UNIVERSITY OF CENTRAL FLORIDA

Remote Area Monitoring

Final Document

EEL 4915L: Senior Design II

Spring 2022

Group 10

Wyatt Vining	**Computer Engineering**
Johan Castillo	**Computer Engineering**
Nicholas Gonzalez	**Computer Engineering**
Abhijeet Malviya	**Computer Engineering**

Table of Contents

1.0– Executive Summary

Our project aimed to reduce the impact of forest fires by providing data to forest management agencies to make informed decisions. While our system is not detecting the forest fires, we have provided information about the environment within the forest. This information consisted of weather data with a specific focus on forest management. During our research, we learned about the current methods for resource allocation and planning behind the prevention of forest fires. Those in charge of managing areas prone to forest fires looked at much of the same data our system intends to supply. The difference is the data currently available to the interested parties is on a macro scale. This meant that the conditions of an area are generalized and averaged. The decisions made rely on historical data and averages that may not be the full picture of the conditions in an area. This had the potential to divert resources for things such as controlled burns and fire breaks away from areas that are in need.

Our system was designed to be placed over a relatively small area of the forest specifically near populated or otherwise at-risk areas. Our system covered the area with a mesh network of nodes providing much more granular data for the managing parties to observe. The goal behind this was to clearly show the health and fire risk of areas buffering populated centers. This may aid in the allocation of resources to these areas. In addition, the data our system gathered was stored to provide a historical record of the area. This may be helpful in forecasting future changes to an area once a large enough sample size of data has been collected.

The research into our competitors revealed a sparsely populated space. Our main competitor Dryad took a similar approach using a mesh network but with a greater focus on detection. While detecting forest fires early was important, equally or of greater importance was preventing them from occurring or spreading in the first place. Beyond this, there were no direct competitors. The current methods for collecting this type of data relied on satellite observation, visual inspections, and sparsely placed weather stations. There were some other methods used by different forest management agencies, though they are not standardized.

We also considered the impacts of our project on the environment. Our system consisted of multiple devices placed throughout a natural environment. We worked to ensure the safety and sustainability of these devices to be remained a priority through the development lifecycle. One way we worked towards this was by creating a modular system to best fit the needs of each individual customer. This helped to cut down on unnecessary electronic waste. We also adhered to applicable standards to increase the compatibility and safety of our system.

As a final note, our system designs was fully open source. Forest fires are devastating and are only going to increase in frequency and severity as our world warms. We want to do our part to preserve these natural areas and make them safer for those living nearby.

2.0 – Project Overview

2.1 Project Narrative

In recent years, we have seen the devastating consequences of global warming. One such consequence was the measurable uptick in wildfires. In 2020 alone over 10 million acres of land burned destroying nearly 18,000 structures. Our motivation originated from one event in 2016, the Chimney Tops 2 fire. One group member was present at the origin of this wildfire. The short of it is, wildfires are terrifying and have a lifelong impact on those affected.

In the days following the start of the fire, we contacted the fire service to offer any assistance we provided from a first-person perspective. The investigator we spoke with had an interesting request. He asked for all the photos we took from the day the fire started. His reasoning was that they needed to determine the fuel load of the area. Fuel load is the amount of combustible materials present. For example, dried brush and dead trees. According to the investigator, this information is hard to ascertain once the fire has passed through an area consuming all the fuel. Being of an engineering mindset, this interaction evoked a line of thinking about how we may help this issue in the future.

We began by examining the options currently available. The options include two main methods of detection both with drastically different approaches. The first method to detect fires is by air. Either from airplanes or satellites. The second method is placing a mesh network of sensors across an area of forest. Both methods have positive and negative features. The fundamental issues with the currently available products that we are aiming to solve is cost and availability. The cost of satellite creation or rental is preventative to many potential users. Additionally, the remaining products are not available for purchase.

We realize that reliably detecting fires is an engineering challenge that may quickly exceed the capacity of two semesters. We are not creating a fire warning system. With consideration of the fire investigator’s insights, we intend to create an open-source monitoring system.

We focused on monitoring the environmental factors leading up to wildfires. Some of these factors include fuel load, moisture, temperature, volatile compounds in air and wind speed/direction. Our system consisted of a resilient mesh network of nodes. Each node reports its data back to an aggregator node where the data is stored and displayed. All nodes are modular allowing only the necessary sensors be equipped for the specific application of the system. The system was to be fault tolerant, maintaining functionality with a minimal number of nodes remaining.

Our hope is that the information gathered by our system aids those responsible for forest management in the prevention of forest fires by prompting preventative actions. As well as informing movement of assets when actively fighting a fire. Finally, and most importantly, placing the mesh near populated areas may aid emergency management in issuing earlier warnings to residents in the event an evacuation is required.

2.2 – Motivation

There have been a noticeable amount of forest fires happening recently and it happens without prior warning. There are various reasons and circumstances of why those fires happen and sometimes these circumstances tend to be underreported because of a lack of proper technology. While knowing about these certain circumstances, we can learn how to properly gather data and analyze them in order to predict the chances of a forest fire from happening.

Therefore, this is the reason why we created the Remote Area Monitoring Device was to gather different types of data such as humidity, temperature, wind speed and direction, and moisture of an area of a specific range so that it could transfer that data to users via a web application. In order for the device to gather these types of data, it would contain cameras, sensors, and anemometers to capture specific nodes. This type of data is then analyzed so that we can then determine the threat of the forest fire of the specific area where it is being measured. When we create our design, we want to make sure that it is efficient and easy to use for any user. It is also important that the product is able to withstand natural damage. The device should be able to work with new technology and software such as Arduino and Python.

2.3 – Goals

2.3.1 – Core Goals

In this section, we listed the goals we believe will lead to project success.

- Created a network of sensing nodes capable of off-grid communications and operation
- Collected the data gathered by each node and aggregate the data in a database. This data may be exported or viewed for a historical representation of the area.
- Each sensing node would adopt a modular design. This is to reduce the overall cost of the system by either: allowing each node to only have the sensors necessary for the application or reducing the number of fully featured nodes in a given monitoring area.
- A minimum set of sensors was implemented including temperature, humidity, smoke, and an anemometer
- All nodes ran on a combination between solar power and a reserve battery for operation at night or in inclement conditions
- The aggregator shall provide a method to visualize the live data. This included a map of each node with a color-coded status
- The system maintained designated persons to receive notifications when a node is no longer communicating with the aggregator

2.3.2 – Stretch Goals

In this section, we listed the nice to have features that may be implement time and resources permitting.

- Predictive data analysis – Use gathered data in conjunction with outside data to offer suggestions to area maintainers. For example, when the system notices an area is dry and there is no rain predicted in the next few days, the system may recommend the fire danger level is increased
- Fault tolerance in nodes – If a node has a degraded component the system shall automatically shut that component down and continue to function with remaining components where possible
- Preventative Maintenance – The system detected a degradation in sensors or batteries of the nodes and suggest preventative actions prior to failure
- Daily or weekly reports with a management friendly layout – Focused on graphics and clear visual representation of the data marked most important by the maintaining agency
- Expanded the capacity of the network – Allowed for covering much larger areas by increasing the number of nodes the system supports

2.4 – Requirement Specifications

2.4.1 – Hardware Requirement Specifications

When we selected our hardware components, we kept various specifications in mind. We had to make sure that our Remote Area Monitoring Device was able to properly function in the area where it was being tested given the constraints and that it produced the most accurate results and data possible. Therefore, the hardware components that we selected must follow a specific accuracy range in order to give us the best value when measuring nodes such as the humidity, temperature, carbon monoxide level, rain level, and soil moisture. The device itself should be large enough at about 6in x 6in so that it can be able to manage and use different types of nodes properly. Also, the device was able to measure and analyze forest fire threats at a certain distance and with a reliable camera or viewing device so that it would give more time and space to report data about it while being able to detect it in a much wider area and distance. It would also need to accurately analyze the type of threat that is coming from the area that is being investigated. The device should also be able to be powered from a reasonable amount of time since it is going to be placed outside so that means that the battery of power reserve should be able to stay fully charged for at least 3 days. The components of the device should be strong enough to withstand circumstances such as rain or high amounts of sunlight and should also be disposable, recyclable, and not that expensive so that it can be affordable for the user who buys that product.

Table 2.4.1.1 – Hardware Requirement Specifications

Requirement ID	Specification	Value
001	Range	400 Meters Minimum
002	Visual Camera	2MP Sensor SPI Interface
003	Infrared Camera	2MP Sensor SPI Interface
004	Power Reserve	3 Days on a Fully Charged Battery
005	Charging	3 Watt Minimum PV Panel – Wall charging option
006	Physical Dimensions	6in x 6in
007	Wind Speed and Direction	+- 5% Accuracy
008	Humidity Sensor	Accuracy +- 3%
009	Temperature Sensor	Accuracy +- 0.5C
010	Barometer	Accuracy: +-5kpa
011	TVOC Sensor	Detects 1000ppm change in TVOCs
012	Carbon Monoxide Sensor	Detects 1000ppm change in CO
013	GPS Module	15m Radial Accuracy
014	Soil Moisture Sensor	Detects 1 Drop of Conductive Liquid
015	Rain Level Sensor	0.2794mm per step

2.4.2 – Software Requirement Specifications

For our device, we included software components as well. The reason for that was in order to send data that we obtained from the nodes that we are measuring with our Remote Area Monitoring Device; we used software that presented those values in a web application. When we selected our software components, we had to make sure that the components were up-to-date, compatible, and were able to store high amounts of information. The way that the data is going to be presented on the web application is by maps, of-grid networks, and databases. With these methods, it can help the user know which specific areas of the forest have a higher risk of forest fires happening while basing from and including other nodes such as temperature. The software was able to communicate and send data without relying on cellular networks and that it was able to store a reasonable amount of data.

Table 2.4.2.1 – Software Requirements Specifications

Requirement ID	Specification	Description
016	Application	Web application to display real time and historical data
017	Map View	Show each node on a map with the status of the node
018	Map Overlay	Display a gradient on the map representing the environmental conditions such as temperature
019	Off-Grid Network	Network for communicating to and from the nodes without relying on established networks such as cellular
020	Database	Store a minimum of 5 years of sensor data
021	Number of Nodes	200

2.5 – Prototype Diagrams

2.5.1 – Node Hardware Block Diagram

Diagram 2.5.1.1 outlines the core systems of a node in the network. These nodes are responsible for capturing and relaying sensor data.

Diagram 2.5.1.1 – Node Hardware Block Diagram

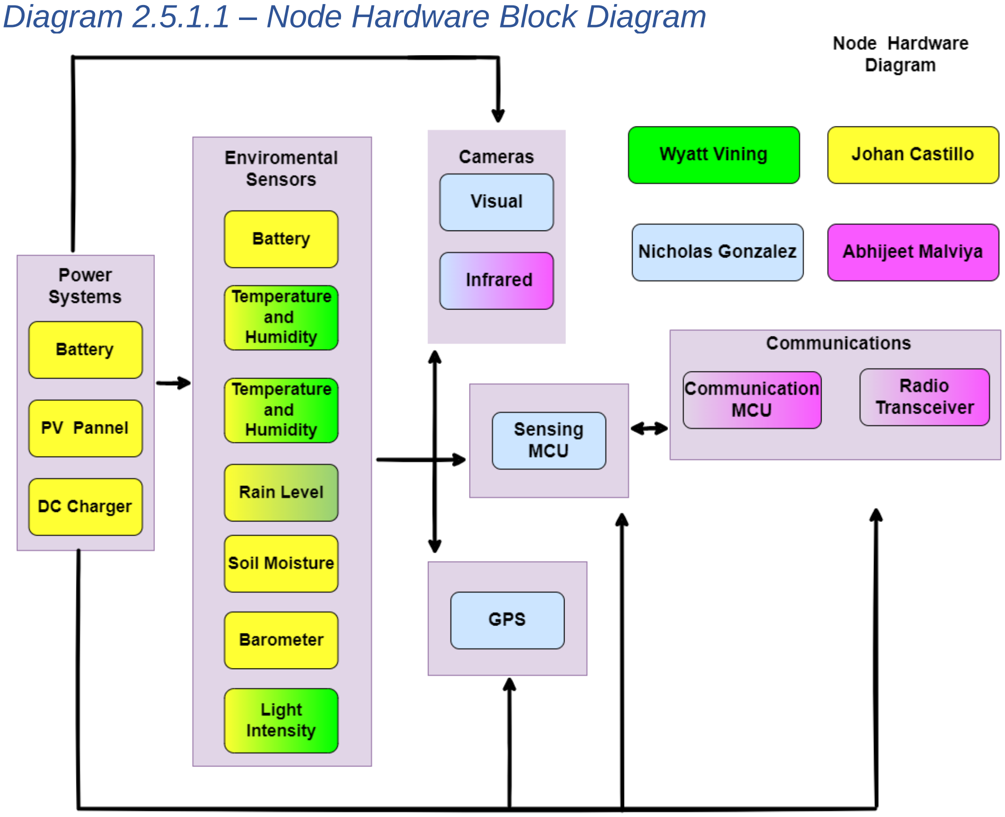
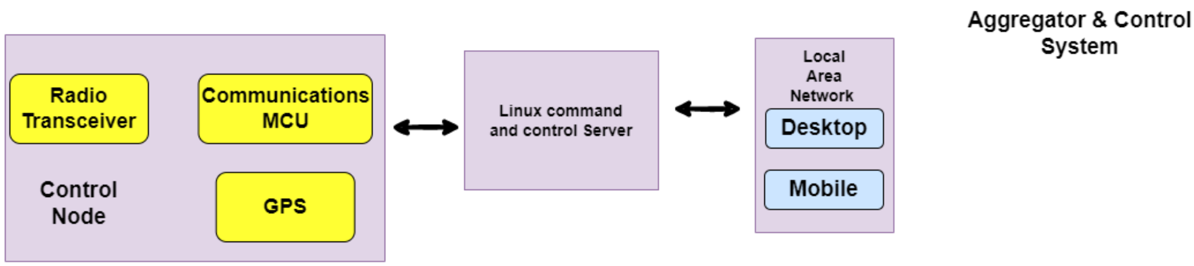


Diagram 2.5.1.2 also describes a node in the network. This is a special node responsible for receiving the data from the wider network and send commands out to all nodes on the network.

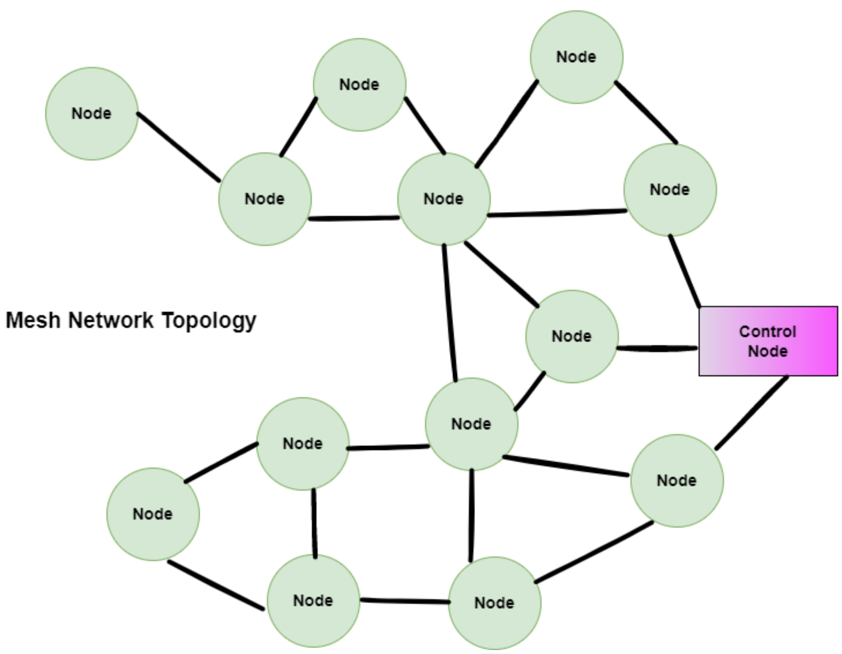
Diagram 2.5.1.2 – Control Node Hardware Block Diagram



2.5.2 – Mesh Network Topology

Diagram 2.5.1.1 describes the network topology. All nodes in the mesh communicated to one another with each having a unique address. The control node listened for data to pass to the aggregator.

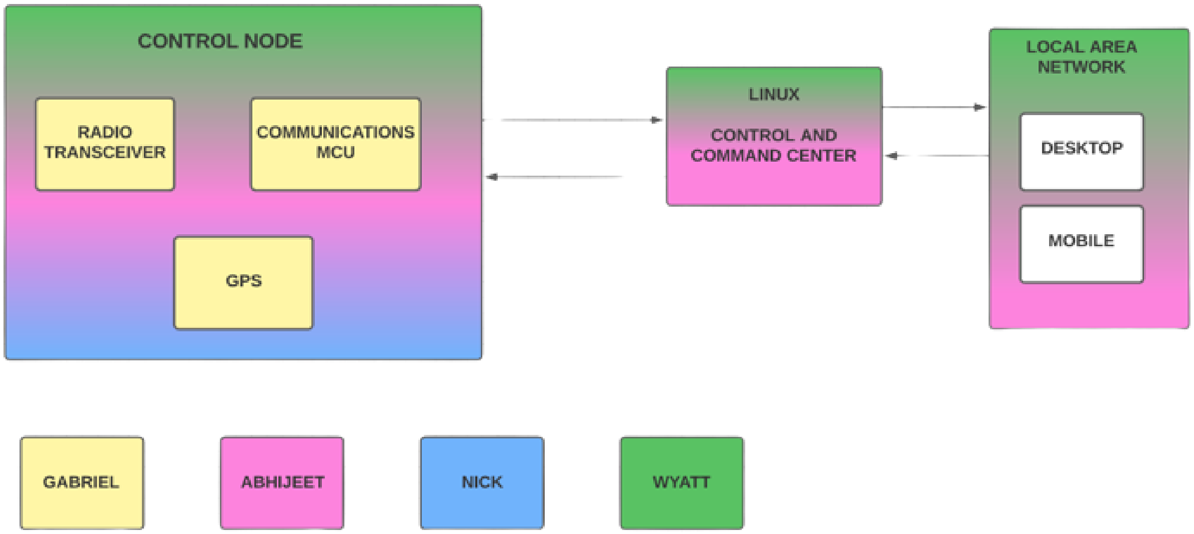
Diagram 2.5.1.1 – Mesh Network Topology



2.5.3 – Command and Control System Block Diagram

Diagram 2.5.3.1 shows the connection between the control node and the server. The server collected the data and passed commands for the network to the control node when needed. The server was also responsible for hosting the user interface.

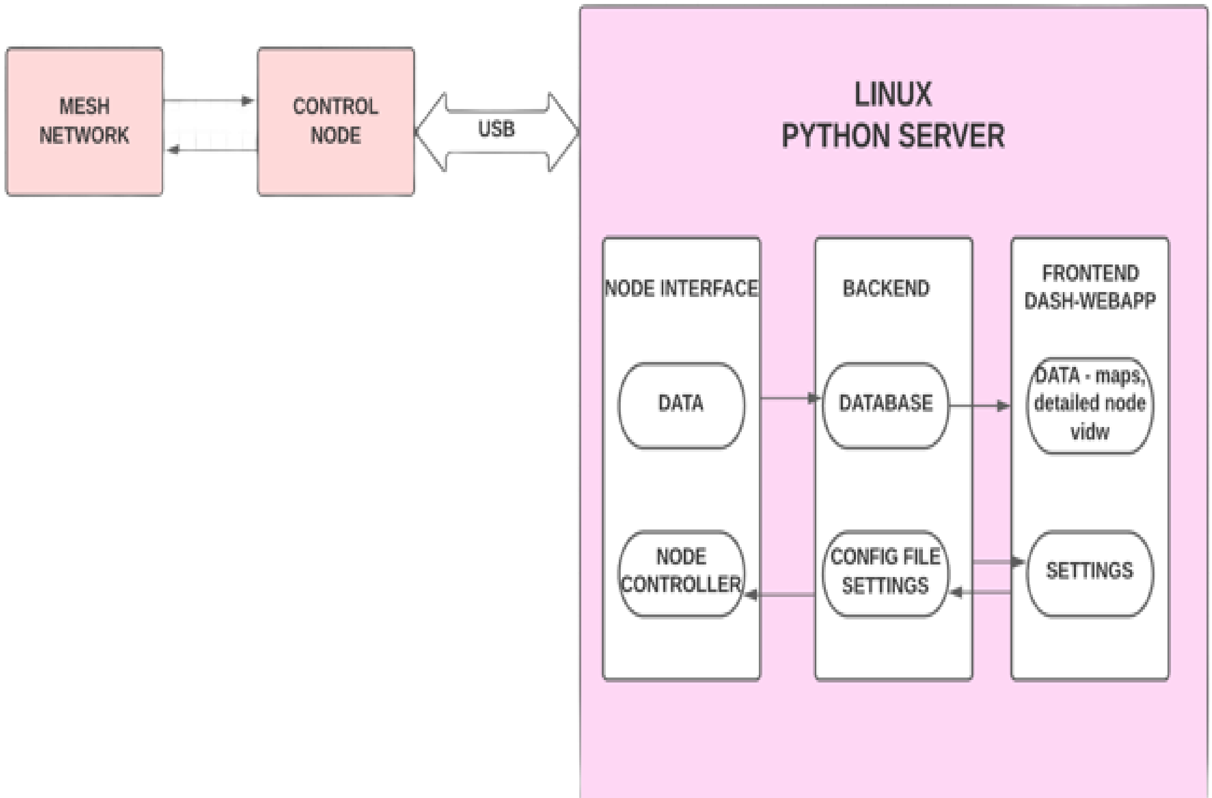
Diagram 2.5.3.1 – Command and Control Software Block Diagram



2.5.4 – Web Server Block Diagram

In this diagram, it showed how the information from the mesh network was transferred to the control node and back to it again. The web server and the control node were responsible for most of the transfer of data in the software. The control node was basically an antenna that was then connected to a USB cable so that it sends that data and other commands from there to the Linux Python Server. Within the Linux Server, it contains three parts called the Node Interface, Backend, and the Frontend Dash-Webapp. The Linux server is responsible for mostly every other action that is happening on the software. In the Node Interface, it contains the data which then stores that information to the database on the backend which it then sends to specific parts of the webapp such as the maps. The database holds collections that contain different data objects in which Python then helps to format them correctly. That interface also sees whether there is garbage from the Mesh Network and the Control Node so that it can take that out and send it to the backend and then to the webapp, which portrays those results visually. The settings on the software can let the user perform certain modifications to the web application such as the type of notifications it receives like updates from the web application. It does this by having the Python backend read the config files so it knows what to do with the settings of the web application. The settings in the frontend then interchanges data with the Configure File Settings from the backend which eventually transfers that information to the Node Controller in the interface. The Linux Server sends back commands to the Control Node so that it knows what exactly to do in the software.

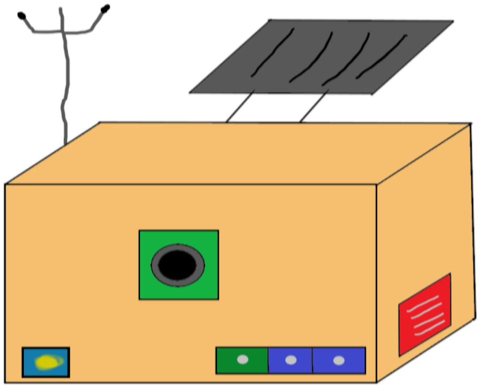
Diagram 2.5.4.1 – Web Server Block Diagram



2.5.5 – Exterior Design Concept

This is how we plan to have the Remote Area Monitoring Device look on the outside. We planned to have all the circuits and components placed inside a cardboard box. Around the box, there would be small holes where we would place the cameras in order to view what is happening outside the environment. It also contains other holes where it would contain other components from the device that would measure specific nodes such as humidity, temperature, rain, and soil moisture. We plan to place the solar panels on top of the box so that it can properly receive power from the sunlight. There would also be an anemometer on top of the box as well in order to measure the wind speed and direction. All of these nodes would help us know the chances of a forest fire happening in the area being measured and the severity of it.

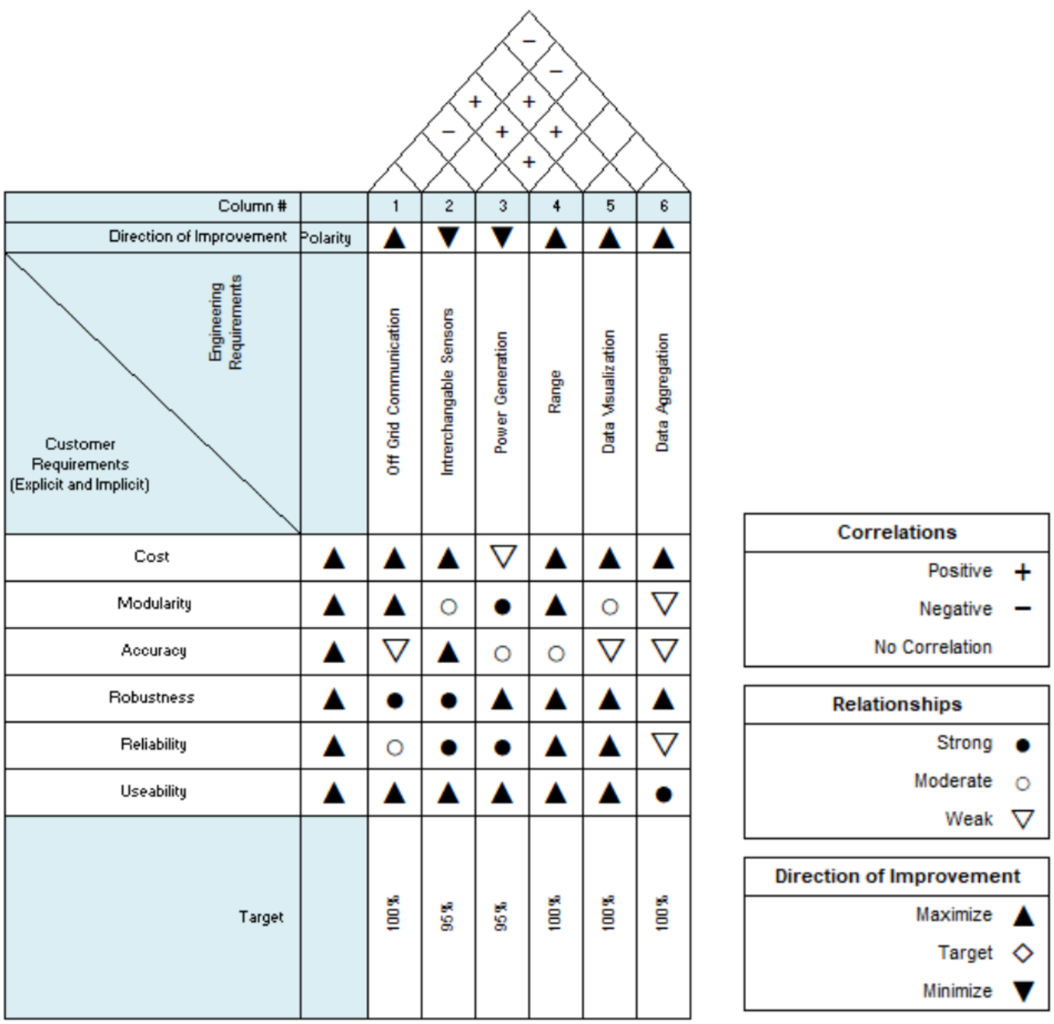
Diagram 2.5.5.1 – Node Exterior Design Concept



2.6 – House of Quality

Our house of quality diagram (Diagram 2.6.0.1) shows our inclination to maximize the value to our customer while minimizing the cost. One of the ways we intend to accomplish this is by using off the shelf parts and making our designs available open-source.

Diagram 2.6.0.1 – House of Quality Diagram



3.0 – Market Research

3.1 – Similar Products

3.1.1 – Dryad

Forest fires are catastrophic not only for the life and environments they devastate, but also because of the lengthy harm they create by emitting large amounts of CO_2 while also killing vegetation that absorbs carbon. The United States is experiencing some of the worst wildfire season in decades, while Australia's 2019-2020 wildfire period was the second worst in history, including over 46,050,750 acres burned. Following unprecedented temperatures, western areas of the United States and Canada are now prepared for the yearly wildfire outbreak, which some experts say may set bleak new marks.

Dryad Silvanet employs solar-powered gas sensors in a large-scale IoT sensor network to detect wildfires as early as possible and to monitor forest health and growth. Dryad aims to avoid undesired wildfires, which generate up to 20% of global CO_2 emissions and have a significant impact on biodiversity. By 2030, we want to have avoided the burning of one million hectares of forest, saving 400 million tons of CO_2.With a large-scale, dispersed sensor network with a central analytics and alerting platform for ultra-early wildfire detection and continuous forest health monitoring. The solar-powered wireless sensors detect forest fires using embedded AI and measure temperature, humidity, and air pressure. Silvanet, which is open to third-party sensors, provides a wireless, solar-powered mesh network architecture for a variety of IoT applications in the forest. Silvanet additionally provides a Cloud Platform for device administration, data analytics, and alerts.

Most options for detecting wildfires rely on optical devices such as cameras or satellites. However, optical techniques might take hours to identify wildfires if they are obstructed by forest canopy or clouds. amera and satellite-based systems, often take many hours or even days to identify a fire since they rely on the smoke plume expanding sufficiently to be noticed from a great distance. This makes it significantly more difficult, if not impossible, for firefighters to put out the fire. Furthermore, developing solutions based on the Narrowband-Internet of Things (NB-IoT) standard are unsuitable for large-scale and isolated woods where the cost of constructing an LTE/4G network is too expensive.

Dryad's Silvanet is focused on identifying wildfires as early as 60 minutes using solar-powered gas sensors put in the forest, detecting wildfires as early as 60 minutes using built-in machine-learning (ML), analyzing gas patterns to consistently detect a fire. Silvanet provides a long-range wireless network (LoRa) that has been enhanced with a patent-pending mesh network design to cover very broad distances because the system cannot rely on mobile network service in the forest. The Silvanet, designed as a general-purpose IoT network infrastructure, can be used by any LoRa compliant 3rd party sensor, enabling health and growth-monitoring applications such as soil-moisture, sap-flow, or tree-growth monitoring, and feeding valuable data into the central Silvanet Cloud Platform, which provides data analytics and alerting services to Dryad's customers.

The Silvanet Mesh Gateway expands the Silvanet Network's reach beyond the conventional single-hop direct connection from sensors and gateways in normal

LoRaWAN networks to massive implementations. The design employs a multi-hop mesh network of Gateways connected using LoRa, with each Gateway acting as a conventional LoRaWAN gateway to Silvanet Wildfire Sensors and third-party sensors. The photovoltaic Powered Mesh Gateways are installed inside the forest, producing a mesh network with an average radius of 2-6 km relying on topological and physical positioning of the Mesh Gateways. The Mesh Gateway only communicates with other Mesh Gateways or a Border Gateway through the LoRa radio. It does not require a direct 4G/LTE radio or Ethernet connection, resulting in low power usage, which is backed by the integrated solar panel.

Figure 3.1.1.1 – Dryad Silvanet Mesh Gateway

Some of features could be futher implemented of the mesh gateway are, Improve LoRaWAN networks to cover a larger region Solar-powered, no-maintenance Any LoRaWAN-compliant sensor can be used. Network setup that is done automatically Mesh network with self-healing and auto fail-over Over-the-Air Firmware Update (FUOTA)

Figure 3.1.1.2 – Dryad Mesh Network

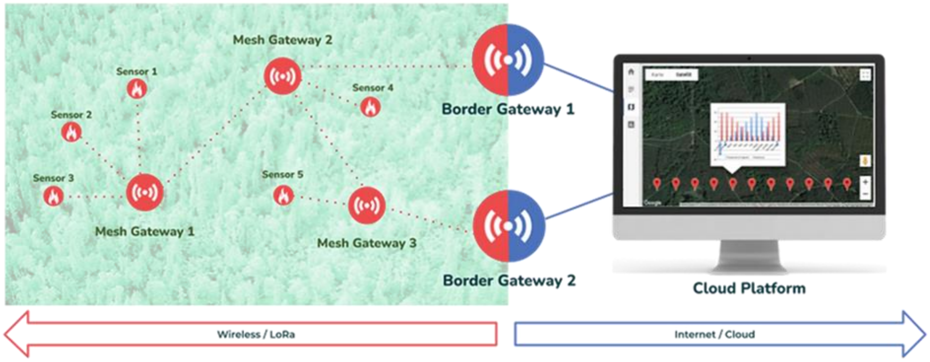


Table 3.1.1.1 – Dryad General Specifications

Maintenance	Maintenance-free (10-15 yr)
Mesh Gateway to Border Gateway (Ratio)	Typically, 20 Mesh Gateways are required for each Border Gateway.
maximum distance between Mesh Gateways.	2-6 kilometers, depending on topography and Gateway placement
Mesh Gateway to Sensor (Ratio)	Typically, 100 sensors per Gateway, depending on topology
Max distance between Mesh Gateways	2-6 km, depending on topology and placement of Gateways
Power source	Solar-powered, battery-free
Energy Storage	Supercapacitors
Installation	Mounted on pole or attached to a tree

SILVANET WILDFIRE SENSOR

Time is of the importance when it comes to wildfires. Silvanet detects smoke, hydrogen, and other gases released by pyrolysis in the beginning phases of a wildfire, providing firefighters critical time and the opportunity to smother the fire before it gets out of control.

The Silvanet Wildfire Sensor is intended to identify forest fires in their early phases (even during the smoking phase, during the first 60 min) and to manage the microclimate by detecting temperature, humidity, and air pressure. The sensor combines ultra-low-power Air Quality detection with a very accurate gas sensing mode. It identifies Hydrogen, Carbon Monoxide, Carbon Dioxide, and other gases at the 0.0001% or parts per million level using built-in machine learning to reliably identify a fire and eliminate false positives. The sensor transmits data wirelessly via LoRaWAN communications and may operate without batteries for 10-15 years, reducing the usage of lithium and other harmful elements.

Figure 3.1.1.3 – Dryad Silvanet Wildlife Sensor

Table 3.1.1.2 – General Specifications Dryad Silvanet Wildlife Sensor

Maintenance	Maintenance-free (10-15 yr)
Mesh Gateway to Border Gateway (Ratio)	Normally, 20 Mesh Gateways are required for each Border Gateway.
maximum distance between Mesh Gateways.	100m radius for 60min detection of 2x2m fire
Mesh Gateway to Sensor (Ratio)	Typically, 100 sensors per Gateway, depending on topology
Power source	Solar-powered, battery-free
Energy Storage	Supercapacitors
Installation	Mounted tree (3m height recommended)

SILVANET BORDER GATEWAY

The Silvanet Border Gateway is often installed towards the edge of the target forest area, such as in a forest home or near a settlement. The Silvanet Cloud Platform interfaces with the Border Gateway, which relays messages from Forest fire Sensors (directly or indirectly via Mesh Gateways). Wireless connectivity is offered via the installed LTE radio (using 4G/LTE-M with 2G/GPRS fallback) or via a wired Internet connection via the built-in Ethernet adapter. The Silva net Border Gateway features built-in functionality for satellite uplink utilizing the SWARM satellite system for distant locations in which there is no mobile data service and no accessibility to mains power. It may be used with either mains electricity or a solar cell.

Figure 3.1.1.4 – Dryad Border Gateway

Table 3.1.1.3 – Dryad Border Gateway General Specifications

Maintenance	Maintenance-free (10-15 yr)
Mesh Gateway to Border Gateway (Ratio)	Normally, 20 Mesh Gateways are required for each Border Gateway.
maximum distance between Mesh Gateways.	2-6 kilometers, depending on topography and Gateway placement
Mesh Gateway to Sensor (Ratio)	Typically, 100 sensors per Gateway, depending on topology
Max distance between Mesh Gateways	2-6 km, depending on topology and placement of Gateways
Power source	Mains powered (PoE) or Solar-powered, battery-free
Energy Storage	Supercapacitors
Installation	Mounted on pole or attached to a tree

3.2 – Area Management Policies and Programs

A fire spreads through radiation when the heat travelled through electromagnetic waves in the air. Fire spreads are so dangerous and can cause so much damage so quickly to open lands. A fire can literally occur anywhere at any time, so you always have to be mindful of your surroundings and what you need to do to avoid it. So, this is why it is imperative to understand the guidelines that safety hazards and officials issue. NPS or National Park Service is one such organization that tries to main their environment and ensure that areas are safe from fire hazards.

NPS works with interagency partners in an effort to prevent unwanted dangerous human-caused hazardous fires. We humans, who have been existing on this planet from ever since the dinosaurs were destroyed by unfortunate and unforeseen circumstances, are responsible for causing approximately 90% of the wildfire in open lands. The key component to stop these scenarios is education and outreach. There are other procedures and precautions taken by park managers as well, including implementing the fire restrictions during times of very high or extreme fire dangers. So, how exactly does a wildfire start? Common causes of wildfire include lightning, human carelessness, arson, volcano eruption, pyroclastic cloud from active volcano. Heat waves, droughts, and cyclical climate changes such as El Nino can also have a dramatic effect on the risk of wildfires. But, according to weatherwizkids, four out of five wildfires are caused by people.

3.2.1 – Environmental Impacts

Fires are adverse and unpredictable events with tangible costs for property and human life. The immediate and direct costs of the fire provide a metric for understanding the economic impact it has on the environment. In addition to the physical costs, fires have a range of less immediate and obvious adverse consequences on the natural environment. These include air contamination from the fire plume (whose deposition is likely to subsequently include land and water contamination). The amount of gas which contains a lot of toxic chemicals mixes with the oxygen in the air which harms all living species that have been existing on this planet for 4.65 billion years. Fire is an important element that is used by a lot of us for different day to day lives, but we don’t understand the impact it has on the environment if we use it incorrectly. There are many beautiful national forests in this country which have been harmed because of the carelessness of people in general.

So, we should understand the fire effects on this environment. It is evident that wildfire is part of our nature. It is sporadic and can cause havoc with anything in its path. Scientists with the Pacific Northwest Research Station are conducting a range of studies pertaining to fire effects on the environment in multiple fields of study, from meteorology to ecology. The range of studies include: Fire Behavior, Fire and Climate Change, Fire Ecology, Fire Preparedness, Fire and Smoke, Fuel Treatments, Fire and Weather.

A fire behavior is heavily depicted by the way the fire’s intensity and the rate of the speed that it is moving at. Scientists who study fire behaviors are keened in factors that influence the fire intensity and the speed of fire such as fuel types, weather and topography.

We already know how the climate change is in full effect at the moment. It has been stated that the change in temperature and weather has already occurred in various regions of this world. The average temperature is increasing slightly per year. Places that are warm are getting warmer while cold places don't see snow anymore. Climate change is in effect and the spread of wildfire is another contributing factor to this. Antarctica is melting slowly. This is because the spread of wildfire mixes with the atmosphere and disrupts the ecosystem. All the toxic particles move through the air towards various locations which alters the ecosystem. As this becomes a phenomenon in the world, farmers won't have much food to grow, aquatic animals won't have a place to live in, etc.

3.3 – Current Standards and Equipment

In this project we faced various different types of standards and constraints. These aspects were necessary since they set guidelines on how we built and implemented our forest fire detection alarm system properly to limit creating errors and to help us determine the scale and limitations of our design. Standards are crucial due to the fact that it helped us to make sure that the device is safe and efficient. There were many different types of standards that we followed based on the project or device being created, so we chose which ones should be the most appropriate for our design, how we planned to deliver it to the customers, and whether or not it was safe in the environment where it was used in.

Constraints were set up to determine the limitations of a design. They were necessary because it helped us focus on more specific aspects of the project, let us know whether or not extra nodes or features were needed to be created and if we're not wasting too much money on resources. This is one of the reasons why we set up a reasonable budget for our project.

3.3.1 – Standards

Engineering standards were used to set up rules and guidelines of how to build, implement, and test a design in order to make sure that it is functioning properly and efficiently with regards to economics and the health and safety of the users and the environment. Understanding and knowing which standards were going to be used is an important aspect in an engineering design project, have we decided to not follow standards, it could have caused difficulty and confusion in building the project and there can be risks of having it malfunction.

Depending on the part or product that was manufactured, there were different types of standards that needed to be implemented. For this project, we followed the IEEE, PCB, ANSI, and Python language standards. The IEEE standards were necessary since they were used to test the functionality of the design such as the communication between the hardware and software aspects of the project. The PCB and ANSI standards were used to build and manufacture the hardware aspects such as the circuits which control the alarm system. We followed local and federal environmental ordinances since our product was used in outdoor areas such as forests.

3.3.2 – Fire Danger Rating System

Forest fires can happen at any moment and how dangerous it can end up being depends on certain circumstances. There is a system put in place called the National Fire Danger Rating System (NFDRS) and that was created so that there are more efficient and scientifically-based standards to detect forest fires which can be applied anywhere and is cost-effective. There are various organizations that use these standards to control fires around their respective areas. These standards helped us determine the possible locations where these fires might occur and the previous activities within those locations.

To determine the severity of fires, NFDRS takes the fuels, weather, topography and risks into account to determine the type of rating it should give in a selected area. While looking at those aspects, NFDRS can finally give a Fire Danger level ranging from Low to Moderate, High, Very High, and Extreme. When the threat is Low, it means that you need to have extremely flammable heat sources to cause fires and it can be easily controlled. When it's moderate, it means that fires can easily be caused in that area by man-made accidents but it would spread really slowly or not at all. When it's High, it means that the fires can be dangerous but it can be put out if it's still small. When it's Very High, it means that any flammable object can cause a severe fire that is hard to stop; and when it's Extreme, it means that the fires can last several days and it's very hard to extinguish.

While we cooperated with different weather stations, they helped us transfer information and guidelines that we used to analyze different types of inclement weather that can cause forest fires. One of these methods was by determining how prevalent lightning strikes are in an area since when they strike a dry object, they can produce flame due to the amount of heat they produce. When we determined the lightning activity level, it was rated in six levels. The lowest level (LAL 1) means that there is no cumulus cloud in that area, which means there isn't a possibility of lightning-producing thunderstorms to happen while when it is at LAL 2, it means that some cumulus clouds are present but the threat is still low. When it’s at LAL 3, cumulus clouds are common and some lightning may happen. At LAL 4, lightning is way more frequent and thunderstorms cover 30% of the sky. When it's at LAL 5, lightning and thunderstorms are very intense and in LAL 6 is where there’s a “dry lightning” which means that precipitation evaporates before it reaches the ground in a lightning storm. When that happens, it increases the likelihood of a fire happening since there is less water and moisture to stop it.

NFDRS is also composed of three parts which are Scientific Basis, User Controlled Site Descriptors, and Data. To determine how threatening fires can be, scientific models that take environmental aspects such as moisture into account to see if the chances of starting a forest fire are high. User Controlled Site Descriptors take account of the area where it's being researched and the plants around it to see how flammable they can be. It also looks at the area’s terrain and climate such as how much rainfall it receives as well. Then, Data is gathered from the controlled site descriptors to calculate the appropriate rating for a location such as the type of vegetation in the area and rain quantity.

When we gathered data, there were two variables that we needed to focus on to determine the fire danger rating: weather observations and other parameters that user finds to calculate the rating such as the vegetation state or the shrub type code. When we gathered weather data, we made sure that the data was recent or able to predict the weather for the next day, and that was gathered via weather stations. When a user was going by their own parameters, he should have noticed that changes can happen in the environment that's being researched such as trees freezing or drying up or the fuels that it may leave out.

Seasonal changes affected how information regarding about fire prevention was obtained. So therefore, we focused and analyzed the different seasons and the greenness factor, which represent the conditions of the environment.

Table 3.3.2.1 – Season Codes and Greenness Factor

Growth Phase	Season	Season Code	Greenness Factor
Plants are dormant	Winter	1	0
Plants begin to grow again	Spring	2	Increase the factor to 20 for each plant
Plant growth is complete	Summer	3	Change the factor based on each plant's growth
Plants are dormant	Fall	4	Decrease the factor to 0 for each plant

While we abided by the NFDRS standards, there were certain equations that we used to perform calculations to help determine how flammable a tree or plant was within the area where it was being tested. One of those equations is the Equilibrium Moisture Content formula. There are three main variations of using that formula:

Equation 3.3.2.1 – Relative Humidity Less Than 10 Percent

$$EMC = 0.03229 + 0.281073 \times RH - 0.000578 \times TEMP \times RH$$

Equation 3.3.2.2 – Relative Humidity Equal to or Greater Than 10 Percent but Less Than 50 Percent

$$EMC = 2.22749 + 0.160107 \times RH - 0.014784 \times TEMP$$

Equation 3.3.3.3 – Relative Humidity Equal to or Greater Than 50 Percent

$$EMC = 21.0606 + 0.005565 \times RH \times 2 - 0.00035 \times RH \times TEMP - 0.483199 \times RH$$

In these equations above, RH represents the relative humidity and TEMP represents the dry bulb temperature, which is the air temperature measured from the environment being tested. While we got the EMC values, they were used to get the observation time, maximum temperature, minimum relative humidity time, and minimum temperature-maximum relative humidity time.

Also, while we calculated the relative humidity, it was important to take the Dry and Wet bulb temperature for the moisture variable into account. To calculate these variables, we had to get the saturation vapor pressure values. Then to get the elevation of the area being tested, we calculated the CORR value.

Equation 3.3.3.4 – CORR Value Calculations

$$SATVPD = EXP\left(1.81 + (TMPOBS \times 17.27 - 4717.31)/(TMPOBS - 35.86)\right)$$

$$SATVPW = EXP\left(1.81 + (TMPWET \times 17.27 - 4717.31)/(TMPWET - 35.86)\right)$$

CORR=6.6*10**-4.*(1.0+(0.00115*(TMPWET-273.16))*TMPOBS TMPWET)*(1013.09/EXP(ELEV/25,000.0))

In these equations above, SATVPS and SATVPW represent the saturation vapor pressure for dry bulb and wet bulb temperature respectively. EXP represents the exponential function of the equation that is selected by the parentheses. TMPOBS represent the dry bulb temperature and TMPWET is the wet bulb temperature. CORR represents the difference between SATVPW and ambient vapor pressure and ELEV is the elevation.

3.3.3 – Natural Resource Management

In our project, we followed federal ordinances such as the National Park Service's Natural Resource Management standards. These guidelines explained how a researched area might be affected by natural phenomena and how to properly take action and used resources efficiently while also cooperating with other similar standards and laws created as well. There are various reasons for why such a phenomenon might happen and it can cause changes to the environment that is being tested, so therefore we saw how certain aspects of the forest such as plants and trees can be restored and whether or not some parts should be replaced or moved to a different location. These standards also focus on the non-living aspects of an environment as well. By looking at this, we saw what can we do when a fire is being detected with our alarm system.

Also, these standards state that in case that something wrong happens in the environment that we are testing our design such as getting injured or having resources get damaged, then that management service can cooperate with other authorities in order to provide compensation and new resources to take care of the damage being done. It also analyzed different areas within the experiment such as the cost, scientific research from other colleges and organizations, park ordinances, and safety in order to find the best way to approach the experiment. For these standards to work, we analyzed the products that we bought for our design and made sure that it doesn't cause that much damage to the environment and whether it's economical. When we researched the tested environment, we also had to make sure that soil is sustainable which helps with determining climate regulation.

3.3.4 – NFPA 1143 Standard for Wildland Fire Management

A standard that we followed would be the NFPA 1143 Standard for Wildland Fire Management. These standards told us how to deal and manage forest fires and by working with other resources and organizations and provided different types of training and equipment. It also tells us what we should do to prevent those fires from happening again in our area, such as coming up with plans beforehand based on analyzing fire seasons and patterns over the years. NFPA 1143 took into account many different areas such as fire department apparatus and divisions and also works with other standards such as the Standard on Emergency Services Incident Management System and the National Fire Danger Rating System.

While we worked with NFPA 1143, it also recommended that we cooperated with local fire departments using mandates from the respective local government to combat these disasters. This standard advised that in a case where you are working with any instruments or laboratories that hasn't been officially sanctioned yet for safety, NFPA doesn't bear the responsibility to approve it. Instead, we should always abide by the federal or state mandates or fire departments to have jurisdiction for these fields. It also required people who are working within those fields to have proper training in first aid and safety and be familiar to agencies that require those guidelines. Programs were established to help people to train and get further knowledge about those fires.

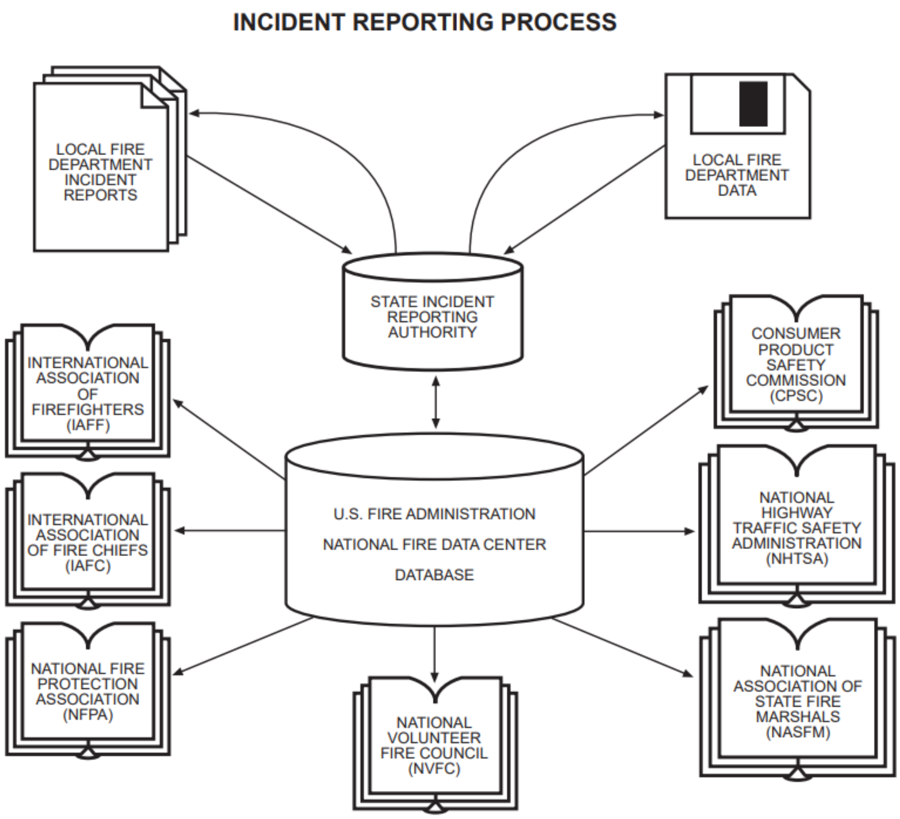
The variables that NFPA 1143 looks at to prevent forest fires include Risks, Hazards, and Values. By taking risks into account, it looks at how fast fire can spread and it does that by comparing that separately to how much fire can burn. It also determines whether or not the fires were man-made or caused by natural disasters such as lightning storms and by how recently have they occurred. When it looks at hazards, it looks at what can cause such fires to happen and how hard it can be to control it. Finally, when it looks at the values, it analyzes what areas of a forest should be taken in the highest consideration to avoid it being burned and how to protect it. After that, we determined how to implement fire protection programs and prioritize that specific place compared to other areas.

In order to stop and combat those fires, there are three variables that we learned: Education, Engineering, and Enforcement. Knowledge should be spread as much as possible informing people about fire prevention. Proper engineering was used to stop those fires from happening by removing fuel or other combustible material in a forest. When we cooperated with law enforcement, they found an inexpensive way to administer laws on forests that restrict fire activity and have close contact with fire departments in order to stop those fires from happening immediately. It was recommended that in each area where the environment is being researched to have at least five well trained firefighters in case of an emergency happening. Fire chiefs and law enforcement are the first types of people that we contacted in those situations and they planned and improved communication tactics and relief crews.

3.3.5 – NFPA 1143 Standard for Wildland Fire Management

NFIRS is a type of fire reporting system that responds to incidents and also analyzes the type of fire disaster that is happening. It allowed us to see and use the type of resources to combat fires. The way how the incidents were reported was by gathering data and incident reports from local fire departments and interchanging them with the state incident reporting authority. Then, the authority passed over that information to the U.S Fire Administration National Fire Data Center Database which it then sends to different governmental agencies such as the IAFF, IAFC, NFPA, NVFC, NASFM, NHTSA, and CPSC.

Figure 3.3.5.1 – Incident Reporting Process Diagram [2]



When we identified the type of incident happening regarding a fire, it required us to select a range of codes to describe it. Some examples that are used are 100 which means that there is a fire happening, 200 means that there is an explosion or overheat but no fire, 300 means rescue and emergency medical service, 700 means that there is a false alarm, 800 means that there is a severe or inclement weather, and 900 means that there is a special incident. When a fire is happening, the level of severity is determined as numbers from a range of 1 to 5, with 1 being the less severe and 5 being the most severe. A level of U is given when the fire threat is undetermined.

Not only does it determine the severity of fires, but it also helped identify causes of why it happens. One of the reasons is because certain fuels are found in important aspects in forests such as tree or plants and they can be flammable. Therefore, NFIRS utilizes a data element called Fuel Model at Origin which analyzes these types of vegetative fuels that appear after a plant is being burned at the point of origin and predicts how dangerous a forest fire can be. This model is divided into different subsections that describe that type of vegetation that is being affected such as Fuel Model A representing annual grasses, Fuel Model B representing mature 6 feet tall brushes, and Fuel Model C representing open pine with grass. Fuel moisture can be classified to different time lag classes such as 1 hour, which represents fuel from dried plants with a diameter less than a quarter inch; 10 hour, which are dried wood with a ¾ to 1 inch diameter; 100 hour, which are plants with a 1-to-3-inch diameter and tend to be the most common one; and 1000 hour, which are plants with a 3-to-6-inch diameter. The fuel moisture levels can also be seen on greenness maps made by the Normalized Difference Vegetation Index data.

3.3.6 – Humidity Standards

Humidity played an important role in detecting and preventing forest fires from happening. A type of humidity that was important to measure of ecosystems where it is being examined is relative humidity, which measures the amount of moisture there is in the air at the same temperature and pressure. When the air is warm, it means that it is more humid since it contains more moisture. Relative humidity is measured by weather stations or by using wet and dry bulb readings. By having more humid air, it meant that the chances of having a forest fire would be lesser because since it contains water, it prevents fire from spreading. If relative humidity is low, then forest fires become more likely to happen because the fuels released from forest vegetation become drier.

3.3.7 – NWGC Standards for Fire Weather Stations

NWCG is a group that was formed to promote training and standards that other firefighting agencies can apply with to prevent forest fires. They also promote standards that exist for fire weather stations and they also work with other standards such as NFDRS since we can cooperate with weather stations on assigning ratings to forests to analyze the threat of those fires.

3.3.8 – Fire Watch Towers

One way of how we detected forest fires from far away is by looking from Fire Watch Towers. They were usually placed in high areas within the location where the fires and smoke are being detected to get a much clearer view of them, since if someone were to view it at ground level, they might not get a clear outlook since it would be covered by trees and other types of terrain. These towers are meant to view areas from a 20-mile radius. While spotting fires over, we planned a way to have some sort of communication to send information to other places. Therefore, fire watch towers contained radios to transfer data. By viewing a selected area over a high elevation, it was easier to locate and map the affected area and pass that information to other sources via satellite so that there can be an accurate view of the situation and what places to avoid.

3.3.9 – Satellite Imagery

Not only can an environment be tested for possible forest fires up close, but it was also detected from different faraway places by the use of satellite imagery. Satellite energy helped us map different parts depending on whether the threat is high by assigning different colors to those areas depending on the threat level. With that data, it is then sent to different users with that software. However, satellites can be expensive to use so we have found more economic ways to detect fires.

Many maps required the use of satellite imagery to properly view an area being researched and what precautions we can take to prevent such incidents. We viewed the fuel moisture levels maps through greenness maps made by the Normalized Difference Vegetation Index data. From these maps, we saw what areas had the healthiest vegetation since it was shaded green.

Other types of maps from satellite imagery that we used is the Keetch-Byram Drought Index (KBDI) which detected which places have been suffered from drought. Drought plays an important part in preventing forest fires since we can tell that if a forest hasn't been receiving too much rain, it might dry up and since there was little to no moisture, then it was harder to control flames. By viewing those maps, we saw the KBDI levels which signified the amount of rain those areas need to be healthy. When the KBDI value is 0–200, it means that soil and fuel moisture were still high; when it was at 200–400, it means that it was starting to become somewhat dry; when was at 400–600, it means that there are drier and duff layers; and when was at 600–800, it means that the drought is severe. Since at lower levels means that it has more moisture, the chance of a forest fire or similar phenomena to happen would be less severe compared to when a KBDI value is high.

3.4 – History of Forest Management and Fire Fighting

Forest fires have existed since the start of human history and can have significant and mostly impacts to the environment such as causing problems to ecosystems, people's health such as getting severely injured or death, and the production of resources. It was also considered to be the biggest threat regarding the destruction of forests. It wasn't until the 1800s in the United States when two people named Franklin Hough and Bernhard Fernow decided that they should find ways to finally control them. They were concerned that forest fires were damaging too many trees and therefore disrupting timber production so they decided to ask the United States Government to set up the US Forest Service. By doing this, it set up standards and made certain areas of land to be protected by the government such as how to control fire, how fire behaves, and how it affected the forest.

However, they still needed to find a way to suppress forest fires. In 1910, there were a series of massive forest fires occurring in three states happening in just two days. The amount of damage that was done was significant and the US Forest Service had to find ways to stop that. They realized that they needed to have more men and equipment with proper training and backing from that service and they then created standards that would help them prepare for those types of situations. It also allowed cooperation from different regions in the United States that are usually far away from each other to manage and transfer data about those types of situations so that other places can know what is happening earlier.

In 1916, scientists like Coert DuBois began to research the causes of forest fires and how to control them and determined that they can be caused by weather hazards. By doing this, they divided up forests into different sections based on the weather they had to determine such causes of those hazards for happening. Then, they began to look at different methods on how to prevent forest fires from happening again.

One tactic that they created was light burning. This meant that small fires should be added around a forest fire to prevent it from spreading and it also improved land conditions. Many people believed that if some plants were burned earlier, then it would help the trees from the forest to be safe from fire outbreaks and it reduced the chances of that area to burn again. However, this proved to be somewhat ineffective because it still destroyed most of the vegetation around that area. So, another method was planned and this time it was stated that other nodes should be created such as better communication systems, fire watch towers, and roads to helped detect forest fires earlier.

In the 1930s, the Civilian Conservation Corps was created. This organization provided new methods to combat forest fires such as allowing more volunteers to help stop fires themselves. They also adopted new technologies such as smokejumpers and fire suppression chemicals to combat these fires anywhere and any place. Better documentation was then created to locate when these fires could happen next in a more accurate and faster amount of time.

Later on, more advanced methods and standards were thought of to understand forest fire control. They decided to split up to different research stations and analyzed more specific and different forest areas in the country. They also began to evaluate resources and determined what caused forest fires to happen. While doing that, they began to examine how to control these fires in a more economical manner. Local agencies began to promote better training to firefighters and other people responsible to look out for forest fires. Newer technologies such as radar and infrared sensors were adopted to detect when cumulus clouds are near.

3.4.1 – Response Times

Fire Response times help provided information of when and how resources from a fire station can be reached to an area affected by a forest fire and how far away they are. If a station was far away from an affected area or if they used all of their resources just on that area, then the response time would be longer since it would take more time to carry all other resources to other areas where it needs to be treated. Therefore, the response times were to be made sure that it was low most of the time.

Fire response times were broken into three rating components: Availability, Capability, and Performance. The availability measured which equipment was ready to use during a fire incident, the capability measured the equipment's ability to stop the fires, and the performance measured how effective the equipment was to combat the fires.

There is a standard that shows how to report fire response times and it is called NFPA 1710. NFPA establishes a criterion of the recommended response times needed for an emergency. Some of the recommended times by this standard include that the alarm answering time had to be around 15 to 40 seconds, the alarm processing time had to be around 64-106 seconds, the turnout time had to be about 80 seconds for fire responses, the First Engine Arrive on Scene Time had to be about 240 seconds, the Second Company Arrive on Scene Time should be about 360 seconds, the initial full alarm for low to medium hazards had to be about 480 seconds, and for high hazards, it had to be about 610 seconds. Also, response times were divided into three subsections called Call processing time, which was the time when the call was sent to a firefighting unit; Turnout time, which was the time when firefighters began to prepare for action, and Travel time, which was the time of when the unit arrived from the station to the area affected from a forest fire.

3.4.2 – How Often are Fires Reported

Here in the United States, fires weren't reported that much until the mid to late 1800s. That was because at that time, there wasn’t that much technology that allowed data to be shared to other places at a fast rate about where and when a forest fire happened. Most of the reporting was done by asking people who heard about those fires second hand. Later around that time, it had been reported that in the state of Idaho that fires happen at least once every four years. The chief of the Division of Forestry, Gifford Pinchot, created a method to analyze and report the occurrence of forest fires better. He cooperated with different agencies to determine the cause of those fires in different states by collecting and analyzing newspaper reports based on those respective areas. By the early 1900s, there was a decrease of forest fires due to having better safety regulations in the lumber industry.

Over the years, fire response times might have an effect on how many times fire incidents have been reported. It had been seen that each year, the number of times incidents have been reported had been increased. This is because there had been more advanced technology that had been able to report those fires at a faster rate and data regarding those incidents can be transferred to other far away areas quickly.

3.4.2 – What Are the Causes of how Fires are Spread?

Forest fires can be started in different ways. One of these is by human activity such as accidentally leaving the fire from a barbecue, burning bushes to clear up land, or playing with matches and other flammable devices. These actions caused these fires to spread to other areas uncontrollably. However, weather such as lightning storms and rain could have a huge impact as well. In an area where there was inadequate rain, it can cause trees to dry up because they need water to survive. Since those trees don’t contain too much water anymore, they had a higher chance to be flammable so when a forest fire happened, they can spread much faster. During thunderstorms, lightning can end up striking those trees and igniting fires. However, rain can have positive effects as well and can actually prevent and control fires from spreading because water is able to put down flames. Therefore, our design included features that detect how much rainfall an area is getting and the chances of how much lighting it got when a thunderstorm happens.

4.0 – Research and Part Selection

4.1 – Hardware Technologies Research

Sensors

We started this section responding to the most basic of the question, what is a sensor? A sensor is a device that transforms a physical quantity to an electrical signal, therefore providing a voltage or current or causing a resistance change. In general, the sensor's impedance, which included capacitive and inductive sensors, may alter. As a result, we were tasked with measuring one of these electrical values. below we explained in further details the subsets of analog sensors since this was the most heavily used technology in our project, the following sections served as the groundwork for explanation of the most used machinery of in our module.

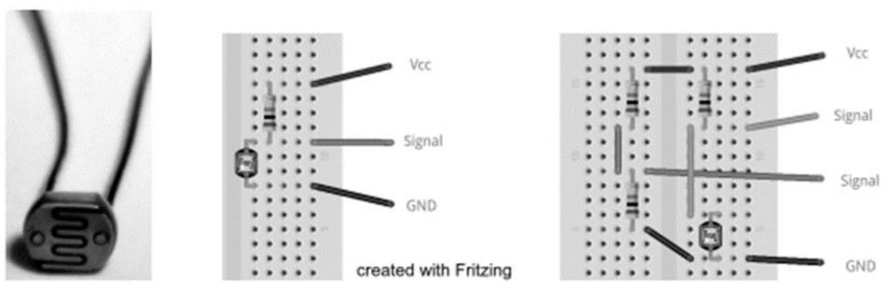
Analog Sensors

Analog sensors are devices that generate analog output in relation to the amount being computed. These sensors also monitored changes in environmental elements such as light intensity, wind speed, and sun radiation, among others. And the output voltage varied from 0 to 5 volts.

Resistance Base Sensors

The first resistance-based sensor we looked at at is a light-dependent resistance (LDR), which was seen bellow. Its resistance varied depending on how much light was exposed to. The variation range varied on the device and normally ran from 100 Ω to MΩ. This sensor schematic needed to be explained in detail since it's the basis of our moisture sensor which was explained in detail in the further sections

Figure 4.1.1 - voltage divider between the source voltage and ground



The figure above produced a voltage divider between the source voltage and ground, we connected a photoresistor in series with a resistor which gave the voltage Vs on the signal terminal. As a result, as the LDR was illuminated, its resistance RLDR changed, and the signal voltage fluctuated accordingly. We placed the resistor R0 in the middle of the RLDR range. As a result, the voltage we measured was about one-half of the supply voltage. As a result, we also utilized a voltmeter in that voltage range.

Very slight fluctuations in light intensity became difficult to resolve and may require amplification. This difficulty alleviated by using a Wheatstone bridge. Wheatstone bridge was an electrical circuit that was used to determine the unknown electrical resistance by balancing two legs of a bridge circuit, one of which contained the unknown component which compared the voltages in two resistor dividers, as illustrated on the right.

Resistance-based temperature detectors (RTD), such as the PT100 temperature sensor, are another type of resistance-based sensor. It is a platinum-based calibrated sensor with a resistance of precisely 100 at 0 oC. It was based on the fact that the resistance of a highly pure metal is governed only by the scattering of electrons in the conduction band with phonons, which are vibrations of the ions that comprise the metal's crystal lattice.

Temperatures generated greater vibrations of the lattice, resulted in increased resistance. At greater temperatures, one may expect the crystal ions to oscillate with bigger amplitudes, producing a larger target for the electrons to scatter and therefore hindering their mobility. Because this was a material attribute, calibration measurements of resistance as a function of temperature are universally valid for all sensors made of the same metal, provided the metal is highly pure and devoid of defects. Platinum wire coiled on a ceramic support body was a common material used in commercial sensors.

Voltage Base Sensors

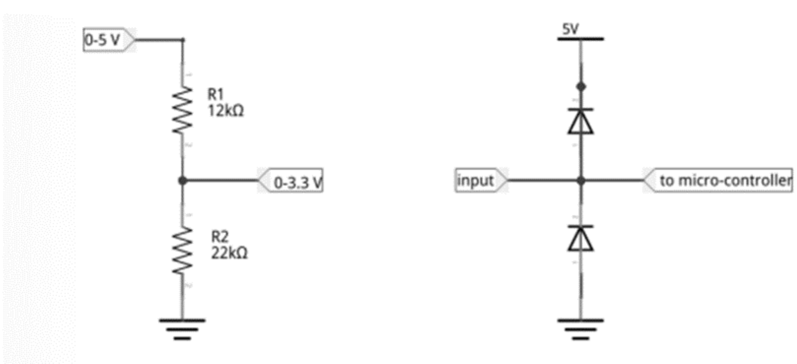
Voltage-based sensor was type of sensor that directly produced a voltage at its output pin. The LM35 temperature sensor was a common example of this type of sensors. The operation was based on the passage of familiar currents. For the sake of familiarity, we named these currents with current densities across the age drop. The voltage differential varied in direct proportion to the temperature. This was explained by inverting the current–voltage curve for the base-emitter junction diode.

One of the components that was good to make remarks of voltage-based sensors were Thermocouples. Thermocouples are temperature sensors that used the effects of temperature and temperature gradient on conductors composed of various materials to detect temperature. The Peltier effect (When an electric current is maintained in a circuit with two different conductors, one junction cools and the other heats; the effect was significantly stronger in circuits including dissimilar semiconductors) created a temperature-dependent current at the junction of the conductors. As a result of rearranging charges inside their crystal structure, certain crystals and ceramics reacted to external stressors by creating a piezoelectric voltage between opposite surfaces of the material.

Current Base Sensors

A current base sensor is a type of sensor, usually a Pin diode, which was a diode having a large, undoped intrinsic semiconductor area situated between a p-type and an n-type semiconductor region. that generated a current of up to 100 nA. Charge-coupled devices, or CCDs, were sensors used in imaging applications such as cameras. They were comparable to a pin diode linked to a tiny capacitor, one for each pixel of the camera. When exposed to light, a tiny charge was transferred to the capacitor.

Figure 4.1.2 – Current Sensor



On the left was a voltage divider that reduced the input voltage from 0 to 5 volts to 0-3.3 volts. The right design demonstrated the usage of clamping diodes to protect the microcontroller input from being between ground and 5 V. Solar cells functioned similarly to photodiodes in photovoltaic mode, providing a voltage to a load. They were, however, tailored to absorb as much of the spectrum as feasible while simultaneously having a wide absorbent surface to maximize the electric power available to the load.

Analog to Digital Conversion

With the previously mentioned sensors we normally wished to convert analog signals to digital format so that they may be processed on a computer. This feat was accomplished through an analog-to-digital converter (ADC), which was thought of conceptually as an extremely fast measuring voltmeter. It was made up of a sample-and-hold circuit, which kept the voltage steady for a brief period while it was measured and digitized. Several digitizing processes were used, and we got through a few of them. ADC, the voltmeter was built using a huge number of comparators that compared the voltage against a series of voltages produced from a resistor ladder. An extra circuit encoded the comparator output into a binary form. ADCs were frequently incorporated into microcontrollers and sensors had digital interfaces, such as those detailed in subsequent sections. Following our examination of the ADC, the workhorse of digital data-acquisition systems, we now turned our attention to the problem of powering our circuits.

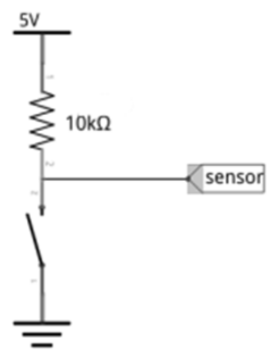
Supply Voltage for Sensors

Our sensors and microcontroller required power to function, and in our case, we had a mixture of both, conventional as well as alternative sources. (Explained in later sections) This was generally supplied by a power source. A transformer was used in a typical kind of power supply to scale down the 220 V or 120 V AC voltage from the wall to a regularly used voltage range of roughly 5 to 20 V**.** Batteries or rechargeable batteries can also be used to power an electrical circuit. Lithium-ion or lithium-polymer (Explained in later sections) rechargeable batteries, which was implemented on this mesh, are of relevance because they could supply very high currents to a circuit, which was especially significant for devices that demand high currents, such as WLAN circuits or motors. Because of their high inherent energy density, lithium-ion cells delivered voltages in multiples of approximately 3.7 V and required sophisticated discharging and recharging circuitry to prevent undercharging and overcharging, as well as physically injuring the battery. We utilized electricity from the USB port to power electrical circuits that was used near a computer. Many RS-232-to-serial converter circuits offer up to 500 mA, which is frequently enough for tiny circuits. We may also utilize solar cells and so-called supercapacitors as temporary power sources to make circuits highly portable in the field. After examining analog sensors, analog-to-digital converters, and power supply, we now moved on to sensors that create digital signals directly.

Digital Sensors

We now referred to sensors that did not require an external ADC and instead send their measurement information directly to the microcontroller in digital form. Some sensors already had an ADC built in, whereas others do not. We started with the latter, of those which buttons and switches are the most visible examples. The most basic digital sensor was either a switch that was either closed or open, or a button whose open or close state was only momentarily triggered. These words were used interchangeably. Our objective was to reliably sense their status, which was often accomplished with a pull-up resistor coupled to the supply voltage.

Figure 4.1.3 - pull-up resistor that is connected to the supply voltage with a switch



The supply voltage may be accurately detected via the microcontroller's sensing pin. The voltage on the pin dropped to zero or ground level only when the switch was pushed. When the switch was open, the voltage potential on the pin was undefined and dictated by stray capacitances in the system. As a result, when sensing the state of a switch, it was usually recommended to employ a pull-up resistor. When the resistor and switch positions were reversed, the resistor worked as a pull-down resistor, and the sensing level was 0 unless the switch was closed. Now we'll look at sensors that directly reported their measurement data in digital form, beginning with I2C devices.

I2C Devices

These devices broke down into their intrinsic form before discussing later further device types (barometric pressure sensor and humidity sensor) which were broken down into detail in further sections. With this being stated let's start laying the groundwork of this section. Language was used when individuals needed to be communicated with one another. When two individuals spoke the same language, they can chat about whatever they desire. Communication was tough if they could not speak the same language. Similarly, electrical components communicated, and each had their own languages too. There are a few commonly spoken languages. TTL Serial, SPI,, I2S, 1Wire, and Parallel are the most prevalent. Barometric pressure sensors include the BMP085, as well as subsequent variants BMP180 and BMP280.

They are based on using a piezoresistive strain gauge that measured the strain induced by the deformation of a membrane that separated an evacuated test-volume from the outside air pressure. The complete system was constructed directly onto a silicon substrate, and the strain gauge was formed by doping a tiny part appropriately. Because the semiconductor-based strain gauge was temperature dependent, a temperature sensor was integrated.

Air Quality Sensors

Air quality sensors are devices used to detect contaminants in the air. This included dangerous pollutants and noxious gases that may be harmful to human health. They were used in various applications like air quality monitoring, various gas detection that harmed the environment in very disastrous ways affecting the environment, combustion controllers and oxygen generators. So, our air quality sensors were placed on our product while activating the mesh network. As the mesh network was activated, the air quality detected the air and tried to obtain all the data it could displayed it on the web interface when the user hovered over the different nodes as they pleased. The air quality measured the how much oxygen and carbon dioxide was released in the air including other toxic chemicals which helped to give a better idea on how clean the air and/or the environment was. All of that information helped in detecting the possible hazardous conditions on the land, also fire hazards.

Other Sensors

Aside from the defined protocols, there were a variety of communication standards developed by device makers. The relative humidity sensor, DHT22, and the DHT11 were two examples of such devices; the latter was depicted on the right in Figure. The temperature was known to calculate the relative humidity, which was done via a thermistor. A capacitive humidity sensor was used to measure the humidity. The operating idea was based on the use of a tiny capacitor with an exposed dielectric that had a high affinity for attracting water, caused a substantial shift in the relative dielectric constant r and capacitance. The density of tiny particles floated in air can be used to measure air quality. Two examples of such sensors were shown above. A PPD42NS particle sensor was seen on the left. A resistor within this detector warmed the air, causing the air containing suspended dust particles to rise and pass through the light generated by an infrared diode. The dust particles scattered the light onto a phototransistor, which caused an output pin to be pulled to a low potential. A cleaned-up signal was provided after signal conditioning and amplification. When particles scatter light, it is low; otherwise, it is high. The sensor has been calibrated such that the time at low signal to total time ratio may be converted to particles per liter. The GP2Y1010AU0F, seen on the right, operates in a similar manner. It detects light dispersed off dust particles as well, but it regularly turns on the infrared diode and integrates the signal from the phototransistor, and the output value must be sampled 0.28 ms after the LED was switched on. The difference between the signal with and without the LED offers enough rejection of ambient light. Both dust sensors' effectiveness may be increased by placing them in the airstream formed by a fan, which must be turned on and off.

Several additional sensors, frequently linked to environmental factors like as humidity or barometric pressure, also report temperature as a byproduct since it is required internally to calibrate the given primary measurement result.

Biological and Chemical Sensors

For many years, researchers have worked on detecting very minute amounts of chemical and biological substances. Carbon monoxide detection for environmental applications, biological and biomedical monitoring, missile fuel leakage detection, and security applications all require high-sensitivity sensors that can quantify chemical or biological agents at low parts per billion (ppb) or parts per million (ppm). The recent advancement of microelectronic technology has created an excellent opportunity for the manufacturing of a sensor and an interface circuit on a single chip. The sensor and interface circuit are manufactured on a single chip, resulting in improved resolution, greater accuracy, and lower-noise signal conditioning and amplification. Furthermore, improved control over sensor working conditions such as temperature and linearity may be offered. When the sensor was near to the interface circuit, removing lengthy wire capacitance and the electromagnetic interface results in increased sensitivity and reduced noise. The array of sensors may be constructed on a single chip to reduce false alarms and increase sensor selectivity. Biochips are biosensors that contain transducers based on integrated circuits. A biochip is often made up of an array of biosensors that may be monitored separately and utilized to analyze numerous analytes. The interplay between the analyte and the bioreceptor is intended to have a quantifiable impact. A transducer is a device that turns a quantifiable impact into a signal (typically electrical) that can be monitored and recorded. Antibody/antigen interactions, nucleic acid contacts, enzymatic interactions, cellular interactions (microorganisms, proteins), and interactions employing biomimetic materials are the most prevalent biosensors (synthetic bioreceptor).

4.1.1 – Microcontrollers

The microcontroller unit or MCU was the brain of each node. The MCU can be thought of as a miniature computer with all of the sensors being its peripherals. There is a wide variety for microcontrollers available for use in today’s modern electronics. There were also application specific and general purpose microcontrollers. For our purposes, we focused on the feature set of general purpose microcontrollers.

These general purpose MCUs had a wide array of functionality with a shared set of core features. The central feature of an MCU is the processor. The processor was single or multiple cores and was responsible for reacting to the inputs and outputs. The inputs and outputs were handled in two ways. The first is general purpose input output or GPIO pins. These GPIO pins are capable of producing or accepting analog and digital signals. Not all GPIO pins were created equal, some were unidirectional or may only have handled digital signal processing. The second way the inputs and outputs were handled was through the use of widely accepted peripheral interfaces. Some of the most notable standards were Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), and Serial. These interfaces have overlapped with the same pins used for GPIO operations or have their own dedicated pins. We dove deeper into these interfaces in other sections of our research. The presence of these interfaces were important to this application as many of the sensors rely on one of these common interfaces for communication. Though, the interfaces themselves were well documented and may be implemented separately, having them built into the MCU reduced complexity and increased reliability of the system.

We also considered the software development flow for the MCU. When creating programs to run on our microcontroller, we used an Integrated Development Environment or IDE.

The IDE was responsible for multiple functions in the software development flow. The first was, accepting the high-level programming language such as C, C++, or MicroPython. Then translating our chosen high-level language into the lower-level operations needed for the microcontroller. Beyond just translating higher level language, many microprocessors also offer human readable constructs to interact with the hardware components. An example of this was, the Wire functionality offered by Arduino for interacting with devices connected to the I2C bus. The prevalence and syntax for these features was unique to the family of microcontroller boards. The next step in the programming flow was actually loading the program onto the microprocessor's local storage. This may have included in the IDE or may require external tools. Lastly, each IDE may had a resource for communicating with the device and debugging code at execution time. This ranges from a basic serial console to a detailed range of tools for viewing the contents of registers or power consumption.

SB Development Board 32. E.s. P 32 is a low-cost system in the chip series made by Espressif Systems, a Shanghai-based Chinese startup, which is fabricated by TSMC using their 40 nm technology. Their Microcontroller had the ability of Wi-Fi and Bluetooth, which makes it chip all that was rounded for the development of our mesh network. When it came to chip characteristics, it featured a dual-core CPU, which meant that it had two processors. It also had Wi-Fi and Bluetooth. And you don't have to plug in the B Dongle every time you want to utilize Wi-Fi or receive a module. This came very useful for our mesh setup since Wi-Fi and Bluetooth were necessary requirements for our communication layout between nodes. The chip also executed a 32-bit application The clock frequency of this board may have reached 240 megahertz, and it may have had 512 kilobytes with 30 or 36 rounds, 15 in each row. It also had a variety of extensions available, including such capacitive touch sensors and digital converters, Universelle digital analogue converter, asynchronous series communication modules, SPI, ice squirty, and others. the dual-core processor also had 2.4 GHz, 802.11 b/g/n WiFi, and 40 MHz of bandwidth support built in. It has Bluetooth Low Energy 5.0 connection and was capable of long-range communication of up to 1 kilometer via the coded PHY layer.

With 2 Mbps transfer capabilities, it also provided faster transmission speeds and data throughput. The ESP32-S3 had a superior RF performance even at high temperatures, which was an outstanding characteristic for our implementation since it could have been exposed to the high heats of the forests. The chips had an ultra-low-power (ULP) core that enables a variety of low-power modes. Because it supported AES-XTS-based flash encryption, RSA-based secure boot, digital signature, and HMAC, the ESP32-S3 was extremely safe, this feature had been from the most help since we were making our modular network restricted from a conventional power source from substantial period. Vector instructions had been added to the Xtensa LX7 core. Vector instructions were a type of instruction that allowed for simultaneous processing of AI datasets, enhancing performance and power efficiency.

These instructions were used and leveraged once we got to the part of image processing for early fire detection with the camera module. Espressif developers were actively working on upgrades to the ESP-WHO face detection library and the ESP-Skai net speech recognition library Our team plans to constantly es committed to use to most updated library to improve usability and efficiency of our module.

4.1.2 – Anemometers

An anemometer is an instrument used to measure wind conditions. Specifically, anemometers may measure wind speed or direction or both. The anemometer is a core component to our project as the state of the wind is a critical factor in predicting fire propagation. With that, we have looked at each of the available types of anemometers.

First up was the most basic type of anemometer, the cup anemometer. These are one of the first variations of anemometer ever created. They were invented by Dr. John Thomas Rodney Robinson in the year 1845 at the Armagh Observatory in Northern Ireland. The operating principle of this type of anemometer was to capture the wind in cups whose openings are placed perpendicular to the wind direction vector. The cups were mounted to a central hub and the apparatus generally contained three cups. As the wind speed changed, the cups interfaced with the wind to create an acceleration or deceleration of the angular velocity at the hub. The number of rotations of the hub was measured providing a means to extrapolate wind speed. The rotational rate of the hub was directly correlated to the measurement of wind speed. As a result, it was important to consider the frictional or other forces contributing to the arrest of this motion. There were two main ways to reduce external forces on the system. The first was to utilize low friction bearings for the hub. The bearings were similar to the ones used in hard disk drives. The second method was to use a sensor for the speed that did not mechanically interface with the hub. A common method to record the number of rotations per time period was by using a hall effect sensor. Hall effect sensors function in the presence of a sufficiently intense magnetic field. The sensor itself was composed of semiconductor material that allowed the flow of current only in the presence of magnetic flux generated by the magnetic field. The Hall effect sensor was placed near the hub of the anemometer. A magnet was then attached to the hub itself. In this configuration, the magnet passed the sensor once per revolution of the anemometer hub. As the magnet passed the Hall effect sensor, current was allowed to flow through the sensor and a brief pulse of voltage was measured. A microcontroller may have observed the frequency of these voltage pulses. The anemometer may be calibrated by directing a set of wind events with a known speed over the anemometer. We characterized the frequency of pulses as wind speed. With a set of known pulse values, unknown wind speeds was extrapolated mathematically.

The cup anemometer measured wind speed only. To measure wind direction, we considered the next type of anemometer, the wind vane. In its most basic form, the wind vane has been around for thousands of years. The first recorded wind vane was created by Greek astronomer Andronicus in the year 48 B.C. The basic design still remains in modern versions of the wind vane. The wind vane uses a vertical rudder to align its axis with the direction of the wind. The rudder is often mounted to a nacelle with the center of the nacelle attached to a fixed vertical axle. The mechanical mate between the nacelle and the axle is important to the functionality of the wind vane. Reducing the friction and other arresting forces at this connection point increases the potential accuracy and precision of the system. For example, at low wind speed the force of the wind moving over the rudder may not overcome the force required to move the nacelle. This would cause the system to register an incorrect measurement. We may have obtained the wind direction by measuring the position of the nacelle in relation to the fixed axle. There were two main methods for measuring the position of a wind vane. The first was a mechanical connection between the nacelle and a continuous positional sensor. A continuous positional sensor was also referred to as a continuous potentiometer. As the input shaft to the potentiometer rotated, the resistance changed. This change was a result of an internal wiper connected to one output moving over a conductive carbon surface connected to another output. The resistance between these outputs may be measured by observing the output voltage with a known input voltage. When the wiper moves in one rotational direction, the voltage increases. In the other rotational direction, the voltage decreased. The input shaft had a continuous range of motion allowing this voltage to be represented the absolute position of the input shaft. A microcontroller may have been used to translate the voltage into positional information. This was possible by using an analog to digital converter (ADC) input to read the voltage of the continuous potentiometer. The ADC counts may have been correlated to the physical position of the sensor by choosing a fixed reference point then, rotating the input shaft 360 degrees. The second method for measuring the direction of the nacelle was by using magnetic sensors. Either by measuring the position of the nacelle in relation to the fixed axle or by measuring the earth's magnetic field to attain a compass azimuth. For measuring the compass azimuth, we may have used a microelectromechanical (MEMS) magnetic field sensor. The basic function of this type of sensor was to supply an output representing the direction of the magnetic field lines passing through the device. It was important to note that magnets elsewhere in the system may have caused interference when attempting to measure the earth's magnetic field. We may have also considered measuring the position of the nacelle itself. This was possible by attaching a diametral magnetized magnet to the center point of the nacelle. The magnetic field produced may then be measured to produce a 0-to-360-degree angle output.

The next type of anemometer was the windmill style. The functionality and principles of the cup anemometer carried over to this style. Where the two diverge was in the implementation of the mechanical interface between the anemometer and the wind. The cups were replaced with a propeller. The propeller was mounted with the hub parallel to the wind direction vector creating a rotation axis around the wind vector. The shape of a windmill anemometer was akin to a propeller airplane with only the tail fin. The tail fin served as an important purpose in this case as the effectiveness of the propeller depended on the face pointing into the wind. For example, if the wind blew across in the same direction the blades extend out from the hub, the blades did not effectively catch the wind. The result was a slower than actual winds speed. A propeller anemometer may have also worked through the use of a Hall effect sensor. This was to retain the same properties of low impedance to the rotational motion of the propeller. The tail fin of a windmill style anemometer also provided another measurement, wind direction. The tail fin kept the nacelle pointed into the wind. As a result, measuring the position of the nacelle allowed the measurement of wind direction. This direction measurement was conducted in the same way the vane type operated.

Another type of anemometer was the hot wire style. The hot wire anemometer worked by heating up a thin piece of wire and placing it in the path of airflow. As the air passed over the wire, the wire was cooled. There were two ways to translate this cooling into a measurement of airflow. First, was the constant current method. This method passed a constant current through the wire and measured the change in temperature. The change in temperature may have then been used to find the airflow. The second method was the constant temperature method. This method relied on the electrical property that when a wire was heated, the resistance decreased. The objective of this method was to maintain the temperature and resistance of the wire. The current needed for this was the measurement that may have been translated to airflow.

The final type of anemometer that we investigated was the ultrasonic anemometer. An ultrasonic anemometer used sound waves to measure both the wind speed and direction. This was accomplished through the use of the Doppler Effect. The Doppler effect described the behavior of sound waves as an emitter moves towards or away from an observer. In this case, the wind was the acting motion, and the transmitter and observed were stationary. An ultrasonic emitter was placed at the center three equally spaced receivers. All four transducers were faced towards a flat surface acting as a reflector. As wind passed between the transducer housing and the reflector, the frequency was seen by each receiver shifts. As the wind pushed the sound waves, the receiver that was downwind observed a frequency increase. While the receiver that was downwind observed a frequency decrease. In this system, the wind acted as the motive force which emulated the movement of the transmitter in relation to the receivers. This was the Doppler effect in action. An important consideration for this system was the sensitivity of the measurements and the related potential interference.

A final note, there were a few types of anemometers that were not mentioned in this section. We did briefly investigated these types, but they were ruled out for cost, complexity, or being analog only with no means to translate to a digital readout.

4.1.4 – Humidity Sensors

The quantity of water vapor in the air is referred to as humidity. The quantity of water vapor in the air in relation to the maximum amount of water that the air can retain is referred to as relative humidity. When determining the fire hazard of a location, humidity is a crucial aspect to consider. There were 3 different types of humidity sensors when it came to humidity detecting technology: capacitive, resistive, and thermal conductivity humidity sensors.

Capacitive Humidity Sensors

In industrial, commercial, and weather telemetry applications, capacitive humidity sensors (CHS) were commonly utilized. CHSs were made up of a substrate and a thin layer of polymer or metal oxide placed between two conducting electrodes. To protect the sensing surface from contaminants and moisture, it was covered with a porous metal electrode. The substrate was usually made of glass, ceramic, or silicon. The fluctuations in a CHS's dielectric constant were nearly precisely proportional to the surrounding environment's relative humidity (RH).

Advantages: Able to function in high-temperature environments (up to 200 °C), near linear voltage output, wide RH range, high condensation tolerance, reasonable resistance to chemical vapors and contaminants, minimal long-term drift, high accuracy, small size, and low cost.

Disadvantages: Limited sensing distance and sensor interface can be tedious and difficult

Resistive Humidity Sensors

Resistive humidity sensors (RHS) detected changes in the electrical impedance of a hygroscopic material such as conductive polymer, salt, or a treated substrate. These sensors were appropriate for use in industrial, commercial, and home control and display equipment. RHS was made up of noble metal electrodes that were either photoresist-deposited on a substrate or wire-wound on a plastic or glass cylinder.

Advantages: included a faster reaction time than CHS, near linear voltage output, great precision, a compact size, low cost, and a wide RH range.

Disadvantages: Lower working temperature than CHS, susceptible to chemical vapors, low contamination tolerance, and low condensation tolerance.

Thermal Conductivity Humidity Sensors

Thermal conductivity humidity sensors (TCHS) calculated absolute humidity by comparing the thermal conductivity of dry air to that of air containing water evaporated sensors were appropriate for applications such as wood drying kilns, textile, paper, and chemical solids drying machinery, pharmaceutical manufacture, cookery, and food dehydration. The TCHS was made up of two matched negative temperature coefficient (NTC) thermistor components in a bridge circuit, one of which was hermetically sealed in dry nitrogen and the other which is exposed to the atmosphere.

Advantages: Extremely long service life, ability to function in high-temperature conditions (up to 600°C), great resistance to numerous chemical and physical impurities, high precision, and high condensation tolerance.

Disadvantages: Respond to any gas with thermal characteristics other than dry nitrogen; this impacted readings.

4.1.4 – Temperature Sensors

Temperature Sensing

In the field of sensors, temperature sensing technology was one of the most extensively utilized sensing technologies. It enabled temperature measuring in a variety of applications and protected against extreme temperature excursions. There were five distinct types of temperature sensors on the market. Each temperature sensor family had benefitted and downsides. One sensor may have been more suited to a certain application than the other.

Thermocouples

A thermocouple was a connection made by two wires of different metals. The voltage generated by the point of contact between the wires was proportional to the temperature. Thermocouples measured across a wide temperature range, up to 2300 °C. They were less appropriate for instances where smaller temperature variations must be measured precisely. Thermistors and resistance temperature detectors (RTDs) were better suited for such applications. Temperature was one application. Kilns, gas turbine exhaust, diesel engines, and other industrial operations were all measured.

Advantages: included a wide temperature range (200 °C–1300 °C), a low cost, good accuracy, little long-term drift, and a quick reaction time.

Disadvantages: The relationship between temperature and thermocouple output signal is not linear, the output signal (mV) is low, thermocouples are susceptible to corrosion, and thermocouple calibration is time-consuming and complicated.

Resistance Temperature Detectors

RTD's were widely utilized in a variety of industrial applications, including air conditioning, food processing, textile manufacturing, plastics processing, microelectronics, and exhaust gas temperature measuring. RTDs were essentially temperature-sensitive resistors. Temperature caused an increase in resistance. The majority of RTD elements were made up of a length of fine-coiled wire wrapped around a ceramic or glass core.

Advantages: Linear throughout a large temperature range, moderately accurate, strong stability and repeatability at high temperatures (65 °C–700 °C).

Disadvantages include low sensitivity, higher cost than thermocouples, and susceptibility to stress and vibration.

Thermistors

Thermistors, like RTDs, were temperature-dependent resistor devices. Thermistors were not as precise or reliable as RTDs, but they were easier to wire, cost less, and were directly accepted by practically all automation panels. Thermistors were built of semiconductor materials having a temperature-sensitive resistance.

Advantages: include high sensitivity, low cost, accuracy over a narrow temperature range, and good stability.

Disadvantages: Nonlinear resistance temperature characteristics, self-heating, and a restricted temperature working range are disadvantages.

Integrated Circuit Temperature Sensors

Most of the sensors mentioned previously were either costly or need the use of extra circuits or components in low-cost applications. Integrated circuit (IC) temperature sensors, on the other hand, are entire silicon-based sensing circuits with either analog or digital output. IC temperature sensors were frequently utilized in applications with minimal precision requirements.

Advantages: low cost, high linearity, and easy-to-read output.

Disadvantages: a limited temperature range, self-heating, fragility, and being somewhat less precise than other varieties.

4.1.5 – Barometers

Atmospheric Pressure

One standard atmosphere of pressure equals 1.01325 105 Pa (N/m2) (14.6960 pounds per square inch, 1.01325 bars, 1013.25 mars, 760.00 mm Hg, or 29.920 in. Hg). This was the mean atmospheric pressure at sea level. At sea level, atmospheric pressure normally deviated from one standard atmosphere by little more than 5%. With increasing altitude, atmospheric pressure falls. Air pressure measuring equipment known was altimeters were used in airplanes to determine altitude. At roughly 5500 m (18,000 ft), the air pressure was half that of sea level. The following instruments were used to measure atmospheric pressure.

Mercury Manometer

Barometric pressure was originally measured with a mercury manometer. This was a 1 m long tube filled with mercury and inverted into an open dish of mercury. The height of the column of mercury maintained in the tube by the external pressure was a measure of the external air pressure. As a result, one standard atmosphere equaled 760 mm Hg. While accurate, this gadget was cumbersome and has been phased out for common usage.

Aneroid Barometer

It was made from a partly evacuated chamber that may have expanded and contract in response to changes in external pressure. The evacuated chamber was frequently a set of bellows, with expansion and contraction occurring in only one dimension. The most basic aneroid barometers, which are still in use, contain a mechanical connection to a pointer that provides a reading on a dial calibrated to read air pressure. Mechanical barometers of high grade can attain an accuracy of 0.1 percent of full scale. Aneroid barometers can also have an electronic readout, removing the need for a mechanical connection; this is more common for serious meteorological measurements. A magnet attached to the free end of the bellows is placed near a Hall effect probe in one way. The output of the Hall probe is proportional to the distance between the magnet and the Hall probe. An aneroid apparatus, which consists of a stiff cylindrical chamber with a flexible diaphragm at one end, is also used to measure barometric pressure. A capacitor is made by attaching one fixed plate adjacent to the diaphragm and another plate to the diaphragm. The capacitance varies when the diaphragm expands or shrinks. Calibration establishes the pressure associated with each capacitance value. For ground-based measurements, a typical range of 800 to 1060 mbars with an accuracy of 0.3 mbars is used. This type of equipment is manufactured by Setra Corporation for the National Weather Service ASOS network, which is manufactured by AAI Systems Management Incorporated. The ASOS network is explained more below. Pressure measurement is also covered in other sections of this chapter.

4.1.6 – Air Quality Sensor

Air quality sensors are devices used to detect contaminants in the air. This included dangerous pollutants and noxious gases that may have been harmful to human health. They were used in various applications like air quality monitoring, various gas detection that can harm the environment in very disastrous ways affecting the environment, combustion controllers and oxygen generators. So our air quality sensors were placed on our product while activating the mesh network. As the mesh network is activated, the air quality detected the air and try to obtain all the data it can and display it on the web interface when the user hovers over the different nodes as they please. The air quality was able to measure the how much oxygen and carbon dioxide are released in the air including other toxic chemicals which helped to give a better idea on how clean the air and/or the environment is. All of that information helped us in detecting the possible hazardous conditions on the land, also fire hazards.

4.1.7 – Visual Cameras

Cameras provide the machine with the ability to see. Cameras were particularly common exteroceptive sensors for detecting the outside environment, especially in mobile and stationary systems. The camera's picture was processed to extract elements of the environment and identify things. The gear used to process pictures has become extremely low-cost, quick, and tiny in size. This has made integrating cameras in robotic systems relatively simple and cost-effective Based on the output signal, there were two types of cameras: analog and digital. Digital cameras were available with a variety of interfaces, including USB, FireWire (IEEE 1394), TCP/IP, and SmartLink. Cameras with built-in RF wireless transmitters are very common and may have been found on autonomous flying vehicles such as Quadrotors and rescue robots.

4.1.8 – Compass

A digital compass sensor determined the bearing and direction of a robot or autonomous mobility platform. The four cardinal (N, E, S, W) and four intermediate (NE, NW, SE, SW) directions were frequently measured. These sensors were critical for robots that must negotiate unfamiliar terrain. For inclination adjustment, it employed a magnetic field sensor and a tilt sensor.

Tilt Sensor

A tilt sensor or inclinometer measured inclination, i.e., slope (or tilt) angles on a single or dual axis. It determined an object's elevation or inclination with regard to gravity. Inclines (positive slopes as seen by an observer looking upward) and declines (negative slopes as seen by an observer looking below) are both measured using inclinometers. Tilt meters, slope gauges, gradient meters, gradiometers, level gauges, declinometers, and pitch and roll indicators were some of the other names for it. These sensors provided analog signals or serial digital data as output. MEMS, accelerometer, liquid capacitive, electrolytic, gas bubble in liquid, and other sensor technologies are often used in inclinometers as well as pendulum

4.1.9 – Soil Moisture Sensors

Excess moisture was undesirable in agriculture, housing, textiles, packaging materials, electrical gadgets, dry food processing, and other industries. Moisture detection was useful in a variety of scenarios. Soil moisture assessment, for example, was beneficial for reducing the quantity of irrigation water used to grow plants and maximizing plant development. Excess moisture also caused "wet feet" in plants, which killed them. Because the moisture content of materials was so important, numerous ways for measuring it have been devised. This section described a variety of soil moisture sensing technologies that were now available on the market, as well as their benefits and drawbacks.

Frequency Domain Reflectometry Soil Moisture Sensor

Frequency domain reflectometry (FDR) was also known as a capacitance sensor. The FDR technique of soil moisture measurement employed an oscillator to create an electromagnetic signal that was carried through the unit and into the soil. The dirt reflected some of this signal back to the device. The FDR probe measured the reflected wave and reported the water content of the soil to the user.

Advantages: Highly accurate, quick reaction time, and low cost.

Disadvantages: They must be calibrated for the type of soil in which they were buried.

Time Domain Reflectometry Soil Moisture Sensor

TDR sensors sent a pulse down a line into the soil, which was terminated at the end by a probe equipped with wave guides. TDR devices evaluated soil water content by monitoring how long it took for the pulse to return.

Advantages: Extremely precise and quick reaction.

Disadvantages: Calibration can be time-consuming, complicated, and costly.

Gypsum Blocks

Gypsum blocks employed two electrodes put inside a tiny block of gypsum to monitor soil water tension. The electrodes' wires were attached to either a portable handheld reader or a data logger. The electrical resistance between the two electrodes within the gypsum block determined the quantity of water in the soil. More water in the soil reduces resistance, whereas less water increases resistance.

Advantages: Low cost and simple installation.

Disadvantages: Must be replaced on a regular basis and are sensitive to the salty content of water.

Neutron Probes

Another method for measuring soil moisture content was to use neutron probes. A probe put into the earth emits neutrons, which were low-level radiation. These neutrons clash with the hydrogen atoms in water, which the probe detects. The more water there was in the soil, the more neutrons were reflected back to the apparatus.

Advantages: Extremely precise and quick reaction.

Disadvantages: Expensive, and users must be registered with the government owing to the usage of radioactive materials to release neutrons.

4.1.9 – Photovoltaic Panels

Energy was often regarded as the driving force behind economic progress across the world. Global energy resources were divided into three categories: fossil energy (oil, gas, coal, etc.), nuclear energy, and sustainable sources (wind, solar, geothermal, air power, biomass, hydrogen, ocean, etc).

Wind power was now the primary source of new renewable energy. Over the previous 10 years, global wind power capacity has increased at a pace of 30 percent on average. In 2007, around 20 GW of additional capacity was built, increasing the global total for that year to 94 GW. This yearly expenditure by a sector that employs 200 000 people and served the electrical demands of 25 million consumers amounts to around 25 billion euros. This significant development has drawn investment from major manufacturers such as General Electric, Siemens, ABB, and Shell, as well as various electrical providers, most notably E.ON and Scottish Power. Wind power had a promising future over the next two decades.

Today's energy pathway was heavily reliant on fossil fuels, and the repercussions in terms of climate change and energy security was already dire. Since the 1970s, solar cells have been recognized as a significant alternative energy source. Solar cells were also promised as a carbon-free energy source capable of mitigating global warming. A solar cell's power conversion efficiency was well characterized as the ratio of electric power generated by the cell to incident sunlight energy per unit of time. Now, the greatest documented cell efficiencies in labs were over 40%, whereas thermal power production power conversion efficiencies can surpass 50%. This, however, does not imply any advantage for thermal generating, as its supplies, such as fossil fuels, are finite, but solar energy is basically inexhaustible.

There was global consensus on the need to reduce carbon emissions, and various policies were being developed both globally and locally to accomplish this. The European Commission unveiled an Energy Package on January 10, 2007, which was approved by the European Council. The EU's greenhouse gas emissions were to be cut by 30% by 2020 if a worldwide agreement is reached, or by 20% unilaterally. The objective to offer a 20% share of energy from renewable energy (RE) sources in the total EU energy mix was a critical component in achieving this goal.

Governments all around the globe have worked to enhance the percentage of renewable green energy in electricity generation. Because of the seriousness of global warming, there is a growing interest in solar energy research. As an alternative energy source, converting sunlight into electricity is critical. Solar energy is regarded as the finest renewable energy source since it is limitless and clean.

Solar radiation may be captured and turned into many types of energy without polluting the environment. Solar collectors, such as solar cells, were required to convert solar radiation. The primary concerns have been energy security, rising carbon-based energy pricing, and mitigating global warming. Global shipping contributes significantly to global greenhouse gas (GHG) emissions, accounting for around 3% of total CO_2 emissions. 1, 2, 3 The International Maritime Organization was presently trying to establish GHG laws for global shipping, and it was under pressure from the EU and the United Nations Framework Convention on Climate Change, among others, to implement measures that had a significant influence on emissions.

Solar radiation may have been captured and turned into many types of energy without polluting the environment. Solar collectors, such as solar cells, were required to convert solar radiation. The primary concerns have been energy security, rising carbon-based energy pricing, and mitigating global warming. Global shipping contributes significantly to global greenhouse gas (GHG) emissions, accounting for around 3% of total CO2 emissions. 1, 2, 3 The International Maritime Organization was presently trying to establish GHG laws for global shipping, and it was under pressure from the EU and the United Nations Framework Convention on Climate Change, among others, to implement measures that have had a significant influence on emissions.

We began our study of PV technology with a brief history of solar energy. Humans were already employing magnifying glasses to focus sunlight and so generating fire around the seventh century BCE. Later, focusing mirrors were used for the same reason by the ancient Greeks and Romans. Horace-Bénédict de Saussure, a Swiss scientist, invented heat traps, which are similar to small greenhouses, in the 18th century. He built hot boxes that consisted of a glass box within another larger glass box, with a total of up to five boxes.

Alexandre-Edmond Becquerel, a 19-year-old French scientist, developed the photovoltaic phenomenon in 1839. This phenomenon was found in an electrolytic cell composed of two platinum electrodes immersed in an electrolyte. An electrolyte is a solution that conducts electricity; Becquerel used silver chloride dissolved in an acidic solution. When he exposed his setup to sunshine, he saw that the current of the cell increased.

The true invention of solar cells as we know them now began in the United States at Bell Laboratories. Daryl M. Chapin, Calvin S. Fuller, and Gerald L. Pearson, three of their scientists, created a silicon-based solar cell with an effectiveness of roughly 6% in 1954 [43]. They are seen in Figure 11.2 in their laboratory. D. C. Reynolds et al. reported on the photoelectric cell of cadmium sulphide (CdS), an II-VI semiconductor, the same year.

Zhores Alferov, a Soviet researcher, invented solar cells based on a gallium arsenide heterojunction in 1970. This was the first solar cell to use III-V semiconductor materials, as discussed in Section 13.2. Dave E. Carlson and Chris R. Carlson founded the company in 1976.Wronski pioneered the use of amorphous silicon in thin-film solar systems while working at RCA Laboratories. The first solar cell was developed by the firms SHARP and Tokyo Electronic Application Laboratory. On the market, there were a variety of battery-powered calculators.

The public's interest in photovoltaic devices for terrestrial applications surged throughout the 1970s as a result of the oil crisis, which resulted in drastically rising oil costs. PV technology was transitioning from a specialty technology for space purposes to a technology appropriate for terrestrial uses at the time. Many businesses began developing PV modules and systems for terrestrial applications in the late 1970s and early

1980s. Solar cells remain vital for space applications, as seen in Figure 11.3, which depicts a solar panel array on the International Space Station (ISS).

The first thin-film photovoltaic arrays based on a copper-sulphide/cadmium-sulphide junction with a conversion efficiency more than 10% were demonstrated at the University of Delaware in 1980. At the University of New South Wales in Australia, crystalline silicon solar cells with efficiency more than 20% were shown in 1985.

The Chinese government had been significantly investing in their PV business since around 2008. As a result, for some years now, China had been the main PV module manufacturer. In 2012, global solar energy capacity exceeded the magical 100 GWp threshold. Thus, between 1999 and 2012, installed PV capacity increased by a factor of 100. In other words, the average annual growth rate of installed PV capacity over the previous 13 years has been over 40%.

For ages, solar energy was already utilized to create heat for decades. Since Gerald Pearson, Daryl Chapin, and Calvin Fuller invented the crystalline silicon solar cell in 1954, solar cells have now become a highly important choice for large-scale solar power production. Photovoltaic power already accounts for 1% of worldwide electricity output in 2015. According to the 2014 IEA Roadmaps for Solar Photovoltaic and Solar Thermal Energy, solar photovoltaic and solar thermal power will account for 27 percent of worldwide electricity output by 2050.

Solar energy was already employed in rural regions to provide tiny amounts of power and heat, helping to the economic growth of these communities. Millions of tiny photovoltaic systems were in use, supplying energy for things like lights and telecommunications. Solar energy systems may have been extremely successfully integrated into the physical environment and were actively contributing to the phenomenal development in solar energy use that we are seeing today. Solar energy may have been utilized to generate electricity on a large scale in power plants using flat plate and concentrator photovoltaic (PV) systems, as well as thermal concentrated solar power (CSP) systems.

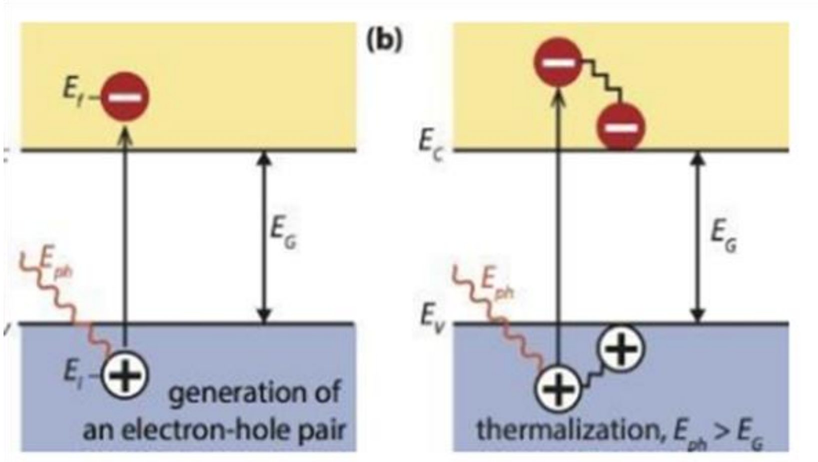
The photovoltaic effect, or the development of a voltage differential at the junction of two distinct materials in response to electromagnetic radiation, underpins the operation of solar cells. The photovoltaic effect was like the photoelectric effect in that electrons were released from a substance that had absorbed light with a frequency greater than a material-dependent cutoff frequency. Albert Einstein realized in 1905 that this phenomenon may be described by assuming that light is made up of well-defined energy quanta known as photons. The energy of a photon of this kind is given by

$$E = Hv$$

where h is Planck's constant and v is light frequency in 1921, Einstein was awarded the Nobel Prize in Physics for his explanation of the photoelectric phenomenon.

When a photon absorbs in a material, its energy is utilized to accelerate the electrons from an initial energy level to a higher energy levelE_f, as shown in Figure. Photons can be absorbed only if the electron energy values $E_i \; and \; E_f$ are present in such a way that the difference between them matches the photon energy,$hv = E_f - E_i$. Electrons in an ideal semiconductor can inhabit energy levels below the valence band edge, E_V, and above the conduction band edge, E_C . There are no permissible energy levels that can be occupied by electrons between those two bands. As a result, this energy differential is known as the bandgap, $E_G = E_C - E_V$.

Figure 4.1.9.2 – PV Panel Functionality [42]



A photon's absorption in a semiconductor with bandgap E_G. is depicted. A photon with the energy is depicted. A photon with the energy $E_ph = hv$ excites an electron from $E_i \; to \; E_f$. A hole is formed at E_i A portion of the energy is thermalized $if \; E_ph > E_G$

The peak power P_{max} , short circuit current density Jsc, open circuit voltage V_{oc}, and fill factor FF are the major characteristics used to characterize the performance of solar cells. These values are calculated using the lighted characteristic. These characteristics can be used to Fsec calculate conversion efficiency.

4.1.10 – Batteries

The term “Lithium Battery” referred to a lot of lithium that were synchronized and connected electrically. Similar to other batteries, lithium battery cells have a negative electro-node, positive electro-node, a separator, and an electrolyte solution. Atoms or molecules with a net electric charge or ions are moved from a positive electrode to a negative electrode through an electrolyte solution. The lithium cells contained and released power by converting potential energy into electrical energy using lithium ions. Electrolyte solutions parsed ions to flow between the electrolyte between the electrodes in a free state.

Lithium-ion batteries utilized lithium in ionic forms instead of lithium in solidified form. These forms were usually rechargeable and most of the time don’t need to be moved away from a device. These batteries power devices like daily use of computers, laptops, cameras, phones, power tools, etc.

The lithium-metal batteries most of the time contained lithium-metal electrodes and were non-rechargeable. These batteries were generally used to power day to day devices like flashlights, watches, calculators, car-keys, etc.

We had seen the positive outcomes while using these batteries, but they also came with a lot of dangerous hazards. Lithium batteries were generally very safe and didn't fail most of the times, however they cannot be without defects for a very long time. When these batteries failed to operate in a safely manner or were damaged, then they can cause a fire or very explosive conditions. Also, with improper use or bad handling can, storing, or charging it may have caused these batteries to perform poorly. It was very important and essential to test batteries, chargers and other equipment in use with these to follow appropriate and safe testing standards, for example: UL 2054, NERT certification, product redistribution and aid identifying other defects in design, quality of material and manufacture. The best way to dispose or throw out Lithium batteries was by recycling them because it may just end up in a landfill and it may cause a lot of harm to the environment due to the toxic elements that it may have been stored in itself. Even though it was recyclable, the procedure of doing that was a little complicated. Since lithium was a highly reactive element, it cannot be mixed with other types of batteries discussed above also including elements when it was sent to a recycling facility. Since recycling material required usage of heat to melt the material, it would not be advisable to work with lithium batteries because they can be flammable.

It doesn't take a long period of time to damage lithium batteries from physical impact, exposure to a specific range of temperatures and improper charging conditions:

- Some physical impacts that can crush the lithium batteries are through dropping, punching, and crushing it.
- Lithium batteries can be badly damaged when temperatures too high (exceeding 130 F). Heat sources like heaters, open flames, matches, etc can also intensify the failure rate in cells with damages from other causes.
- Lithium batteries can be badly damages when temperatures are too low as well (low below 32 F) during the process of charging. Charging batteries in sub 0 degrees Celcius can lead to lifelong metallic lithium buildup on the anode which increases failure significantly.
- It is important to follow the manufacturer's instructions as charging the battery blindly can cause damage to rechargeable lithium-ion batteries. For instance, if some manufacturer-authorized chargers cycled the power of the battery to be on/off before it is completely charged to disallow overcharging. Since ultra-fast chargers may not cycle power too much, it is important to not use them unless the manufacturer's instructions includes an option to do so.

It was important to be safe from the injuries caused by damaging the battery or battery defected of lithium as they can be preventable with the help of some of safety guidelines: 100% know that lithium batteries, chargers and associated equipment were tested in accordance with appropriate testing standard (UL 2054) as it was applicable and certified by the NRTL or Nationally recognized testing laboratory while changing the batteries and chargers for the specified electronic device that the user used. Ensuring that these devices were specifically designed and approved for safety and were ready to be purchased from the device's other manufacturer. It was imperative to remove lithium powered devices and batteries from the charger once it is 100% charged and ready to use. If these batteries were used in generalized day to day product or in our product for the project then it was imperative and crucial to inspect them for signs of damage, for instance hissing, leaking, budging, cracking, temperature rise, and any leakage of smoke. It was important to immediately remove a device or battery from the service and put it in an area far away from flammable materials when these circumstances rise. The methods mentioned helped prevent fire and/or fire hazardous situations in any environment or surrounding.

NiMh batteries were utilized in over 95% of HEVs and other major manufacturers have so far invested a lot in the past ten years. The major advantage from a manufacturing perception is the safety and injury-free environment of NiMh batteries compared to lithium batteries, and, so far, no incidents have occurred or been reported in the press before. NiMh batteries were preferred in the real-world or industry and consumer applications rose due to their design flexibility (like raning from 40 mAh to 250 mAh), environmental acceptability, very low maintenance, high strong power and energy densities, cost and importantly: the safety (discharge and in-charge modes, most especially at high voltages). NiMh batteries were currently prices at an amount of $250 to $1500 per kWh, so the total price of the battery pack provided for a hybrid (Toyota Prius, though the newer the models use 5.2kWh lithium battery packs) varied between $600 and $4000 per vehicle. The NiMh batteries were patented in 1986 by Stanford Ovshinksky who was the founder of Ovonics, when recharging hydrogen storage models. Ovshinsky also described NiMh battery as the hydrogen ion or protonic battery by an analogy with Lithium atteries as the NiMh electrochemical reaction involved the transfer and insertion of H+. The major components of NiMh batteries consisted of anode of hydrogen absorbing alloys or MH, a cathode of nickel hydroxide or NiOH2 and a potassium hydroxide or KOH electrolye Nickel-metal hydride batteries as in most aspects of their manufacture and design, the manufacturing procedure was similar to the NiCd batteries. The key difference was the change of placings of the negative cadmium based electrode with an electrode using a hydrogen storing metal alloy. Nearly all NiMH batteries operating in this field in these times employed a very rare earth Mischmetal called the nickel based metal allow (MmNi5-type) with some chunks of cobalt, manganese and aluminum. The statements in the previous aspect of nickel-based substances including the hazardous risks of the chaotic electrolyte also applied for the NiMH system. The absence of the cadmium made recycling process of spent products. The soft-chemical process of Ni-MH batteries have also been studied. The chemical composition of the recovered material from the large-format NiMH cell is provided to show how it can be recyclable.

Nickel Cadmium or NiCd battery cells were completely instilled with electrolytes and often needed to be handled, stored and moved with vents facing north or upwards. It was important to avoid using the NiCd batteries in direct sunlight, high temperature, high humidity and dry conditions. The NiCd batteries could be stored in a cool and dry area where temperature ranged from 10 degrees Celsius to 30 degrees Celsius which was 50 to 86 degrees Fahrenheit and with a humidity between 44% and 86%. Ensuring that one does not connect a positive terminal to a negative terminal with electrically conductive materials. Always stored and operated the Nickel-Cadmium batteries in separate rooms where the lead-acidic batteries can be stored and used. Also, keep NiCd batteries away from water in cooling areas with well-ventilated conditions. Do not put any other materials of use on the top of the batteries as given the features of any specific battery type and how it can be performed, that certain battery can be beneficial for our product as it performed efficiently and last longer in any circumstances comparted to the usage of Lithium batteries.

4.1.11 – Battery Charge Controllers

Battery charge controllers were vital to ensure that our batteries worked desirably for long hours so that the product was effective in full use. Battery charge controllers or charge regulators or battery regulators were mainly used to charge a solar deep cycle battery safely at the accurate charge rates and to protect the battery from overcharging or it might overheat and destroy itself. If it does destroy itself, then the battery won’t be able to function properly which won’t help the product in any situation at all. There were a significant variety of solar charge controllers available such as: AC chargers, solar lighting controllers, etc.

Solar charge controllers were used to optimize the charge from solar panels, regulate the charge and protect the batteries overall located in the solar power systems. The solar lighting controllers comprised of both a solar charge controller and a lighting controller which were programmable in one-unit direct charge or DC lights were ran directly off the solar lighting controller at accurate scheduled period of times which potentially eliminated the requirement of a load controller or an alternative timer. Alternative current or AC battery chargers were used as an AC source. An AC source like a wall outlet which was used to directly charge a direct current or DC battery bank. There were a lot of different versions available which allow fast charging, or an input/output of different voltages. The load controllers were significantly capable of being solar charge controller or a direct current (DC) load controller, or a diversion load controller. It was key to determine which load controller does it want to operate as, but since our battery was charged through the beams of the sun rays directly falling on the solar panel, then it was beneficial for the direct current load controller to enact like a solar charge controller, because the more power it contained, the product performed its tasks for the users for longer durations. In order to make sure that the battery charged controllers worked properly, then the low voltage disconnected or the LVD was an important voltage that allowed one to connect to a DC source load which was the same voltage as the battery bank, which allowed the controller to be turned it off when the battery bank power was low. This process allowed the battery bank to be protected from the battery becoming completely drained which is also known as deep discharging.

4.1.12 – Power Regulation

A power or a voltage regulator was a circuit that maintained a fixed output voltage, even though it wouldn’t matter how the input voltage or load conditions changed. The way this worked, was where a voltage regulator maintained the voltages from a power supply within a range that was compatible with other electrical units. Voltage regulators were often used for DC/DC power conversions, AC/AC or AC/DC performed conversions too. There were two types of voltage regulators: Linear and Switching. Both types of regulators controlled and regulated a system’s voltage, however linear regulators operated with very low efficiency whereas switching regulators operated with very high efficiency. All, if not most of the input power gets transferred into the output without dissipation.

Linear regulators used devices such as BJT or MOSFET which were active pass and were controlled by a high gain amplifier which was operational. In order to maintain a constant output voltage, the linear regulator modified the pass device resistance by comparing the internal voltage reference to the sampled output voltage. This had driven the error to zero. It was important to notice that linear regulators stepped-down converters, so the output voltage was always below the input voltage. There were few regulators that gave a few advantages which were generally simpler to design, cost-efficient, and offered low noise as well as low output voltage ripple. MP2018, which was a linear regulator, required an input and output capacitor to operate. Their simplicity and reliability made them intuitive and simple devices for engineers.

Some of the known parameters were know when we used voltage regulators as the input voltage, output voltage and output current. There were other parameters considered but it depended on the application. Some of these parameters included: quiescent current, switching current, thermal resistance, and feedback voltage. We used quiescent current when the efficiency needed to be increased during the light-load or standby modes. Feedback voltage was another parameter that needed to be considered importantly because it found out the lowest output voltage that the voltage regulator can support.

In order to get the right results, the right voltage regulator needed to be selected for the design. So, to select the right voltage, the designer was first understood when the key parameters were used. The key parameters consisted of Vin, Vout, Iout, etc. There were other system priorities considered too such as: performance, efficiency, cost, power good indication and/or enable control. Once these problems or requirements were defined, a parametric search table was used to find the best device to meet the requirements which made things simpler in functioning. The parametric search table was nothing but a valuable tool for designers. This tool provided different features and packages to meet the required parameters.

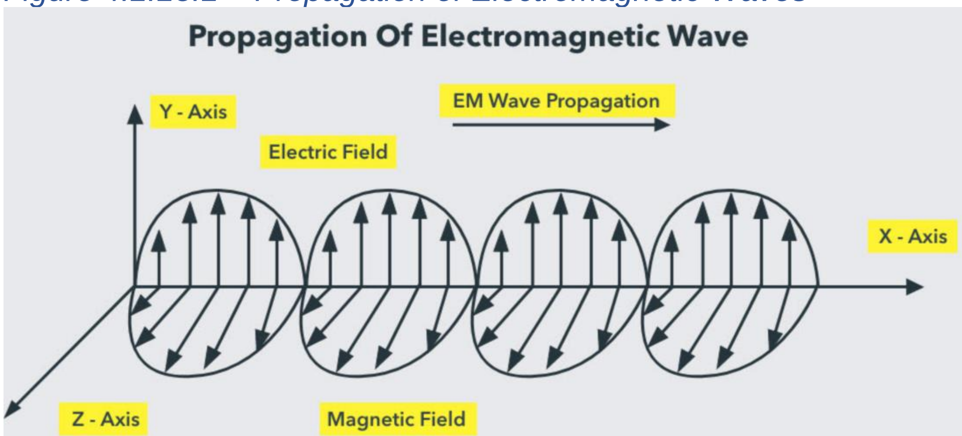
4.1.13 – Wireless Communications

As evident from recent advances in technology in our world, wireless communication had been the fastest growing and expanding technology over several networks and nodes in this world. It is one of the most vibrant technological areas in regarding the communication field. Wireless communication is a process of transmitting information from one point to another without utilizing any direct connection through wires, cables or any other physical mediums. This sort of communication is seen with Bluetooth available in portable devices such as mobile phones, portable computers such as laptops, tablets, iPads, etc. Usually, information is transmitted from one transmitter to a receiver which are placed over a limited distance. With the aid of a wireless communication, the transmitter and receiver can be stored anywhere between a particular amount of distance which could range anywhere from a few meters to a thousand kilometers. In today’s world, communication is key. It has always been vital regardless of any century we lived in. Without proper communication, no tasks can be done. So, communication is highly important to thrive today regardless of the career that one worked in. Due to which, wireless communication had integrated into our lives as a key component.

Any type of communication system, wired or unwired, can be guided or unguided. When wired communication was examined, the medium was a physical path such as cables, optic fiber links, axial cables, etc. These materials were used to guide the signal to spread from one point to another. This already seemed like a lot of work and can lead to serious conjunctions in terms of wiring and connecting devices to function properly. So, wireless communication was the optical way to ensure proper connection between two devices. The mesh networks used a wireless connection in order to connect the circuit board which collected different data points of the area, which then were stored into a database, and that displayed the information onto the webapp for the users. There wasn't any other physical medium used. The circuit board had its own wiring and functioning done for the different components which were connected to a database wirelessly. The signals from one device to another were propagated through space. Space only permitted for signal transmission without any guidance, the medium used in these sorts of wireless communications, or any wireless communications was called the unguided medium. But how do these signals travel through space?

Even though there were no cables or any other physical mediums to send signals, antennas were used to accomplish the transmission and reception of signals. Antennas were nothing but electrical devices that transformed the signals acquired from electrical to radio in the form of Electromagnetic or EM waves and vice versa. These electromagnetic waves propagated through empty space. So, both the receiver and transmitter consisted of an antenna. The electromagnetic waves were important because they transported the electromagnetic energy of the electromagnetic field through empty space. Electromagnetic waves consisted of Gamma Rays (y-rays), X-rays, Visible light, ultraviolet rays, microwave rays, infrared rays, and radio waves. Electromagnetic waves are also called radio waves which were also used in wireless communication to transport the signals. These electromagnetic waves consisted of both electric and magnetic fields in the form of time varying sine waves or sinusoidal waves. Both fields were oscillating perpendicular to each other and direction of the propagation of the electromagnetic wave was perpendicular to both fields. Scientifically or mathematically, an electromagnetic wave was explained using Maxwell's equations. These equations were extremely helpful to use the waves effectively when we connected the nodes in our mesh network to accurately display the retrieved data from the system.

Figure 4.1.13.1 – Propagation of Electromagnetic Waves



The picture displayed above, Figure 4.1.13.1 showed a better idea of the type of calculations which were used to attribute for the electromagnetic waves to ensure better signals while connecting devices.

Wireless communication helped in mobility. Other than mobility, wireless communication helped in offering flexibility and ease of use, which made it increasingly better day by day. Wireless communication like mobile telephony could be created anywhere and at any instance with a considerable high performance. Another important point to consider was infrastructure when it came to wired communication systems. The installation and setup of infrastructure was expensive and was also very time consuming. Whereas the infrastructure for wireless communication could be installed very easily and came with a low cost. Wireless communication could be a viable option when emergency situations arise. The setup of wired communications was time consuming and tough to untangle whereas there isn't much to worry about when it came to wireless communication.

4.1.14 - GPS

The GPS or a Global Positioning System is a satellite-based navigation system consisting of at the very least 24 satellites. GPS can be constructed in any weather conditions, anywhere in the world and operated every second of the day without any additional setup. The satellites were placed into orbits for military use by the US gov initially but then civilians used it in the 1980s and beyond.

GPS satellites circle or revolve around our planet Earth twice a day in a precise orbit. Each of the satellite transmitted a signal and orbital parameter that ensured that the GPS devices were able to compute and decode any accurate location of the satellite orbiting. Then, the GPS receivers utilized this information which calculated the user's precise location. Eventually, the GPS receiver was able to measure the distance from one point to each satellite by the amount of time it took to receive a transmitted signal. With more measurements from more satellites, the receiver could determine a user's position and showed the results electronically to measure any running route, finding the way back to home or any adventure anywhere.

First the 2-D position was calculated (the latitude and longitude) and that tracked the movement which was a GPS receiver which must be locked to a particular signal of at least 3-4 satellites. If there were 4 or more satellites then the receiver could tell one's 3-D position (latitude, longitude and altitude). Usually, a GPS receiver does track 8 or more satellites, but it all depended on the time of the day where one was located. In today's world, GPS systems are very accurate due to their multi-channel designs. The receivers are quick to lock onto satellites when it is turned on. It maintained a decent tracking lock in dense tree surroundings or in urban settings with tall buildings. What and how was the signal sent? The gps satellites transmitted it at least 2-3 low-power radio signals. The signals moved by line of sight wherein it passed through cloud, glasses, and plastic, however it didn't go through the most solid objects, such as buildings and mountains. There were three types of information that a GPS signal gives: Pseudorandom code, Ephemeris data, almanac data. Pseudorandom code is an ID code that finds out which satellite is transmitting information. The ephemeris data is needed to determine if a satellite's position and shows important information about the health of a satellite, current data and time. There were many factors that affect the GPS signal and accuracy include:

Ionosphere and troposphere delays where satellite signals slowed down as they pass through the atmosphere and then the GPS system used a built-in model to partially correct for any errors that arise. The signal multipath was another factor that affected the GPS signal and accuracy where the GPS signal reflected off objects like tall buildings or very large rock surfaces before it would reach the receiver which improved and increased the travel time of the signal and created errors. Receiving clock errors was another factor which affected the GPS signal and accuracy where a receiver's built-in clock had a little off timing errors because it was less than usual accuracy than the atomic clocks on the GPS satellites. Another factor to be considered was the number of satellite visible where the more satellites a GPS receiver could perceive the better the accuracy was. When the signal gets blocked, there were position errors or potentially no position reading at all. GPS units usually did not work underwater or underground, but there was a new high-sensitivity receiver that could track some signals when they were inside buildings. Satellite signals were more effective though when satellites were placed at wider angles relative to each other.

We intended to place a GPS system into our product as well. The nodes were essentially pinpoints on the map depicting different cities. The GPS system was able to track all the information that was gathered from that specific node and then all the information was transferred into a data server which was discussed earlier. The data server was made using python. GPS system was important because it was reliable in terms of delivering accurate information to a specific server. So, if we placed a GPS then the users would have a convenient and easy interface to use it.

4.2 – Part Selection

4.2.1 – Microcontroller

We needed an MCU that had a good selection of both digital and analog general purpose input output (GPIO) pins. Next, we considered the communications standards the MCU supported. The most common of these being SPI, I2C, and Serial. We also considered the wireless communication standard; this was built-in or external. We also took note of the processor memory and speed as this was important as we temporarily stored sensor data or participated in the mesh communication. Lastly, we needed to consider the programming language and supporting software. The native language to each of the microcontrollers we researched allowed different features. Some examples of these features were object-oriented design, code portability between microcontrollers of the same family and faster prototyping in certain languages. As for the supporting software, we needed to consider the ease of development. This project had a limited scope in terms of time and budget. With this, a family of MCUs with an easy-to-use software suite delivers a strong value proposition. Ease of use was subjective though the basic ground rules we applied. First, the IDE should be capable of compiling and uploading the code to the device with minimal steps and do so reliably. Second, there should be support for external libraries. These libraries allowed us to interface with sensors without the need to develop our own low-level code. Lastly, the development environment was stable and easy to set up.

Our initial design concept used two separate microcontrollers. One microcontroller handled communications while the other handled the sensors. We expected this to be the best option under the assumption that each sensor would need its own set of pins to communicate with the microcontroller. Further research had revealed that many of the sensors could share the inter integrates circuit bus. As a result, we searched for a single microprocessor to handle all the functions for a node.

We quickly selected the ESP32 system on a chip as the heart of our nodes. There were a few key reasons for this. First, the ESP32 integrated wireless communications into the same package as the microcontroller. This allowed us simple access using a library of functions provided by the manufacturer of the ESP32. The second reason was the amount of general-purpose input output pins. The ESP32 provided more pins that we expected to use by a wide enough margin to allow for potential feature creep given the time and budget. The third reason was the performance. The ESP32 offered a dual core processor with 512k bytes of memory space. These two specifications wee important when we were communicating across the mesh network and gathering sensor data simultaneously. The fourth reason was the availability of both development boards and raw modules for programming. Development boards were widely available for creating our system prototypes. When it was time to deploy our system, the ESP32 module was purchased on its own. Additionally, there were external programmers available allowing us to avoid adding the complexity and cost of on-board programming to our design. The fifth and final reason we selected the ESP32 was its compatibility with the Arduino family of boards. The Arduino IDE was simple and reliable. Also, the Arduino family of boards had a wide user base with extensive documentation and libraries. All of these factors combined to alleviate some stress points during the early development and the late-stage deployment of our project.

4.2.2 – Anemometer

While investigating each sensor in our design, we considered the build versus buy framework. Our project was self-funded by our group members requiring cost to be an important consideration. We also had a finite window of time for development of the subsystems and the project. These previsions gave us strong arguments for both sides of the build versus buy proposition. In the case of the anemometer, it quickly became clear that we needed to investigate the probability of building the sensor ourselves. The cost for an anemometer of any applicable type started as the most expensive single component and way above our budget margin from there. As a result, we invested time and developed our own lower cost anemometer.

As we begin the development process, we needed to reflect on our research and selected the most likely candidate for purposes. We immediately eliminated one option: the hot wire anemometer. This type of anemometer had a few insurmountable challenges when considering our design requirements. The first was that the core functionality of this type of anemometer required the heating of a wire for detection of airflow. Heating was a generally energy intensive process. Additionally, there was a requirement for a mechanical design to prevent the generated heat from propagating to other components of the device. The final issue was that the measurement range of the hot wire anemometer did not scale linearly. This meant that the device could measure very fine variation in a light breeze then became overwhelmed in a strong gale. For this reason alone, the hot wire anemometer was not a good fit for highly variable natural weather conditions.

The elimination of the hot wire anemometer left us with two diverging paths. The first was the tried-and-true mechanical design consisting of a wind vane for detecting direction and a spinning apparatus for detecting speed. This path consisted of both the cup style and the windmill style as the electrical implementation of both styles was similar. The second path we considered was the ultrasonic anemometer. This path had a few key benefits over the other. First, the mechanical design was much simpler. There were no dynamic elements or forces to consider. The mechanical design was also relatively simple and compatible with rapid iteration using 3D printing. This leads into the second benefit, the ultrasonic anemometer which had no moving parts. Our system was deployed in areas that may hinder regular preventative maintenance. Reducing the complexity and need for preventative maintenance was an important consideration. A further benefit to the ultrasonic anemometer was that a single apparatus provided both wind speed and direction measurements. This was not a common trait of anemometers, there were typically multiple sensors needed to give both measurements. Lastly, the components used to construct an ultrasonic anemometer were inexpensive. Costing only a few dollars compared to tens or hundreds of dollars for an off the shelf offering. These benefits combined necessitate further investigation.

Ultrasonic Anemometer Investigation

As we began our investigations into the specific methods for building an ultrasonic anemometer it may be beneficial to first visualize how this subsystem works.

Figure 4.2.2.1 – Ultrasonic Anemometer Functionality

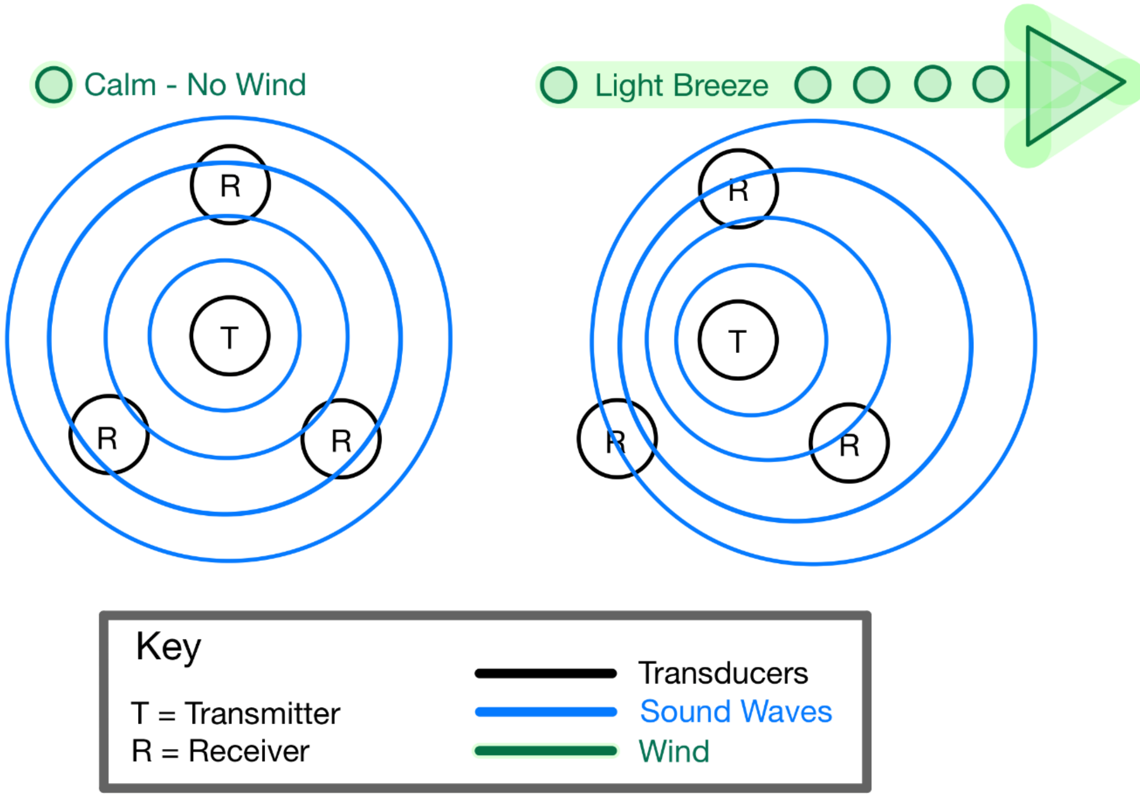
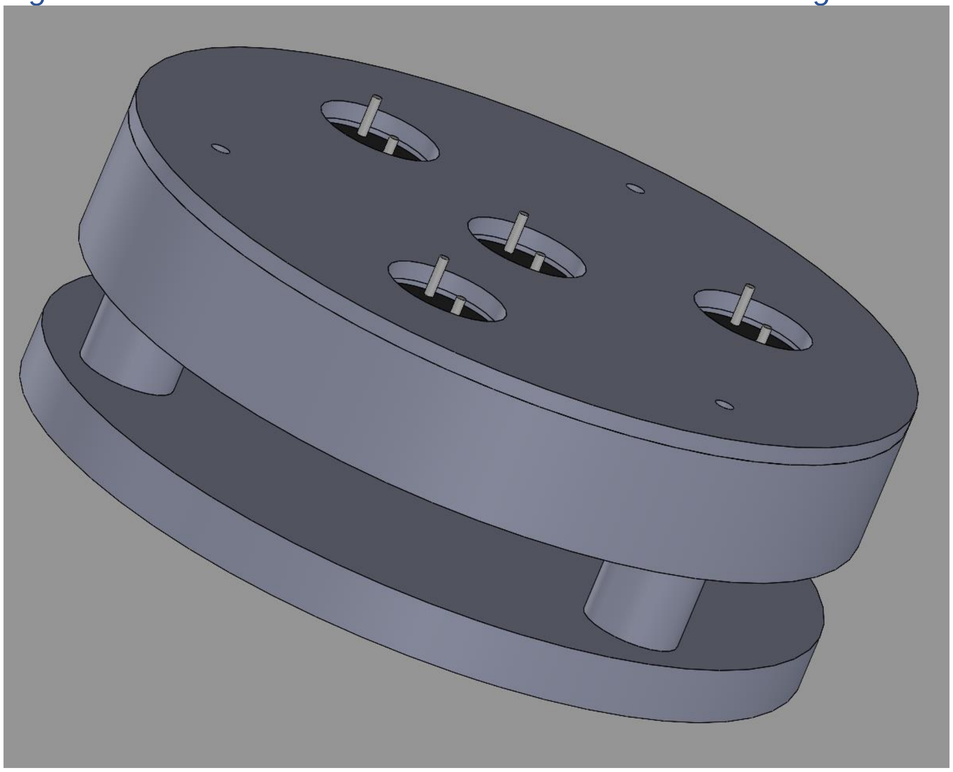


Diagram 4.2.2.1 showed how the wind passing through the anemometer affected the sound waves. The diagram depicted two different wind conditions. On the left, we saw the conditions in a calm environment. There were no distortions or changes to the form of the sound waves. On the right, there was a light breeze propagating from the left of the sensor array to the right. In this condition, we observed that the sound waves became compressed when observed from the far-left receiver. In contrast, the waves were expanded when observed from the far-right receiver. This was a representation of the core functional mechanism of this anemometer, the Doppler effect.

Now that we had a visual idea of the effects the sensor experienced, we needed to consider a means to measure these effects. As we began to investigate the specific methods for interpreting this data, it was apparent that the available information on this subject was limited. There was a single readily available resource for a complete solution. Beyond this single source, there were a handful of research documents and available off the shelf components that we used for reference. Keeping in mind, one of our core design limitations were the timeframe, which we attempted to build a proof of concept using the single source. This supplied us with an in-depth understanding of the feasibility of this anemometer design.

Figure 4.2.2.2 – Ultrasonic Anemometer Mechanical Design

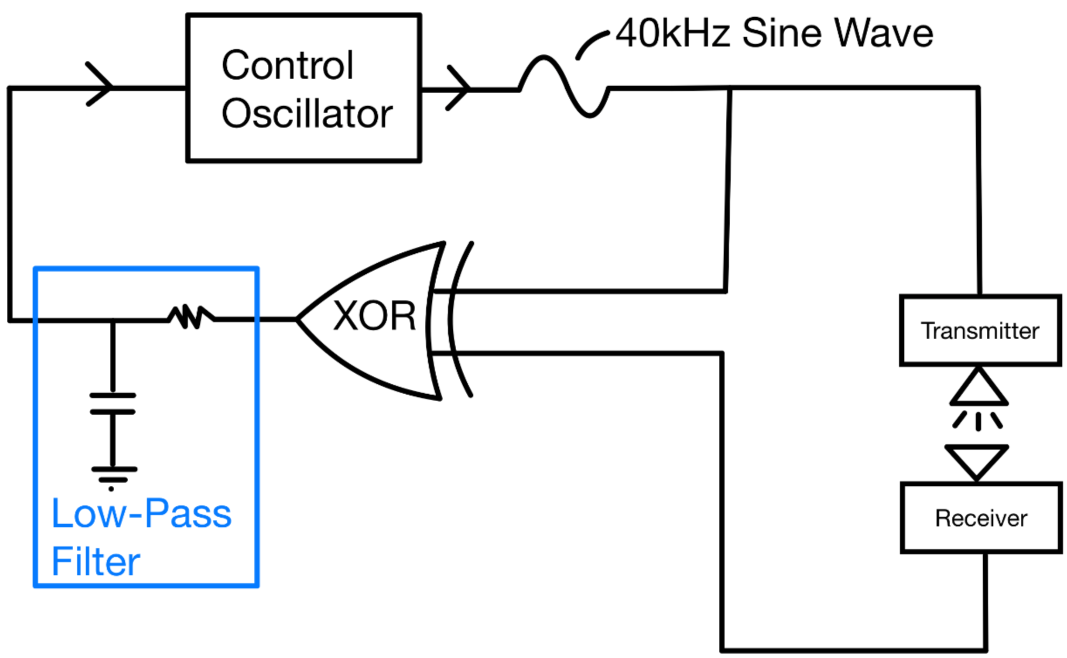


The functionality of the ultrasonic anemometer depended on the reflection of sound in a uniform manner from the transmitter to the three receivers. This required the use of a planar reflection surface. Additionally, the reflection surface must be the correct distance from the transducers. If the reflector was too far, then the sound energy may have escaped the system before it was properly measured. If the reflector was too close, then the sound energy was overpowered over the wind reducing the sensitivity of the system. To reduce the impact of these and other mechanical factors, we bult this prototype, we created a computerized model of the mechanical design. A preview of this design may be seen in Figure 4.2.2.2 as a capture from within the Solidworks design program.

The first step in the mechanical design was to carefully measure and model the transducers themselves. With this model created, we then created the arrangement of the four transducers. The transmitter was placed in the center of the design. We then used the transmitter as the reference point for the three receivers. The center of each receiver was placed 25mm from the center of the receiver. Each receiver was placed equidistant along a circular path centered on the transmitter. Once the transducers were all in place, we designed a housing to retain them. This housing was comprised of three pieces, the body holding the transducers in the center, the reflector on the bottom and the retaining lid on the top. The assembly was designed with 3D printing in mind. This consideration maintained that we conceptualize the model to be created one layer at a time. Now that we had a mechanical design; we changed our focus to the electrical functionality.

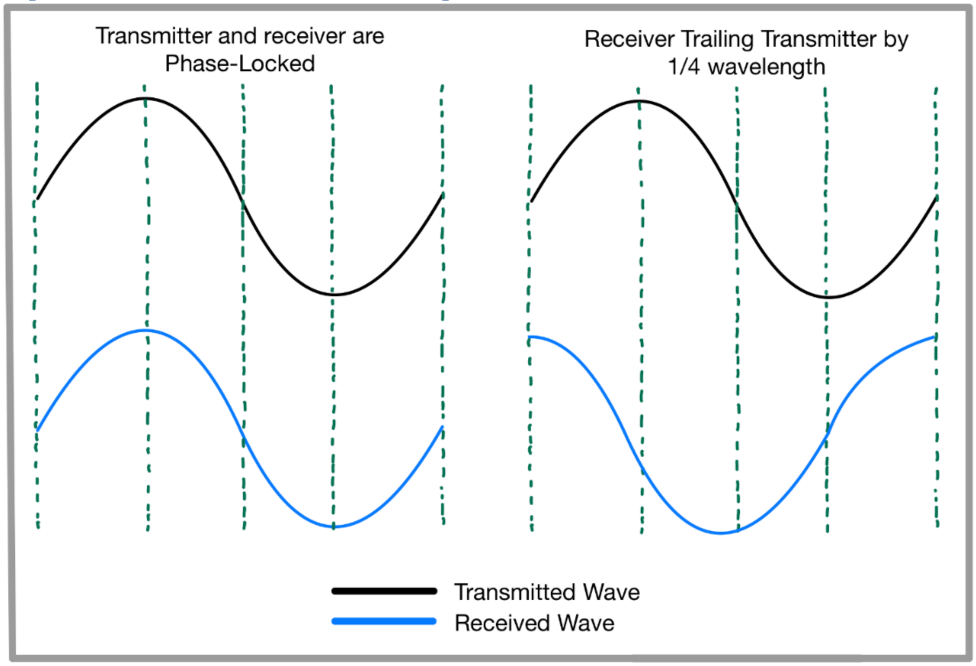
For the ultrasonic anemometer to function, there needed to be a stable emission of 40kHz from the emitter transducer. The 40kHz specification was important as that was the frequency these transducers were tuned to and where they worked the best. The stability was important as we measured small changes as a result of wind flowing through the sensor. If the transmitter had a fluctuating signal itself; errors may be introduced into the system. There were a few methods to generate the output signal, for this prototype we implemented a phase-locked loop. A basic example of this loop is observed in Figure 4.2.2.3.

Figure 4.2.2.3 – Phase-Locked Loop



A phase-locked loop consisted of three basic components. The main component was the control oscillator. This oscillator took a direct current voltage as its input. This DC voltage was the control voltage and contributed to setting the output oscillation. The output of the oscillator was tuned by its inputs to be a set frequency. In our case, the target frequency was 40kHz. The output oscillation was used to create a feedback loop that had a high propensity to resist any deviation from the tuned frequency. The output was used to drive the transmit transducer. The output was also fed into one side of an exclusive or gate. One of the three receiver transducers were connected to the other side being exclusive-or gate. This created the output for the feedback to the control oscillator.

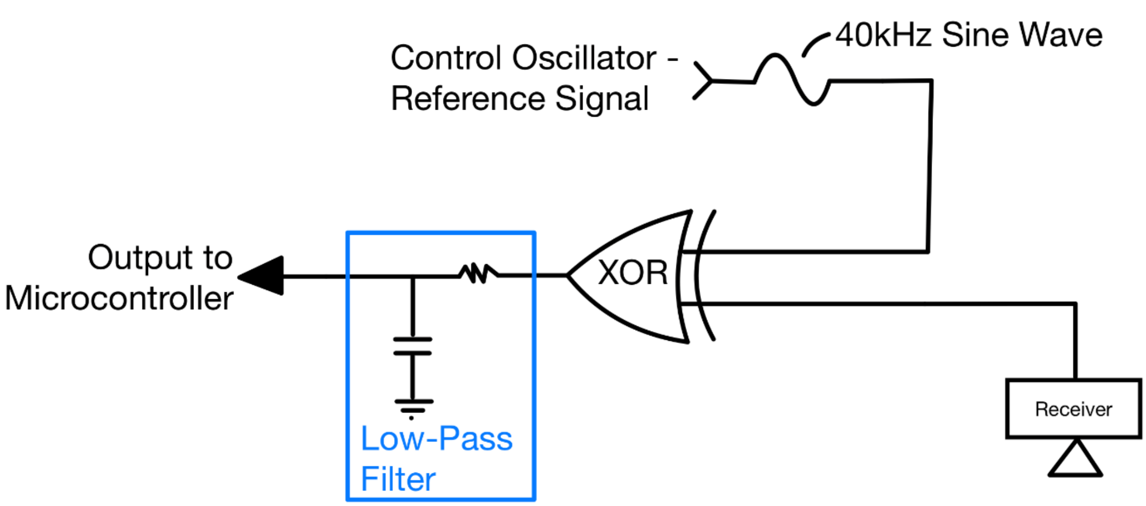
Figure 4.2.2.4 – Phase-Lock Signals



The sine two sets of sine waves depicted in Figure 4.2.2.4 showed the action of the phase-lock loop. On the left was the stable condition where the loop was locked in phase between the output and the input. In this condition, the output of the exclusive-or gate was zero because both waves match each other. The right showed the transmission which always stayed stable at the preset frequency even if the received wave had shifted in phase or frequency. When the condition on the left occurred, the output of the exclusive-or gate began to periodically transition from zero to one. This was a result of the waves being out of phase. This was the mechanism we used to adjust the control voltage to the voltage-controlled oscillator. We observed an example of the real exclusive-or output in Figure 4.2.2.5. The output of the exclusive-or was then passed through a low pass filer with the result being an average volage between the voltage input and reference. This average increased and decreased over time as the square wave was high and low respectively.

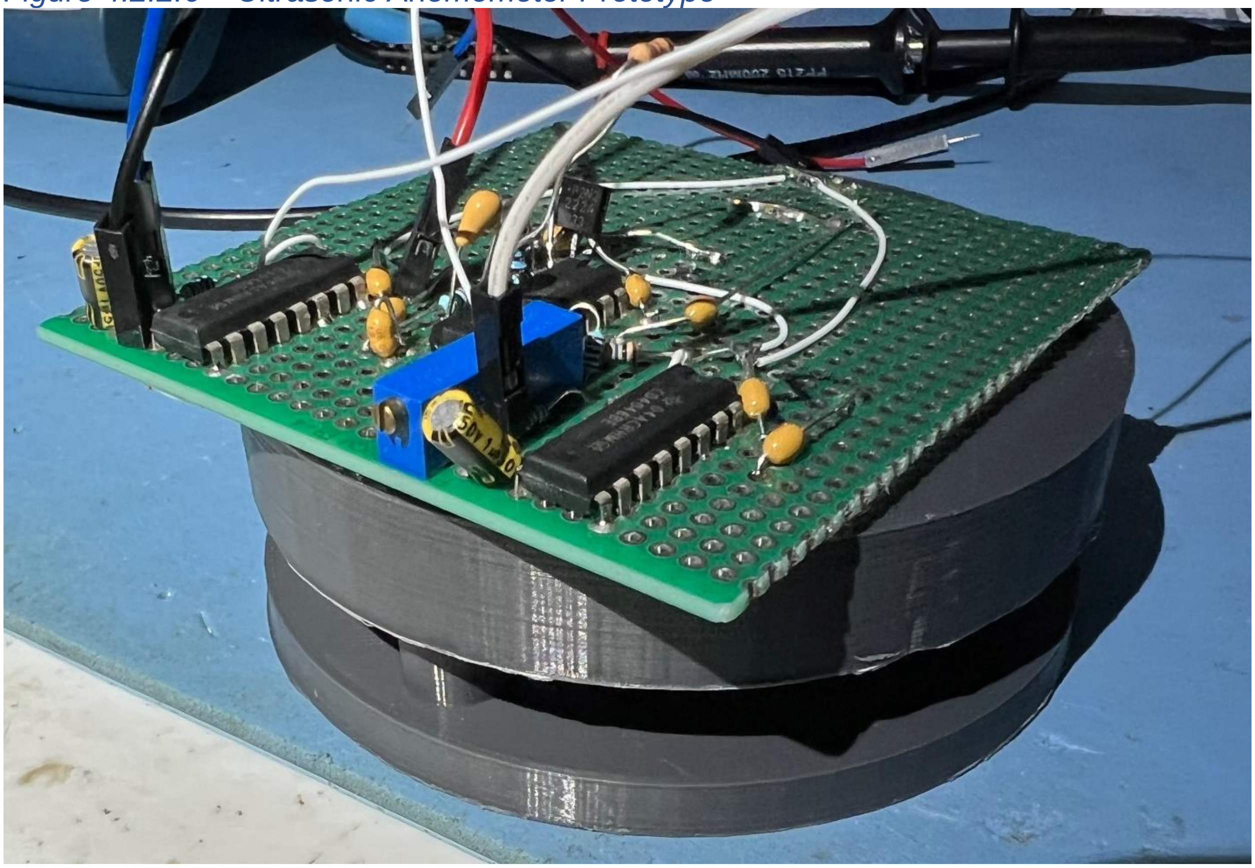
Figure 4.2.2.5 – Output of the Exclusive Or

Figure 4.2.2.6 – Ultrasonic Anemometer Receiver



The phase-locked loop used one of the three receivers as the feedback to maintain the 40kHz signal. That left the other two receiving transducers to be used as inputs to the microcontroller. Figure 4.2.2.6 showed how each of the inputs to the microcontroller was generated. The same basic principle applied from the input to the feedback loop. The 40kHz signal became one side of the exclusive-or gate. The other side of the exclusive-or gate was the exclusive-or gate as the receiver. When passed through the gate together, the resultant signal was a square wave. This wave was fed through a low pass filter. The filter removed the alternating current component and supplied a direct current voltage to the microcontroller. This direct current voltage was the average of the square wave. This voltage was measured by the analog to digital converter present in the microcontroller. This measurement was then used to extrapolate the wind conditions.

Figure 4.2.2.6 – Ultrasonic Anemometer Prototype



Before we were able to connect our prototype ultrasonic anemometer to an actual microcontroller, there were a few challenges we attempted to overcome. The first was that the reference circuit was designed to work with a five-volt logic level microcontroller. The microcontroller we were using had a maximum typical input of 3.3v and was not 5v tolerant. With some reading of the datasheet for the CD4046B phase-lock loop integrated circuit, we began to adjust the input voltages. This was accomplished by adjusting external capacitors and resistors. These components were used to set the frequency of the phase-locked loop. This also meant that the input voltage and voltage reference remained stable during measurements. This was an important factor to be considered as we were deciding to use this anemometer. We adjusted the component values to reach an input voltage of 5 volts with an output of roughly 3 volts on both output lines. During the adjustment and investigations, we located a second challenge, the variability of the output was well beyond what we expected. The measurements we were attempting to make in this system were microscopic. The changes in voltage were expected in the order of tens of millivolts. Instead, we saw a minimum and maximum voltage readout with a delta of three quarters of a volt. This was also resulting in the maximum reading of our output exceeding the maximum of our microcontroller. These measurements were taken using the minimum and maximum function on our digital voltmeter and were visible in Figure 4.2.2.7.

Figure 4.2.2.7 – DC Output Voltage of Ultrasonic Anemometer

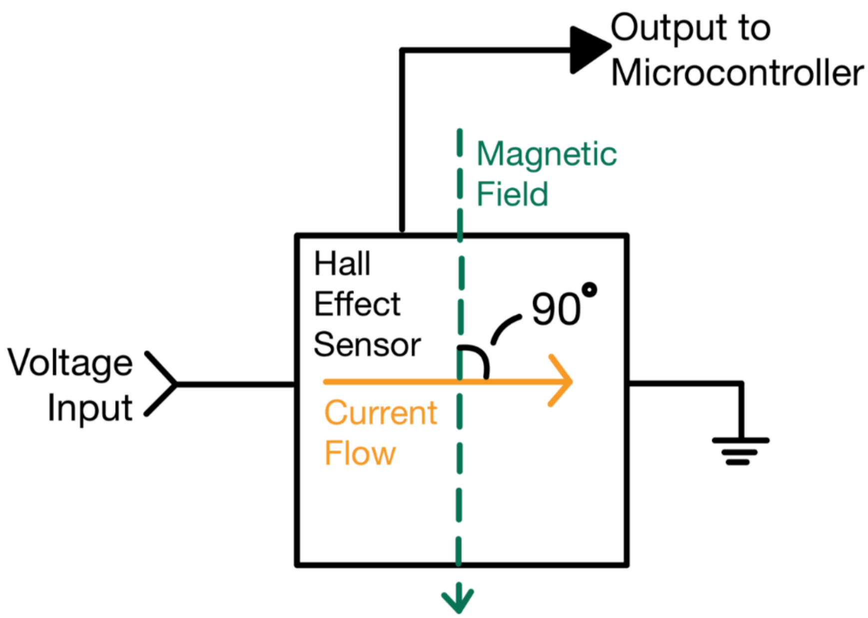
At this point, we stepped-back and considered the timeframe design constraints. We believed that this design had the potential to function properly with more tuning of the component values. We also used the given values that may cut back on the variation in the signal. This introduced much more complexity to the design with the need for analog line level translation and a 7-volt supply. These issues compounded to lead us to the decision to move toward another solution for the anemometer.

Cup Anemometer and Wind Vane Investigation

The next style of anemometer we investigated was the cup type with a wind vane. We were investigating both types of wind sensor because the cup style gave us the wind speed and the wind vane gave us direction. The first sensor we focused on is the cup style wind speed sensor.

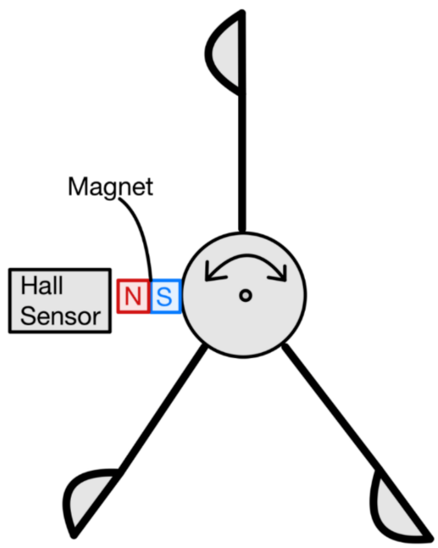
The circuitry design of the cup anemometer was much simpler than that of the ultrasonic anemometer. In this design, we were using a single sensor, the Hall effect sensor. A Hall effect sensor functioned much like a switch. The Hall effect sensor resisted the flow of current under nominal conditions. When a magnetic field was placed in proximity to the hall effect sensor, the switch closed and began to allow the flow of electrons. It was important to note that the optimal positioning between the magnet and the sensor was such that the magnetic field lines passed through the Hall effect sensor at a 90-degree angle to the direction of the current flow. For a basic reference of the Hall effect sensor, see Figure 4.2.2.8.

Figure 4.2.2.8 – The Hall Effect Sensor



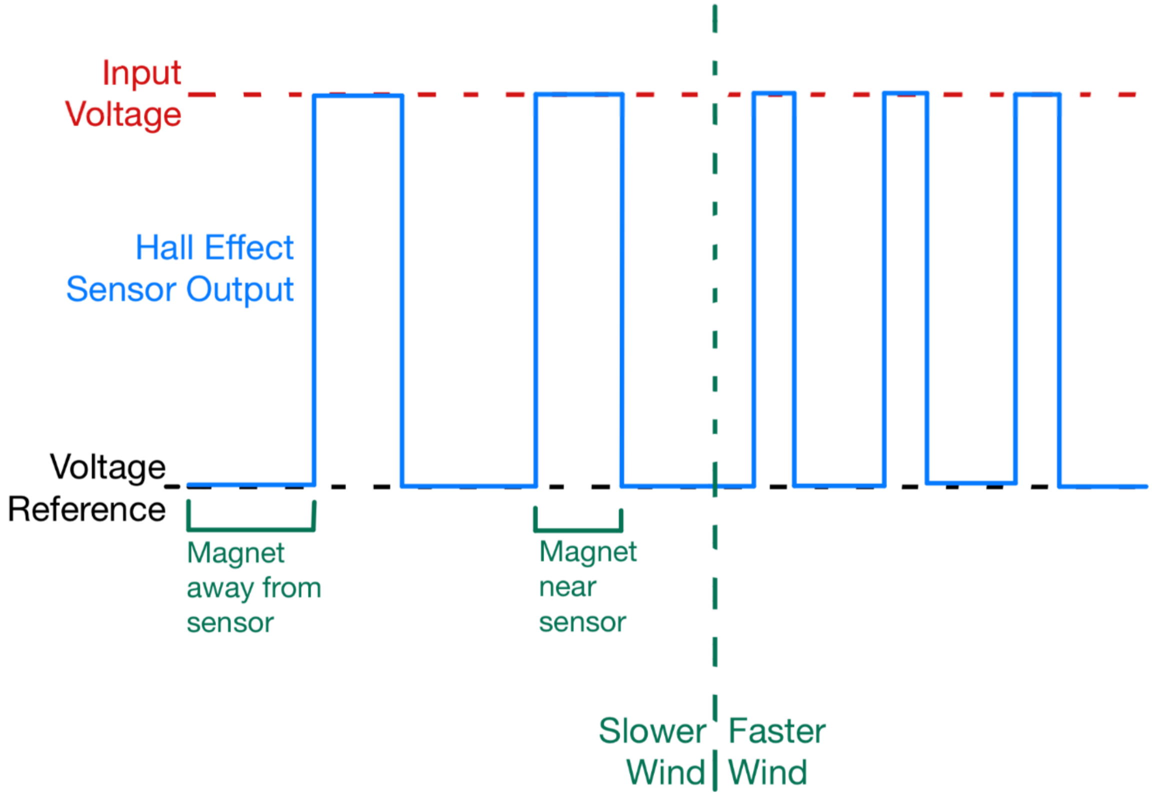
When the magnetic field activated the Hall effect sensor, the output to the microcontroller became nearly equal to the voltage input. We utilized this characteristic to create a signal to represent the wind speed by leveraging the mechanical design. As we discussed in the research section for anemometers, cup style anemometers relied on the ability to move freely. Figure 4.2.2.9 showed a top-down view of the basic mechanical design of the cup style anemometer. The cups were attached to a hub. The hub was then attached to a low friction bearing and allowed to spin freely. Also attached to the hub was a magnet. The Hall sensor was mounted adjacent to the hub. The Hall sensor was fixed in position. As the hub spun, the magnetic field activated the Hall sensor producing a signal output.

Figure 4.2.2.9 – Cup Anemometer (Top-Down View)



As the magnet passed the hall effect sensor, the sensor became active. When the magnet moved away, the sensor became inactive. Using a comparator, a filtered square waveform was generated. The transition from active to inactive was nearly instant. The result was the spinning of the hub of the anemometer producing a square wave output from the Hall effect sensor. The pulse frequency and width was used to extrapolate wind speed.

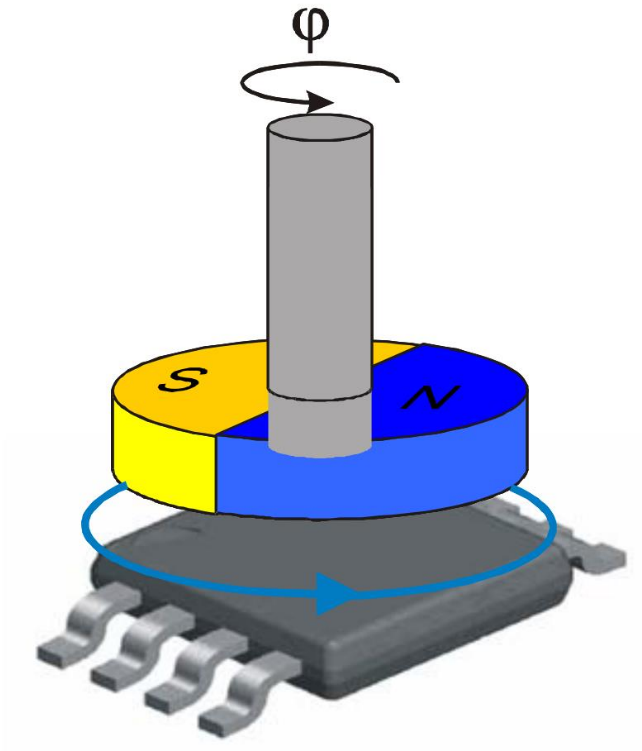
Figure 4.2.2.10 – Hall Effect Sensor Example Output



We supplied the Hall effect sensor with the high logic voltage of the microcontroller and connected both to a shared ground. The resulting signal produced by the Hall effect sensor was a square wave from logical low to logical high. This gave us the ability to interpret the signal as a digital input to the microcontroller. The microcontroller was able to determine the pulse width and frequency. This data at different known wind speeds allowed us to extrapolate the wind speed at unknown speeds. The complexity of this system was much lower than the ultrasonic anemometer. Next, we conducted practical testing to ensure our theory holds true in the real world.

The other half of creating this anemometer required a wind vane to determine direction. The wind vane was connected to a free spinning shaft. Detecting the absolute position of a shaft proved to be challenging. The main issue was that there was no reference point to measure against after the node is power cycled. We encountered this issue if we used a Hall effect sensor like the cup anemometer or a traditional encoder. We were able to count the number of turns of the shaft but were not able to ascertain its position in reference to the node or other objects. Our objective with the wind vane was to detect the compass azimuth direction the wind was blowing in. We considered a few methods to tackle this challenge. The first consideration was to attach a compass module to the anemometer itself. This would fix the direction of the compass to the direction of the wind vane. The issue with this was the electrical connection to the compass became a limiting factor. The vane needed to spin freely in any direction. This meant that the connection to the compass would need to do so as well. This required contacts and spring-loaded wipers to connect from the shaft to the sensor. Which introduced two additional issues, the contacts needed to be periodically cleaned. Also, the friction force of the interface between the contacts and the wipers was reduced for the accuracy of the wind vane. The next option we considered was using a continuous potentiometer. A potentiometer functioned by passing a wiper over a linear resistance carbon contact. As the wiper traversed the contact, the resistance from the input to the output of the potentiometer increased or decreased. A typical potentiometer had a stop preventing the wiper from rotating 360 degrees. A continuous potentiometer removed this stop. This allowed the wind vane to spin freely. We then passed a known voltage through the potentiometer and measured the voltage drop across it. This was done by using the analog to digital converter on the microcontroller. The functionality and implementation of this option was straightforward in terms of hardware and software. There were however two issues with this approach. The first was that there was a non-trivial amount of force needed to start and turn the potentiometer. This reduced the accuracy of the wind vane. The second Issue was the cost. The cost of a continuous potentiometer was high enough that we began to approach the off the shelf options for anemometers. This brought us to the final option we considered, an absolute position sensor using a magnet attached to the wind vane shaft.

Figure 4.2.2.11 – Magnetic Rotary Position Sensor [17]



We located a company named Melexis that sold a magnetic rotary position sensor. The integrated circuit chip and magnet shown in Figure 4.2.2.11 allowed us to detect the absolute position of the wind vane. According to the specification data sheet, the Melexis chip gave a voltage output to represent the position in degrees. This voltage was read by the microcontroller.

We developed a validation testing methodology for both the Melexis position sensor and the Hall effect speed sensor. Before we developed this methodology, it was important to consider the design requirements of the overall project. First, we were not evaluated on the mechanical design of our system or subsystems. Second, the scope of this paper covered the design concepts and did not realize the final design. With these factors in mind, we focused on ensuring the sensors we had selected for the anemometer that would meet our expectations. We hadn't designed the mechanically complex subsystems to provide actual wind measurements. Alternatively, we developed a simple, repeatable test apparatus for both wind speed and direction sensors. This sensor developed by Melexis needs to meet two main requirements to be considered functional in our system.

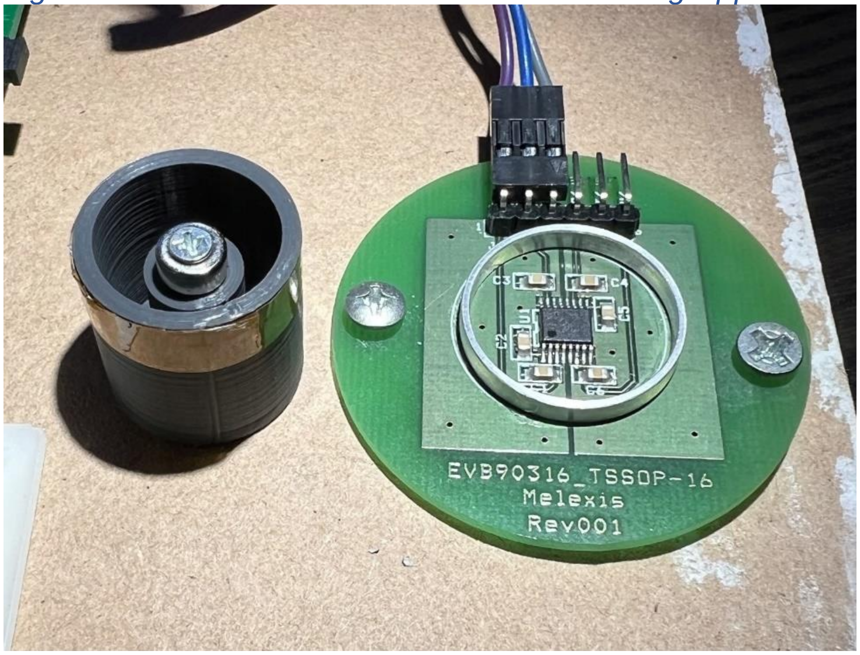
The first requirement was that the sensor provided the absolute shaft position of the nacelle shaft. Providing the absolute position meant that the sensor did not need an external reference to give the position of the shaft. This was tested by cycling the power of the device under two conditions. The first condition was to cycle power and leave the shaft in the same position. The passing criterion for this condition was that the sensor output was unchanged from before power was removed to after power is reapplied. The second condition was to power cycle the device and move the shaft with power removed.

The passing criterion for this condition was that the sensor output changed from when the power was removed to when the power was reapplied.

The second requirement was that the position was reliably and repeatably measured. The objective of this test was to ensure that the sensor output was correlated to a human readable degree measurement. For this to be the case, the full range of the sensor output must be linear and repeatable. We used two test conditions to satisfy this requirement. For the first condition, we marked the shaft and the sensor body at the rollover point. The rollover point was the point where the sensor voltage output changed from the maximum value to the minimum value or vice versa. This was like the point on a compass where the degree heading changes from 360 degrees to zero degrees. For this condition to be satisfied, the marks visually lined up with one another after each revolution. This was repeated and observed for three revolutions in each direction. The second condition for this requirement was for an arbitrary position selected in the sensors path of travel to be repeatably reached and measured. We marked the arbitrary point and the shaft. Then the sensor read at the point which was recorded. For this requirement to be satisfied, the value after the shaft had been rotated needed to be within plus or minus 40 analogs to digital counts of the initially recorded reading. The number of analogs to digital counts were chosen to be 1 percent of the 4096 counts representing the full range of the microcontroller.

We conducted the tests outlined previously by creating a testing apparatus. This apparatus consisted of a knob in place of the shaft. The knob contained the diametrically magnetized magnet and was marked to indicate its position. Kapton tape was applied to the knob at the interface between the knob and the evaluation board. For our testing, we used an evaluation board created by Melexis. This evaluation board was assembled with a guide ring for our knob to rest in. This guide ring also acted to center the knob over the sensor itself. The evaluation board was then connected to the microcontroller for taking our test measurements. The testing apparatus was observed in Figure 4.2.2.12. Pictured on the left is the knob, placed inverted to show the magnet housed within. Pictured on the right was the Melexis evaluation board itself.

Figure 4.2.2.12 – Wind Direction Sensor Testing Apparatus



We now have a repeatable means to test the wind direction sensor. As well as a set of testing requirements and conditions. We proceeded with gathering the required testing data. First, we gathered all the data needed to satisfy the first requirement in Table 4.2.2.1.

Table 4.2.2.1 – Wind Direction Sensor Requirement 1 Test Results

Condition	Position Before Power Cycle (Counts)	Position After Power Cycle (Counts)	Delta (counts)	Passing
Power Cycle – No Movement	640	640	0	PASS
Power Cycle – Arbitrary Movement	640	1298	658	PASS

Our testing showed that this sensor met the specification of the first requirement. We maintained the positional reading even after power cycle. As well as the position changed even in the event the shaft was moved while the power was removed.

Next, we began testing for the second requirement. The first condition for the second requirement stated that we needed to arrive at the same transition point after each revolution. We rotated the knob first in the clockwise direction three times. Each full rotation, we lined up the marks for the zero point and record the sensor reading. We repeated this testing for the counterclockwise direction. For both directions, the test was considered passing if the sensor reading was within 40 counts of 4096, the maximum value of the sensor. The results of this testing were displayed in Table 4.2.2.2.

Table 4.2.2.2 – Wind Direction Sensor Requirement 2 Condition 1 Test Results

Condition	Rollover Position (Counts)	Measured Position After Full Revolution (Counts)	Delta (Counts)	Passing
CW Rotation 1	4095	4091	4	PASS
CW Rotation 2	4095	4086	9	PASS
CW Rotation 3	4095	4095	0	PASS
CCW Rotation 1	4095	4087	8	PASS
CCW Rotation 2	4095	4090	5	PASS
CCW Rotation 3	4095	4079	16	PASS

Our testing showed that the sensor met the first condition for requirement 2. For all the gathered data, the delta between the expected and measured values were less than 1 percent of the analog to digital converter range. This held true in both the clockwise and counterclockwise directions.

For the final condition, we began by selecting an arbitrary point in the travel of the knob standing in for the shaft. We recorded the sensor reading at this position and marked the position. This served as our reference starting point. From here, we rotated the knob one full turn in the clockwise direction. We visually lined up the reference mark. We then recorded the sensor output and compared the value to the initial reference reading. This was repeated two more times in the clockwise direction. Then, we repeated the same steps modifying the direction of rotation to be counterclockwise. Each test was considered passing if the trial measurement was within 40 counts of the previous reading.

Table 4.2.2.3 – Wind Direction Sensor Requirement 2 Condition 1 Test Results

Condition	Reference Position (Counts)	Measured Position After Full Revolution (Counts)	Delta (Counts)	Passing
CW Rotation 1	783	781	2	PASS
CW Rotation 2	783	788	5	PASS
CW Rotation 3	783	797	14	PASS
CCW Rotation 1	783	782	1	PASS
CCW Rotation 2	783	761	22	PASS
CCW Rotation 3	783	785	2	PASS

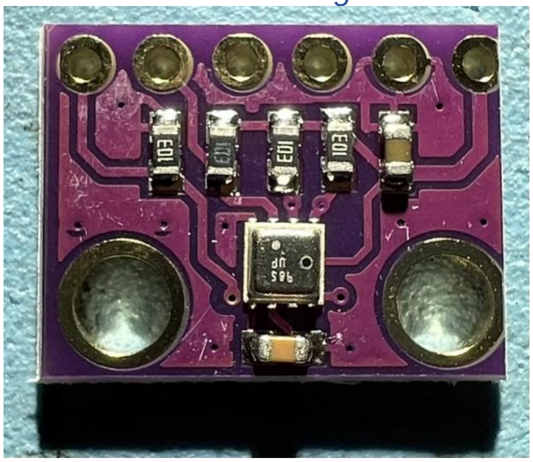
Our test results for the final condition of the second requirement shown in Table 4.2.2.3 demonstrates that our wind direction sensor passes all requisite testing. Both for this requirement and those before it. This testing indicates that the sensor should reliably and accurately read the wind direction in an implemented design. The next steps for our design and testing required more complex mechanical design than we have the time resources to create during our investigation stages. This testing does however validate the concept of this sensor and allowed us to proceed with the other design items to be completed in this document.

4.2.3 – Temperature, Humidity, and Barometric Sensor

Selection Methodology:

We thoroughly assessed various components and components, considered our design and cost limits, before making a final choice on what would be used in the final design. We used the research from the preceding chapters to guide our decision-making and ensured that we were properly examining all our possibilities. In addition, we weighed the benefits and drawbacks of each component in relation to our goals, objectives, and criteria. The parts that followed went into further depth on the process and general logic behind our final component and part selections. Similar Technology Information and Investigating similar products and technology were already on the market that could help us design our own technology. While no product perfectly fulfilled the need for monitoring land in fire-prone regions, with an emphasis on prevention, elements that have been incorporated into other items and initiatives had been discussed in length in earlier chapters. (See also Chapter 3) When choosing a temperature sensor, it was helpful to understand the different types and how they were operated so that the best choice could be made when we were ready to choose a component. Thermistors, Resistance Temperature Detectors (RTD), thermocouples, and semiconductor-based integrated circuits were examples of common temperature sensors (ICs). While all these sensors were used to monitor temperature, they did so in various ways and with different materials. As a result, it was advantageous to separate these gadgets into smaller classes.

Figure 4.2.3.1 – BME280 Sensor Used for Testing



After a Lengthy consideration we finalized our decision with the BME280. The reason of the selection was because BME280 was a humidity sensor designed specifically for mobile applications and wearables where small size and low battery consumption were critical design characteristics. The unit combined high linearity and high precision sensors, making it ideal for low current consumption, long-term stability, and strong EMC resilience. The humidity sensor had a very rapid reaction time and hence met the performance criteria for developing applications such as context awareness and high accuracy across a wide temperature range. The high accuracy capacity of the sensor was the main selling point for our team of engineers, the device integrated individual pressure, humidity, and temperature sensors with excellent linearity and precision in an 8-pin metal-lid 2.5 x 2.5 x 0.93 mm3 LGA box. The BME280 was developed with low current consumption (3.6 pA @1Hz), long term stability, and good EMC robustness in mind. The humidity sensor had a very rapid reaction time, which met the performance criteria for new applications such as context awareness and high accuracy across a wide temperature range. The pressure sensor was an absolute barometric pressure sensor with extremely high precision and resolution while producing very little noise. The inbuilt temperature sensor had been designed with very low noise and good resolution in mind.

Parameter	Technical data
Operation range	Pressure: 300.. 1100 hPa Temperature: -40...85°C
Supply voltage VDDIO Supply voltage VDD	1.2...3.6V 1.71...3.6V
Interface	I²C and SPI
Average current consumption (typ.) (1Hz data refresh rate)	1.8 A @ 1 Hz (H, T) 2.8 A @ 1 Hz (P. T) 3.6 A @ 1 Hz (H.P.T) T . temperature
Average current consumption in sleep mode	0.1
Humidity sensor Response time (T63%) Accuracy tolerance Hysteresis	1 S +3% relative humidity <2% relative humidity
Pressure sensor RMS Noise Sensitivity Error	0.2Pa (equiv. to 1.7cm) +0 25% (equiv. to 1m at 400m height change)
Temperature coefficient offset	+1 5Pa/K (equiv. to +12.6cm at 1°C temperature change)
RoHS compliant, halogen-free, MSL1	
Package dimensions	8-Pin LGA with metal

and high precision sensors, made it ideal for low current consumption, long-term stability, and strong EMC resilience. The humidity sensor had a very rapid reaction time and hence met the performance criteria for developing applications such as context awareness and high accuracy across a wide temperature range. The high accuracy capacity of the sensor was the main selling point for our team of engineers, the device integrated individual pressure, humidity, and temperature sensors with excellent linearity and precision in an

8-pin metal-lid 2.5 x 2.5 x 0.93 mm3 LGA box. The BME280 was developed with low current consumption (3.6 pA @1Hz), long term stability, and good EMC robustness in mind. The humidity sensor had a very rapid reaction time, which met the performance criteria for new applications such as context awareness and high accuracy across a wide temperature range. The pressure sensor was an absolute barometric pressure sensor with extremely high precision and resolution while producing very little noise. The inbuilt temperature sensor had been designed with very low noise and good resolution in mind.

BME280 Sensor Operation

The I2C and SPI (3-wire/4-wire) digital serial interfaces were supported by the BME280. The sensor had three power modes: sleep mode, normal mode, and forced mode. In normal mode, the sensor alternated between measurements and standby periods. This mode was recommended when employing the BME280's built-in IIR filter to filter short-term disturbances (e.g., blowing into the sensor). In forced mode, the sensor provided a single measurement on demand and then returned to sleep state. This mode was appropriate for applications that required a low sample rate or host-based synchronization. Several oversampling modes, filter modes, and data rates were set to customize data rate, noise, reaction time, and current consumption to the demands of the user. The sensor, when combined with multiple short-term disturbance filter settings, had tuned in a very flexible manner to respond to application and power management needs. Default settings optimized for numerous sample use-cases, such as weather monitoring, elevator/staircase identification, drop detection, or interior navigation, were supplied to facilitate the design-in phase.

4.2.4 – Air Quality

Air quality was an important factor in the health of an area and those residing within that area. For our application, we were taking some inspiration from the air quality index or AQI. The air quality index was a complex measure of the gasses and particles in the air over time. This index was typically used to indicate the pollution level of an urban area. Our application was focusing on natural areas that were prone to forest fires. For this, we focused on the air quality in the event there was an active forest fire in or near the area. Our research showed us that there were a few gasses from carbon dioxide to methane. The largest contributor to the emissions were carbon dioxide. The carbon dioxide produced by forest fires did generally dissipate relatively quickly though does cause measurable increased. Another factor in the air quality changed during a forest fire was the presence of volatile organic compounds. These compounds were chemical compounds such as, aldehydes, alkyl benzenes and other hydrocarbons. Volatile organic compounds made up part of the smell of smoke. Volatile organic compounds also contributed part of the smell of smoke. The typical concentration of these compounds in the atmosphere was neat zero. For reference, the measure for volatile organic compounds were parts per billion. Our research also revealed there were many types of air quality sensor. There were gas sensors each of which measured a different gas. There were particulate sensors that measured the microscopic particulate matter that may be harmful to humans. There were also sensors that measured the volatile organic compounds. We elected to measure a subset of the changes to the air. We measured carbon dioxide and total volatile organic compounds. This gave our system a good balance of measurements considering the exact combination of elements in the air during a fire. We were also not attempting to detect fire using this sensor. This reduced the need for a wide spectrum of accurate air metrics. The CCS811 sensor module was expected to fit our requirements.

We were using the measurement of carbon dioxide and total volatile organic compounds to give a general overview of the air quality of an area when there were fires nearby. With this in mind, we developed our testing methodology for the subsystem to focus on stability and the ability to measure deltas. We started by taking a measurement of the known atmospheric baselines for both measurements. The average concentration of carbon dioxide in the atmosphere were 400 parts per million. The concentration of total volatile organic compounds were near zero parts per billion. We expected our sensor to read zero parts per billion total volatile organic compounds when placed in a well-ventilated space. The first requirement for our testing was that when placed outdoors, the sensor reads no more than 10 parts per billion total volatile organic compounds and no more than 420 parts per million carbon dioxides. With the first requirement, we established the sensor which can accurately measure our expected values. From here, we tested additional values. These additional values were less a test for accuracy and more a test for precision. The second requirement was that the sensor read the value and outputted a stable measurement for 5 measurement cycles. Stability was defined as a change of no more than 10 parts for either measurement from cycle to cycle. One cycle was one second. The second requirement was tested in two ranges. The first range was an occupied room of a house. The expected concentration of carbon dioxide was between 400 parts per million and 1000 parts per million. As for volatile organic compounds, there were many contributing factors to the indoor concentration. For our purposes, we looked for two conditions to represent a passing test. The first condition was an increase from outdoor to indoor concentrations of volatile organic compounds. The second was that the concentration of total volatile organic compounds was no more than 500 parts per billion [38]. The second condition was intended to test an extreme condition. For this condition, we placed the sensor inside a candle that had been recently extinguished. The passing criteria for this test was that both measurement values read in the multiple thousands of parts. We recognized that this condition was not well defined or accurate. In later testing, we characterized the subsystem. The final requirement was that the sensor recover to indoor levels once removed from the extreme environment. The testing apparatus for the extreme condition was observed in Figure 4.2.4.1.

Figure 4.2.4.1 – Air Quality Sensor Inside Candle

Table 4.2.4.1 – Air Quality Sensor

Air Quality Sensor Test Results					
Test Condition	Expeted Max CO2 Concentration (PPM)	Actual CO2 Concentration (PPM)	Expected Max TVOC Concentration (PPB)	Actual TVOC Concentration (PPB)	Passing
Outdoors	400	400	0	0	PASS
Indoors	400 - 1000	949	500	83	PASS
Candle	Greater than 1000	8224	Greater than 500	14011	PASS

The results of our testing in Table 4.2.4.1 reflected our expectations. All testing criteria were met. During our testing, we had not made an additional observation. The behavior when the sensor was moved from the extreme environment back to the indoor environment was unexpected. Both measurements increased and decreased quickly and stabilized as expected. The unexpected behavior occurred when the concentration was abruptly decreased. Both measurements decreased past the indoor measurement to the baseline outdoor measurement. We expected the measurements to return to near the indoor readings noted just prior to the extreme measurements. The sensor took approximately 15 minutes to return to the indoor measurements for both values. After referencing the datasheet for the ccs811 sensor [39] we believed we had been within the burn in period for the sensor. As a result, the measurements had not reacted properly when exposed to large changes in the environment. We monitored this as we began our full system level testing.

4.2.5 – Camera

We began the selection process for the camera module by considering its function with our system. We were using the camera to record a history of the fuel load in the area around each node. Fuel load referred to the amount of combustible material in each area. Images were used as a means for forestry management services that were used to monitor this factor. The image needed to effectively convey the environment. To do this, the image did not need to be high quality. This was an important benefit when considering the limitations of our design. We used a microcontroller to manage each node. Microcontrollers were inherently low computing power devices. The ESP32 for example had a 520-kilobyte memory size. For comparison, a typical smartphone image had between 1 and 2 megabytes. Additionally, there were some memory requirements for the program and the mesh network running alongside the image processing. With these factors in mind, we searched for a low cost and low image quality camera.

The first camera module we selected is the OV7670. This camera was capable of a resolution of 640x480 also known as VGA. We initially selected this module because the cost was low at around ten dollars each and the VGA resolution was expected to be sufficient. As we began the testing process with this module, we ran into a few issues that guided our future selections. The first issue we encountered was the lack of support documentation. There were not many existing code libraries that met our somewhat narrow field of requirements. The ESP32 was the microprocessor we had chosen. This board did run software made for the Arduino platform. The issue we encountered was that each microcontroller had a lower-level architecture that was exposed to the Arduino software through a driver layer. The libraries that were available for the OV7670 were not compatible with this driver layer. The result was a compilation time error. There were several approaches to troubleshooting this issue. The first path we investigated was to simply find alternative libraries for the OV7670. This proved unsuccessful as we encountered the same or similar issues. The next path we considered was to modify the existing library code to run on architecture of the ESP32. This was also unsuccessful. The library files had errors in many different files. Fixing these errors required a significant time investment into understanding both the ESP32 architecture and the library structure. The final option we explored was creating our own library for the camera module. The basic functionality of the camera was straightforward with an I2C control interface and an 8-bit digital output for pixel data. The first issue with this option was the lack of documentation for the camera module itself. The second issue was the image processing and movement of pixel data. All the existing library functions we explored had extensive handling and processing of the pixel data. This combined to rule this option out again for a time investment beyond the constraints of our project. We tried one additional camera module of a similar specification and interface. We compiled a library for the ESP32 for this new module. The issue is this library would require the ESP32 to be connected to an external non-volatile storage media. The added complexity of adding this storage had not outweighed the cost of stepping up to the next camera option we tried.

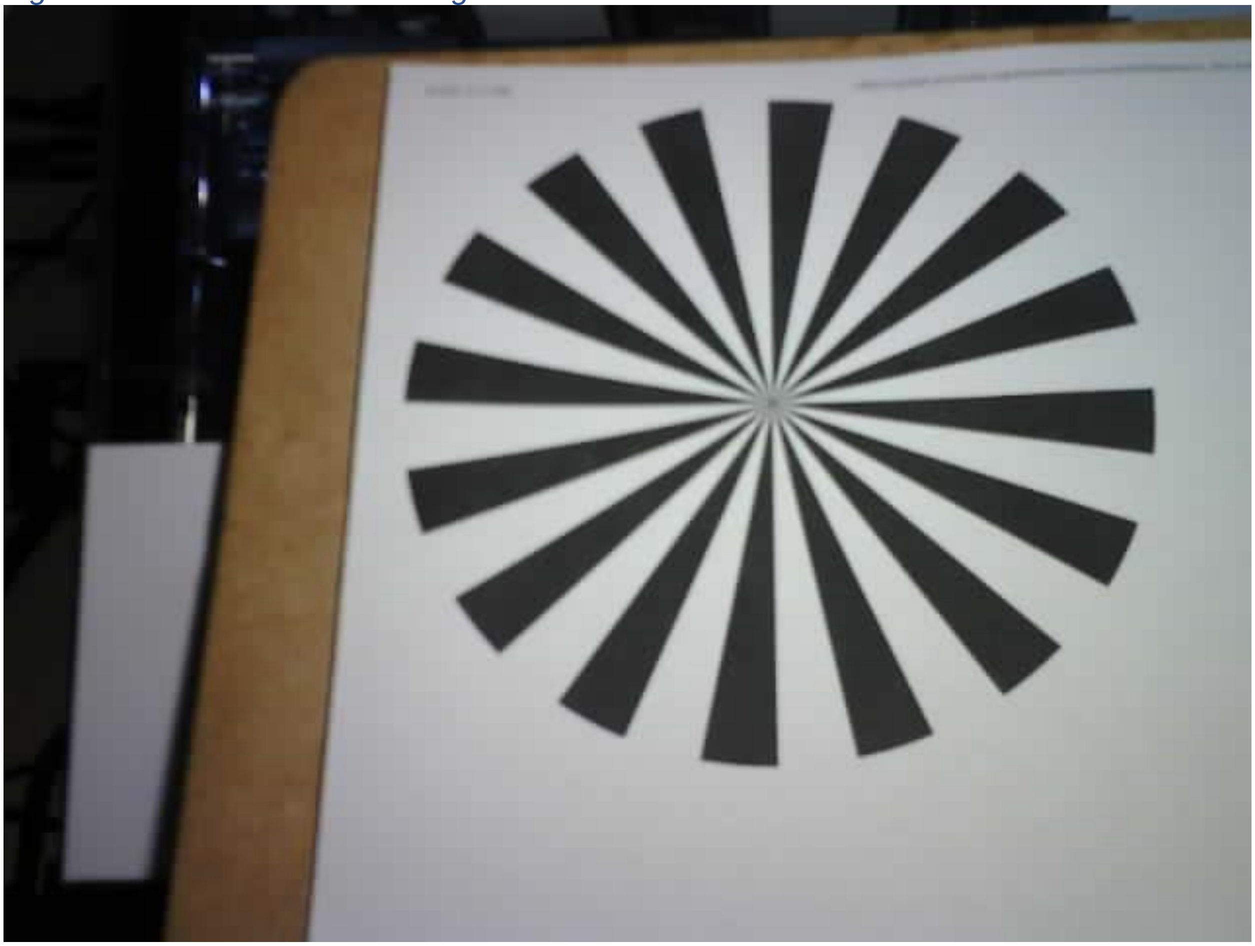
The third camera module we tested was notably different than the previous two. In contrast, this new module used a serial peripheral interface (SPI) connection to transfer pixel data. This module also carried the burden of capturing and preprocessing the pixel data. The cost for this additional functionality was roughly double the previous two options. The new module was the ArduCam mini 2MP plus. This module used the OV2640 camera assembly. Moving the pixel data and preprocessing to this external module afforded us a few additional features beyond our initial design criteria. The first was the ability to capture higher quality photos. The pixel data was buffered in the module and may be read, sent and discarded before repeating the process. We had not implemented this feature due to other limitations of the system. The next benefit was the reduction in complexity from a hardware design perspective. The number of pins were reduced from 18 on the other options to just 8 on the ArduCam module. The third benefit was the company itself supplied their own library designed to be portable across most boards running on the Arduino platform. This library had some minor issues though the ability to compile and run it allowed us to begin working on solutions rather than fighting our tools.

The manufacturer of this module supplied a set of example software for testing the configuration of the hardware. This software consisted of two parts. The first part was a Windows 10 application for controlling the camera and adjusting its settings. The second part was the Arduino code to be loaded onto the ESP32 that received the controls from the computer. Thought the Arduino code was open source, the control software was closed source. Also, the documentation for the open-source Arduino code did not specify how to adapt the camera for our use case. We attempted to take a single low-quality photo and sent the data over serial for testing purposes. While accomplishing this goal, we needed to do some reverse engineering to understand the software on both sides. First, we investigated the Arduino code to find the available commands. Each command was run by sending a single byte over serial from the computer to the ESP32. The issues were the commands which seemed to mix decimal and hexadecimal values. Additionally, the built-in serial interface for the Arduino IDE only allowed sending ASCII characters. To get around this limitation, we designed our own control software in python. We sent the exact format of bytes and received image data. This presented the next issue for us that we overcame eventually, the image data had not contained a decipherable image. After some troubleshooting, we decided to intercept the commands from the closed source drive software to understand the command structure for capturing an image. The first method we attempted was to create a man in the middle attack for the serial com port. We created a python script that would create a virtual com port for the closed source control software to connect to. The script would then print the commands to the terminal and passed the commands to the physical com port of the Arduino. In practice, this method resulted in a breakdown of the serial com port protocol and prevented any communication. In the process we concluded that we only needed the commands sent by the control software. This simplified the problem because the Arduino was already receiving these commands. To have them retrieved, we added a block of code to the Arduino side to save the serial commands it had received and printed them out to the serial monitor when it was prompted. Now that we had the byte commands, we used our python control software to replicate them. This proved to be the key, the next block of data we received in the python driver contained the header and footer of a jpg image file.

We then extracted the bytes between these begin and end points and wrote the bytes to a file. Now that we had gained the ability to control the example software, we began the process of breaking down the command structure. There are three commands needed to capture an image. The first command set the type of camera module and image resolution. The second command instructed the camera mode to single capture. The final command instructed the camera module to capture and send the data over serial. Using these commands as a starting point, we were able to rewrite the example code to be purpose built for our application and increase efficiency. This basic command structure was sufficient to complete our testing for part selection.

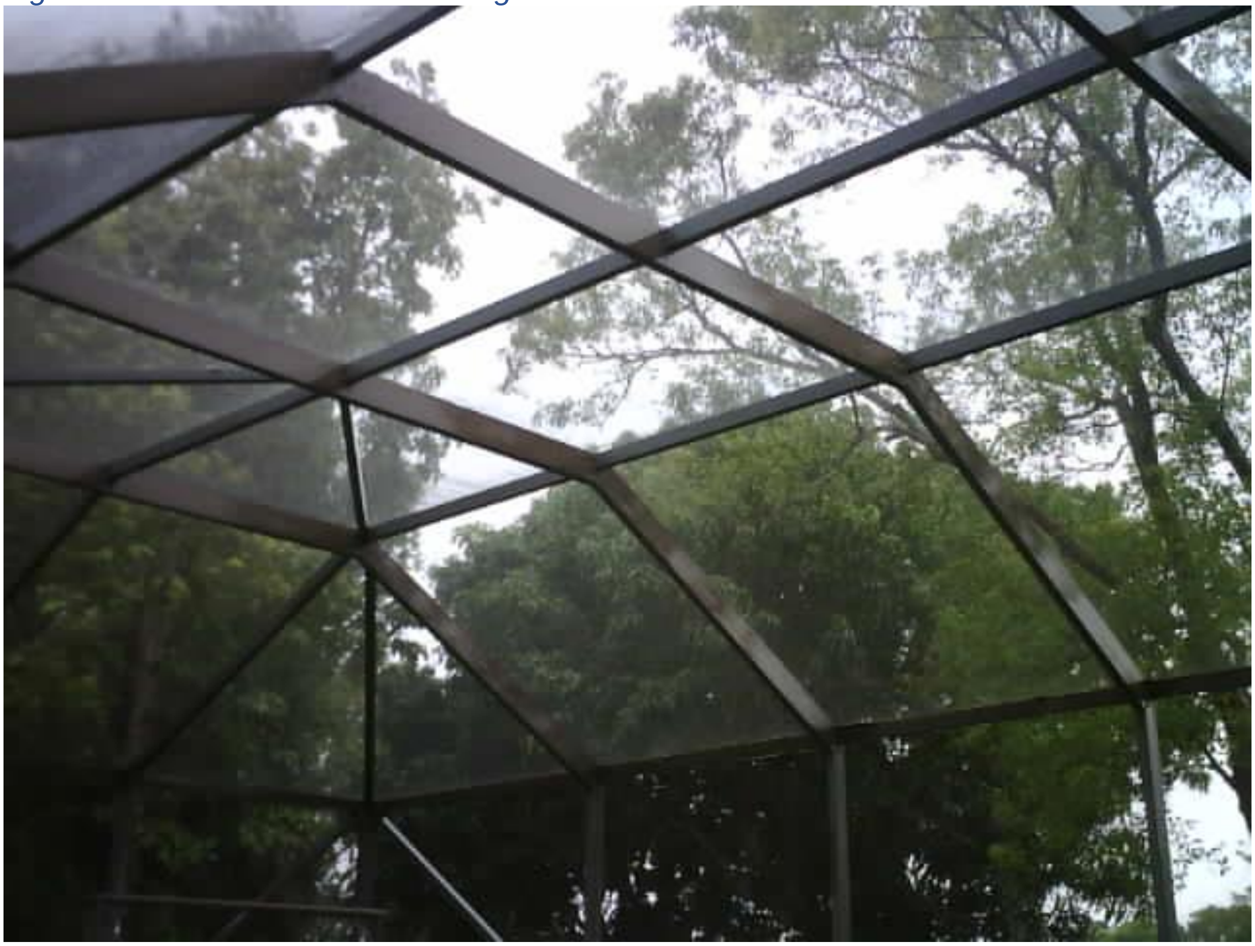
We conducted most of the testing for the camera module at the system level. The movement of pixel data through the node and over the mesh network presented challenges separated from the functionality of the camera module itself. The testing methodology for the subsystem level camera module was to capture two images. There were two requirements for this test to be considered passing. The first requirement was the python control software which was capable of both commanding the camera to take a photo and saving the capture data to a jpg file. The second requirement was both photos be relatively focused to the eye. We left this requirement up to the eye of the test agent.

Figure 4.2.5.1 – Close Test Image



The image in Figure 4.2.5.1 satisfied both requirements for a successful test. This image was captured and saved using the python command script. The image subject was spoke target used for testing the optics and resolution of cameras. This gave us a clear metric for grading our image. Considering the resolution of the image, the capture was considered passing. We also captured an additional image to test the focus at longer ranges.

Figure 4.2.5.2 – Distance Test Image



The test at distance in Figure 4.2.5.2 showed a much clearer passing result for the focus test. We clearly observed the definition of the edges of the trees as well as the leaves on the branches. We also observed some shortcomings of the camera module when it came to dynamic range. The dynamic range was the ability to accurately represent both the highlights of the sky and shadows of the underside of the trees. This was an area we further investigated at the system level. For now, we concluded that this camera module was reasonably capable of filling its utilitarian purpose for our project.

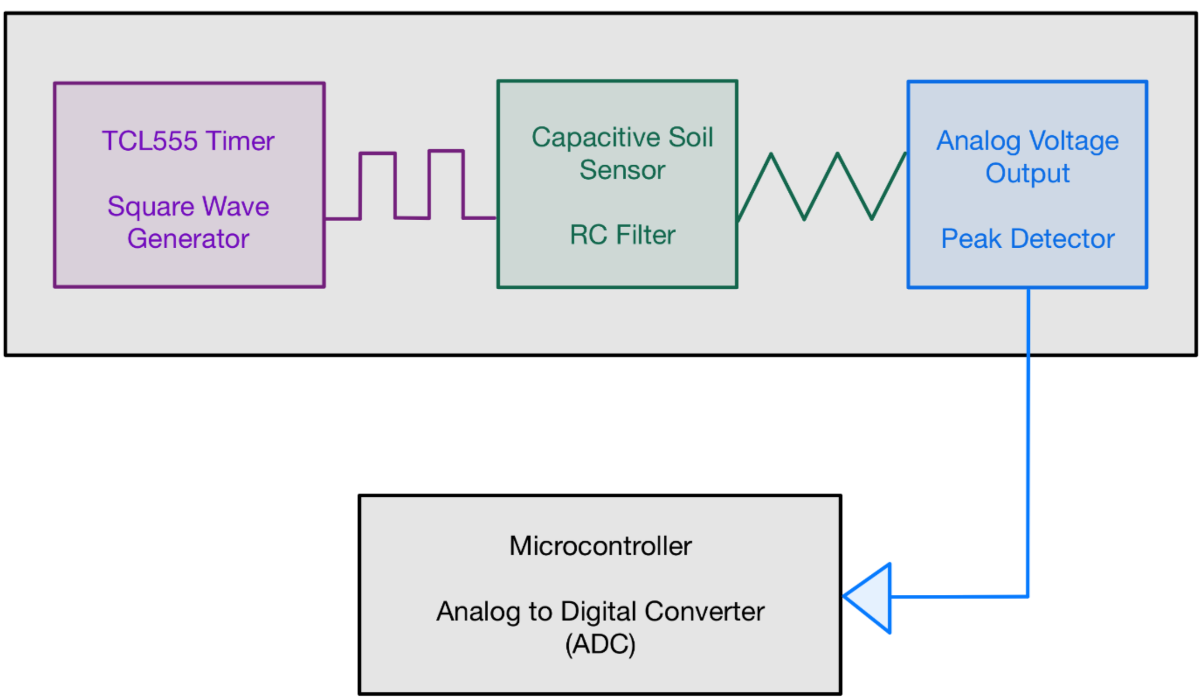
4.2.6 – Soil Moisture Sensor

We began our selection of the soil moisture sensor by considering the application and design requirements for this sensor. We used this sensor to monitor the relative water content of the topsoil. The system was expected to measure this data point as many times as the end user specifies. This was likely to be a few times per hour. A core design consideration of our product was reducing the preventative maintenance of the system. With this, we needed to consider the correlation between the number of measurements and the useful lifetime of the sensor.

We found that there were two main types of soil moisture sensor. The first type was a resistive soil moisture sensor. The basic operation of this type of sensor was relatively straightforward. The resistive soil moisture sensor had two probes separated by a fixed distance. These probes were inserted into the soil. The sensor then passed a known voltage across the probes. By measuring the voltage drop across the probes, the water content of the medium was estimated. This mode of operation caused an issue with prolonged use, corrosion of the probes. Each time the sensor was used, the voltage passing though the soil caused electrolysis to take place. The process of electrolysis accelerated the ever-continuing corrosion of the sensor. This meant that the sensor must only be used when necessary. This led to increased preventative maintenance and repair cost even if the system design limited the number of readings of the soil moisture sensor. Fortunately, the second type of soil moisture sensor solved these issues.

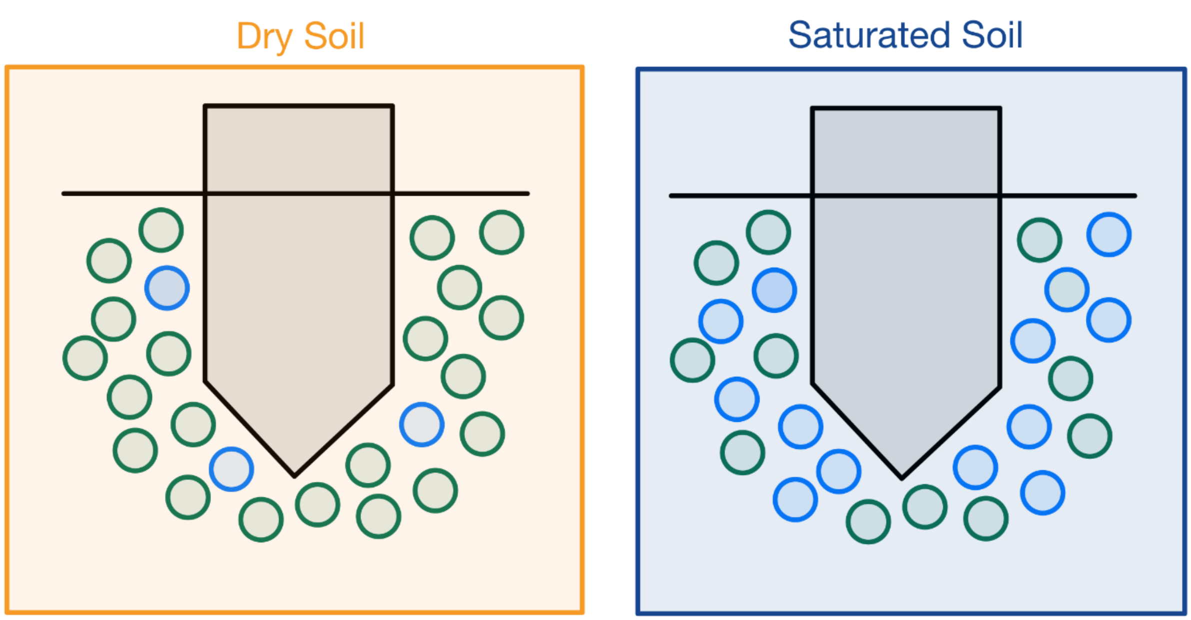
The second type was the capacitive soil moisture sensor. This sensor was immediately the more interesting choice for our project. The reasoning was twofold, no exposed contacts and there was almost no cost difference between the resistive and capacitive types. Having no exposed contacts, the preventative maintenance and ongoing costs for the system was reduced. Additionally, the cost difference was on the order of pennies. With those factors in mind, we moved forward with testing the capacitive soil sensor and validating it for use in our system.

Figure 4.2.6.1 – Capacitive Soil Moisture Sensor Hardware Block Diagram



Before we developed our testing methodology; we first gained a deeper understanding of the inner workings of the capacitive soil sensor. In Figure 4.2.6.1, we illustrated the basic hardware responsible for creating and reading the sensor value. The first thing to understand is the method to which the microcontroller communicated with the soil sensor. The communication was one way and in the form of an analog voltage signal. This signal was produced by the soil sensor and read by the analog to digital converter on the microcontroller. There were three subcircuits in the capacitive soil sensor that produced this analog voltage signal. The first subcircuit generated a square wave signal. This was done using a ubiquitous timer integrated circuit the TCL555. The frequency and other characteristics of the square wave were set by a set of external resistors and capacitors. The values of which were responsible for determining the actual value of these characteristics. This square wave was then passed through the next component of the soil moisture sensor, the soil probe. The soil probe was the actual sensing surface of this subsystem. The probe itself was two traces within the printed circuit board that were not directly connected. These traces were placed adjacent to one another to create a capacitive effect. The electrolytic medium in this case was the soil or any other material the sensor was placed into. The capacitive probe acted as the capacitor in an RC filter. The output of this filter was a triangular saw tooth waveform that was then passed into the final subcircuit of the soil moisture sensor, the analog voltage output. The analog voltage output was generated by a peak detector circuit. A peak detector used a capacitor, diode and a sufficiently large resistor. The diode prevented current from flowing in the reverse direction back through the capacitive probe. The resistor and capacitor were placed in parallel. The capacitor was charged and discharged according to the frequency of the triangular wave input. The voltage of the capacitor became the output of the sensor. As the dielectric medium encompassed the capacitive probe increased in conductivity; the output of the sensor reduced in voltage. This reduction in output voltage was then measured by our microcontroller.

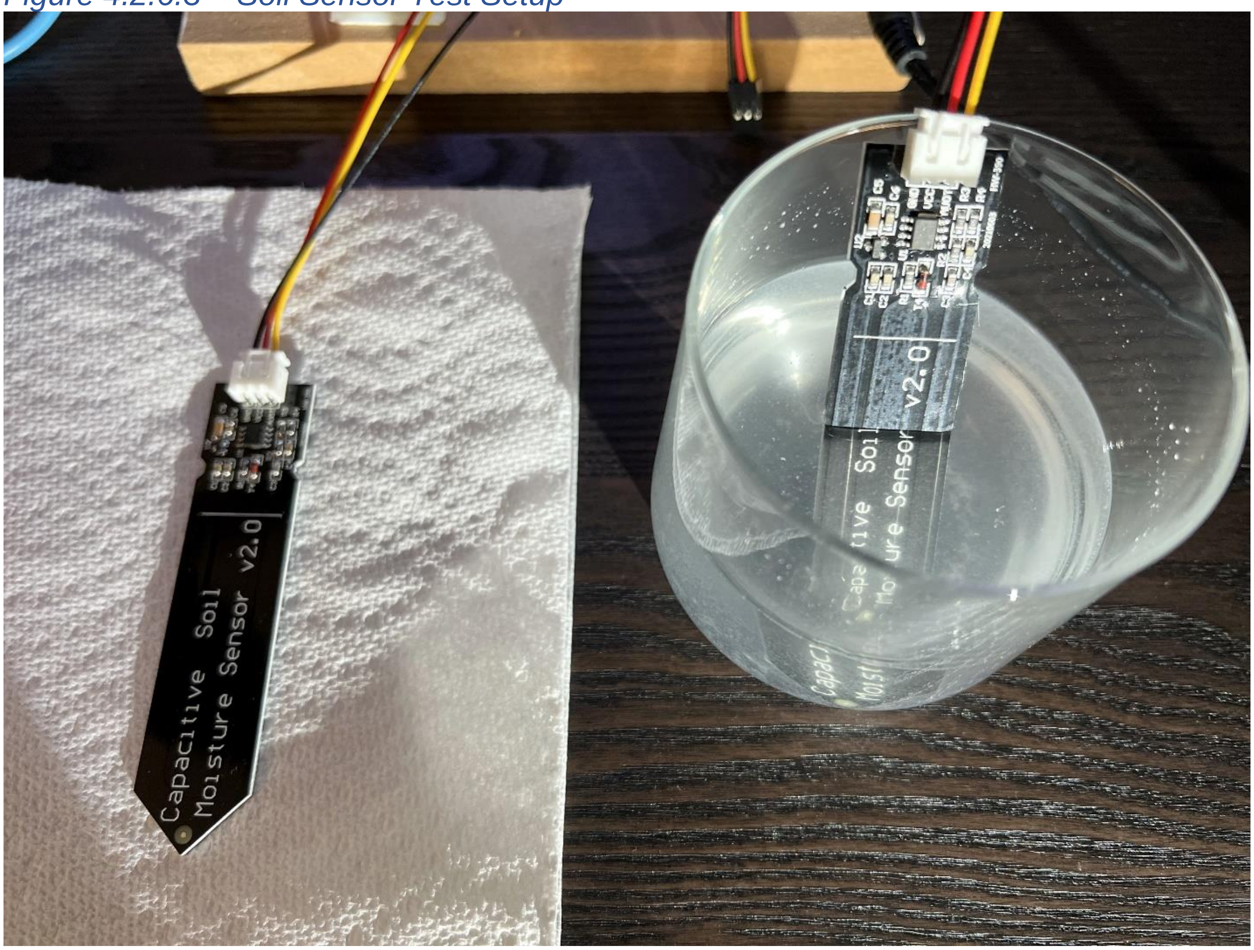
Figure 4.2.6.2 – Soil Sensor in Dry and Saturated Soils



Now we understood how the soil moisture sensor functioned, we considered the environment we were subjecting the sensor to. We measured the concentration of water in the soil. It was important to note that, water on its own is not conductive. In nature, water moved through the environment in the water cycle. Along the way, particles dissolved in the water. These particles produced the ions that were the medium of conductivity within water. As a result, when the soil was saturated with ion-rich water, the conductivity of the soil increased. An illustration of the physical changes in the saturated soil were observed in Figure 4.2.6.2.

With an understanding of the physical changes in our environment and how they correlated to the electrical properties of the medium, we developed a testing methodology. Our methodology focused on the extremes of possible environmental conditions. We focused on extremes because this demonstrated that the sensor had the necessary range. As well as the ability to quickly react to changing conditions. The second point became less important in the real world as changes in soil conditions happened over long periods of time. We were however interested in the ability to adapt and the rapidness to which that adaptation happened was helpful in the testing environment. Additionally, we had built an off the grid network of sensing nodes with limited power constraints. This meant, we also measured the continuous current draw of the sensor.

Figure 4.2.6.3 – Soil Sensor Test Setup



Our testing methodology consisted of three cycles from one extreme to another. The first extreme was placing the dry sensor on a paper towel in open air. This represented extremely dry soil. The second extreme was to place the sensor in saltwater. This saltwater solution had 3/4 cup of water to 1 teaspoon of table salt. The saltwater represented fully saturated soil. This condition hadn't occurred naturally but demonstrated the sensors functional range. The sensor was measured dry then submerged in the saltwater solution. The sensor was dried thoroughly when removed from the solution. The submersion – drying cycle was repeated three times for each sensor. We repeated this testing on two sensors to verify repeatability from one sensor to another. The basic testing setup is shown in Figure 4.2.6.3.

Table 4.2.6.1 – Soil Sensor 1 Test Results

Soil Sensor 1 Test Results			
Continuous Current Draw (mA)		5.64	
Cycle Number	Dry Output (Counts)	Submerged Output (Counts)	Delta (Counts)
1	2870	953	1917
2	2909	961	1948
3	2903	946	1957

Table 4.2.6.2 – Soil Sensor 2 Test Results

Soil Sensor 2 Test Results			
Continuous Current Draw (mA)		5.66	
Cycle Number	Dry Output (Counts)	Submerged Output (Counts)	Delta (Counts)
1	2863	942	1921
2	2858	954	1904
3	2857	945	1912

The results of our testing were displayed in Table 4.2.6.1 and Table 4.2.6.2. We drew three main conclusions from this result data. First, the sensors had a wide range of counts. Across all 6 trials and 2 sensors, the delta from the low to high measurement was greater than 1900 counts. This was roughly half of the full 4096 counts the microcontroller was capable of measuring. This was a high enough delta to measure the change in soil moisture. The second observation we made were that the measurements were precise. For all trials, the sensors recovered to their dry measurement after being submerged. This was true of the submerged measurements as well. This gives us confidence in the measurements to be stable when the environment was stable. The third observations were that the current draw was consistent between the two sensors tested. The sample size was small though the measurement was within 2 tenths of a milliamp indicating this was likely accurate. Considering we were relying on solar power and batteries to power each node; we considered power waste mitigations in the future.

4.2.7 – Power System

Several factors were considered when determining which photovoltaic module or modules were utilized for fire protection subsystem project. The panels used should be able to provide a considerable amount of the electrical needs. However, the project budget was critical, and committing too much of it to solar panels for energy generation jeopardized functioning in other aspects of the project.

4.2.7.1 – Solar Panel Testing Setup

The engineering team anticipated using an 5x5 polysilicon solar panel. They provided a panel with a relatively small size of around 5 square feet and a maximum power production of 3W and a Peak output spec (500mA) at 6 volts was enough to power the ~200mA system and charge during peak daylight hours. Our engineers expected this to be sufficient, and it was validated through testing, which was discussed further in the next sections.

Key Reasons Selection:

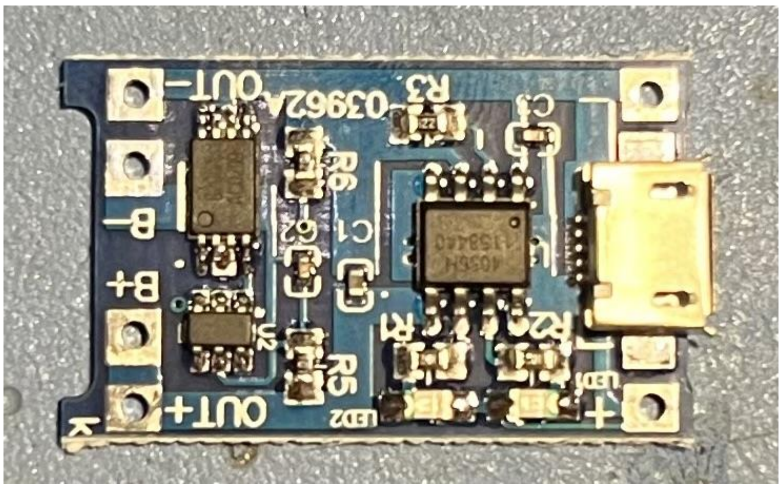
- -Size fits design requirements ~ 5x5in
- -Peak output spec (500mA) at 6 volts is enough to power the ~200mA system and charge during peak daylight hours
- -6-volt panel selected for a 5v input device because the voltage of the panel drops under load – even at 6v the input of the TP4056 tolerated

- The engineer team considered a potential upgrade up to a larger size at the same 6v

Solar Charge Controller

The TP4056 chip was a single-cell lithium-Ion battery charger that protected the cell from over and under charging. It featured two status outputs, one for charging in process and one for charging completed. It also had a charge current of up to 1A that may be programmed.

4.2.7.2 – TP4056 Charge Controller



Key Reason of Selection

- The TP4056 offered variable current control, which allowed us to compensate for the solar panel's actual current capabilities as needed.
- All charging logic was housed on a single integrated circuit (IC) in the TP4056.
- The input voltage was 5-8 volts, which gave us the range we needed for solar panel fluctuation.
- The TP4056 supplied and received 1A of power, which was more than enough for our planned use case.

4.2.7.3 lithium-ion foil Battery

A lithium-ion battery pack, often known as a Li-ion battery, was a type of rechargeable battery made up of cells in which lithium ions travel from the negative electrode to the positive electrode via an electrolyte during discharge and back again during charging.

Key Reason of Selection:

- These were high-energy batteries.
- They enabled the engineers to change the capacity and form factor without affecting the voltage or charging characteristics
- Once later stage of testing began this type of battery made modularity accessibly for the engineering team.

Testing Methodology:

Battery Regulator, battery pack and solar panel was evaluated all together during 3.5-day period. We developed a multi-axis graph consisting of Minutes for the horizontal Axis and a double vertical axis feature with Current (mA) in the left Vertical axis and Voltage in the right horizontal axis. During the day, the battery voltage occasionally rose but remained constant, and the current swungto the negative, indicating that the battery was beingcharged.

Figure 4.2.7.1 – Power System Current and Voltage Overview

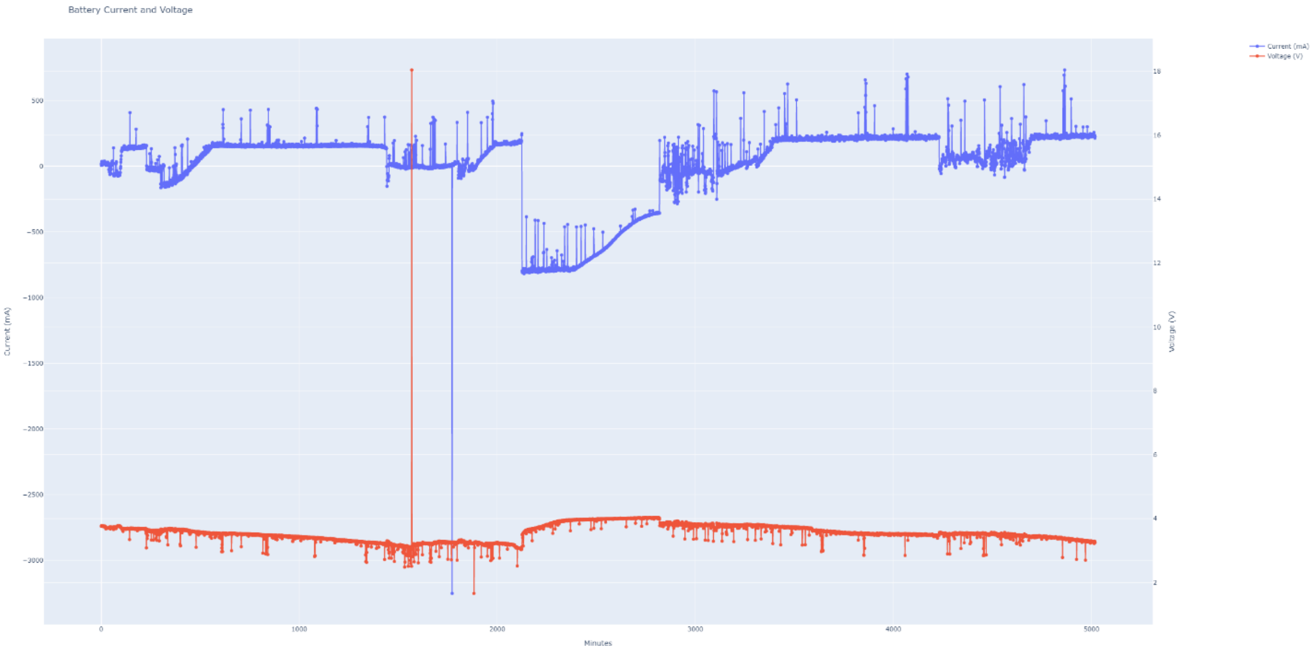
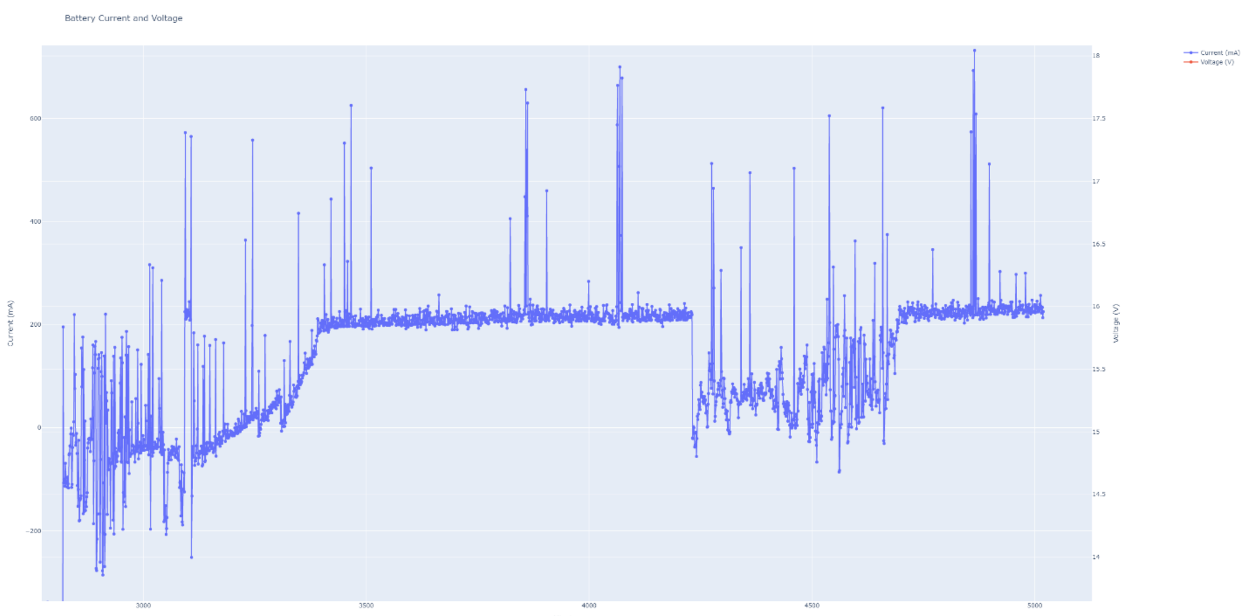


Figure 4.2.7.2 – Battery Current Draw for Day and Night



Power Systems Conclusion

We concluded this chapter with some of the tasks that the engineering team was developing to further improve this subsystem. We currently faced the task that we needed to implement 2 voltage regulators: 3.3 V and 5 V regulators, the reason for this was because we had different sensors with different power needs. An example of this would be our wind direction sensor (MLX90316LDC-BCS-000-SP) with an operating supply voltage of 5 V in contrast to the BME 280 with an operating supply voltage of 3.3 V, in fact, all our sensors ran 3.3 V except for our wind direction sensor, however the engineering team had decided that it wouldn't be aligned with the culture of this group to simple give up and disregard the wind direction sensor, for that reason currently this group was trying to implement a boost-buck converter which was a type of DC-to-DC converter in which the output voltage magnitude was more or less than the input voltage magnitude It was like a flyback converter that used a single inductor rather than a transformer.

One of the boost Regulators being considered was the MCP1642 A synchronous step-up DC-DC converter that was small, high efficiency, and had a set frequency. It offered a simple power supply solution for applications that required one-cell, two-cell, or three-cell alkaline batteries.

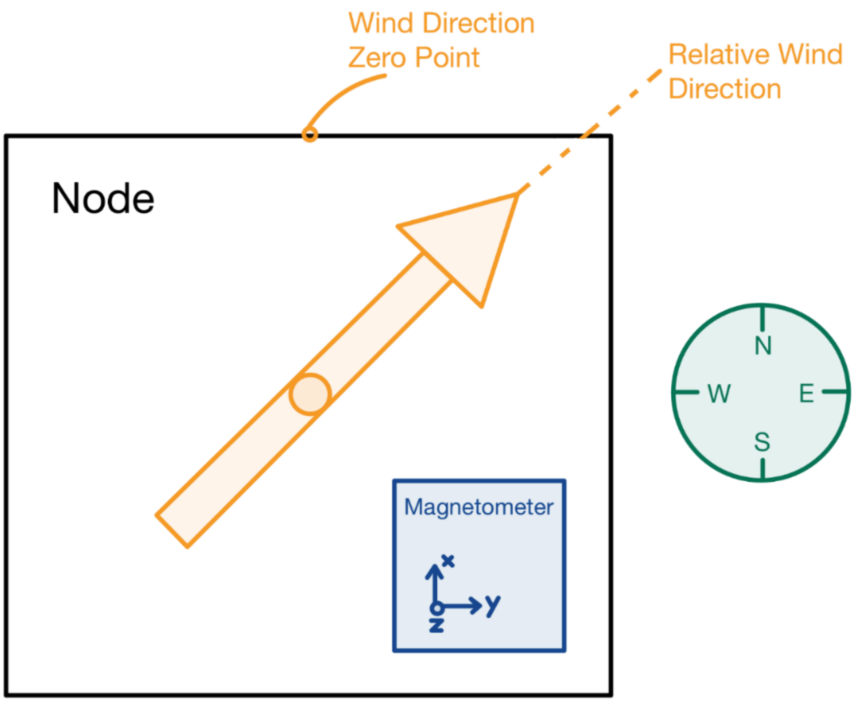
4.2.8 – Compass

We began our selection of the compass sensor by considering the purpose of a compass in our design. When designing the anemometer, we realized that we needed a reference point to base our measurements on. Specifically, we needed a reference for the wind direction measurement. A reference point allowed us to produce a measurement that was human friendly and easily usable. The measurement for wind direction was a compass heading from 0 to 360 degrees. When the node lacked a compass, we determined only the heading relative to the node itself. This would have not provided a useful measurement. The compass allowed us to determine the node's position relative to the earth. We used this reference to determine the offset needed to extrapolate the wind direction according to the earth.

During our research for the compass, we concluded that there were two main options for our application. The first option was, using a handheld compass to set a reference at setup. In this option, the user would mount the node in its operating location and note down the compass heading in relation to one face of the node. This reference could then be added to the software configuration for the node. The argument for this option was that the node should not move once it has been placed. Though this was true, we also aimed to reduce the complexity of the system from a user perspective. As a result, we chose to proceed with the second option. The second option was to use a magnetometer to detect the earth's magnetic field. This option added minimal cost and complexity to the system, and it also reduced human error and required input when setting up the system.

As we began developing our testing methodology for the compass, it was important to gain an understanding of how the compass was implemented in our design. Figure 4.2.8.1 showed the compass module mounted within a node. Also visible was the wind direction sensor. From this figure we drew a few important observations. The magnetometer X-axis was aligned with the zero point of the wind direction. For the purposes of this example, the zero point of the wind direction was also aligned with the north compass heading. We purposefully aligned the wind direction zero point and the magnetometer x-axis as this was the axis the magnetic heading is measured. This example showed that we were able to take the wind direction measurement and add the degrees to the magnetometer compass heading. The result was the actual wind direction.

Figure 4.2.8.1 – Compass Implementation within Node



The testing methodology for the magnetometer consisted of four comparisons of the magnetometer heading against the compass app within a smartphone. We oriented the magnetometer and the phone to align their direction of measure. We then moved both devices such that the phone read out each of the major directions, north, south, east, and west. These readings were then compared with the reading of the magnetometer.

We needed to consider factors that would have introduced error into our testing. The first factor was the magnets in our test smartphone. The smartphone itself contained many strong earth magnets in its construction as well as in the case for the phone. The magnetometer that was the unit under test was very sensitive to magnetic fields. We removed the case from the phone to reduce the number of magnets present. We also moved the phone sufficiently far from the magnetometer as to reduce the risk of skewing the results. A second factor considered was the accuracy of the smartphone itself. We had not known the exact device used by the phone to read the magnetic heading. We also had no other reference to ensure the smartphone direction was reasonably well calibrated. One final factor considered was the physical alignment between the magnetometer and the smartphone. The magnetometer was mounted in a female header on a prototype board. That board was then mounted to a wood board. The phone was placed against the edge of this wood board. All these placements had been completed by eye. Added together, these physical attributes of our test had introduced alignment errors. With these factors considered, we reran this test in future prototypes with an analog compass mounted square in reference to the magnetometer. This rerun avoided many of the issues noted with our current testing methodology. See Figure 4.2.8.2 for the overview of our testing apparatus.

Figure 4.2.8.2 – Magnetometer (Compass) Test Apparatus

The consideration of the potential errors in our testing lead us to widen the passing criteria for this set of testing. The test was to be considered successful so long as the magnetometer reading was within plus or minus 20 degrees of the smartphone reading. The results of this test indicated whether we had confidence in the sensor to function at all in the final design. More testings were completed when we were validating at a system level.

Table 4.2.8.1 – Magnetometer Test Results

Magnetometer (Compass) Test Results				
Continuous Current (mA)			1.05	
Major Direction	Smartphone Reading (Degrees)	Magnetometer Reading (Degrees)	Delta (Degrees)	Result
North	0	16	16	PASS
South	180	196	16	PASS
East	90	71	19	PASS
West	270	251	19	PASS

The results of our testing for the magnetometer were observed in Table 4.2.8.1. The continuous current draw was approximately 1 milliamp. This was important to be considered when we were designing our power system. Our testing did show that the sensor was passing. Though, during testing it was clear the unit under test was exceptionally susceptible to magnetic interference. The magnets for the wind sensors needed to be removed from the test bench. The smartphone also needed to move farther than pictured in Figure 4.2.8.2. Without these changes, the magnetometer would fall outside the 20-degree delta. This was an important design consideration when we finalized the physical proximity of the magnetometer to magnets within the systems.

4.2.9 – GPS

The purpose of including a Global Positioning System or GPS in our system was to allow each node to be plotted on a map. This was necessary as the location of the node was important to determine the area of physical land the node was sensing. During our research we learned there were some notable drawbacks to including a GPS system. The first was the power requirements. Our system was intended to operate off grid with minimal maintenance. This meant that we were limited to the battery and solar charging capacity of the node. The GPS system drew on average 40mA of current. This may have not seemed high but this was more than any other single sensor in the system. The next issue which also tied into power consumption was the warmup time required. When the GPS started, it first needed to acquire a signal lock with multiple satellites to accurately determine its location. This took as much as five minutes or longer depending on the location and surroundings of the device. For this period, the GPS had no useful output and was only consuming energy. The last issue we considered was the added design complexity. The unit required additional components to be added to the device. As well as an external antenna that must be mounted. All these issues combined to drive our consideration of alternative methods for determining location.

One important consideration was the real-world use case of the node. The node was deployed semi-permanently in the area where it was monitoring. This meant that after the initial set up, the node should not be moved. For this reason alone, the inclusion of a built in GPS unit did not have a practical design justification. The alternative we proposed was the use of a handheld GPS unit. These portable units that were widely available for sale and provided reasonably accurate location data. Included in this location data was the latitude and longitude needed to determine the location of our node. At the time of setup, the personnel deploying the node was asked to use a handheld GPS unit to note the latitude and longitude. Along with the node ID, this may have been saved in the configuration of the network and should not be needed to be changed unless the node was redeployed.

4.3 – Software Technologies Research

The list of software tools we used in this project consisted of python, which was used on an IDE, PyCharm, which was used for server development, web interface, data parsing, and commanding mesh. So, we used python to create a data server and a web interface for the user. The data server was responsible for handling all the data that the nodes contained. The plan was to indicate each node as a region or city on the map and that'll depict the area's land features. For example, Chicago was to be a node which showcased the city's wind speed, temperature, surrounding land information, etc. Then all this information got stored in the server as saved data for that date. Each passing date parsed the information obtained from the sensors and then stored it into the server. The web interface was created for the user to hover over different nodes or locations on the map in order to see if that area had a chance in dictating how good the area was for possible hazardous conditions. We used Android Arduino for making the mesh networks. We had used and seen Arduino before. They were so popular in different projects. They were used as microcontrollers which power systems. They were able to read inputs and lights on a sensor, a finger on a button operated to do many functions. Mainly, C++ was used to operate Arduinos. So, we were planning to use Arduinos and connect it to our mesh network. Then, that way, we could control the mesh network to display the different nodes as desired. We used SolidWorks to build our mesh network from the beginning to the end. Initially, it was used for designing and building mechanical, electrical, and software elements. AutoCad was used for 2D related work. We used Eagle to build our electrical network designs and then created schematics. We used Eagle to check for DRS errors and that helped us in identifying if our PCBs were ready to go for the next stage. We also used Cura which was used for printing 3D models. Whenever we required any materials, we used it to manufacture materials in our product.

4.3.1 – Mesh Networking

Mesh. This library handled the networking with built in error handling and automatically connecting new devices. This method of mesh networking did reduce the range and capacity of the network. We had designed the software to be modular supporting future upgrades to the mesh networking if required.

4.3.2 – Dash/Plotly

Dash was a python framework made by plotly for developing interactive web applications. Dash was written using Flask, Plotly.js and React.js. Through dash, HTML was not needed to be learnt. The same goes with HTML, CSS and JavaScript. In order to make interactive dashboard, only python was needed. Dash was an open source and the application is developed using the framework which could be viewed on the web browser. The building blocks of Dash consisted of Layout and Callbacks. Layout described the look and feel of the app, which defined the elements like graphs, dropdowns, etc. The placement, size, color, etc. Dash had Dash HTML components through which the style of HTML could be created like headings, paragraph, images, etc. using python. Dash core components were used to create elements such as graphs, dropdowns, sliders, etc.

Callbacks were used when one needed to bring interactivity into the dash application. This function helped in defining the activity during which a user could click a button or a dropdown menu. Layouts were used to create web-based layouts using Plotly dash. There were some important required packages needed to be installed. Packages like dash, dash-html-components, dash-core-components, Plotly. These packages helped in creating graphs, dropdown, etc and the Plotly packages helped in creating plots and reading various datasets. HTML was used for styling the web application. So, for our web application for the mesh networks, we used these tools to create a graphs or scatter plots to depict the different temperatures, wind speed, etc in different formats or colors to signify the extremes in high and low.

Callbacks in dash were used to make the web application or any application interactive. First, a callback was initialized using @app.callback(), which was then followed by a function definition. In this function, there were some arguments used to define what could change the values of the dropdown which could be interacted by the user. Some arguments included: output and input. The output function was used to define the components within the layout which was later updated after another function was called. The output functions in our application were developed to display the temperatures, wind speed, etc. The input function was used to define the components which were used to change the values when triggering the callback. The callback was needed to get triggered which was subjected on the change in the value of the dropdown.

4.4 – Stretch Goal Technologies Research

We have two main stretch goals. The first was regarding the fire danger rating system. This system helped officials to convey the risk level of a forest fire starting. Given the time and ability, we liked to explore implementing a machine learning component. This component was used for the data gathered by our system to make recommendations on setting the current fire danger level. Our second stretch goal was to implement trends page. This page would be displayed to the user all the trends for the measurements of the area. These trends helped to give a better picture of the health of an area over time.

4.5 – Overall Project overview

We selected measures wind speed and direction. During our research into anemometers, we observed a high cost for a sensor capable of measuring both speed and direction of wind. For example, one popular option was the Davis Anemometer with a price tag of $160 each. Our project was self-funded with a limited budget. We recognized a need to carefully examine the build vs buy concept. In the end, we decided to build our own anemometer. After several iterations of prototypes, we settled on two separate assemblies both using magnets to determine the wind characteristics.

The wind speed sensor used an A3144 hall effect sensor. The sensor was mounted in place and outputs an active low signal in the presence of a magnet.

The wind direction sensor used a Melexis MLX90316 rotary position sensor. The key feature of this sensor was that the position was absolute as opposed to relative. A relative sensor used the position when the unit was powered on as the zero point and measured everything relative to that point.

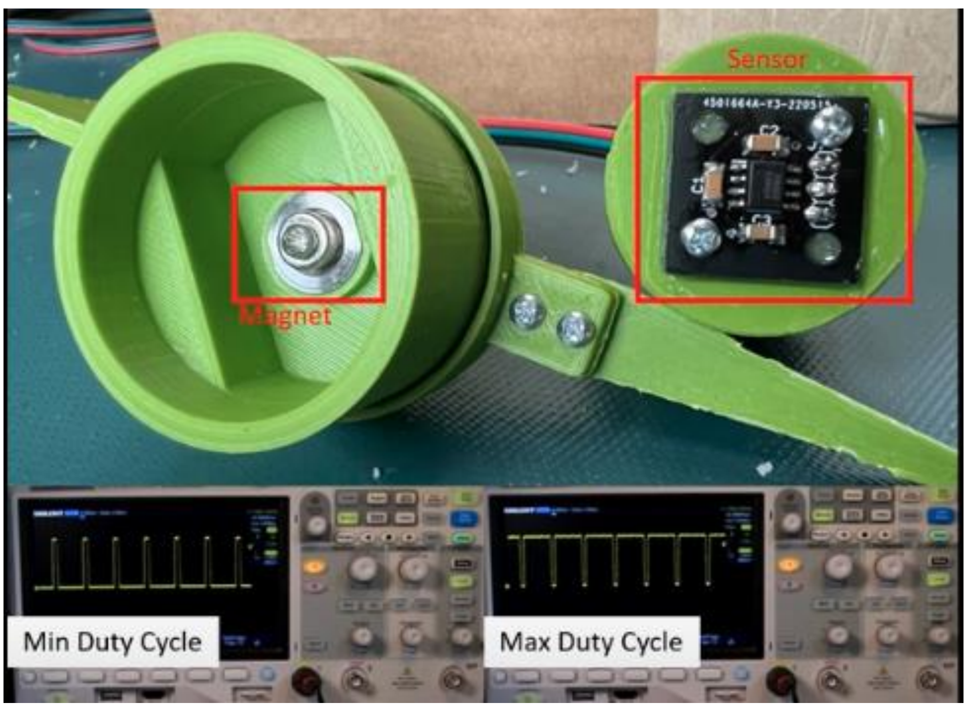
As we mentioned, cost was a consideration as we designed our wind speed sensor. We started by extracting the bearings from old hard drives. We then built a 3D model around the bearing to encompass the remaining parts, the hall effect sensor, magnets and of course the wind-catching cups. Altogether, the parts used cost less than $2.

The wind speed sensor worked by using the microcontroller to measure the digital pulse width produced by the sensor. This pulse width was equated to speed by obtaining a constant calibration factor. To do this, we attached the wind speed sensor to a vehicle. The vehicle’s speedometer became our reference speed. We recognized this method of calibration may have introduced some errors from inconsistencies in the real-world environment such as wind direction against the vehicle's travel. With some patience, we obtained a calibration factor that consistently yielded results within our plus or minus 10 percent accuracy goal. We also tested the wind speed sensor at different speeds to ensure a consistent measurement.

The wind direction sensor came programmed to output a variable pulse width modulation or PWM signal. This signal was produced in the presence of a diametrically magnetized magnet. We mounted the magnet to one end of an axle. A wind vane was attached to the other end.

The two halves pictured above come together to position the magnet just above the surface of the sensor. As the magnet rotates, the duty cycle of the PWM output changes.

The cycle time increased until reaching its maximum at a fixed transition point. On the other side of the transition point, the duty cycle rolls over to its minimum. The oscilloscope output on either side of the transition was pictured in the bottom left and right images. The transition point became the zero point for wind direction. We marked the zero point and performed several rotations in either direction to ensure the zero-point remained consistent.



With the consistency verified, we calculated the degrees using the noted equation. We obtained our calibration factor by positioning the sensor at the transition point and noting duty cycle time for both minimal and maximal values. We then solved the equation for calibration factor using 360 for degrees and the maximum for duty cycle time.

For the software component of our design, we built a web app that was run using Python, Plotly dash, and Tiny DB. Python was used because it can quickly adapt to any prototype environment and had a strong data manipulation. To program with Python, we used an IDE called PyCharm. This IDE was used to record and parse the data from the Python program that we are running. To build the web application, we used an interface called Dash, which allows us to develop web interfaces using python. Dash's core components were used to create elements such as, dropdowns, buttons, sliders, etc. Plotly is a data visualization package that is tightly integrated with Dash. The Plotly packages help to create interactive graphs and maps.

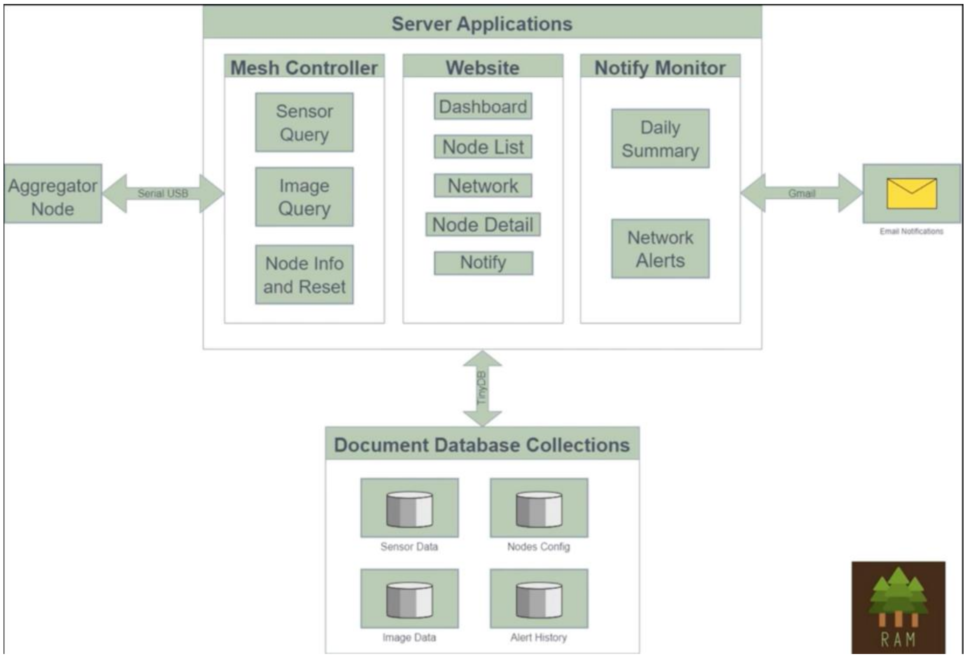
TinyDB is a document-oriented database written purely in python with no other external dependencies. It does every action using JSON and is scalable, has a flexible data model, and is easy to use for development.

We designed the software running on each node to be highly adaptable and portable. A core goal of our system is to give users the ability to pick and choose what sensors are required for their unique needs. To that end, we used the Arduino Platform. Arduino allows the code to be recompiled to run on a wide range of devices. Additionally, there is string community support for creating and maintaining the software libraries responsible for interfacing with many of the sensors and the mesh network.

Painless mesh allows the node to send and receive commands from the root aggregator and adapt to any new changes in the connections of the network.

When the node received a message, the message may be 1 of three commands. The first command asked the node to collect sensor data and send that data back to the aggregator. The second command asked the node to capture and image and send packets of image data to the aggregator. The final command was a reset command allowing the user to reset the node remotely in the event of an issue.

In our server software design, we had developed three applications each with a unique purpose. The flow of data through our system starts on the left with the Mesh Controller. This application was responsible for requesting, parsing, and storing data from the nodes. The communication to the mesh network was handled by the aggregator node which was an ESP 32 connected to the server by serial USB.



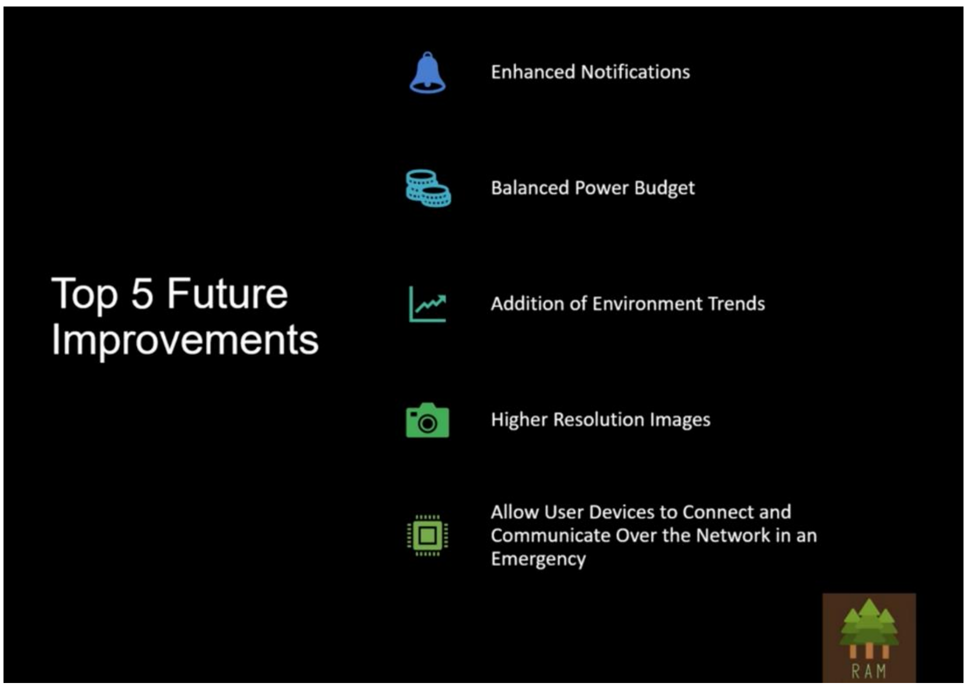
The next application was the web interface. The website was available to anyone on the same network as the server. The website was the primary user interface displaying all data and controls to the user. The website had an assortment of graphs and maps to display both current and historical data.

All of these applications share data stored in a document-based database managed by TinyDB. We have four collections in this database. The first stores the sensor data gathered from the nodes. The second stored the configuration data such as location and ID for each node. The third collection stored the raw pixel data for the images collected by the nodes. The final collection saved a history of notifications.

The final application, Notify Monitor was responsible sending out important alerts. Users had the option to enter their email through the web interface to begin receiving these notifications. Currently the system alerts a user if a node stops communicating with the network. Additionally, the users receives a daily overview summary of the node data from the previous 24 hours.

We decided to improve our project by adding five new features: enhanced notifications, balanced power budget, addition of environment trends, higher resolution images and allowing user devices to connect and communicate over the network in an emergency.

One of our top 5 features to be improved is the notification feature as it currently showcases a basic structure of how users are alerted of any issues in the mesh network. We would improve it by adding notifications to alert the user for a lot of different factors in the environment. For example, sending an alert when the temperature goes above or below a pre-determined limit. Another improvement we would make is to better balance the power budget of our system. When we originally designed the node hardware, we had not yet selected a camera. The camera we did select doubled our power need. We would increase the output of the solar cell and improve the heat management in the charger circuit for a future version of our design. Another improvement would be to introduce environmental trends utilizing machine learning to predict future environment conditions and recommend amendments to the fire danger level. Another thing that we plan to improve for our design in the future is the camera. We hope to find a way to implement higher resolution images because as of right now the cameras are capable with 2 megapixels but the network isn't able to send more detailed images. Though this is not required, we want the users to have the option if they so choose. The final improvement we would make is to allow users devices to connect to the network. With a small low power device, like an ESP32, the user would be able to connect and send plain text messages in an emergency.



5.0 – Standards, Equipment, and Safety

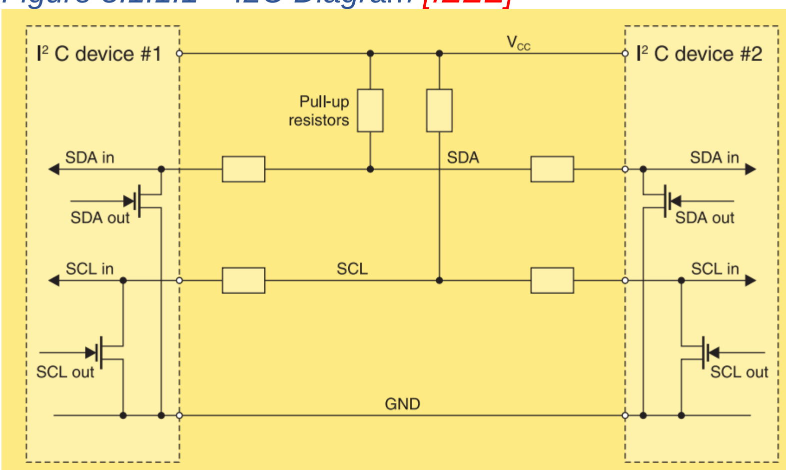
5.1 – Standards

5.1.1 – I2C

One of the communication standards that we planned to use was the Inter-Integrated Circuit, which was also known as I2C. They were used to connect integrated circuits with only two wires that worked in both back and front directions called SDA and SCL into other processors and microcontrollers as a peripheral address and it helped to make PCBs easier to build. As of right now, they usually had a bus speed from a range of 100 kb/s to 3.4 Mb/s. In its protocol, it recommended that it used 7-bit slave addresses and had its data be divided into 8-bit bytes.

The way I2C functions was that first the master sent a “Start” condition for the integrated circuit to listen for data being sent. Then, the master sent the address so that it could access a device and perform a read or write operation so that the circuits compared with their own address and send an “Acknowledge” signal when the address matches. When that happened, the master could send or receive until it ended, in which the master sendt a ‘Stop” signal.

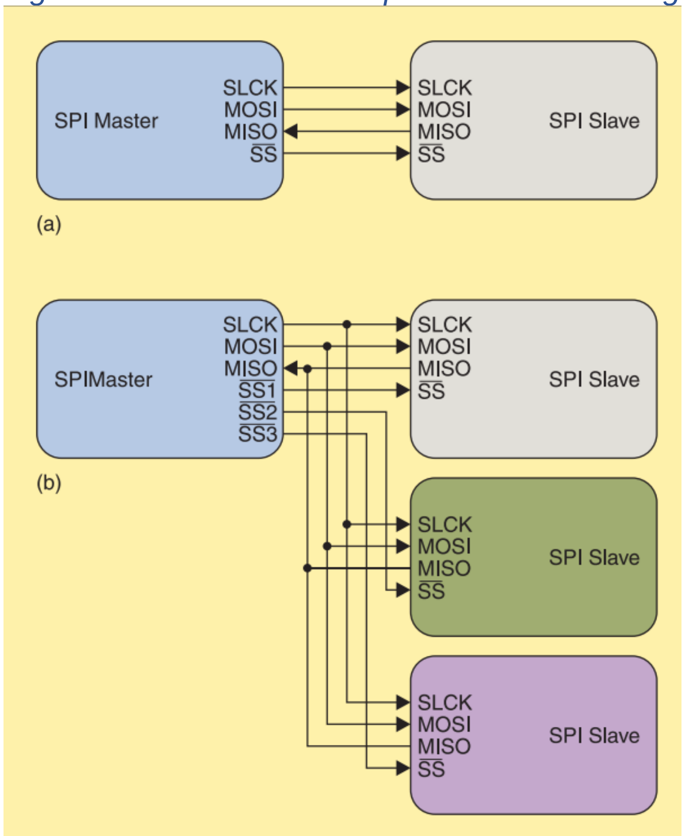
Figure 5.1.1.1 – I2C Diagram [IEEE]



5.1.2 – Serial

Another communication protocol that we worked on was the Serial Peripheral Interface, also known as SPI. It was mainly used to communicate to different systems in a short distance in a full duplex mode and it also consisted of using Masters and Slaves like in I2C. However, unlike I2C and other communication standards, it used more signal lines. The way how SPI worked was that it had four signal lines: SCLK, SSn, MOSI, and MISO. It only contained one type of master that performed all the communication to the slaves. To send data, the master pulled the slave and selected “SSn” line so that it can set up the clock. By doing this, it created data which was then sent to the MOSI line and received data via the MISO line.

Figure 5.1.2.1 – Serial Peripheral Interface Diagram [IEEE]



5.1.3 – Wi-Fi (IEEE 802.11)

For the design we planned to use WiFi in order to transmit the data that we obtained regarding possible fire threats. The standard that we planned to abide by that pertained to WiFi was IEEE 802.11. It allowed us to carry and send data to other types of equipment where it could listen to other channels. Usually, it can be divided into other protocols that contained different types of frequencies:

Table 5.1.3.1 – IEEE 802.11 Protocols

Protocol	Frequency Band	Bandwidth	Maximum Data Rate
a	5 GHz 3.7 GHz	20 MHz	54 Mb/s
n	2.4 GHz 5 GHz	20 MHz 40 MHZ	150 Mb/s
g	2.4 GHz	20 MHz	54 Mb/s
b	2.4 GHz	20 MHz	11 Mb/s

While we worked on the design, we chose the most appropriate WiFi protocol so that it could be transmitted to other devices and networks in a fast and efficient manner.

5.1.4 – LoRa

LoRa, also known as “low range” was a type of networking standard created to connect battery-operated systems to internet networks in a wireless manner. It sent data using the chirp spread spectrum “CSS” technology. It used three different bandwidths which were 125kHz, 250kHz and 500kHz and used spreading factors from a range of SF7 to SF12 to transmit data. To get the symbol rate, we used this formula:

Equation 5.1.4.1 – Symbol Rate

$$R_s = BW / (2^{SF})$$

In that formula, Rs represented the symbol rate, BW represented the bandwidth, and SF represented the spreading factor.

5.1.5 – Power Standards

Certain power standards were determined to see what's the best and most efficient way for a design to function. We planned on using solar energy and reserved batteries to power our product and made sure that they could be safe and still function during inclement conditions. On a fully charged battery, the power reserved ran for three days. By using solar energy, solar panels were going to be connected to the node on top of the product and then to the database.

5.1.6 – Printed Circuit Board Standards

A lot of the design consisted of using Printed Circuits or PCBs. The standards that pertained to printed circuits are IPC PCB and it told how certain aspects should be put in place in those PCBs such as how to properly solder the design and how the cables and wires were assembled. When a PCB was being built, IPC divided that into three classes: general electronic products, which were products where the printed circuit was the main aspect; dedicated service electronic products, in which the PCB must always be reliable and safe from malfunctioning; and high-performance electronic products, where it performed on demand.

5.1.7 – Programming Language Standards

For this project we used using python and dash and created a Web App that displayed real time and historical data of an environment that we were observing. These languages had some sort of standard so that we could understand it better while programming the application

Python

An important standard to follow while using Python was PEP 8. It instructs you how to properly style that code when you wrote it. It sometimes changed overtime because new conventions and standards were modified every year. An important aspect that this standard recommended while writing code was that it had to be consistent so that other people reading the code can understand it. It also recommended that the python code had 4 spaces in every indentation level. The lines that succeeded the past lines from the code should be indented and there shouldn't be any arguments on the first line. It doesn't matter if the conditional lines on the if-statement were written similarly or differently from each other, and lines should have at most 79 characters.

Another standard that was used in Python was PEP 526, which took the syntax of variable annotations into account. It also modified how the syntax was written compared to previous standards such as PEP 484. In that case, it allowed for changes to be made regarding how the code was annotated. This was also a modification of a previous standard known as PEP 3107, which allowed arbitrary annotations to definitions of functions and promoted static analysis.

Dash

Dash was another programming language that we planned to use on the web application, and it was related to C and C++ so it might have shared a lot of standards with that language. Therefore, it abided to C++ standards such as C++14, C++17, and C++20. C++14 which was a recent standard that was created in 2014 as a modification to other standards such as C++11 and worked with other compilers as well.

C++11 back then helped implement changes to the core language and added new features to the standard library to make it safer. Some of the modifications that C++14 implemented included function return type deduction, alternate type deduction, variable templates, and digit separators. In 2017, C++17 modified C++14 by removing trigraphs and updated the memory allocation on the program. C++20 was another update that was done in 2020 that upgraded the programming language's syntax and library, and it also improved the compiler support.

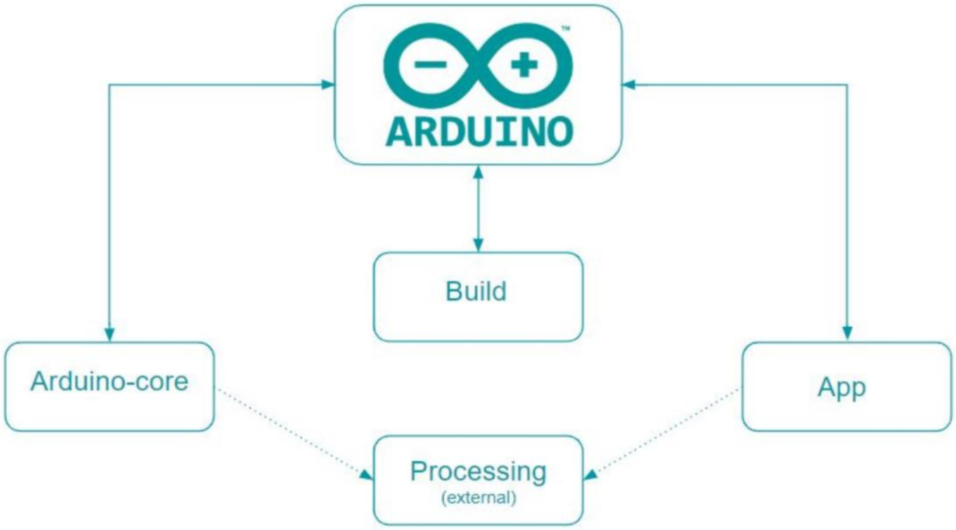
Arduino

In this project, we also planned to use Arduino to help run the ESP32's. To make sure that we were using a proper Arduino board, we made sure that it included a regulator, USB-to-serial interface, and an SPI Programming interface. It was also able to work with both the hardware and software aspects of the design. To implement an Arduino board, a similar design should be followed.

While we worked with Arduino, other standards were taken into account such as the ROHS 2 Directive 2011/65/EU and the Directive 2014/35/EU. These were safety standards that advised you on how to properly handle hazardous substances in electrical equipment and made sure that it used an appropriate voltage amount so that it wouldn't cause any problems. When we looked at the power and electrical components, we had to make sure that they were in a proper temperature range such as being less than 85°C and greater than -40°C so that the product wouldn't overheat or freeze.

To program the ESP32, we used a component called the Arduino IDE, which was an open-source software where we wrote code to program and interact with the board. The IDE was written with C, or C++ so it followed similar standards as I mentioned before. It also provided software applications and microprocessors as well for Windows and Linux and implemented customized libraries and command line tools as well. Arduino IDE was composed into seven parts: Developers, Build Requirements, Testing Framework, Target Platforms, Communicators, Dependencies, and Language. It also used JUnit4 to perform unit tests. The way how Arduino was developed can be shown in this diagram below.

Figure 5.1.8.2 – Arduino Development View [The Arduino IDE]



So basically, Arduino interacted with those modules in a two-way fashion with the Arduino-core, which contained the main code for the IDE; Build, which were extra files that build the distributable zip; and the App. The Arduino-core and the App also shared its dependency modules with the external processing component. To test it, we did it by building the distribution zip by typing "ant dist" on the IDE and then started it where we typed "ant run". To compile what had been written, we typed "avr-gcc" on the IDE.

It was also important that when using the Arduino IDE that proper documentation should be used. When writing and merging code, it had to be consistent and follow proper code style guidelines and those standards were added to its repository.

5.2 – Equipment

5.2.1 – Electrical Fabrication

In some parts of our design, we used different types of electrical circuits and microcontrollers such as the ESP32 Wi-Fi & Bluetooth MCU. They are considered the most important part of the nodes of our design since worked with the wireless communication aspect of the microcontrollers and allowed us to have access to a variety of functions provided by its manufacturer.

5.2.2 – Software

For the software component of our design, we built a web app that was run using Python and Dash. We also used integrated development environments such as the Arduino IDE in order to run the ESP32 boards on our device and the PyCharm IDE in order to record and parse the data from the Python program that we ran. The purpose of that application was to showcase the real time and historical data of the area that we observed. It contained a local area network and a Linux command and control center as well. Besides the web application, the software on our design also contained other nodes as well such as a map view and overlay, off-grid network, and a database.

5.2.3 – 3D Printer

For our design, we printed some of our components for our nodes by using a 3D printer. We used software that allowed us to build or gather information on how the 3D model looked like so that its data can be sent to the printer itself either online or as a SD file. When the data of the model was sent, it started to print the three-dimensional object using a special type of plastic filament and that process usually took a couple of hours. There were various types of 3D printers and the type that we used correlated on the complexity, or the specific part being built for the design. The main types of printers that we used are the FDM or SLA printer, and these printers were known to use different types of material or approaches to print a model.

FDM

The Fused Deposition Modeling printer, also known as the FDM printer was a type of three-dimensional printer that was known for printing models using solid plastic filaments that were available as a string such as PLA and ABS. The filament connected and passed through a nozzle where it got heated hot enough for it to melt. The melted filament then touched the base where it was being printed in which it then dried up and printed on top of the dried filament layer by layer so that it produced a solid 3D model. Also, some commands were written as well so that the machine knew what exact areas, corners, or coordinates it needed to print on the base. It was also one of the most common 3D printers that was used since it was economical and is simple to use. Our group had accessibility to a FDM printer.

There were benefits for using the FDM printer compared to other printers such as the SLA printer. It was user friendly, and it was also not that expensive since it was usually available at around $100. Therefore, it was used by people who were just starting to learn how to use 3D printers. Also, it allowed us to print basic small parts that we threw away and printed anytime due to it being low-cost and simple to use. It was used for creating basic prototype models that we tested first before we implemented it on the design to see if it worked or not. It also didn't have any post-processing when it printed basic designs.

However, there were some downsides to using FDM. Due to the low-cost price of that printer, it was sometimes not that reliable and efficient to use so that meant that we had to make sure that we paid attention that the machine didn't malfunction in the long run. When it printed, it printed the model layer by layer and sometimes there were complications in the heating nozzle that hindered the printing process. The shape of the model sometimes appeared somewhat inaccurate compared to the original design from the data because those details were too complicated to print so that machine can avoid implementing that function, eventually leaving it as printing a simpler variation of the original model that was designed. The model appeared as of low resolution as well. If it was printing a more complex model, then a longer post-processing time was required. FDM was also not compatible when obtaining certain designs that were drawn and created due to its limitations that it had while printing the model.

SLA

The stereolithography printer, also known as the SLA printer, was another type of three-dimensional printer that unlike FDM, used a liquid resin as its source instead of a solid plastic filament. This liquid was inserted inside a small tank of the printer. When it printed, a lifting platform touched the liquid near the bottom part of the tank so that the platform could absorb parts of it. Then, there were UV lasers connected to the printer that shot beams to the liquid in specific directions so that parts of it dried up into a solid piece and began to take the shape of the designed model. It completed this action layer by layer. When a part of the layer was printed, the solid piece was lifted to the top by the platform so that parts of the liquid resin dropped and flowed from the solid model. After that, the platform then lowered the 3D printed model back to the liquid resin to print the other layers. SLA was used to build and print more complex models than was implemented on our design so that it was made more secure and of better quality so that the design didn't malfunction when it was being tested or used.

Some of the benefits of using the SLA printer was that it produced much more accurate-looking models in respect to the original design model being drawn. It also gave the printed object a better surface finish. Due to the material that it used and the light intensity from the laser beams that it shot on each layer which formed a chemical bond, the solid model ended up having much smoother surfaces and shapes. It even allowed the model to have specific details not available in other models such as ridges, edges, lines, and other isotropic parts so therefore it gave the printed object a better resolution in a fast-printing speed. Because of that, SLA was very compatible and allowed us to print a variety of designs drawn using software such as Solidworks due to the efficiency it had while printing. Also, the printer didn't seem to be affected too much by outside temperature since it used the light intensity of the laser to produce the solid model instead of relying on heat like the FDM. Therefore, the model hadn't suffered from thermal expansion which was seen by having small threads of plastic hanging from parts of the model.

However, there were some downsides to using SLA. Unlike other printers like FDM, it was expensive to obtain. Buying one of those machines cost around $3,000 to $10,000. Also, sometimes when the model was finished printing, it needed to be washed and sanded in order to make it smoother. If we wanted to print a bigger model, we needed to buy larger parts to implement on the SLA printer for the model to have enough space when it printed and kept its structure. Since SLA printers required the use of UV laser to print the object, it was exposed to too much ultraviolet violet in which the device was sensitive.

5.2.4 – 3D Printer Materials

Working with three-dimensional printers required the use of different types of material such as filament to print the model. Some of the types of material that were needed to print include PLA, ABS and Resin. Since the FDM 3D printer was more accessible for use, we focused more on the PLA and ABS but also considered other types of filaments as well.

PLA

Polylactic acid, also known as PLA, was a type of thermoplastic polyester that was normally used in FDM printers. The PLA was inserted through the nozzle of the FDM 3D where it was melted from the heat that the nozzle produced so that it printed the 3D mode layer by layer in a given shape. It was able to melt around 190–220 °C and printed at a speed of 10–100 mm/s. This type of plastic was known to be very reusable, and it was also biodegradable. When a 3D model was finished printing using PLA, it had to be washed and sanded in order to make it smoother. While we printed the first parts of the design; it was recommended to use PLA because it can be reused, and we can print many times we want.

Some of the positive impacts that PLA had was that it was cheap since it costs around $20 to $40 and it was easy to use. It was also simple to use in order to format it to print and it was available with different types of colors. It was also able to print at corners more accurately and faster than ABS. Compared to other filaments, it was not that heavy so that meant that we could print as many objects as possible without worrying how much the weight could affect while implementing that in our design and besides that it was also durable. The fact that PLA was environmentally friendly was important for our design since it was used outdoors and involved collecting data about forest fire causes and prevention. Therefore, for the most part, it didn't cause any further problems to the environment where it was being tested even if it was placed there for a long amount of time. It was able to print using a base with a cold temperature without causing further problems to the design.

Even if using PLA came with a lot of benefits, there were negative impacts as well. Since PLA was biodegradable, it wasn't that much of a good idea to leave it outside in the middle of a forest for a long time if most of the nodes were composed of that material because it decomposed and even melted a bit, especially during chances of extreme temperature such as experiencing too much sunlight outdoors. For PLA to start to degrade or even melt, it had to be in a temperature of a range between 50°C to 60°C. So, when we were experimenting with something that used this type of material, we had to avoid being close to fire, hot water, or sunlight. However, if it was placed in an area with normal room temperature, it was able to remain durable.

PLA, like other types of thermoplastic polyester, brought risks of causing environmental impacts such as pollution and landfills. A good thing about PLA was that it was somewhat biodegradable since it was not produced synthetically and contained natural chemicals that came from plants. This meant that if PLA was discarded somewhere on Earth, it was able to break down without causing that much of a landfill and done in a short amount of time. We also recycled models made with PLA when we realized that we were not going to use it anymore, but we had to consider that it had a low melting point. This meant that recycling it with other types of plastic together, we had to make sure that they were separated from each other so that they were recycled properly based on their melting point. We also knew which recycling factories were able to recycle materials containing PLA since some factories focus just on certain types of plastic.

ABS

Acrylonitrile butadiene styrene, which was also known as ABS, was another type of strong thermoplastic polyester that was used in FDM printers. Like PLA it came with different colors and sizes, and it was able to melt at temperatures of a range of 230 - 250°C.

Some of the benefits of using ABS was that it was feasible and able to work with other components such as electronic hardware and machine parts. It was also very durable and like PLA, it was cheap and reusable but however, it withstood high temperatures better, so it was very useful when building a product that required to stay outside in the sunlight for a long amount of time which was case with the Remote Area Monitoring Device. It was also much stronger and withstood environmental impacts better than PLA. This meant that there was a chance that we were able to use models printed with that type of filament when we tested for our final design instead of a rough draft. A model printed using ABS didn’t really need that much care when it was being sanded to make it smoother unlike PLA. ABS was also recyclable so that meant that we were able to melt and reuse the parts that were left out from the original design so that we were able to improvise it.

Despite its benefits, it also had its downsides. ABS tended to warp due to high temperatures which meant that it misprinted certain parts of the model and caused it to detach it from the base where it was being printed. When it was printing a figure, it required the use of a heat bed to keep the filament warm which wasn’t really required while using PLA. We also avoided cooling the 3D model that was being printed with ABS in a quick amount of time because it was able to crack easily.

Like PLA, ABS was also recycled but it didn’t recycle that well because once we did that, the modified product was less durable than it was before. When they were recycled however, it was able to transform into a rubber-like state at 105°C because it was an amorphous compound. Since ABS was not considered to be a “typical” plastic, most recycling factories didn’t accept ABS as a type of material to be recycled, so therefore we had to make sure we knew the exact locations where it can be recycled. Also, since ABS was being reheated during the recycling process, it was therefore weakened in later iterations and couldn’t properly resist heat so that meant that ABS didn't have a long lifespan. In order to increase its lifespan, we had to add new ABS pieces into the recycled ones.

When an ABS model was finally printed, it was not toxic anymore and that made it safe for the environment. However, ABS was very flammable when placed on an area with very high temperature and can end up producing carbon monoxide. Sometimes when a model was being 3D printed using ABS, it released small particles in the air which could be harmful for the user’s health.

ABS was not considered to be biodegradable, and it had a higher chance of ending up in landfills compared to PLA. When it is left on the earth, it takes a very long time to decompose.

Not all recycling factories were able to recycle products with ASA filament since they mostly focus on standard plastics, so that meant that we had to find a proper factory that was able to dispose of it. Also, when the ASA filament was being recycled, we had to make sure that it hasn't been tainted or contaminated with other elements such as water.

PETG

Polyethylene terephthalate glycol-modified, also known at PETG, is another type of thermoplastic polymer filament used for printing objects using a FDM three-dimensional printer. It was basically a modified form of the PET compound that was combined with other elements such as glycol so that was more suitable for 3D printing. Using these elements helped the PETG filament melt at a lower temperature based on the heat that the printer's nozzle produced, and it also made it more bendable like a string. The temperature needed to print something with PETG was between the regular temperatures that printed the PLA and ABS filaments.

The advantages of using PETG was that PLA produced smooth layers while it was printing the design. However, like ABS, the filament produced a strong and sturdy model that was safe from external damage and from breaking apart. It was also recyclable so that meant when we reused old filament to produce new models any time we wanted for our design. Like ABS, it was able to withstand high temperatures and it didn't melt that easily. This also meant that the model was able to remain compact and sustainable from damage even when it was overheated. Another important aspect from it was that it was able to withstand excess moisture which was necessary since our Remote Area Monitoring Device was be placed outside in forests for the most part where rain and high levels of humidity occurred since it was one of the nodes that we hadn't measured. It was not that complicated to manage and to print something with that filament for beginners.

Some of the disadvantages of using PETG was that the final product being printed contained problems that were not seen with other filaments that we mentioned such as scratches on the surface of the model. Also, the product had to be placed away from certain elements such as ultraviolet light since it caused damage to it. At higher temperatures, the PETG model was not able to remain firm and compact; at above 176°F, it ended up being softer.

The reason why PETG was easy to recycle was because it was made from a plant-based product called glycol which was also environmentally friendly. It was also accepted by many factories as an appropriate plastic to recycle, so that meant that we hadn't really been that concerned about finding a faraway place to discard material made from PETG in order to recycle it. However, it was not that biodegradable because it was mostly made from oil-based products instead of raw materials. Therefore, it was able to last a long time in a landfill and had caused environmental problems in oceans due to the elements that PETG contained.

ASA

Acrylonitrile Styrene Acrylate, also known as ASA, was another type of thermoplastic polymer filament used for FDM printers as well. It was known to be a modification of the ABS filament and it's usage was recommended for building and working with outside products and for building prototypes.

Some of the positive impacts for using the ASA filament in the FDM three-dimensional printer was that it was resistant and had not gotten damaged that easily when there were heavy changes in temperature or when it reheated unlike its counterpart ABS. Therefore, it was useful when building a piece for the circuit used that material since our device was used outdoors for a long amount of time with noticeable amounts of sunlight. The reason why it was so resistant was because it had high durability and it withstood heavy amounts of ultraviolet light, which was mostly produced from the rays of the sun. It was also resistant from environmental impacts and hadn't corraded that easily. Therefore, ASA was more durable and had a longer lifespan compared to other materials like ABS.

However, it also brought some negative impacts. It was not always that safe and environmentally friendly because when printing something using that filament, it released toxic chemicals and it was also more expensive compared to its counterpart, ABS. It was complicated when printing a 3D model. ASA had a higher melting point so that meant that we had to make sure that our 3D printer's nozzle was able to produce a strong amount of heat so that it was able to melt properly. It was somewhat expensive and hard to find a printer with a nozzle that was able to fill that requirement.

Not all recycling factories were able to recycle products with ASA filament since they mostly focused on standard plastics, so that meant that we had to find a proper factory that were able to dispose of it. Also, when the ASA filament was being recycled, we had to make sure that it wasn't tainted or contaminated with other elements such as water since elements from the ASA filament could have easily spread to the water and spread toxic chemicals around it.

5.3.5 – Digital Voltmeter

In this project, we worked a lot and gathered data for our nodes such as the voltage and the current. We saw how the PCBs and the microcontrollers were functioning. One of the types of equipment that we worked on was the Digital Voltmeter. The voltmeter helped read the alternating or direct voltage current of a device and it also displayed the value of the current as a numerical value. It also measured the input voltage, and it did this by converting the analog voltage to the digital voltage of the device being measured. The voltmeter was then divided into five parts which are the Pulse Generator, Voltage Control, Counting Clock Pulses, Analog to Digital Conversion, and the Latching and Display Section. For our project, we used the DC voltage of certain nodes that we are going to test.

The Pulse Generator helped create a pulse in order to generate voltage. The Voltage Control looked at the voltage from the input and to the one that passes through the capacitor. To make sure that it got an input current that was close to zero, it used an Operational Amplifier chip that took the voltage from a positive and negative value and determined whether the output voltage was high or low based if the positive input was greater than the negative input. The voltmeter also checked and counted the clock pulses between the charging signals and got those results as a binary number value. The Analog to Digital Converter helped convert that voltage into a binary number and the Latching and Display Section displayed the results of all those calculations from the capacitors as an output.

For our design, we used two types of digital voltmeters when we designed and tested the product which were the BK Precision voltmeter and the Hyelec digital multimeter. Both voltmeters functioned similarly but the BK Precision voltmeter was used more for our project since it gave out more accurate outputs and contained more functions. Also, we used a mini-3-wire standalone voltmeter to read out the solar panel voltage in the first iteration of the test circuit before obtaining the Arduino microcontroller. It was simple to use but the output results were less accurate compared to the other voltmeter. The way the small standalone voltmeter worked was that it connected the red wire to a power supply and the black wire to the common ground. It also contained a microcontroller that read the voltage value. It attached to almost any place on the circuit.

The way how the digital voltmeters were used for our project was that it gathered data from certain nodes on our design. We also made sure that the microcontroller from the circuit had a maximum input of 3.3 V before connecting it to an ultrasonic anemometer. By getting the values, we decided whether we had to adjust the input voltages for the circuit in order to get an appropriate DC output voltage for the anemometer or not. When we obtained these values, we used the minimum and maximum function on our digital voltmeter. From these values, we saw whether we needed an analog line level translation and a 7-volt supply and improved our design for the anemometer.

5.3.6 – Oscilloscope

The oscilloscope was an electronic device that was used for measuring and displaying electrical voltage waves as a graph. In the graph, it showed the voltage level over time from the circuit that was being measured and it also showed the frequency. We saw how much DC and AC current flowed and if there was too much noise in the signal. We also noticed whether there were irregularities in the oscillations which meant that there were problems within the transfer of voltage in the circuit.

When the voltage was displayed on the oscilloscope as a graph, it was displayed in different types of waves such as a sine, square, or triangle wave. We could tell that there was a change in voltage based on the height of each wave and we were also able to adjust the waveform as well. When there was no change in voltage over time, it was shown as a straight, horizontal line on the graph. We also determined the amplitude and phase as well. In order to obtain voltage data to send it to the oscilloscope, we used probes that were connected to the channels of the device so that we can connect it to the circuit.

The way we used oscilloscopes for our project was to also look at the amount of voltage that it flowed for our ultrasonic anemometer. We also saw whether there was a change in stability based on the wind that was being blown on the circuit's sensor through a phased-locked loop. We then took the current voltage as the input and then get the output oscillation to create a feedback loop which was shown as sine waves. Then with the exclusive or gate, the output passed through a low pass filter in which was then shown a square wave. Eventually, these measurements gave us an idea of how the wind of the area that we tested was like.

5.3.7 – Power Supply

The power supply was what it was needed to power certain aspects of our design such as the circuits. It also enhanced the circuit's electrical current in order to convert it to the appropriate current, voltage, and frequency. Usually, they were either 3v or 5V regulated power supply circuits. There were also different types of power supplies such as Switched Mode, Uninterruptible, AC, DC, and Regulated.

For our project, we mostly focused on the DC power supplies. Some of the devices that we used include the Yescom 110V AC 30V 10A DC Power Supply. This device had the ability to produce constant voltage and current and had over voltage, open circuit, and over temperature protection and it was used for 24 hours and contained a fan that prevented the power supply from overheating. The device also contained some probes that we connected from the channels of the power supply and then we connected them to circuits so that voltage can flow through them. We also used the Hewlett Packard DC power supply as well.

Most of the equipment that we used such as the digital voltmeter, oscilloscope, and power supply were accessible for us to use because one of the members of our group owned those pieces of equipment.

5.3.8 – Soldering

We worked with a lot of circuits and microcontrollers, and an important aspect of working with those objects was that you had to connect a lot of pins and wires to those microcontrollers, and we made sure that they remained connected for a long time. So therefore, we needed to melt small pieces of metal into small holes where the wire and the microcontroller were connected so that the metal dried up and became solid so that these two pieces remained stuck. This created an electrical connection between these two nodes. The process of doing that is called soldering.

For this project, we soldered pieces of the circuits such as the camera, ultrasonic anemometer, battery, sensors, and other microcontrollers to other wires and pins so that power be transferred from the supply to the specific part of the circuit for it to function.

Soldering Iron

To do this process, we used a soldering iron. This device was connected to a power supply to power it up so that the soldering pen heated up. The pen heated up at around 365°F so that it melted the string of metal alloy when it touched the pen, but we were able to adjust the temperature whenever we wanted so that it allowed the metal alloy to melt faster or slower. Also, when we finished melting a piece of metal alloy through a part of the microcontroller, we had to clean the tip of the pen with a small sponge so that it removed all the pieces of the molten metal so that it wouldn't interfere with having excess metal when we soldered another piece of the microcontroller. When there was too much metal alloy in the microcontroller which happened to each touch with another hole that has been soldered, it caused problems to the power connection of the circuit; sometimes, it prohibited the circuit design from running.

Soldering Gun

Another tool that helped us out with the process of soldering parts of the circuit was a solder gun. It was like a soldering iron, but it was a more simplified version of it. Like the soldering iron, the gun contained tips that was heated so that it melted the metal alloy when it touched it. But unlike the soldering iron in which we had to manually control the temperature to make it hot enough to melt metal alloys, the solder gun contained a trigger that allowed us to control the temperature of the tip so that we melted the metal any time we wanted without just touching it and waiting for it to heat up. It did this at a faster rate compared to the iron and can make it hotter. It also tended to work better on surfaces that weren't smooth such as edges and small corners and can even repair parts that had been already soldered before.

Some of the advantages of using a soldering gun is that it heated and cooled down quickly and that it used an exceptional power voltage, but it was not recommended when working with small pieces. The soldering iron was much simpler to use and it's good when working with less heavy equipment and the pen doesn't produce too much excessive heat. We had access to both types of soldering devices since one of our group members owned them.

There were also different variations of soldering guns to use. One of them was the solder air gun. Unlike the regular soldering air guns, this version melted the metal alloy by blowing heated air to the area where it was being soldered instead of physically touching the metal alloy string to melt it. Then soldering gun modification was beneficial in the fact that not only it melted the metal alloy solder, but it also melted away the alloy in order to de-solder it by blowing it away with heat unlike using a heated soldering pen to pick it out. In the original method, it sometimes pushed the alloy further directly through the hole which made it even harder to take it out and even caused the PCB to malfunction. Also, they weren't that expensive to obtain compared to other soldering devices. However, they came with their disadvantages as well. Soldering air guns weren't always that accurate; when we used it to solder something, the tube's area from where the hot air came from was too big and heated up more space than the necessary portion needed for the circuit. This ended up causing some damage to the circuit's surface.

Reflow Oven

When soldering pieces for our design, we then had to put together electrical components on top of PCB's. In order to do this, we used a reflow oven. The reflow oven helped us save time compared to soldering circuits the traditional way such as using the soldering iron. It was also very efficient to use and heated up on specific areas of the circuit more accurately compared to other methods. We had used other methods such as reflow ovens since other soldering tools aren't able to adequately solder the wires to the PCB. It was especially necessary when we were soldering very small pins and if they were placed in the bottom of the circuit, since it was kind of difficult to reach it in a regular soldering tool.

The way that it worked was that it used solder and flux to connect the electrical components to the circuit by first configuring a design of the holes of the PCB where it needed to be soldered and then adding the solder with a screen printer. After that, the oven placed and aligned the electrical components to the soldered paste that was added on certain parts of the PCB. It was then placed into a special oven where it heated up the paste and then it dried up, leaving the two parts stuck together. When the product was being heated or cooled through the reflow oven, it preheated at a range of 4°C to 150°C and it also soaked the connected PCB in a temperature range of 150 to 170°C. When it actually reflowed, it heated the product in a temperature a bit greater than the metal's melting point in about a minute. Finally, the oven cooled the soldered circuit at a temperature of about 4°C.

Reflow ovens were beneficial in some ways because it hadn't produced too much thermal shock and it hadn't wasted too much soldering metal alloy. These ovens were also recommended for low-volume production as well. Since most of the operation took place inside an oven, it hadn't really excreted that many chemicals such as flux from burning metal, so it was environmentally safe. However, there were negative consequences from the usage of reflow ovens such as that it was difficult to control when using more complicated circuit boards and that it sometimes had temperature control problems. It was also slow and difficult to obtain that device because it was expensive.

5.3.9 – Microscope

When we were soldering and connecting the wires to the PCBs and the microcontrollers, it was sometimes complicated to see exactly where it needed to be soldered from the naked eye. That was why we used a microscope in order to carefully see where the piece is being soldered and connected. By doing that, it helped to prevent committing an error in the process that can end up causing the circuit to malfunction. One of these types of errors was that some of the holes of the PCB were too small to see or are placed very close to other holes. When those cases happened, we couldn't see whether we were soldering too much or too little. If we soldered too little, then the cable won't be able to remain stuck to the PCB. If we soldered too much, it wasn't allowed the power to flow efficiently. It became harder to carefully remove the soldered metal alloy off without damaging it and after we removed it, there was a chance that the PCB won't be able to function properly again after to re-solder it. Also, since some holes appeared to be too close to each other, we could have accidentally melted the metal in a way that it touches both holes which would have fatal for the PCB because it won't be able to transfer any data or power. Using a microscope was also necessary when working with small pieces and nodes such as the circuit connecting to the camera, the wireless controller, the small power supply, and the magnetic wind speed detector.

5.3 – Safety

5.3.1 – RoHS

When working with the production of electronic devices, we had to make sure that anything that's hazardous was under control. Therefore, there was a standard that we abided upon called the Restriction of Hazardous Substances Directive, also known as RoHS. These standards helped let us know which proper electronic devices were being used because it prohibits the usage of ones that can be hazardous or dangerous such as Lead, Mercury, Cadmium, Hexavalent chromium, Polybrominated biphenyls, and Polybrominated diphenyl ethers.

Abiding these standards was crucial for our project because when we were looking for elements to implement on our design, not all of them were used and we had to throw them away and if those elements didn't follow the RoHS guidelines, then it could have caused dangerous environmental impacts such as pollution. The way how electronic devices were tested for RoHS compliance was by using X-ray fluorescence, which detected harmful substances.

RoHS was divided into six different levels or modifications based on the elements that it restricts which was seen in this table below. The “x” value in the boxes represented which element was restricted by the respective RoHS standards.

Figure 5.3.1.1 – Element Classification per RoHS Standard

	RoHS 1	RoHS 2	RoHS 3	RoHS 5	RoHS 6
Lead (Pb)	x	x	x		x
Mercury (Hg)	x	x	x	x	x
Cadmium (Cd)	x	x	x	x	x
Hexavalent chromium (CrVI)	x	x	x	x	x
Polybrominated biphenyls (PBB)	x	x	x	x	x
Polybrominated diphenyl ethers (PBDE).	x	x	x	x	x
Bis (2-ethylhexyl) phthalate (DEHP)		x	x		
Butyl benzyl phthalate (BBP)		x	x		
Dibutyl phthalate (DBP)		x	x		
Diisobutyl phthalate (DIBP)		x	x		

5.3.2 – Solder Safety

When working with a soldering tool, we had to be aware of certain risks. It was important to learn how to avoid them. The soldering iron’s pen reached very high temperatures such as 400°C so it was important for us to always take proper precaution and we made sure that we were wearing protective material such as gloves to prevent ourselves from getting burned. When cleansing the soldering iron pen after using it, we had to make sure that the sponge that we use to clean was wet because since the pen had a high temperature, it had a potential of being flammable. After that, we made sure that the soldering iron’s power was off and paid attention where we put the pen and not leave it unattended, due to the same reason why we needed to make sure that the sponge was wet.

Not only was the soldering iron flammable, but it also had the possibility to release toxins and chemicals such as flux, lead, and rosin that without proper precaution, could be harmful for our health. That is why when we were worked with it, we had to make sure that our eyes and hands were always protected and avoided using soldering machines that contained lead and rosin because elements like lead can pass through our skin and produce harmful fumes. When we soldered a PCB, flux was produced as fumes when the metal alloy was being melted into the circuit. Flux contained an element called Rosin.

Inhaling rosin, even in small quantities could be extremely hazardous and can cause lung and skin damage and headaches.

There were many other methods to control those risks. One of those methods was that we had to make sure that we were in a safe environment when we were soldering. That meant that we worked on that task in an area with proper ventilation or even outdoors. Since toxic fumes could come out when soldering, we also acquired carbon or bench top filters that extract those fumes. The filters also required constant revisions to make sure they were working properly. Also, proper training was necessary when soldering and it was also recommended that at least two people were present when completing a task using that tool and make sure that the group members hadn't had any health problems beforehand. There were no people in the group that presented certain health problems so that meant that we are all safely contributed to the soldering process.

We made sure that no broken cables on the soldering device caused electrical problems. Also, while working on that, the soldering iron was placed on a fireproof surface and away from other electrical cables since it had a high temperature, it could have caused sparks while touching the cable and therefore cause flames. We used a grounded outlet when the soldering iron had a short circuit. We had access to first aid close by as well. When throwing away waste from working with the soldering device, we had to make sure that it was discarded in a protected box or container and knew what exact type of waste was being thrown away, so we did this by dividing the containers with the appropriate waste.

Solder Fume Fan and Filter

Since soldering can release too many fumes that was unhealthy to the user operating the device, we found devices that helped us eliminate the excess fumes such as solder fume extractors. There were many different types of fans and filters available, but we decided to use the Solder Sentry or the Weller WSA350 120v Bench Top Smoke Absorber.

The Bench Top Smoke absorber was a device that was used to removed toxic fumes from soldering by using filters and fans. Since we soldered mostly in an indoor area, using that smoke absorber was necessary especially when a high quantity of flux was being released. The way the flux was being filtered was by absorbing them through carbon filters and then replacing the filter if they stopped working by any chance. Compared to other types of filters, it was reliable and had better performance since it ran for a long period of time, which was necessary because a major component to produce our fire prevention device included soldering. It also had a high-rated airflow so that it filtered more toxic fumes. We were also able to adjust the absorber into different positions based on where and how we wanted it to absorb the toxic chemicals. However, a downside of using that absorber is that it hadn't included a power rating. It also mostly seemed to work with filtering flux from solder that it hadn't contained lead.

The Solder Sentry was able to retract fumes from the flux being produced from soldering without relying on a carbon or charcoal filter. Instead, it used a high efficiency particulate air filter also known as HEPA, which was more efficient since it was able to filter out almost all air particles that came out from the fumes while the other types of filters only took out odor. Even without odor, harmful particles still came out from flux so that was the reason why we used more improved filters such as the solder sentry.

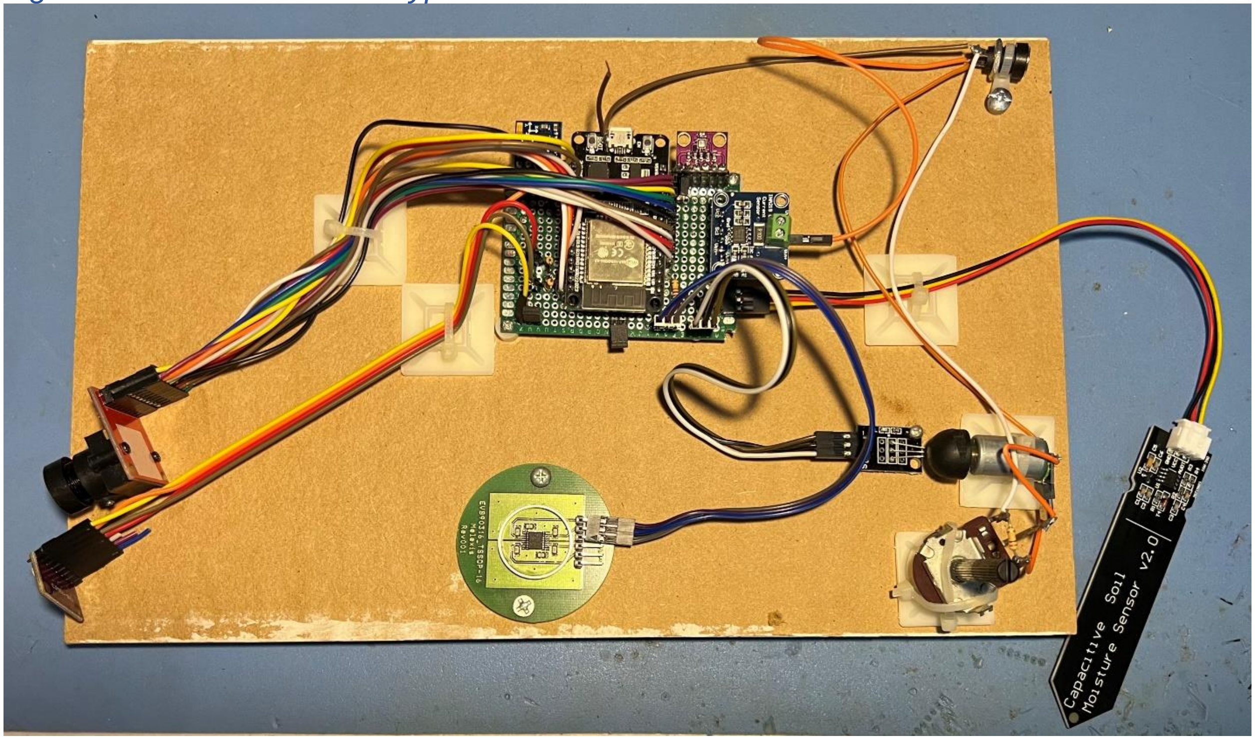
Also, the Solder Sentry’s fan was placed in a different position compared to other filters. It was placed in front of the filter while other machines placed it on the back of the filter. The fan positioning was important because it pulled the air toward the middle of the assembly and took out the excess air after it. The way how the air flew determined the durability of the filter. Since the Solder Sentry placed the fan in that manner, it dispersed air more efficiently all throughout the filter which then allowed it to capture more flux particles.

6.0 – Prototyping and Testing

6.1 – Prototype Design

We developed a basic system prototype as we progressed through the testing of our individual sensors. This prototype represented a complete set of sensors and hardware. The only exclusion was the power system was replaced by a wall power supply. From a software perspective, this prototype was designed to act identically to a real node. We also included the ability to simulate wind for the wind sensors. This prototype acted as the development platform for our early software design. Here we quickly exchanged parts as required during testing. This also allowed us to have tested and functional software running before we received all the full prototype parts and PCB. For an overview of our development test bed, see Figure 6.1.0.1.

Figure 6.1.0.1 – Basic Prototype Test Bed



6.2 – Testing

The testing for the major hardware components of our project was captured in the part selection section 4.2. This section stepped through each component and tested their functionality individually. Once we began integrating these components into the prototype design, we focused on validating our existing test results. This meant that we ensured the operability of each sensor and that the output wads as expected based on previous testing.

The testing of our software design evolved as we began to create the system code. At this juncture in the project design, we laid out a broad overview of the testing we expect to complete. First, for each node, we ensured that the sensors communicated and reported their data properly at the local node level. This was completed by directly connecting to the node with a development computer. Next, we tested the basic functionality of the mesh network. This was completed by ensuring that the nodes were connected to the network automatically. The network needed to handle unexpected disconnection of a node. The network also needed to function as a mesh meaning that a far node could be connected to the aggregator through an intermediary node. The next set of tests for the software occurred on the aggregator and web server side. We verified that the aggregator computer could be communicated through the attached aggregator node to the wider network. This was also the point we verified the network capacity could handle our data throughput for both sensor data and images. The final test for the software was a full system test. This test was a validation of all pieces of the system. We started off as a new system installation and progress to a fully functional network with a minimum of three nodes.

7.0 – System Architecture

7.1 – Electrical Schematics

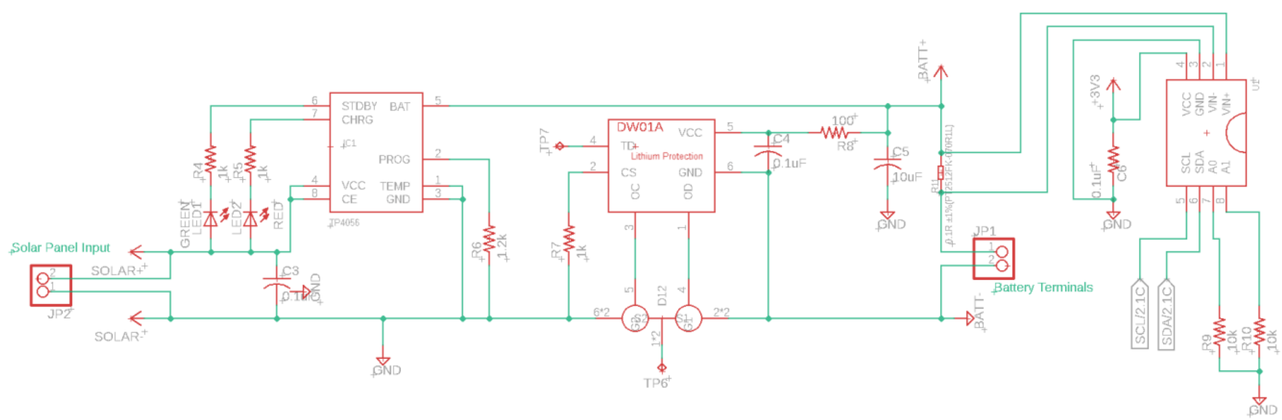
The electrical schematics depicted the connections between components within our system. These connections defined the functionality of our system. The components were depicted in red often with pin names and numbers. These pins were connected to nets in green. A net name was assigned to each group of interconnections. When a connection passed from one side of a component to the other, the net name changed. These net names were the map for laying out our physical circuit board in the next sections.

Throughout our design process, we maintained a prototype design methodology. We understood that this was an early stage in the product design lifecycle. With that, the design needed some revision as we progressed through the testing process. Our design methodology incorporated a few key points of consideration. First, we connected test points anywhere that may have benefitted. For example, all unpopulated pins of the microcontroller and all power supply lines. These test points gave us a good place to probe the prototype with test equipment like an oscilloscope. Additionally, these points gave us an easy to work with point to solder to. This greatly increased the speed and reliability of any board-level rework we may need to carry out. The second design consideration we made was to include spots for optional components even if we chose not to populate them initially. A good example of this was the pullup resistor on the address pins of some of our sensors. The address can either be grounded or pulled up to power. This change allowed us to select a different address for the sensor. This may be beneficial if two devices shared an address as this caused communication issues with I2C. By adding the required components to the design, we retained the ability to use this feature should the need arise. We also left the physical pads empty without impacting operability or cost. The third design consideration we made was in part selection. All the components came in different packages. The package referred to the physical characteristics of the device. One such characteristic was the size and spacing of the pins of the device. Where possible, we selected components with externally exposed pins with the largest spacing available. The reasons for this are two-fold. First, this style of component was much easier to solder by hand without the need for any specialized equipment. Second, in the event we needed to troubleshoot the device, the pins were easily accessible. This helped for both visual inspection and rework. This consideration was possible because space was not at a premium in our design and the cost difference for this style of parts was minimal.

7.1.1 – Power System Schematic

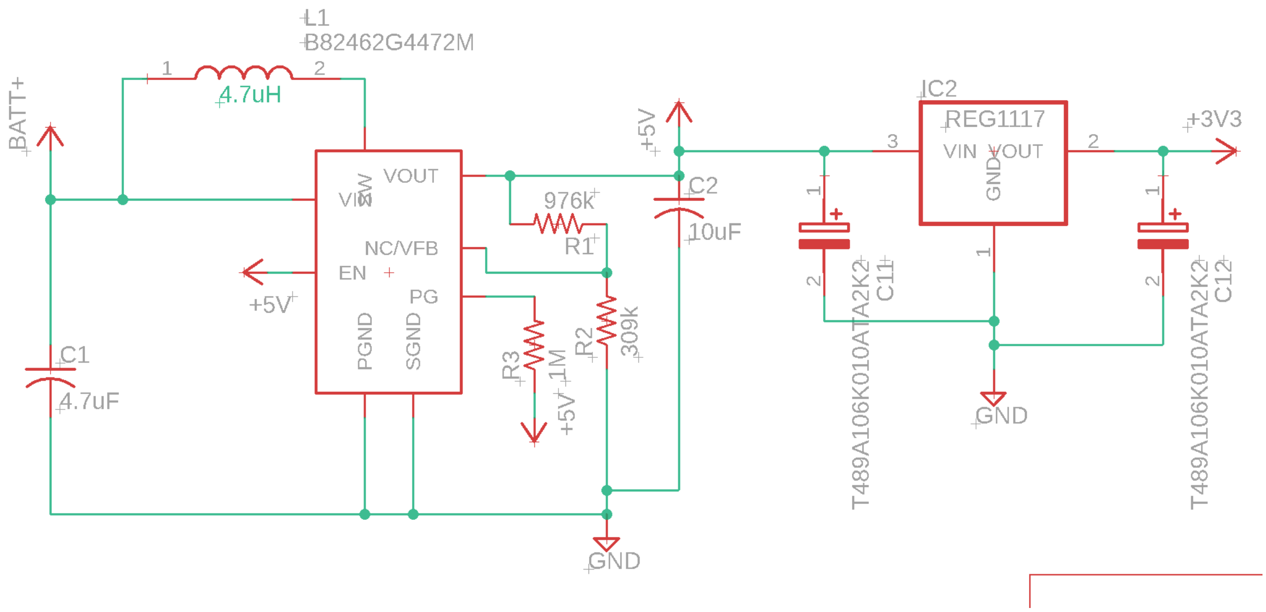
The first subsystem to discuss is the power storage and generation design. We used a TP4056 charge controller that regulated the unstable input of the photovoltaic panel. See the left-most IC in Figure 7.1.1.1. Throughout the day, the power generated by the photovoltaic panel was likely to change often with the conditions of the environment. The TP4056 took this unstable input and produced a stable output to charge the battery.

Figure 7.1.1.1 – Power Generation and Storage



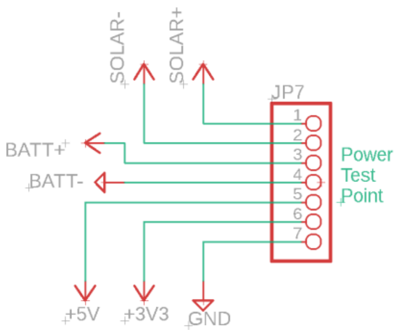
The next major component of the power system was the DW01A lithium battery protection circuit. See the center IC in Figure 7.1.1.1. This device provided protections for current, and voltage passed to the lithium battery. This component was important for safety as well as prolonging the lifetime of the lithium cell. This was because lithium-based batteries were sensitive to over or under discharge and over-current conditions. The DW01A mitigated these risks. The final major component in the power generation and storage subsystem was the INA219 current sensor. See the right-most IC in Figure 7.1.1.1. This sensor was used to monitor the health of the power generation and storage system. The INA219 measured the voltage drop across a 0.1-ohm resistor. With a known resistance and a measured voltage, the current was calculated using ohms law. This sensor also provided the voltage between the reference ground and the positive terminal of the battery. This data was available to the microcontroller over I2C. The current allowed us to measure the power input / output and detect any issues with the solar charging system. The battery voltage allowed us to monitor the charge status. If the battery voltage fell below expected operating conditions, a system administrator was notified.

Figure 7.1.1.2 – Voltage Regulation



The next power subsystem to be discussed is the voltage regulation circuitry. Each node required two voltages to properly function. Most of the device functioned on 3.3 volts. The wind speed sensor required 5 volts. The leftmost IC in Figure 7.1.1.2 was the MCP1642B, a switching boost regulator. We selected a boost type regulator because the battery voltage was always supposed be below 5 volts. The MCP1642B came in adjustable and fixed voltage output packages. During part selection, we noted the adjustable package had wider availability over the fixed voltage package. We preferred the fixed voltage package to reduce the number of supporting components required. Our design allowed us to use either package type by populating the resistors R1 and R2 to set the adjustable input if required. The rightmost IC was the REG1117 3.3-volt regulator. This regulator took the output of the 5-volt regulator and reduced the voltage to 3.3 volts. We chose to use this type of regulator to reduce the design complexity. In contrast, switching regulators required a more parts and specify a layout for best performance. We expected a 10mA quiescent current drew from the REG1117. During testing of the fully integrated prototypes, we monitored the current draw to quantify the impact of this design choice. We considered a different regulation method for 3.3 volts in future design revisions.

Figure 7.1.1.3 – Power Test Points

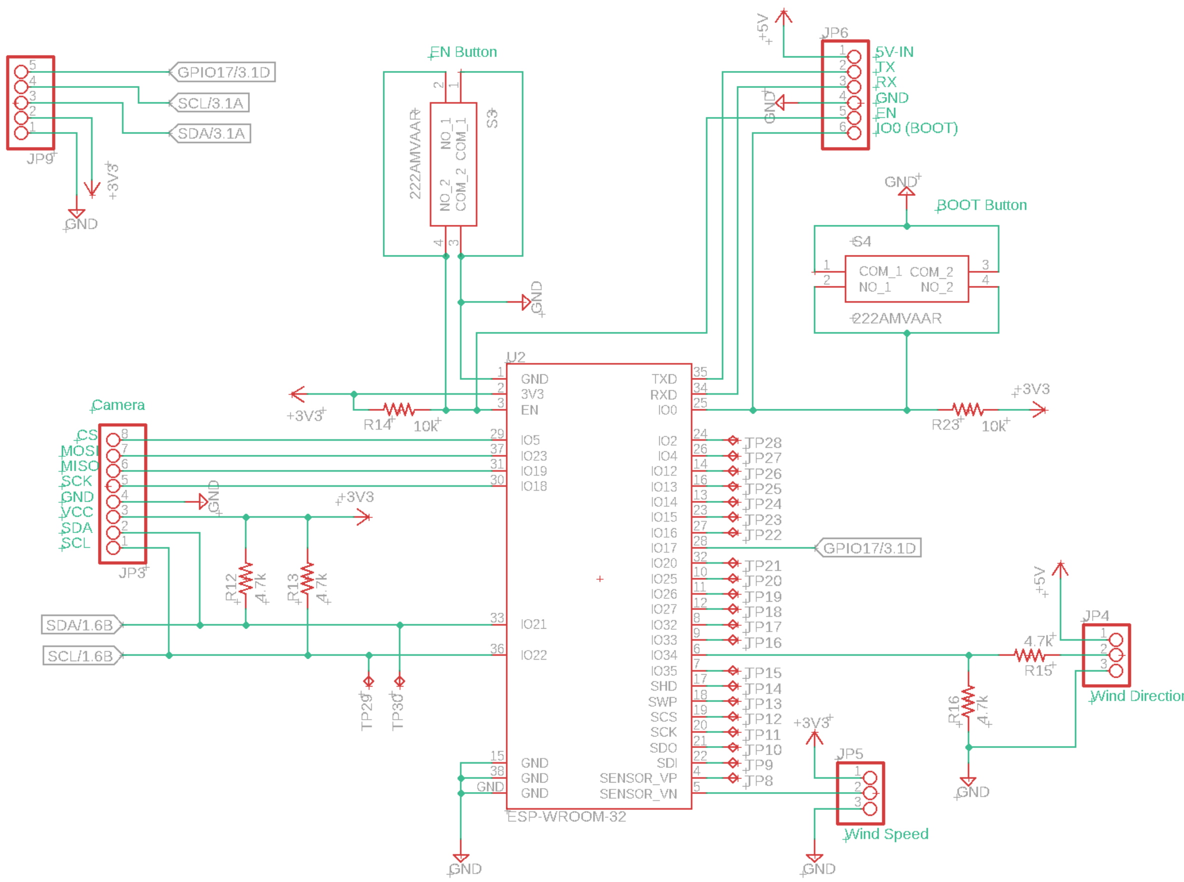


The final component of our power subsystems was a test point for each of the inputs and outputs. This test point allowed for troubleshooting and rework. We used this to monitor noise or voltage levels and supplied external power or route power to somewhere it was needed.

7.1.2 – Microcontroller Schematic

The microcontroller was the center of each of our nodes. This section discusses each of the features and connections to the microcontroller. These consisted of features from on board programming to sensor communications.

Figure 7.1.2.1 – Microcontroller Schematic



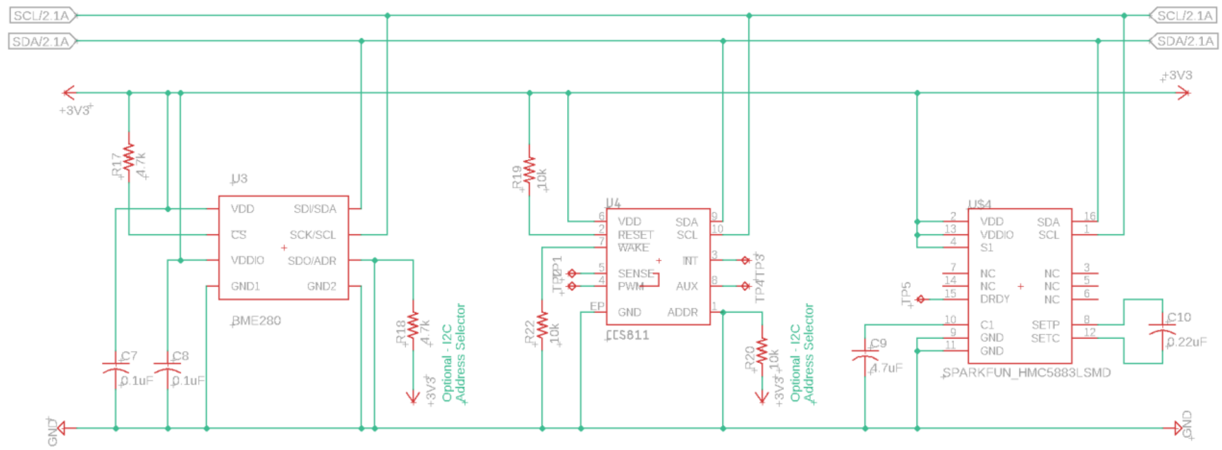
We began from the top left of Figure 7.1.2.1 and worked our way around the microcontroller. The first component was a standard tenth inch header for connecting our microcontroller to some of the environmental sensors. The first of which was GPIO17. GPIO stands for general purpose input output. This pin was responsible for the 1-wire communication to the DS18B20 temperature sensor. The next two lines are SCL and SDA for I2C communications. These lines were often useful to monitor using an oscilloscope during development. The final two lines were power and ground. Moving clockwise around the microcontroller, the next set of components were used for programming the microcontroller. These components consisted of two buttons and a programming port. The buttons were for selecting boot mode and enable. These buttons were not required depending on the programmer we chose to use. They were included to afford us the ability to pivot to a more general programming device if the device specific to our microcontroller did not function as expected. The programming port itself exposed the transmit and received lines of the microcontroller. Along with a 5V power input and the microcontroller specific boot selected and enabled lines. The next component was a labeled net for GPIO17. This net was connected through the header to the sensor itself on the sensor portion of the physical circuit board. After this, we came to a set of two headers. Both headers were for connecting to the wind speed sensors. The wind direction sensor produced an analog signal with a maximum voltage above the microcontroller's analog to digital converter maximum voltage tolerance. As a result, we passed the signal through a voltage divider with the center tap connecting to the input of the microcontroller. The next set of components on the right side of the microcontroller were the communications lines SDA and SCL for I2C. These lines were pulled up to power. They were also connected to the camera connector. The camera module was controlled by I2C and sent pixel data over the serial peripheral interface (SPI). The camera connector was based on standard tenth inch center headers. This allowed for wide compatibility when we selected the internal connector for the camera. This connection was made using a cable to reduce the mechanical design complexity.

As a final note on the microcontroller schematic, we had included test points for all unused pins. These test points were removed for a production design. They were prototype aids. The test points gave us a good place to connect test equipment or rework the board layout without the need to order a new board and accompanying parts.

7.1.3 – Sensor Schematic

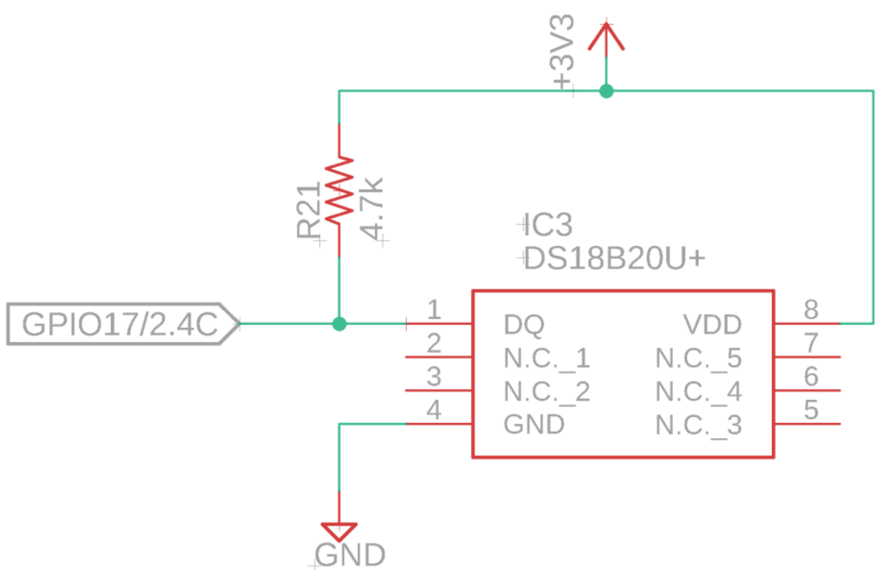
The sensor schematic consisted of three I2C sensors and one 1-wire sensor. Though the sensors were populated on the same board as the microcontroller, the connections were made through the header in Figure 7.1.3.3 for prototype considerations discussed in the physical layout section 7.2.

Figure 7.1.3.1 – I2C Sensors



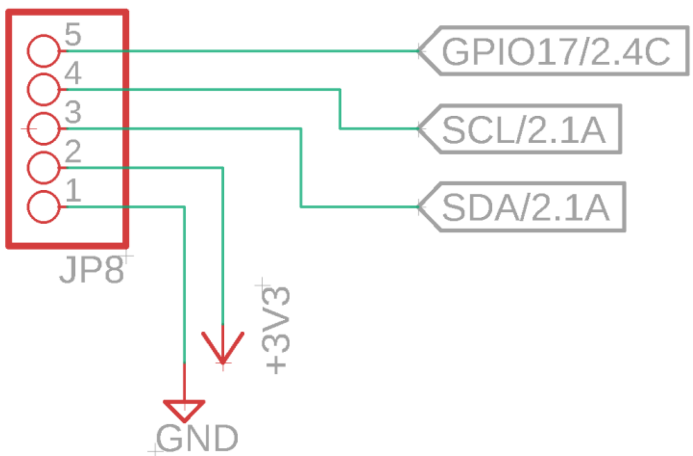
Three of the sensors in our design communicated using I2C. These sensors were all hooked to their respective supporting components individually while sharing the I2C lines. The first two sensors from left to right have settable addresses. These addresses may be set using a pull up resistor. We left this resistor unpopulated for both sensors to set the default address. The leftmost sensor was the BME280. This sensor was capable of either I2C communication or SPI communication. We pulled the chip selected and VDDIO pins high to enable the I2C interface of this sensor. All the peripheral minor components for each sensor were chosen based on the respective datasheets.

Figure 7.1.3.2 – DS18B20 Temperature Sensor



The DS18B20 sensor in Figure 7.1.3.2 was for accurately measuring temperature. The communication standard used for this sensor is 1-wire. This connection standard was developed by Dallas Semiconductor Corporation, the same company that manufactured the sensor. The 1-wire standard was tolerant to long distance for low-speed communication. The DS18B20 was available in several package types. We selected the largest surface mount part available. Size was not a restriction for this part of our design. We also gained the benefit of large pins when it came time to build the prototypes. The package had eight pins, only three of which were connected either internally or externally. This sensor was redundant to the BME280. However, the simplicity and low cost allowed us to add the DS18B20 with minimal impact to the progress of the project.

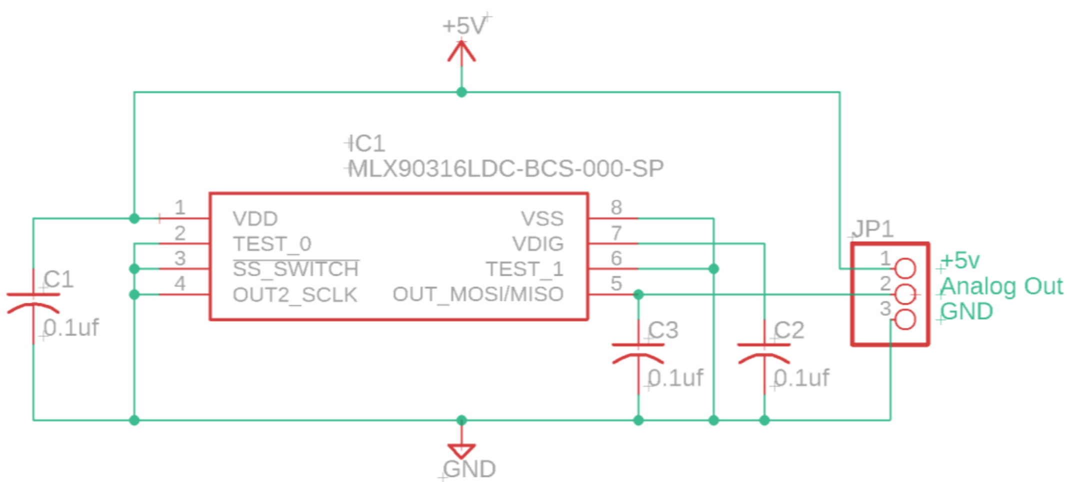
Figure 7.1.3.3 – Sensor Communications and Power Header



7.1.4 – Wind Direction Sensor Schematic

The wind speed sensor was the Melexis MLX90316 with its connections shown in Figure 7.1.4.1. This sensor was a rotary position sensor operating by measuring the magnetic flux passing through the part. During development, we used an evaluation board designed by Melexis featuring one of their 16 pin options. The 16 pin package offered two full sensors together on the same part. This additional functionality was not necessary for our purposes as we only need a single output. For this reason, we selected the 8 pin package for our prototype design. The function of the sensor we were using remained the same.

Figure 7.1.4.1 – Wind Direction Sensor Schematic



The MLX903016 was the only component in our design to require a 5-volt supply. It was important to note that this circuit was separate from the main circuit board. This explained the overlapping of component names. The header was tenth inch centers and connected to the corresponding input header on the main circuit board.

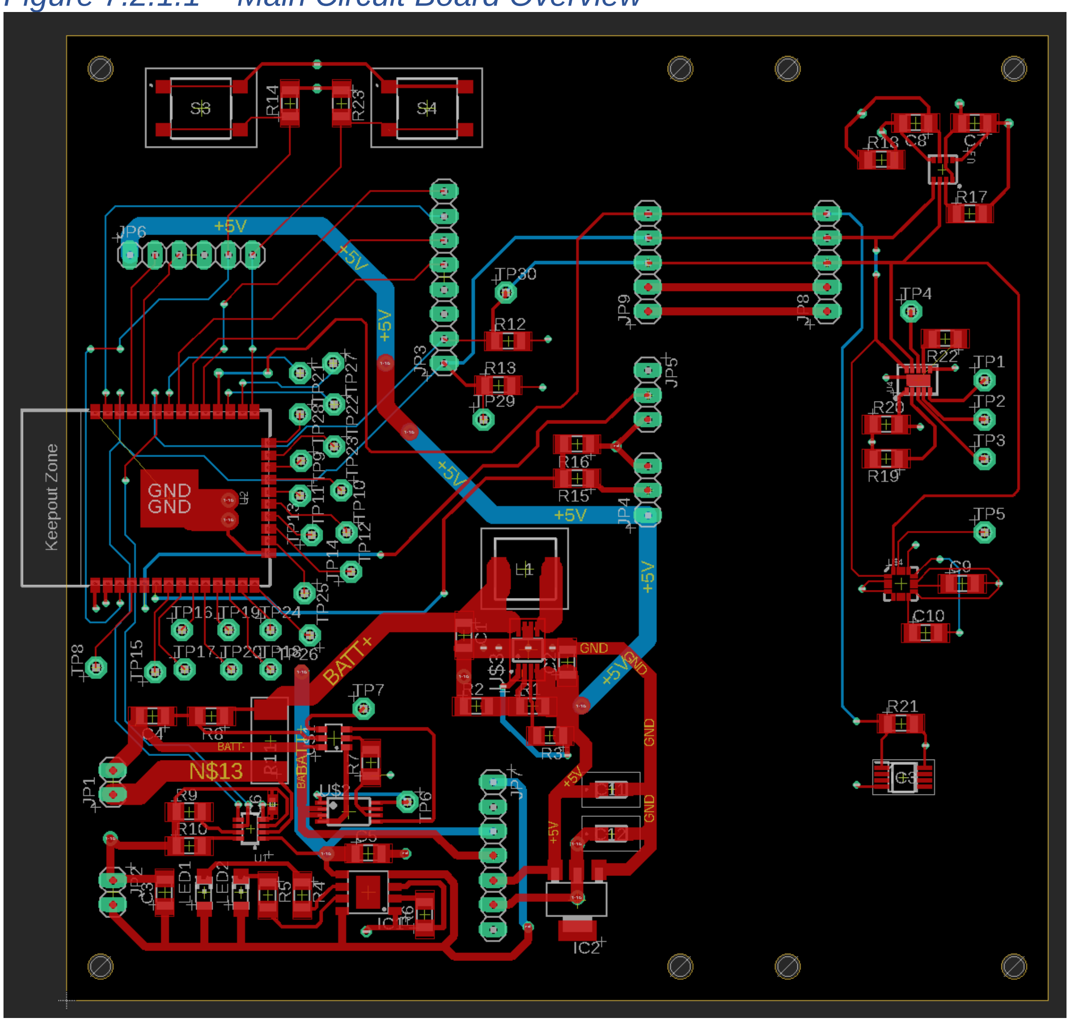
7.2 – Printed Circuit Board (PCB) Layout

The printed circuit board layout was the physical location and connection of each component in the electrical design. The layout had a direct impact on the physical mechanical design of our system. We needed to consider the size of the finished circuit board and the location of the components. We also needed to consider any manufacturer recommendations for the physical layout of the parts.

7.2.1 – Main System PCB Layout

The main system PCB contained the microprocessor, power systems, and all sensors excluding wind sensors. The main PCB also housed all supporting minor components. We designed the main system PCB layout with prototyping and testing in mind. Additionally, we elected to use a 100mm square size to reduce the cost. We found that limiting the size and choosing a typical size reduced the cost of the PCB from the vendors.

Figure 7.2.1.1 – Main Circuit Board Overview



The main PCB was laid out to locate related parts near one another. We also carefully considered the physical layout of the 5-volt switching power supply. The layout was important for switching power supplies for three main reasons. First, the efficiency was below the manufacturers rated capability. Second, the switching power supply was introduced noise into the power system. Noise on the power system had caused issues with other components that can be hard to detect. Third, the regulator had unexpected behavior under high loads. We likely would not run into the third issue as; the supply was capable of approximately four times the expected current draw of the system. The manufacturer recommended a layout design in the datasheet for the part. We matched the electrical characteristics of the supporting components. We also matched the layout as closely as possible given the different footprints of our parts.

Regarding power, we had chosen to create a four-layer board. The top and bottom layers contained all the traces. The middle two layers were 3.3 volts and ground. We chose 3.3 volts as opposed to 5 volts for this layer because nearly all the components ran on 3.3 volts. With these power and ground planes, we accessed either one from nearly anywhere on the PCB. This access was provided by small holes called vias. Eagle allowed us to create a via and name it to the same net name as either power or ground. The connection to that layer was then made for us and we connected our component to the via using a trace. We also recognized the benefits of noise reduction and shielding provided by the ground layer. A ground layer had a much lower impedance than a ground trace for each device. We also considered creating our ground pour on the bottom of the board. We ultimately decided against this as the noise reduction benefits were reduced as the traces on the bottom layer divided the ground plane. A final benefit to this design was the heat dissipation capability of these additional layers. By connecting devices to either continuous layer, the copper of the layer had spread the heat evenly across the PCB. This was beneficial to our power circuitry as well as the microcontroller itself. An important consideration regarding the heat dissipation was the effect on the sensors. In testing, the BME280 was very sensitive to waste heat from the other components.

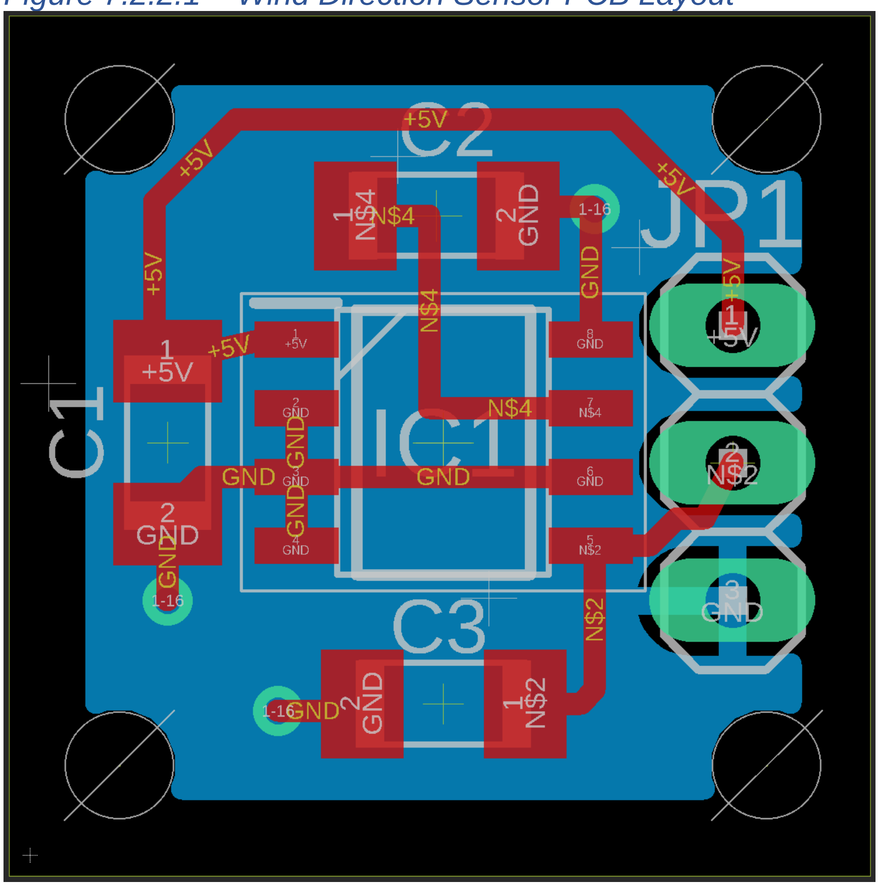
This consideration lead us to the next feature of our main PCB layout design, the division between the sensors and the rest of the circuitry. In Figure 7.2.1.1, we had observed some of the design considerations we made to support prototyping and testing. The first feature was the addition of two extra mounting holes on both the top and bottom center of the board. The second feature not pictured was the division of layers between these two sets of centers mounting holes. There was no copper running through this area of the board. The only connection between halves was through the two five pin headers near the top center of the board. These design choices had two main explanations. First, the separation of ground and power layers had isolated any waste heat from interfering with the sensors. Second, should we need to move to further isolate them from the main circuitry, we would have cut the circuit bord between the holes. We also needed to utilize this feature either for mechanical design of the node or isolation from heat of the circuitry. We had left this as an option to allow for quick modification during our testing. Also, the cost was lower for one larger board than two smaller boards. Lastly, this form of design rework would save us time and money with minimal upfront impact. If we needed to move the placement of the sensors otherwise, we would need to reorder a new PCB and many of the components. This was a potential cost that our self-funded budget would not support.

The main PCB has all the circuitry required for programming the microcontroller on board. A programming device was connected to the J6 header which exposed the transmit, receive, boot selection, and enable pins of the microcontroller. Additionally, the programming header allowed the device to be temporarily powered from the programming device with a 5-volt input and ground lines. We also included buttons for the enable and boot selection pins. The buttons allowed us to manually configure the microcontroller for accepting new programs should the need arise. The enable button was also used as a reset button to reboot the microcontroller during troubleshooting.

7.2.2 – Wind Direction Sensor PCB Layout

The wind direction sensor was a separate PCB from the main PCB. The first reason for this was the wind direction sensors used a strong magnet. During our testing, we found that the magnetic field from this magnet interfered with the accuracy of the compass. We were able to mitigate this interference by moving the magnet approximately one foot away from the compass. The second reason was to support a mechanical design. Both the wind speed and direction sensors required significant mechanical design. We may not reach the mechanical portion of the design within the scope of this project. In the event we could design a mechanical solution for these sensors, having them as satellite modules greatly reduced the complexity of the design.

Figure 7.2.2.1 – Wind Direction Sensor PCB Layout



The physical design of the wind direction sensor PCB focused on a compact size. The compact size allowed us to embed the device in a mechanical apparatus. We were able to create a board with a 0.65-inch square size. One of the methods we used to reduce the size was locating any parts possible on the main PCB. For example, the voltage divider consisting of two resistors was placed near the connector on the main PCB. The wind speed sensor was a two-layer board. The bottom layer was an uninterrupted ground plane. The top layer housed all the traces. This simple small design was low cost and fit the requirements of our system.

8.0 – Administrative

8.1 – Project Budget and Funding

The budget and the items that we used for our project and the estimated price were shown here on Table 8.1.1.1 - Cost Estimate. We had a budget of around $400-500. According to the table, the estimated total price would be $260.67, which satisfied the budget of $400-500. However, due to certain constraints we had to change the estimated price because we bought more materials, and their prices were different. We expected the total estimated price of all the items to be higher, most likely around $400 due to the vendors and because the shipping price of some products depending on the area was more expensive.

The table below listed all the items that we bought for our design and listed how many nodes of each item we bought and the price of each node. It also listed the cost of the total amount of nodes from each item that we used. As we researched further in our project and tested some materials, we noticed that some products were replaced or modified and that some costs were changed. When we researched the product that we bought, we searched from many different types of vendors and knew the exact product number. We also noticed which products were currently in stock and the lead time. These types of information were presented in the last two table of the Bill of Materials below.

When we bought the materials for the design, we did it by buying them from online companies like Amazon due to its availability and price. Also, most of the material that we used as the central part of our design such as the ESP32 microcontrollers were manufactured by ESP. Also, HiLetgo was another mostly used source for other nodes in our design such as other MCU's, atmospheric pressure sensors, humidity sensors, and camera modules. Also, to test the solar charging, solar panels and batteries were bought as well.

8.2 – Self Funding Cost Constraints

Some of the constraints that we had while we bought the products for our design included cost. We made sure that the total price of the materials that we bought doesn't exceed over $500. This influenced the types of products that we ended up buying because depending how expensive they are, it could have affected the quality.

8.3 – Bill of Materials

Table 8.3.1 – Bill of Materials with Manufacturer and Part Number

Part Number	Part Name	Manufacturer	Mfg. Part #	Quantity	Unit Cost
001	MCU-ESP32	Espressif Systems	ESP32-WROOM-32-N4	3	$8.00
002	BME280 - Temperature/ Humidity/ Pressure	Bosch Sensortec	BME280	3	$9.01
003	DS18B20 Digital Temperature Sensor	SUNFOUNDER	DS18B20	3	$12.99
004	CCS811 Air Quality Sensor	Digilent, Inc.	CCS811	3	$24.99
005	Current Sensor – INA219A	Texas Instruments	INA219A	3	$2.63
006	Solar Charge Controller – TP4056	JMoon Technologies	TP4056	3	$30.00
007	Battery Protection – DW01A	Unbranded	DW01A	3	$7.99
008	IC Part for Eval Board	Melexis	MLX90316KDC-BCG-300-SP	1	$6.71
009	Solar Panel	Solar Panel	6v 500mA PV	2	$15.99
010	Battery	Miisso	10000mAh	2	$16.99
011	3.3V Voltage Regulator	Texas Instruments	UA78M33CKVURG3	1	$0.92
012	5V Voltage Regulator	BINZET	B00J3MHRNO	1	$9.98
013	ArduCam Mini 2 MP	Arducam	OV2640	2	$8.99
014	Hall effect sensor	Module	OH3144	1	$1.62
Total					$156.81

Table 8.3.1 shows the bill of materials for the major components. This table gave us a good estimate of the cost of the project. For a single node, we expected a cost of approximately $150. This meant to build our goal of 3 nodes, the cost was $450 which is reasonably close to our agreed budget.

Table 8.3.2 – Component Availability

Part Number	Part Name	Vendor	Supplier Part #	Current Stock	Lead Time
001	MCU-ESP32	Espressif Systems	165-ESP32-WROOM-32-N4CT-ND	0	8 weeks
002	BME280 - Temperature/ Humidity / Pressure	DigiKey	828-1063-6-ND	220	In stock
003	DS18B20 Digital Temperature Sensor	Mechanic Surplus	DS18B20	In stock	In stock
004	CCS811 Air Quality Sensor	DigiKey	1286-1233-ND	2	9 weeks
005	Current Sensor – INA219A	Texas Instruments	296-23978-6-ND	0	35 weeks
006	Solar Charge Controller – TP4056	Roborium	JMTP40561	4	3-5 days
007	Battery Protection – DW01A	Ebay	DW01A DW01 SOT-23-6	2	Approx 5 weeks
008	IC Part for Eval Board	Mouser Electronics	482-90316KDC BCG300SP	80	7 weeks
009	Solar Panel	Ebay	CNC145x145-6	5	3 weeks
010	Battery	Amazon	B07H87HKKM	In Stock	2 days
011	3.3V Voltage Regulator	Newark	29AH8357	6,721	2-4 days
012	5V Voltage Regulator	Amazon	LTC0389-X	(In stock)	2 days
013	ArduCam Mini 2 MP	Amazon	M0023	(In stock)	Approx 3 days
014	Hall effect sensor	Module	OH3144	10	3 weeks

The electrical design was completed with component availability in mind. Table 8.3.2 shows the expected lead times of each of the major components. We also attempted to select components available from multiple vendors in the event a vendor runs out of stock of a component.

8.4 Milestones

Table 8.4.1 – Project Milestones

Senior Design Timeline	
Senior Design 1	
Date	**Details**
1/10/22	Project Brainstroming, and forming ideas with Computer Engineering group
1/11/22	Group formed
1/12/22	Decided project idea
2/3/22	Divide & conquer v1.0. Complete the D&C
2/4/22	Divide & conquer v1.0. Complete the D&C. Submit the completed D&C
2/9/22	Discuss the D&C with Dr. Richie
2/17/22	Updated Divide & Conquer Draft ready
2/18/22	Divide & Conquer v2.0 Due
2/21/22	Order parts for basic prototype
3/1/22	Assemble and test basic prototype
3/11/22	New Assignment on Standards
3/22/22	60-page draft final draft
3/25/22	**60 Page Paper due**
4/7/22	100 Page Paper final draft
4/8/22	**100 Page Report submission**
4/23/22	Final Documents draft
4/26/22	**Final Documents due**
4/29/22	Start assessing parts and order
Senior Design 2	
5/16/22	Classes begin
5/17/22	Plan May goals - Build and test components
6/20/22	June - Execute project and test software components
7/25/22	July – Finalized Prototype & Documentation
7/28/22	**Final presentations**
8/5/22	**Final day of class and submit document**

We as a group discussed the major components that were needed to be met by a particular date. Table 8.3.1 displayed all the dates that showcases everything we planned from brainstorming and writing our first draft to production. The dates were divided by two semesters where the first semester contained the planning and drafting the product in documentation form from 1/10/22 to 4/26/22. The second semester began from 5/16/22 and concluded on 8/5/22 which comprised of development and production.

References

[1] Management of National Park Service Programs. National Park Service, 2006, https://www.nps.gov/subjects/policy/upload/MP_2006.pdf.

[2] "National Fire Danger Rating System." Cibola National Forest and National Grasslands - Resource Management, https://www.fs.usda.gov/detail/cibola/landmanagement/resourcemanagement/?cid=stelprdb5368839

[3] Schlobohm, Paul, and Jim Brain. Gaining an Understanding of the National Fire ... - NWCG. NWCG Fire Danger Working Team, July 2002, https://www.nwcg.gov/sites/default/files/products/pms932.pdf.

[4] Cohen, Jack D., and John E. Deeming. United States Agriculture the National Fire ... - Fs.fed.us. May 1985, https://www.fs.fed.us/psw/publications/documents/psw_gtr082/psw_gtr082.pdf

[5] "Understanding Fire Danger (U.S. National Park Service)." National Parks Service, U.S. Department of the Interior, 2021, https://www.nps.gov/articles/understanding-fire-danger.htm

[6] "IPC Standards for Printed Circuit Boards: PCB Design." Mcl, Millenium Circuits Limited, 10 Mar. 2022, https://www.mclpcb.com/blog/ipc-standards-for-pcbs/#:~:text=IPC%2D2221%20is%20the%20standard,rigid%20and%20MCM%2DL%20PCBs

[7] van Rossum, Guido, et al. "Python Enhancement Proposals." PEP 8 – Style Guide for Python Code, 1 Aug. 2013, https://peps.python.org/pep-0008/#:~:text=The%20Python%20standard%20library%20is,inside%20parentheses%2C%20brackets%20and%20braces.

[8] Smith, Diane M. Sustainability and Wildland Fire The Origins of Forest Service Wildland Fire Research. USDA, May 2017, https://www.fs.usda.gov/sites/default/files/fs_media/fs_document/sustainability-wildlandfire-508.pdf.

[9] National Fire Incident Reporting System Complete Reference ... FEMA, Jan. 2015, https://www.usfa.fema.gov/downloads/pdf/nfirs/nfirs_complete_reference_guide_2015.pdf.

[10] "Anemometer." University of Technology Sydney, 11 Aug. 2017, https://www.uts.edu.au/partners-and-community/initiatives/after-da-vinci/models/anemometer#:~:text=The%20four%2Dcup%20velocity%20anemometer,at%20Armagh%20Observatory%2C%20Northern%20Ireland.

[11] Ouellette, Veronica. "Types of Anemometer." Sciencing, 2 Mar. 2019, https://sciencing.com/units-anemometer-measure-8146408.html.

[12] Max, et al. “The Complete Library of Types of Anemometer - Industrial Manufacturing Blog.” Linquip, 20 Aug. 2021, https://www.linquip.com/blog/the-complete-library-of-types-of-anemometer/.

[13] National Geographic Society. “Anemometer Functionality.” National Geographic Society, 9 Oct. 2012, https://www.nationalgeographic.org/encyclopedia/anemometer/#:~:text=The%20arms%20are%20attached%20to,used%20to%20calculate%20wind%20speed.

[14] Beach, Emily. “Differences between a Wind Vane and an Anemometer.” Sciencing, 2 Mar. 2019, https://sciencing.com/differences-between-wind-vane-anemometer-4801.html.

[15] “Physics Tutorial: The Doppler Effect.” The Physics Classroom, https://www.physicsclassroom.com/class/waves/Lesson-3/The-Doppler-Effect.

[16] Electrical, Wat. “Hot Wire Anemometer : Definition, Equation, Working, Uses and Benefits.” WatElectrical.com, 29 Nov. 2021, https://www.watelectrical.com/what-is-hot-wire-anemometer-construction-its-working/.

[17] MLX90316 EVB-DMB REV01 - Melexis. https://www.melexis.com/-/media/files/documents/datasheets/evb90316-datasheet-melexis.pdf.

[18] Choifin, Mochamad, et al. “A Study of Renewable Energy and Solar Panel Literature through Bibliometric Positioning during Three Decades.” DigitalCommons@University of Nebraska - Lincoln, 21 July 2021, https://digitalcommons.unl.edu/libphilprac/5749/

[19] Dryad. “Silvanet: Border Gateway.” Dryad, https://www.dryad.net/bordergateway

[20] Dryad. “Silvanet: Border Gateway.” Dryad, https://www.dryad.net/bordergateway

[21] Dryad. “Silvanet: Ultra Early Wildfire Detection.” Dryad, https://www.dryad.net/silvanet

[22] Dryad. “Silvanet: Wildfire Sensor.” Dryad, https://www.dryad.net/wildfiresensor

[23] Dryad. Uplink, https://uplink.weforum.org/uplink/s/uplink-contribution/a012o00001pTfQzAAK/dryad-silvanet

[24] “Fevertir: Fever Detection System: Body Temperature Measurement.” VisionTIR, 14 Dec. 2020, https://visiontir.com/products/fevertir-fever-detection-system/

[25] Freris, Leon. “Renewable Energy in Power Systems by Leon Freris, David Infield - Books on Google Play.” Google, Google, https://play.google.com/store/books/details/Renewable_Energy_in_Power_Systems?id=0VLnK_kTVUUC&gl=US

[26] Huot, Fantine, et al. “Deep Learning Models for Predicting Wildfires from Historical Remote-Sensing Data.” Google Research, 1 Jan. 1970, https://research.google/pubs/pub49935/

[27] Journal, IJRTE. “An Efficient and Effective Forest Surveillance System to Prevent Malicious Activities Using LoRa.” International Journal of Recent Technology and Engineering (IJRTE), IJRTE Journal Https://Www.ijrte.org/Wp-Content/Uploads/2021/02/IJRTE-Logo-W-300x252.Png, 29 Oct. 2020, https://www.ijrte.org/download/volume-8-issue-4/

[28] Kim, Hee-Je. “Solar Power and Energy Storage Systems.” Amazon, Pan Stanford Publishing, 2019, https://www.amazon.com/Solar-Power-Energy-Storage-Systems/dp/9814800139

[29] “Multispectral Vision Technology.” VisionTIR, 24 Dec. 2021, https://visiontir.com/

[30] Smets, Arno, et al. “Solar Energy: The Physics and Engineering of Photovoltaic Conversion, Technologies and Systems by Arno Smets.” Amazon, Amazon, 1 Apr. 2016, https://www.amazon.com/gp/product/B0198VHPHM/ref=dbs_a_def_rwt_hsch_vapi_tkin_p1_i0

[31] “Arduino Uno Pinout, Specs, Layout, Schematic in Detail(Updated).” ETechnophiles, 26 Feb. 2022, https://www.etechnophiles.com/arduino-uno-pinout-pin-diagram-specifications/.

[32] Kools, Pieter, et al. The Arduino Ide, https://se.ewi.tudelft.nl/desosa2019/chapters/arduino-ide/.

[33] Edmallon, et al. “Hacking a Capacitive Soil Moisture Sensor (V1.2) for Frequency Output.” Underwater Arduino Data Loggers, 3 Apr. 2022, https://thecavepearlproject.org/2020/10/27/hacking-a-capacitive-soil-moisture-sensor-for-frequency-output/.

[34] “PEAK Detector: Diodes and Rectifiers: Electronics Textbook.” All About Circuits, https://www.allaboutcircuits.com/textbook/semiconductors/chpt-3/peak-detector/.

[35] “555 Timer Tutorial - the Monostable Multivibrator.” Basic Electronics Tutorials, 23 Apr. 2020, https://www.electronics-tutorials.ws/waveforms/555_timer.html.

[36] ago, Shawn 2 years, et al. “Soil Moisture Sensor - Getting Started with Arduino.” Latest Open Tech From Seeed, 17 Mar. 2021, https://www.seeedstudio.com/blog/2020/01/10/what-is-soil-moisture-sensor-and-simple-arduino-tutorial-to-get-started/.

[37] “Capacitance (Charge).” SoilSensor.com, 25 Sept. 2018, https://soilsensor.com/articles/capacitance-charge/#:~:text=When%20you%20add%20water%20to,determine%20the%20soil%20moisture%20content.

[38] “Indoor Environmental Quality: A How-To Guide.” Worldgbc, https://www.worldgbc.org/sites/default/files/bp-resource/BPFP-IEQ-Guidance-Note.pdf.

[39] CCS811 - Sciosense.com. https://www.sciosense.com/wp-content/uploads/documents/SC-001232-DS-3-CCS811B-Datasheet-Revision-2.pdf.

[40] Karvinen, Kimmo, and Tero Karvinen. Getting Started with Sensors: Measure the World with Electronics, Arduino, and Raspberry Pi. Maker Media, 2014.

[41] Sathish, Janani. LEARN ESP32 ARDUINO INTERFACING - A STEP BY STEP GUIDE.

[42] Webster, John G., and Halit Eren. "8-Single-Chip Chemical and Biological Sensors." Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement, 1st ed., vol. 1, ser. 1, CRC Press, Boca Raton, FL, 2018, pp. 155–172. 1.

[43] Ziemann, Volker. A Hands-on Course in Sensors Using the Arduino and Raspberry Pi. Taylor Et Francis, CRC Press, 2018.

[44] "FDM vs. SLA: Compare Filament and Resin 3D Printers." Formlabs, https://formlabs.com/blog/fdm-vs-sla-compare-types-of-3d-printers/.

[45] "What Is Pla? – 3D Printing Materials Simply Explained." All3DP, 1 May 2019, https://all3dp.com/2/what-is-pla-3d-printing-materials-simply-explained.

[46] "What Is Pla? – 3D Printing Materials Simply Explained." *All3DP*, 1 May 2019, https://all3dp.com/2/what-is-pla-3d-printing-materials-simply-explained/.

[47] "What Is Pla? – 3D Printing Materials Simply Explained." *All3DP*, 1 May 2019, https://all3dp.com/2/what-is-pla-3d-printing-materials-simply-explained/.

[48] Slack, Ian. "Soldering Safety." *Department of Engineering Health & Safety*, 27 Mar. 2018, https://safety.eng.cam.ac.uk/safe-working/copy_of_soldering-safety.

[49] "RoHS Guide in Electronics: RoHS, WEEE and Lead-Free FAQ." *Electronics Tutorial* | *The Best Electronics Tutorial Website*, 14 Nov. 2020, http://www.electronicsandyou.com/blog/rohs-guide-in-electronics-rohs-weee-and-lead-free-faq.html.

[50] Sentryair. "Comparing Solder Fume Extractors: It's All about the Filter." *Sentry Air Systems, Inc.*, 27 July 2020, https://www.sentryair.com/blog/industry-applications/soldering/comparing-solder-fume-extractors-its-all-about-the-filter/.

[51] Admin. "Pet & Petg Plastic: 6 Facts You Need to Know." *Indepak*, 10 Aug. 2020, https://www.indepak.com/pet-petg-plastic-6-facts-need-know/#:~:text=PET%20%26%20PETG%20Plastic%20Is%20Easy%20To%20Recycle&text=PET%20and%20PETG%2C%20which%20is,option%20for%20eco%2Dfriendly%20packaging.

[52] Heinecke, Robert. "How We Calculate Our Air Quality Index and Why We Need It." Breeze Technologies, Robert Heinecke Https://Www.breeze-Technologies.de/Wp-Content/Uploads/2017/04/Breeze-Logo-Header.png, 28 Sept. 2021, https://www.breeze-technologies.de/blog/what-is-an-air-quality-index-how-is-it-calculated/

[53] “Volatile Organic Compound.” Wikipedia, Wikimedia Foundation, 31 Mar. 2022, https://en.wikipedia.org/wiki/Volatile_organic_compound

[54] Smith, Dianna. “What's in Wildfire Smoke? A Breakdown of the Air Pollutants Created by Wildfires.” Kaiterra, https://learn.kaiterra.com/en/air-academy/whats-in-wildfire-smoke

[55] Smith, Dianna. “Understanding TVOC: What You Need to Know about Volatile Organic Compounds.” Kaiterra, https://learn.kaiterra.com/en/resources/understanding-tvoc-volatile-organic-compounds

[56] “How Do CO2 Emissions from Forest Fires Compare to Those from Fossil Fuels? | CBC Radio.” CBCnews, CBC/Radio Canada, 14 Sept. 2018, https://www.cbc.ca/radio/quirks/sept-15-2018-summer-science-camping-under-a-volcano-plastic-in-beluga-bellies-and-more-1.4821942/how-do-co2-emissions-from-forest-fires-compare-to-those-from-fossil-fuels-1.4821944

[57] A;, Graf M;Frey U;Taschini S;Hierlemann. “Micro Hot Plate-Based Sensor Array System for the Detection of Environmentally Relevant Gases.” Analytical Chemistry, U.S. National Library of Medicine, https://pubmed.ncbi.nlm.nih.gov/17007499/

[58] “Carbon Dioxide (CO2).” Carbon Dioxide (CO2): Environmental Health in Minnesota, https://www.health.state.mn.us/communities/environment/air/toxins/co2.html

[59] “Preventing Fire and/or Explosion Injury from Small Devices and Wearable Lithium Battery Powered Devices“

https://www.osha.gov/sites/default/files/publications/shib011819.pdf

[60] “Battery Types and Safety” https://batterysolutions.com/learning-center/battery-types/

[61] “Nickel Metal Hydride Battery” https://www.sciencedirect.com/topics/chemistry/nickel-metal-hydride-battery

[62] “Department of Environment & Conservation: NiMH and NiCD batteries”

https://www.sciencedirect.com/topics/chemistry/nickel-metal-hydride-battery

[63] “Hydrogen Tools” https://h2tools.org/lessons/nickel-cadmium-battery-explosion

[64] “Charge Controller | Charge and Battery Regulators | altE.” *altEstore.Com*, www.altestore.com/store/charge-controllers-c432. 21 Apr. 2022.

[65] “Voltage Regulators Types and Working Principles.” *MPS*, 2022, **Error! Hyperlink reference not valid.**.