Distributed Sound Event Location - DiSEL

University of Central Florida
Department of Electrical & Computer Engineering

EEL4915 - Senior Design 2
Dr. Samuel Richie & Dr. Lei Wei

Group D

12/8/20

Brandon LaGuerre - Electrical Engineering

Mark Judy - Computer Engineering

Christopher Santana - Electrical Engineering

Drew King - Computer Engineering

Table of Contents

Contents

Table of Figures

1 Executive Summary

Sound plays a major role in our world, from being used for communication by humans and animals, to echolocation used by dolphins and other animals to hunt in low visibility conditions, to the distant sound of thunder warning of an incoming storm. From a technical standpoint, sound is a very powerful tool which can be utilized in many unique and valuable ways. Advancements in modern technology, and a large amount of research in audio signal processing, have paved the way for many technologies that incorporate audio broadcasting, audio synthesis, 3D sound localization, noise cancellation, audio fingerprinting, sound recognition, and much more.

This project will focus on implementing two specific applications of audio signal processing: 3D sound localization, audio fingerprinting, and sound recognition.

DiSELs purpose is to detect shared sounds at different locations, calculate the time of arrival at each location, determine a possible origin of the sound (category and location), and then serve this data through a web application. This technology would be designed to be modular to have use in a variety of applications ranging from safety and security, hunting, research, and conservation of animal species of interest, or even general hobbyist activities such as birdwatching.

This technology would be extremely useful for law enforcement in active shooter situations, by connecting their smart devices to our DiSEL sound network installed at that location, personnel will be able to receive valuable tactical information on the shooter's firearm and current whereabouts so that they can make an informed plan-of-action. In addition, civilians at the location would also be able to connect to the network and be given information that will aid them in making decisions that would increase their safety.

The DiSEL network will be expandable in order to provide optimal coverage in a variety of different scenarios (i.e. using a low number of devices in a small, enclosed area or a higher number of devices in a larger outdoor area).

In outdoor applications where it would be difficult to maintain the battery levels of each device in the DiSEL network, the devices will use solar power in order to maintain operable power for extended periods of time.

2 Project Summary

The following section will go on to discuss our group's goals for this project, along with the required specifications, requirements, and the distribution of subsystems between the members.

2.1 Goals and Objectives

The project will satisfy the following goals and/or objectives shown in Table 2-1.

An array of multi-directional audio sensors to detect multiple categories of sound events. Connected to both solar and battery power for indoor and outdoor use.
An application to allow users to interact with data that is retrieved and analyzed from the devices.
A microcontroller connected to the sensors to analyze sound events.

Table 2-1*: Goals and Objectives*

2.2 Specifications

The project will satisfy the following specifications shown in Table 2-2.

The array of sensors is connected to a central server.
The central server analyzes the sound events from multiple sensors. It will determine if all sounds being analyzed are of the same origin by filtering out other noise.
The sensors will timestamp all audio that is transmitted to the central server.
The central server triangulates the location of the sound event using time of arrival.
The central server is supported by machine learning and a database of sound samples consisting of multiple categories including discharging of different firearms, weather such as thunder, and bird calls.
The program will adjust for environmental factors such as temperature that may affect the distance a sound might travel when determining the originating location of a sound event.
Sensors will be mountable to pre-existing light or utility poles for outdoor applications, and wall or ceiling mountable for indoor applications.
Sensors will be able to withstand various environmental conditions for outdoor use.

Table 2-2*: Specifications*

2.3 Requirements

The project will satisfy the following requirements shown in Table 2-3.

Identify the originating location of the sound event within:	5	Meters
Verify the same sound is being analyzed from all sensors with greater than:	90%	Accuracy
Identify what originated the sound event with greater than:	90%	Accuracy
Analyze the location and label the sound event within	5	Seconds
Have a minimum required distance between sensors of less than:	5	Meters
Have a maximum operating distance between sensors of greater than:	100	Meters
Determine the temperature of environment within:	1	℃
Maintain a sensor weight of less than:	5	lbs.
Have a sound event capture range of greater than:	100	Meters
Achieve an IP rating for sensor equal or better to:	I45	IP
Maintain a cost of less than:	500	USD

***Table 2-3**: Requirements*

2.4 Quality of House Analysis

This section will show the physical house of quality diagram, along with a description of how to interpret the meaning of each section of the diagram. This information will be used to process our development of the prototype and its features into the context of the customers desires, and the engineering capabilities we have access to and can achieve to meet the needs of the customer.

2.4.1 Understanding House of Quality

Inside of a matrix there are 6 of both engineering requirements and customer requirements. These requirements are then compared with each other to determine and display the relationships between each of them. This provides a visual layout on how each of the requirements both engineering and customer may influence each other. This provides the ability to view possible issues that may occur if a change is required in future project renderings as one change may affect others causing more problems than solutions. From the diagram shown below there are color coded regions that show how strong of a relationship there is between each of the requirements. Red symbolizes a strong relationship between two requirements, meaning that pairing of requirements has a stronger bond correlation. Yellow symbolizes a moderate relationship between two requirements, meaning the relationship may have some effect on each other but is considered quite minimal in the grand scheme of things. Green is then used to symbolize a weak relationship between two requirements, this means that there is no relationship between the two requirements.

The top of the HOQ diagram, also called the “roof”, shows the potential conflicts between the engineering requirements that are being used to satisfy the customer requirements. A plus sign is used to symbolize a strong correlation meaning those two engineering requirements have a direct cause and effect on each other. A negative sign is used to symbolize a weak correlation meaning those two engineering requirements have a minimal to no cause and effect on each other.

With this diagram, it provides the ability to have the priorities of the project to align both with the customer’s wants and needs and the engineering requirements that will be used to meet those goals. It also allows for the group to be able to know what areas of the project need to have more attention paid towards it as everything begins to come together.

2.4.2 House of Quality Diagram

In order to help our team adequately manage the development and design process of DiSEL, we have adopted the House of Quality technique to aid our process. The House of Quality will be our primary tool for making decisions on budgeting our time and resources in order to produce the best product we can.

The five main functional requirements we identified for the House of Quality are:

- Time Spent Learning
- Product Weight
- Battery Life
- Product Size
- Sound Identification Accuracy
- Cost

These functional (engineering) requirements are used to satisfy the requirements outlined by our customer to have a product that is designed to have the most benefit for them. Those customer requirements are:

- Usability
- Accuracy
- Reliability

- Length of Use
- Ease of Use
- Cost

Target goal for each of these engineering requirements to accomplish satisfying the customer needs have also been established. For the time it takes for a user to learn how to correct use our product we are targeting a maximum of ten minutes. For the products weight we have established a target of a maximum of eight pounds. For battery life are target will be a minimum of sixteen hours. A target of less than ten inches in diameter has been established for our products size. A ninety-five percent sound identification accuracy will be targeted. And the total cost of the project has been given a target amount of 750 USD.

Figure 2-1 shows our groups determined customer requirements along with the technical requirement that will be used to satisfy those of the customer. A correlation matrix shown above the functional requirements is a representation of the correlation between each of the functional requirements and how one being improved/accomplished may influence another.

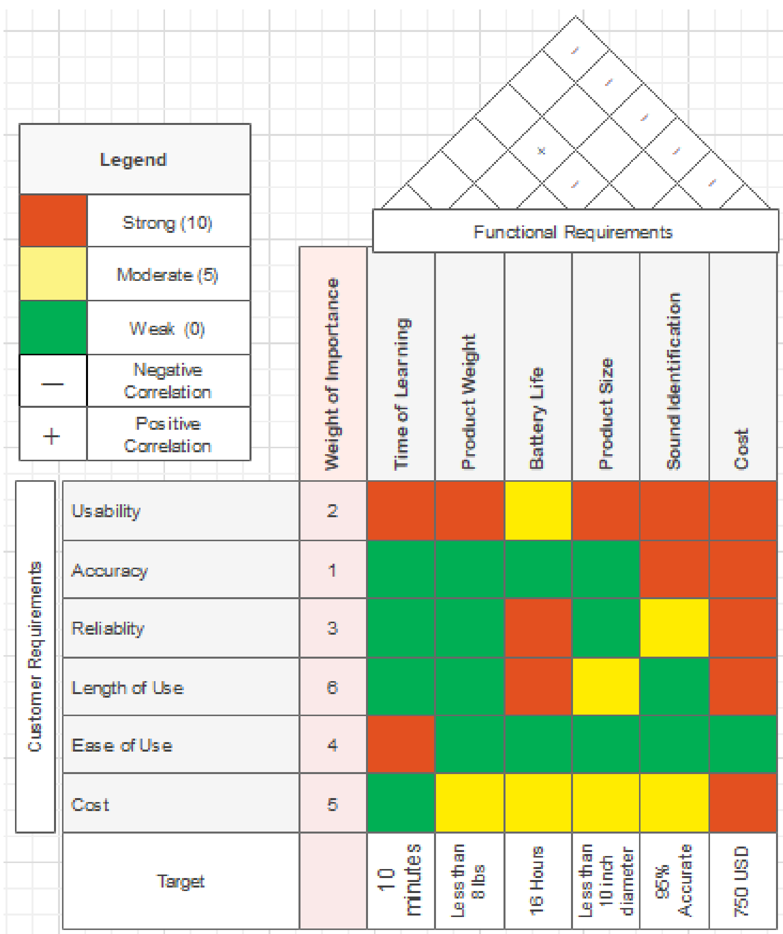


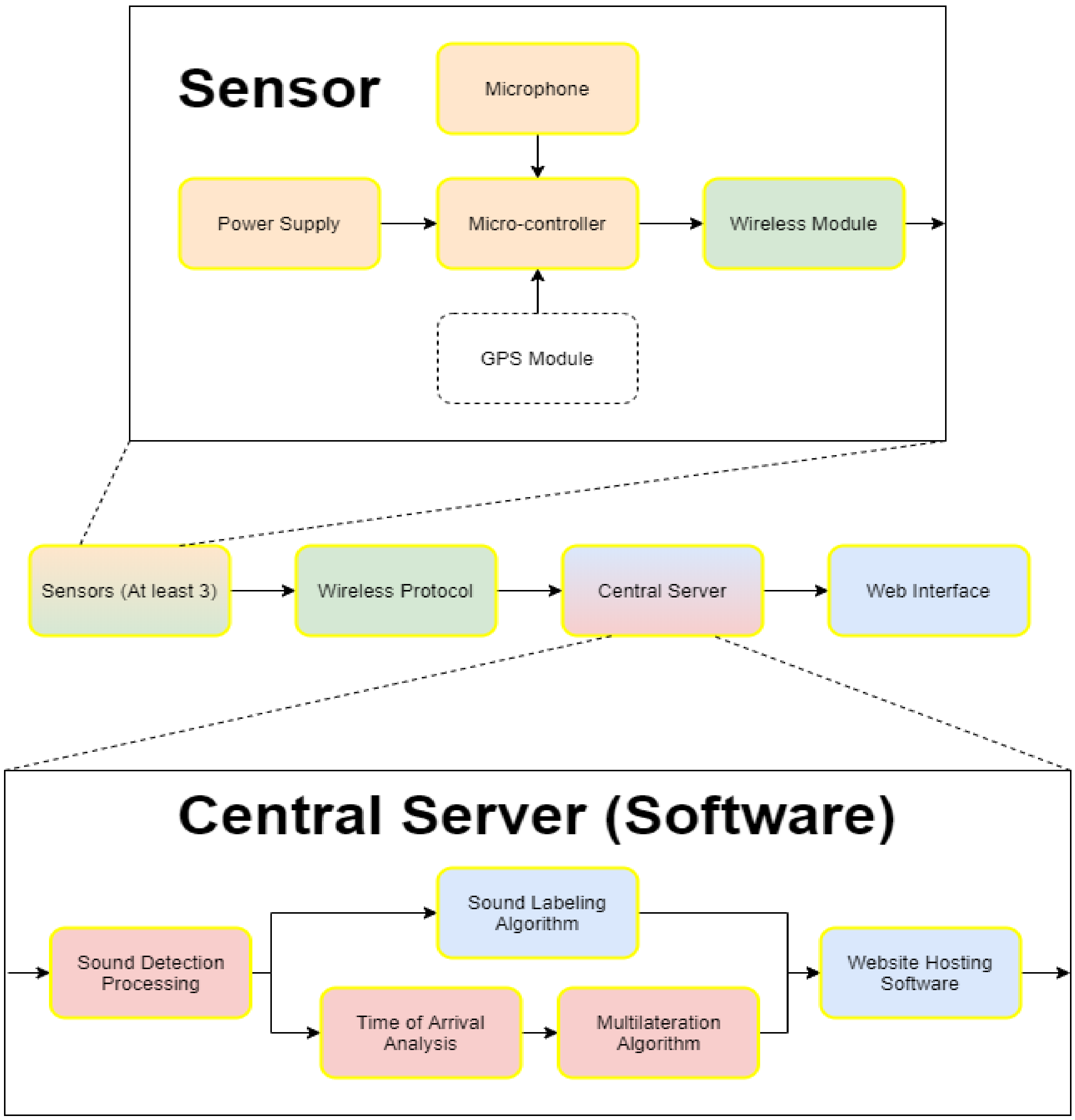
***Figure 2-1**: House of Quality Diagram*

2.5 Block Diagram

Table 2-4 gives the description of each block associated with the projects block diagram. Figure 2-2 provides the block diagram for our project showing the different sections of the prototype along with the division of responsibilities to each member in respect to each of the project's sections.

Block Name	Description
Power Supply	Provides power to the units of the system.
Sensors	Listens to the sounds being emitted from the surrounding environment.
Communication	Relays information and data between both the multiple units in the systems and the central server.
Central Server	A central hub that will connect the multiple system units and provide the ability to analyze all sounds acquired by the units.
Sound Location	Software that will determine the location of the object that is emitting the sound by analyzing data received from the system's sensors.
Sound Labeling	Software that will determine what object is emitting the sound by analyzing data received from the system's sensors.
Sound Detection	Software to determine if at least three sensors heard the same sound, and when each one did relative to the others.
User Interface (Web Application)	A web application formatted user interface that will allow a user to view data acquired by the system.

Table 2-4: *Block Descriptions*



Block Diagram Key	
Assigned Blocks	**Status**
Christopher Santana	Research
Brandon LaGuerre	Acquired
Mark Judy	Design
Drew King	Prototype
	Completed

Figure 2-2: *Block Diagram*

3 Research and Background Information

The following section will discuss research within other projects related to ours and the technologies and theories required to build our prototype to establish both application uses and the feasibility of creating a project like this.

3.1 Existing Projects

Products and projects that deal with sound identification, and using sound waves to locate specific targets, have been utilized in the defense industry for decades to increase the safety of military personnel and give them a tactical advantage on the battlefield. More recently, similar technologies have begun being implemented into the civilian world to alert police and civilians to active situations. Additionally, two previous senior design projects have attempted to design simpler, more affordable devices using these concepts.

3.1.1 Project GLASS

GLASS was a previous senior design project sponsored by Boeing and Sunstone Circuits that aimed to provide a modular security alarm system in the event of an active shooter situation. This project was intended to be used in the civilian world, with its primary function to automatically alert local law enforcement during an emergency.

For its primary function, the GLASS module continuously monitor's audio signals while checking for three specific events before it triggers its alarm: the event sound is above 140 dB, the peak frequency caused by the gunpowder after the firearm is discharged to determine the caliber of the firearm, and that there is proof of subsonic frequency as the round is fired to affirm that the sound event was not caused by a recording.

The secondary function of GLASS, was to attempt to triangulate the location of the firearm discharges and display the location of the event on a digital map of the institution that it is installed in.

Some additional interesting aspects of GLASS that are to be considered are its modular nature for being able to add extra features to the system in the future and its self-sustaining power design utilizing photovoltaic solar panels.

3.1.2 Project Minutemen

Minutemen was another senior design project that attempted to tackle the challenge of developing a system that can detect firearm discharges in an active shooter scenario, approximate the location of the discharge and attempt to alert local law enforcement about the situation.

The project's primary objectives when developing the Minutemen system were to make the detection/location system a very simple and affordable system for use in schools and other public places, while also having the option of being installed outdoors.

The Minutemen system originally was intended to use the results of multiple modules to triangulate the location of the sound event but ended up using multilateration techniques to estimate the origin of the sound event using only one module.

3.1.3 Guardian Indoor Active Shooter Detection System

Guardian is an indoor security system by Shooter Detection Systems that is designed to automatically identify gunfire and relay the location of the shooter to designated personnel utilizing a digital floor map of the facility.

The individual Guardian module has a small, inconspicuous design and is mounted flush with the wall or ceiling of the room that it is monitoring. It utilizes two sensor systems to identify whether a gunshot was fired. The two sensor systems used within the Guardian are an audio sensor that is used to record and identify the acoustic bang generated by the firearm discharge, and an infrared sensor that detects the infrared light from the muzzle flash.

The Guardian's most attractive feature for maximizing the safety of a particular buildings occupants in the event of an active shooter situation is the systems total processing time, being able to identify and send out an alert and approximate location to authorities within less than a second of the original gunshot.

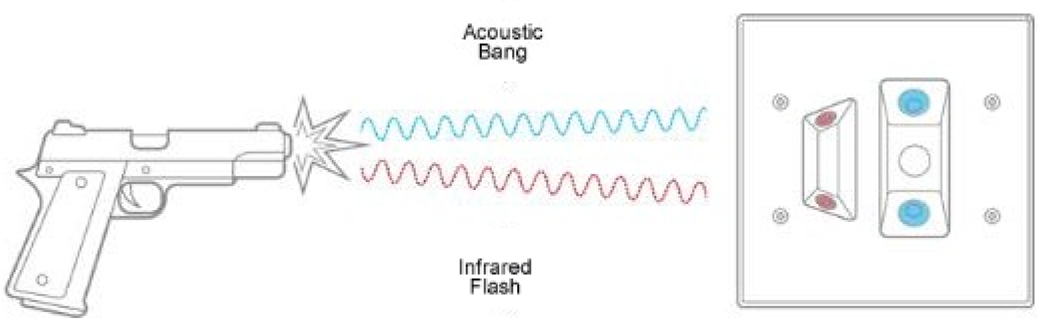


Figure 3-1: *Image of wall-mounted Guardian interacting with a physical effect of a firearm discharge*

3.1.4 Boomerang

Originally contracted by DARPA and developed by Raytheon's BBN Technologies, the Boomerang system can detect and locate the origin of small arms fire that come within 30 meters of the device. The system can be mounted to vehicles and operated regardless of whether the vehicle is moving or not. Boomerang can ignore sounds that are like gunshots such as firecrackers or door slams and does not trigger an alert when shots are fired from the vehicle it is mounted to or the site that it’s guarding.

The system operates by using an array of seven microphone sensors placed in different directions around the system. The alert is triggered by the sensors measuring the muzzle blast and supersonic shock wave that is produced by a supersonic bullet passing by. Due to the placement of the microphones on the array, each microphone will detect the sound event at slightly different times, this information is then used to determine the direction, elevation, and distance of the origin point of the gunshot.

The Boomerang utilizes both an LED screen and a speaker to inform the user of the location of the gunman. The system will announce the direction of the shot over its integrated speaker, and the LED display will show the direction of the gunman using a clock-like display and display the elevation, range and azimuth of the shot

Figure 3-2*: Image displaying the multi-directional microphone array of the Boomerang.*

3.1.5 Louroe Gunshot Detection

The Louroe Gunshot Detector is a dedicated gunshot detection software that can be integrated with other devices like microphones or a video surveillance system to allow those devices the capability of alerting a user to a gunshot. Developed as an add-on to other devices is a promising idea, however the greatest downfall of this device is that it is only compatible with other devices designed by Louroe which takes away from the novelty of being able to attach this software modularly.

An additional downside to Louroe's system is that it has a slow response time, requiring a few seconds to process the gunshot characteristics and alert the user.

3.1.6 ShotSpotter

Currently the most popular commercial gunshot locator on the market, ShotSpotter utilizes a series of sensors, advanced filtering algorithms and machine learning to accurately filter out ambient and non-ballistic noise while maintaining a timely and high accuracy detection response and location mapping.

This system has the capability of tracking whether there are multiple gunmen, whether the firearm is automatic, the number of rounds that have been fired, determining the exact latitude, longitude, and direction of the shots being fired.

The ShotSpotter technology has been utilized by law enforcement agencies in many cities to aid in reducing gun violence and helping to enact faster response times in the event of an emergency. The ShotSpotter was even a vital tool for the FBI and Franklin County Sheriff's department during the 2003 Ohio highway sniper attacks.

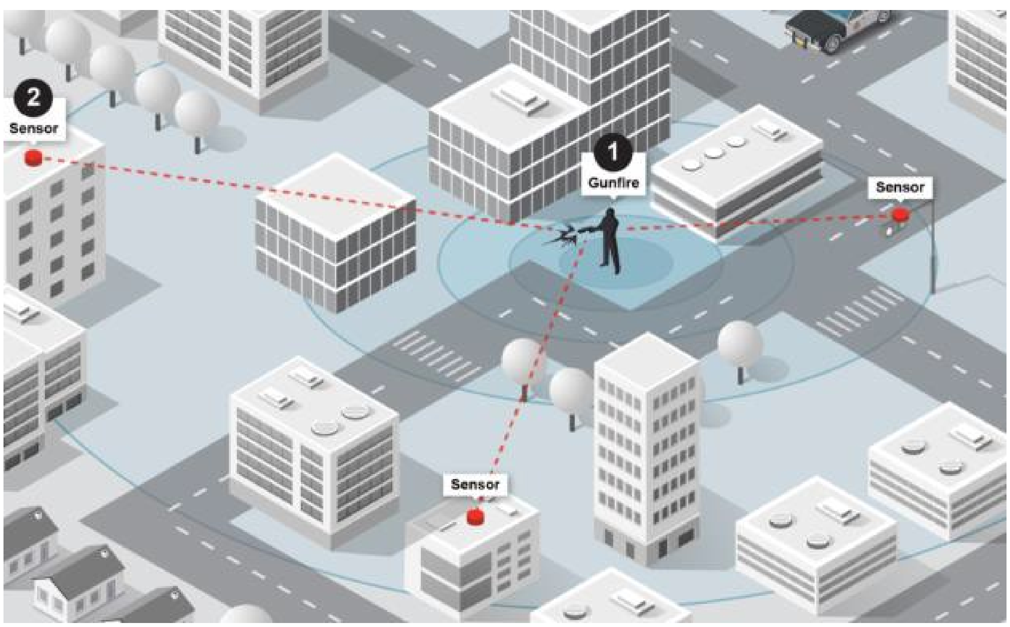


***Figure 3-3**: Example of how the ShotSpotter technology works.*

3.2 Technologies and Theory

The following section will be a discussion of research related to the technology and theory required for this project. A comparison of these technologies will be presented to determine which of these products will be used with our project.

3.2.1 Batteries

The following will discuss the different types of batteries that we are comparing for the use in our prototype. This includes: Lithium-Ion, Nickel-Metal-Hydride, and Lithium Iron Phosphate batteries.

3.2.1.1 Lithium-Ion

Lithium-ion batteries work by passing lithium ions from the anode through the electrolyte and to the cathode. Lithium-ion batteries have several advantages such as a high charge storage per unit mass and a high voltage of 3.7 volts. This means that they can deliver large amounts of current for high power applications. Lithium-ion batteries have no memory effect, meaning that repeated partial discharge of the battery will not reduce the overall capacity of the battery. In addition to this, lithium-ion batteries have a low self-discharge rate of about 1.5% per month. ["Lithium-Ion Battery"] So, when the batteries are not in use, they will hold their charge. Lithium-ion batteries are also very reliable. They require little maintenance and can withstand anywhere from 300 to 500 charge cycles. ["BU-808: How to Prolong"] The main disadvantage of lithium-ion batteries is its safety. The batteries are subject to overheating and sometimes combustion. Another disadvantage is that, like most batteries, they start to fail with age. ["Lithium-Ion Battery"]

3.2.1.2 Nickel-Metal-Hydride

Nickel-Metal-Hydride batteries were invented to be a more environmentally friendly alternative to the Nickel-Cadmium battery. Nickel-Metal-Hydride uses nickel hydroxide for the positive electrode, hydrogen for the negative electrode, and potassium hydroxide for the electrolyte. ["NiMH Battery Charging Basics"] The Nickel-Metal-Hydride battery has an output voltage of 1.2 volts and an energy density of 50 to 70 wh/kg. The advantage of Nickel-Metal-Hydride is that they offer a very similar voltage to disposable alkaline batteries. This means that any device that uses alkaline batteries can also use Nickel-Metal-Hydride batteries. Nickel-Metal-Hydride batteries can also be deep cycled. The disadvantage of the Nickel-Metal-Hydride battery is that it has a very high self-discharge of 20% to 50% within a month. ["Nickel Metal Hydride"]

3.2.1.1 Lithium Iron Phosphate

Lithium Iron Phosphate batteries use iron phosphate as the cathode material and graphite as the anode material. These batteries have a nominal voltage of 3.2 volts and an energy density of 90 to 120 wh/kg. The main advantage to Lithium Iron Phosphate batteries is that they are safe. They have excellent chemical and thermal stability. This means they can withstand high temperatures and still perform normally. Lithium Iron Phosphate batteries are not likely to experience thermal runaway and as a result, the phosphate cathode will not burn or explode during overcharging or overheating. The chemicals used in Lithium Iron Phosphate batteries are considered non-toxic and can be easily disposed of. Another advantage of Lithium Iron Phosphate batteries is their longevity. They can withstand anywhere from 1000 to 10,000 charge cycles. These batteries can also be stored for up to 350 days without losing significant charge. Lastly, Lithium Iron Phosphate batteries are inexpensive to manufacture. [Beck]

3.2.1.2 Battery Comparison

When comparing the three common rechargeable batteries, Lithium-Ion and Lithium Iron Phosphate have similar characteristics while Nickel-Metal-Hydride appears to be very different. The only reason to choose Nickel-Metal-Hydride over the other batteries, is that it has a voltage close to that of standard disposable batteries. This means that standard battery configurations can be used. However, Nickel-Metal-Hydride is not as reliable as its counterparts and has a low energy density. Lithium-Ion and Lithium Iron Phosphate both have desirable traits. They have high energy densities and have a low self-discharge rate. Lithium-Ion batteries have a higher energy density and voltage, while Lithium Iron Phosphate can withstand more charge cycles. Lithium-Ion batteries are known to overheat and sometimes combust. This is far less likely to happen with a Lithium Iron Phosphate Battery. Lithium Iron Phosphate is also better suited to operate in higher temperatures.

Considering all of this, either Lithium-Ion or Lithium Iron Phosphate would be an acceptable choice for a battery. Even though Lithium-Ion and Lithium Iron Phosphate both have advantages and disadvantages, neither comes out as a clear winner. Even though Lithium-Ion is volatile and has the chance of combusting and overheating, this rarely happens. If Lithium-Ion batteries were not safe, they would not be as common as they are today, and they would not be used in so many different electronics. Although Lithium Iron Phosphate is cheap to manufacture, Lithium-Ion batteries are mass produced to the point that the price difference between Lithium-Ion and Lithium Iron Phosphate is comparable. Overall, if the battery is to be used in an area with high temperatures a Lithium Iron Phosphate battery should be used. If the battery needed requires high energy density, then a Lithium-Ion battery should be used.

Comparison Factors	Lithium-Ion	Nickel-Metal-Hydride	Lithium-Iron-Phosphate
Voltages	3.7 V	1.2 V	3.2 V
Specific Energy	100 - 265 wh/kg	50 - 70 wh/kg	90-120 wh/kg
Charge Cycle Life	300 - 500 cycles	180-2000	1,000 – 10,000 cycles
Self-Discharge	1.5%/month	20%-50%/month	1.5%/month
Safety Comments	Known to potentially overheat	Non-toxic	Less likely to overheat than Lithium-Ion

***Table 3-1**: Battery Comparison*

3.2.1.3 Battery Management System

When batteries are going to be charged and discharged, a battery management system is recommended. A proper battery management system should perform three main functions: overcharge protection, over discharge protection, and maintaining charge balance amongst different cells. When charging a battery, it is dangerous to continue charging it over its operating voltage. A battery management system detects when the battery has been fully charged and uses a MOSFET to cut power to the battery. In the case of DiSEL, the battery pack will be charged with a solar panel. Without overcharge protection, if the battery pack is fully charged and the solar panel is in direct sunlight, then the battery pack is being overcharged and could be destroyed or possibly combust. The opposite is also true. If there is no sunlight and DiSEL is still consuming power, then the battery pack is at risk of over discharging. This could ruin the battery pack and can cause it to no longer hold a charge. [“Battery Management Systems”]

In order to get desirable voltages from batteries, they are often combined in series to form battery packs. Due to differences in age, construction, and imperfections, not all batteries are the same. This means that if all the batteries in a battery pack are charged at the same time with the same voltage and current, they all will not charge at the same rate. Charging all the batteries like this would eventually cause an imbalance in the battery pack. Some batteries could be charged at 75% while others are charged at 90%. If the batteries at 75% were charge to full capacity, the batteries at 90% would be overcharged. If overcharging did not happen, the battery pack would never be at full capacity. A battery management system balances the different cells in the battery pack by monitoring each cell and determining which cells need to be charged. [“Why Proper Cell Balancing”]

There are two types of cell balancing: active and passive. Passive cell balancing works by discharging cells through a dissipative bypass route. This is a very inexpensive method; however, all the energy is wasted away as heat. Active cell balancing uses capacitive or inductive charge shuttling to transfer charge from different cells. So, a battery with a full charge is used to charge a battery that is not at full charge. This is significantly more efficient than passive cell balancing because no energy is wasted. [“Why Proper Cell Balancing”]

3.2.2 Solar Energy

Solar panels are made up of different semiconducting materials that are combined to form a photovoltaic cell. A photovoltaic cell consists of an n-type and a p-type semiconductor that create a p-n junction. When light is shines on the p-n junction, the energy from a photon is transferred to an electron. This transfer of energy causes the electron to move from the valance band to the conduction band, creating electricity. This is known as the photovoltaic effect. Photovoltaic cells can only convert about 25% of sunlight into electricity. This is because sunlight consists of many photons, each with various energies. For an electron to jump from the valance band to the conduction band, it needs to receive a specific amount of energy known as the band gap energy. For silicone, the band gap energy is 1.2 electron volts. If a photon has energy less than 1.2 electron volts and it hits an electron, the electron will remain in the valence band and the energy will be absorbed as heat. If a photon has an energy greater than 1.2 electron volts and it hits an electron, the electron will move to the conduction band. However, the excess energy will be absorbed as heat. [“Photovoltaic Cell”].

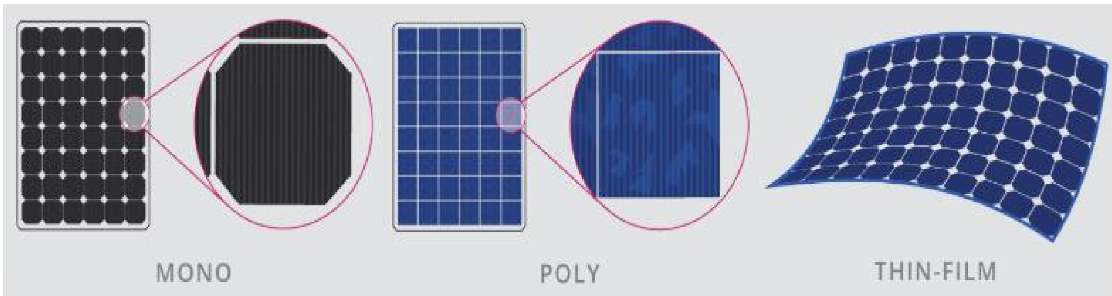


Figure 3-4: Solar Panel Material [“Types of Solar Panels”]

3.2.2.1 Monocrystalline

Monocrystalline solar cells are made from a single, pure crystal of silicone. In order to achieve this, a solid piece of silicone is cut into wafers. This process causes a lot of waste material. Since the photovoltaic cells are made of a pure silicone crystal, monocrystalline solar panels produce some of the highest efficiencies compared to other types of solar panels. Monocrystalline solar panels have efficiencies of around 20%. Monocrystalline solar cells also appear black because of the way light interacts with the pure silicone crystal. The main advantage of monocrystalline solar panels is their high efficiency. Due to this efficiency, monocrystalline solar panels are preferred when there is not a lot of space. However, achieving such a high efficiency requires an expensive manufacturing process with a lot of waste material. [“Types of Solar Panels”]

3.2.2.2 Polycrystalline

Polycrystalline solar cells are made from fragments of silicone crystals that are melted together in a mold. This manufacturing process is inexpensive and there is no waste material. Polycrystalline solar panels are blue in color because of the way the different silicone fragments reflect light. Because of the color and the silicone fragments, polycrystalline solar panels are said to be visually unappealing. Polycrystalline solar panels are average when it comes to efficiency. They offer an efficiency of around 15%. Polycrystalline solar panels offer a balance between efficiency and cost. [“Types of Solar Panels”]

3.2.2.3 Thin-Film

Thin-film solar panels are made from a variety of materials. The most common material is cadmium telluride (CdTe). In order to make the solar cell, a layer of cadmium telluride is placed between transparent conducting layers. Another material commonly used is amorphous silicon (a-Si). This is a non-crystalline silicone and the cells are not made up of a solid wafer. The last material that can be used is copper indium gallium selenide (CIGS). Thin-film solar panels have a very low efficiency of around 11%. This means that many panels would have to be used to achieve and desirable energy output. Due to the manufacturing process and depending on what type of material is used, thin-film solar panels are relatively inexpensive. They are also, as the name describes, reality thin. This means that thin-film solar panels are small, lightweight, and flexible. [“Types of Solar Panels”]

3.2.2.4 Solar Panel Comparison

As far as efficiency goes, monocrystalline solar panels are the best. They offer more efficiency than both the polycrystalline and thin-film solar panels. This means that if a small solar panel was desired, a monocrystalline would be the smartest option. The disadvantage to monocrystalline solar panels is that they are the most expensive. Thin-film solar panels are not very efficient, but they are flexible and lightweight. This means that if weight were a constraint, or if the solar panel was to be mounted to a curved surface, a thin-film solar panel would be the best option. The only reason to use a polycrystalline solar panel would be to have a balance between efficiency and cost. For these reasons, we chose monocrystalline.

Comparison Factors	Monocrystalline	Polycrystalline	Thin-Film
Efficiency	20%	15%	11%
Cost	Most Expensive	Average Cost	Least Expensive
Size Needed to Produce Same Power	Smallest	Average	Largest
Flexible	No	No	Yes
Life Span	25 years	25 years	20 years

***Table 3-2**: Solar Panel Comparison*

3.2.3 Temperature Sensors

The following section involves three different types of temperature sensors, including a thermistor, a thermocouple, and a resistive temperature detector. It is then discussed how each one functions to determine temperature.

3.2.3.1 Thermistor

A thermistor measures temperature by measuring the resistance of a semiconductor. This works because the resistance of a semiconductor is a function of its temperature. There are two types of thermistors: positive temperature coefficient and negative temperature coefficient. In a positive temperature coefficient thermistor, the resistance increases as the temperature increases. In a negative temperature coefficient thermistor, the resistance decreases as the temperature increases. The negative temperature coefficient sensor is the most used thermistor. In a thermistor, the relationship between the resistance and temperature is non-linear. [Davis] This relationship can be described by:

$$R_T = R_{25C} \cdot e^{\{\beta[(1/(T+273))-(1/298)]\}}$$

- R_{25C} is the nominal resistance of the thermistor at room temperature
- β is the material constant of the thermistor in Kelvin
- T is the thermistor's temperature in Celsius [Davis]

There are ways to linearize the relationship between temperature and resistance. These methods are known as resistive mode and voltage mode linearization. With resistive mode linearization, a resistor is connected in parallel to the thermistor. When the resistor has the same value as the resistance of the thermistor at room temperature, then the relationship between resistance and temperature will be linear. [Davis]

Voltage mode linearization works by placing a resistor in series with the thermistor, creating a voltage divider. The voltage divider is then connected to a stable voltage reference. This produces an output voltage that is relatively linear with respect to temperature. Like the resistive mode linearization, the resistor's value is the same as the thermistor's resistance at room temperature. [Davis]

3.2.3.2 Thermocouple

A thermocouple is made up of two different metals that are joined together to form a junction. Due to the Seebeck effect, the junction produces different voltages at different temperatures. The Seebeck effect occurs when two connected materials are heated. The heat causes electrons to flow from the hotter material toward the cooler material, creating a voltage. [Seebeck Effect] Thermocouples can be made from a wide variety of materials that effect cost and performance. Thermocouples are able to withstand extreme temperatures. The operating range of thermocouple is anywhere from -200°C to 1450°C, allowing them to operate in almost any condition. Thermocouples they suffer from reduced accuracy of plus or minus 2°C. [Davis]

3.2.3.3 Resistive Temperature Detector

A resistive temperature detector works much like the thermistor. A resistive temperature detector works by measuring resistance as a function of temperature. The measuring device consists of a coil of wire wrapped around a ceramic or glass core. The operating temperature of a resistive temperature detector ranges from -260°C to 850°C. This means that it can operate in a wide variety of temperatures and applications. Resistive temperature detectors also offer an accurate reading of temperature of plus or minus 1°C. The output is naturally linear. The disadvantages are that resistive temperature detectors can be expensive and fragile. [Davis]

3.2.3.4 Temperature Sensor Comparison

The following chart compares the three sensors researched and discussed above, using temperature range, temperature accuracy, cost, and size as the factors of comparison.

Comparison Factors	Thermistor	Thermocouple	Resistive Temperature Detector
Temperature Range	-40°C to 125°C	-200°C to 1450°C	-260°C to 850°C
Temperature Accuracy (+/- °C)	1	2	1
Cost	Lowest	Average	Highest
Size	Small	Small	Average
Durability	Durable	Average	Fragile

Table 3-3: *Temperature Sensor Comparison*

Selected Device: *Thermocouple*

3.2.4 Camera Technologies

The following section will involve the research and comparison of camera and image sensing technologies for the use in our project.

3.2.4.1 Image Sensor

The image sensor is the heart of a camera. It consists of an array of photodiodes that collect photons and converts them into electrons. Just like a photovoltaic cell, when a photon with energy higher than the bandgap energy collides with an electron in the valance band, the electron absorbs the energy and moves to the conduction band, creating electricity. With an array of photodiodes, each collecting a different number of photons, it is possible to capture an image. [“Understanding The Digital Image Sensor”]

There are two types of image sensor technologies: Coupled Charged Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS). Both CCD and CMOS image sensors use photodiodes to collect photons and convert them into and analog electrical signal that is stored as a charge. These analog signals must be amplified and then converted to digital signals. With a CCD image sensor, all the different charges form each pixel are all transferred together into a horizontal shift register and then into an amplifier. The output of the amplifier is still and analog signal and needs to be converted to a digital signal. With a CMOS image sensor, each pixel has its own CMOS transistor switch, allowing each pixel to be

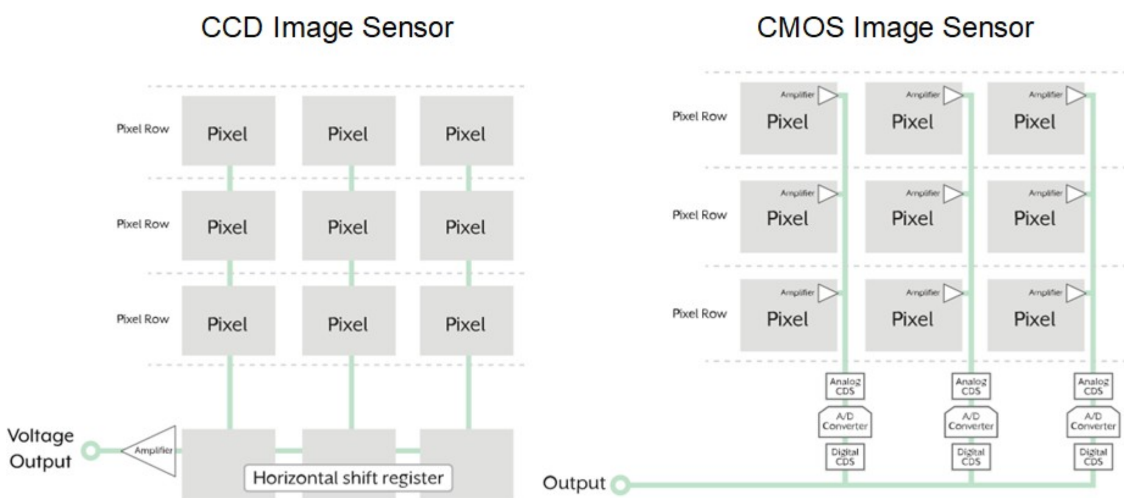


Figure 3-5*: CCD vs CMOS Image Sensor [“Understanding The Digital Image Sensor”]*

individually amplified. Each column of pixels also has its own analog to digital converter. CCD image sensors used to outperform CMOS image sensors, however as technology and manufacturing processes improved, CMOS image sensors now perform nearly as well as CCD image sensors. CMOS sensors have become more popular because of they are cheaper to manufacture and use much less power. [“Understanding The Digital Image Sensor”]

3.2.4.2 Color Detection

Photodiodes can only measure light intensity. They cannot determine the intensity of different colors. This means that an image sensor only consisting of photodiodes can only capture a grey scale image. In order to capture color, different light filters must be placed over the photodiodes so that only certain colors of light can pass through. Three different filters are needed for this process: a filter that only passes red light, a filter that only passes green light, and a filter that only passes blue light. These colors are used because they are the primary colors of light. So, every color is a combination of red, green, and blue. With these filters, an image sensor can measure the intensity of each primary color that an object produces and determine its color.

The most common type of filter array is known as a Bayer array. It consists of alternating rows of red-green and green-blue filters. This means that a Bayer array

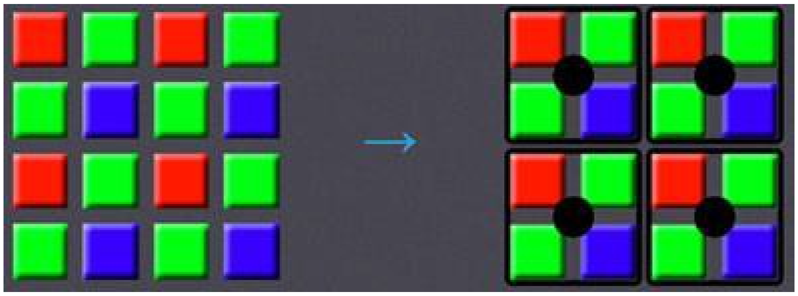


***Figure 3-6**: Bayer Array [“Digital Camera Sensors”]*

contains twice as many green sensors as it does blue or red sensors. This means that the image sensor is not receiving an equal amount of each primary color. This is done because the human eye is more sensitive to green light than blue or red light. The extra green pixels produce an image that appears less noisy and has greater detail than if there were an equal amount of red, green, and blue pixels. [“Digital Camera Sensors”]

3.2.4.3 Resolution vs Sensor Size

The resolution of a camera is defined by the number of horizontal pixels multiplied by the number of vertical pixels. This can be written as “Number of horizontal pixels X Number of vertical pixels” or by the total number of pixels in Megapixels. For example, an image sensor with 2048 horizontal pixels and 1536 vertical pixels has a resolution of 2048 X 1536 or 3.1 megapixels. The more pixels an image sensor has, the higher the image quality will be. However, pixel count does not solely

Name	Resolution	Number of pixels
VGA	640 x 480	0,3 megapixels
SVGA	800 x600	0,48 megapixels
XGA	1024 x 768	0,78 megapixels
SXGA	1280 x 1024	1,3 megapixels
UXGA	1600 x1200	1,9 (2) megapixels
SUXGA	2048 x 1536	3,1 megapixels
-	2048 x 2048	4,0 megapixels
-	2452 x 2054	5,0 megapixels
QUXGA	3200 x 2400	7,7 megapixels
HD	1280 x 720	0,92 megapixels
Full HD	1920 x 1080	2,1 megapixels

***Figure 3-7**: Common Resolutions For C-Mount Cameras [“Resolution of Sensors”]*

determine image quality. The size of the image sensor, and therefore the size of the pixels, also play a major factor in determining image quality. Larger pixels can capture more light. This means that larger pixels can produce images with more dynamic contrast. Larger pixels also produce less noise in the image.

3.2.4.4 Bit Depth

Bit depth is defined as the amount of color information stored in an image. The higher the bit depth is of an image, the more colors it can represent. A 1-bit image can only store two colors: 0 (white) and 1 (black). An 8-bit image can store 256 different colors, while a 24 but image can store over 16 million colors. Images with a higher bit depth require more storage space because there is simply more information in the image. ["What is Bit Depth"]

3.2.4.5 Exposure

The exposure of a camera is the amount of light that the image sensor receives. Exposure is defined by three different elements: ISO speed, aperture, and shutter speed. ISO is an acronym that stands for International Standards Organization and it describes how sensitive the image sensor is to light. Each value of the ISO represents a "stop" of light and each increment of the ISO represents doubling or halfling the image sensor's sensitivity to light. The lower the ISO is, the less sensitive the less sensitive the image sensor is. This means that the image sensor will take in less light and will produce less noise. The higher the ISO rating is, the more sensitive the image sensor is. This means that the image sensor will take in more light and therefore produce more noise. [Kun]

Aperture controls the diaphragm of the lens, which controls the amount of light going through the lens to the film plane. Aperture is indicated by the f-number which represents a "stop" of light. With a small f-number, a large amount of focused light can pass through the lens, even at a fast shutter speed. At a high f-number, only a small amount of focused light can pass through the lens, even at a slow shutter speed. The f-number only refers to the amount of light passing through the lens. The focal length of a lens does not affect the f-number. The f-number indicates the same amount of light passing through a 35mm lens as it does a 100mm lens. [Kun]

Shutter speed is the speed at which the device covering the lens opens and closes. The shutter speed determines the duration that light is hitting the image sensor. Each shutter speed value represents a "stop" of light and is measured in fractions of a second. Shutter speed allows motion to be recorded in different ways. If the shutter speed is faster than the moving object or background, the image will be sharp. If the shutter speed is slower than the moving object or background, the image will be blurry. [Kun]

***Figure 3-8**: Different Types of Exposure [Kun]*

3.2.5 Voltage Regulators

A voltage regulator is a circuit that maintains a constant output voltage regardless of varying input voltage or load current. Voltage regulators are necessary when the input voltage is greater than desired. Voltage regulators can also add a layer of protection to a circuit by preventing voltage spikes that would normally damage electrical components. There are two types of voltage regulators: linear voltage regulators and switching voltage regulators. [Sattel]

3.2.5.1 Linear Voltage Regulator

A linear voltage regulator acts as a voltage divider in order to maintain a desired output. This means that all of the voltage not being used gets converted into heat energy. Converting usable electricity into heat is incredibly inefficient. The efficiency of linear voltage regulators decreases as the input voltage decreases. Due to this, linear voltage regulators are used for low power applications and when the difference between the input and output voltage is small. [Sattel]

A basic linear voltage regulator can be built from an op-amp and some type of transistor. However, one of the most common three terminal linear voltage regulator is the LM7805. The LM7805 has an input, a ground, and an output. Linear voltage regulators use up some power in order to operate. The dropout voltage is the number of volts that the linear voltage regulator uses up. In the case of the LM7805, the dropout voltage is 2 volts. This means that in order to achieve the desired output voltage, the input voltage must be equal to or greater than the input voltage plus the dropout voltage. So, to produce an output voltage of 5 volts, the input voltage must be at least 7 volts. [“µA7800”]

3.2.5.2 Switching Voltage Regulator

A switching voltage regulator maintains a constant output voltage by switching the input voltage on and off and storing that energy. A switching voltage regulator takes a DC input and switches it on and off at a frequency in MHz. The switched voltage is then converted to a DC output using a rectifier and a low pass filter. Since the input voltage is constantly being switched on and off, switching voltage regulators are much more efficient than linear voltage regulators. However, this constant on and off switching of the input voltage introduces noise into the circuit. This makes switching voltage regulators not ideal for audio applications. [Davis]

There are two types of control for switching voltage regulators: pulse width modulation and hysteretic. In a pulse width modulation switching regulator, there is feedback from the input voltage back to the pulse width modulator controller which varies the amount of time the input is kept on. This is can also be referred to as the duty cycle. As the frequency of the switching increases, the size of the storage inductor or capacitor needed is decreased. Higher frequency also contributes to higher noise introduced into the circuit. [Davis]

A hysteretic switching regulator consists of a comparator with a hysteretic input that compares the output voltage to a reference voltage. When the output voltage goes above the hysteretic reference voltage, the input voltage is switched off. When the output voltage goes below the hysteretic reference voltage, the input is switched on. Due to this design, a hysteretic switching regulator does not need a voltage-error amplifier. This means that it can respond to a change in the load current or input voltage almost instantaneously. This makes the hysteretic switching regulator the fastest DC to DC converter control technique. The disadvantage of hysteretic switching regulators is that their switching frequency varies with the output capacitor's equivalent series resistance. This means that as the capacitor ages, the frequency of the switching voltage regulator will change. [Davis]

3.2.5.3 Voltage Regulator Comparison

The following chart will compare voltage regulation, using efficiency, heat, noise, high power application capability and time of adaptation to load current change as the factors of comparison.

Comparison Factors	Linear	Pulse Width Modulation	Hysteretic
Efficiency	Low	High	High
Heat	High	Low	Low
Noise	Low	High	High
Can Be Used for High Power Applications	No	Yes	Yes
Amount of Time to Adapt To Load Current Change	Average	Average	Fast

Table 3-4: *Voltage Regulator Comparison*

3.2.6 Switching Devices

The following section involves the research and discussion of the three switching devices being considered for our prototype. The devices being considered are electromechanical relays, Bipolar Junction Transistors, and MOSFET transistors.

3.2.6.1 Electromechanical Relay

An electromechanical relay is an electromagnetic switch that turns on or off with the use of a control signal. Inside a relay is a coil of wire wrapped around an iron core. When the control signal is passed through the coil of wire, a magnetic field is generated that attracts the metal contacts within the relay. This is what forms the electrical connection between the relay's inputs. The magnetic field holds the electrical contacts together if the control signal remains on. When the control signal is turned off, there is no longer a magnetic field being produced and the contacts are then separated from each other by a spring. ["Electrical Relay"].

Figure 3-9 provides a graphic of how an electromagnetic relay is designed and how it functions.

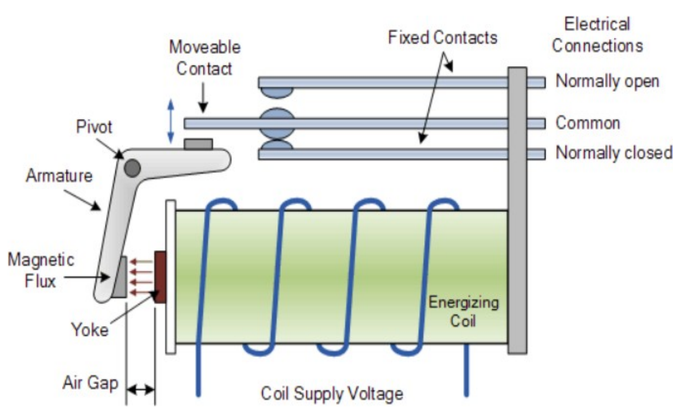


***Figure 3-9**: Electromagnetic Relay ["Electrical Relay"]*

Electromechanical relays come in several different configurations. The relay can be designed so that the contacts are either normally open or normally closed. Relays also have configurations that include single pole – single throw, single pole – double throw, double pole – single throw, and double pole – double throw. In these cases, a "pole" is a contact and they can be connected, or "thrown", together. When contacts are connected, they are described as a "make". When contacts are separated, they are described as a "break". So, a relay could be described as a "break before make". ["Electrical Relay"]

Figure 3-10 provides shows the different possible configurations of a electromagnetic relay.

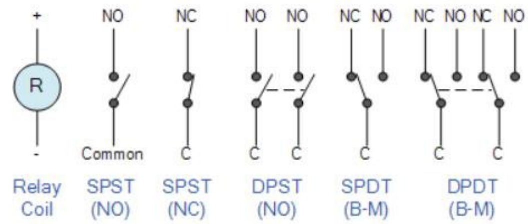


Figure 3-10*: Configurations of an Electromagnetic Relay [“Electrical Relay”]*

One of the dangers of electromechanical relays involves damage to the circuit providing the control signal. The main component of an electromechanical relay is a coil of wire. Even though the main purpose of the coil of wire is to produce a magnetic field to attract the contacts, the coil unintentionally acts as an inductor. When the control signal sends current trough the coil and then shut off, a large electromotive force voltage is produced as the magnetic flux collapses within the coil. This voltage is very high compared to the control signal voltage and can damage the electrical components sending the control signal. To prevent this from occurring, a diode can be connected from the control signal to the relay coil so that the diode is reversed biased when the coil generates an electromotive force voltage. [“Electrical Relay”]

One of the main disadvantages of the electromechanical relay is that it is a mechanical device. Because of this, its ability to switch on and off quickly is limited. The mechanical parts within the relay can also wear out. The metal contacts can corrode with time, making the relay unusable. Electromechanical relays are also noisy due to the contacts suffering from contact bounce. [“Electrical Relay”]

3.2.6.2 Bipolar Junction Transistor

Bipolar junction transistors (BJT) are semiconductor devices that can be used for amplification or switching. A bipolar junction transistor is made up of three sections. These sections can consist of either two N regions and one P region or two P regions and one N region. These transistors are known as NPN and PNP transistors. A N region is one that is doped with extra electrons while a P region is one that is doped to have extra holes, or lack of electrons. When silicone is used as the base material, it is doped with either boron or phosphorous. Since boron has three valence electrons and silicone has four valance electrons, this causes there to be a lack of electrons when the boron atoms bond with the silicone atoms. The opposite is true for phosphorus. Phosphorus has five valance electrons and when the silicone atoms bond with phosphorous atoms, there is a surplus of electrons. [“Bipolar Transistor”]

The three regions of bipolar junction transistors are known as the emitter, base, and collector. The base is the middle region of the transistor while the emitter and the collector are the end regions of the transistor. Depending on how these regions are connected, bipolar junction transistors can be operated in four different modes. These modes are forward active, reverse active, saturation, and cut-off. For amplification, the forward active mode is the most important. In forward active, the majority carriers of the emitter flow through the base and end up in the collector. Reverse active is essentially the same as forward active except the original location of the collector is swapped with the emitter. This results in a poor amplifier because bipolar junction transistors are designed to have a small emitter and a large collector. This is done so that the collector has the greatest chance of catching the majority carriers from the emitter. Since the original emitter and collector are switched in reverse active, the emitter is large, and the collector is small. This means that the collector will catch very few majority carriers. Saturation and cut-off are used for switching applications. When in saturation the transistor acts as a short circuit. When the transistor is in cut-off mode, it acts as an open circuit. ["Bipolar Transistor"]

$I_E = I_B + I_C$ $I_C = I_E - I_B$ $I_B = I_E - I_C$	$\alpha = \frac{I_C}{I_E} = \frac{\beta}{1+\beta}$ $\beta = \frac{I_C}{I_B} = \frac{\alpha}{1-\alpha}$
$I_B = \frac{I_C}{\beta} = \frac{I_E}{1+\beta} = I_E(1-\alpha)$	
$I_C = \beta I_B = \alpha . I_E$	$I_E = \frac{I_C}{\alpha} = I_B(1+\beta)$

Figure 3-11*: BJT Formulas ["Bipolar Transistor"]*

When in forward active, the bipolar junction transistor can be connected in three different ways: common base, common emitter, and common collector. A common base transistor has its base connected to ground. The base is also common to both the input and the output signal. With this connection, the transistor output has a voltage gain, but no current gain. ["Bipolar Transistor"] The common base voltage gain is:

$$A_v = \frac{V_{out}}{V_{in}} = \frac{I_C * R_L}{I_E * R_{IN}}$$

When a bipolar junction transistor is in the common emitter configuration, the emitter is connected to ground. The input signal is connected between the base and the emitter and the output is connected between the collector and the emitter. The common emitter configuration is also known as an inverting amplifier because an increase in the base voltage causes a decrease in the output voltage and a decrease in the base voltage causes an increase in the output voltage. This configuration produces the highest current and power gain out of the three transistor configurations. This is because the input is connected to a forward biased PN junction, giving it a low input impedance, and the output is connected to a reversed biased PN junction, giving it a high output impedance. [“Bipolar Transistor”] The common emitter voltage gain is:

$$A_v = \frac{V_{out}}{V_{in}} = -\frac{R_L}{R_E}$$

When a bipolar junction transistor is in the common collector configuration, the collector is connected to ground. The input signal is connected to the base and the output is connected to the emitter. Another name for this configuration is “emitter follower”. The common collector configuration is useful for impedance matching because it has a high input impedance and a low output impedance. When a transistor is connected in the common collector configuration, the output has a current gain but no voltage gain. [“Bipolar Transistor”] The current gain is:

$$A_i = \frac{I_{out}}{I_{in}} = \frac{I_C}{I_B} = \beta + 1$$

To act as a switch, a bipolar junction transistor takes advantage of the cut-off and saturation regions. When the transistor is in the cut-off mode, there is zero input base current, zero output collector current, and maximum collector voltage. This creates a large depletion layer and prevents current from flowing through the transistor. If the bipolar junction transistor is treated like two PN junctions, both PN junctions are reversed biased in cut-off mode, therefore there is no current flowing. In saturation mode, the transistor is biased so that the maximum amount of base current is applied. This causes the collector current to be at its maximum and the collector-emitter voltage drop to be at a minimum, resulting in a small depletion layer. If the bipolar junction transistor is treated like two PN junctions, both PN junctions would be connected in forward bias. [“Transistor as a Switch”]

When used as a switch, a bipolar junction transistor acts as a single-pole single-throw solid state switch. The most effective way to switch moderate to high amounts of power is to have an open-collector output and with the emitter connected to ground. When a real-world bipolar junction transistor is operating in the cut-off region, a small amount of leakage current flows through the transistor. This means that a non-ideal transistor operating in cut-off does not act like a true open circuit. A non-ideal transistor also has problems when operated in saturation mode. A non-ideal transistor has a low resistance value that creates a small saturation voltage (V_{CE}) across it. Despite these flaws, bipolar junction transistors still operate well as switches. [“Transistor as a Switch”].

Figure 3-12 provides a schematic view of a Bipolar Junction Transistor functioning a mode of cut-off and a mode of saturation, providing the switching behavior discussed above.

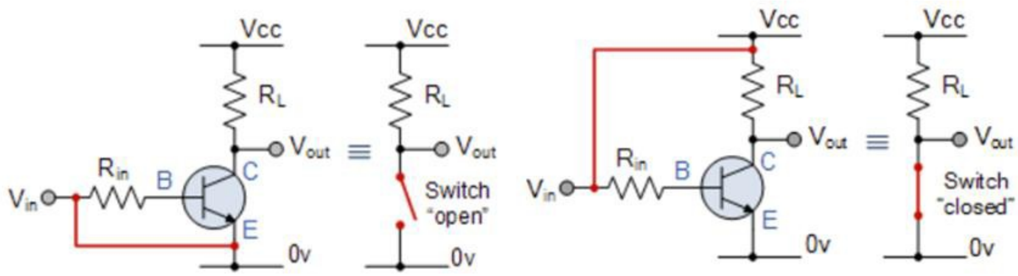


***Figure 3-12**: BJT Operating in Cut-Off Mode and Saturation Mode ["Transistor as a Switch"]*

3.2.6.3 MOSFET

MOSFET stands for metal oxide semiconductor field effect transistor. MOSFETs are mainly used for switching or amplifying electronic signals. MOSFETs have four terminals: drain, gate, source, and body. There are two types of MOSFETs: a n-channel and a p-channel. In a n-channel MOSFET, the majority carriers are electrons, and in a p-channel MOSFET, the majority carriers are holes. ["The MOSFET"]

MOSFETs are offered in two different forms: depletion and enhancement. Depletion mode MOSFETs is normally conducting without the application of a gate bias voltage. For a n-channel MOSFET, a negative gate-source voltage will deplete the channel of its free electrons and therefore stop conducting. For a p-channel MOSFET, a positive gate-source voltage will deplete the channel of its free holes and stop conducting. An enhancement mode MOSFET is not conductive when the gate bias voltage is zero. In a n-channel MOSFET, applying a positive voltage to the gate attracts more electrons toward the oxide layer around the gate, increasing the thickness of the channel and causing conduction. Increasing the positive gate voltage causes channel resistance to decrease and the drain current to increase. For a p-channel MOSFET, applying a negative voltage to the gate causes the channel to thicken and conduct. Enhancement mode MOSFETS are more common than depletion mode MOSFETs. This because they have low resistance while conducting, high resistance when not conducting, and a high input resistance. All these characteristics allow enhancement mode MOSFETs to perform well in switching applications. Enhancement mode MOSFETs are used to make complementary metal oxide semiconductors, which perform as logic gates. ["The MOSFET"]

3.2.6.4 Switching Device Comparison

The following chart compares the three switching devices discussed above. The chart uses efficiency, construction, switching speed, high power application ability, and size for factors of comparison. Due to the nature of how we are to use the switching device, a priority was placed on the efficiency of the device and the size of the device when comparing these three components.

Comparison Factors	Relay	BJT	MOSFET
Efficiency	Average	High	Highest
Construction	Mechanical	Semiconductor	Semiconductor
Switching Speed	Slow	Fast	Fastest
Can Be Used For High Power Applications	Yes	No	Yes
Size	Large	Small	Small

Table 3-5: *Switching Device Comparison Chart*

3.2.7 Microphone Types

The following section includes the research of three different types of microphones being considered for our prototype, which include: dynamic, condenser and ribbon microphones. A chart is then used to make comparisons between the three types of microphones. The comparison of these microphones focuses both on the functionality of each type of microphone and the acutance to the sounds it can receive and send.

3.2.7.1 Dynamic Microphones

Dynamic microphones work by using a coil of wire and a magnet. The coil of wire is connected to a diaphragm that moves back and forth with the vibrations in the air. The coil of wire vibrates around a stationary magnet, creating the electrical signal. Dynamic microphones are able to handle high sound pressure levels, meaning they can record loud sounds without distorting. Since the microphone uses a coil of wire and magnet to record sound, it does not require external power to operate. [Briones]

3.2.7.2 Condenser Microphones

Condenser Microphones work by using a conductive diaphragm that sits close to a metal backplate. This creates a capacitance that changes as the diaphragm vibrates, converting sound into an electrical signal. Due to this design, condenser microphones offer high sound quality and accuracy. However, condenser microphones cannot record at high sound pressure levels. Condenser microphones also require external power to operate. [Briones]

3.2.7.3 Ribbon Microphones

Ribbon microphones use a light metal ribbon suspended in a magnetic field that senses the velocity of the air as well as air displacement. This allows for improved sensitivity to higher frequencies while at the same time making them less harsh. [Briones] Ribbon microphones are a vintage technology and they are not very popular today. Due to the thinness of the metal ribbon, ribbon microphones are somewhat fragile. [“Ribbon Mics”]

3.2.7.4 Microphone Type Comparison

The following chart compares the three types of microphones discussed above. This chart uses sound quality, durability, quiet sound detection capability, and loud sound detection capability as its factors of comparison. Due to the nature of our project several of these factors a paramount for a successful device. The leading being sound quality as well as the sensitivity of the microphone. Though durability is a concern, we must focus on the quality of the recordings provided by the microphone of choice and the audio range that it will give as this will allow our device to be as versatile and accurate when determining the location and classification of the picked-up sound.

Comparison Factors	Dynamic	Condenser	Ribbon
Sound quality	Average	High	High
Durability	High	Average	Low
Detects Quiet Sounds	Yes	Yes	Yes
Detects Loud Sounds	Yes	No	No

Table 3-6*: Microphone Comparison Chart*

3.2.7.5 Microphone Polar Shapes

Cardioid microphones only capture sound in the front and block out everything else. Super-cardioid captures sound from the front of the microphone and a small amount of sound from the back of the microphone, while rejecting the sound from the sides. Figure-8 microphones capture an equal amount of sound from the front of the microphone as well as behind the microphone. Omnidirectional microphones capture sound from all directions and produces a more natural sound. [Briones] The figure below shows the various microphone polar shapes.

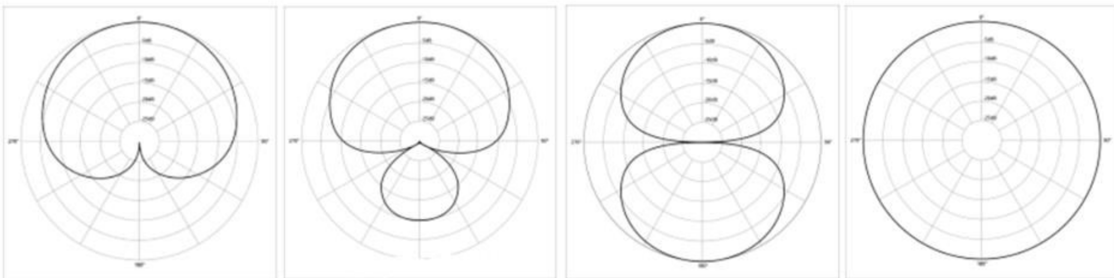


Figure 3-13: *From left to right: cardioid, super-cardioid, figure-8, and omnidirectional [Briones]*

3.2.8 Operational Amplifiers and Filtering

After the microphone receives the incoming audio signal, the signal typically must pass through an amplifier to increase its gain before entering the analog to digital converter. In addition to amplifying the signal, the operational amplifier circuit will also be operating as an active low-pass filter to remove the DC component of the signal.

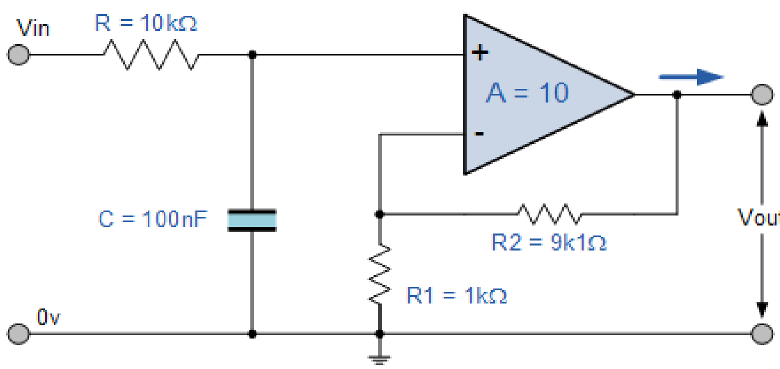


Figure 3-14: *Non-Inverting Low Pass Filter*

Notice that in the low pass filter above, the gain for the circuit is as follow:

$$A = (1 + \frac{R2}{R3})$$

A low pass filter works by only allowing frequencies below a certain cut off point to pass the filter, with any frequencies higher than this cut off subsequently being removed from the signal. In order to achieve the desired bandwidth for the Nyquist-Shannon sampling theorem, a low pass filter is used to minimize the aliasing effect during the ADC process.

Since amplification is also desired in this process for DiSEL's application, an active low pass filter is built using op amps. This will provide the required increase in gain needed and the filtering required to our input signal.

The frequency response of an active low pass filter is shown below:

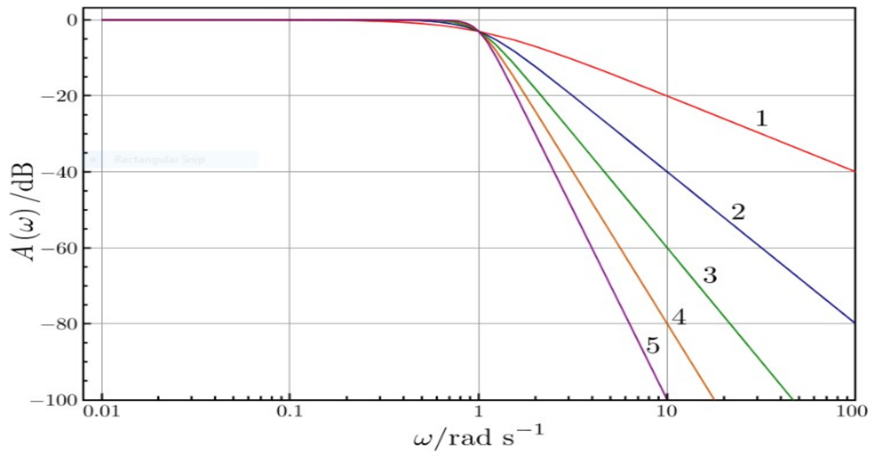


***Figure 3-15**: Frequency response of a low pass filter, showcasing how the order of the filter affects the slope of the drop off*

The following operational amplifiers were compared based the team's prior experience with op amps in other UCF courses and their ease of availability at the university.

3.2.8.1 LF351 JFET Op Amp

The LF351 op amp was the primary operational amplifier that is utilized during the Analog Filter Design course at UCF. Due to this, many low pass filter topographies are readily available in notes and other course resources that use this amplifier.

Some of the LF351's specifications are shown in the datasheet below:

Parameter	Value	Unit
Supply Voltage	±18	V
Differential Input Voltage	30	V
Input Voltage Range	±15	V
Power Dissipation	500	mW
Operating Temperature	0 ~ +70	C
Storage Temperature Range	-65 ~ +150	C
Input Resistance	1000	MΩ
Slew Rate	13	V/μs

***Table 3-7**: LF351 Op Amp Specifications*

This operational amplifier also has the added benefit of being almost identical to Texas Instrument's TL072, allowing even greater options for purchasing or acquiring these op amps.

3.2.8.2 LM741 Operational Amplifier

The LM741 operational amplifier is primarily used by UCF for their digital communications course. Since the course did not focus on designing different types of filters, the number of low pass filter topographies on hand that could be used to for the purposes of DiSEL are much small than that of the LF351.

Below are the specs for the LM741 operational amplifier:

Parameter	Value	Unit
Supply Voltage	±22	V
Differential Input Voltage	30	V
Input Voltage Range	±15	V
Power Dissipation	500	mW
Operating Temperature	0 ~ +70	C
Storage Temperature Range	-65 ~ +150	C
Input Resistance	2	MΩ
Slew Rate	0.5	V/µs

***Table 3-8**: LM741 Op Amp Specifications*

3.2.8.3 Operational Amplifier Comparisons

The DiSEL team has experience using both the LM741 and the LF351 operational amplifiers for designing active low pass filters. Although both op amps have similar specifications overall, when comparing the two operational amplifiers side by side, there are two major factors that make it clear that the LF351 op amp should be chosen for the applications desired for DiSEL over the LM741. The first major reasoning is due to the low input resistance of the LM741 when compared to the LF351. This low input resistance means that the resistor used for R1 must be carefully selected as to not heavily divide the voltage between R1 and the input resistance of the operational amplifier. The second reason is due to the low slew rate of the LM741, which has the potential of causing distortions. The following tablle compares the different operational amplifiers being considered by our team using supply voltage, differential input voltage, input voltage range, power dissipation, operating temperature, storage temperature range, input resistance and slew rate as parameters of comparison

Parameter	**LF351 Value**	**LM741 Value**	**Unit**
Supply Voltage	±18	±22	V
Differential Input Voltage	30	30	V
Input Voltage Range	±15	±15	V
Power Dissipation	500	500	mW
Operating Temperature	0 ~ +70	0 ~ +70	C
Storage Temp Range	-65 ~ +150	-65 ~ +150	C
Input Resistance	***1000***	***2***	***MΩ***
Slew Rate	***13***	***0.5***	***V/µs***

Table 3-9*: Comparison of the Specifications of the LF351 and the LM741*

3.2.9 Analog to Digital Converter

Analog to Digital Converters (also known as ADCs) play a pivotal role in the process of modern signal processing. These devices allow analog signals to be converted into digital signals that can be read, analyzed, and manipulated by computers through the use of digital signal processing. For the purposes of DiSEL, our main focus is going to be using ADCs to convert the analog audio signals recorded by the microphones into a digital signal. Once the converted into a digital signal, the microcontroller can begin using the fast Fourier transform algorithm to generate the signals frequency spectrum and attempt fingerprinting to identify whether or not the sound is considered a “trigger” sound, thereby activating DiSEL’s other primary functions.

Analog to digital converters work by taking the continuous time-domain signal and quantizing the signal based on the ADC’s sampling frequency. Based on Nyquist’s theorem, the minimum sampling frequency required to be able to adequately sample a given signal is:

$$f_{Nyquist} = 2 * f_{max}$$

Where fmax is the max frequency of the given signal. Using a sampling frequency greater than this can increase the quality of the reconstructed signal but does so at the cost of using greater bandwidth.

Another factor that can determine the quality of a digital signal when compared to its original analog signal is the bit resolution of the ADC. The more bits that an ADC has at its disposal, the more quantization levels are available for use when quantizing the analog signal. By increase the amount of quantization levels, the values of the digital signal can more accurately represent the values of the analog signal.

Number of ADC bits = N

Number of Quantization Levels = 2^N

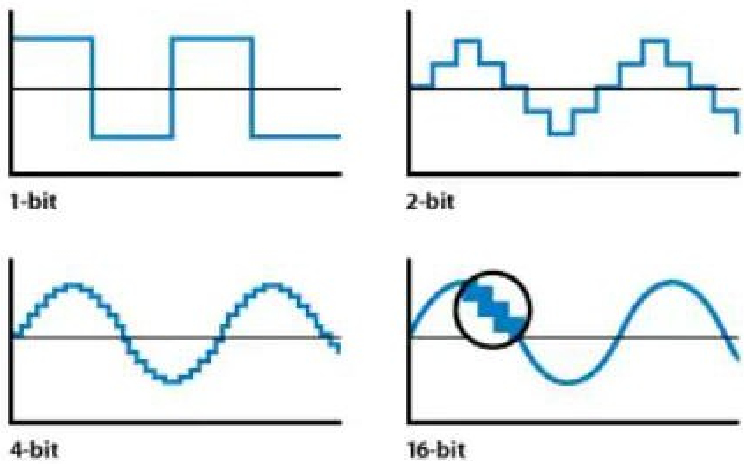


Figure 3-16*: Graphs showcasing how the number of bits in an ADC affects the number of quantization levels and the precision of the digital signal.*

3.2.10 Machine Learning

Machine learning is the study of algorithms and data that allows a computer system to complete a specified task without being given explicit instructions to perform that task. A formal definition of algorithms that are studied in machine learning is given by Tom Mitchell, “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”[Mitchell] Meaning that when a program performs a specific task is accuracy and effectiveness at performing that task will increase with the experience or number of times that task is performed by the program. The most important quality that is provided by machine learning is that it allows the system to become as efficient as possible when performing the given task. Machine learning is being used today in everyday life from providing curated ads to internet users, to the multiple applications used by computer vision.

3.2.10.1 Subsystems of Machine Learning

There are three main subsystems of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning is like that of the traditional education system that we use. Much like a student-teacher model a dataset that is used for training the system. The learning process is accomplished by having the correct answer already being contained in the system, the algorithm then makes predictions on the data contained in the dataset and is then corrected by the user. The algorithm is then continuously trained and corrected until a satisfactory level of performance is achieved by the algorithm. The problems involved in supervised learning can be further broken down into the two subsets of classification problems and regression problems. Classification problems are when the output can be put in a class, for instance a certain animal sound can be classified as a specific animal such as a dog, cat, etc. Regression problems are when the output can be provided as a real number such as numerical characteristics of an animal to determine subclasses of animals from images or the weight of an object to determine what the object is.

Unsupervised learning, unlike supervised learning which provides input and output sets, provides the algorithm with only input data and no outputs. This provides the algorithm to gain a greater understanding of the structure or distribution of the data. This in turn allows the algorithm to discover patterns in the provided data. These problems can be further broken down into two subsets of clustering and association problems. Clustering problems involve the grouping of data forming clusters based on discovered patterns. Association problems involve the grouping of data based on rules that are discovered such as a cause an effect where if one action is done, then another action is also likely to occur. Though usefully in some aspects of our problem, these subsets are further from the desired results we are needing for our project.

The final type of technique brings the reinforcement learning. This subsystem of machine learning takes both aspects of supervised and unsupervised learning methods and combines them. In this subsystem only a portion of the input data will be labeled (correct answers) while the rest of the input data will remain unlabeled. This system is used to have some of the capabilities that are provided by supervised learning while still limiting the cost that is required to label the input data. The information in table 3-10 provides a side by side comparison of the three types of learning using the comparison aspects discussed above as its comparison factors.

In conclusion, we will be using a supervised learning system as it will provide us the benefits found with classification problems of a supervised learning system, that outweigh the costs caused by it that could be reduced with the other two types, but that reduction could also hinder the overall results of the classification model, since training a classifier requires labeled data to classify at a much higher rate of accuracy.

Comparison Factors	Supervised Learning	Unsupervised Learning	Reinforcement Learning
How is the training data given?	The network is given training inputs with their associated outputs.	The network is only given training inputs with no associated outputs	The network has a percentage of training inputs given with their associated outputs. The rest are given without.
How does the network learn?	The network predicts the label for each input, neural weights are then physically adjusted to improve accuracy	The network looks for patterns and distributions in the input's structures, learning how the inputs relate to each other.	The network uses both adjustments to neural weights, and patterns seen in the input data.
Types of Problems	Classification and Regression	Clustering and Association	Classification, Regression, Problems with long-term vs. short-term reward trade-off
Cost	High. Requires all inputs to be labeled for training, weights must be adjusted by hand	Low. Has no labeling requirements for its training inputs?	Medium. Only requires a percentage of inputs to be labeled.

***Table 3-10**: Comparison of the Types of Machine Learning Subsystems*

3.2.10.2 Machine Learning Framework

The typical framework that is used with machine learning is an artificial neural network (ANN). This type of computing system is loosely designed and inspired by the neural network of the human brain. This framework is based on a collection of nodes often referred to as artificial neurons, modeling neurons in the brain. These artificial neurons can then be used to transmit signals. A neuron can also receive a signal which it then in turn processes the signal and can then signal another neuron that it is connected to. Using a non-linear function, the output of each neuron is computed using the sum of its inputs. The connections between the neurons called edges and the neurons themselves contain a weight. This weight is adjusted as the learning progresses, increasing, or decreasing based on how strong the signal is at the specified connection, allowing for the system to establish a more efficient performance when completing the given task. The pattern of neurons is typically designed in a layered fashion where different transformations are applied to the data as it passes through each layer.

3.2.10.3 Convolutional Neural Network (CNN)

A typical convolutional neural network consists of several different layers stacked together in a deep architecture: an input layer, a group of convolutional and pooling layers, a limited number of fully connected hidden layers, and an output (loss) layer.[Piczak] In the convolutional layers the goal is to abstract features from the images. Typically, the first layer abstracting low-level features such as edges, gradient orientation, and color. Additional convolutional layers can be added to obtain higher-level features. Pooling layers are used to lower the complexity of the results created by the convoluting, by reducing the spatial size. The two types of pooling are max pooling which will return the maximum value from the combination of values inside the specified kernel being used for pooling, and average pooling which returns the average of all values that are inside the kernel used for pooling. A fully connected hidden layer is then used to classify the image using the preceding layer as its input. This layer outputs an N dimensional vector where N represents the number of classes the algorithm must choose from. ***Error! Reference source not found.*** shows a convolutional neural network.

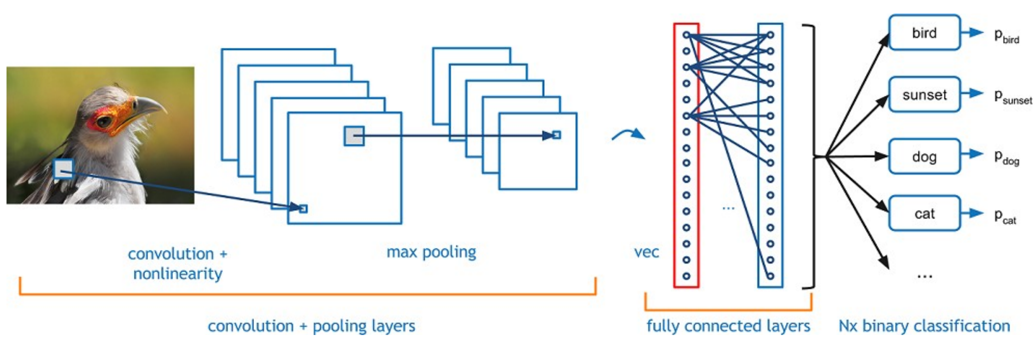


Figure 3-17: Convolutional Neural Network Visual [Adit]

3.2.10.4 Activation Functions

An activation function is a non-linear function that allows one to introduce those non-linear properties inside of an artificial neural network. The main use of these functions is to convert the input of a node into an output, which can then in turn be used as the input for the succeeding layer in the network. Functions that have been previously used include the Logistic and Hyperbolic Tangent functions but have fallen out of favor. The function that has become their replacement is the Rectified Linear Units Function known as ReLU. One problem that the ReLU function presents is that some gradients can be fragile in training resulting in dead neurons. This was addressed by a modification called Leaky ReLU which introduced a small slope that will keep updates alive. This eliminates neurons from dying off during training. *Equation 3-1* shows the mathematical expression of ReLU. Figure 3-18 shows the graphical representation of both ReLU and Leaky ReLU.

$$R(x) = \max(0, x)\ ;\ ie.\ if\ x < 0, R(x) = 0\ and\ if\ x \geq 0, R(x) = x$$

***Equation 3-1**: ReLU*

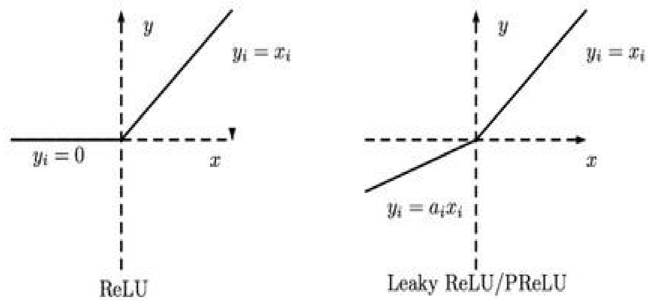


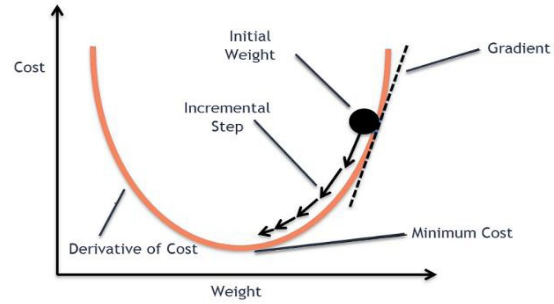
***Figure 3-18**: Graphical Representation of ReLU and Leaky ReLU*

3.2.10.5 Dropout Learning Algorithm

Deep neural networks in general are susceptible to overfitting where the estimated values are much greater than the training cases. This can cause a high rate of accuracy when new data is given to the system. Essential the network learns that it can be right a high enough average, for example 80 percent if it simply chooses one certain label. This can be caused by have a network that is developed where the problem being asked to solve is too simple for the complexity of the network. Another reason this may occur is a poor dataset being used where data being presented to the network is well represented in one label and not in another. One solution to this problem is using the dropout learning algorithm, this regularization technique accomplishes this by randomly removing with a predefined probability every hidden unit during each training iteration. This in turn prevents complex co-adaptations on training data. Though this step may not be necessary in regard to this project, it should always be kept in mind along with the risk of overfitting when assessing the quality of the networks being developed. This reminds us that if the results of the network do not match with the entire picture there needs to be made an assumption that overfitting may be occurring within the network and steps must be taken to address it, either by using a dropout algorithm, assessing the data being used by the network, or adjusting the network itself.

3.2.10.6 Optimization Algorithms

The idea of optimization algorithms inside of machine learning begins with the concept of gradient descent. This is an iterative optimization algorithm for finding the minimum of a function. The algorithm takes steps proportional to the negative gradient of the function at the current point. [Kapil] This in turn allows the determination of parameters (or weights) causing a reduction in the cost. This process is repeated over every training iteration until the minima of the cost function is reached.



$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Figure 3-19*: Gradient Descent Equation and Graphical Representation [Kapil]*

There are three separate ways to implement this algorithm: Batch Gradient Descent, Stochastic Gradient Descent, and Mini Batch Gradient Descent.
Batch Gradient Descent uses all the training data for each single step. The averages of the gradients of all training data is then calculated and the mean is then used to update the weights. Because of this, the number of times that the weights are adjusted is dependent on the number of epochs used. The advantages of this method are that it provides a steady track as the minima is converged upon, which also results in a stable error gradient. The disadvantages are that since BGD requires that all the training data is to be used, when dealing with a large dataset, an increase of memory may be needed to be able to process the entire dataset. The stability of the error gradient may also become a problem as a local minimum may be reached without the ability of escaping it, causing the inability to reach the global minima.

Stochastic Gradient Descent on the other hand uses a single training sample that is passed in the neural network. The gradient is then computed, and the weights of the layers are adjusted. Whereas in BGD the parameters (weights) are only updated one time, with SGD the number of training data samples that are used, results in the number of times the parameters are updated. For example, if there are 5000 images within a dataset the parameters are then updated 5000 times using 1 image for each training iteration. This operation of having a high number of updates addresses an issue seen with BGD as the steps towards the minima of the cost function has an oscillating pattern rather than a smooth one, this allows the ability to escaping local minima within the cost function. Another advantage of SGD, especially when dealing with a larger dataset is the decreased need for large memory as it uses a single training sample for each iteration rather than all training samples. Disadvantages are still present within this method. Due to the oscillating pattern of steps, there is a possibility of an increased amount of time required to converge upon the global minimum.

The Mini Batch Gradient Descent method is a combination of the other two methods. This is accomplished by taking the training set and dividing them into multiple smaller groups. Each of these groups contain an equal number of training samples. Each of these groups is then sent into the network, which in turn computes the loss of each sample, the average is then taken and used to update the parameters of the network. For example, if there are 5000 images within a dataset. The dataset could then be divided into 50 mini batches resulting in 100 images within each. The gradient descent equation would then be iterated over by the number of mini batches that are created which in this example would be 50. This allows for the advantages of both SGD and BGD to be utilized.

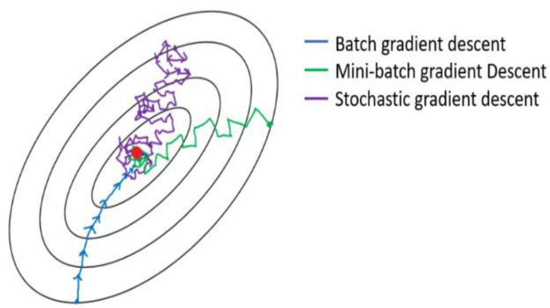


***Figure 3-20**: Comparison of Gradient Descent Methods Path to Global Minimum [Dabbura]*

3.2.11 Sound Classification

The focus of our sound classification model will be of environmental sounds, specifically in the area of bird calls with the possibility of expansion in the future. The idea of sound classifying bird calls is to take a bird call from a sensor, process that audio signal into an image for the model to take in as an input, and after having a model accept the image, return the result of the classification. The classifier focuses on 29 bird species. Training is necessary for the classification to work, to do this a custom query was issued using the Xeno-Canto API that resulted in over 17000 bird calls from these 29 bird species to be training, validated, and tested on.

3.2.12 Spectrogram and MFCC’s

A problem that arises when using convolutional neural networks with sound classification is that CNN’s are predominantly built for the purpose of classifying images. The question is then raised of how do we create an image for the CNN to use, while our data are audio files? One solution is the use of a spectrogram. A spectrogram is a graphical representation of an audio file. This is created by placing the audio signals spectrum of frequencies along an x-axis of time. The amplitude of the signal in respect to time is then plotted as a third dimension with the use of color, typical with darker colors symbolizing a lower amplitude and brighter colors representing higher amplitudes. This image can then be used alongside a CNN to both train the network and classify new sounds that are introduced to the system. The other major problem that must be tackled when using a solution like spectrograms is when background noise is introduced into audio files that are being converted. Since this product will be placed into a live environment, careful consideration must be taken with how to deal with this issue. One such solution to this is having peaks being the focus when a spectrogram is analyzed with the CNN. This solution is used because in most cases sound even when recorded in combination with other background noise will have a similar profile when focusing on the peaks allowing the separation of the desired audio signal and other signals when analyzing. Figure 3-21 shows an example of a spectrogram created from an audio signal.

Another feature extraction option to transform an audio signal into an image for classification is an MFCC. This plots the coefficients of the cepstral representation of the audio signal. The frequency is then equally spaced out along the Mel scale. This scale relates the perceived frequencies of the signal to the actual measured frequency. The idea is to match what a human ear can hear. We used MFCC since they provide enough features and data to accurately classify, while using a lot less memory and data in their representation of the signal

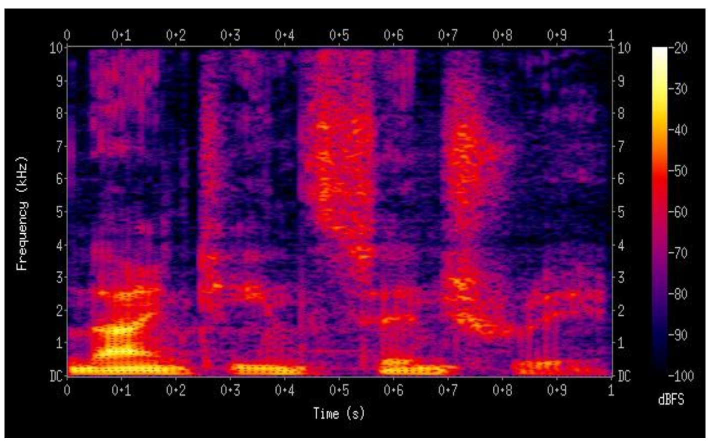


Figure 3-21: *Spectrogram*

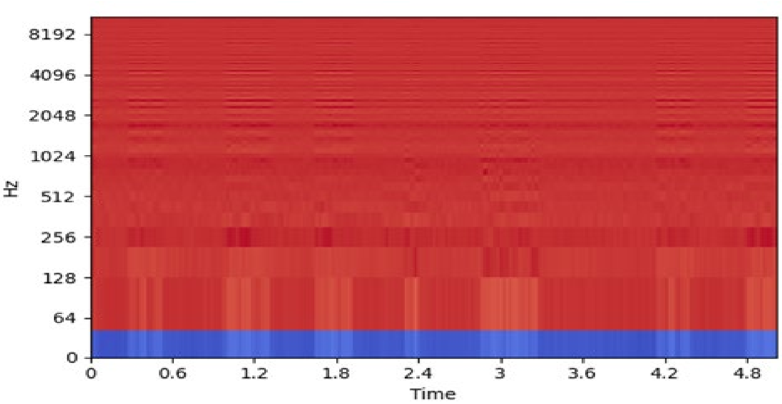


Figure 3-22: *MFCC*

3.2.13 Web Application

The web application will encompass the user's ability to view data that is obtained from the sensors and developed programs. This will allow a user to easily interact with the product without the need for any technical background as all information will be provided directly to them.

3.2.13.1 Frontend Development

The main goal of the frontend is to develop a robust and user-friendly interface that will allow easy access to the provided data by any type of user. Data must be displayed in an easy to view manor. From a technological view several programming languages will be required to accomplish a properly created user interface. This would include HTML which is used as the building blocks of the web application. This will provide the structure of the interface, CSS which will provide the layout and rendering of the web page, and TypeScript which will provide a dynamic website by giving functionality to it.

3.2.13.2 API

Application Programming Interface (API) is how the frontend and backend are connect providing a way of communication. With the use of different format such as JSON data can be transmitted between server and the browser. This allows for real-time data to be sent, received, processed, and viewed by both the server and the user of an application. This allows our product to provide data to the user in real-time.

3.2.13.3 Backend Development

The main goal of the backend is to develop a database that will be able to hold data received from the sensors and in turn organize and relay that data to the client. To build the database MongoDB will be used to provide the structure and relationships of the data received along with information required from the user, such as account information. This will allow users to only view data that is related to their own account and not someone else's. NodeJS with ExpressJS will be used as the server-side scripting language to send data and information from the server to the client.

3.2.14 Multilateration

Multilateration is a technique to determine the location of a subject based on the time of arrival (TOA) of multiple signals that have a known propagation speed. This technique can be used for either navigation or surveillance. If used for navigation, a subject receives signals from multiple known stations that are synchronized. If used for surveillance, the stations receive a signal from the entity to determine its location based on the relative arrival time at each station. A popular use case of multilateration used for navigation is GPS (Global Positioning System), and a couple use cases for surveillance include Seismic Event Location, and sound ranging. For this project we will be using multilateration in a surveillance configuration to determine the location of a sound source.

There are two kinds of multilateration:

(a) Determine the time of transmission (TOT) of synchronized signals being emitted by n + 1 stations with known locations. This is used for navigation.
(b) Create n time difference of arrivals (TDOAs) of a signal at n + 1 stations to determine n coordinates. This is used for surveillance.

As this is a surveillance system, TDOAs must be used because the TOT of the sound is not initially known. Assuming the propagation speed of the signal and the location of each station are known, using one TDOA, a hyperbolic curve of locations for the source can be created. Another TDOA can be used to create a second hyperbolic curve, and the points where the two curves intersect are the sources possible location.

It may make sense to use a GPS module on the microcontroller to maintain up-to-date sensor location information automatically. This would enable the sensors to be mobile and they would require less setup. Sensors could be distributed over an area without keeping track of where each one is. For example, this could be used to deploy a sound event location system over an area by dropping sensors from an airplane. GPS can consume a decent amount of power, and this needs to be understood when choosing the rest of the hardware.

3.2.14.1 Solution Algorithms

There are many well studied methods available to calculate a sources location, and some algorithms have code readily available. There are two types of algorithms: iterative solutions and closed-form solutions.

Iterative solutions were the first methods to be found and are used the most often in real-time systems. They often require a good initial approximation of the target's location and can sometimes take hundreds of iterations. However, iterative solutions have an advantage over closed form solutions because closed form solutions only exist for a specific number of measurements. This allows data from redundant (greater than n + 1) sensors to be used.

Closed-form solutions do not use an initial guess for the targets location but can only use a specific number of measurements. One reason they are less commonly used in real-time systems is because they were not developed until after iterative solutions were already being used in the real world. However, they are used more often in offline studies.

Most real-time systems provide the continuous location of a target and have two stages:

1) A cold-start mode where the targets location is determined from the current measurements only.
2) A steady-state mode where the previous location of the target is used in addition to the current measurements.

Most often, our system may only utilize the cold-start mode because a sound might be received from a source only a single time. Examples include, a single lightning strike, a single gunshot, or any other single sound. This means that it may be more common to determine the sources location using only the cold-start mode. This has the consequence of being more demanding and/or potentially less accurate. Additionally, it may be complex to determine if a measured sound originated from a previously located source. The sound identification portion of this project may be useful for this. If time allows, this should be investigated.

One of the most common iterative approaches is the Gauss-Newton non-linear least squares algorithm. It can be used for both modes and is widely used in actual systems.

The library that we ended up using uses the Nelder-Meade iterative method and is surprisingly accurate. Using example data, the computed origin location was only off by ~5 meters with data from 3 sensors, and with 4 sensors it went down to around 1 meter.

3.2.14.2 Accuracy

The accuracy of multilateration depends on many variables and can be solved for. This could be used to display an adaptive range of locations of the source.

Although not implemented, it may be possible to calculate the accuracy of our design to determine and display the system's current "accurate" service area. This would allow users to determine the size of the area the sound may have originated from.

Overall, the accuracy of our system will depend upon:

- The recording quality of the microphones
- The clock accuracy of each sensor
- Diffraction or reflection of the sound
- How accurate the known sensor locations are
- How accurately the sources signal can be synchronized at each location to determine its relative time if arrival

The accuracy can be calculated using the Cramér-Rao bound.

3.2.14.3 Service Area

The location of the source can only be determined if the source is within the perimeter of the stations, because the only the relative time of arrival is known. If a source is located outside the perimeter then its absolute location cannot be determined, however it can be determined to be closer to certain sensor(s).

It may be useful to display sources as being outside the perimeter and give a relative direction of the sound if possible.

A typical service area using three stations can be seen below:

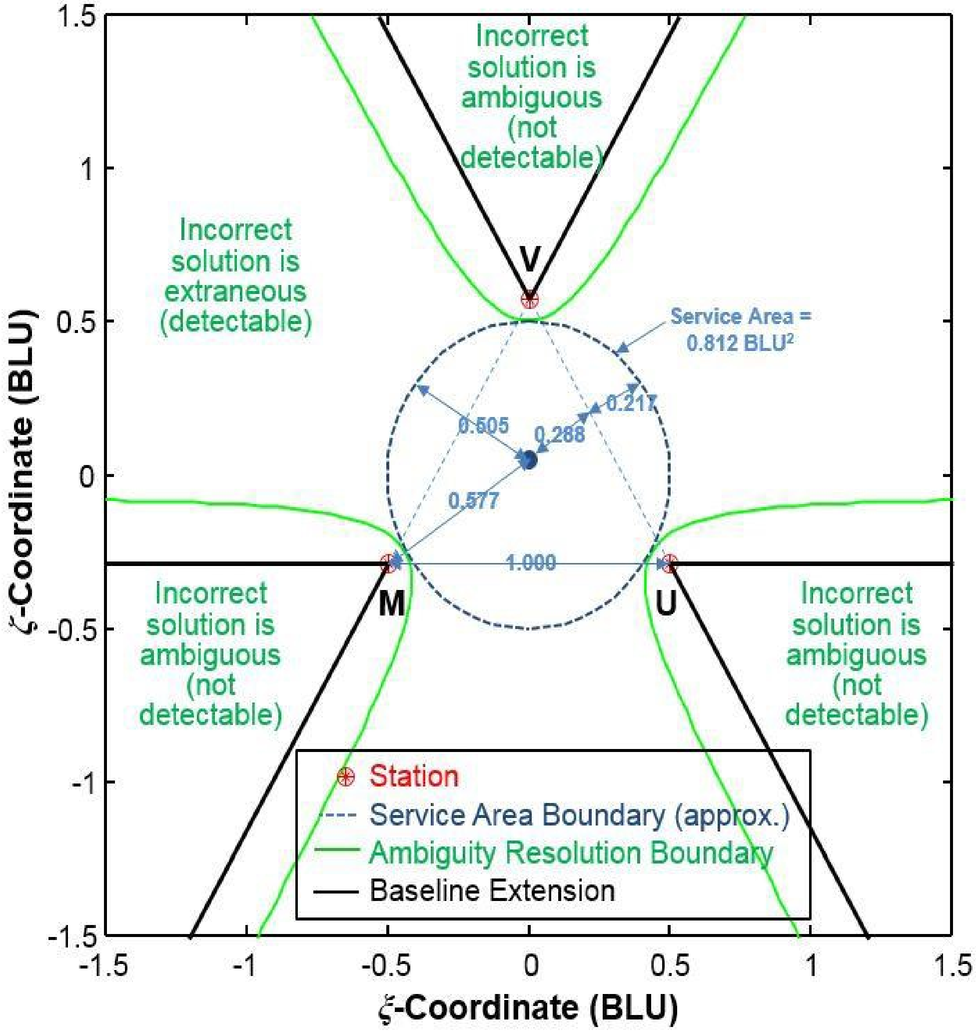


Figure 3-22: Service Area [Taken from Wikipedia using the Creative Commons Attribution-Share Alike 4.0 International license]

3.2.15 Audio Identification Through Fingerprinting

To make use of multilateration, the time of arrival of a sound at each sensor location needs to be found. This first involves identifying that multiple sensors recorded the same sound event, and then it involves accurately determining when exactly the sound arrived at each location.

One solution to recognize shared sounds is to use audio fingerprinting. An audio fingerprint is a content-based compact signature that summarizes an audio recording. A popular audio fingerprinting application is Shazam.

To be able to identify shared sounds well, a design needs to be robust. Ideal fingerprinting systems share the same requirements:

- Identify items regardless of the level of compression and distortion
- Identify small clips of audio
- Properly deal with the lack of synchronization between fingerprints
- Properly deal with degradation due to pitching, equalization, background noise, conversion between digital and analog, coding with various formats
- Efficient in terms of compactness and computation

Audio identification through fingerprinting follows this simplified block diagram:

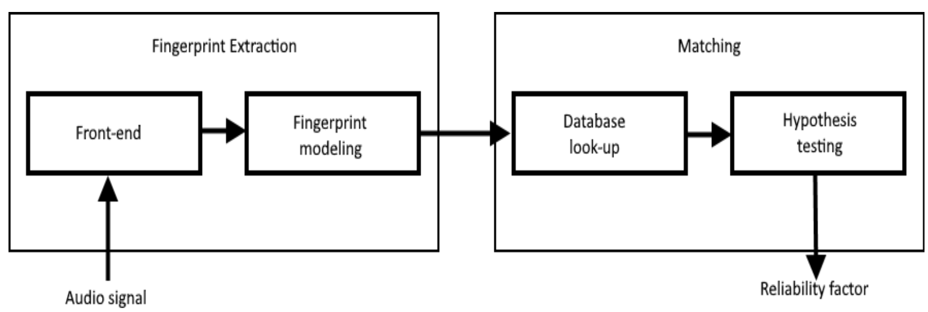


Figure 3-24: *Audio Identification Block Diagram*

Typically, the goal of audio recognition systems is to find metadata that relates to the audio signal, such as song artist and title. The reliability factor is a score of how sure two fingerprints match. Usually, systems only accept scores over a certain threshold.

In our case we would want to identify if there is a shared fingerprint between audio signals and when it was heard at each sensor. Audio is normally split into frames and then fingerprinted and matched, so the metadata we would be storing and searching for should tell us about when each frame was recorded and at what location. Our system would also be slightly different in that we would not need to keep fingerprints for longer than it takes a sound to reach all the sensor. I. e. once the sounds have left the service area and can no longer be recorded by other sensors the fingerprints can be disposed of.

Our sensors would need to surround the UCF campus in the situation that we wanted the entire campus to be our service area, because origin locations can only be determined for sounds originating within the shape made by the sensors. The furthest apart our sensors would be in this case is no larger than a couple of miles. This means that if a sound is loud enough to be detected at both sides of campus, and it originated on one side of campus, then the maximum amount of time it would take to reach a sensor on the other side of campus is two miles divided by the speed of sound in miles per second:

$$\frac{distance}{velocity} = time \Rightarrow \frac{2\ miles}{0.21313\ miles/second} = 9.3839\ seconds$$

***Equation 3-2**: Maximum Time for Sound to be Detected Across Campus*

This means that we would only need to keep fingerprints for at most around 10 seconds, which should help keep the false positive rate low. As a reminder, the accuracy of multilateration depends upon how accurately the TOA can be determined at each location. Since we want to accurately identify when a sound was heard at each sensor it makes sense to raise the frame rate to find more precise time of arrivals. This would also help identify sounds that only last a short amount of time. Raising the framerate means that more fingerprints will be stored in the database, but this should not be a problem because we only need to keep 10 seconds of fingerprints.

Here are a few of opensource audio fingerprinting and identification libraries found with a quick google search:

- Audio fingerprinting and recognition in Python.
  https://github.com/worldveil/dejavu
- Open source audio fingerprinting in .NET. An efficient algorithm for acoustic fingerprinting written purely in C#.
  https://github.com/AddictedCS/soundfingerprinting
- C library for generating audio fingerprints used by AcoustID
  https://github.com/acoustid/chromaprint

One interesting idea to decrease the performance load on the central server is to create the fingerprints on each sensor. This could also potentially lower overall network traffic, because fingerprints are often much smaller than the raw audio data. However, the performance of the microcontrollers will determine if fingerprinting is able to be done on each device in real-time. This also depend on what the microcontroller supports. For example, most MSP430 programming is done in C, but a C fingerprinting library might not be available, and it may take too long to develop ourselves.

We ended up using the Teensy 3.2, which has a significant Audio processing library and has hardware specifically for performing FFTs (Fast Fourier Transform), which means it should be capable of doing it in real-time. However, we ended up only listening for a single tone, as this was much easier to determine an accurate TOA with. It may be possible to train a neural network to determine accurate TOAs from fingerprints. [A Review of Algorithms for Audio Fingerprinting]

3.2.16 Digital Signal Processing

Digital signal processing is the process of analyzing a signal in the digital domain and using various techniques to improve the performance of the signal, observe the signal's behavior and overall characteristics, or to modify the signal to produce more desirable effects. There are many subfields that fall within the digital signal processing umbrella, however the most researched and utilized subfield is audiosignal processing, which is primarily used in applications such as speech processing and recognition. The high volume of research and number of resources available in the audio signal processing subfield is beneficial for developing DiSEL, as this project will be extremely reliant on digital signal processing for continuously monitoring the sounds that it hears and being able to accurately identify when a particular sound should trigger an alert or when a sound will simply be ignored.

When it comes to analyzing incoming signals for the purposes of digital signal processing, the most common method of signal analysis is use of the Fourier transform. However, another method of frequency analysis also exists that may prove to be of use, this method is called the wavelet transform. Both signal analysis methods will be discussed to determine which one will prove to be more beneficial for use in DiSEL.

3.2.16.1 Fourier Transform

The Fourier transform is a mathematical tool that is used to transform a time domain signal into its frequency domain counterpart. Observing these time domain signals in the frequency domain gives the observer more information about the signal than simply viewing it in the time domain. Typically, the most useful information that can be provided from the Fourier transform revolves around observing the characteristics behind how a signal's magnitude frequency response and magnitude response behave.

The relationship between a time-domain signal, g(t), and its frequency-domain counterpart, G(f), obtained using the Fourier transform is shown below:

$$g(t) = \int_{-\infty}^{\infty} G(f) e^{j2\pi f t}\, df \iff G(f) = \int_{-\infty}^{\infty} g(t) e^{-j2\pi f t} dt$$

***Equation 3-3**: Obtaining frequency-domain signal, G(f), using the Fourier transform of g(t)*

These relationships are broken down into a graphical view showing the time-domain pulse signal and the frequency-domain magnitude response in the figure.

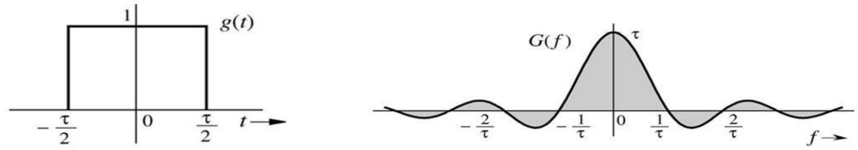


***Figure 3-25**: A time-domain pulse signal (left) and its frequency-domain magnitude response (right) after being obtained using the Fourier transform.*

Fourier transforms are often used in signal processing when designing and characterizing filters, with the low pass filters being used most often for removing high frequency noise especially in sound processing applications.

The biggest downside when it comes to utilizing the Fourier transform for signal processing, is that Fourier transform will only decompose a signal into its frequency components and will lose all information about the signal in time.

3.2.16.2 Short-Time Fourier Transform

One solution to this is the Short-Time Fourier Transform which is useful for non-stationary signals - like many audio signals are. While Short-Time Fourier Transform can provide the same information as a traditional Fourier transform in the event of observing a stationary signal, STFT has the advantage of also providing information on how the frequency components of a signal change over time by computing a series of Fourier transforms in windowed sections of the signal. Due to this effect, STFT has become a powerful general-purpose tool for audio signal processing and defines a useful class of time-frequency distributions which specify complex amplitude versus time and frequency for any signal [Smith].

Once the original signal is divided into multiple windows and the Fourier transform is applied to each of these windowed signals, the information on how the frequency spectrum of the signal changes with time is then plotted into a spectrogram. A spectrogram is a two-dimensional plot with frequency on the y-axis and time on the x-axis.

The drawback to using the Short-Time Fourier Transform is that it can face data resolution problems due to the size of the windows that are used. When a wider window is used, this results in better frequency resolution but a lower time resolution. Whereas, if a narrower window is used, this gives better time resolution but lower frequency resolution. This poses a problem if a signal is varying significantly in frequency over time.

The narrow and wider windows are shown below in figure 3-26, providing a view of the difference in the time and frequency resolutions provided by each window.

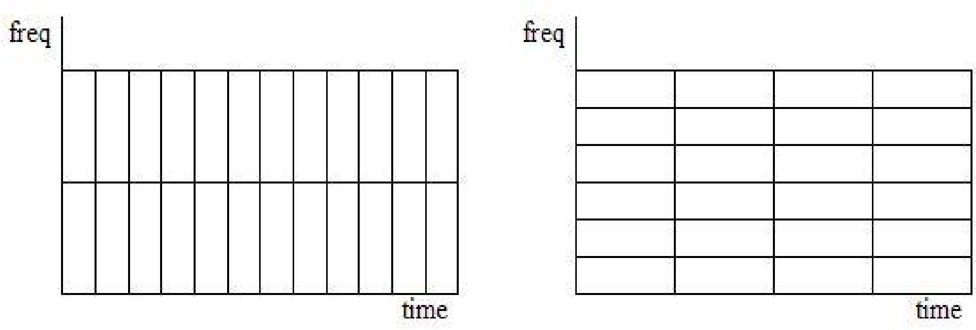


Figure 3-26*: The narrow windows on the left provide a good time resolution, while the wider windows on the right provide a better frequency resolution.*

To combat this resolution problem, another signal analysis method was developed to account for adjusting windows to provide high resolution in both high frequency components and low frequency components.

3.2.16.3 Wavelet Transform

The Wavelet Transform provides characteristic information of a signal for both frequency and time like the short-term frequency transform, however what sets it apart is its ability to adjust its windowing to maintain optimal resolution for high frequency and low frequency variations in time.

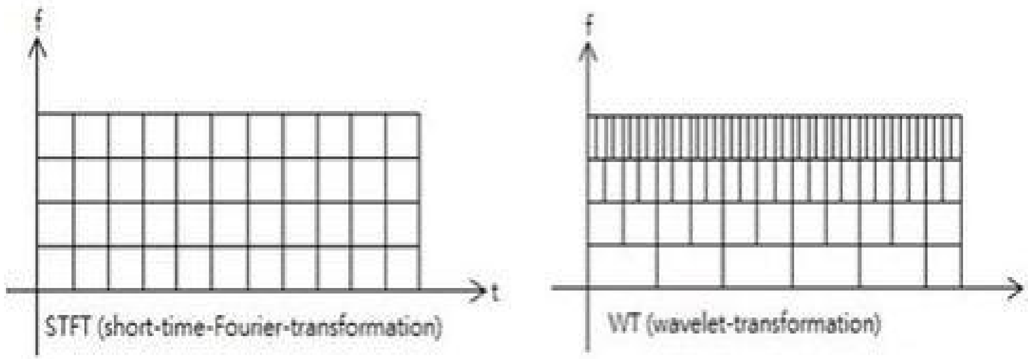


***Figure 3-27**: Comparing how the windows are developed for short-time Fourier transforms versus wavelet transforms.*

In the figure above comparing STFT versus wavelet transforms, it is clear to see that the resolution for time is greatly increased for higher frequencies when compared to lower frequencies, where the resolution for frequency is much higher instead.

The following table provides a side-by-side comparison of the wavelet resolutions, showing the temporal and frequency resolution responses as the frequency is either in a high or low state.

Wavelet Resolutions Depending on Frequency Content		
Frequency, Hz	**Temporal Resolution**	**Frequency Resolution**
High	Increased	Decreased
Low	Decreased	Increased

***Table 3-11**: Comparing how the resolutions of time and frequency are affected in the Wavelet transformation based on whether the frequency components are high or low.*

Just as the Fourier transform is derived using Fourier series, the wavelet transform can be derived in an integral form from a wavelet series which is constructed using a Hilbert basis.

The integral wavelet transform is defined as:

$$X(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \Psi\left(\frac{t-b}{a}\right) x(t)\ dt$$

***Equation 3-4**: Integral wavelet transform, where it takes a (scaling factor) and b (time) as parameters.*

In addition to providing information on both the frequency and temporal characteristics of a signal, wavelet transforms are capable of being used for data compression (particularly image and audio compression), denoising a signal, and have reduced computational complexity for specific frequencies when compared to Fourier transforms.

Transform	**Equation**	**Inputs**
Fourier	$G(f) = \int_{-\infty}^{\infty} g(t) e^{-j2\pi f t} dt$	t, time f, frequency
Wavelet	$X(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \Psi\left(\frac{t-b}{a}\right) x(t)\ dt$	a, scaling b, time

***Table 3-12**: Wavelet Transform vs Fourier Transform*

3.2.17 Distributed Computing

Distributed computing is the study of distributed systems. A distributed system is a system whose components are spread across multiple networked computers. A few examples of distributed systems include telephone networks, the World Wide Web, and cloud computing. This project is an example of a distributed system because multiple sensors (microcontrollers) and potentially a central server will work together over a network. Three formal requirements of a distributed system include: a lack of a global clock, concurrency of components, and independent failure of components. [Tanenbaum] Each microcontroller will have its own clock, and all of them will be concurrently recording and transmitting sound data, and it is possible for a part of this system to fail without the entire system failing. Such as if a redundant sensor fails, or even if a needed sensor fails while the website still loads.

The study of distributed computing also covers the use of distributed systems to solve computational problems. Some famous examples include the Great Internet Mersenne Prime Search, and Einstein@home.

Parallel computing is not the same thing as distributed computing. Parallel systems share memory, and distributed systems use message passing.

Distributed computing comes with issues and design considerations that need to be understood. The next two sections will cover them in detail.

3.2.18 Centralized Computing

Centralized computing is a type of architecture where most of the processing is done by a central server. This kind of design has advantages over decentralized designs, but there are also some drawbacks. A couple advantages include greater security because only one location needs to be protected, and it avoids the major complexity associated with decentralized designs. The biggest drawback is total reliance on the central computer. If the central server fails, then the entire system fails. The overall system performance also heavily relies on the performance of the central server. It may be slow compared to a distributed solution because it is not able to split up work or take advantage of locality. Another thing to keep in mind is the added cost of a central server.

For example, if this project were to use a single central server to do most of work such as hosting the web server, receiving, and processing all the sound data from each sensor, then all the network traffic for the system would run through it. We ended up using several servers to split this work up. One for audio processing, one for classification, and one for the web server. This may become an issue if the number of users requesting data from the webserver becomes too great, or if there are too many sensors. A possible centralized architecture that has three sensors and four clients can be seen below:

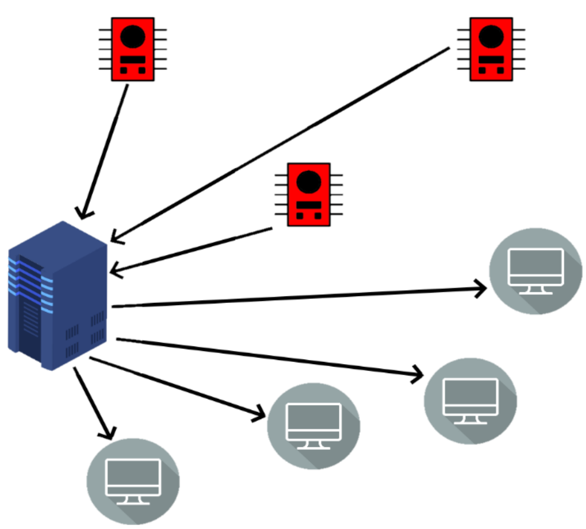


Figure 3-28: *Centralized Architecture*

3.2.19 Decentralized Computing

A decentralized system is a distributed system. In a distributed system, work is split up across multiple computers, but decisions may still be centralized and use complete system knowledge. A decentralized system on the other hand has no single computer that has complete system knowledge.

One advantage of making this project decentralized is that the hardware for this system could be sold so that people could set up their own sensor networks without having to purchase or set up a central server. Additionally, it could be designed so that each new sensor contributes to a global sensor network, which would allow for a larger service area.

A couple of important algorithmic issues include finding other sensors on the network and determining which sensors to compare sound data to. If a sound is heard by a sensor it should only be possible to detect it from so far away. So only comparing sound data with some of the closest other sensors would make sense, rather than making each microcontroller receive and compare data with every other sensor on the network. Sensors only need to send data to the server in a centralized system, but in a decentralized system they need to figure out what each other's networks address is without initially knowing. All that is needed in the centralized system is a domain name that points to a dynamic IP address or a static IP address. One solution for the decentralized system is to use elected controllers that maintain locations and addresses about nearby controllers. The elected controllers could themselves use dynamic IP addressing so that when a sensor gets added to the network, it checks a list of domains that point to elected controllers. The elected controllers could then tell the new sensor which elected controller(s) to communicate with to find the locations and IP addresses of the nearest sensors around it.

The two problems above are already quite complicated, and they are only the basic first steps. There are many other problems, such as the problem of election, how to distribute the multilateration algorithm, and how to host the webserver using only a series of microcontrollers. For the purposes of this project, a centralized system is the easiest to implement, however it may be interesting to try and implement a decentralized version. A possible decentralized architecture can be seen below:

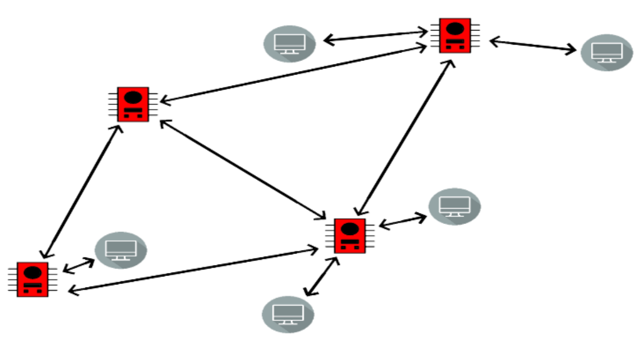


Figure 3-29: *Decentralized Architecture*

3.2.20 Networking

The microcontrollers will require some form of networking to send data to the central server. There are a few different options available including, Wi-Fi, Ethernet, and cellular networks. These options come with different pros and cons, and they should be taken into consideration while determining what we want the system to be capable of, and what it is given requirements are.

Ethernet is the fastest and most reliable of the three available options, however it is not mobile at all. A building with it installed will need to be nearby to use it. Ethernet cable would need to be ran from the building to each sensor, which would add to the installation cost and time. However, many microcontrollers have ethernet hardware built-in, and it is more commonly supported than Wi-Fi.

Cellular networks have the opposite properties of ethernet. They are extremely mobile, but they can be potentially unreliable and slow depending on the signal strength and network technology being used. Additionally, it usually costs money to use cell networks, and the microcontroller would need additional hardware, which is not as common as ethernet and Wi-Fi hardware.

Wi-Fi is kind of a middle ground between ethernet and cellular data. It is faster, more reliable, and more common than cellular technology, but it is slower and less reliable than ethernet.

Wi-Fi is the best option for our project because it meets our requirements of mobility, cost, and performance. Wi-Fi is readily available around UCF, so we would be able to test freely and easily around the entire campus. Additionally, it is possible to configure most cellphones as mobile Wi-Fi hotspots so it should be possible to test where the Wi-Fi is weak or unavailable. This could also be used to make testing easier because we can configure static SSIDs and passwords for on our phones, that the microcontrollers can easily look for without having to change their configuration. This is what we ended up doing.

If our system were to be provided as a service or sold, it would make sense to provide multiple technologies on each device. If we wanted to target a large service area where Wi-Fi and ethernet are not readily available, it would make sense to use a cell network. If cell networks were not available in our targeted service area, something like satellite networking would need to be investigated. If the required reliability and performance of each sensor were especially important then it may be necessary to investigate installing a wired internet connection at each sensor location.

The central server also uses networking, but it makes sense to use Ethernet or Wi-Fi as it should be in a stable environment. Also, because the entire system depends on the reliability and performance of the central server, ethernet should be used because it is the fastest and most reliable option. Although it takes time and money to install ethernet, it is not really a problem because it only needs to be done once for the central server. Additionally, if we use a service such as AWS for the central server then we will not have to configure its networking or hardware. We ended up using Digital Ocean to host all our servers, so networking was not an issue.

We used TCP from our sensors to the central server along with COBS (Consistent Overhead Byte Stuffing) encoding. COBS made it easy to parse data as we were able to delimit each packet with a zero byte. This made it easier to parse, because each packet is a different length depending on the data being sent.

3.2.21 Microcontroller Audio Recording and Encoding

Each sensor records audio using a microphone, and then digitally encodes it. In our case audio on the Teensy is encoded into PCM (Pulse Code-Modulation). A block-diagram representing each stage of the audio processing needed to be done on each microcontroller can be seen below:

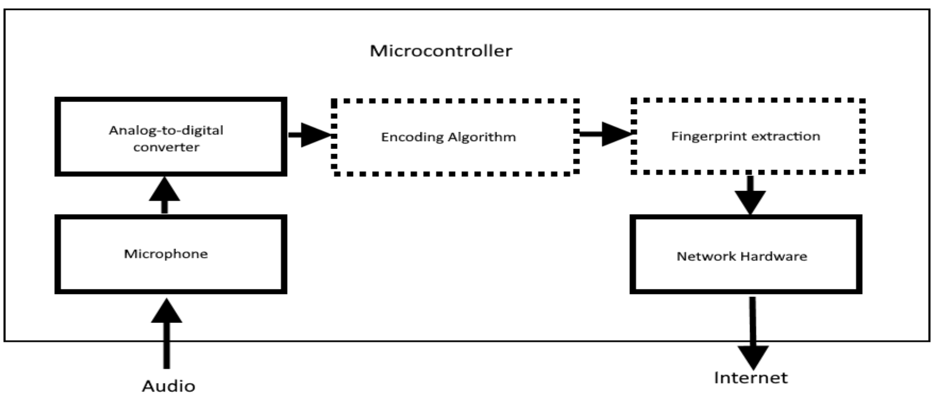


***Figure 3-30**: Microcontroller Audio Recording and Encoding Diagram*

There are many different microcontroller microphones available, but each of them has different power consumption requirements, recording quality, cost, portability, and compatibility. These properties will need to be kept in mind when deciding which microcontroller and microphone to use.

Sound is an analog signal, and microcontrollers operate on digital data, so analog data from the microphone will need to be converted into digital data to be used by the microcontroller. This is typically done with an analog-to-digital converter, which are available on many different microcontrollers including MSP430s. Converting to digital involves quantifying the analog input, which introduces a small amount of noise. Additionally, an ADC cannot convert data continuously, so it must sample the input, which results in a limited bandwidth. Although the audio fingerprinting technology should be robust enough to handle these problems, it should still be kept in mind when deciding what sampling rate and resolution to use.

After conversion to digital, the data is encoded using Opus, although MP3 could also be used. If the networks data rate was an issue, an encoding algorithm that compresses the amount of digital data representing the audio may need to be used. Opus is extremely efficient, so it is probably optimal in that situation. However, not every algorithm is created equally. The computational complexity and distortion due to compression varies between different encoding algorithms. Some algorithms are free and open source, while others are not. If we had used fingerprinting, then it would have to hand the distorting due to encoding/decoding. If it is too great sound recognition performance may drop. All these factors will need to be considered when deciding which microcontroller to use, and which algorithms to use.

It may be possible to do fingerprint extraction on the Teensy itself.

3.3 Strategic Components and Part Selections

The section above goes over how things work and could be done in theory. This section goes over hardware options available that could be used to implement the ideas above. Concerns related to each topic will be considered for each individual piece of hardware, as well as concerns as to how the hardware has to function as a complete system.

An important requirement to keep in mind is that our final hardware sensor design uses one PCB. No breadboards or loose wires allowed.

3.3.1 Microcontrollers/Microcomputers

This section goes over various development boards available today. An important requirement for our chosen microcontroller is that we can design a single board using the same components as the development board. If a schematic and a bill of materials (BOM) are available, then it is much easier to create our own.

Microcontrollers are typically small computers on a single integrated circuit chip. Microcontrollers are typically programmed at a bare metal level. A microcomputer is different from a microcontroller in that it has hardware spread out over the entire board and can typically run actual operating systems.

The microcontrollers discussed in this section have development kits that can make them easy to prototype with.

Additionally, our chosen microcontroller needs to have an available bootloader. Initially we went with the Teensy 4.0, but because it’s bootloader is still not available, we had to switch to the Teensy 3.2.

3.3.1.1 MSP430 Microcontrollers

Of the available options, we were initially the most familiar with MSP430 microcontrollers. MSP430s are considerably basic but have some features that make them great to work with.

Their most important feature is their outstanding support from Texas Instruments.

They have fantastic documentation from TI, a variety of configurations, and an extensive set of add-on modules that really expand what they are capable of. Although most evaluation boards from TI only have basic features, they are compatible with many different add-on boards from third parties or even TI themselves. The documentation even features board schematics as well as BOMs (bill of materials). MSP430s are typically programmed at the register level in C. This means that they are not capable of using large libraries for things such as fingerprinting. However, there may even be a module that could make fingerprint extraction on the device itself possible. TI's Low-Energy Accelerator module is capable of complex vector math processing such as FFT's, which are used during fingerprint extraction. It may make sense to do the fingerprinting on a central server first, and then later implement it on each sensor.

MSP430s are also unbelievably cheap, and often exceptionally low power, which makes solar power achievable. However, MSP430s may only be capable of recording and transmitting audio at an insufficient rate.

MSP430 launchpads vary in price from $10 to $18.

3.3.1.2 Raspberry Pi Microcomputers

Raspberry Pi's are the most famous microcomputers today. Raspberry Pi's can run a variety of operating systems including Android, FreeBSD, Linux, Plan9, and Windows 10 Arm64.

There are several versions available that feature different amounts of memory capacity, processing power, and peripheral support. Many addon modules should be able to be used if needed.

A Raspberry Pi would be less complex to set up because it can run a wide variety of software and programs. However, its hardware is much more power hungry than an MSP430, and it may be harder/more expensive to power with solar panels.

One issue is that the documentation and support is not as good as the MSP430.

The price of Raspberry Pi's ranges from $10 to $55.

3.3.1.3 BeagleBone Microcomputers

BeagleBone's are another single-board computer like Raspberry Pi's but are produced by Texas Instruments. They are open source, designed to be low power, and have all the functionality of a basic computer. They can run Linux, FreeBSD, OpenBSD, RISC OS (Operating System), Symbian, and Android.

These are a compromise between a Raspberry Pi and an MSP430, because they have most of the features that make Pi's attractive as well as most of the features that make MSP430s attractive.

Although the documentation and support are better than the Raspberry Pi's, they are still not as good as the MSP430 launch pads. BeagleBone's are more capable than MSP430s, but they also use more power.

BeagleBone boards are the most expensive of the options discussed so far. Their price ranges from $25 to $99.

3.3.1.4 Teensy Microcontrollers

Teensy microcontrollers are designed to be small and fast. They are so small and simple that they lack a lot of features that other development boards have. However, this also makes them extremely cheap relative to how fast their processors are. Creating a "production" board using the components from a Teensy board was somewhat easy because of how simple the development board design is.

Teensy boards use 32-bit ARM Cortex-M processors. According to the designer, the Teensy 4.0 features the fastest microcontroller available today.

There is not a lot of documentation for these boards, partially because they are so simple. However, the documentation lacks important things such as official CAD (Computer Aided Design) schematics.

Additionally, the creator of the Teensy boards has made extensive tutorials, addon boards, and programming libraries available online. This board can be programmed with either C or C++, and it compatible with most Arduino software.

Teensy boards vary in price from $12 to $34.

3.3.1.5 Microcontroller Comparison

The following chart compares the different microcontrollers being considered for our project.

Device	Typical Power Usage	Software Support	Hardware Support	Processing Capability	Cost
TI MSP430FR 5994	- 1.9 mA @ 16 MHz - 500 nA LPM3 5V	- C programmable - TI libraries - Some other C libraries	- TI BoosterPacks - UART, SPI, I2C compatible devices - SD card support	- 16-bit RISC Architecture up to 16 MHz - Low-Energy Accelerator	$17.99
Raspberry Pi Zero W	- 170 mA 5V	- Linux - Any software compatible with ARMv6	- Built-in Wi-Fi - SD card support - Full USB support	32-bit ARMv6 at 1 GHz	$10
BeagleBone Black	- 210-460 mA 5V	- Linux - Any software compatible with ARM	- Full USB support - BeagleBo ne capes	32-bit ARM Cortex-A8 up to 1 GHz	$45
Teensy 3.2	- 100 mA @ 600 MHz - Coin cell shutoff mode 5V or 3.3V	- C/C++ Programmable with their libraries - Teensy libraries - Arduino libraries	Teensy shields UART, SPI, I2C compatible devices SD card support	32-bit ARM Cortex M7 up to 600 MHz	$18.00

***Table 3-13**: Microcontroller Comparison*

Selected Device: *Teensy 3.2*

Due to our project needs and the size and power goals we have set. We have determined to use the Teensy 3.2 microcontroller. The Teensy 3.2 is quite powerful, but not too power hungry, which will benefit our device as they are tobe battery operated devices rather than having a direct line of power, and it features an extremely plain design that will be easy to work with.

3.3.2 Networking Hardware

Some of the development boards discussed above feature built-in networking hardware, but others will need an add-on module. Setting up and using the modules below will take more time and research than using a board with built-in networking. Additionally, both boards are designed to take commands through SPI, I2C, or UART.

3.3.2.1 ESP32

The ESP32 wireless module is a fully-fledged microcontroller with built-in Wi-Fi and Bluetooth. It is a versatile and robust device that features ultra-low power technology. It is powerful enough that it may be interesting to consider using it as the actual microcontroller for this project. Additionally, it is powerful enough to use as a coprocessor alongside the actual microcontroller. It contains two 32-bit low- power processors, that can be turned on and off. This allows it to save power by turning off components that are not in use. It may be possible to use the low power processor to monitor the sensors on our device, and then wake the other processors when they are needed to do intense processing. It is also possible to use this device for Ethernet.

The documentation for this module can be lacking, and there are still bugs being discovered. The ESP32 development kit costs around $10. We actually encountered a problem with SPI between the Teensy and ESP32 that was the result of a bug in the ESP32 libraries.

3.3.2.2 SimpleLink™ Wi-Fi CC31XX BoosterPack™

This series of modules from TI are designed for use with any microcontroller. They can be easily connected to any of TI's MCU Launchpad kits. The documentation for this module is particularly good. However, it is not used as often as the ESP32. The CC31XX BoosterPack™ costs around $30.

3.3.2.3 ESP8266

The ESP8266 is another low-cost Wi-Fi chip from Espressif Systems. This device is what made Espressif Wi-Fi modules popular. It became popular because it has very few external components and is very inexpensive. Just like the ESP32 it is also able to connect to Wi-Fi without an additional microcontroller because it is a fully fledged microcontroller. Although the ESP32 is newer, many people still use the ESP8266 today.

3.3.2.4 Networking Hardware Comparison

The following chart compares the networking hardware being considered for this project. The chart uses power usage, hardware compatibility, networking capability, and cost as its comparison factors. Since our project has data transfer has a central requirement, the networking capability was considered the most crucial factor to determine the device to be used for the project.

Device	Typical Power Usage	Hardware Compatibility	Networking Capability	Cost
ESP32-WROOM-32D	200 mA	Control via UART, SPI, I2C, or I2S	802.11 b/g/n 150 Mb/s 20 dBm output power	$2.50
SimpleLink Wi-Fi CC3120 BoosterPack	200 mA	UART or SPI	802.11 b/g/n 16 Mb/s 18 dBm output power	$29.99
ESP8266	80 mA 2.5 – 3.6V	Control via UART, SPI, I2C, or I2S	802.11 b/g/n Wi-Fi 10 Mb/s 20 dBm output power	$3.00

Table 3-14: *Networking Hardware Comparision*

Selected Device: *ESP32-WROOM-32D*

The ESP32 is the newest Wi-Fi module from Espressif Systems, has many excellent features, and is extremely cheap. Additionally, the Teensy 4.0 has a breakout board with a spot for an ESP32.

3.3.3 Microphones

None of the discussed microcontrollers have built in audio recording capabilities. There are quite a few low-quality microphones available for microphones, but we want to be able to pick up faint sounds. The microcomputers have built in USB ports and good driver support so they should be able to support USB microphones as well as microphones connected through GPIO.

3.3.3.1 USB Microphones

To use a USB microphone with the Raspberry Pi or BeagleBone kits, a compatible sound card is needed. There are a wide variety of sound cards available as well as many different USB microphones. Using this option may result in the best audio quality. However, it may be difficult to design a single board using a proprietary sound card and microphone.

Prices vary for each piece of equipment.

3.3.3.2 TI Audio Signal Processing BoosterPack

This BoosterPack features a built-in microphone and speaker and can be connected easily to any TI LaunchPad MCU. It is capable of sampling rates of up to 20 kHz. It is also capable of automatically switching to microphones plugged into the included aux jack.

This module costs around $30.

3.3.3.3 Teensy Audio Shield

This is adapter features high quality 16-bit, 44.1 kHz sample rates. It also has DSP instructions and is fast enough for real-time FFTs. There are no official Eagle schematics available for this development board, but there is a regular PDF schematic. It is designed to stack on top of the Teensy microcontrollers, and there are C and C++ libraries, as well as tutorials, and many example projects available online.

This module costs around $14.

3.3.3.4 Microphone Comparison

The following chart compares different microphone models being considered for our project. The chart uses power usage, hardware compatibility, recording capability, and cost as its comparison factors.

Device	Typical Power Usage	Hardware Compatibility	Recording Capability	Cost
SunFounder USB Mini Microphone	2A @ 5V	USB Only	Omnidirectional	$6.99
TI Audio Signal Processing BoosterPack	@ 3.3V	SPI	12-bit ADC 20 kHz sample rate	$29.99
Teensy Audio Shield	58 mA @ 3.3V	I2C and I2S	16-bit ADC 44.1 kHz sample rate	$13.75

Table 3-15: *Microphone Comparison*

Selected Device: *Teensy Audio Shield*

The Teensy Audio Shield is designed to interface easily with the Teensy 4.0 and features high quality recording specifications. It is also quite cheap and has available schematics and software examples.

3.3.4 Camera Hardware

An optional camera can be used to capture video after a sound event has occurred. After a sound event, the sensors will collect data from the cameras and send a live video feed to the user interface. This allows the user to see what is happening and monitor the situation. The desired camera will provide a high-quality picture for a decent price. Their also we be a possibility depending on the capability of the camera and the sharpness of the image to provide additional details to the user with the use of machine learning and the images that are captured. This however will be considered a stretch goal for our project, as this will be a feature that will require the camera to obtain good enough photos that we do not know yet if it will be capable at the distances, we hope to be able to reach.

Cameras can be purchased with just the image sensor itself, the image sensor and a breakout board, or the image sensor and a microcontroller. The cameras are also offered with a variety of lenses.

3.3.4.1 Micron

Micron produces CMOS image sensors that use a Bayer color filter arrangement. Their image sensors include a microcontroller and a sophisticated image flow processor (IFP) with a real time JPEG encoder. Their sensors also come with a programmable general purpose I/O module (GPIO). Their features include the following:

- Low light performance
- Low power consumption
- Electronic Rolling Shutter (ERS)
- Automatic image correction and enhancement
- Two wire serial interface
- 10-bit output
- Programable I/O slew rate
- Support for external auto focus, optical zoom, and mechanical shutter.
- Xenon and LED flash support with fast exposure adaptation

3.3.4.2 OmniVision

OmniVision produces CMOS images sensors that use a Bayer array color filter. OmniVision image sensors include microcontroller and a compression engine for increased processing power. Their features include the following:

- High sensitivity for low light
- Low operating voltage
- A programable serial port
- 8 bit and 10-bit output
- Supports LED and flash strobe mode

- Supports scaling
- Supports compression
- Variable frame rate control
- Automatic exposure control
- Automatic gain control
- Automatic white balance
- Automatic band filter
- Video or snapshot operation

3.3.4.3 Camera Comparison

The following chart compares the different camera models being considered for use with our prototype. The chart uses megapixels, resolution, pixel size, and cost as comparison factors.

Device	Megapixels	Resolution	Pixel Size	Cost
Arducam Camera Shield OV2640	2	1600x1200	2.8µm x 2.8µm	$25.99
Waveshare OV9655	1.3	1280x1024	3.18 µm x 3.18 µm	$6.99
Arducam MT9D111	2	1600x1200	2.8µm x 2.8µm	$31.25
Yosoo OV7670	0.3	640×480	3.6µm x 3.6µm	$11.99
Arducam MT9M001	1.3	1280×1024	5.2µm x 5.2µm	$54.99
OV7670 by Atomic Market	0.3	640×480	3.6µm x 3.6µm	$10.99
SainSmart Surveillance OV5647	5	2592×1944	1.4µm x 1.4µm	$26.99
Arducam Mini Shield OV5642	5	2592×1944	1.4µm x 1.4µm	$39.99
ESP32-CAM with OV2640	2	1600×1200	2.8µm x 2.8µm	$8.60

Table 3-16*: Camera Comparison*

Selected Device: *OV2640*

The OV260 is a decent image sensor with a high resolution, large pixel size and reasonable price. It is also known to interface well with the ESP32.

3.3.5 Batteries

As mentioned in the research section, there are three types of rechargeable battery chemistries that have been considered: Lithium-Ion, Lithium Iron Phosphate, and Nickel-Metal-Hydride. Of the three battery chemistries, it was determined that a Lithium-Ion battery is best suited for DiSEL. This is because Lithium-Ion batteries can survive a high number of charge cycles, they have a high charge density, they have a high voltage, and they are reasonably priced. However, there are many options for Lithium-Ion batteries. These options will be considered and compared in this section.

3.3.5.1 Battery Comparison

The following table compares different Lithium-Ion batteries based on their charge capacity, physical size, and cost.

Name	Charge Capacity	Battery Size	Cost
Xstar	800 mAh	14500	$4.45
Fenix ARB-L14	800 mAh	14500	$6.95
Klarus	800 mAh	14500	$5.95
UltraFire UF AA	900 mAh	14500	$4.50
Samsung 25R INR	2500 mAh	18650	$3.95
Samsung 30Q INR	3000 mAh	18650	$4.95
Panasonic NCR	3400 mAh	18650	$6.50
Nitecore NL1834	3400 mAh	18650	$17.95
LG MJ1 INR	3500 mAh	18650	$5.35
Samsung 40T INR	4000 mAh	21700	$7.95
Nitecore NL2140	4000 mAh	21700	$14.95
Olight ORB-217C50	5000 mAh	21700	$26.95
Fenix ARB-L21	5000 mAh2	21700	$23.95

Table 3-17: *Battery Selection Comparison*

Selected Battery: *Samsung 25R INR*

This battery was chosen because it has a balance of charge capacity, physical size, and cost. Samsung is a well-known and reliable brand. Their batteries are guaranteed to be high quality and produce numbers close to their ratings. The reason this battery is so inexpensive compared to other brands such as Fenix or Nitecore is because Samsung mass produces the 18650 battery to use in battery packs such as a laptop battery. Fenix and Nitecore are high quality battery manufacturers that only produce single cell batteries, making their cost higher. The Samsung 25R INR is on the lower side in terms of charge capacity. However, 2500 mAh is still a lot of battery life and is more than enough for DiSEL applications.

3.3.6 Central Server

The requirements for the central server are much more relaxed and do not need to be considered as carefully. The main two requirements for the central server are that it has enough performance and is reliable. There are two options:

- Host the server ourselves using one of our own computers
- Use a service that provides us with the needed infrastructure.

Using one of our own computers as the server allows us to have full control over it and configure it any way that we want to. However, we may not be able to provide the same level of reliability and performance that something like AWS or DigitalOcean provided. Additionally, these services can be very inexpensive, and may even be free to us as students.

3.3.6.2 Personal Computer

Using one of our own personal computers would allow us to have full control over the central server. It is a free option and will need to be used for testing anyway. However, a personal computer may not be a robust solution. Power outages could affect our service as well as annoying things like Windows Update. This option would be ideal for local rapid testing. Additionally, if this were a real product it would be quite hard to scale our own system.

3.3.6.3 Heroku

Heroku is a cloud application platform that makes it easy to deploy web applications. It supports many different programming languages and can be scaled with different service plans. There is a free option, and we can get account credit with the GitHub student pack. Its documentation and support are quite excellent and features many online tutorials. However, Heroku does not provide as much control as other options. It is possible to deploy the application automatically with pushes to a GitHub repository. The only control over the server is with a basic web and command line interface (CLI).

3.3.6.4 DigitalOcean

DigitalOcean is another cloud infrastructure platform like Heroku. Just as with Heroku there are free options available from DigitalOcean, as well as student credit through the GitHub student pack. DigitalOcean is typically used by SSHing into a server and then configuring everything manually. Although things are configured automatically like Heroku, DigitalOcean provides much more control over every detail of the central server. DigitalOcean also has additional service plans that would allow us to scale the central server's performance if we needed it.

3.3.6.5 AWS

Amazon Web Services (AWS) is the most popular web service platform today. There are many ways to configure it and it offers a variety of performance and scaling options. There are so many options in fact, that to use AWS well it practically requires training. The GitHub student pack can provide us with some AWS credit, but its costs can quickly add up, and are hard to manage. This conclusion is based on experience with this system from group members, as the learning curve presented by this service to create a functioning result is high. Though the challenge and options may be beneficial in some respects, the overall cost is too high when other service options can be presented with similar options and benefits without the presentation of a learning curve that is this high.

3.3.6.6 Central Server Comparison

The table below provides a comparison of central server solutions being considered for our project. This table uses the configuration options provided by the service, the scaling options, their performance, and their cost as factors used to compare each one of them. A decision is then presented for which service we have decided to go with based on the factors presented both from the above discussion and the table below.

Service	Configuration Options	Scaling Options	Performance	Cost
Personal Computer	Full software and hardware control	Very minimal Possible to buy a new computer	Possibly reliable Might be fast enough	Free
Heroku	Basic web interface / CLI	Pay for a better service plan	Extremely reliable and fast	Free/Paid
DigitalOcean	Full software control	Pay for a better service plan	Extremely reliable and fast	Paid
AWS	Full software control Advanced access control	Pay for a better service plan	Extremely reliable and fast	Paid

Table 3-18: *Central Server Comparison*

Selected Service: *DigitalOcean*

DigitalOcean will provide us with the performance and configuration options that the central server will need to be reliable. Although it takes more configuration than Heroku, it will allow us to have better control over the software running on the server. We can get credit through the GitHub student pack. Additionally, it may be useful to do testing on a local server before deploying DigitalOcean because it takes less time to test locally.

3.3.7 GPS modules

It was initially thought that a GPS module would not be required, but they are actually very important for syncing the clocks on each sensor. There are a lot of options when it comes to GPS hardware. The size, update rate, power requirements, number of channels, antennas, and accuracy. Preferably we want our design to be as small and low power as possible. However, we also want to make sure that the location is maintained quickly and accurately enough for multilateration to work properly. Additionally, we want to make sure that the cost of the GPS module makes sense.

3.3.7.1 NEO-6

The NEO-6 family of GPS receivers from ublox are versatile, cost effective, compact, and power efficient. They use extraordinarily little power and take up little space.

Receivers support the UART, USB, I2C, and SPI interfaces. A development kit only costs around $15.

3.3.7.2 NEO-M9

This is a newer series of GPS receivers from ublox with many upgraded features. These features are quite neat but may not be necessary for what we need. The GPS requirements of our system are quite minimal. Our sensors will be in outdoor areas and will not be moving. This means that we only need to update our location occasionally. This should allow us to save the most power. However, this also means that even if a GPS sensor uses a lot of power it will be off for most of the time, so its overall effect on power requirements will be minimal.

A development kit costs around $70.

3.3.7.3 Adafruit Ultimate GPS Breakout Board

This development kit from Adafruit has good specs and should be capable of what we need for this project. Again, the GPS requirements for this project are quite low, and most of the above modules are quite similar. They all have subtle differences, but their price is what distinguishes them the most. This board features a connector for an external antenna to make it extremely accurate and precise. Adafruit provides schematics and tutorials to make using this board even easier.

A development kit costs around $40.

3.3.7.4 Adafruit Mini GPS

The Adafruit Mini GPS is like the Adafruit Ultimate GPS breakout board, but it is designed to be as small as possible. It lacks the ability to connect an external antenna, but this should be ok because our GPS performance requirements are not that extreme. It can be interfaced with either UART or I2C. It also features a low-power mode and a standby mode with a WAKE pin. CAD schematic diagrams are available, and there are even EAGLE schematics.

The development kit costs around $30.

Device	Typical Power Usage	Hardware Compatibility	GPS Capability	Cost
NEO-6M	37 mA @ 3.3V	UART, USB, SPI, I2C	-162 dBm sensitivity	$16.99
NEO-M9	36 mA @ 3.3V	UART, USB, SPI, I2C	-167 dBm sensitivity	$64.95
Adafruit Ultimate GPS	25 mA @ 3.3-5V	USB	-165 dBm sensitivity	$39.95
Adafruit Mini GPS	30 mA @ 5V	I2C, UART	-165 dBm sensitivity	$29.95

***Table 3-19**: GPS Module Comparison*

Selected Device: *NEO-6M*

The NEO-6M meets all our GPS performance requirements while being extremely small. It has excellent product support and seems to be a popular choice in the maker community.

3.3.8 Lithium-Ion Charging IC

These charging ICs are specifically designed to charge Lithium-Ion batteries. Various ICs are compared in the following table.

Device	Charging Voltage (V)	Input Voltage (V)	Charging Current (A)	Cost
TP4056	4.20	4-8	.500, 1.000	Average
MCP73831	4.20, 4.35, 4.40, 4.50	4-6	.100, .505	Least Expensive
MAX8900A	4.20	3.4-6.3	.101, .500, 1.200	Most Expensive

***Table 3-20**: Charging IC*

Selected Device: *TP4056*

The TP4056 is widely used throughout many different projects. This means that its use and implementation is well document, making it easier to design a BMS.

The TP4056 IC is designed specifically for charging Lithium-Ion batteries. Although Lithium-Ion batteries are rated at 3.7 volt, they can actually be charged to maximum of 4.2 volts. All charging circuits take advantage of this fact, and the TP4056 is no different. The TP4056 uses 4.2 volts to charge its Lithium-Ion batteries.

3.4 Sound Reflections

One characteristic of sound saves that will prove to be challenging to overcome if we choose to have DiSEL operate as an indoor system is sound reflection.

As sound waves leave its origin point and encounter another medium, such as a wall, a portion of the wave will try to pass the wall while the other portion will simply reflect off the wall. When these sound waves reflect, this can cause the sound wave to interfere with other sound waves inside the building, causing them to combine which will drastically change the wave's characteristics or simply cancel the two waves.

The DiSEL system operates primarily on acoustic processing and recognition. This makes DiSEL incredibly susceptible to errors when being operated indoors. When operating indoors, the sound wave reflections could cause a few different errors:

- False Positive: When the waves interfere with each other, the final distorted wave has characteristics that are close enough to the desired trigger.
- False Negative: A sound wave would typically count as a trigger collides with another sound wave, the waves combine, and the resultant wave no longer has the characteristics needed to count as a trigger.
- Characteristic Error: In this scenario, the system correctly detects a trigger event, however due to the sound waves combining, the resultant wave's characteristics are significantly different than the original trigger wave characteristics thereby leading to the system providing the user with incorrect information. Such as, the incorrect location of the origin point.

These errors would be devastating for the DiSEL system, in addition, these errors would also prove to be very time consuming and pricey to develop solutions for. Even testing would be difficult in those conditions. Due to this, the bulk of DiSEL's testing and operation will be performed outside to minimize these reflections and allow for more accurate readings.

3.5 Possible Architectures and Related Diagrams

This project has many different architectures. We can choose to create a centralized network, or an entirely decentralized network. To do 2 dimensional multilateration we will need at least three sensors, but more sensors can make the system stronger in: service area, accuracy, location time, redundancy, and several other areas.

However, some of these architectures are harder to implement than others. An entirely decentralized network has many complex design problems that need to be solved. On the other hand, a centralized network is much straighter forward, and should be simple and easy to implement. Even if we decided to create a decentralized network, it would be easier to create a centralized version first because it may provide insights as to how a decentralized version would work. Still, the design problems associated with decentralized networks are numerous and complex, so it may not be realistic to create within our time and resource constraints.

It also makes sense to start with the minimum number of sensors, because it may provide insights into how more sensors can be added to the network, and how a general sensor management algorithm would operate.

One system architecture can be seen below in Figure 3-31. This design is not too specific as to how the actual hardware for our prototype will be connected, but it does show how all the hardware will communicate together. As we continue to refine our design of the project, our model will also be refined down into a more specific layout of how the hardware for this project will be connected both visual and with schematics. This will be done in phases as the research in these areas of connection are discovered, and by trial and error when we begin to build our prototypes hardware.

We ended up using a design very similar to this diagram. We used a NEO-6M instead of the Adafruit Mini GPS, the Teensy 3.2 instead of the Teensy 4.0, and the Audio Shield used I2C instead of SPI.

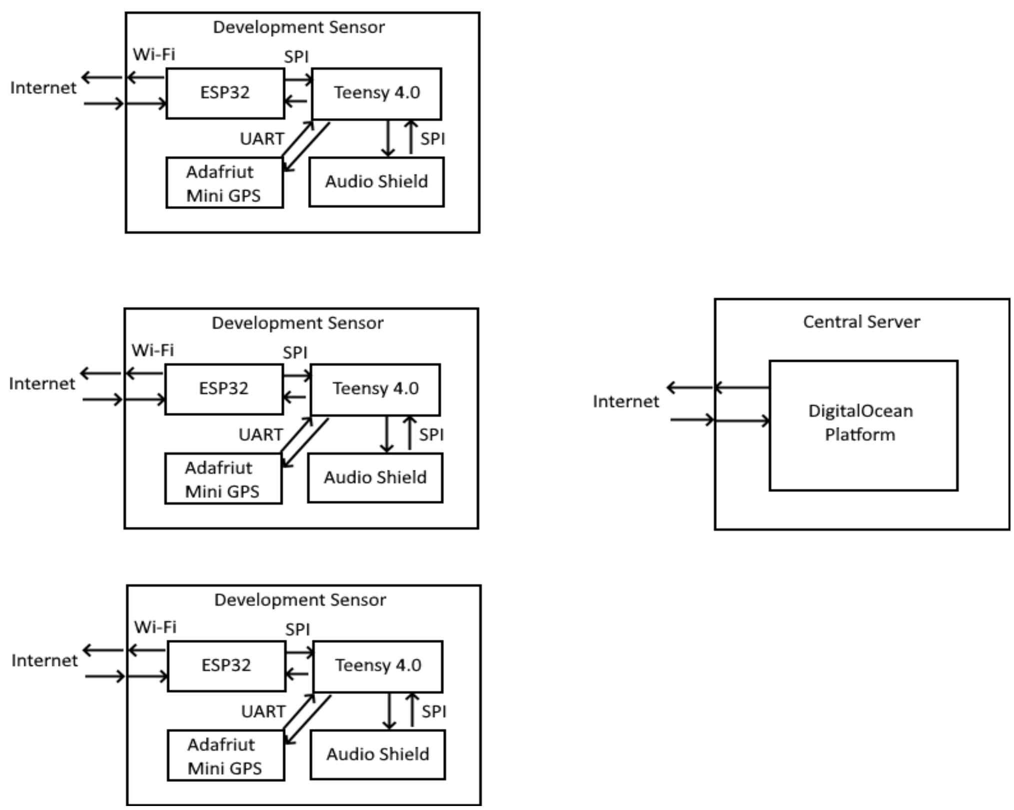


***Figure 3-31**: Hardware Architecture*

3.6 Major Parts Selection Summary

Table 3-21 gives a summary of the major part selections made for one development sensor. It features some additional parts to facilitate connecting everything together.

The figure below shows the parts that we have already collected for our projects. This includes the Teensy4 audio shield, an electret condenser microphone, a Teensy 4.0 breakout board, a Teensy 4 ESP32 breakout board, and an ESP32 Wi-Fi module.

The photos containing the green board of the figure display the Teensy4 audio shield and the microphone. The photos containing the purple board display the Teensy 4.0 breakout board, the ESP32 breakout board and the ESP32.

Unfortunately, prior to having the photos taken, the parts have already been combined into the first prototype for our audio sensor. Though these components have been put together, each of the individual components are still present and have been received. The remaining components have already been ordered, but we are still waiting for these orders to be completely processed and delivered due to back ordering as a result of the current business and factory closures.

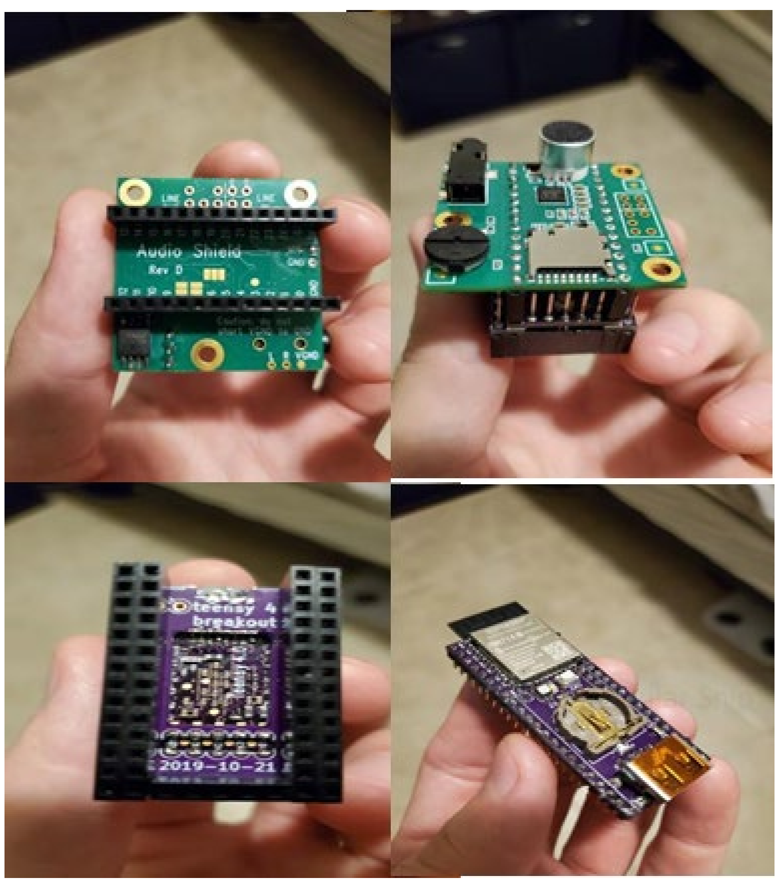


***Figure 3-32**: Collected Parts*

Quantity	Item	Part Number	Manufacturer	Total
1	Audio Shield	TEENSY3_AUDIO	PJRC	$13.75
1	Electret Condenser Microphone	AOM-6738P-R	PUI	$1.25
1	Thumbwheel Potentiometer	POT_THUMB_25K	PJRC	$1.15
1	Teensy 4.0 Breakout Board	v2019-10-21b	OSH Park	$2.10 ($6.30/3)
1	Teensy 3.2	TEENSY32	OSH Park	$18.00
1	Teensy4 ESP32/SD/USB Host Breakout Board	Teensy4 ESP32/SD/USB Host	OSH Park	$3.30 ($9.90/3)
1	ESP32-WROOM-32E	ESP32-WROOM-32E	Espressif Systems	$2.50
1	*NEO-6M GPS Module*	GT-U7	Geekstory	$11.00
1	NOR Flash spiFlash	W25Q128JVSIM	Winbond	$1.40
			Total	$33.85

***Table 3-21**: Part Selection Summary*

4 Related Standards and Realistic Design Constraints

The following section will involve the discussion of the multiple standards related with our project along with the design impact in relation to each of these standards. Our design constraints will then be discussed including both constraints defined by our customer, and those defined from outside of that relationship such as environment and economics.

4.1 Standards

Standards are the building blocks to having a common language and set of guidelines when designing, creating, and using products and services across an array of industries, countries, and the world. They are used to allow new products and service that are introduced into the marketplace to be compatible with other existing technologies as well as safety for both producers and users. Standards can either be used voluntary or enforced by law, depending about a standard and the views nation or governing body of which the standard is being considered.

4.1.1 802.11 Standard

The 802.11 standard refers to a family of standards that contain techniques for wireless communication and their protocols. The first subset standard is 802.11a. This standard can provide up to a 54 Mbps data rate using the 5 GHz frequency band. The next standard is 802.11b which can provide up to a data rate of 11 Mbps using the 2.4 GHz frequency band. Other standards included are the 802.11g which is like that of the 802.11a however it utilizes the 2.4 GHz frequency band. 802.11n and 802.11ac are also included both providing greater speeds. However, the 802.11ac does sacrifice distance in indoor applications. When acquiring items for the project that will be using Wi-Fi, close attention must be paid to the frequency each item must utilize for compatibility since this IEEE standard provides more specs rather than information of equipment compatibility. This is due to the face that Wi-Fi is a term that can be used for any of these individual standards, and not of the compatibility between products.

Comparison Factors	802.11a	802.11b	802.11g	802.11n	802.11ac
Frequency	5Ghz	2.4Ghz	2.4Ghz	2.4/5Ghz	5Ghz
Max Data Rate	54 Mbps	11 Mbps	54 Mbps	600 Mbps	1 Gbps
Typical Range Indoors	100 ft.	100 ft.	125 ft.	225 ft.	90 ft.
Typical Range Outdoors	400 ft.	450 ft.	450 ft.	825 ft.	1000 ft.

***Table 4-1**: Wireless Standards*

4.1.2 Design Impact of 802.11 Standard

The impact of our design with respect to the 802.11 Standard will be significant. With the intent for our application being used in both an indoor and outdoor environment; the indoor and outdoor ranges along with the data rates of the different subset of the 802.11 standard must take into consideration with our product's design, as it will influence the performance and usability of the system.

4.1.3 IEC 60529 Standard

Standard 60529 is the classification and rating of a device's protection from both foreign bodies such as dust and water. One desire for this project is to have our device be capable of function in an outdoor environment. Thus, this standard must be taken into consideration. The rating system used by this standard involves the testing of products based on both how they perform in environments of water as well as how easily foreign particles can enter the system. In terms of water protection, a rating ranging from 0-8. The rating of 0 informs a product as having no protection against water, 8 shows the ability to be protect from continuous water immersion. Table 4-2 shows details of the ratings 0-8. For foreign body protection a range of 0-6 is used from being non-protected to being dust tight. Table 4-3 shows details of the ratings 0-6. An IP code is then established by combining these two numbers or by replacing one of the numbers with a X denoting that a rating has not been formally assigned to the product. Therefore, codes can either be shown as a combination of the two numbers such as "IP58", or only having a formally declaration of one of the protection codes such as "IPX7" or "IPX6".

Degree of Protection	Meaning
0	No special protection
1	Protected against falling water equivalent to 3-5mm rainfall per minute for a duration of 10 minutes
2	Protected against falling water when tilted up to 15 degrees
3	Protected against spraying water
4	Protected against splashing water
5	Protected against water jets
6	Protected against heavy seas
7	Protected against water immersion
8	Protected against water submersion

Table 4-2: *IP Water Protection Codes and Meanings*

Degree of Protection	Meaning
0	Non-Protected
1	Protection against a sphere of 50mm
2	Protection against “Test finger”/sphere of 12.5mm
3	Protection from a rod of 2.5mm diameter
4	Protection from a wire of 1mm diameter
5	Dust-Protected
6	Dust-Tight

Table 4-3: *IP Foreign Body Protection Codes and Meanings*

4.1.4 Design Impact of IEC 60529 Standard

With the desire for our project sensor to have the capability to operate in an outdoor environment, the consideration of the IEC 60529 Standard will have a direct impact with our design. Adhering to this standard when choosing the casing for our design will allow for an outdoor operational environment to be feasible.

4.1.5 C Programming Standards

The purpose of C programming standards is to provide consistency and universal understanding of the meaning and function of lines of code, both by other programmers and compilers. According to a committee formed at AT&T’s Indian Hill Labs, “Good C coding style and standards should encourage consistent layout, improve portability, and reduce errors.” [Cannon]

4.1.5.1 Comments

Comments are essential to good code as they allow a programmer who is unfamiliar with the code quickly understand what is happening within the code and you those functions are being completed. They can also increase clarity on the meaning of parameters and where global variables are used or changed. The use of *TODO* comments are also an essential practice when producing code, especially in a group effort, as it conveys what sections of code are still needed to be completed.

4.1.5.2 Naming Conventions

Naming conventions within C provide a standard way to understand certain identifiers as well as invoking a greater understanding. The first major staple of a proper naming conventions is to have the names of variables provide information, either the functionality or symbolism, that a variable/function is representing. For example, a variable representing an average would be better suited as being named "*average, or avg"* rather than simply "*x"*. Other naming conventions include names with leading and trailing underscores to be reserved for system purposes. To assist in avoiding confusion by other programmers one should also avoid names that differ only in case, such as, *'foo'* and *'Foo'*; and ones that look similar on terminals and printers, such as, *'1'*, *'l'*, and *'I'*. (The constant one, lowercase l, and uppercase I). [Cannon]

4.1.5.3 Formatting

Standardized formatting allows the viewing of one's code to be much easier for other programmers. Some general formatting guidelines include, lines of code should be limited to 80 to 120 characters long to allow a universal fit for most editors used. This also would include the reduction of the amount of nested indentation within code. Tabs should be set to agree upon length typically between 4-8 spaces, if the tab length is not agreed upon, misalignment of code can occur, causing the code to become difficult to read and understand.

4.1.5.4 Header Files

A header file is a file with extension *.h* which contains C function declarations and macro definitions to be shared between several source files. [Tutorialspoint] Every program written in C contains requests for header files with the using of the directive *#include.* Header files should be used to keep all constants, macros, system wide global variables, and function prototypes with the header file. [Tutorialspoint] To avoid double inclusion which would result in an error the following wrapper can be used.

```
#ifndef EXAMPLE_H
  #define EXAMPLE_H
  ... /* body of example.h file */
  #endif /* EXAMPLE_H */
```

Figure 4-1*: Define Statement with Double Inclusion Avoidance Wrapper*

4.1.6 Design Impact of C Programming Standards

With consideration of C Programming Standards will impact the software design of our project. All code written for this project within the C language will be impacted by these standards. All members' code written in C will be reviewed to insure adherence to these standards. Due to the multiple member nature of this project, our design of programs written in C will be of a uniform state by conforming to these standards.

4.1.7 C++ Standard

C++ is a general-purpose programming language that has many different features, and is designed to perform just as well if not better than C. It is an extension of C, so it supports all features from C as well as the entire C standard. The language has object-oriented, functional, generic, and low-level memory manipulation. C++ is standardized by the International Organization for Standardization (ISO). The most recent standard is C++17, which is supported by most compilers, but the C++20 standard should be published soon, and some compilers already support it. C++ enables programmers to create software more quickly and in most cases with better performance than even C because of the wide variety of language features that it supports.

4.1.8 USB Standard

The USB industry standard establishes specifications for cables and connectors. This includes the type of connectors used, power provide, and data transfer rates. The following table shows the connector, power, and data transfer rates for generations of USB.

Specification	Connector	Power Delivery	Max Data Transfer Rate
USB 1.0	Type A/Type B	5V, 5 A	12 Mbit/s
USB 2.0	Type A, B, microA, microB, miniA, miniB, miniAB	5V, 500mA	480 Mbit/s
USB 3.0	Type A/Type B SuperSpeed, microB SuperSpeed	5V, 900mA	5 Gbit/s
USB 3.1	Type A/Type B/Type C SuperSpeed, microB SuperSpeed	5V, 900mA	10 Gbit/s
USB 3.2	Type A/ Type C	5V, 900mA/1500mA	20 Gbit/s

***Table 4-4**: USB Standards*

4.1.9 Design Impact of USB Standard

When considering a USB microphone, we must consider the USB standard to ensure that proper power is provided to the microphone. In our design we must also consider the data transfer that will be required to have a fully functioning system.

4.1.10 JTAG Programming Interface

Being able to test the device and to provide updates to the software of the device is going to be vital to making sure both the hardware and software are running properly after the device is manufactured. JTAG was a method that was developed in order to provide a solution to this issue by allowing for a simple Test Access Point (TAP) that could be used for debugging. JTAG standards are provided under IEEE 1149.1.

Under IEEE 1149.1, the following pins are specified to meet proper JTAG standards:

- **Test Clock (TCK):** This pin sends a signal that controls the TAP clock.
- **Test Mode Select (TMS):** This pin controls the actions the JTAG operates.
- **Test Data-In (TDI):** This pin communicates with the chip and feeds data to it.
- **Test Data-Out (TDO)**: This pin receives outbound data from the chip.
- **Test Reset (TRST)**: This pin is used to reset the JTAG.

The i.MX RT1062 processor that is being utilized for the device follows the JTAG standards and uses the five pins mentioned above. According the processor's datasheet, in addition to the pins above, there is a sixth pin named JTAG_MOD that is used to switch between an IEEE 1149.1 compliant interface and an interface that can be used for common SW debug by adding all the system TAPs to the chain. The datasheet also states that external resistors are not necessary for on-chip termination. A summary of the JTAG controller interface is below:

JTAG	I/O Type	On-Chip Termination
JTAG_TCK	Input	100 kΩ
JTAG_TMS	Input	47 kΩ
JTAG_TDI	Input	47 kΩ
JTAG_TDO	3-state output	Keeper
JTAG_TRSTB	Input	47 kΩ
JTAG_MOD	Input	100 kΩ

Table 5-1: *JTAG Controller Interface Summary*

The JTAG interface will be instrumental in providing firmware updates to the device as the software team patch and update the initial program.

4.1.11 Electrical Safety Standards

With our project involving the use of electricity we must be mindful of the safety standards created by the Occupational Safety and Health Administration (OSHA) regarding the use of electricity. This will allow for our group to design and test our system while staying safe in the process.

Electric Power Generation is written to establish safety guidelines when an activity results in the generation, transmission, or distribution of power. This includes being aware of potential hazards that may be within your work area and steps on how to prevent accidents from happening.

Personal Protective Equipment (PPE) is in reference to items that can be worn to protect against hazards involved when dealing with electricity. This may include safety glasses, face shields, insulted rubber gloves, insulated sleeves and flame-resistant clothing.

Another section involved with this standard is Insulating Protective Equipment (IPE) this includes items that are not worn but may be used to protect someone from encountering electrically charged items. This may include insulating rubber line hose, blankets, and hoods, insulating barriers made of fiberglass, hot sticks, switch sticks, and shotgun sticks.

General Wire size and the Ampacity is another standard set with the ESS. In general, as the wire gauge (AWG) increases, so does that of the ampacity. When choosing wiring for our system this substandard must be considered. The following table gives a representation of a few wire AWG to ampacity ratios.

Copper Wire size (AWG)	Ampacity in free air	Ampacity 3-conductor cable
14 AWG	20 Amps	15 Amps
12 AWG	25 Amps	20 Amps
10 AWG	40 Amps	30 Amps
8 AWG	70 Amps	50 Amps

Table 4-5*: AWG to Ampacity Table*

4.1.12 Design Impact of Electrical Safety Standards

These standards will have an enormous impact of the design of our project as we must always be aware of these safety standards. Not only will this affect how we choose to design our system, but also how we will test the designed system in the future.

Following electrical safety protocols also falls within Ethical constraints as this device is intended to be used for the benefit of the user and not put the user or anyone else at risk of harm or injury.

4.1.13 IEEE 1149.1 Standard

The IEEE 1149.1 standard discusses how to implement the standard integrated method for testing and programming printed circuit boards. The standard specifies using a dedicated debugging port, known as an on-chip Test Access Port (TAP) that can communicate with the chip without requiring direct access to the chip's sensitive data.

According to IEEE 1149.1, in order to be compliant with the standard, the following set of instructions are required by a device:

- **BYPASS**: causes the TDI and TDO lines to be connected, allowing for other devices in the JTAG chain to be tested without extra overhead.
- **EXTEST**: causes the TDI and TDO lines to be connected to the Boundary Scan Register (BSR), sampling the device's pin states, shifting new values into the BSR and then sending those values to the device pins
- **SAMPLE**/PRELOAD: causes the TDI and TDO to be connected to the BSR, but leaves the device in normal mode. This allows the BSR to be accessed and for the data entering and leaving the device to be sampled.
- **IDCODE**: causes the TDI and TDO lines to connect to the IDCODE register
- **INTEST**: causes the TDI and TDO lines to connect to the BSR. This instruction deals with a device's core logic signals

This standard is going to be very important to our design as it will be the standard that we will be following for designing the JTAG interface on our PCB. The JTAG interface will allow us to debug and program our device.

4.2 Design Constraints

Constraints are important and necessary within the design phase and must be given careful consideration when designing a product. This is because all constraints are imposed outside of the control of the member of the group. The following constraints are discussed below along with the effect they will have in the design of our product. The project will subject to the following constraints shown in Table 4-6 which:

Requires a minimum of three sensors.
The sensors should not have a weight of more than 8 pounds.
The project must stay at a cost below 750 USD.
All components of the projects must share the same clock rate when communicating.

***Table 4-6**: Design Constraints*

4.2.1 Economic and Time Constraints

Due to this project being conducted in a class setting, certain time constraints should be considered. The research, design, and testing phase should be completed by April 21, 2020. After the research, design, and testing phase are completed, the building phase of the project will begin. Our project will be presented at the end of the Fall 2020 semester, with a date set sometime in November of 2020. With the completion of the research, design and testing phase being set for the end of the Spring 2020 semester, we should have plenty of time to then build and troubleshoot our project by the time our November deadline expires.

Due to our agreed upon project budget, economic constraints will result in desired parts not being able to fit within the parameters of our budget. Considering the various parts required to implement this project this constraint will result in the possible use of older technology with less capability than other newer product to stay within our budget.

4.2.2 Environmental, Social and Political Constraints

Social and political constraints will impact the design of our system because of the nature of it being a system that records sound events. One major constraint that is poised both social and political is the increased desire for privacy and transparency when dealing with the recording of one’s environment. Our design must address this constraint by providing a way to protect data that is recorded by our system and include a way to inform our user’s what type of data we will be receiving, and transparency of how that data will be handled.

Outdoor environmental constraints such as weather must be taken into consideration with the design of our system to insure the usability of our product outside. Indoor environmental constraints such as noise levels and echoing effects must also be considered when designing our system to allow for accurate results.

4.2.3 Ethical Constraints

Since the nature of this device includes the recording of sound events of the surrounding area, we must consider constraints in regards of providing security for user’s personal data, and data that is recording. This must be an essential part of the device to ensure other users cannot use the device to spy on others. Measures must be taken to prevent the ability to access and use data collected from one that is not the owner of the account.

4.2.4 Manufacturability and Sustainability Constraints

Manufacturability constraints will apply restrictions on the material and components that will be used in our design. For us, this will be limited to the materials and components that we can access, along with the services available to use to integrate those materials and components.

Due to the desire to have our system operating outdoors, sustainability constraints will be of immense importance in the design of our system. Though we will try to assess both manufacturability and sustainability constraints in ways to attempt to obtain a positive correlation. We understand that in most cases these two constraints typically have a negative correlation.

5 Project Hardware and Software Design Details

While taking into consideration all aspects of our research and prior knowledge in the technologies, theory, components, standards, and constraints related to this project's hardware and software requirements; the design of the product must now be determined. The following section will introduce detailed design plans to implement the DiSEL system.

5.1 Hardware Design

The main components involved in our system include the PCB, microphone, microcontroller, wireless module, GPS module and a battery system. The following design will show how all these components will be included into a singular system.

Autodesk's Eagle PCB software was used for creating the schematic and board layout for the DiSEL PCB. Eagle was chosen as our electronic design automation of choice due to its availability for students, its integration into Autodesk's large library of computer aided design software, and for the experience that the DiSEL team has with the software.

5.1.1 Initial Design Architectures and Related Diagrams

The Printed Circuit Board is a critical part of DiSEL, the PCB and its related components are what will be constantly receiving incoming audio signals that will then be filtered and converted into a digital signal for the microcontroller to run a fingerprint analysis to determine if the original sound will trigger an alert.

When coming up with an initial design for the PCB, we chose to split each of the six main subsystems/components into sections on the PCB to try to simplify the design process as much as possible. The six sections that the PCB is split up into are as follows:

1. The Main Power System
2. The Solar Power System
3. The Microcontroller
4. The Audio Receiver Subsystem
5. The GPS Module
6. The Wi-Fi Module

Breaking up the PCB into these sections allows the DiSEL team flexibility and simplicity when designing the board layout of the PCB, while still minimizing the overall board layout as much as possible. A diagram of the desired board layout is shown below:

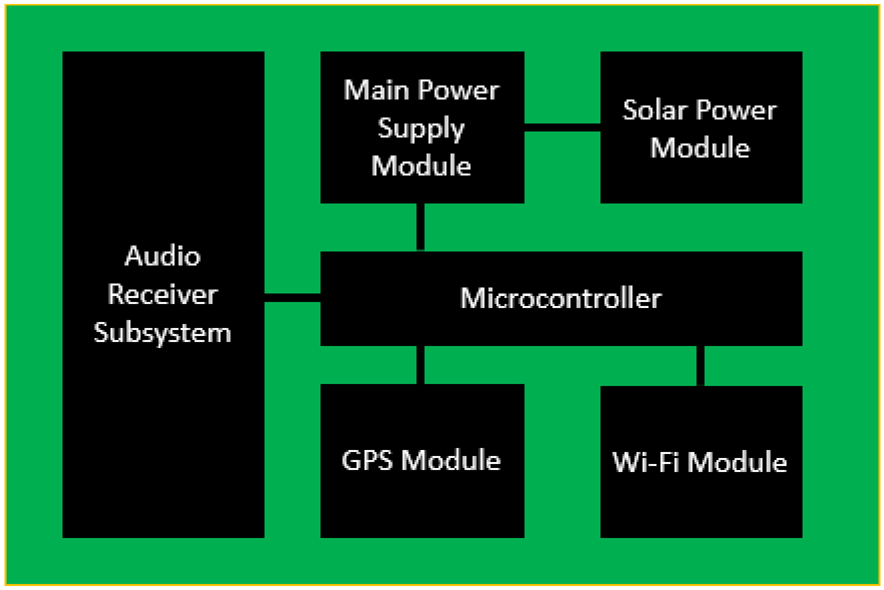


Figure 5-1*: Block diagram used to aid in the PCB's design process*

When designing the PCBs, the size of the PCB and the number of layers is what drives their cost. Since a goal of the team is to make DiSEL a low-cost product, one of our main priorities when designing the PCB is to limit the area of the PCB as much as possible and to only utilize a 2-layer PCB design.

When a PCB is chosen by a design team to be multi-layered, that typically means that the total number of layers used by the design team will be a multiple of two, as two layers are naturally added to a PCB when adding an extra stack and only using one of these layers provides no cost savings to the design team.

With this knowledge in mind, if it becomes evident in the design process that the DiSEL PCB will require more than two layers, the extra layer provided will likely be used as a grounding layer to limit the amount of space that grounding traces use in the PCB, leaving more room for other traces to be placed in the active layers.

One standard that must be addressed when designing the traces on the board layout is that the traces should not make 90° turns and should instead make 45° turns. The reasoning behind this tradition is that it is thought that the sharp corners produced by a 90-degree angle would cause electromagnetic interference due to radiation produced by high frequency signals. Additionally, an added benefit of using 45 degree turns instead of 90-degree ones, is that 45-degree traces take up less space on the board, leaving more useable room on the PCB. This also will allow for the design of the PCB to be more compact resulting in the PCB being a smaller overall size. This reduction in the PCB footprint will enable the overall design of the devices to be smaller reducing the overall weight, and the ability for the devices to function in additional areas.

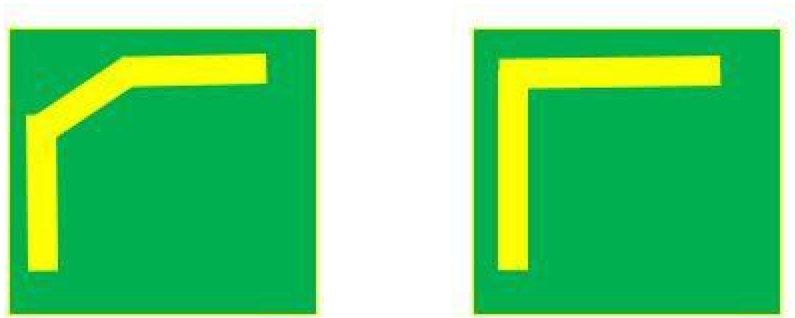


***Figure 5-2**: The standard method of taking corners in traces using a 45-degree angle (left), and incorrectly making a 90-degree trace turn (right)*

The traces being shorter on 45-degree turns will greatly help in reducing the overall cost of the PCB as the price of the PCB is directly correlated to how large it needs to be, with this price increasing substantially as the PCB grows in size.

The development board being used to prototype the first sensor is the Teensy 3.2. In order to maintain code compatibility between the firmware created for the development board and the firmware for the finalized sensors, the microcontroller portion of the PCB was designed using the same I/O pins as the Teensy 3.2. Additionally, a bootloader that is pre-programmed with the Teensyduino firmware was also required in order to allow the use of the Teensyduino audio libraries.

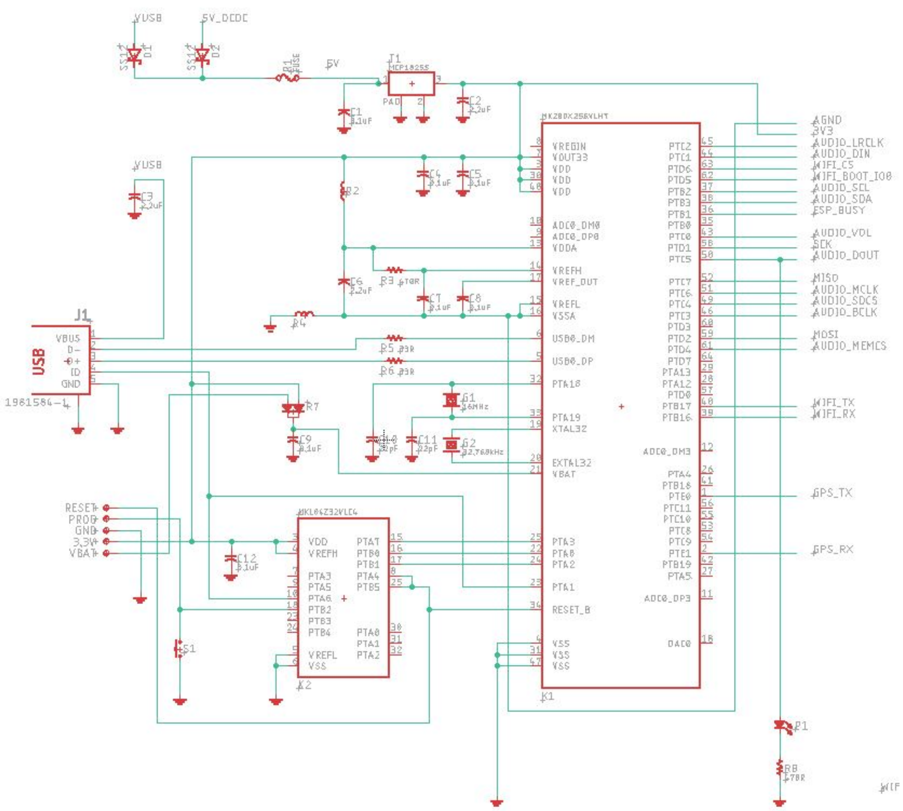


***Figure 5-3**: Schematic for the microcontroller section of the PCB*

5.1.2 Main Power Supply Subsystem and Schematics

The Teensy 4.0 microcontroller that DiSEL uses to operate requires a 5V supply voltage for it to be powered. The batteries that have been selected to power the device are two 3.7-volt Lithium-ion batteries, providing a total maximum voltage of 7.4 on full battery charge. Since the Teensy 4.0 requires a 5V supply voltage, the total 7.4V that is provided by the batteries needed to be dropped down to 5V to not overload the device.

A switching regulator will be utilized to design a DC-DC converter that will be able to maintain the desired supply voltage of 5V to the microcontroller. Additionally, it is important to note in the design that the maximum input current into the Teensy board is noted in the datasheet to be 250mA, therefore when designing the DC-DC converter it is imperative to ensure that its output current is less than 250mA.

When designing the DC-DC converter, Texas Instrument's Webench Power Design tool proved to be an enormous asset. Webench allows the power designer to simulate and customize various designs and voltage regulators to help the user decide which DC-DC converter design would be most beneficial to the project at hand. One important feature of Webench, is that the user can define specific component sizes for their design. For the case of DiSEL, component sizes were selected to be at a minimum size of 0603 to simplify the soldering process when assembling the PCBs.

Two DC-DC Converter designs were tested against each other, using two different voltage regulators: a TPS62175 (Design A) and a LMZM236000SILR (Design B). Schematic diagrams comparing the two DC-DC converter designs are shown below:

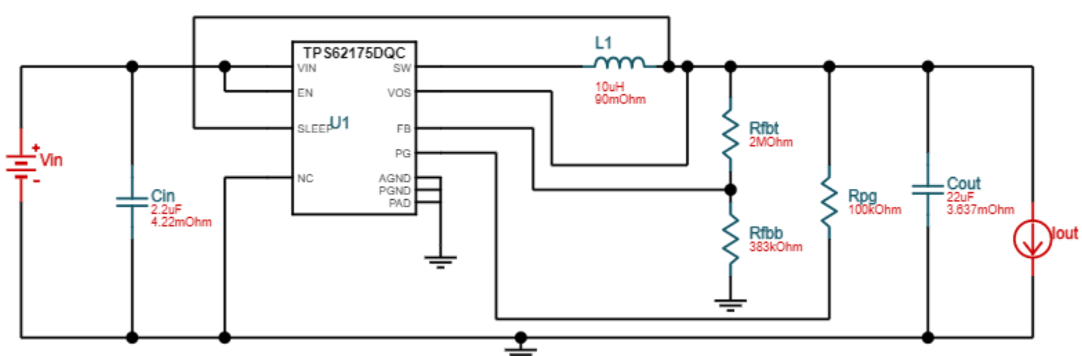


Figure 5-7: *Schematic diagram of **Design A** Using the TPS62175 Switching Regulator*

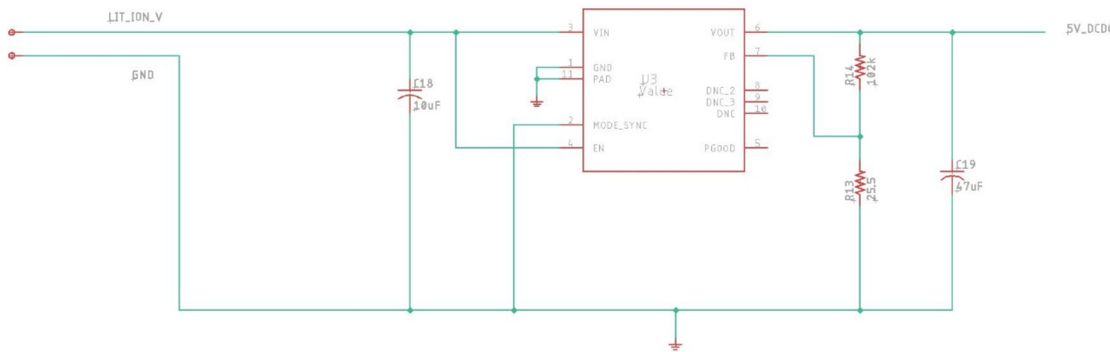


***Figure 5-8**: Schematic diagram of **Design B** Using the LMZM236000SILR Buck Regulator*

When comparing the two designs, the following factors were considered:

	Efficiency	IC Cost	BOM Count	Frequency	Footprint
Design A	94.1%	$0.48	7	1.01 MHz	111 mm^2
Design B	93.8%	$0.67	4	444.56 kHz	61 mm^2

***Table 5-2**: Table comparing important features of the two DC-DC converter designs*

The most important aspect of a voltage regulator is its efficiency, as we do not want to waste power from the batteries. When looking at the two designs, we can see that both are at fairly high efficiency percentages that are fairly close together so that is good on both. The Efficiency vs Output Current charts for both designs are shown below:

Design A

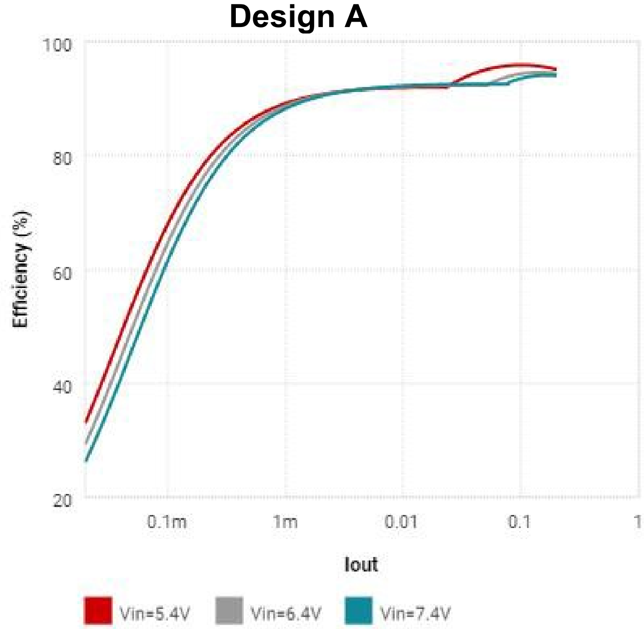


***Figure 5-8**: Efficiency vs Output Current Chart for Design A, Logarithmic Scale*

Design B

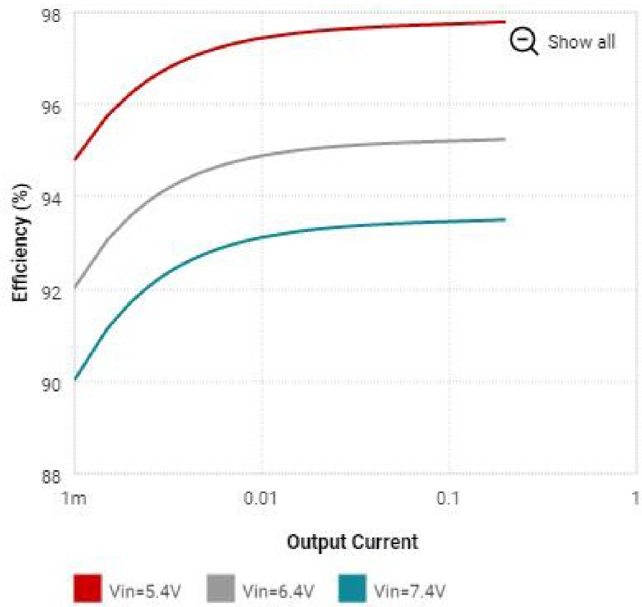


***Figure 5-9**: Efficiency vs Output Chart for Design B, Logarithmic Scale*

The next most important aspect is the footprint. The overall size of the PCB will be the largest determining factor for its overall cost. The DC-DC converter will only be one component of the overall PCB so in order to keep the PCB manufacturing costs as low as possible, we need to make sure the DC-DC converter does not take up any unnecessary space. When comparing the two PCB designs, it is clear that Design B beats out Design A in this category. Design B only has a footprint of 61 mm^2, while Design A has a footprint that takes up 111 mm^2 which is almost doubt the size of Design B.

While Design A does have a smaller IC cost, the overall cost of the DC-DC converter will end up being much higher than in design B simply due to the fact that Design A is twice the size of design B. Due to this, Design B is more attractive than Design A. Additionally, the one extra component that Design B has over Design A still does not make it more beneficial in the long run.

With the main goal of keeping the DC-DC converter small, while still maintaining a high efficiency, Design B was chosen to be implemented to the PCB. Keeping the DC-DC converter design small will allow more room and options for the team when implementing the other components into the overall PCB design.

For the majority of DiSEL's operation time we would like the device to operate in low-power mode. In this mode, only DiSEL's most basic and core functions will be operating, this is done to save power from the battery and increasing the device's overall life span in between charges. However, while we would like for the device to always operate in low-power mode out in the field, we still would like to have an option of easily disabling the device's battery and turning the system completely off. Whether it be for maintenance, transporting the system, rebooting the system, it is always a good idea to design an off switch within the main battery system.

For the sake of this design, the input voltage and ground of the battery will connect to the PCB using header pins that the battery holder can connect to.

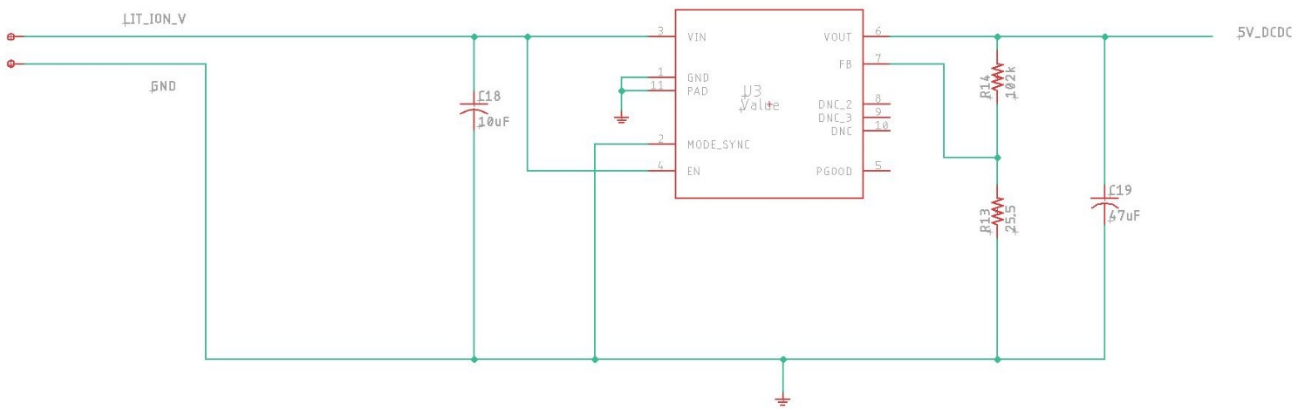


Figure 5-10: *DC-DC Converter schematic on Eagle (with additional switch) that will be used to power the PCB*

The DC-DC converter also performs well in simulations, reaching its average output voltage of 5.13V in 0.72 ms.

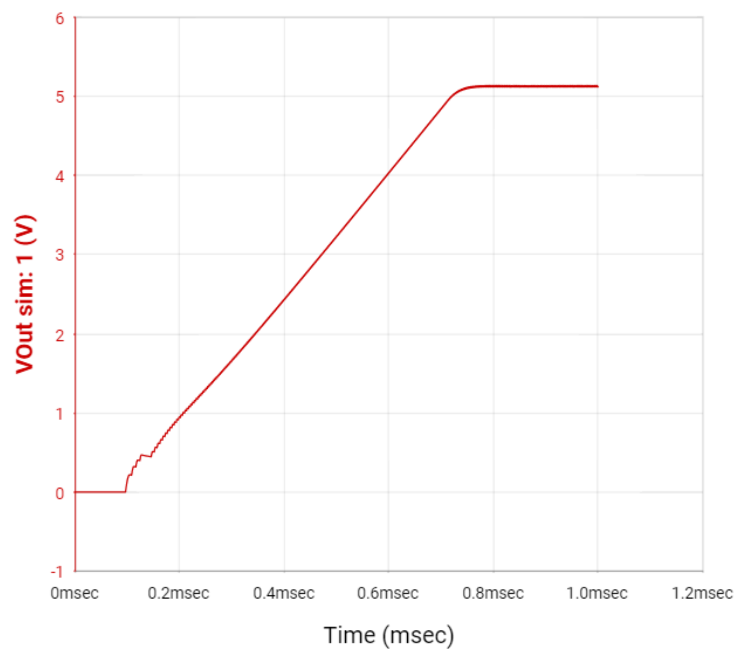


Figure 5-*11: Graph showing the startup response of the selected DC-DC Converter*

5.1.3 Audio Module Subsystem

The audio-receiver is the most critical hardware system of this project as it will oversee the recording of all the necessary audio data that will be analyzed and classified by the DiSEL software systems.

The audio module comes equipped with an Analog to Digital converter. An A-D converter as its name says, converts and analog signal into a digital one. Since we will be using analog microphones, this piece of hardware is required to allow our software to complete the analysis of the collected signals. The way this is accomplished is by converting an analog input with continuous time and voltage levels, to a discrete time and voltage level digital output. The discrete time that is established by the ADC is known as the sampling period, this is determined by the difference in time between two consecutive input samples. The sampling rate is determined by the inverting of the sampling period, resulting in the frequency of the ADC.

Our system network will contain three individual sensors at the minimum. These three sensors will then be placed in a triangular pattern that will be used in the sensor server for the multilateration algorithm. Each of these sensors will be designed to be placed a preexisted pole or be mounted on a wall or ceiling. Due to this design expectation the weight of the sensors will be created with a weight limit of under 10 pounds.

The figure below shows a schematic of the necessary connections between the Teensy Audio Shield and the microcontroller.

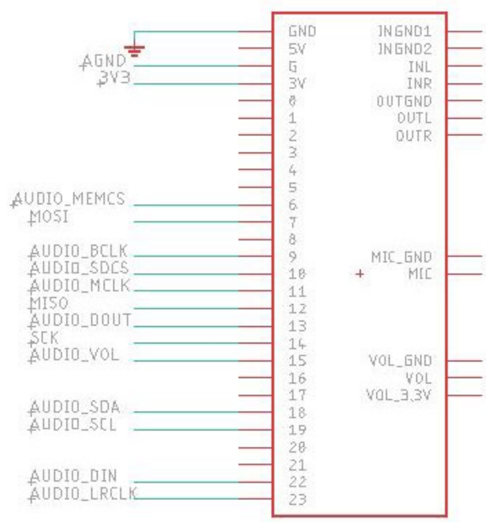


***Figure 5-12**: Schematic of the Teensy Audio Shield Connections to the MCU*

5.1.4 GPS Module Subsystem

Each sensor in the network will include a NEO-6M GPS module onboard. The data received by the GPS modules are vital in syncing the clocks of each sensor, determining the origin location of a sound using the multilateration algorithm, and in displaying sensor locations on the web interface.

The GPS module communicates with the microcontroller solely using UART. It uses its TX pin to send all of the GPS satellite data to the microcontroller. Its RX pin is used to have its settings configured, such as the Pulse Per Seconds.

In reducing the overall size of the board was important to cutting down on PCB manufacturing costs. In order to save board space, the GPS module was connected to the PCB using headers instead of placing the entire module on the board.

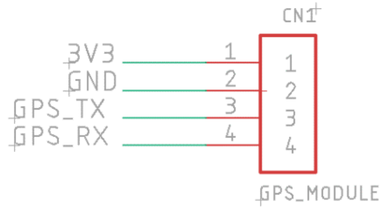


Figure 5-13*: Schematic of the Teensy Audio Shield Connections to the MCU*

5.1.5 Wi-Fi Module Subsystem

The onboard Wi-Fi module is critical to the overall project as it is used for transferring the audio and GPS data collected by our sensor to the sensor server. Consideration was made to utilize Bluetooth instead of Wi-Fi, however due to the large distances used between each individual sensor, the range of Bluetooth was just too small.

The Wi-Fi module is an ESP32 and communicates with the microcontroller using SPI and UART. The SPI communication is used to transfer the audio and GPS data from the MCU to the ESP32 which will then send the data to the sensor server through Wi-Fi. UART is used to program the ESP32 through the MCU.

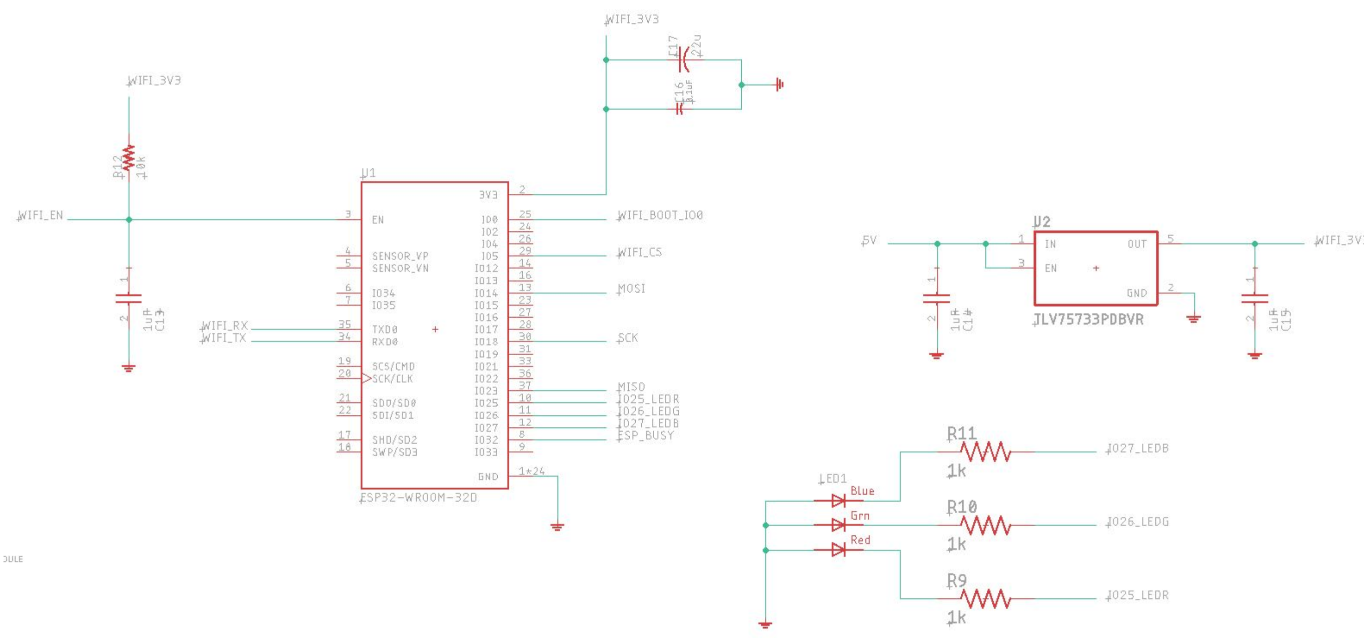


Figure 5-13*: Schematic of the Teensy Audio Shield Connections to the MCU*

As shown in the schematic above, the ESP32 will also have its own voltage regulator connected to its power pins. The current that is outputted by the MCU is relatively low, which has the potential of causing power issues with the ESP32. With this voltage regulator connected, it will output the same 3.3V but will drive it using a higher current. This should effectively remove the potential power concerns.

5.1.6 Backup Solar Power Subsystem

The backup solar power subsystem will consist of a solar cell, a battery management system, and a battery pack. The solar panel will collect photons from the sun and create electricity through the photovoltaic effect. The solar panel used must have a balance between size, weight, and efficiency. The ideal solar panel will have a high efficiency but be small and lightweight. However, as efficiency is increased, size and weight are both increased. Having an efficient solar panel that produces the same amount of energy that is being used by the circuity would allow DiSEL to run indefinitely without its main power source or in non-ideal weather conditions. Depending on the amount of time the backup solar power subsystem will be used, efficiency of the solar panel can be sacrificed. Having a solar panel that is lightweight and small would allow for more mounting options and locations for DiSEL. This means that all the mounting equipment can be less robust.

The solar panel will connect to the battery pack through a battery management system. A battery management system will prevent the battery pack from getting damaged. The battery management system will monitor the charge of the battery pack and ensure that the solar panel does not keep sending power to the battery pack once it is fully charged. Without this feature, the battery pack can overheat and possibly combust. If DiSEL is not drawing power from the solar power subsystem, then the solar panel will eventually completely charge the battery pack, making a battery management system necessary to prevent overcharging.

The battery management system will also prevent damage to the battery pack if too much power is being drawn. This can happen if DiSEL is using the solar power subsystem and the solar panel cannot provide enough energy to keep the battery pack charged. Draining the battery pack past a certain point, can cause permanent damage to the battery pack. The battery management system will cut off power to DiSEL before any damage occurs and will turn on the power once the batteries have recharged to a certain level.

The battery pack will consist of multiple battery cells connected both in series and in parallel. This is done to provide enough voltage and watthours for DiSEL to operate comfortably. The voltage of the battery pack will be higher than the voltage required for DiSEL. Due to this, a voltage regulator will be used to maintain a steady, usable voltage. Since the main function of DiSEL is to locate and identify sound, it would be beneficial to keep noise from the circuitry to a minimum. So, a linear voltage regulator might be the best component for this application. However, if the noise does not affect the sound readings for DiSEL, then a switching voltage regular would be the better component since it is more efficient and produces less heat.

5.1.7 Hardware Enclosure

The enclosure to be used for this project will contain the sensor system, including the MCU, the PCB, and any required sensors and components. Due to the desire to have the product function in an outdoor environment this enclosure must provide protection to water and dust to allow for such functionality to occur. The enclosure will also provide protection to accidental damage when being used in an indoor setting, or when being handled by the user. This enclosure will be made most likely from plastic material due to the reduced cost and its durability. The enclosure may simply be purchased since the time and cost required to properly create a proper casing may be deemed not worth it when many options are already available. If the enclosure is to be purchased, the casing will be a plastic junction box, rated for water and dust proof and the ability to be used with electronic components.

Once the enclosure has been determined, modifications will be needed to be made for desired functionality. The main modification to be performed will be to provide a way to affix the device to a desired location. A strap harness will be added to the enclosure so that the device can be mounted on an outdoor pole.

The next modification required for our enclosure to function properly with our system is to include a notch for the inclusion of an on-off switch for each of our sensor modules. This will allow for easy access by the user to turn the system on and off when in and out of use for battery conservation and simple convenience.

The following figure gives a proposed design of an enclosure for our project.

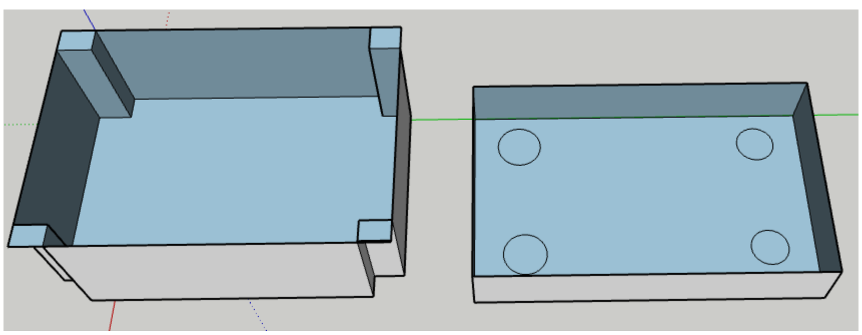


Figure 5-13*: Enclosure Design*

The forethought in the design of our projects housing unit is vital to the overall success of our project. This will be our first defense against outside interference to our system and must be flexible in many different environments to allow our system to perform at its greatest potential.

5.2 Summary of Hardware Design

In summary our hardware design will include a few major subsystems. The main power supply, the audio module, the GPS module, the Wi-Fi module and the backup solar power. Each section has an important factor to the overall success of our prototype, and special consideration must be made when implementing our design for production. This design aims to provide ease and functionality. The physical cost on the system from each subsystem must be closely paid attention to. This will allow our goal of a product being efficient and reliability when in use.

Below is an image of the finalized PCB design, combining every subsystem from this section into one board:

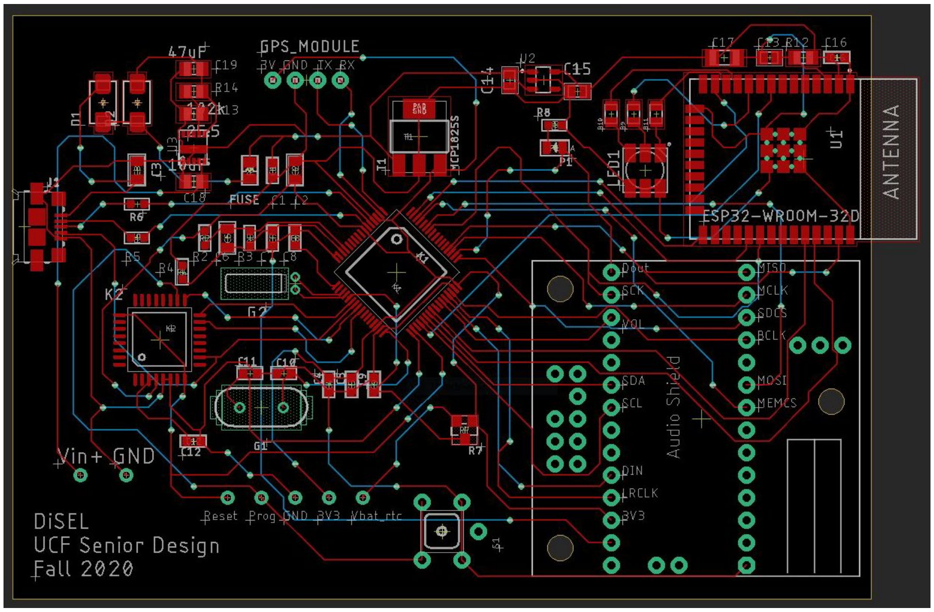


***Figure 5-15**: Finalized PCB design*

5.3 Software Design

In our software design there are two subsets of interaction. The first subset of interaction is the user to web application. This will include the front and backend design and how our users will access the product's acquired and analyzed data. Figure 5-14 shows our use case diagram of a user's interaction with our web application.

The user will begin by accessing our web application using a browser of their choosing. They will then be presented with two options. They can either login if they are a returning user who has already created an account with our system, or they can choose to create an account with a signup option. Once they have completed this process they will return to the same path as a user that was logging in. The user will then be sent to their dashboard. This will be provided for all users once they have logged in either through the login feature or the signup feature. They will then have to option to access the features provided by our system. Once they access the features, they will be presented with two options. They can either stay inside of the system and continue viewing their collected data, or they can choose to logout of the system once they have finished using it.

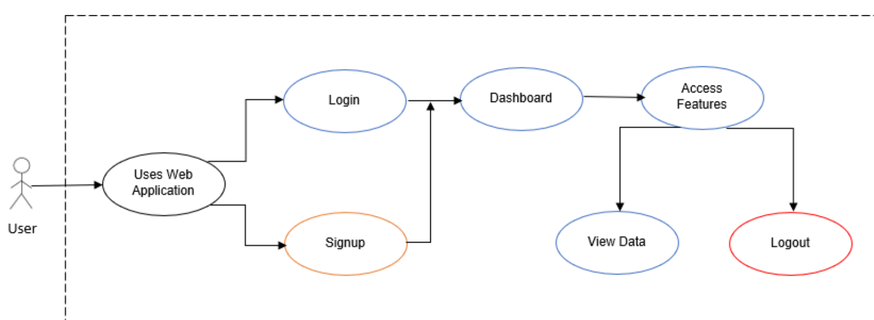


***Figure 5-14**: Use case diagram*

The second subset is the sensor to web application interaction where the sensor records data that will be analyzed and processed into information that will be sent to our web application. Figure 5-15 shows the use case diagram of a sound event interaction with our system.

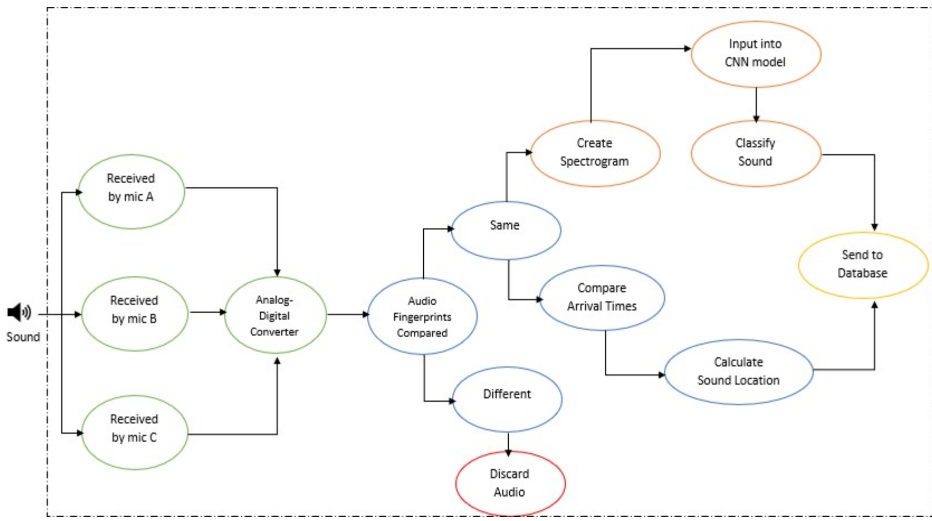


***Figure 5-15**: Sound Interaction with System*

The sound event will begin in the system by being received by the three audio sensors placed throughout an area. All three sensor will then have the audio event pass through an analog to digital converter. This processed audio will then be compared using and audio fingerprint algorithm to determine if all three audio signals from the individual sensors is the same sound. If the audio is different than the audio data will be discarded. If they are the same, then the audio will continue through the system for further analysis. Two processes will then occur simultaneously. A spectrogram will be created based on the audio being passed, as well as the arrival times from each of the sensors will be determined. Once the spectrogram is developed it will be placed in our CNN model which will then classify the sound. While that is occurring the determined arrival times will be sent to be used to calculate the sound location. These two results will then be sent to our database.

5.3.1 Methodology

When building the software for DiSEL, our team will be using an *AGILE* design methodology. Many of the technologies that we will be using to implement the requirements of our system are either new to our team, or our team has only a small amount of experience with them and combining them together. The main goal of *AGILE* is to encourage a group of programmers to be adaptive for when inevitable changes occur. The core values for an *AGILE* project are: Interaction, Working Software, Collaborating with Customers, and Responding to Change.

The process of Agile when developing code is more of a circular progression rather than a linear one. What this allows is instead of developing code where each step is created in order; or group will have the ability to consistently develop and improve code while continuously testing. Since our experience with the technologies required, we can then go back to previously developed code and improve it based on knowledge gained over the time of using these modern technologies.

5.3.2 Operating System

When determining how to design our central server, a discussion had to be made on the operating system to use for it. Our team has decided on using a Linux operating system. Though some consideration was made for using a windows server, the ultimate decision was for Linux for several reasons.

The first benefited that presented itself was pure experience with this operating system when dealing with servers. Group members in previous classes were required to develop server systems and had used a Linux server, therefore the learning curve to learn a Microsoft server is much greater.

Linux also provides more capabilities is the realm of modifications and open-source access. This allows for a varying types of application support and modifications that can be made when necessary, rather than having to have a harder time if an issue arises, and a less likelihood of incompatibility between services.

The final benefit that the Linux operating system can provide is its overall reliability. Due to the nature of how Linux is designed, there is fewer resources being required to allow the system to function, creating a more efficient system.

The sensors will not use a traditional operating system. They will be programmed directly in C++. This means that any libraries that we use will need to be compatible with C or C++.

5.3.3 Programming Language Comparison (Machine Learning)

When discussing the language to be used to develop our machine learning algorithms, the languages of Python and C++ were compared. Python is a high-level general-purpose language with syntax that closely resembles everyday writing, resulting in code that is easy to read and comprehend. C++ on the other hand is a general-purpose language with roots in C, requiring more syntax rules and other conventions. Due to its popularity in the realm of machine learning Python also has the advantage of a large selection of datasets and code examples that involve machine learning. Due to the nature of Python resembling the common English language, writing code in Python is generally shorter and more concise in comparison to C++. Some of this is due to Python requiring only indentations, whereas C++ requires a more conventional syntax with the need of curly brackets and semicolons. An advantage the C++ does hold over Python is the fact that C++ is a statically typed language, this means that type errors will not show up during runtime.

When discussing the language to be used to develop our machine learning algorithms, the languages of Python and C++ were compared. Python is a high-level general-purpose language with syntax that closely resembles everyday writing, resulting in code that is easy to read and comprehend. C++ on the other hand is a general-purpose language with roots in C, requiring more syntax rules and other conventions. Due to its popularity in the realm of machine learning Python also has the advantage of a wide selection of datasets and code examples that involve machine learning. Due to the nature of Python resembling the common English language, writing code in Python is general shorter and more concise in comparison to C++. Some of this is due to Python requiring only indentations, whereas C++ requires a more conventual syntax with the need of curly brackets and semicolons. An advantage the C++ does hold over Python is the fact that C++ is a statically typed language, this means that type errors will not show up during runtime. The following table gives a quick view on the comparison of Python and C++ in the context of machine learning.

Comparison Factors	Python	C++
Type of Language	High level general-purpose language	General-purpose language
Convention Levels	Low. Syntax conventions resembles everyday writing	High. Syntax resembles C programming language
Syntax Conventions	Indentation	Curly Brackets, Semicolons
Language Type	Dynamic	Static
Machine Learning Relation	High number of machines learning modules due to its popularity with using python in machine learning	Not as popular with programmers of machine learning, resulting in a lower number of modules that deal with ML.

Table 5-3: *Machine Learning Programming Language Comparison Chart*

In conclusion, due to the easy and simplicity of the syntax in Python, and the availability of libraries, information, and datasets available in the realm of machine learning. We will be choosing Python as our programming language to be used for the machine learning portion of our project.

5.3.4 Development Tools

With any type of software development there are tools that are available to make the process of developing in a group setting seamless and organized. The following will discuss the varying tools we will be using to implement our software design in a consistent and accurate manor.

An Integrated Development Environment typically shortened to the acronym IDE is an application that provides an environment that are geared towards software development. All IDEs typically consist of the same base features: a source code editor, build automation tools and a debugger. Examples of such are NetBeans, Eclipse and Visual Studio. Another setup to use when developing code is to simply use a source code editor such as Sublime Text or Notepad++ to edit the code and compile and run the code through a terminal such as the Linux terminal. Depending on the preferences and experience of the programmer, either solution or a combination of the two will suffice for our project. Compatibility between the different environments would be the only concern, however in most cases and IDE and source code editor will be compatible if the code being written is of the same language.

A valuable tool that is almost a necessity when programming within a group setting is the use of VCS, short for a Version Control System. A VCS is a system that will keep track of the versions of source code developed. This allows for the system to record changes occur to the code and who made those changes. Another great feature of this type of system is to have to ability to revert to previous versions of code in the case of a bug arising with the addition of new code. The VCS that we will be using in this project is the popular Git, a local VCS that can be download to a local computer. Git can also be used in conjunction with the online version GitHub allowing for a team to view changes to the source code in real time.

Within the development of databases, MongoDB Atlas is an essential tool that we will be utilizing. MongoDB Atlas provides a visualization of a created database along with the relationships to be used between each of the columns. What this software provides is a way to design organized databases with having to read through lines and lines of code to ensure of that organization. This also provides a way to view data being inserted and updated to verify that all sections of the database are functioning as they should be.

Our system will be developed within a Linux environment. To allow for easier development and integration between team members we will be utilizing Oracle's VM VirtualBox to allow group members that do not have Linux as their native operating system. VM VirtualBox is a framework for running virtual machines on computers running a different operating system. This will invoke consistency with the designing and implementation of our software.

In the development of our sensor modules two tools will be vital in the creation of the module's software. The first tool is the Teensy Loader Application. This application allows for communication with the Teensy board we are using to enable the downloading of innovative programs onto the board. There is a version of this application both in a standalone format and a command line format that will both be used based on the preference of the group members invoking new software unto the boards.

The other development tool to be used with the developing of our sensor modules is the Arduino IDE. This is an open-source program provided by Arduino allowing for an effortless way to build code for our board and run it. This software is provided in both a web format, allowing for cloud-based collaboration, and a standalone format. There is also a possibility of transferring the Arduino coding into Virtual Studios with a use of an Arduino addon.

When developing a system within a group setting, one of the most important tools that is needed is a way to communicate with each other. For this project, our team will be using discord to communicate, this allows for an easy use of text, and voice calls to exchange ideas, get help on areas that other members may have experience with and to display progress. To establish this communication, a discord server has been created specifically for this project with all the members being invited to it. One of the most notable features that discord provides us is the voice channel which allows for conference calls between members of the group. This has proven especially beneficial when creating this document, and when trying to develop and begin the development phase of the project when face to face meetings are not available.

Microsoft SharePoint will be used for our projects documentation collaboration and sharing. SharePoint provides our team the capability of accessing and developing documents related to our project, allowing for real time editing and collaboration. This will increase our groups efficiency when creating the required documentation by reducing moments where writing could be overwritten or repeated if the document is not updated at the same time for each member.

5.3.5 Frontend and Backend Development

The frontend of a website or application is like that of a first impression. With the first few moments of interaction the user a determined if the design and features included are worthy of further interaction. With the development of our web applications frontend, we will take into consideration both the look and features it will include. Features that will be included with this interface is an interactive map showing the location of sound being analyzed with our system. A place to view data that has previously been analyzed, and the ability to view the different producers that have been determined of making that sound. The goal of our website's frontend design is to be simple, and robust; where the user's interface is both good to the eye and easy to navigate. The implementation of styling will allow for a customization of the look and feel of the website that will produce a user's feeling of wanting to interact with it. The use of a system that will allow the user to navigate the entire website with the least number of clicks required to reach their destination will be used. General the smaller number of redirections inside of a website the better the navigation appears to the user. For this to work a dashboard design will be used to hold most of the data; with catalog and setting tabs being used for those specific cases. Everything else should be accessible by the user by simply logging into their account. The framework to establish such a design will be implemented using HTML. This markup-language is widely used on almost every website running currently on the internet. However, the framework is what HTML is limited to. To invoke to stylist desires of the website a combination of custom CSS and Bootstrap will be implemented.

To allow an interactive and dynamic webpage, a scripting language is required. This is what allows a website to be transformed by simply styled text, to one that allows real-time interaction with the user. For this project we will be using JavaScript, a widely used scripting language commonly used in conjunction with HTML and CSS. With the addition of JavaScript our frontend can then begin moving. Active data can be updated onto the user's dashboard along with the location of sound events and other relevant data. Figure 5-16 shows a graphical view of our frontend design.

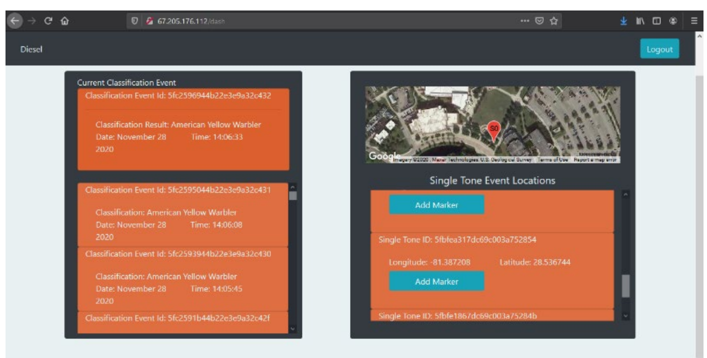


Figure 5-16: Frontend Dashboard Design Layout

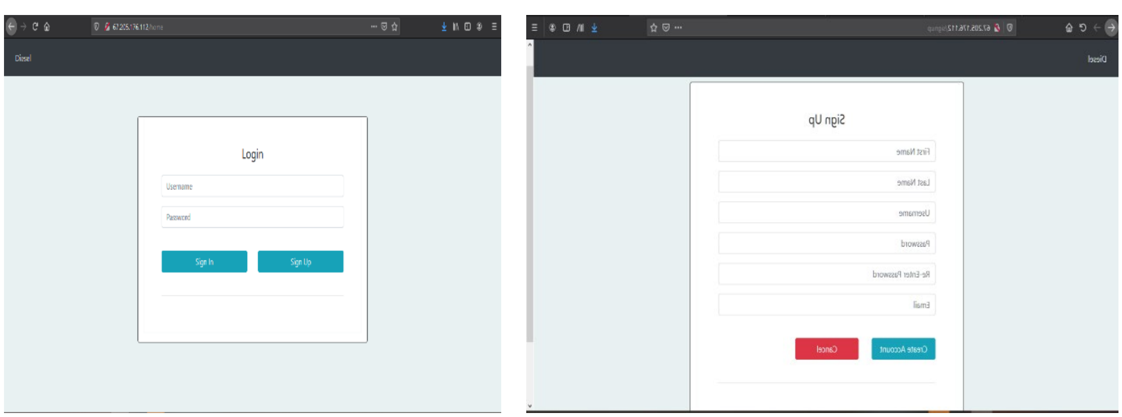


Figure 6-17: Frontend Signup and Login Design Layout

The goals for the backend are very similar to those of the frontend development. However, these goals are more for our team rather than the user since users will have no direct interaction with the backend. The main item involved with the backend is the development of a database. One that is simple to navigate by the group, and can hold all the necessary information, including that of the user and of our system. The creation of the database will be implement using SQL, a query language used to both create tables within the database and traverse the held data when it is needed.

The final piece required for the development of the web application is to connect to frontend to the backend. This will allow the website to be dynamic as it will send and receive information from the user inputs being sent to the server, and the server responding with the user's requested data, or the user's data they are wanting stored. This piece also known as an API will be created with the combination of NodeJS with ExpressJS and JSON. NodeJS is a server-side scripting language framework that provides the functionality to send and receive request on the server side to either add data to the database or search the database for information being asked by the client. JSON is a data interchange format that can take client script and convert it into readable code by the server scripting language and vice versa. This enables the connection between the client and server.

5.3.6 Database

Having a good database design ensures that a web application can have all required data to be stored to all for optimal functionality and quick data searches. The design of our database will include several tables that will contain all the required information necessary for our web application to properly function. The tables that we will have included are a user table, a current sound event table, and a historical sound event table. A user identification key will be used as the primary key to allow the various tables to relate to each other. The user table will contain columns holding user's id key, user's full name, username, hashed password, and email. The current sound event table will hold a user's latest sound event information. This table will include columns holding the user identification key, the events time stamp, latitude, longitude, and classification. The final table in our database the historical sound event table will contain sound event information that has been collected over the time the user's system has been used. This table will hold the same columns as the current sound event table. Figure 5-17 contains a graphical view of the database's tables.

Users	
User_id	*int*(11)
Full Name	*varchar*(255)
Username	*varchar(255)*
Hashed_PWD	*varchar*(255)
Email	*varchar*(255)

Current_Data	
User_id	*int*(11)
Time	*varchar*(255)
Latitude	*int*(11)
Longitude	*int*(11)
Classification	*varchar*(255)

Historical_Data	
User_id	*int*(11)
Time	*varchar*(255)
Latitude	*int*(11)
Longitude	*int*(11)
Classification	*varchar*(255)

*Figure 5-**18**: Database Tables*

5.3.7 MEAN Stack

The idea of a web application stack is to use technologies that together can be used to create a completed web application from frontend and backend, also known as full stack development. The stack that was chosen for this project was the MEAN stack, and acronym for the used technologies MongoDB for the database, ExpressJS which is an API framework for NodeJS, Angular which is a frontend framework, and NodeJS which in conjunction with ExpressJS is a flexible, yet durable API and server-side framework. Utilizing a stack gives a proven combination of technologies that also provide seamless transitions between them. The beauty of the MEAN stack is that all of the web application components for both the frontend and API are written in the same language JavaScript. This reduced some of the time needed to have been spent on learning multiple new languages, as our experience with JavaScript helped in utilizing this stack, only then having to learn new ways of using JavaScript, rather than starting from scratch.

5.3.8 Acquiring and Developing Datasets

There are two different methods to acquiring datasets to use with a neural network. The first way is to use datasets that have already been created, available for use in research. These datasets have been previously curated by others to accurately label the included data which is one of the main time-consuming factors of machine learning.

The second method is to acquire data on our own and then label the collected sound data. This may achieve a wider span of capability with our network, however due to the time constraints that are given by this project this method is not favorable.

The method that we will be choosing to proceed with in our project is to create our own database using an already created API through Xeno-Canto. This will give us the opportunity to keep within our time constraints by having a single source of audio signals that are already labeled, while also using data that has previously been used in neural networks giving us the ability to have a general baseline in our network based on the data, we our using. If time allows, we will also consider using a combination of the two methods by collecting and labeling data on our own and adding it to previously created datasets. This would allow our network and product as a whole increase its capability by providing additional sounds that we can classify. This will also allow our product to go into more depth with our classification. For example, rather than classifying a sound simply as a bird, we could add specific sounds and labels of specific species of birds, creating a network that can provide a species classification of the sound event.

5.3.9 Preparing and Loading Data

A python program will be designed to prepare our data to be used in the development of a working convolutional neural network. This program will begin by taking the audio data files through a spectrogram function. This function will take each audio file as an input and return the related spectrogram to the audio. This spectrogram will now be used as our image for classification purposes in our CNN. The program will then take the file paths of all images that will be used for both the training and testing phases. Each spectrogram image will be resized to size to be determined based on the data being used. This value will be determined by trial and error resizing to different dimensions until the result is satisfactory. The training images will then with the use of python's pickle module will be split into two pickle files with an associating features file and label file. The testing images will only be resized with no further steps necessary to prepare them. The testing files will then be loaded into the CNN with the use of python's pickle load function.

5.3.10 Developing and Training the Convolutional Neural Network

The development of a CNN model will be conducted with the use of Keras, a machine learning API within the TensorFlow python module. We will be using a multi-layer with each performing a specified task by manipulating and performing calculations from the images we will be passing through the network. Due to the nature of CNN's, generally the accuracy of the network is due to the number of layers used to develop the network. The balance required however when developing how deep the network goes is due to the issue of overfitting. Overfitting will result in the network learning how to predict based on the knowledge that a guess will be right most of the time, rather than predicting based on features and patterns learned. This can occur when a network is to deep and complex for the task given to it. To determine the correct depth and complexity we must use to perform the task of classifying our sound events, a trial-and-error techniques must be used which will compare the accuracy and confidence of different network models as they learn. To analysis these comparisons, the program Tensorboard will be used. Tensorboard takes the results of each proposed model, charting the accuracy (percentage that the model correctly predicted a classifying label for an image) and loss factor (the confidence of which the model made its prediction. ex. Predicting an image label as ninety percent a dog and ten percent a cat) of each pass also known as an epoch and the validation set. Figure 5-19: Tensorboard Output Example shows an example of the Tensorboard interface and comparing results we will be viewing. The main determining factor of this comparison will be of the validation loss factor. To creation of this data involves training each of the proposed models. This is accomplished by taking a defined training dataset consisting of related images along with classifier labels. This dataset will then be split with a 90/10 split with ninety percent of the dataset being used for the training of the model, and the other ten percent being used for validating the model. We will then pass the training set through each of our proposed model's network. This process will be repeated until three acceptable models have been created with an accuracy of ninety percent or greater. These models can then be moved to the testing phase for further evaluation.

5.3.11 Testing Convolutional Neural Network

The final phase of completing a functioning CNN is the testing phase. We will carry out this testing by acquiring and creating a dataset of images that have never been seen by any of the three proposed models being taken to the testing phase. We will then load this dataset set as we did with the training dataset, however we will only include the images without any labels or identifying names. This set of images will then be passed through each of the models and the resulting predictions will be analyzed. The model that produces the highest accuracy will be the model put into production. If there are similar results between any of the models, the model's loss factors will be compared with the model with the lowest loss factor being used in production.

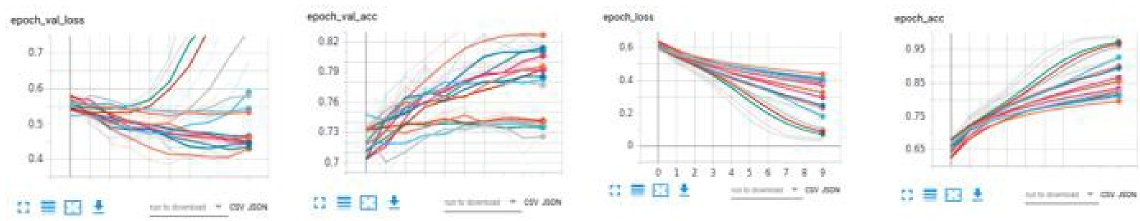


Figure 5-19: *Tensorboard Output Example*

5.3.12 Flask Server

With our system being designed around a web application for user interaction, their then posies the issue of how to implement the classification model with data captured from the sensors and display it to the user. Since the machine learning model was designed in python, the easiest way to implement the classification model to service the data and user was to create a server based in the same language. This is where the flask server contained came in. Flask is a server container designed and developed in python, allowing for an easy implementation and transition for utilizing our model. This allows the model to both run on a dedicated machine learning server, taking in data from the sensors central server, and sending the classification results to be stored on the web application database and server by only have to use simple http requests such as posts gets. This simplified this process greatly as any other attempt at this would involve translating the python code to and from another language such as JavaScript to run on another server framework which is both difficult and prone to errors.

5.4 Summary of Software Design

In summary, our software design has two main phases. The hardware to software integration phase, analyzing sound data, and the software to user interaction phase, taking analyzed data and creating a form that is easily understood by any user. The focus of this design is to use techniques that will provide a streamline system. To achieve this design the cost of each section must be kept in mind in implementation and integration of our design to allow our data to interact with our user with speed and efficiency.

6 Project Prototype Construction

The following section will discuss the overall creation of our prototype. This will include the integration of all our hardware parts schematically, along with how we will be creating our PCB with a vendor.

6.1 Integration

Integration is the combining of power and data within a system of electrical components. Within this context the data being sent (transferred) provides protocol and information to allow the network of components to function cohesively. The following diagram gives an overview of the interaction between the various electrical components.

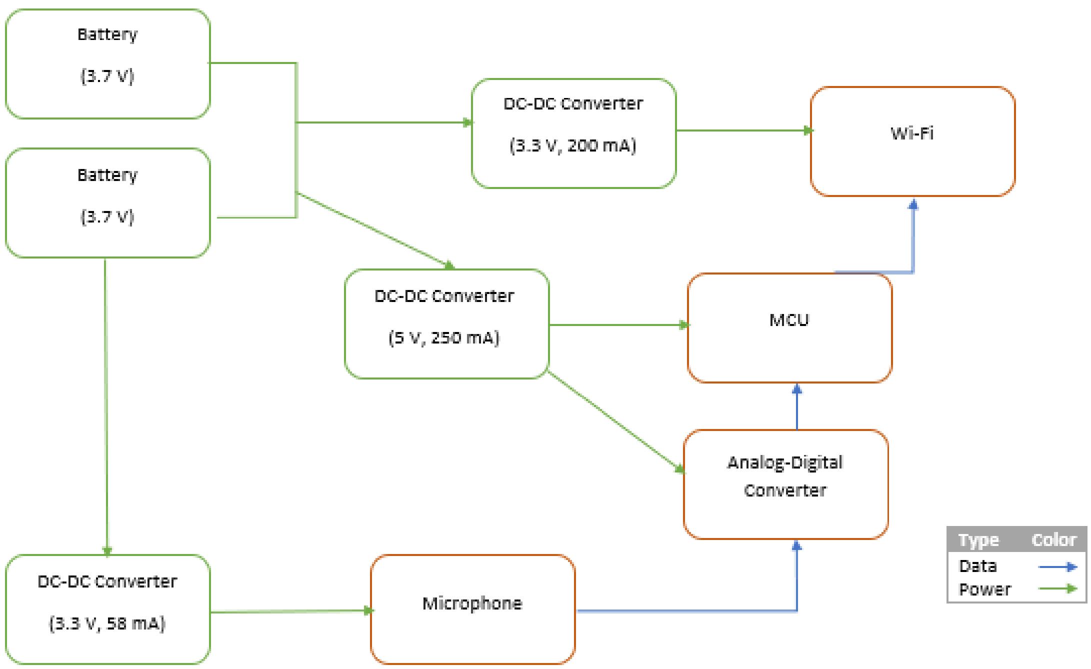


***Figure 6-1**: Component Integration*

As the above figure shows there are multiple sections to the integration of our system. Due to the use of different components, there are varying power requirements needed for each of the components to properly function. As the power from the batteries are distributed, the flow of the hardware design leading to our software can be seen in the blocks outlined in orange. As the microphone receives a signal, that signal is pushed to the ADC. This converted signal is then sent to the MCU where analysis of the signal occurs. Everything leads to a Wi-Fi module which will transmit the analysis data to a central server to be further analyzed and recorded to our web application to allow viewing by our user.

6.2 PCB Vendor and Assembly

The fabrication house that will be used for the DiSEL prototype PCB is OSH Park. OSH Park produces high quality PCBs for prototyping for a low cost and has a fast-shipping time of around 9-12 days (about 1 week 5 days) for its lowest cost prototype board. Since there may need to be modifications to the initial PCB design, having the opportunity to receive these boards as soon as possible, while also keeping the costs within budget, OSH Park is the best choice. An added benefit to using OSHPark is that they will ship three copies of your PCB so if we accidentally damage our first one, we can use the second or third copy.

OSHPark PCB Options			
	Layers	**Price/in^2**	**Shipping Time**
Prototype	2	$5	9-12 days
After Dark	2	$5	12-21 days
Super Swift	2	$10	4-5 days
4 Layer	4	$10	9-14 days

Figure 6-2*: The fabrication options that OSHPark provides that we are interested in.*

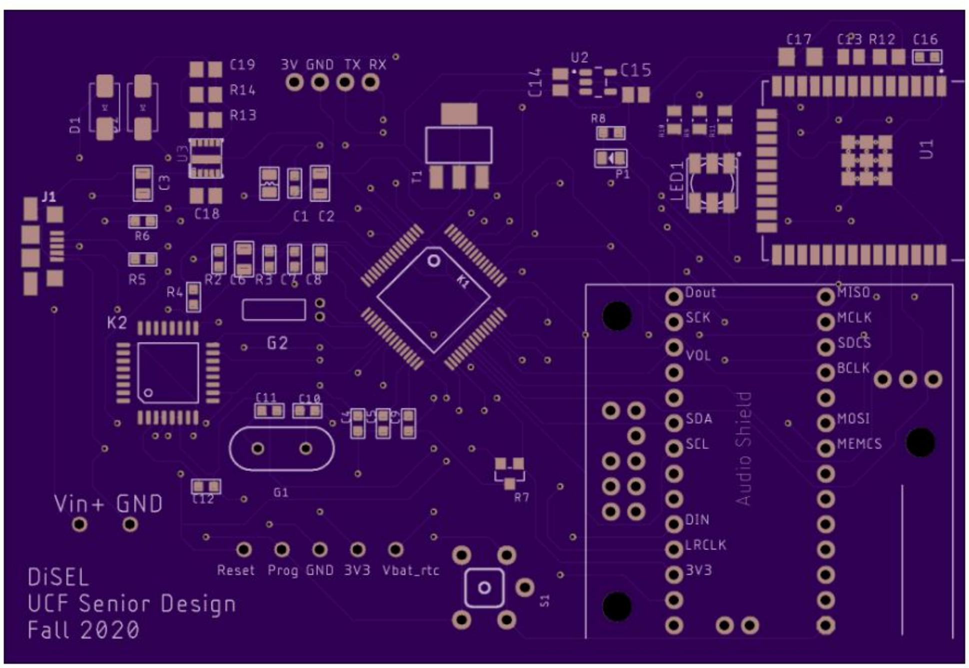


Figure 6-3*: The layout of the bare PCB from OSHPark.*

Once the printed circuit board arrive from fabrication and the ordered components also arrive, then it comes time to assemble the PCB. The PCB is designed to primarily using surface mount devices. Since the PCB uses so many SMD components, the team's initial plan was to use a reflow oven to quickly and efficiently mount each component to the board. However, it was later decided to manually solder each component to the board using a soldering iron. While this method is significantly more tedious than using a reflow oven, the team had more experience with it.

Another important fact that the DiSEL design team considered when selecting the components for their PCB and for designing the PCB is the size of the components. As mentioned above, the PCB needed to be kept as small as possible to maintain a low manufacturing cost, so the components used for the board had to be as small as possible. Another consideration for selecting the component sizes were how easy they were going to be to solder since a regular soldering iron and heat gun were going to be used. For as many parts as possible, components no smaller than 0603 were selected in order to strike a balance between these two constraints.

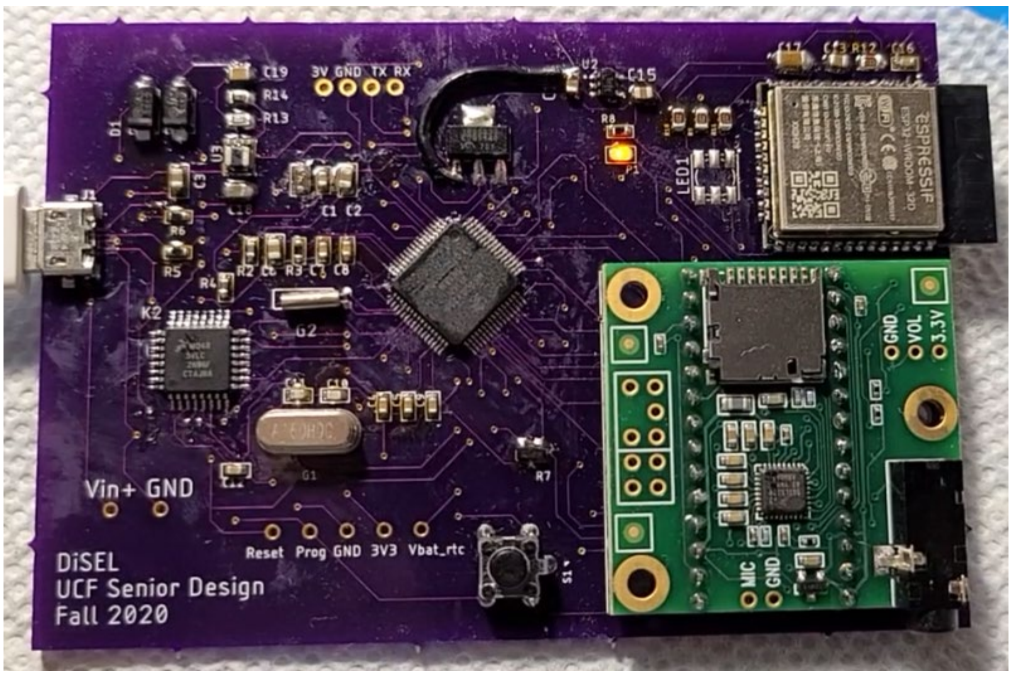


Figure 6-4*: Finalized PCB with all the components soldered on*

6.3 Battery Management System

Each PCB is powered using two 3.7-volt, Lithium-Ion, 18650 batteries. Each battery is charged using a TP4056 IC, which is designed specifically for charging Lithium-Ion batteries. Although Lithium-Ion batteries are rated at 3.7 volt, they can actually be charged to maximum of 4.2 volts. All charging circuits take advantage of this fact, and the TP4056 is no different. The TP4056 uses 4.2 volts to charge its Lithium-Ion batteries.

To add protection for the batteries, a DW01A IC is used in the TP4056 charging circuit. The DW01A IC offers two main features: over-charge protection, over-discharge protection. If the battery voltage goes above 4.3 volts, the DW01A cuts off power to the battery. Once the battery voltage drops below 4.1 volts, the DW01A will restore power to the battery. If the battery voltage drops below 2.5

volts, the DW01A cuts off the battery from the output. The DW01A reconnects the battery to the output once the battery voltage goes above 3 volts.

For DiSEL, two TP4056 charging circuits were used to create a BMS. This allows each battery to be individually monitored, charged, and discharged, ensuring proper cell balancing. The outputs of the TP4056 charging circuits are connected together in series providing a maximum voltage of 8.4 volts.

Since the two TP4056 charging circuits are connected in series at their outputs, the inputs cannot share a ground. If both TP4056 charging circuits are powered by the same voltage source, one charging circuit will get shorted out and bypassed. To solve this problem, there are two solutions: Use two different power supplies (two solar panels) or isolate the grounds from each other.

For this project, a B0505S-1W was used to isolate the grounds of the charging circuits. The B0505S-1W is a DC-DC converter with isolation. It takes an input of 5 volts and outputs 5 volts. The B0505S-1W is able to isolate the input from the output by using a transformer.

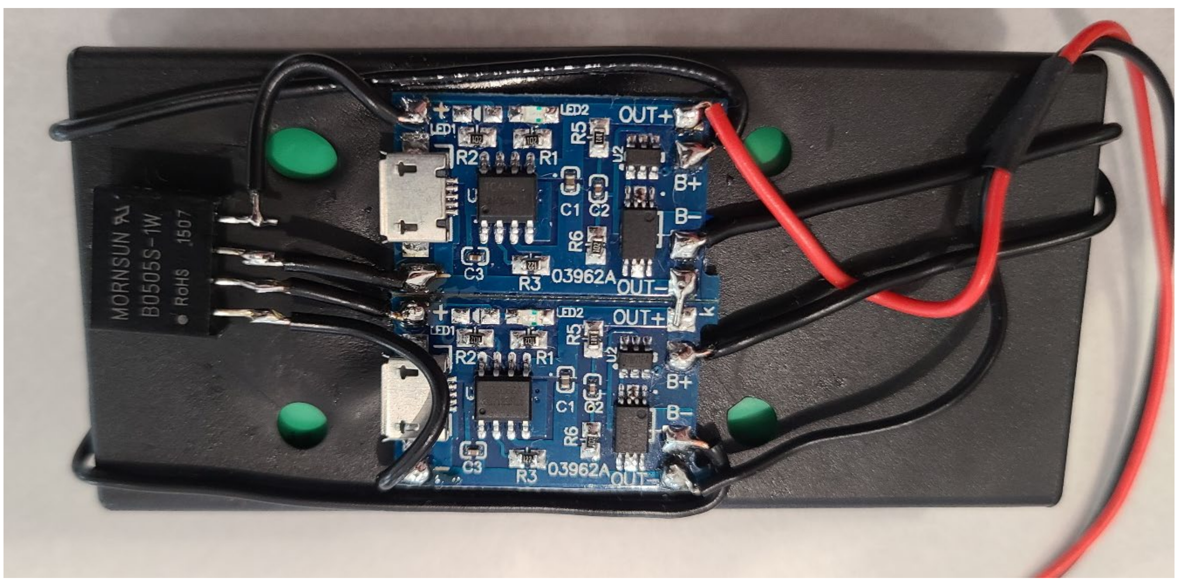


Figure 6-5: *BMS*

6.4 Final Coding Plan

The software for this project will be developed incrementally. This is a simple and popular build model, and it should work well for this project as there are many components. The central server software and the sensor software can be developed at the same time. Most of the software components for the sensor will need to be developed and tested in a specific order.

Sensor software development plan:

1. Configure audio recording with microphone
2. Configure the RTC using the ESP32

3. Configure Wi-Fi using the ESP32
4. Configure GPS using the Adafruit Mini GPS module
5. Configure audio fingerprinting
6. Configure board to send data to test server
7. Configure board to receive camera activation command from test server
8. Configure board to send camera data to test server upon receiving camera activation command
9. Power optimization
10. Reimplement on final sensor hardware
11. Connect final sensor to “production” central server

Central Server software development plan:

1. Configure a basic backend
2. Connect backend to database
3. Create basic development UI
4. Configure fingerprint recognition using test fingerprints
5. Configure multilateration algorithm using test data
6. Configure sound identification algorithm using fingerprints
7. Create sample data that packages audio fingerprints, time, and GPS data together
8. Configure software to take sample data, recognize shared sounds, and then compute a possible location
9. For each possible location try and classify the sound
10. Configure server to receive data from a sensor
11. Configure server to automatically receive data from any sensor that connects
12. Reconfigure algorithms to use actual data from sensors rather than sample data
13. Reconfigure UI to be much cleaner and free of unnecessary development information
14. Configure server to send camera activation command to closest sensor upon detecting a shared sound
15. Configure server to receive camera data from sensors
16. Reconfigure UI to show camera data
17. Clean everything up for use on “production” server
18. Configure “production” server with software

7 Project Prototype Testing Plan

The following section will discuss how we plan to test our prototype, both its hardware and software, first establishing an environment for testing, and then establishing the process for which we will follow for the multiple sections of hardware and software involved with the prototype.

7.1 Hardware Test Environment

The simulation and prototyping phase are a critical step in the design process to test a design for feasibility and identify any defects in the system as a whole or in individual components. These tests will begin in simulation using Multisim and Eagle, and then progress to the experimental phase where a breadboard will be used for prototyping and the various test equipment and resources provided by the Senior Design Lab will be used to conduct the tests themselves. The tests that will be conducted using the various multimeters, oscilloscopes, power supplies, and function generators that are available.

Our hardware testing phase will begin in simulation first in order to see if our setup will operate as we expect in theory. For this simulation portion, the Multisim SPICE software will be utilized due to its ease of availability for this project and due to the hands-on experience that many of the team members have with it from prior projects. In addition to Multisim, Autodesk's Eagle software will also be utilized for its own schematic and SPICE options that are available for use. Creating these simulations of the hardware will allow the DiSEL team to modify and test how each subsystem operates and how the whole system works in tandem, without risking damaging any equipment that is used.

The greatest difficulty for the simulation is the potential that there may not be libraries available for some specific components that we might need for the prototype. If a component is missing from the default Multisim or Eagle libraries, then the component must be added to the respective software(s). There are two options available for adding these components to their designated software, finding premade libraries online for the desired component or manually creating the library ourselves. The simplest solution would be to find a library for that component online that was already made by another user, this would save on time for having to develop our own library. If there are no premade libraries online and we had to develop our own custom-made component library, then we would need to utilize the component's datasheet to provide an accurate simulation environment.

Once the simulation phase is complete, then the prototyping phase must begin. For this phase, individual components are tested to ensure that there are no underlying manufacturing errors with them. If each individual component was not tested for defects and a defective component was integrated into our prototyped system, this would greatly increase the difficulty of troubleshooting the problem and waste valuable development time.

7.2 Hardware Specific Testing

The following section discusses the steps for which we will use to test the different areas of hardware being used. This includes individual component testing, subsystem testing, and a full system test.

7.2.1 Individual Component Testing

The initial individual component testing phase is the most critical phase in prototype testing. All electrical components that are used have a chance of developing defects during the manufacturing process. Some of these defects are minor enough to be ignored, but major defects in the component could cause it to not operate as intended, or not operate at all.

Due to this knowledge, it is very important to individually test the components or in small sections before assembling them into a device prototype to simplify troubleshooting problems as much as possible. If these initial tests were not completed and a hardware problem occurred on the final prototype build, it would be very difficult to assess the problem as the number of possible causes would be enormous. The error could simply be an accidental invalid pin connection, an electrical design flaw, or perhaps one of the many components that were used in building the prototype simply does not work. Testing these components beforehand will help prevent many of these troubleshooting frustrations. This process will involve testing all necessary resistors and capacitors to ensure they are within their proper component value ranges, testing the pins on all ICs and microcontrollers that are used making sure that the individual pins are receiving the correct voltage and that they are outputting correct signals, and using the power supply to ensure all switches are operating as intended.

One method that will be used to test these components are using a multimeter to check the resistance of resistors, and the use of a breadboard to test other components.

Another method that may be used in some cases to test these components is to use them to build a simple circuit that will have an easily known input, output, and setup. For example, if an operational amplifier is needing to be tested, setting up a simple low pass filter circuit and using an oscilloscope will work as a simple test to see if that operational amplifier, and the assorting resistors and capacitors being used here, are functioning as intended.

The audio board was connected to the Teensy board and a short LED based program was uploaded. The program was then performed to check if both boards were operational and if they were connected and soldered properly. An image below shows how the connected boards appear together and show the LED operating. This gives visual evidence of a properly connected board that can be further developed knowing that a simple connection has been established.

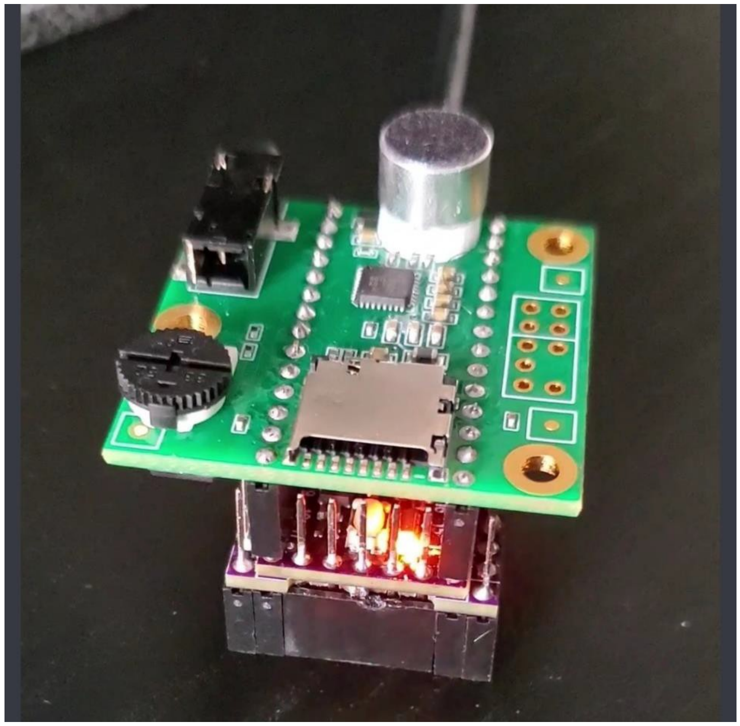


Figure 7-1*: Audio board and Teensy 3.2 development board testing*

7.2.2 Subsystem Testing

Once the individual components, ICs and microcontrollers are tested to ensure they function as expected, the next step in the prototyping process is to build each subsystem of the device on the breadboard so that each section may be tested and added to the larger prototype sequentially.

7.2.2.1 Main Power Supply Subsystem

The first subsystem that will be prototyped and tested is the main power supply system. The main power supply system used for DiSEL is a fairly simple design, utilizing a DC voltage source provided by the rechargeable batteries. A battery holder will be placed on the breadboard and connect to a switch and a DC-DC converter (voltage regulator). Once the power system circuit is built, the batteries can then be placed to provide power to the circuit. The two main testing locations will be at the output end of the switch, to ensure that it is receiving the voltage from the batteries, and at the output end of the DC-DC converter. The DC-DC converter will be the most critical component of the power supply system and will likely be one of the larger components in the overall prototype. Ensuring the inductor functions properly is typically critical to a DC-DC converter's operation.

7.2.2.2 Audio Receiver Subsystem

Next, is to build and test the audio receiver subsystem. This system is responsible for recording the incoming audio signals from the microphone and converting them into a digital signal that the microcontroller can interpret and manipulate. The microphone is connected to an operational amplifier which will be used to amplify the received audio signal and filter out the DC component. Then, the resulting signal will be fed into the analog-to-digital converter to convert the audio signal into a digital signal that the microcontroller can read. Each of these signals will be monitored to ensure that the audio signals are being properly transmitted and interpreted by the microcontroller.

7.2.2.3 Wi-Fi Module

Once the audio signal has been interpreted and the signal analysis done by the microcontroller has determined that the audio signal matches a trigger sound, the data is then sent to the central database via Wi-Fi. The Wi-Fi module at this point should have already undergone a basic test to ensure that its core functions are operational. Upon a trigger, the microcontroller should send the necessary audio signal characteristics to the Wi-Fi module to be sent to the central server. The software used for the database should be able to confirm that a successful transmission was completed between itself and the Wi-Fi module.

7.2.2.4 GPS Module

When the GPS module is used, the GPS receiver will attempt to get a fix on a satellite. Once a fix is acquired, it will receive date, time and coordinate data from the satellite and send this information to the microcontroller through UART which will then send the information to the Wi-Fi module through SPI to be processed in the sensor server along with the audio data. This will help in the process of locating the origin point of the trigger sound. If the Wi-Fi module works properly, the GPS module can be tested by checking if the central server has received the accurate location of the GPS. Alternatively, to test the GPS module without using the Wi-Fi module, a computer can be used to directly read the GPS data from the microcontroller.

7.2.2.5 Camera

Once DiSEL has detected a sound event, the camera will allow the DiSEL user to assess the threat and monitor the situation that caused the sound event. Once a sound event occurs, the microcontroller will send a turn on signal to the camera and the camera will start. In order for the camera module to operate, it requires 3.3 volts for power and a clock signal connected to the system clock pin (XVCLK). Using this clock signal, the camera module generates three clock signals: horizontal reference (HREF), pixel clock (PCLK), and vertical synchronization (VSYNC). These three signals are used to synchronize when the camera captures an image. The camera module can operate in either master mode or slave mode. In master mode, these signals are generated by the camera module and read by the microcontroller. In slave mode, these signals are generated by the microcontroller and read by the camera module. For DiSEL, the camera module will be operating in master mode.

When the camera module captures an image, pixel data will be sent to pins Y0 through Y9. The camera module can produce either a 10-bit image or an 8-bit image. In 10-bit mode, the camera module uses all 10-pixel data pins. In 8-bit mode, the camera module only uses pins Y0 through Y7. As mentioned in the research section, bit depth only affects color accuracy. An 8-bit image can produce 256 assorted colors while a 10-bit image can produce 1,024 distinct colors. Even though this seems like a significant difference in color accuracy, an 8-bit image is still extremely detailed. For this reason, the camera will be operated in 8-bit mode and only pins Y0 through Y8 will be used.

To test the camera module, 3.3 volts will be supplied to the power input pin and a clock signal will be connected to the XVCLK pin. The HREF, PCLK, and VSYNC pins will then be checked to see if they are outputting clock signals. Then pins Y0 through Y7 will be checked to see if they are outputting pixel data in synchronization with the HREF, PCLK, and VSYNC pins.

7.2.2.6 Backup Solar Power Subsystem

Now that the core DiSEL systems have been prototyped and tested, the final subsystem that is built and tested is the backup solar power subsystem. This system was the last to be tested since the DiSEL device can operate without it and its inclusion serves more as a convenience than an integral piece of the overall system. The backup solar power subsystem consists of a solar panel, a battery management system, and a battery pack. The solar panel is responsible for providing the power to charge the battery pack. The photovoltaic cells from the solar panel can directly be tested at its output to determine if it is properly generating power when in contact with sunlight.

The battery management system is responsible for ensuring that the battery pack is charged safely. The battery management system will prevent overcharging, over-discharging, and will ensure that the cells within the battery pack maintain the same level of charge. To test the battery management system, the voltage output to the battery back will be measured. If the system is working properly, the battery pack will stop receiving power once they are fully charged. Next, the output from battery management system to the DC-to-DC converter will be measured. Once the charge of the battery has been depleted past a certain point, the battery management system should no longer provide power to the DC-to-DC converter. Testing the cell balancing feature of the battery management system is difficult because it would take time for the battery cells to become unbalanced. However, testing the battery cell balancing feature is not as important as the other features.

The last component to test in the backup solar power subsystem is the battery pack. The battery pack will consist of two lithium-ion batteries connected in series. This will provide a total output voltage of 7.4 volts. To test the battery pack, the batteries will be charged and then measured to see if they have a voltage of 7.4 volts. The batteries will then be discharged to see if they last the proper number of milliamp-hours.

7.2.3 Full System Test

Now with each individual subsystem tested and proven to be operational and functioning as expected, the next step in the hardware testing process is to run a complete system test. For these tests, all subsystems are connected and operating in tandem as if they were out in a live environment. These tests will include audio signals that will not trigger the alert to ensure that false positive situations are not occurring. In addition to audio signals that will trigger an alert will also be intermittently played to test how well the system handles all the components operating at once and if the system responds correctly to a trigger. If the system worked properly then a trigger should result in the microcontroller sending the audio signal's information and its GPS location to the central server to then do further processing.

7.3 Software Test Environment

Our software testing will begin in a local controlled environment. Our web application will be tested on a local server, and software that involves the analysis and classification of sound events will use simulated data and sounds. This environment being used in our initial phases will give us more control over how the system can function and provide us with baselines (control group) as we continue in our testing leading up to a live environment.

Once our simulated testing has been deemed successful in all area, we will move our testing into a live environment. Our web application will be placed on an internet host. Our data that we will analyze our software will be of a live environment with sounds being produced by living objects, rather than a sound produced by a computer. Though our simulated testing will provide us with our base on which to tweak and adjust the software, this live environment will show how we must adjust when the environment's conditions are not ideal or unexpected events occur when the software is in use. This might include white noise within the audio being received by the sensors, or echoing effects being present when the device is being attempted to operate inside of a closed building.

The software features of our development sensors will be programmed in the following order:

1. Microphone configuration and recording to SD card
2. ESP32 configuration and basic Wi-Fi test
3. Adafruit Mini GPS configuration

4. Real-time clock configuration
5. Fingerprinting algorithm configuration
6. Configure data transmission to local server
7. Configure power saving features
8. Configure data transmission to “production” server

7.4 Software Specific Testing

Several tasks must be accomplished by the software of this product. It must be able to connect with the sensors. It must then receive and access the data from the sensors. The software must then be able to process the data to be usable with our CNN. Then it must accurately locate the sound event and classify the creator of the sound. The software must provide a user interface that will then provide information of both the sensors and the sound data that are captured. These tasks can be tested by acquiring data from the sensor and physically analyzing the results after each task is attempted to be complete by the systems software.

7.4.1 Accessing Web Application

The first step in testing the web application’s accessibility is to ensure that it is being properly hosted on the internet. This will be tested by opening a web browser and entering our web applications address. This test will be passed if the web application is being viewed when the web address is entered.

The other step of accessing the web application is its viewability across a multitude of platforms. To perform this task three web browsers will be used:

1. Firefox
2. Chrome
3. Microsoft Edge

A visual inspection will then be used to test that the web application’s features are viewable across the different web browsers.

The viewability of the web application will also be tested on mobile platforms. The mobile platforms that we will use are:

1. Apple iPhone Safari
2. Android Internet

A visual inspection will then be used to test that the web application’s features are viewable across these mobile platforms.

7.4.2 Creating an Account

To aid in our ability to protect user's information and data, a user will be required to create an account prior to using the product's web application. The user will be given an option when first accessing the web application's homepage.

1. The user will be shown an option to create a new account if one has not already been created.
2. The user will enter their:
    a. Full Name
    b. Email
    c. Username
    d. Password
    e. Re-entry of Password
3. The application will then verify that:
    a. An account does not already exist with the associated email or username
    b. Verify the account was successfully created within the database
    c. Bring the user to their dashboard

This will be tested by trying to create an account with the web application and observer that all the steps laid out above were completed successfully.

7.4.3 Logging into Account

When a user accesses the web application's homepage, they will be presented with a form to enter their accounts username and password. The application will then:

1. Verify the username and password combination are valid
2. One the username and password combination is validated; the application will bring the user to their dashboard.

This will be tested by creating an account and then proceed to try to login with those credentials. We will then observe if the steps presented above are then completed successfully.

7.4.4 Logging out of Account

The user will have a button option on their dashboard to log out of their account. When the user presses the button, the application will then leave the user's dashboard, returning them to the web application's homepage. To gain access to their account will require the reentry of their username and password.

7.4.5 Web Application Features

There are three key features of the web application that will need to be tested. They include:

1. Mapping which provides a map that shows the location of a sound event
2. Classifier which provides the classification information of the creator of a sound event
3. History which provides historical data of maps and classifications of sound events that have been previously analyzed by the user's system.

The mapping and classifier features will be tested by taking a phone and playing a sound with a known distance from the sensors. The web application will then be accessed and will be observed if the data is present on the dashboard. This will be done several times to test the accuracy of the web application's data being provided. This will also allow for the historical feature to be tested to ensure that the data is being properly saved in the database and displayed on the user's dashboard.

7.4.6 Convolutional Neural Network

Testing our CNN as an individual entity is important as discussed earlier. However, we must also test our network as it relates with the system. To accomplish this part of testing we must do so in phases. The first phase will involve:

1. Input simulated sound events being used as inputs into our system.
2. Place the outputs of our network into a csv file.
3. A visual inspection will occur. Each outputted prediction from the network will be compared to the expected classification for each sound.

If the network performs at a comparable rate to that of is individual testing, we will deem our connection in the system to be partly successful.

The next phase for testing our network within the system is to use live sound events. To implement this phase our system will:

1. Be placed into a live environment where sound will enter the system's sensors.
2. The sound will be analyzed through our system as well as by our own ear.
3. Classify the sound on our own while the system makes its prediction.
4. Once the network has completed its prediction, the classification made by us and the one made by our network will be compared.

This will be done multiple times. If the accuracy of the network's responses is comparable to that of its isolated test, we can officially concur that the network is successfully connected to our system.

7.4.7 MFCC Function

One important aspect of our program portion of the project is our function that will produce MFCC of incoming audio data to allow our CNN to classify. To test this function, we will take several sound clips that have had MFCCs produced and verified for them. We will then pass these clips through our function. We will then give a visual inspection to the created spectrograms and compare them with the ones that have previously been verified for these clips. If matching images our produced by our function, we can ensure that our spectrogram function correctly.

7.4.8 Server API

To test our connection between our frontend and backend, we will use Mongo a command line feature of MongoDB and simulated data. This will provide a visual of our database, while simulated data will be passed through the system, we will then observe if the data is accurately being represented in our database and being recorded in the current tables.

To test that our backend is correctly connected to our frontend, we will pass several queries from the user's dashboard to retrieve data from our database. If the data displayed on the user's dashboard matches that of which is represented in the database, we will have verification of a working connection with our API.

8 Administrative

The following section discusses all the administrational areas of our project. In this section a detailed budget is established along with how we will finance our project. A set of time stamped milestones are then created to help our group stay on task.

8.1 Project Budget and Financing

Table 8-1 shows the projected budget and financing for the project.

Part	Quantity	Price	Total
Teensy Audio Shield	3	$14.25	$42.75
Printed Circuit Board	3	---	$97
MK20DX256VLH7 (MCU)	3	$8.66	$25.98
Solar Panel	3	$50	$150
Lithium-ion 18650 battery	6	$6	$36
GT-U7 GPS Module with NEO-6M GPS Receiver	3	$10.98	$32.94
ESP32-WROOM-32D (Wi-Fi Module)	3	$4.50	$13.50
LMZM23600V3SILT (5V Voltage Regulator)	3	$5.10	$15.30
Misc Components (Res, Caps, Diodes, Fuse, etc)	----	----	$50
Overall Cost			$463.47

***Table 8-1**: Project Budget and Financing*

Managing the budget of the development was critical for the DiSEL team. One of our main goals when designing the device was to keep the price down as much as possible. This meant limiting any excess or inefficient equipment or components as possible. Limiting the design was especially important for keeping the PCB costs down, as PCB cost increases greatly as its size increases.

Another important financial decision the DiSEL team had to make when beginning our initial research and design phase, was the potential of a sponsor. Having a sponsor would allow the team more flexibility with our design choices, allowing us to utilize higher performance equipment and even expanding on the initial idea to include more modules and devices to the network. However, after careful consideration, the team decided it would be best to forgo the sponsorship route as it would allow us to keep autonomy and control over our design and goals.

8.2 Milestones

Table 8-2 shows the list of our project milestones that are known at this time.

Task	Start	End
Research		
Group Formed	1/10/2020	1/20/2020
Divide and Conquer V1	1/10/2020	1/31/2020
Initial D&C Group Meeting	2/5/2020	
Divide and Conquer V2	1/31/2020	2/14/2020
Power	1/31/2020	2/14/2020
Communication	1/31/2020	2/14/2020
Sensors	1/31/2020	2/14/2020
Sound Labelling	1/31/2020	2/14/2020
Sound Location	1/31/2020	2/14/2020
Sound Detection	1/31/2020	2/14/2020
Web Application	1/31/2020	2/14/2020
Software-Hardware Integration	1/31/2020	2/14/2020
Assignment on Standards	3/9/2020	3/13/2020
60 Page Draft	2/15/2020	3/20/2020
Design		
Power	2/15/2020	4/3/2020
Communication	2/15/2020	4/3/2020
Sensors	2/15/2020	4/3/2020
Sound Labelling	2/15/2020	4/3/2020
Sound Location	2/15/2020	4/3/2020
Sound Detection	2/15/2020	4/3/2020
Web Application	2/15/2020	4/3/2020
100 Page Draft	3/20/2020	4/3/2020
Final Document	4/3/2020	4/21/2020
Implementation/Testing	8/21/2020	11/30/2020
Power	8/21/2020	TBD
Communication	8/21/2020	TBD
Sensors	8/21/2020	11/04/2020
Sound Labelling	8/21/2020	11/05/2020
Sound Location	8/21/2020	TBD
Sound Detection	8/21/2020	11/15/2020
Web Application	8/21/2020	11/15/2020
Final Project Presentation	12/01/2020	12/01/2020

***Table 8-2**: Milestones*

9 Conclusion

Through our process of designing and developing our Distributed Sound Event Location prototype, several challenges were encounter and overcome by the research of individual sections of the project, along with some trial and error. The design requirements that have been laid out within this document have be taken into consideration for all hardware and software to be used and developed for our prototype.

The Diesel system contains multiple system of both hardware and software, that when integrated as we have discussed, will create a product that will have many applications throughout a multitude of industrial and residential landscapes. The idea of using machine learning in sound detection, though not new when involving speech recognition, can still be considered in its infancy when dealing with sound outside of speech. Though there can be pitfalls within sound classification, careful consideration has been made to insure a successfully and functioning product will be created.

The workload for this project has been divided evenly between group members. Though some of the division is based on the members' individual computer and electrical engineering disciplines, there is also carry over of task between these disciplines based on the experience within each section, and the desire and interest each member has for those areas. This way of task division will allow us to both improve on skills we have experienced prior to this project, as well as learn and develop new skills that can be taken into our future endeavors.

As we reflect upon the Diesel system. This project offers the ability to experience knowledge gained through our classes and new areas within our engineering disciplines in a hands-on environment, while also offering the opportunity to build something that can be useful to a multitude of people.

Appendix A - References

1. *Lithium-Ion Battery.* Clean Energy Institute, University of Washington, https://www.cei.washington.edu/education/science-of-solar/battery-technology/. Accessed 13 February 2020

2. *BU-808: How to Prolong Lithium-based Batteries.* Battery University, https://batteryuniversity.com/learn/article/how_to_prolong_lithium_based_batteries. Accessed 13 February 2020

3. *Nickel Metal Hydride (NiMH).* Johnson Matthey Battery Systems, 2016, http://www.jmbatterysystems.com/technology/cells/nickel-metal-hydride-(nimh). Accessed 13 February 2020

4. *NiMH Battery Charging Basics.* PowerStream Technology, 31 July 2019, https://www.powerstream.com/NiMH.htm. Accessed 13 February 20, 2020

5. *Battery Management Systems*. Engineering.com, 2020, https://www.engineering.com/productshowcase/batterymanagementsystems.aspx. Accessed 19 March 2020

6. *Why Proper Cell Balancing is Necessary in Battery Packs.* Battery Power, https://www.batterypoweronline.com/blogs/why-proper-cell-balancing-is-necessary-in-battery-packs/. Accessed 19 March 2020

7. Beck, Anton. *Lithium Iron Phosphate vs. Lithium-Ion: Differences and Advantages.* epec Engineered Technologies, 20 September 2019, https://blog.epectec.com/lithium-iron-phosphate-vs-lithium-ion-differences-and-advantages. Accessed 18 March 2020

8. *Photovoltaic Cell*. Energy Education, 25 June 2018, https://energyeducation.ca/encyclopedia/Photovoltaic_cell. Accessed 18 March 2020

9. *Types of Solar Panels*. energysage, 21 January 2020, https://www.energysage.com/solar/101/types-solar-panels/ Accessed 19 March 2020
10. *Understanding The Digital Image Sensor*. Lucid Vision Labs, https://thinklucid.com/tech-briefs/understanding-digital-image-sensors/ Accessed 20 April 2020
11. *Resolution of Sensors*. Vision Doctor, https://www.vision-doctor.com/en/camera-technology-basics/resolution-of-sensors.html Accessed 20 April 2020
12. *Digital Camera Sensors*. Cambridge in Color, https://www.cambridgeincolour.com/tutorials/camera-sensors.htm Accessed 20 April 2020

13. *What is Bit Depth*. Tech-Ease*win https://etc.usf.edu/techease/win/images/what-is-bit-depth/ Accessed 20 April 2020
14. Kun, Atilla. *Exposure*. Exposure Guide, https://www.exposureguide.com/exposure/. Accessed 20 April 2020

15. Davis, Nick. *Introduction to Temperature Sensors*. All About Circuits, 05 June 2017 https://www.allaboutcircuits.com/technical-articles/introduction-temperature-sensors-thermistors-thermocouples-thermometer-ic/. Accessed 19 March 2020

16. *Seebeck Effect*. Search Networking, December 2008 https://searchnetworking.techtarget.com/definition/Seebeck-effect. Accessed 19 March 2020

17. Sattel, Sam. *What is a Voltage Regulator?* AUTODESK, 2016, https://www.autodesk.com/products/eagle/blog/what-is-a-voltage-regulator/. Accessed 19 March 2020

18. *µA7800 Series Positive-Voltage Regulators.* Texas Instruments Incorporated, May 1976 – Revised May 2003, 2004, https://www.sparkfun.com/datasheets/Components/LM7805.pdf. Accessed 19 March 2020
19. *OV2640 Color CMOS UXGA. OmniVision*, 23 February 2007, https://www.uctronics.com/download/OV2640_DS.pdf. Accessed 20 April 2020
20. Davis, Sam. *Switching Voltage Regulator ICs FAQ.* Power Electronics, April 2009, https://www.powerelectronics.com/technologies/power-management/article/21856505/switching-voltage-regulator-ics-faq. Accessed 19 March 2020

21. *Electrical Relay*. Electronics Tutorials, https://www.electronics-tutorials.ws/io/io_5.html. Accessed 19 March 2020
22. *Bipolar Transistor.* Electronics Tutorials, https://www.electronics-tutorials.ws/transistor/tran_1.html. Accessed 3 April 2020
23. *Transistor as a Switch.* Electronics Tutorials, https://www.electronics-tutorials.ws/transistor/tran_4.html. Accessed 3 April 2020
24. *The MOSFET.* Electronics Tutorials, https://www.electronics-tutorials.ws/transistor/tran_6.html. Accessed 3 April 2020
25. *MOSFET as a Switch.* Electronics Tutorials, https://www.electronics-tutorials.ws/transistor/tran_7.html. Accessed 3 April 2020
26. Briones, Alexander. *The Different Types of Mics and Their Uses*. Gearank, 18 December 2015, https://www.gearank.com/articles/types-of-mics. Accessed 3 April 2020

27. *Ribbon Mics — How They Work and When to Use Them.* Sweetwater, 28 April 2016, https://www.sweetwater.com/insync/ribbon-microphones-how-do-they-work/. Accessed 3 April 2020
28. Mitchell, T. (1997). Machine Learning. McGraw Hill. p.2. ISBN 978-0-07-042807-2. Accessed 18 March 2020

29. Piczak, Karol J. Environmental Sound Classification with Convolutional Neural Networks. Institute of Electronic Systems, Warsaw University of Technology. 2015 IEEE International Workshop on Machine Learning for Signal Processing, 17-20 September 2015, Boston, USA. ISBN 978-1-4673-7454-5. Accessed 18 March 2020

30. Tanenbaum, Andrew S.; Steen, Maarten van (2002). Distributed systems: principles and paradigms. Upper Saddle River, NJ: Pearson Prentice Hall. ISBN 0-13-088893-1.

31. Adit, Deshpande. A Beginner's Guide to Understanding Convolutional Neural Networks., Engineering at Forward, UCLA CS, 20 July 2016, https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks. Accessed 18 March 2020

32. Kapil, Divakar. Stochastic vs Batch Gradient Descent. Medium, 6 January 2019, https://medium.com/@divakar_239/stochastic-vs-batch-gradient-descent-8820568eada1.

33. Dabbura, Imad. Gradient Descent Algorithm and Its Variants. Towards Data Science, 21 December 2017, https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3.

34. Cannon, L.W.; Elliott, R.A.; Kirchhoff, L.W.; Miller, J.H.; Milner, J.M.; Mitze, R.W.; Schan, E.P.; Whittington, N.O.; Bell Labs, Spencer, Henry; Keppel, David; EECS, UC Berkeley CS&E, University of Washington; Brader ,Mark. Recommended C Style and Coding Standards. https://www.doc.ic.ac.uk/lab/cplus/cstyle.html.

35. Tutorialspoint. C – Header Files. https://www.tutorialspoint.com/cprogramming/c_header_files.htm.

36. Smith, J.O. Spectral Audio Signal Processing, http://ccrma.stanford.edu/~jos/sasp/, online book, 2011 edition http://ccrma.stanford.edu/~jos/sasp/, online book, 2011 edition

Appendix B - Copyright Permissions

Contact the Electronics Tutorials Team

We always encourage you to share your ideas and improvements with us, so if you have any questions about our Electronics Tutorials website, please feel free to contact us using the form below. Many thanks for your show of support.

Email (required)

blaguerre47@knights.ucf.edu

Message (required)

Hello, I am a student at the University of Central Florida. My team and I are currently working on our senior design project and would like to use some of your images in our report. I was hoping to get in contact with someone who could give us permission to use these images.
The images we would like to use: https://www.electronics-tutorials.ws/filter/filter_2.html

Thank you for your time,
Brandon LaGuerre

Submit

Your Email (required)

blaguerre47@knights.ucf.edu

How can we help?

Hello,

I am a student at the University of Central Florida. My partners and I are working on a senior design project where we will be trying to detect and locate specific sounds. Our project concept features some similarities to your Guardian Active Shooter Detection system and we were wondering if we could get permission to reference some information about your product and also use some of the images that are on your website.

Thank you for your time,
Brandon LaGuerre

Subject

UCF Senior Design Permission Request

SEND

To corporatepr@rtx.com Bcc

Cc

UCF Senior Design Project - Raytheon Media Permission Request

Hello,

I am a student at the University of Central Florida. My group partners and I are currently working on our senior design project which involves identifying desired sounds and locating the origin of the sound. After doing some research, we noticed that your Boomerang III shooter detection system had some similarities to what we were looking to design. We were wondering if we could get permission to utilize some of Raytheon's Boomerang images on our senior design report and also reference some information about your product as well.

Thank you for your time,
Brandon LaGuerre

What is the nature of your inquiry?*

General Information

Additional Details

Please be as specific as possible so we can route your information to the right person.

Hello,
I am a student at the University of Central Florida. My group members and I are currently working on our senior design project in which your ShotSpotter product has some features that are similar in concept to what we were looking as making. We were wondering if we could have permission to utilize some of your images and reference some technical aspects of your product for our school project.
Thank you for your time,

To sales@louroe.com Bcc

Cc

UCF Senior Design Project - Louroe Media Permission Request

Hello,

I am a student at the University of Central Florida. My group partners and I are currently working on our senior design project which involves identifying desired sounds and locating the origin of the sound. After doing some research, we noticed that Louroe's gunshot detector software shares some similarities to the concept that we are looking to design. We were wondering if we could get permission to reference some information about your company and your product on our senior design report.

Thank you for your time,
Brandon LaGuerre