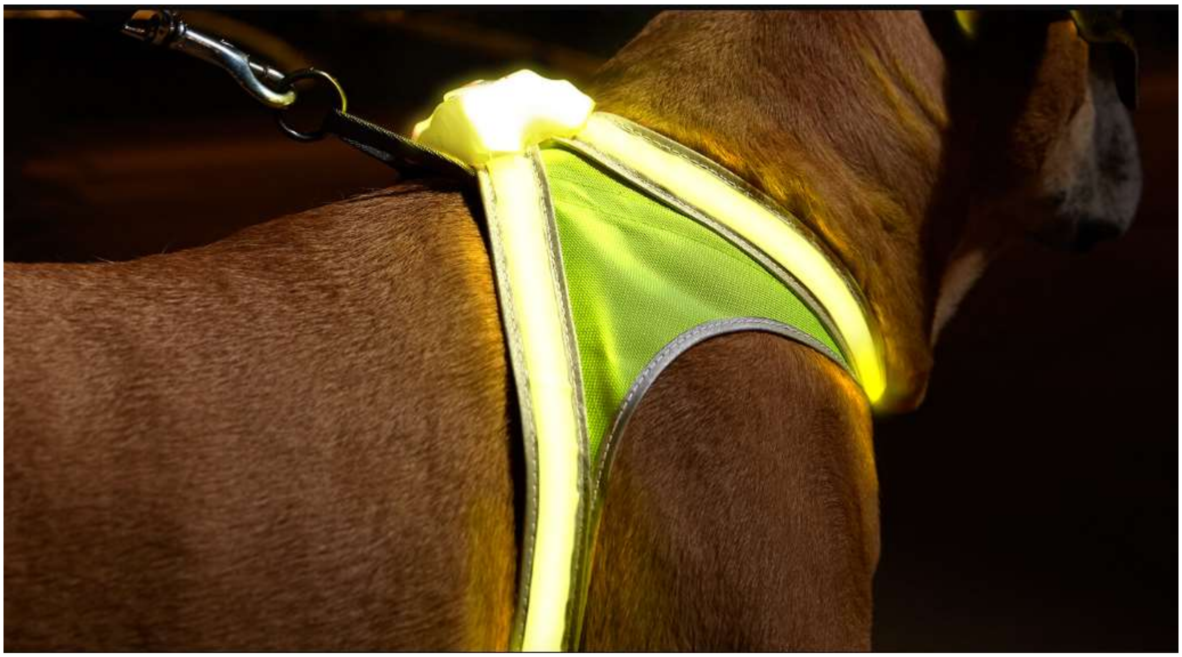
Senior Design Project

PetAid Harness



University of Central Florida

Department of Electrical Engineering and Computer Science

Dr. Samuel Richie

Senior Design 1

Term: Summer 2020

Group 8

Pedro Rocha *Electrical Engineering*

Pascal Bahms *Electrical Engineering*

Edgar Gonzalez *Electrical Engineering*

Adam Henderson *Computer Engineering*

Contents

1. Executive Summary

The concern about the lack of information regarding health and location of a pet owner's dog is the main problem and an overall issue for the general public audience in which it is decided to be approached for our senior design project. The smart harness called "PetAid Harness" is the best object or method for an electronic device to be suitably placed and attached to the dog while providing an improved level of comfort compared to placement of a smart collar around the neck. The goal of this project is to monitor the temperature, heart rate, location, and position of any dog by wireless communication. This information obtained is constantly collected and displayed on a phone app designed and implemented specifically with the PetAid harness. The purpose of this report is to outline the research, budget, testing, and implementation as many other aspects that must be considered to produce a project of this importance and scale. Information is obtained and explained in this project report with careful consideration in detail, while within a showcasing environment to complete our senior design capstone requirements.

The overall project design of the PetAid harness can be divided into two parts. The first half being the actual harness and the electronic components attached together, and the second half of the design being the implementation of an IOT platform and smartphone application with WiFi communication to store and display information. The reason for utilizing a mobile application such as ThingView is that it can be downloaded and used on any smart phone for better connectivity and it is also freely accessible with the PetAid harness utilizing a Internet-of-Things website such as ThingSpeak. Thus, the PetAid Harness involves several features to achieve this goal. These features can include sensors such as a Thermometer, Heart Rate module, Accelerometer module, GPS and GSM module, and WiFi/Bluetooth module. The GSP and GSM module was not included for the final product design and demonstration. Each sensor contains several important specifications that must be considered during the design of the project to meet the desired needs while remaining at low cost.

For a project design to be constructed, the project idea must have a specified beginning to create and initiate it. One of our group members has a family member who has a service dog by her side and the importance of how the service dog has been not only to the owner, but also how service dogs are crucial to any pet owner throughout the world. It can also be for any type of dog considering a pet owner who is concerned about their dog. The idea is to also ensure the dog is in healthy condition, and in a known location. If the dog has a medical condition that requires more supervision than the owner can typically provide, this can provide vital information to ensure that the dog receives care faster in times of emergency. The pet owner can be informed with an alert system such as a push notification and email on their phone to indicate the medical treatment that is required. The Thermometer and Accelerometer help provide the current body temperature, motion and position of the dog to be sent to the IOT platform and a compatible phone app. GPS module is an option that can be included to supply the location and coordinates of the dog, should there be a situation where the dog must be found for treatment or if the dog is at an unknown location. A Heart Rate module helps in checking for an abnormal heart rate of the dog, such as when having a seizure and provides a graph of their daily heart rate patterns. WiFi/Bluetooth is also needed along with a GSM module to send all these various vitals reported from the sensors integrated to the PCB (Printed Circuit Board). The PCB (Printed Circuit Board) is designed to be the controller of the PetAid Harness and will send this information to the ThingSpeak cloud for a smartphone device of the user to access this information. Overall, the outcome of the senior design project hopes to become helpful and bring peace of mind to dog owners and give them a better chance to ensure that their dog is safe and healthy.

2. Project Description

The PetAid Harness is a smart harness designed to be a biometric and location tracking device worn by any size of dog, as desired by the pet owner. There are many practical uses of this type of device, whether it is to help improve a dog's fitness and facilitate weight loss, or more closely monitor a pet dog with a medical condition that could require a quick response time in order to properly care for them. The PetAid Harness provides vital information needed by the pet owner to easily track and maintain the health and fitness of their dog, while also keeping this data collected for a veterinarian to observe and record the dog's health information during checkups to help in any diagnoses that may be needed. Since the app is meant to be accessible for dog owners of all levels of technological literacy, the interface is made to be user friendly to ensure a short-to-nonexistent learning curve for any pet owner when utilizing the PetAid Harness app such as ThingView. The PetAid Harness has several important specifications in allowing the user to monitor the current status or condition of their dog in real-time. Consequently, the several aspects that will be important are the desired requirements for the smart harness and electronic components such as the sensors in order to meet the budget of our group; low cost while preserving the power use, sensor accuracy, and ability to multitask. Size and weight dimension must also be taken into consideration to achieve a smart harness that is compact, comfortable, and portable. Dimension size includes length, width, and height (LxWxH) for all electronic components such as all sensors, casing, and harness. Notable solutions are explored which include if it is possible for these sensors to be surface mounted onto the PCB board of the PetAid Harness. The Temperature sensor will provide the body temperature of the dog, and the heart rate module continuously keeps track of the heart rate of the dog to make sure it is in normal condition. In the future design, once the GPS module is implemented it has the ability to function alongside the accelerometer for position and motion of the dog, which provides the pet owner an extended amount of awareness for where their pet can be located. A microcontroller such as the Arduino Uno MCU, is the brain of the PetAid Harness due to connecting all the data received from the sensors and then processing it to the smartphone device of the user or pet owner. The Alkaline battery is currently the best battery source option now and was decided the best option to be connected as a power supply to the PCB board of the PetAid Harness.

Various users are considered with project design of the PetAid Harness. These users consist of parents assisting their child in caring for their first pet, a working pet owner who must leave their pet dog at home without supervision, and a veterinarian safe-keeping several puppies to better supervise their current condition, among others.

2.1. Project Motivation and Goals

The goal of this project is to design a harness that is adjustable to fit the size and shape of a wide range of dogs, including those that have a physical impairment or other medical conditions. With the great extent of pet ownership in the United States, as described by the 2019-2020 National Pet Owners Survey conducted by the American Pet Products Association (APPA), a total of about 67% of U.S. households own a pet and this harness can be of high demand to many of these pet owners.

This device can have multiple features integrated with the harness, these features include GPS, heart rate module, thermometer module, accelerometer module, vibrating tactile feedback, GSM module, and WiFi/Bluetooth transmitter. GPS allows the user or pet owner to track their pet with the locator through the phone. Heart rate monitor warns if there is a sharp rise or drop in heart rate, which will help if the pet is having a seizure or other medical emergency while the owner is not present. Thermometer can also be useful by giving the user plenty of information regarding their pet's body temperature to determine whether it is normal or at an abnormal condition. The accelerometer module will help display the position of the pet while it is in motion. All this information will be transmitted to and received by our own created mobile phone application using a WiFi/Bluetooth module integrated alongside the GSM module for cellular compatibility. GSM module becomes very useful with utilizing the PetAid Harness during situations where the pet owner is at work or substantial distance away from the dog where Bluetooth is incapable due to its limited range of distance. This GSM module supports network connectivity between the smartphone of the pet owner and the PetAid Harness in order to display proper heart rate, temperature, and location tracking information on the phone application. This network connectivity provides an almost unlimited distance range if cellular service is available with good telecommunication signal.

Placement of all these sensors will be attached on the harness around the neck and chest in a 3D printed case that protects the components from the elements. The electronic components must be sealed from dirt, dust, and water. Weather along with temperature is also another constraint that must be considered to prevent high and low temperatures from affecting all the electronic components in terms of their maximum and minimum temperature operation rating. The final design of the harness must be lightweight, portable, low cost, easy to use, and receive/transmit accurate data. Size and weight of the PetAid Harness must be considered with the final project design. If the electronic components are too immense in size and have an abundance of weight, the PetAid Harness will not only be considerably large for a pet owner to carry around with portability, but also the comfortableness will be of low standard and unpleasant for their pet.

The motivation for this project is due to creating a device that could monitor a sick pet that has a disability such as impairment from blindness, deafness, or disease. The idea or goal is to keep track of the heart rate of the pet, along with other additional features that can further employ usage of a device that can keep an owner more aware of the current condition of the pet, and determine if heart rate, temperature, position are abnormal. Other uses can also be used, such as the occurring scenarios where pet owners leave their animals left inside the vehicle while stopping at a small grocery store or another location. If needed, once the body temperature or the temperature inside the car is not at the recommended temperature, a notification will be sent through the designed mobile phone application only within Bluetooth 5.0 range which can stretch up to 400 meters in a clear line of sight. A notification can also be sent through the mobile phone application at any desired distance between the user and the PetAid Harness with the usage of a GSM module, if there is cellular data communication available.

2.2. Project Objectives

The general project objective of the PetAid Harness is to allow many pet owners the opportunity to successfully and efficiently monitor their dog's activity daily. To be completely successful for our project design, the PetAid Harness must demonstrate itself as simple to access at any location where there is an internet connection. Accommodation for the general project objective is achieved through smaller objectives laid out to better plan the project scope within the last two semesters of the Electrical Engineering track. The smaller objectives also provide targets to reach and attain, by adding all these smaller objectives to sum up the overall project objective. These smaller objectives are listed below without any certain order of sequence.

1. Durability:

    a.) Long-lasting condition of the PetAid harness must be achieved. Through long durability the better choice of reliability is given to the user to consider utilizing the PetAid harness within a longer period as their first choice of smart harness compared to other competitors.

    b.) PetAid Harness battery life must be long-lasting and must meet the max capacity of the components attached. The best battery type is considered to meet these expectations.

2. Lightweight:

    a.) Pet owners realize the amount of weight they must carry around daily. Being able to keep the PetAid harness lightweight, is helpful to allow elderly pet owners to move around and during attach/removal of the smart harness.

    b.) The pet's perspective is also taken into consideration to provide better comfort while being worn around their neck.

3. Compact:

    a.) Storage can be limited at certain locations and can also be limited at a pet owner's home such as an apartment. PetAid Harness is small and compact to find an easy storage location when not in use.

    b.) Traveling is necessary for a pet owner, whether it is to go out for a walk or going out for a ride in their car to the store. Thus, PetAid Harness must be compact to remove any trouble in which the pet owner can experience, and instead worry about other important items.

4. Safe:

    a.) Standards are designated with safety requirements that must be met according to the safety and health of any living creature, such as a dog's safety when wearing the PetAid harness.

    b.) The PetAid Harness will be worn by the dog, and thus must meet safety and health requirements to ensure it is not too tight or loose on the dog which can cause choking or pinching.

5. Resistant:

a.) PetAid Harness design is rugged to withstand harsh weather or environment conditions. These weather conditions can occur if the outside temperature is too hot or too cold, affecting the maximum or minimum operating temperature of a sensor or electronic components.

b.) During rainy weather conditions or when washing the harness, where the PetAid Harness can get wet, the smart harness must be rugged to be water-resistant. This will protect all the electronic components inside the PetAid Harness from any type of damage.

6. Data transmission:

a.) Monitor and Tracking information obtained from sensors positioned on PetAid Harness must have access to and from a mobile phone application. The Pet owner or user should have access to this information easily.

b.) Access is available all the time if the PetAid Harness is powered on.

c.) Engineering requirements involve obtaining accurate real-time data information and then given to the pet owner or user.

2.3. Requirement Specifications

The first requirement specification is the size of the project. The overall size will be small to fit around an average sized dog. The important part being that the components must be as small as possible without sacrificing quality. The idea of designing a Harness style, instead of a dog collar style provides better ways of arrangement when placing the microcontroller.
The second requirement specification is the weight of the project. The Comfort level of the harness is better when the harness is low in weight. For similar reasons as described in the size requirement specification, the components to research must also be low in weight.

There are project constraints that can occur when the pet harness is being worn and utilized. One possible constraint is the limited distance that can be reached for the user to communicate with the pet harness. Such as during the day when the owner is working away from the house, the location where the pet is located could be too far away. This long-distance range will be desired for the owner to receive any information about the condition of the dog and is not possible with only WiFi/Bluetooth integrated. There are two possible ways that the PetAid Harness can transmit information to the pet owner. One way is an IOT website database to collect data from the sensors on the harness, then send it to the cloud where the data can then be manipulated and displayed. Second method is performed by implementing a self-created mobile phone application to access information from the PetAid Harness and display it to the user’s phone. Accessibility to a designed mobile phone application platform is important because a message or notification must be sent to the smartphone of the pet owner or user of the PetAid harness in case any data monitored and collected is in abnormal condition from the heart rate sensor.

Another project constraint is the durability of the harness. Most dogs have the likelihood to scratch or destroy objects, especially if the harness is uncomfortable. The reason for comfortableness is a factor of why our group chose to use a harness instead of a collar, as well as adjusting a better fit.

If the harness is uncomfortable and fragile to being ripped or broken, then the harness would be useless if it cannot withstand the rough play put on the harness by the pet. The operating temperature of the electronic components must also operate at a high and low temperature ranges in order to withstand the various seasonal weather conditions including being weather resistant. The house of quality tables describes both the user and engineering requirements of the PetAid harness in terms of accuracy of the sensors and user interface. Notation for house of quality is included to determine whether a certain requirement has a positive effect or a negative effect to the overall performance of the PetAid Harness.

The last project constraint takes place during the project design testing process. Since the PetAid Harness involves a dog or another animal, testing is not accepted on animals. Animal testing is not a good alternative due to Universities not allowing animal testing when utilizing a project design such as the PetAid Harness.

2.3.1. Size

Overall size of the PetAid Harness will be small to fit around an average sized dog. The important part being that the components must be as small as possible without sacrificing quality. Research is developed for the different sensors that are used, such as the idea of designing a Harness instead of a dog collar provides additional ways of arrangement when deciding how to place the microcontroller. The biggest component of the project design is most likely the main battery used for power supply of the PetAid Harness. Depending on the type of battery will determine its size, in which size is a factor that is recognized for the PetAid Harness design. Table 1 shown in the following page represents detailed specifications on the dimension size of the various modules that will be utilized for the PetAid Harness. These dimensions are subject to change for the final project design according to prototype test results. The 3D case is ultimately built to store all these electronic components and conforms to the smallest case size possible once all modules are decided and arranged.

2.3.2. Weight

Comfort level is highest when the PetAid Harness is low in weight. For similar reasons as described in the size requirement specification, the components to research must also be low in weight for related causes such as providing less neck strain put on the dog when the PetAid Harness is being worn. The pet owner or user is also given a better lightweight harness to be able to carry and move around when traveling to different locations. Providing better comfort keeps the pet in healthy condition, safe, and prevents from the dog scratching or damaging the PetAid Harness due to neck pain. Likewise, the battery is the biggest component in size just like it is also the heaviest component. The 3D case can also be heavy due to its wood material which is sought to be heavy. Alternative materials or casing can be considered if needed to be changed for the final project design, if the 3D case is too heavy to meet our standards. Table 1 as represented below describes both the weight and size specifications containing important information for the overall health of the dog. Therefore, the overall design of the PetAid Harness is made to be small and low in weight to meet the pet owner's expectation.

Component Name	Size (mm) (LxWxH)	Weight (grams)
GPS Module	35.306x30.48x6.35	8.754
Accelerometer Module	21.59x19.05x2.0	1.4
Temperature Module	21x21x2	2.93
Heart Rate Module	16x16x3.2	3
Bluetooth Module	20.4x41x4	4.4
Battery (mm^3)	60x50x7.9	47
Microcontroller Module	3.35x2.2	1.4
3D Casing	230x185x69	200
Collar	304.8x19.05	90.72

Table 1. Pet Harness component specification

2.4. House of Quality Analysis

The PetAid harness must have a house of quality, such as what is needed for most project designs. There are two types of requirements that are incorporated for the PetAid Harness project design and shown in Table 2. These two requirements are described from both the engineering and user perspective. Engineering requirements are necessary as part of the house of quality analysis to define project requirements and ensure proper research is performed for hardware components and software applications. User requirements are included to achieve standards expected and desired by the user or pet owner. User requirements are also set within the house of quality to ensure the interface has easy accessibility for the pet owner to have an extra sense of assurance of their pet's condition. Consideration is put into place for pet owners who do not want to worry about operating a complex smart harness and focus only on their pet's health. The build of these components and applications will be designed to fulfill the best possible requirements of power use, sensor accuracy, cost, simple user interface, reliability, and maintenance.

<u>Quality requirements</u>

			Engineering Requirements			
			Power Use (<10 Watts)	Sensor Accuracy (Temperature) (±2%)	Sensor Accuracy (Location) (±7%)	Sensor Accuracy (Heart rate) (±1%)
User Requirements		Polarity	-	+	+	+
	Cost	-	↓	↓	↓	↓
	Ease of use	+	↓	↓	↓	↓
	User Interface	+	↓	↑	↑	↑
	Reliability	+	↑	↑	↑	↑
	Maintenance	-	↑	↓	↓	↓

Table 2. House of Quality

Strong Positive Relationship	↑↑
Positive Relationship	↑
Negative Relationship	↓
Strong Negative Relationship	↓↓
Positive Polarity	+
Negative Polarity	-

Table 3. Correlation Factors

There are correlation factors also incorporated into the house of quality, as represented in Table 3. These factors take into consideration whether there is a positive or negative relationship between the engineering and user requirements.

Block Diagrams:

Shown in Figure 1, our division of focus was that Pedro will be dealing with selecting the thermometer and the heart rate monitor, Pascal will decide the GPS module and the microcontroller, Edgar will decide the accelerometer and the power supply interface, and Adam will decide the wireless communication module, and handle the creation of the phone application. This was soon proven to be inaccurate in delegation, because we have touched into everyone else's areas in some way or another, whether it was to vote on the product we think is best to use for the PetAid Harness, or having suggestions to change the product selected to either fit the budget or to otherwise make the project progress more smoothly in the long run before we commit too strongly on any particular area of the production. While still in the stages of pre-production, we have had examples of Pascal focusing into the budget, ethics, and standards surrounding the product, Edgar looking into the selection of the microcontroller to assist Adam, and Pedro working across all regions of the project, giving assistance where it is necessary.
In Figure 2, we see some of the internal behavior of the software that will be running on the microcontroller of the PetAid Harness, and how it will interact with the phone application.

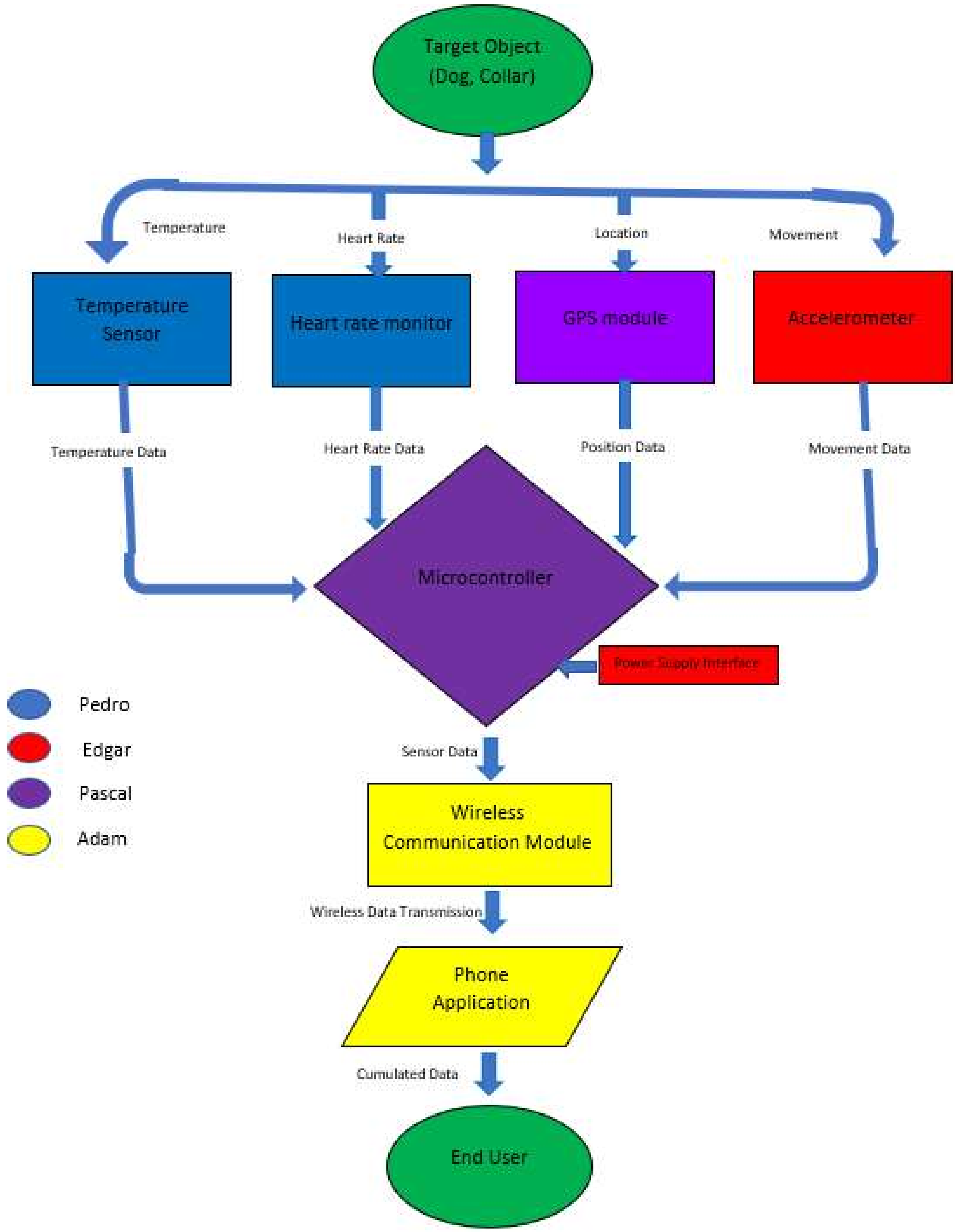


Figure 1. Hardware Block Diagram

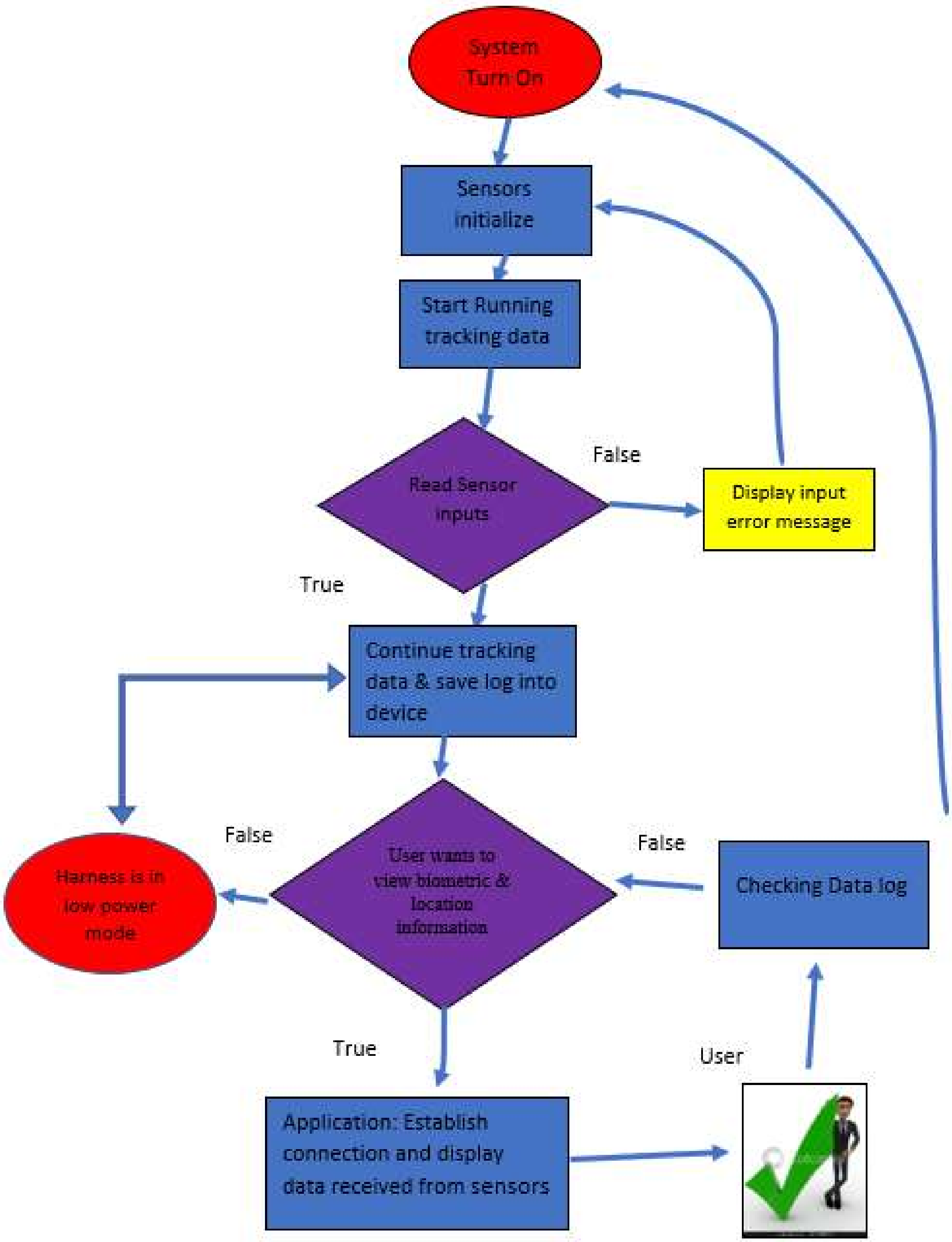


Figure 2. Software Block Diagram

The following block diagrams displayed in Figure 1 and Figure 2 provide a clear and concise structure to provide the team for future planning on parts needed throughout the design process. Figure 1 provides a drawing created by the group for comprehension of the hardware side. The hardware section begins with the electronic components needed, such as the sensors, until finally the end user. Figure 2 illustrates a diagram which explains the software side of the PetAid harness without undergoing into substantial detail. The drawing represents a process that is taken of how the PetAid harness will basically go into operation. It is exceptional to understand beforehand what are the tasks required to be completed, since if the prototype is finished early, then modification can be achieved for a superior quality PetAid harness. Each of the blocks are designated a "leader" whose responsibility is to provide the proper documentation and correct hardware part model to fit the specifications of the PetAid harness, but duties can still be divided based on any issues or substantial workload encountered. On the other hand, almost all parts will involve a combination of work tasks between the hardware and software section of the project for all team members.

3. Research Project and Specification for Fulfilling the Requirements

The research conducted for the PetAid Harness is found in this section of the report. When researching relevant technologies for the PetAid Harness, an extensive amount of research was put toward the hardware components and software. The hardware components that were researched were separated into 7 parts, in order, starting with the microcontroller, WiFi/Bluetooth module, heart rate module, thermometer module, accelerometer module, GPS and GMS module, power supply and finally the Dog Harness. The software research begins in the second part of research conducted for the PetAid Harness. In the software research, we introduce software requirements for the project and software applications that were utilized to complete the PetAid Harness. Our group decided to only include the most relevant technologies that would operate successfully with the PetAid Harness with the least amount of cost. As with the software, it was decided to choose an application that included a programming language that our team members were familiar with, in order, to make things easier for our group to learn.

3.1. Related Projects Research

To get a better understanding on how our group would implement the PetAid Harness, our group decided to research different projects that were related to the same objective of the PetAid Harness. The related projects we researched had some similar ideas to that of our project, such as connected WiFi/Bluetooth module, Temperature sensor, Accelerometer module and others. The research of the related projects played a huge part in our project. From the three related projects we choose to include in our report, our team was able to get an estimate on how much it would cost for us to design our project. We also got an image on how the final design should look like. Finally, the most important research we obtained was, what we were expected to research, such as the hardware components and software. In the following subsections, are the selected few related projects our team viewed as prominent.

3.1.1. Whistle GO Explore

The Whistle GO explore is a clip on collar attachment that allows the user to collect detailed data about their pet's fitness in coordination with the Whistle App. It features real-time tracking using a GPS unit which interacts with the AT&T network and google maps so that owners can locate dogs at any time through their smartphone. The app allows for the designation of "safe zones" which will notify you when the collar breaks the boundaries of these zones. It also gives a detailed 24-hour activity timeline describing rest periods, periods of high physical stress, scratching and licking periods, and even emails the information to your vet once a month. Multiple users can be added on an account so that multiple people can monitor the health trends of their pets at the same time. This helps to ensure that the pet is always fed properly, is given medication at the proper time, and is getting enough exercise everyday by coordinating its health info on the app between all caretakers. It further coordinates your pet's health routine by allowing you to set fitness goals based on breed, age, and weight. It will recommend how long your pet should be active per day and track distance traveled and calories burned. If changes in behavior deemed "regular" over time occur, then the app will notify the caretakers. The app also works for multiple pets at once. It is priced at $129.95 with a $10 a month subscription fee and available only in the United States.

Summary of Features

- Built in night light, controlled from an app for nighttime visibility.
- 1 Hook-and-Loop collar attachment
- 1 Snap collar attachment
- Catalog of fitness data, including sleeping, scratching, vet updates, and notifications for behavioral changes.
- Real-time location tracking using GPS
- Reminders for medication, vet appointments, feeding, etc.
- Nutrition calculator
- Customized fitness plan based on breed, age, and weight.
- Daily feedback on your pet's day accessible for review at any time from the app
- Allows for multiple caregivers on same account

Technical Specifications

- Recommended for pets 8 lbs. and up
- Contains a 3-axis accelerometer
- Secures to any collar or harness up to 1" wide
- Up to 20 days of battery life
- Uses 2 satellite systems (GPS and GLONASS)
- Leverages local Wi-Fi and cell tower data
- Requires a Wi-Fi connection
- Can connect to multiple Wi-Fi networks
- Only 2.4 GHz networks are supported
- Uses Bluetooth Low Energy (BLE)
- Requires Apple device with iOS 11 or later or Android device with 6.0 or later

Physical Specifications

- .96 Ounces
- Width: 1.4” | Height: 1.8” | Thickness: 0.7”
- Shock resistant
- Waterproof rated IPX 8

Inspirations from design

- Lightweight and compact design is clearly possible
- GPS tracking
- Clip on attachment instead of full collar
- WiFi connect ability
- Accelerometer required

3.1.2. LINK AKC Smart Collar

The LINK smart collar is a tracking/fitness product created by The American Kennel Club (AKC), the organization known for hosting dog competitions and keeping a registry of purebred dog breeds in the United States. Using their over 130 years of expert experience with dogs and the needs of dog owners they designed their image of the ideal pet tracker. With an app, it gives real-time location using a cellular connection and gives notifications whenever the pet leaves a designated “safe zone”. When needing to track your pet in a short proximity or in low light conditions the app allows you to activate a digital audio “chime” on the pet collar or activate an LED attached to the collar. The chime feature can also be used as a training aid as training with sound is a common method for building consistent dog behavior. The app also provides precise activity tracking using data from a 3-axis accelerometer sensor to track periods of rest and high activity. The app’s algorithm even uses data from activity over time so that it gets an accurate definition of what constitutes “high activity” for your pet. Using this and data the user enters about their dogs’ weight, breed, age, and size the app creates a custom activity recommendation for your pet every day. The collar will also provide notifications via the smartphone app when your dog is in an environment deemed too hot or too cold based on settings entered by the owner. The app also logs periods of activity and records them in a “virtual scrapbook” for review so that the owner can have a detailed catalog of previous activity. Using cloud storage, the app allows for an unlimited number of logged workouts. Unlike other pet trackers, the LINK AKC Smart Collar is not a clip on attachment but is instead a device that wraps fully around the pets’ neck. Therefore, sizing must be considered and done carefully to ensure the comfort and safety of the pet, along with the durability of the tracker. It costs $59.99 with a $10 a month subscription fee.

Summary of Features

- Real-time location tracking using smartphone app, GPS, and cellular data
- Notifications when leaving “safe zone”
- LED built on collar, controllable via app
- Audio cue built on collar, controllable via app
- Precise activity tracking
- Recommended daily activity based on logged activity data and dog traits
- Notifications for “extreme” temperature

- Virtual activity scrapbook with unlimited cloud storage
- Base charging station and charger

Technical Specifications

- Recommended for pets 10+ lbs.
- 3 days battery life
- Minimum operating system requirements are iOS 9.0 or Android 5.0
- Contains 3-axis accelerometer
- Leverages local Wi-Fi and cell tower data
- Requires a Wi-Fi connection

Physical Specifications

- Waterproof rated IP67
- Collar = 1” wide, between 9”-25” long
- Tracking Module = 1.5 by 3.9 by 0.5 inches
- 2.55 ounces
- Reflective fabric

Inspirations from design

- Temperature alerts (potential hot car alert)
- Easy access to medical records like vaccinations
- Ability to log dog workouts much like a running app
- Full collar makes sizing difficult

3.1.3. PetPace Health Collar

PetPace is the most comprehensive health monitoring collar we researched by far. It is meant to be integrated with your pet's veterinary service and functions to prolong the life of your pet with innovative monitoring and diagnosing technology. It comes as a full collar, not a clip-on attachment, and sizing is essential for the well-being of both the tracker and the pet. The collar continuously tracks your pet's physiological responses like temperature, activity, pulse, respiration, calories consumed and burned, and heart rate variations all non-invasively. For pets recovering from injuries, it also can track posture, like periods of laying or standing, to track areas of discomfort. It even tracks position using GPS and uploads all its monitored data every 2, 15, or 30 minutes. If any abnormalities are detected in your pet’s physiological responses a notification is sent to both the pet owner and the veterinarian so that they can act as quickly as possible. All the PetPace health monitoring features work by having the collar wirelessly connect to a Gateway directly connected to your modem via Ethernet. Using the integrated Health Monitoring Service PetPace analyzes all the data relating to your pet's physiological responses, changes in behavior, and levels of activity. If PetPace notices any abnormal trends, like a high temperature or heart rate during low activity, in your pet then it quickly sends notifications to the caretaker and their vet. Depending on the subscription service selected it costs between $159.95-$349.95.

Summary of Features

- Stores health data when not within Gateway range

- Uploads data every 2, 15, or 30 minutes when within Gateway range
- Posture tracker for injury recovery
- A comprehensive health report that tracks pulse, temperature, respiration, activity levels, calories burned, and variations in heart rate
- GPS tracking
- Alerts for veterinarian and caretaker
- Analysis of pet health data to help provide early diagnosis and response
- Gateway and Gateway charger

<u>Technical Specifications</u>

<u>Tracker</u>

- Lithium polymer 250mAh battery
- Max 3 weeks between recharges
- Recommended for pets 8 lbs. and up
- Requires Apple device with iOS 11 or later or Android device with 6.0 or later

<u>Gateway</u>

- Must be connected via Ethernet to modem
- Maximum 1000 ft range in open space
- micro-USB power port
- Ethernet 10/100 BASE-T
- Single gateway can handle up to 30 collars

<u>Physical Specifications</u>

<u>Tracker</u>

- Comes as a complete collar and must be sized correctly
- 1.5 ounces
- 1.57” x 1.27” x .59”
- Waterproof rating IP-X7
- Shockproof

<u>Gateway</u>

- 3.07” x 2.11” x 2.00”
- 2.18 ounces

<u>Inspirations from design</u>

- Heart Rate Monitor
- Potential “Gateway” to store data for later access
- Potential posture tracker

3.1.4. Relevant Technologies Research

This will not be the first time and likely not the last time that an augmented dog harness is created by a senior design group, and it likely will not be the last. Because of this, we have drawn inspiration from other projects, and we've seen what we can improve, as well as pitfalls to avoid.

3.2. Microcontroller Research

A microcontroller has many uses: in consumer electronics like toys and appliances, in communication devices like cell phones and computers, and in medical instruments like personal blood pressure monitors and heart rate monitors. Also, in industrial settings, it can be used to control temperature and pressure, count time, or measure speed. Because microcontrollers are custom-made for specific tasks, it is important to choose a microcontroller that best fits the PetAid harness. The microcontroller will be the main component of the system. One major reason is the microcontroller will transmit the data from the PetAid harness to the user who is monitoring the PetAid harness from a mobile device or tablet. The data is then passed on through a wireless connection either WiFi, Bluetooth, or WiFi + Bluetooth module. During the selection process, the group made a list that considered a few important key factors in determining the microcontroller we would use for the PetAid Harness.

1. Pin Count

   The more pins on the microcontroller indicates there will be more ports available for interfacing devices to the microcontroller. Most pins are general purpose input/output (GPIO) pins and some pins have specific purpose. The microcontroller our group will choose must have enough pins to connect all the IO devices used in the PetAid harness but reduce the number of unused pins.

2. CPU Speed

   CPU is the abbreviation for central processing unit and is the brains of the computer where most calculations take place. The speed of a microcontroller is typically measured in megahertz (MHz) and million instructions per second (MIPS). We were taught in Embedded Systems that a typical microcontroller with bigger bus widths tend to have greater speeds. For the PetAid harness, an 8-bit microcontroller running between 8 to 16 MHz is enough.

3. Power Consumption

   The power consumption relies heavily on the current being drawn by both the microcontroller and any external devices attached to it. Because the PetAid harness will have more than one external device such as a temperature sensor, pulse sensor, and an accelerometer, it will require more current to operate. The current demands of the devices used in the PetAid harness were heavily considered when selecting a microcontroller. Also, the microcontroller chosen for the PetAid harness had to meet the proper current ratings to cover the current consumed by both the microcontroller and the external devices.

4. Sleep Mode

   Most microcontrollers are designed to have a low power sleep mode to help conserve power. The microcontroller wakes up through the existence of an event, but usually spends most of the time in sleep mode. For the PetAid harness, we will be using batteries. Which means that if we want the PetAid harness to work for long periods of time the sleep mode was an important factor when choosing a microcontroller.

   While in Sleep Mode, we still wanted the essential functions of the device to be running, such as the data being collected, and a low-power mode above sleep mode that would initialize when the app was running in the background.

5. Instruction Set

   We were taught in Embedded Systems that a typical microcontroller with larger data bus widths will have a more complex microcontroller instruction set. For the PetAid harness an 8-bit or 16-bit microcontroller meet the processing demand. Given the constraint of a smaller processor than the computers we have typically used to program, our team focused on having the least amount of computational strains with our functions that operated in the microcontroller.

6. Program memory

   Microcontroller program memory is typically in the format of non-volatile FLASH or electronically erasable programmable read-only memory (EEPROM). The size of the code written by the computer engineer determined the actual size of program memory needed for the PetAid harness. But we concluded that using a microcontroller with 8 to 16 KB of program memory was enough to store the written program code. Because it was possible to keep all the stored data and its analyses on the phones application , we were able to erase all the data collected on the microcontroller after the functions were executed.

7. Data memory

   The microcontroller data memory is typically in the format of volatile RAM and can be in the form of FLASH or EEPROM. The data memory was important because it stored data obtained from the sensors in the PetAid harness. For the PetAid harness 8 to 16 KB of data memory was enough to store the logged data on ThingSpeak.com.

8. Analog-to-Digital Converter (ADC)

   The ADC takes analog physical quantities, observes the information, and then converts them to digital values (Binary digits) that can be processed by the program stored in the microcontroller program memory. Because the PetAid harness used sensors that senses physical quantities, the microcontroller we choose had to include ADC pins.

9. Digital-to-Analog Converter (DAC)

   Because microcontrollers are digital, digital values need to be changed into analog form prior to being sent to any output device that requires an input signal in analog form. The DAC works by converting digital signals to analog signals. This is an essential feature that the microcontroller was required to have so, that output devices that only received an analog signal would function.

10. Real Time Clock (RTC)

The RTC is a computer clock that keeps track of current time. The typical RTC works together with an interrupt and is developed to interrupt the CPU after a specified amount of time. Because the PetAid Harness would record and log data through different times of the day and for long periods of time, the microcontroller was required to have an RTC.

3.2.1. MSP430FR6989 LaunchPad Development Kit

The MSP430FR6989 LaunchPad Development Kit provides a low cost and simple to use rapid prototyping platform for analyzing the MSP430FR6989 Ultra-Low-Power FRAM microcontroller. The 16-bit MSP430FR6989 features an on-board segmented display which is operated directly from an on-chip controller as well, 128 KB of high speed and ultra-low power FRAM memory. It also featured a 16 MHz system clock, 2 KB RAM, 12-bit ADC, 16 channel analog comparator, 3-channel internal DMA, five 16-bit timers, SPI, I^2C, and UART. This launchpad used the Texas Instruments 100-pin MSP430FR6989 MCU in a low profile Quad Flat Package (QFP), eZ-FET emulator with EnergyTrace++ technology support for real-time current measurements, a USB Micro-B connector, 2 user configurable push buttons, a hardware reset button, 2 LEDs in green and red that are user configurable, and removable jumpers to disconnect the debugger circuit from the main processor circuit. The MSP430FR6989 LaunchPad is entirely programmable by 3 popular IDEs such as Code Composer Studio, Energia, and IAR Embedded Workbench. The cost for the MSP430FR6989 LaunchPad on TI was $20.00. Some additional MCU specifications provided from TI are shown below in bullet points.

- Operation voltage from 1.8V to 3.6V
- 16-bit RISC architecture up to 16-MHz system clock and 8-MHz FRAM access
- Up to 128KB of nonvolatile memory
- 230 µA/MHz active mode and 350 nA standby with RTC and 3.7-pF crystal
- Integrated LCD driver with contrast control for up to 320 segments
- Extended Scan Interface (ESI) for background water, heat, and gas volume measurement
- Operating temperature range (°C) between -40 to 85

3.2.2. MSP430G2 LaunchPad Kit

The MSP430G2 LaunchPad Development Kit provided a low cost and simple to use rapid prototyping platform for analyzing the MSP430G2 Ultra-Low-Power microcontroller. The 16-bit

MSP430G2553 included in the LaunchPad Kit featured integrated capacitive touch I/Os, a 16 MHz system clock, 10-bit SAR ADC, 16 KB Flash, 512 B RAM, two 16-bit timers, I^2C, SPI, and UART. This launchpad contains Texas Instruments 20-pin MSP430G2553 MCU in a Plastic dual in-line package (PDIP), a USB mini-B connector, hardware reset button, a user configurable push button, 2 LEDs in green and red that are user configurable, JTAG emulation, and removable jumpers. The MSP430G2 LaunchPad can be entirely programmable by 3 popular IDEs such as Code Composer Studio, Energia, and IAR Embedded Workbench. The cost for the MSP430G2 LaunchPad on Amazon was $23.00. Some additional MCU specifications provided from TI are shown below in bullet points.

- Operation voltage from 1.8 V to 3.6 V
- 16-bit RISC architecture up to 16-MHz system clock
- 16KB of flash memory and 512 bytes of SRAM
- 230 µA at 1MHz active mode and 0.5 µA standby mode
- Operating temperature range (°C) between -40 to 85

3.2.3. Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328. The Arduino consisted of both a physical programmable circuit board and a piece of software, or IDE. The Arduino IDE used a simplified version of C++, which was easy to learn. The major components for of the Arduino UNO were the USB connector, Power Port, Microcontroller, Analog input pins, Digital pins, Reset switch, Crystal Oscillator, USB interface chip, and TX RX LEDs. Next, we briefly discuss each important component of the Arduino UNO.

USB CONNECTOR:

The USB connector is a printer USB port used to load a program from the Arduino IDE onto the Arduino board. This board can also be powered through this port.

POWER PORT:

The Arduino board can be powered through an AC-to-DC adapter or a battery. The power source can be connected by plugging in a 2.1 mm center-positive plug into the power jack installed on the board. The Arduino UNO board typically operates at a voltage of 5 volts and can resist a maximum voltage of 20 volts. But if the board is supplied with an even greater voltage, there is a voltage regulator that will protect the board from being damaged.

MICROCONTROLLLER:

The microcontroller used on the Arduino UNO board is ATmega328 by Atmel, which is a major microcontroller manufacturer. ATmega328 has flash memory of 32 KB, RAM of 2 KB, CPU, and 1 KB of EEPROM. ATmega328 is pre-programmed with a bootloader, which allows us to directly upload a new Arduino program into the device without having to use any external hardware programmer.

ANALOG INPUT PINS:

The Arduino UNO board has 6 analog input pins, each which provide 10 bits of resolution and are labeled "Analog 0 to 5." These pins can read the signal from an analog sensor such as a temperature sensor and convert it into a digital value for system understanding. These pins measure just voltage and not the current because they have a high internal resistance, which means only a small amount of current flows through these pins. These pins can also be used for digital input or output even though they are labeled analog and are analog by default.

DIGITAL PINS:

The Arduino UNO board has 14 digital pins and are labeled "Digital 0 to 13.". These pins can be utilized as either input or output pins. When used as an output pin, they act as a power supply source for the components connected to it and when they are used as input pins, they read the signals from the components connected to them.

RESET SWITCH:

The reset switch is very useful when the code does not repeat, and it needs to be tested multiple times. By clicking on the switch, it sends a logical pulse to the reset pin of the microcontroller, and then runs the program again from the initial start.

CRYSTAL OSCILLATOR:

The Crystal Oscillator is a quartz crystal oscillator which ticks 16 million times a second. At every tick, the microcontroller performs one operation such as addition or subtraction.

USB INTERFACE CHIP:

The USB interface chip acts as a signal translator, which converts signals in the USB level to a level that the Arduino UNO board can understand.

TRANSMIT (TX) RECEIVE (RX) INDICATOR:

The TX and the RX indicators are indicator LEDs which blink every time the UNO board is transmitting or receiving data.

Some additional specifications provided from the Arduino datasheet are shown below in bullet points.

- Operation voltage 5 Volts
- 32KB of in-system self-programmable flash program memory and 2KB SRAM
- Clock speed of 16 MHz
- Active mode: 1.5mA at 3V - 4MHz
- Power-down mode: 1µA at 3V
- Operating temperature range (°C) between -40 to 125

3.2.4. Raspberry Pi 3 Model B

The Raspberry Pi 3 Model B is neither a microcontroller nor microprocessor but was more like a minicomputer. The Raspberry Pi 3 can be configured to operate like a microcontroller. Unlike the microcontrollers considered up to now, the Raspberry Pi 3 includes an on-board Wi-Fi and Bluetooth connectivity. So, this device will eliminate the purpose of implementing a microcontroller that can communicate through wireless communication either by Wi-Fi or Bluetooth module. The Raspberry Pi 3 has a Quad-Core 1.2 GHz Broadcom BCM2837 64bit CPU, 1GB RAM, 40 pins extended GPIO, 4 USB and 2 ports. The cost for the Raspberry Pi 3 is $41.99 on Amazon.

3.2.5. Microcontroller Choice

MCU	MSP430FR6989	MSP430G2553	Arduino Uno	Raspberry Pi 3
CPU	16-bit RISC	16-bit RISC	Microchip AVR (8-bit)	ARM Cortex-A53
Frequency (MHz)	16	16	16	300
Memory	FRAM and RAM	FRAM and RAM	Flash memory and SRAM	RAM
Memory Size	128KB and 2KB	16KB and 0.5KB	32KB and 2KB	1 GB
UART	2	1	1	2
I2C	2	1	2	2
SPI	4	2	2	3
Operation Voltage (V)	1.8 to 3.6	1.8 to 3.6	5	5
GPIO	83	24	14	40
Dimensions (WxL)(mm^2)	16x16: 256 mm^2	9.4x24.33 mm^2	68.6x53.3 mm^2	85x56x17mm^2 (WxLxH)
Price	$20.00	$23.00	$22.95	$41.99

Table 4. MCU comparison

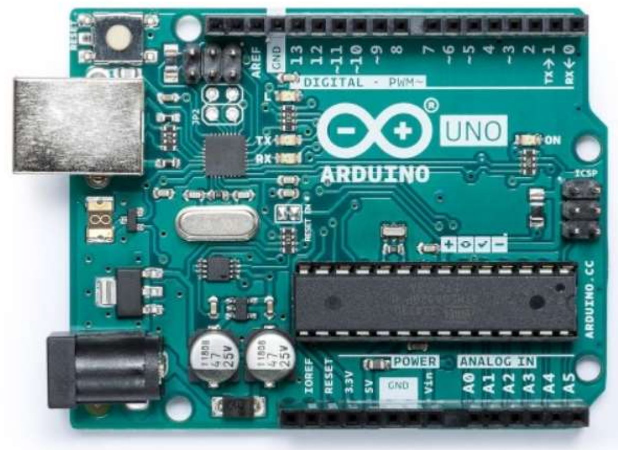


Figure 3. Arduino Uno (Courtesy of DigiKey)

Table (4) displays a comparison between all 4 microcontroller boards that were considered for the PetAid harness project. Using the table, we created, we were able to compare and make a clear decision on which MCU board we would use for the project. During the MCU selection process, we concluded that we would go with the Arduino Uno shown in figure 3 because it met most of the requirements for the project. The Arduino UNO had enough pins to connect all the IO devices that were going to be used in the PetAid harness. The CPU runs between 8 to 16 MHz, which is enough for the project. The power consumption is only 5.8 µA for the 5-volt version and is enough to function. The Arduino UNO had enough program memory and data memory. Also, the Arduino IDE uses a simplified version of C++, which everyone in the group was familiar with. So, it was easier to code and test our hardware components. Since the PetAid harness included sensors, it was important for the MCU to have enough ADCs, in which the Arduino UNO had 6. Even though the MCU was cheap, but the Arduino UNO was a little expensive, it was the right board to test our components and the right chip for the PetAid Harness.

3.3. WiFi + Bluetooth Module Research

Both WiFi and Bluetooth played an important part in the Pet Aid harness as it would provide wireless communication between the PetAid Harness and the internet. The main difference between the WiFi and Bluetooth is that, the Bluetooth is typically used to connect devices in short ranges for sharing data, whereas WiFi provides high-speed internet access and has longer ranges for connectivity. For our project, we tried implementing both the WiFi and Bluetooth module on the PetAid harness to track the activity of the pet each day on ThingSpeak.com. Based on the information obtained from the sensors, the owner could make clear observations whether his/her pet was healthy or not and could make better decisions with his/her pet. Also, the Veterinarian could use the information in the database either through an application or website to help the owner understand what was wrong with his/her pet. For the PetAid harness Wifi and Bluetooth module, there were a few important key factors that were considered. Those key factors included: Bandwidth, Frequency, Protocol, Voltage Supply, Current consumption, Current Transmitting, Frequency range, Dimension, Operating temperature range, and Cost. The next few sections provide a few Wifi + Bluetooth modules that were considered for the PetAid harness project.

3.3.1. Type 1DX

The type 1DX module was a very small module that came with a 2.4GHz WLAN and Bluetooth functionality. This module was smaller than a dime. But it required external connections to complete a radio design, such as a power source, an antenna, clocks, a processor, and associated interface hardware to complete the radio hardware design. Table (5) displays some key factors for the Type 1DX.

Key factors	Description
Bandwidth	65 Mbps
Frequency	2.4 GHz
Protocol	802.11b/g/n, Bluetooth v4.0
Voltage Supply	3.3V to 4.2V
Current Transmitting	370mA
Current Receiving	60mA
Dimension	6.95 x 5.15 x 1.1 mm
Operating Temperature Range (°C)	-30 to +70
Costs	$12.60

Table 5. Type 1DX key factors and descriptions

3.3.2. CC3100 SimpleLink

The CC3100 WiFi module was part of the SimpleLink family from Texas Instruments. The CC3100 device was able to connect to just about any low powered microcontroller to the internet of things. The device combines all the protocols for WiFi and the internet, which meant it would minimize host microcontroller software requirements. Table (6) displays some key factors for this device.

Key factors	Description
Frequency	2.4 GHz

Protocol	802.11b/g/n Radio, Baseband, and Medium Access Control
Voltage Supply	2.1V to 3.6V
Low Voltage deep sleep mode	115µA
Dimension	9.0 mm x 9.0 mm
Operating Temperature Range (°C)	-40 to +85
Costs	1ku | $3.25

Table 6. CC3100 SimpleLink Key factors and descriptions

3.3.3. SH-HC-08

The SH-HC-08 from DSD TECH was a Bluetooth 4.0 ble module. This device was based on TI CC2541 Bluetooth 4.0 ble chip. It was a class 2 Bluetooth, which means it operated in the range from 10 to 20 meters, but on the datasheet provided by DSD TECH it stated that this device had an effective distance range for connectivity of 10 meters. It was fully compatible with ios7.0 or later and the Android 4.3 or later. The size for the device was 26.7 x 13 x 2 mm (LxWxH) and weighed close to 0.16 ounces. The operating temperature ranged from -40 °C to +85 °C. It also had an operating frequency band of 2.4 GHz. The operating voltage for the SH-HC-08 was 3.3V. This device can be bought on AMAZON for $7.99.

3.3.4. BLE Link Bee

The Bluno Bee was a serial to Bluetooth 4.0 module based on TIs CC2540 chip. The transmitting distance from this device was from 15 to 20 meters indoor and 30 meters in free space. It was compatible with Arduino UNO and was supported by devices like Androids and iPhones. It required an input DC voltage of 3.3 volts. Table (7) displays a few more key factors for this device.

Key factors	Description
Frequency	2.4 GHz
Voltage Supply	3.3v to 5v
Power Consumption	working: 10.6mA, ready mode: 8.7 mA
Dimension	32mm x 22 mm
Operating Temperature Range (°C)	-40 to +85

Costs	$9.90

Table 7. BLE Link Bee Key factors and descriptions

3.3.5. nRF51822

The nRF51822 was a Bluetooth device from Nordic Semiconductor. It had some key features that we considered important like, a 2.4 GHz radio frequency, 32-bit ARM Cortex M0 processor, 256 KB/128KB flash and 32 KB/16KB RAM, 10-bit ADC, Low cost external crystal, supply voltage ranged from 1.8 volts to 3.6 volts, 31 configurable GPIO, and 16 channel PPI. This device also supported devices such as Androids and iPhones. The dimensions for the nRF51822 were 21 mm x 18.5 mm. The temperature operating ranged from -25 to +75 °C. It also covered 10 meters indoor. The cost for this device was priced at $7.00 on robotshop.com.

3.3.6. ESP32

The ESP32 also known as ESP WROOM 32 was a WiFi and Bluetooth chip. It supported WiFi HT40 and classic Bluetooth/BLE 4.2. It also supported Arduino IDE and was engineered for mobile device and wearable electronics. The integration of Bluetooth and wifi provided for different ranges of applications to be accessed or targeted, and the module was an all-around device. Using the wifi allowed for a large physical range and connection to the internet through a wifi router, whereas the Bluetooth allowed the user to connect to the phone or broadcast low energy ranges at short ranges. Table (8) displays more important key factors and specifications for the ESP32. The cost for this device was $6.49 on dfrobot.com.

Key factors	Description
WiFi	
Frequency	2.4 GHz to 2.5 GHz
Protocol	802.11 b/g/n
Bluetooth	
Radio	Class-1, class-2 and class-3 transmitter
Voltage Supply	3.0v to 3.6v
Operating current	Typically, 80 mA
Dimension	18.00 mm x 25.50 mm x 3.10 mm
Operating Temperature Range (°C)	-40 to +85

Costs	$6.49

Table 8. ESP32 Key factors and descriptions

3.3.7. DFRobot WT8266-S1

The WT8266-S1 WiFi module was based on the ESP8266 chip. It functions in the wifi 802.11 b/g/n standard at 2.4 GHz. It comes with 18 pins each with a pitch of 2 mm, 9 which are GPIO, 1 ADC and UART. It also comes with a built in PCB antenna. The dimension of the device was 19 x 15 mm. The device was compatible with Arduino IDE if the operating voltage was 3.3 volts. Some additional key features for this device are shown in table (9).

Key factors	Description
Frequency	2.4 GHz
Operating voltage	3.3v
Protocol	802.11 b/g/n
Power Consumption	Standby mode: Modem Sleep 15 m, Light Sleep 0.9 mA, Deep Sleep 20 µA, OFF 0.5 µA.
Dimension	19 mm x 15 mm
Operating Temperature Range (°C)	-40 to +85
Costs	$6.90

Table 9. DFRobot WT8266-S1 Key factors and descriptions

3.3.8. WiFi + Bluetooth Module Choice

Wi-Fi/Bluetooth Module	SH-HC-08	WT8266-S1	ESP32(WROOM32)
Criteria	Bluetooth module	Wi-Fi Module	Wi-Fi and Bluetooth Module
Operating Temperature	-40°C to +85 °C	-40°C to +85 °C	-40°C to +85 °C
Supply voltage	3.3V(3.0V - 6.0)V	3.3V (3.0V - 3.6V)	3.3 V (3.0-3.6V)

Frequency	2.4 GHz	2.4 GHz	2.4GHz to 2.5GHz
Dimensions	(35.7x15.2x2.54)mm	(20x12.2 x2.9)mm	(18x25.5x3.1)mm
Cost	$7.99	$6.90	$6.49

Table 10. WiFi + Bluetooth Module comparison

Figure 4. ESP32 (ESP WROOM 32) WiFi + Bluetooth (Courtesy of Digi Key) First choice for project

Many of the WiFi/Bluetooth modules have adequate specifications listed that can be available and compatible with a microcontroller. However, since the desired microcontroller was the Arduino Uno as mentioned previously, there were a few WiFi/Bluetooth modules that were easily compatible with the Arduino. The Type 1DX was a small module, but its availability in stock was limited on the internet and our group did not want to wait for this device to be available in the future to have access to purchase the 1DX. The CC3100 SimpleLink was a Wi-Fi network processor that had no Bluetooth and only Wi-Fi Internet-on-a-Chip network. The plan as determined for the PetAid Harness was to reduce unneeded capacity and make it compact/portable and this could not be done by having a separate Wi-Fi and Bluetooth module. That ruled out the CC3100 SimpleLink as well as the WT8266-S1 WiFi module. The SH-HC-08 from DSD TECH and BLE Link Bee were both considered a Bluetooth 4.0 ble module that involved usage of Bluetooth wireless technology but did not have a Wi-Fi network integrated into it the chip.

When contemplating the SH-HC-08, was a very highly utilized Bluetooth module and most common due to its impressive specifications, ease-of-application, high cost-to-performance ratio, low power wireless technology and reliability. Due to the project design and reducing modules used in our project, the group decided it would be better to use one single module for both uses of Wi-Fi and Bluetooth technology. In comparison to the BLE Link Bee, the size of the chosen ESP32 Wi-Fi/Bluetooth module was smaller in dimension size considering that the ESP32 had an integration of Wi-Fi and Bluetooth, while the BLE Link Bee was the most immense compared to the other Wi-Fi/Bluetooth modules at 32mmx22mm (LxW) and with only a Bluetooth mode operation. The BLE Link Bee may have had longer distance range for a range of 30 meters to detect connectivity between devices. The first choice of a Wi-Fi/Bluetooth module was the ESP32.

It supported WiFi HT40 and classic Bluetooth/BLE 4.2. It also supported the Arduino IDE and was engineered for easy use with mobile devices and wearable electronics. Table (10) displays a comparison of the different WiFi and Bluetooth Modules considered for the PetAid Harness

For the PetAid Harness after failing with the intended first choice, which was the ESP32, our team decided to use the ESP8366-01. The ESP8266 had 1 mega Byte of flash memory, 2 available GPIO pins, VCC, TX, RX, Chip Enable, Reset and Ground pins. It had a small antenna on the PCB, which had a wifi connecting range of 5 meters. The ESP8266 operated at 3.3 volts and did not come with a 3.3 volt regulator. Because our data was being sent from the ATmega328P-PU chip and the chip used a 5 volt logic and the ESP8266 used a 3.3 volt logic, we had to step down the voltage on the chip by using a voltage divider. The chip enable pin on the ESP8266 required 3.3 volts but did not require any current, so we connected a 10 Kohm pull-up resistor between the VCC line and the CH_EN. The ESP8266 also had a current consumption that ranged between 20 mA up to 240 mA of current, which we had to consider choosing a new power supply because it was too much current needed from the ESP8266. Because the ESP8266 did not have a built-in USB to serial converter, we had to find a different way of programming it. Luckily, we had the Arduino UNO. When we bought the ESP8266, it already come with a pre-installed AT firmware. Because the AT firmware was compatible with the Arduino IDE, it was possible for us to flash the chip with another firmware. Through a few AT commands we were able to set the wifi module as a station mode and connect to our internet connection, providing our project with wireless communication.

3.4. Heart Rate Monitor Research

A basic heart rate sensor includes a light emitting diode and a detector. The detector can either be a light detecting resistor or photodiode, and when the tissue of the body for example a finger is strike with the light, it is mirrored or reflected as shown in figure (5). The light is then received by the detector and based on the reflection, the detector will transmit an electrical signal that is proportional to the heart rate, which is exactly what we want for the PetAid harness. This was the basic concept that was used to apply a heart pulse sensor in the PetAid harness.

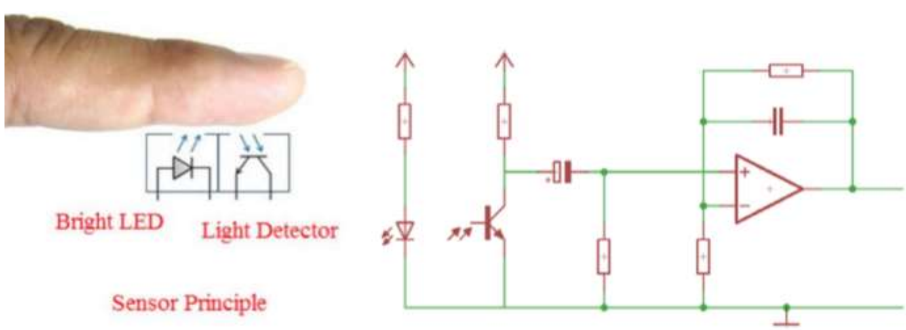


Figure 5. Basic concept of the Heart Rate sensor with the circuit

Today, the optical heart rate monitors are the most common pulse sensors in wearable electronics. So, our group decided to go with an optical heart rate monitor. By choosing the optical heart rate monitor, we had to consider the placement of the device on the friendly dog. It would need to be placed on the body where there is no or less dog fur to obtain the best results from the sensor. There were 4 options we considered for the heart rate monitor for the PetAid Harness.

3.4.1. SEN 11574

The SEN-11574 pulse sensor was a low-cost optical heart rate sensor that was compatible with the Arduino UNO and other microcontrollers. It was designed and made from the World Famous Electronics. The great thing about this heart rate sensor is that it was easy to use and could be used by anyone who wanted to incorporate live heart rate data into their projects. The sensor was perfectly sized and could clip onto a fingertip or earlobe. It also included an open-source monitoring app that plotted the pulse in real time. The dimensions of the heart rate monitor were 16 mm x 3.8 mm (W x H). Also, the cost for this device was $24.95, which was one of the most expensive on the market. Also, in table (11) we listed the absolute maximum ratings for the heart rate monitor.

Absolute Maximum Ratings	MIN	TYP	MAX
Operating Temperature Range (°C)	-40		+85
Input Voltage Range (V)	3		5.5
Output Voltage Range (V)	0.3	Vdd/2	Vdd
Supply Current (mA)	3		4

Table 11. Absolute maximum ratings for the SEN 11574

3.4.2. Max30100

The Max30100 is a heart rate monitor from protocentral.com. This device is a combination of two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart rate signals. The operating voltage ranges from 1.8 volts to 3.3 volts. It is perfect for wearable devices, fitness assistance devices and medical monitoring devices. The dimensions for this device are 5.6 mm x 2.8 mm x 1.2 mm, which is a good size because we need very small sensors for this project. This device also costs $12.56 on amazon.com. The MAX30100 is compatible with Arduino. Table (12) displays some important ratings in this device.

Absolute Maximum Ratings	MIN	TYP	MAX
Operating Temperature Range (°C)	-40		+85
Power Supply Voltage (v)	1.7	1.8	2.0
Supply Current (µA)		600	1200

Continuous Input current into any terminal (mA)	-20		20

Table 12. Absolute maximum ratings for the MAX30100

3.4.3. AD8232

The AD8232 single LED heart rate monitor from SparkFun is used to measure the electrical activity of the heart. The electrical activity can be reported as an electrocardiogram and sent out as an analog reading. Electrocardiograms are usually noisy, and the AD8232 can act as an op amp to collect clear information from the PR and QT interval easily. In other words, this device allows for an ultralow MCU to obtain the output signal easily. The AD8232 has dimensions of 4 mm x 4 mm in a 20-lead LFCSP package. The cost for this device is $19.95 on sparkfun.com. Table (13) displays the absolute maximum ratings for this device.

Absolute Maximum Ratings	**MIN**	**TYP**	**MAX**
Operating Temperature Range (°C)	-40		+85
Power Supply Voltage (V)	2.0		3.5
Voltage at any Terminal (V)	-0.3		0.3

Table 13. Absolute maximum ratings for the AD8232

3.4.4. Heart Rate Monitor Sensor Choice

Heart Rate Module	SEN 11574	Max30100	AD8232
Input voltage range	3V - 5.5V	1.7V - 2.0V	2.0V - 3.5V
Input current	3mA - 4mA	600µA - 1200µA	50nA - 200nA
Operating Temperature	-40°C - +85°C	-40°C - +85°C	-40°C - +85°C
Dimension	16 mm x 3.8 mm (W x H)	14.4 mm x 3.0 mm (W x H)	4 mm x 4 mm (W x H)
Type	Pulse Sensor	Heart-rate monitor sensor	Heart-rate monitor
For use with/related products	Arduino	Arduino	Raspberry Pi or Beaglebone Black

Price	$24.95	$12.56	$19.95

Table 14. Comparison for three different heart rate sensor/monitors

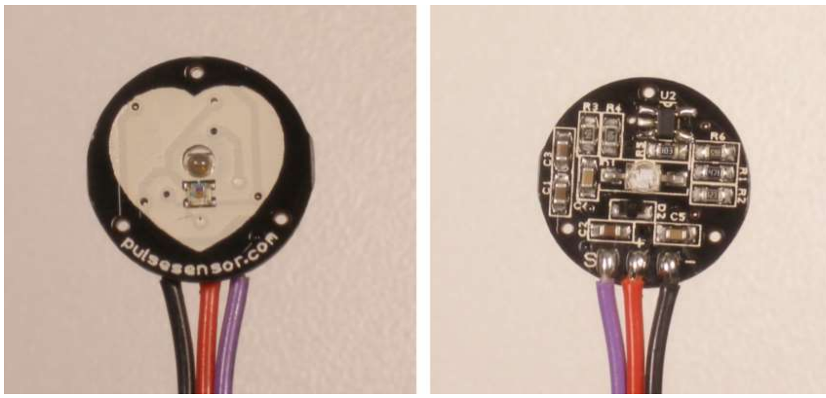


Figure 6. Displays the front and the back of the SEN 11574 Heart Rate Sensor (courtesy of SparkFun)

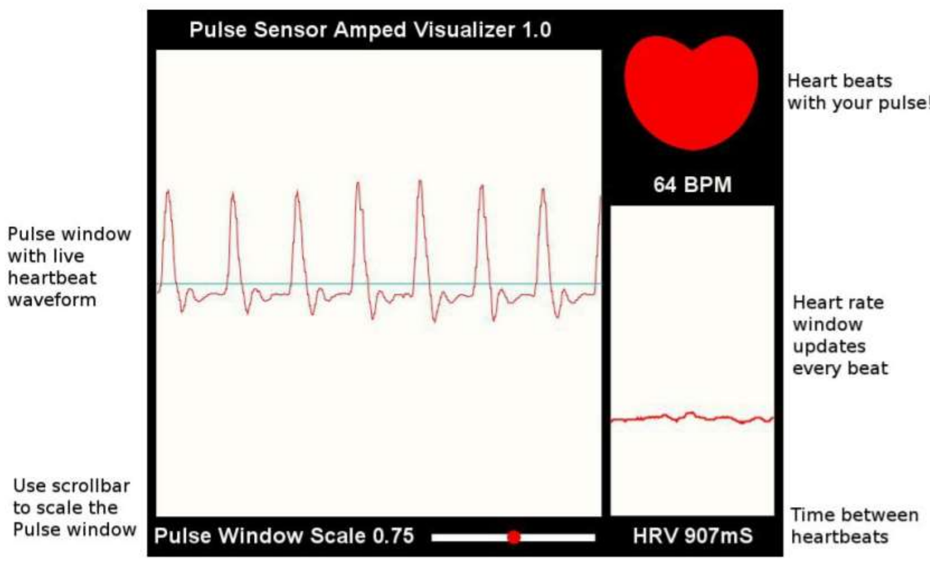


Figure 7. Graph of raw sensor data over time (Courtesy of SparkFun)

Table (14) displayed a comparison between all 3 heart rate monitors/sensors considered for the PetAid harness project. Using the table, we were able to determine which device we would choice for the project. From the three heart rate monitors/sensors in table (14), our group decided to go with the SEN 11574. The pulse sensor was based on photoplethysmography technology, which is based on a very simple fact. Blood absorbs green light, and each time the heart beats, it pumps a blood pulse into vessel. As a result, more green light is absorbed. Therefore, the intensity for the reflected light is captured by the optical sensors giving us the density of the blood pulse. The SEN 11574 was not the cheapest, but it was very easy to use. It was an easy plug and use heart-rate sensor that was compatible with Arduino IDE. Sparkfun.com provided plenty of resources such as codes and schematics for the SEN 11574. The price might be a little more than the other devices because it included a pulse sensor board, a 24-inch color-coded cable with standard male headers, a Velcro finger strap, and transparent stickers to protect the sensor. All those features came as a package when we purchased the SEN 11574. Also, the dimensions of the device were perfect for the project because they were neither too small nor too big to attach to the harness we used on the pet. Finally, the SEN 11574 has absolute ratings that meet the requirements for the MCU we have chosen for the PetAid harness. Figure (6) and figure (7) display this device and real time data from the SEN 11574.

3.5. Thermometer/Temperature Sensor Research

Thermometers are widely used to sense the temperature in just about any environment. They all work in a similar way but have small different features. Typical Thermometer devices are Thermocouples (TCs) or Resistance Temperature Detectors (RTDs). TCs work by using two metal wires to make a voltage relative to the temperature present in the junction between them. There are different special kinds of TCs, which can integrate different metals to measure temperature ranges and produce specialized calibrations. An RTD sensor works by measuring temperatures based on the resistance changes in a metal resistor side. The most used RTDs are called PT100 sensors, which use platinum and possess a resistance of 100 ohms at 0 °C. Besides the RTDs and TCs, there are other options such as integrated circuit temperature sensors, which can be digital ICs and analog ICs. An integrated temperature sensor is a two terminal integrated circuit temperature transducer that generates an output current proportional to absolute temperature. For the case of digital ICs, digital IC sensors provide an output that has been computed through an integrated A-D converter and is available for input into digital control and monitor systems. The IC sensor does not need any external circuitry done and is much cheaper than RTDs. The three biggest advantages for IC sensors are they are the most linear, have the highest output, and are inexpensive. Table (15) provided by OMEGA compares the RTDs, TCs and ICs for research purposes.

Sensor Type	Thermocouple	RTD	IC temperature sensor
Typical Temperature Range	-200°C to 1750 °C	-200°C to 650 °C	-55°C to 150 °C
Typical Accuracy	0.5 to 5 °C	0.1 to 1 °C	Depends on device
Long term stability at 100 °C	Variable	0.05 °C/year	Varies
Linearity	Non-linear	Fair linear	Most linear

Power Required	Self-powered	Fixed voltage or current	Power Supply required
Response time	Fast	Slow	Slow
Likely to be influenced due to electrical noise	Very likely	Rarely	Likely
Cost	Cheap	Expensive	Cheap

Table 15. Comparison of the TCs, RTDs and ICs

3.5.1. DS18B20+PAR

The DS18B20 is a digital thermometer that comes with a 9-bit to 12-bit Celsius temperature measurement. This device communicates through a 1-wire bus. This device works great with any microcontroller using a single digital pin. Also, it has a usable temperature range from -55 ℃ to +125 ℃ with a ±0.5 ℃ accuracy over the range from -10 ℃ to +85 ℃. In addition, the DS18B20 does not need an external power supply because it derives power directly from the data line. Another cool feature is that this device also comes as a waterproof type. Finally, the size of the DS18B20 is smaller than an American quarter. Table (16) displays more important key features for the DS18B20.

Sensor Type	Digital Thermometer ICs
Typical Temperature Range	-55°C to +125 °C
Typical Accuracy	0.5°C to 2 °C
Long term stability at 125 °C for 1000 hours	±0.2 °C
Power Supply range	3V - 5.5V
Supply current	1 mA (Typical)
Likely to be influenced due to electrical noise	Very likely
Mounting Type	Through Hole
Cost	$3.95

Table 16. key features for the DS18B20

3.5.2. TMP36 (TMP36GT9Z)

The TMP36 from NEWARK is a low voltage, precision centigrade temperature sensor. This device provides an output voltage that is linearly proportional to the Celsius temperature. This device operates at a voltage supply of 2.7 V to 5.5 V. The supply current runs below 50 µA, which provides low self-heating (less than 0.1°C in room still air). Also, it has a usable temperature range from -40 ℃ to +125 ℃ and is compatible with the LM50. This device is also compatible with Arduino and there are many resources on the web using this sensor. Table (17) displays more key features of the TMP36.

Sensor Type	Analog thermometer ICs
Typical Temperature Range	-40°C to +125 °C
Typical Accuracy	± 2°C
Long term stability at 150 °C for 1000 hours	0.4 °C
Power Supply range	2.7V - 5.5V
Supply current	Less than 50 µA
Likely to be influenced due to electrical noise	Very likely
Mounting Type	Through Hole
Cost	$1.40

Table 17. Key features for the TMP36

3.5.3. MCP9800/1/2/3

The MCP9800/1/2/3 from Microchip Technology is a digital temperature sensor, which changes temperature readings from -55°C and +125°C to a digital word. The device has an accuracy of ± 1 °C. This device is manufactured with an industry standard 2-wire, I2C/SMBus compatible serial interface, which allows for sophisticated multi-zone temperature-monitoring applications. So, communication with the sensor is via a two wire bus that is compatible with industry protocols. This device is also a small physical size, low installed cost and easy to use with any microcontroller. Table (18) displays more important key factors we considered when researching this device.

Sensor Type	Digital Thermometer ICs
Typical Temperature Range	-55°C to +125 °C

Typical Accuracy	0.5°C to 2 °C
Long term stability at 125 °C for 1000 hours	± 1 °C
Power Supply range	2.7V - 5.5V
Supply current	200 µA (Typical)
Likely to be influenced due to electrical noise	Very likely
Mounting Type	Surface mount
Cost	$1.20

Table 18. Key features for the MCP9800/1/2/3

3.5.4. Thermometer/Temperature Sensor Choice

Thermometer sensor/Temperature sensor	DS18B20	TMP36	MCP9800/1/2/3
Criteria	Digital Thermometer ICs	Analog thermometer ICs	Digital Thermometer ICs
Precision	0.5°C to 2 °C	± 2°C	0.5°C to 2 °C
Supply voltage	3V - 5.5V	2.7V - 5.5V	2.7V - 5.5V
Supply current	1 mA	Less than 50 µA	200 µA
Size	small	small	small
Cost	$3.95	$1.40	$1.20

Table 19. Thermometer/Temperature sensor comparison

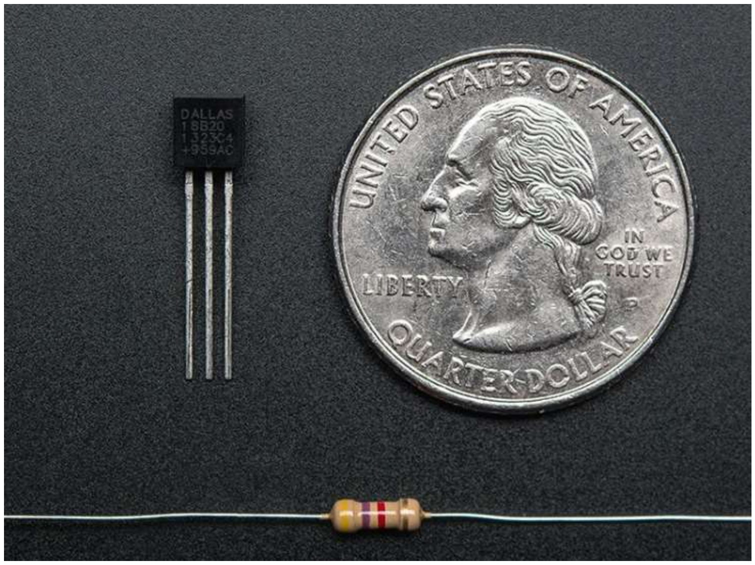


Figure 8. DS18B20 Temperature sensor next to a quarter-dollar and resistor (Courtesy of adafruit)

Table (19) displays the comparisons made between the three sensors researched for our project. Based on the research that was done on three different heart rate monitors and from the comparison made from table (19), we concluded that the DS18B20 sensor displayed in figure (8) was the temperature sensor we would pick for this project. The DS18B20 was capable of measuring temperatures between negative 55 degrees Celsius and 125 degrees Celsius, with a plus or minus 2 degrees Celsius accuracy. The connection of this sensor was also straightforward, all it needed, was an input voltage between 3 volts and 5.5 volts, a ground, and a digital pin. A 4.7 Kohm pull up resistor between the output of the sensor and VCC was needed, without this pull up resistor, the sensor would not work. The DS18B20 cost more than the other two devices researched in this project, but a cost at $3.95 was not that bad. The size was not a problem because it was smaller than a quarter-dollar. Another major reason we chose this device is because it was available to purchase as a waterproof version.

3.6. Accelerometer Module Research

The accelerometer module will be one of the main components for the PetAid harness. An accelerometer module is a sensor used to sense both static and dynamic acceleration. An example for static acceleration is gravity and for dynamic acceleration is sudden starts/stops. One of the most used applications for accelerometers is tilt-sensing. That is, because accelerometers are affected by the acceleration of earth's gravity and the accelerometer module can inform you how it is being oriented with respect to the earth's surface. In addition, an accelerometer can be used to sense if a device is in a state of free fall. For this project the accelerometer is implemented to monitor the different levels of activity in dogs in terms of speed of movement. Because accelerometers can detect changes in specific activities or behaviors, these devices can help indicate early signs of possible health problems for the dog. Some characteristics our group considered when searching for an accelerometer were Range, Interface, Number of axes measured, and Power usage. We also took in consideration the size, cost, and compatibility with our choice of microcontroller for this project.

3.6.1. ADXL345BCCZ

The ADXL345 from ANALOG DEVICES is a small, thin, ultralow power, 3-axis accelerometer module with both I2C and SPI interfaces. This device is well suited for mobile device operations. It also measures static acceleration of gravity when in tilt-sensin//g-applications, and dynamic acceleration resulting from motion or disturbances. The dimensions for this device are 3 mm x 5 mm x 1 mm Land Grid Array (LGA) package. The cost on this device on Digi-Key goes for $7.33. Table (20) displays additional key features for the ADXL345BCCZ.

Sensing Axis	X, Y, Z
Acceleration Range	±2g, 4g, 8g, 16g
Sensitivity (LSB/g)	256 (±2g) ~ 32 (±16g)
Output Type	I2C, SPI
Supply Voltage	2V (Min) – 3.6V (Max)
Operating Temperature range	-40 °C to 85 °C

Table 20. Key features for the ADXL345BCCZ

3.6.2. MPU-9250

The MPU-9250 from InvenSense is a second generation 9-axis Motion Processing Unit that works well with smartphones and wearable sensors. This device has a digital-output triple-axis accelerometer with a programmable full-scale range of ±2g, ±4g, ±8g, ±16g and integrated 16-bit ADCs. The device features I2C and SPI serial interfaces, and a small board design. The operating voltage ranges from 2.4V to 3.6V. The dimensions for this device are 3 mm x 3 mm x 1 mm Quad Flat No-Leads (QFN) package. The cost for this device on sparkfun.com is $14.95. In addition, this device is compatible with Arduino. Table (21) displays additional main features for the MPU-9250.

Sensing Axis	X, Y, Z, Compass, rotational
Acceleration Range	±2g, 4g, 8g, 16g
Sensitivity (LSB/g)	±4800
Output Type	I2C, SPI

Supply Voltage Min	2.4V
Supply Voltage Max	3.6V
Supply current	450 µA
Operating Temperature range	-40 °C to 85 °C

Table 21. Additional key features for the MPU-9250

3.6.3. MMA8451

The MMA8451 is a tiny, low cost, but high precision triple-axis accelerometer with 14-bit ADC that can detect motion, tilt, and basic orientation with a digital accelerometer. It has an acceleration range from ±2g to ±8g and is compatible with Arduino or other microcontrollers. This device has a built-in tilt/orientation detection to let the user know whether the project is being held in landscape or portrait mode, and it lets the user know if the project is tilted forward or backward. It can be used on tablets, phones and other mobile things in the house. So, attaching this device would not be a problem for the project. Also, this device communicates over an I2C, which means it can be shared with other sensors on the same two I2C pins. This device is also safe to use with a 3V or 5V power supply because it has a built-in 3.3 low-dropout regulator and level shifting circuitry on the board. In addition, this device also is equipped with an 8-pin 0.1" standard for the case of using it with a breadboard and two 2.5 mm (0.1") mounting holes for an easy fitting. The dimensions for this physical small device are 3 mm x 3 mm x 1 mm. Lastly, this device can be purchased for $7.95 on Adafruit.com. Table (22) displays additional key features for the MMA8451 that we consider important for our project.

Sensing Axis	X, Y, Z
Acceleration Range	±2g, 4g, 8g
Sensitivity mode	Counts/g
2g	4096

4g	2048
8g	1024
Output Type	I2C
Supply Voltage Min	1.95V
Supply Voltage Max	3.6V
Current consumption	6 µA to 165 µA
Operating Temperature range	-40 °C to +85 °C

Table 22. Additional key features for the MMA8451

3.6.4. LSM303

The LSM303 is a high precision 3-axis accelerometer + compass sensor designed to add motion and direction sensing to any wearable project. This device when purchased on adafruit.com comes with two sensors, one is the 3-axis accelerometer, which can tell the user what direction is down towards earth or how fast the device is accelerating in 3D space. The other sensor included is a magnetometer which senses where the strongest magnetic force is directed into the device. This board/chip uses a digital I2C 7-bit interface. This device is also usable with Arduino and other microcontrollers. The dimensions for this device 3 mm x 5 mm x 1 mm and can be purchased for $14.95 on adafruit.com. Table (23) displays additional key features for the LSM303.

Sensing Axis	X, Y, Z
Acceleration Range	±2g, ±4g, ±8g, ±16g
Output Type	I2C
Supply Voltage Min	2.16V

Supply Voltage Max	3.6V
Current consumption	110 µA (Typical)
Operating Temperature range	-40 °C to +85 °C

Table 23. Additional key features for the LSM303

3.6.5. Accelerometer Module Choice

Accelerometer module	ADXL345	MPU-9250	MMA8451	LSM303
Range	±2g, ±4g, ±8g, ±16g	±2g, ±4g, ±8g, ±16g	±2g, ±4g, ±8g	±2g, ±4g, ±8g, ±16g
Interface	I2C, SPI	I2C, SPI	I2C	I2C
Sensing Axis	X, Y, Z	X, Y, Z	X, Y, Z	X, Y, Z
Voltage operation	2V - 3.6V	2.4V - 3.6V	1.95V - 3.6V	2.16V - 3.6V
Current Consumption	140 µA	450 µA	6 µA to 165 µA	110 µA
Type	Digital	Digital	Digital	Digital
Size	3 mm x 5 mm x 1 mm	3 mm x 3 mm x 1 mm	3 mm x 3 mm x 1 mm	3 mm x 5 mm x 1
Cost	$7.33	$14.95	$7.95	$14.95

Table 24. Comparison for three different Accelerometer modules

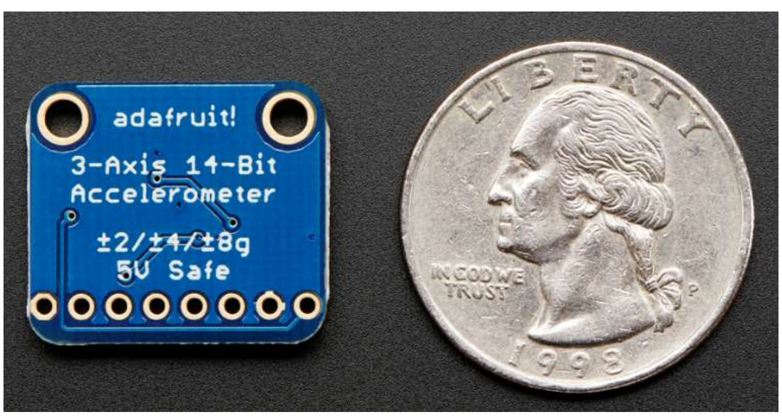


Figure 9. The MMA8451 Accelerometer Module (Courtesy of Adafruit.com)

Table (24) displays the four accelerometer modules our group considered for the project. Based on our needs for the project we decided to go with the MMA8451 breakout module. This device has small full-scale ranges, which are ±2g, ±4g, and ±8. Having small full-scale ranges means there is a more sensitive output, which means we will get a more precise reading out of this accelerometer. Because this device is a digital accelerometer, it is less susceptible to noise and could be integrated to any microcontroller with I2C interface. The Arduino UNO we chose for this project did have the I2C serial interface, which means integrating this accelerometer to the microcontroller should not be difficult. Also, because it is compatible with Arduino, there are a ton of information sources available online for this device. This device also uses 1.95V to 3.6V, which is good if we need the PetAid harness battery to last longer. Figure (9) display the MMA8451 on a board, which means the dimension of this device are now 21mm x 18mm x 2mm and the cost if purchased on adafruit.com is $7.95, which is still cheap considering the price for the MPU-9250 and the LSM303. Overall, our group decided to choose the MMA8451 Accelerometer module for the PetAid harness project.

3.7. GPS and GSM Module Research

The Global Positioning System (GPS) + GSM module is the last sensor or module to be added and covered within our project design. This type of module is a 2-in-1 module that contains both cellular network connectivity and GPS tracking. Initially during the brainstorming process of the PetAid Harness, there was solely going to be a GPS module for tracking the location of a pet instead of also including a GSM module. The group instructor provided this information about a GSM module which can be integrated to be used to communicate between a phone application within a much larger range of distance instead of relying on only Bluetooth or Wi-Fi. Bluetooth and Wi-Fi lack in communication between a mobile phone and the PetAid Harness when it is being utilized at longer distances such as at different locations. GPS is widely used throughout the world and highly recognized by countless mobile phone users. Examples of usage include location for determining a basic position, navigation for getting from one location to another, tracking for monitoring the movement of people and things, mapping for creating maps of the world, and timing by bringing precise timing to the world. This GPS module is a tracking system that is a very important aspect required by the user or pet owner. It is one of the several goals mentioned for the PetAid Harness in the Project Description section, which consists of monitoring and tracking the location of their pet while the pet owner is away from their dog either at home or being taken care of at another location.

The GPS is a worldwide radio-navigation system formed from a constellation of 24 satellites and their ground stations. GPS uses these reference points to calculate positions accurate to a matter of meters. Therefore, implementation of the GPS is a feature that must be incorporated with the PetAid Harness, since the user requirements must be achieved. Wifi/Bluetooth was going to be determined as the main communication between the PetAid Harness and the mobile phone of the user for transmitting and receiving information from the indicated sensors of temperature, heart rate, movement, and location tracking. Then, an addition was introduced to the group by the senior design professor as integrating a GSM as well due to its compatibility for connection of mobile phones. The long distance range of communication is a requirement that was desired by the group which can be completed with a GSM module.

The Global System for Mobile Communications (GSM) is a standard developed by the European Telecommunications and is the network standard for much of the world. While talking about a mobile device, it is either a GSM device or a CDMA device, but the network standard for much of the world consists of Global System for Mobile Communications (GSM). The GSM module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. This will allow the location brought by the GPS to be transmitted and displayed to all mobile phones that contain GSM functionality. Most mobile phones today have GSM available on their 2G or higher SIM card, since the module allows the user to connect to any global GSM network. It either operates at 900 megahertz (MHz) OR 1,800MHz frequency band, and is easy to unlock and transfer to other networks compared to CDMA without SIM slots which are tied to their mobile phone carriers.

3.7.1. Raspberry Pi GPRS/GPS HAT Module

The Tracker HAT for Raspberry Pi is specially designed for tracking applications. It includes necessary parts to easily develop the location-based applications. Accurate GPS positioning with built-in LNA provides better sensitivity. Quad-band GSM/GPRS module at heart of the board will provide reliable data connectivity across the globe. This kind of GPS/GSM module comes with embedded antennas, so there is no need for the mess of antenna cables and connector right out of the box. There are power-sensitive projects in which this module can provide the needed advanced power controls and sleep mode features that come with the board. The GPRS/GPS Tracker HAT is a module to make location-based applications on Raspberry Pi. There are a few important specifications that are obtained about the Raspberry Pi GPRS/GPS Tracker HAT, as shown in the Table (25).

Module	Raspberry Pi GPRS/GPS Tracker HAT
Support AT command	Yes
GSM network	Quad-band: (850/900/1800/1900)MHz
Operating Temperature	-40°C to +85 °C
Supply voltage	5V
Tracking sensitivity	-165dBm

Dimensions (LxWxH)	(65x56.5x8)mm
Cost	$49.00

Table 25. GPRS/GPS Tracker HAT specifications

3.7.2. SIM808 GPS/GPRS/GSM Module by iTead

The SIM808 GPS/GPRS/GSM Module by iTead is a GSM and GPS two-in-one function module. It is based on the latest GSM/GPS module SIM808 from SIMCOM. This supports GSM/GPRS Quad-Band network and combines GPS technology for satellite navigation. It features ultra-low power consumption in sleep mode, and it has an integrated charging circuit for Li-Ion batteries. That makes it obtain a substantial standby time and convenient for projects that use a rechargeable Li-Ion battery. It has high GPS receiving with 22 tracking and 66 acquisition receiver channels. Indoor-localization is available due to support of A-GPS. The module is controlled by AT command via UART and supports 3.3V and 5V logic level power supply. What is included in the box is only the SIM808 GSM/GPRS/GPS module, but there is a possibility that the GPS and GSM antennas can be obtained separately. There are a couple important specifications that were researched and obtained for the SIM808 by iTead, as presented in the Table (26).

Module	SIM808 by iTead Studio
Support AT command	Yes
GSM network	Quad-band: (850/900/1800/1900)MHz
Operating Temperature	-40°C to +85 °C
Supply voltage	5V
Tracking sensitivity	-165dBm

Dimensions (LxWxH)	(27x27x2.8)mm
Cost	$34.00

Table 26. SIM808 by iTead Studio specifications

3.7.3. SIM808 GPS GSM GPRS Module by DFRobot

This GPS/GPRS/GSM module is similar in terms of the chip module used compared to the SIM808 by iTead Studio. The main difference is that the SIM808 GPS/GPRS/GSM is a Shield for Arduino that comes with an available Shield, GPS Antenna, and GSM Antenna. It is very important that an antenna is received along with the SIM808 module, this will help reduce shipping time as well as simplicity when it comes to the hardware subsystem of the GSP+GSM module integrated with the PetAid Harness. The SIM808 GPS/GPRS/GSM Arduino Shield is an integrated quad-band GSM/GPRS and GPS navigation technology Arduino shield. The size might be large when meeting the size requirements of the PetAid Harness, but it should be expressed that its size is still comparable with a credit card size. This module is set to the standard Arduino pin packaging, which makes it compatible with Arduino UNO, Arduino Leonardo, Arduino Mega, and other mainboards from Arduino. Compared to the previous generation SIM908, SIM808 made some improvements on performance and stability. Another GPS/GPRS/GSM module like the SIM808 by DFRobot, is the Elecrow SIM808 GPRS/GSM/GPS/Bluetooth Shield. This Elecrow module is like the SIM808 by DFRobot and can be selected if the SIM808 by DFRobot is not available in stock since it is very popular and due to the constraint of not being able to acquire parts in the quantity desired. There are a couple important specifications that were researched and obtained for the SIM808 by iTead, as presented in the Table (27).

Module	SIM808 by DFRobot
Support AT command	Yes
GSM network	Quad-band: (850/900/1800/1900)MHz
Operating Temperature	-40°C to +85 °C
Supply voltage	5V
Tracking sensitivity	-165dBm
Dimensions (LxWxH)	(18x25.5x3.1)mm

Cost	$39.99

Table 27. SIM808 GPS/GPRS/GSM Shield for Arduino (Courtesy of DFRobot.com)

3.7.4. GPS + GSM Module Choice

GPS+GSM Module	Raspberry Pi GPRS/GPS Tracker HAT	SIM808 by iTead Studio	SIM808 by DFRobot
Support AT command	Yes	Yes	Yes
GSM network	Quad-band: (850/900/1800/1900) MHz	Quad-band: (850/900/1800/1900) MHz	Quad-band: (850/900/1800/1900) MHz
Operating Temperature	-40°C to +85 °C	-40°C to +85 °C	-40°C to +85 °C
Supply voltage	5V	5V	5V
Tracking sensitivity	-165dBm	-165dBm	-165dBm
Dimensions (LxWxH)	(65x56.5x8)mm	(27x27x2.8)mm	(18x25.5x3.1)mm
Cost	$49.00	$34.00	$39.99

Table 28. Comparison of three different GPS+GSM Modules

Figure 10. SIM808 GPS GSM GPRS Module (Courtesy of DFRobot.com)

There are three possible GPS+GSM modules that are taken into consideration during our research and decision-making process. The goal when deciding our GPS+GSM module is to make sure that it can utilize both the Global Positioning System (GPS) and Global System for Mobile Communication (GSM). These two standards allow us to track the location and display real-time information from the PetAid Harness. All three GPS+GSM modules provided in this section can perform both GPS and GSM operations. The Raspberry Pi GPRS/GPS Tracker HAT module is specially designed for tracking applications and reliable data connectivity, but two important specifications for which it is the best possible choice is that it is not easily compatible with Arduino and the cost is highest at $49.00. Balancing our budget and ease of integration are two factors for which the Raspberry Pi GPRS/GPS Tracker HAT module is not selected.

On the other hand, the SIM808 GPS/GPRS/GSM Module by iTread is compatible with Arduino and is of the cheapest cost at $34.00 to better meet our budget, but there is a drawback for which the SIM808 by iTead can't be selected as the best possible choice. The GPS and GSM antenna are both not included with the SIM808 module by iTread. This should not be of much issue to obtain these antennas, but if they are ordered separately, it will drastically increase the total cost of our finances and budget that is attempted to keep as low as possible. Increase of the time for the parts to be shipped and received, will also produce a delay for prototype parts of the PetAid Harness. Especially since given the change of the class from on-campus lecture to online and the required social distancing, the entire group has been affected in ways that neither we nor the professors could predict or control. Thus, shipping for the items to be received by the group will have a duration that is longer. Table (28) represents some specifications considered for making our decision of the GPS+GSM module.

Due to time limitation for this project, our group did not use the GPS and GSM module for the PetAid Harness. Even though we bought the module, we were unable to integrate the module to our project. There was not enough time to get the code to work like we wanted with the PetAid Harness. If we had more time, our group would have implemented the GPS and GSM module to the PetAid Harness.

3.8. Power Supply Research

During the power supply selection process, our group decided to focus on using a battery power supply. Batteries offer a way to store electrical potential energy in a portable container. A battery's capacity is a measure of the amount of electric charge it can deliver at a specific voltage. These capacities on most batteries are rated in amp hours (Ah) or milliamp hours (mAh). The nominal voltage of a battery is the voltage stated by the manufacturer. Usually, the nominal voltage refers to the average voltage of the battery over its discharge cycle. First, we considered whether the project required primary or secondary batteries. Primary batteries are typically alkaline, lithium primary or carbon-zinc batteries and are single use, which means once they lose their charge, the batteries are unavailable to use again. Secondary batteries are typically nickel-cadmium, lead acid or lithium ion batteries and are rechargeable batteries, which means once the battery discharges, they can be charged again and reused on any load. For this project, our group decided to go with primary batteries and concluded we would focus on Alkaline batteries. Alkaline batteries are the most popular batteries today. These batteries are sold in every store, so they are great for projects that need to be user serviceable. The alkaline batteries come in different sizes that range from AAA to AA, C, D and 9-volt. Figure (11) displays a comparison of the first four types of batteries.

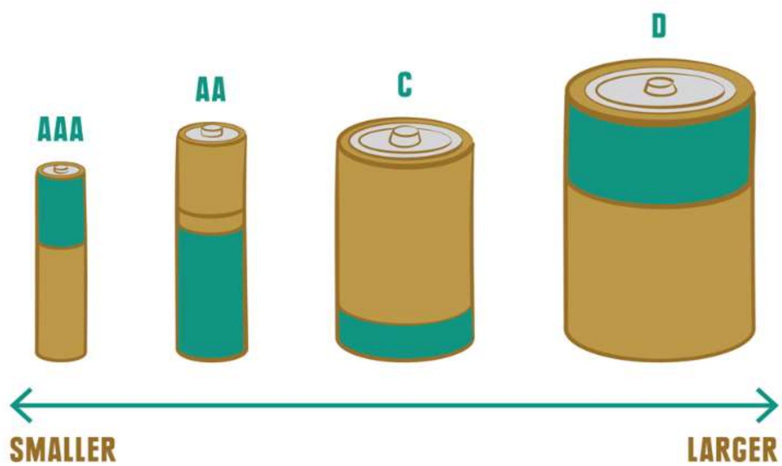


Figure 11.

Figure (11) displays the four types of batteries, which the letters indicate the size. So, the higher you get through the alphabet, the larger the battery. Also, when the letter is used more than once, such as AAA, then the smaller the battery. During the research of information for the battery power supply for the PetAid Harness, there are four important characteristics that were taken into consideration to meet the engineering and user requirements.

1. Safety
2. Size/Weight
3. Rechargeable/Non-rechargeable
4. Battery cycle life/ Time duration of Recharge cycles

If the voltage is too high or too low for our circuit, we will likely need a DC/DC converter.

3.8.1. AAA Battery

The Alkaline AAA battery is the smallest of the 4 types of batteries compared in figure (11). This battery is typically used in small electronic devices such as MP3 players, CD players, digital cameras, radios, and toys. The AAA battery has a battery capacity of 1000 mAh and has a typical drain of 10 mA. A single battery cell has a nominal voltage of 1.5 volts. It weighs 11.5 grams and has an external diameter of 10.5 mm and an external height of 44.5 mm. The cost for one single AAA battery cell is $2.49. But most stores sell them in packs of 4, 8, and 16. Buying them in larger packs is recommended because they are much cheaper. In addition, most stores sell them in packs, and you would need to shop for individual batteries on the internet. The price may also very depend on the brand of the battery. The three most common AAA Alkaline battery brands are Duracell Ultra Power Alkaline Battery, Panasonic Evolta Alkaline Battery, and the Kirkland Signature Alkaline Battery.

3.8.2. AA Battery

The Alkaline AA battery is small but bigger than the AAA battery. This AA battery are used in many similar devices such as the AAA battery. The AA Alkaline battery can also be used for electronic devices such as Wall Clocks, Remote Controllers, Flashlights, and many outside devices that control lights. The AA battery has a battery capacity around 2400 mAh and has a typical drain of 50 mA. A single battery cell has a nominal voltage of 1.5 volts. It weighs 23.0 grams and has an external diameter of 14.5 mm and an external height of 50.5 mm. The cost for one single AA battery cell is $1.42. As mentioned before, it is recommended to purchase the Alkaline AA battery in packs rather than individually to save some money.

3.8.3. C Battery

The Alkaline C battery is a bigger size than the AA battery. This C battery is also used in most similar electronic devices that use AA batteries. In addition, C batteries are used in specific applications that have a steady demand such as motorized toys, portable game consoles, and electronic musical instruments. The C battery has a battery capacity around 6000 mAh and has a typical drain of 100 mA. A single battery cell has a nominal voltage of 1.5 volts. It weighs 66.0 grams and has an external diameter of 26.2 mm and an external height of 50.0 mm. The cost for one single C battery cell is $3.00. As mentioned before, it is recommended to purchase the C Alkaline battery in a pack, which the pack typically comes in 2 or 4 batteries.

3.8.4. D battery

The Alkaline D battery is the biggest from the four batteries compared in figure (11). This battery is typically used in high current drain applications, such as large flashlights, radio receivers, and transmitters. They are good for devices that require longer running time. The D battery has a battery capacity around 13000 mAh and has a typical drain of 200 mA. A single battery cell has a nominal voltage of 1.5 volts. It weighs 139.0 grams and has an external diameter of 34.2 mm and an external height of 61.5 mm. The cost for one single D battery cell is $2.59. These batteries are typically sold in packs that come in pairs of 2, 4 8, and 16. As mentioned before, it is recommended to purchase these batteries in a pack rather than individual purchases.

3.8.5. 9 Volt Battery

For the purpose of power supply research only, our group decided to include the 9 volt Alkaline battery as an additional battery supply. This battery is typically used for walkie-talkies, Clocks, and most home smoke detectors. The 9 volt battery has a battery capacity around 550 mAh and has a typical drain of 15 mA. A single battery cell has a nominal voltage of 9.0 volts. It weighs 45.0 grams and has an external diameter of 26.5 mm and an external height of 46.4 mm. The 9-volt battery can be purchased at Walmart for $8.99 in a 2 pack. The price may vary depending on the brand. The better the brand the more expensive.

3.8.6. Power Supply Choice

Our thought process in deciding the optimal power supply is to get the most battery life for the cost, size, and weight, and to make sure it operates at a temperature that's safe to have on a piece of wearable technology. In Table 29, we have a comparison of four common batteries and their properties that we looked at.

Classification	Alkaline AAA	Alkaline AA	Alkaline C	Alkaline D
Nominal voltage	1.5V	1.5V	1.5V	1.5V
Operating Temperature	-18°C - 55°C	-18°C - 55°C	-18°C - 55°C	-18°C - 55°C
Weight	11.5 g	23.0 g	66 g	139g
Shelf life	10 years at 21°C	10 years at 21°C	10 years at 21°C	10 years at 21°C
Battery Capacity	1.25 Ah	2.85 Ah	8.35 Ah	20.5 Ah
External diameter	10.5mm	14.5mm	26.2mm	34.2mm
External Height	44.5mm	50.5mm	50mm	61.5mm
Cost	$2.49 (each)	$1.42 (each)	$3.00 (each)	$2.59 (each)

Table 29.

Table (29) displays important specifications for a AAA battery, AA battery, C battery and D battery.

The knowledge of recognizing the information displayed in table (29) is important and crucial when the PetAid Harness is being used and worn by the dog. Most of the time, the PetAid Harness will remain on 24 hours a day whether the pet owner is constantly receiving information or just monitoring once a day at the end of the day. The assumption is that the user or pet owner will need the PetAid Harness to be running for long hours and for at least a full 24 hours preferably. Accessibility is key when it comes to having a reliable and long-lasting battery which should provide an access for the pet owner to keep track of when to replace or recharge the battery. An example of this accessibility is if the pet is lost or in an abnormal condition, location tracking information must always be needed to provide the pet proper care and treatment. Comparison of the pros and cons about whether a rechargeable or non-rechargeable battery should be used for this project is obtained. A sample of data is considered to provide an example calculation between rechargeable batteries versus non-rechargeable batteries to determine which battery type is the better investment for the budget of our group and reliability. Information presents that rechargeable batteries have a lower mAh than non-rechargeable batteries. However, it is later pointed out that after two years, rechargeable batteries will suffer from a greater internal discharge that increases with time.

Component	Current draw	Input Voltage
AtMega328P-PU	1 mA	7V - 12V (recommended)
ESP8266	240 mA	3.0V - 3.6V
SEN 11574 Heart-Rate sensor	4 mA	3V - 5.5V
DS18B20 Temperature sensor	1.5 mA	3V - 5.5V
MMA8451 Accelerometer module	11 mA	1.95V - 3.6V
GPS + GSM module (Not implemented in final design)	500mA	3.4V
Result	The total current drawn is 258 mA	Nominal voltage at 3.5V. 9V for MCU

Table 30.

Figure 12.

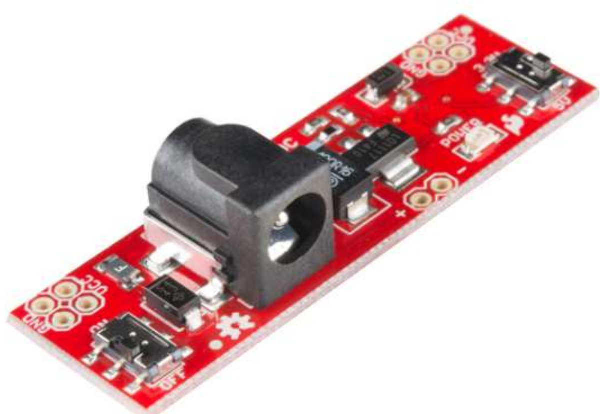


Figure 13: Breadboard Power Supply 5V/3.3V output.

Before our group decided on which type of Alkaline battery we would use for the project, we built a table that displayed the total current consumption and voltage needed for the PetAid to function. Table (30) display the devices our group have chosen for this project and their max ratings. Table (30) displays the total current drawn from the system would be around 146 mA and from the voltages provided, we decided to set the nominal voltage at 3.5V for all the sensors using a voltage regulator. We decided to use two power supply sources for this project. This includes the 9V battery with DC barrel jack 5.5x2.1mm and also a breadboard power supply with a 5V/3.3V regulated output. The 9V Alkaline batteries have a 550mAh battery capacity, which means with most sensors included, the system would function for about 1 hour and 18 minutes. That is not enough time considering the presentation time is 25 minutes. The formula used to calculate the battery life was, Battery life = Battery Capacity in mAh/Load in mA. In addition, using a 5V/3.3V power supply would be enough to operate the ESP8266 WiFi module at a current consumption of 240mA. To connect the power supply, we would need to buy a battery case that can hold a 9V battery. The battery case costs our group decide to go with cost $3.95 on Digi key and has the required power plug fitting for the Arduino board. It has DC barrel jack dimensions of 5.5x2.1mm diameter. The case is shown in figure (12).

3.9. Software Requirements

In order to collect and analyze the data from each of the biometric and geolocating peripherals in real-time, we need programming that allows for multiple streams of data to be recorded, have statistics taken, and displayed simultaneously to a GUI. With the MSP family of microcontrollers manufactured by Texas Instruments, we conveniently have TI's Code Composer Studio, an Integrated Development Environment (IDE) that has a host of tools to help develop and debug applications that the microcontrollers will have. Firstly, we'll need to specify what we're going to need the microcontrollers to be doing, and when we'll need them to do it.

3.9.1. Engineering Specifications

The first distinction among specifications that we'll want is the different values being recorded. Our current goal is to monitor 4 values with the dog harness: body temperature, heart rate, location, and velocity. We monitor the temperature and heart rate to verify the health condition that the dog is in, and the location and velocity verify the position and potentially the method of travel that the dog is in. These metrics will also need to be analyzed over time to recognize patterns and may need to be analyzed in conjunction to gain more information.

According to VCA Animal Hospital, the healthy body temperature for a dog is between 101 to 102.5 degrees Fahrenheit (38.3 to 39.2 degrees Celsius) and is most accurately measured either digitally--in the ear--or rectally.[1] Our best bet would likely be a digital thermometer, and our design would need to minimize irritation as much as possible to avoid scratching, and possible forced removal by the dog. Seeing as how this temperature is not likely to drastically change very often, we can save processing power by checking it as often as once a minute, or even larger spans of time.

According to Rover, the heart rate of a dog is typically anywhere from 60 to 140 beats per minute (BPM) and is accurately measured by counting pulses felt in the chest over 15 seconds, and multiplying the number by 4.[2] This is likely significantly easier and safer to measure than temperature, as it is non-intrusive. Seeing as how there is a given timeframe for heart rate measurement, and this is a value that is likely to shift drastically fairly often from minute to minute, it seems sensible to measure constantly, and store the value counted every 15 seconds, with a counter incrementing by 4 for every pulse noticed by a touch sensor.

The accelerometer and locator are simpler to test and implement because they are measuring the position and velocity of the entire dog as opposed to the internal activities or conditions of the dog. These conditions, however, are the most resource-intensive to record, because they are best measured with as fast a frequency as possible, require more frequent statistical analysis, and in the locator's case, requires the use of a GPS. If we wanted to conserve resources, we could remove the accelerometer and extrapolate average velocity with the locator, and potentially lower the frequency of the location pings, and/or activate the locator only when requested by the user. The accelerometer and locator should be able to switch between an on and off state at the very least, and ideally the locator should have an adjustable refresh rate to let the user decide on performance vs. battery life. Illustrated in Table (29) are the different variables that are contained in each of the 4 measurements that will likely be modular in the Settings Menu.

Name	Refresh Rate	Uptime	Adjustable	Power Usage
Heart Rate Monitor	15 seconds	Constant	No	Low
Body Temp. Monitor	10-120 seconds	Constant	Yes	Low
Accelerometer	1/30th second	On request	Yes	Med
GPS Locator	1-300 seconds	On request	Yes	High

3.9.2. Statistical Analysis

The second distinction we need to make is what we want to do with the history of recorded values of each type. Each metric has a unique refresh rate, a different unit of measurement, and different meaning for its value.

With body temperature we want to notify the owner if the dog falls below 99 degrees Fahrenheit or exceeds 104 degrees Fahrenheit,[1] and with heart rate we want to notify if the dog's heart rate falls below 60 BPM, or above 140 BPM.[2] Since each dog's state of health is unique, these have potential for some adaptive processes, such as raising or lowering the bar for what is healthy for your dog after receiving an alert. Useful records can be compiled by taking the average, low, and high temps and heart rates for the dog daily, and taking the weekly, monthly, and yearly averages of those values over time. This helps in noticing patterns for the dog, and is vital reference material for veterinarian visits, either for a checkup or an emergency visit.

While the accelerometer has little use for analysis over time, the GPS locator could benefit from it the most of all the metrics. If the home location and X-meter radius from its home are established, an alert can be set up for when the dog has potentially become lost or has been taken. With the implementation of subdermal chips for pets to more easily be located, the vest could function alongside the chip, giving more accurate results, or compiling the information given by the chip with the other metrics and interpreting it for the PetAid app.

With these values being recorded, certain conclusions can be drawn when looking at the combination of 2 or more values; if both temperature and heart rate are high, the dog is likely running outdoors, and if heart rate is high and the accelerometer shows little movement, the dog is likely anxious, or otherwise in an urgent physically stressful state. If the accelerometer shows a speed that is well over what a dog could run--20 miles an hour[3]--it can be safely concluded that the dog is in a vehicle. These conclusions can all be checked for and have alerts as responses when the conditions are met. These conditions can also be read as a 'Status' in the GUI of the PetAid app.

3.9.3. PetAid Application Requirements

The phone application that will display the information collected and deduced by the microcontrollers on the vest must be simple, user-friendly, and effective. To meet these goals, we need to keep all the information relevant to the user readily available. Based on our personal experience with phone applications, we will try to keep all the important information reachable within 3 taps. Important information includes the body temperature, heart rate, velocity, location, status/condition, and settings to adjust measurement frequency. A 'Help' option will also be available for troubleshooting: this is where FAQ could be, a 'Contact Us' button that lets the user submit a ticket with feedback on bugs with the application, and possibly access to the console text, or even a debug window. Those options, however, will not necessarily need to be immediate, considering that they are used in cases where the app is having problems, which is not a state that should be happening often.

3.9.4. PetAid Application Layout

The Application layout should be as follows: upon first opening, a register window should open, connecting an email address for an account, and fields for the phone number and home address. There can be an optional checkbox for allowing push notifications and text alerts, but it should be strongly suggested to check these off, as they help warn the owner of unusual dog conditions. Once the registration is complete, the Home screen should have the option to connect with a PetAid vest by pairing, and either a toolbar on a side of the UI that contains the Home, Settings, Help, and Sign Out buttons. Once a PetAid vest is registered, the Home window then shows the interface for that collar, and the button to add another collar should be beneath this interface.

In the main menu of the app detailed in Figure 13, in each vest's interface there will be the name and a picture of the dog it belongs to, and five values underneath: the Status, Body Temp (Fahrenheit), Heart Rate (BPM), Velocity (miles per hour, kilometers per hour, or meters per second) and Location shown with a pin in a satellite view. Location is measured by the proximity of the vest with its home location and is given the Far value when outside of a certain radius from that point.

Each condition status has a different level of severity, and there should be checkboxes for what to be notified about in the Settings area to better suit the desires of the owner. If Notifications are a section of the Settings menu, there would only be two clicks to reach it (one to open the app, and one to open the Settings menu) but Notifications can be its own sub menu and still meet our convention of 3 maximum clicks. In Figure 14, the most important states to check for are just abnormal readings of any of the 4 variables checked: abnormal heart rate and body temperature, unexpected distance from home radius, and to a lesser extent the kinetic activity of the dog. Table 26 shows the current idea for what the Boolean values will be that need to be met for each of the basic conditions to be met. These will be used in the statistical analysis of the linked lists of data collected over time.

Status	Temp	BPM	Velocity	Location
Active	High	High	High	Close
Overheating	High	Low	Low	Close
Anxious	Low	High	Any	Close
Relaxed/Asleep	Low	Low	None	Close
Roaming	Any	Any	Any	Far

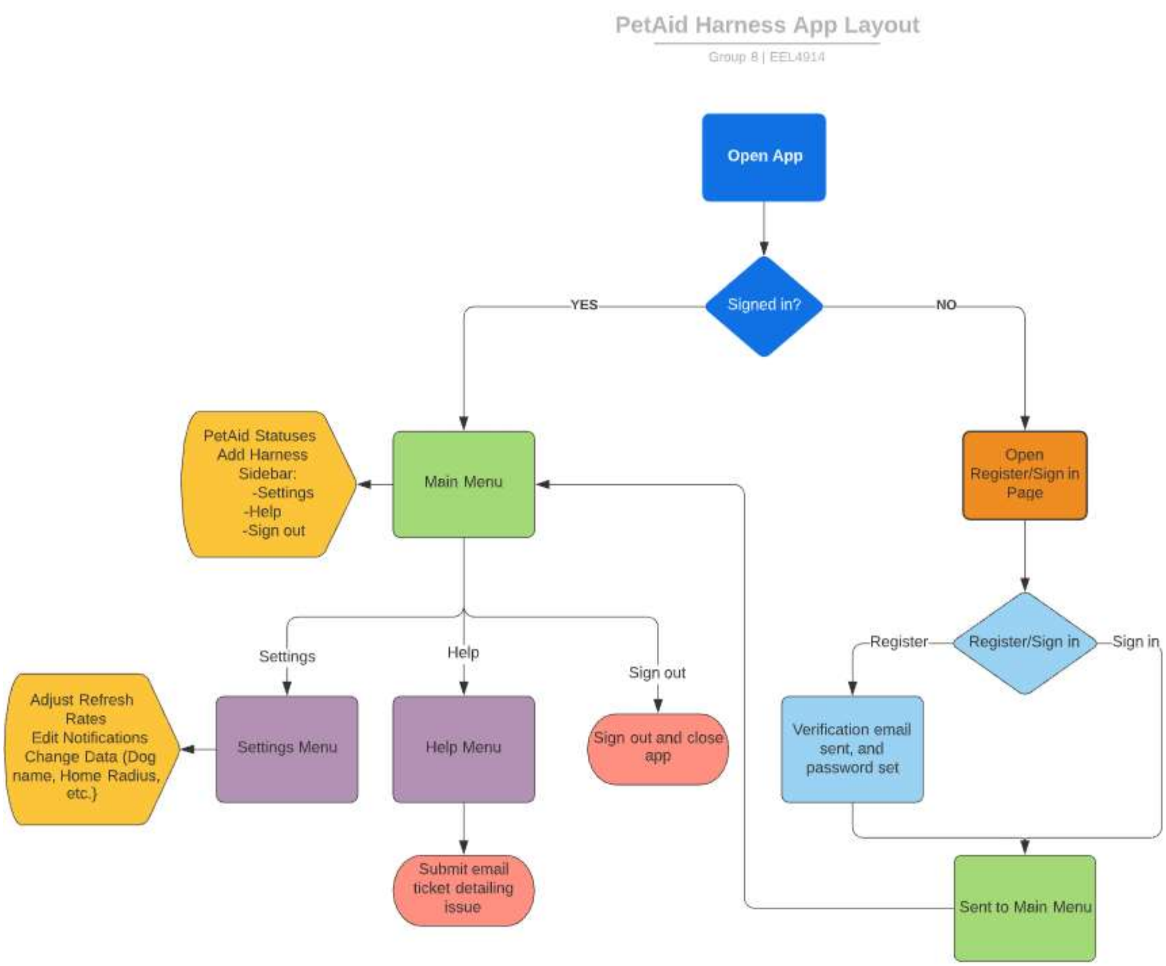


Figure 14: PetAid application layout

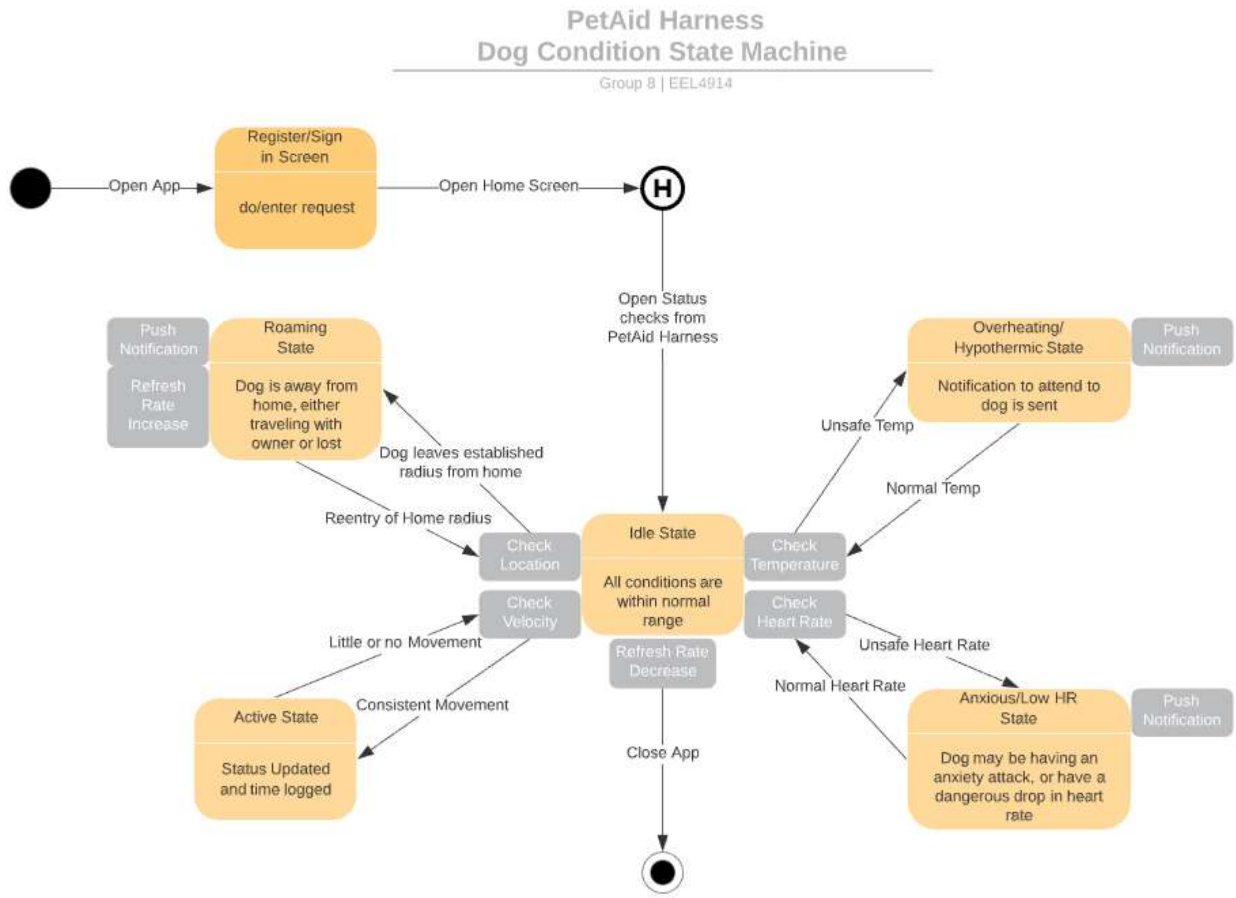


Figure 15: PetAid pet condition state machine

When in Settings, the option to adjust the frequency of the refresh rate for the Locator should be available to save power. The checkbox for turning the Accelerometer on can be changed into having it monitored only when the app is open. The Help button should lead to a screen with text boxes to fill the subject and description of a complaint, which is sent directly to a team email address. Sign Out does as stated, sign the user out, and return to the sign in/register menu.

3.10. Code Composter Studio

The IDE that supports the MSP family microcontrollers with the most resources for optimization of processor cycles, debugging tools, and uniformity in coding language would be TI's very own Code Composer Studio. This IDE comes with an app center that contains various useful tools, such as a GUI composer, task viewer, a resource explorer for you to find the latest libraries, datasheets and more, among several other features that are meant to make the programming of these devices maintain either C or C++ consistently and smoothly throughout all the microcontroller's functions. The GUI composer could prove to be useful for us in the development of the app, as it would be most efficient to maintain the same coding language throughout the development cycle, especially in a language like C, which most if not all of the team is familiar with. The App Center has a plethora of microcontroller specific compilers, which provide additional support for whichever device we decide to run, a task viewer which improves communication for group projects, and helps for when a project is paused for a long enough time to lose track of what the next goal is, 3D modeling support with MATLAB, compatibility with a Git repository for multidimensional edits and version management of a set of code, and multiple methods of testing and debugging code.

3.10.1. IAR Embedded Workbench

The IAR Embedded Workbench is meant mostly for debugging and compiling in C and C++. The codes that are most likely to have bugs are the Status condition code and the statistical analysis for the four primary sources. Viewing the project of coding realistically we acknowledge the inevitability of bugs in the code and will undoubtedly need to tweak code to better fit our purposes during our testing.

3.10.2. Arduino IDE

While MSP family microcontrollers have a large amount of support for coding in C, Arduino also has its own IDE that uses a higher level language: C++. While not that far from C in terms of sophistication, this higher level of abstraction and greater human readability can prove to be invaluable for the speed of our project, and the ease of communication between members, as well as for those unfamiliar with the project. Arduino has both an online and offline IDE, with the former being what was used for our project. The Arduino IDE has multiple libraries available for each of the devices that we used for this project, to the point where some were too specific to be used for the general family of devices that they were meant for. This however ensured that when the correct library and method of coding were employed, the exact desired output we looked for would be the outcome.

3.11. Dog Harness Research and Choice

To guarantee the long term safety of both the electronics and the pet wearing the device we performed significant research on the dog harness that would hold the device. Listed below is a brief summary of the products we came across with a summary chart presented in Table #

- RabbitGoo Adjustable Oxford No Pull Dog Harness

  A large amount of dogs pull with a significant amount of force while on a leash which can lead to restricting their throat and overall pain for the pet. The RabbitGoo Adjustable Oxford No Pull Dog Harness aims to remove this issue with a patented no-pull design which gives you better control of the dog by attaching the leash to the front clip. When the dog pulls it will be turned around and in the opposite direction of where it was initially trying to go, encouraging it not to pull and to move at the pace its owner is moving at. It comes with multiple rings attached to the harness so that the owner can attache at different points for different walks. It has four easy adjusting straps around the body which allow a perfect fit for your dog along with some room for growth. The harness is made of a durable nylon oxford and padded with soft cushioning for a pleasant dog walk each and everytime. The mesh is also breathable enough to keep you dog cool while minimizing potetial injuries. It also has reflective strips for safe night time walks. We believe this is excellent for our design as the adjustable harness allows for room for the installation of our Pet-Aid electronic devices. We also appreciate the versatility that comes with the multiple different rings in different places on the harness. This will carry on to the consumer and allow them flexibility when using our product with the Pet-Aid electronics incorporated in it. We also think that the breathable mesh will help prevent the electronics from breaking our temperature standards and harming the dog. It also comes in multiple colors which allows for customization.

- Blueberry Pet Reflective Multi-Colored Stripe Dog Harness Vest

The Blueberry Pet Reflective Multi-Colored Stripe Dog Harness Vest is made of a lightweight mesh, polyester, and oxford materials to maximize comfort. It also includes reflective strips and a large reflective triangle built into the harness to ensure the dog is visible at all times of the day. Its connector is a heavy-duty metal D-ring which provides maximum durability and ease of leash attachment. The chest and neck straps are adjustable and the harness is dual layered for pull control and weight distribution. Polyester seams keep the mesh layer and oxford layer tightly together and contrast the vest in color for added style. It is sold at a reasonable $26.99 and is well within our price range. It features even more colors than the Rabbitgoo Adjustable Oxford No Pull Dog Harness, but cost more as well which is a negative. It is more than large enough to fit our electronics, but it is not breathable as the other dog harnesses.

- Lifepul No Pull Dog Vest Harness/Body Padded Dog Vest

The Lifepul No Pull Dog Vest Harness/Body Padded Dog Vest was designed for owners looking for a harness for a dog that pulls itself a lot or needs to be trained. The padding is soft and breathable to ensure your dogs comfort even in very hot weather. The harness has a no pull design to bring better control of your dog. The stitching is made of reflective material along with reflective strips built into the harness. It has an easy buckle system to make putting the harness on your dog go as smoothly as possible and without any hiccups. the padded nylon material is long lasting and washable to provide a long and extended product lifetime to the consumer. The most unique feature that the Lifepul No Pull Dog Vest Harness/Body Padded Dog Vest provides is a handle on the top of the harness. This is great for owners with service dogs wo need to be in close contact with their dog for their own health purposes

- Petsafe 3 in 1 Harness

The Pet safe 3 in 1 Harness is designed to be an all-around harness. It was designed with the dogs comfort as the top priority and is promoted to be perfect for everyday use. It has multiple leash attachment options to provide the owner with flexibility when taking their dog on a walk. It is unique in that it has a car safety restraint option designed to decrease your dog’s movement in the car which increases the overall safety of all passengers including the pet. It can be ordered with a special 2-point control leash for additional no pull encouragement. It is an excellent option for very active dogs who will be moving constantly throughout the day.

For our final harness decision we decided the best fit for our product and mission statement was the Rabbitgoo: Tactical Dog Harness, Adjustable Oxford No Pull Dog Harness. The adjustable harness was the first major plus for this harness as our pick. The next deciding factor was the fact that the harness had multiple different attachment points which gave the user multiple degrees of freedom. The harness also has plenty of room for our sensitive electronics while still providing a safe and comfortable feel for the dog. The breathable material limits thermal upkeep and provides the dog with an added safety measure. It is made of the highest quality material which assures our consumers that our product is made with thought and the dog in mind. The cushioning and cheap price provide the final selling point for this to be the harness that represents our product. An example of the harness is provided in Figure # and a quick comparison chart is presented in Table #.

Figure #: Rabbitgoo: Tactical Dog Harness, Adjustable Oxford No Pull Dog Harness

Name	Cost	Size	Manufacturer
Rabbitgoo: Tactical Dog Harness, Adjustable Oxford No Pull Dog Harness	$23.99	Adjustable	Rabbitgoo
3M Reflective Mutlit-Colored Stripe Dog Harness Vest	$26.99	Small, Medium, Large	Blueberry Pet
Lifepul No Pull Dog Vest Harness/Body Padded Dog Vest	$25.79	Small, Medium, Large, Xtra Large	Lifepul
PetSafe Three in One Harness	$31.99	Xtra Small, Small, Medium, Large	PetSafe

4. Design Standards

Design standards allow us to take a function or idea and easily implement it using previously determined technologies. These standards not only allow our product to serve its basic functions, but also serve to make it as modular as possible and allow for interoperability between devices. Standards also allow us to compare different ways of accomplishing a function to see which standard is most efficient and economical for our product. By ensuring as many design standards are followed as possible in a design an engineer encourages simplicity for future work on that design and for any future users. Design standards allow economical mass production of parts so that individual custom pieces are not required for each new design. They protect consumers by ensuring that engineers can understand each other's design which encourages product durability and safety. In other words, standards provide a common rhetoric so that engineers can measure and evaluate performance in as efficient and safe an environment as possible.

4.1. IEEE 802.15.1 Bluetooth Standard

The IEEE Project 802.15.1 derived a Wireless Personal Area Network standard based on the Bluetooth v1.1 Foundation Specifications in 2002. The IEEE 802 standards committee published this standard on June 14th, 2002 (1). IEEE no longer maintains this standard by reviewing it every 5 years like it does with other standards, but it still holds and is used today. This standard state that the device must operate at least in the 2.4 GHz range and support the transmission of voice and data at the same time. (1)

4.2. Microcontroller Data Communication Standard

There are multiple different standards of communication for microcontroller devices. These standards are called "protocols" and have very different requirements for communication, speeds of data transmission, and ease of implementation. The difference between microcontroller communication protocols and regular communication protocols like Universal Serial Bus (USB) is the lack of a need for high speed data transmission. Having the correct communication protocol is essential because the code used to transmit data is built around a specific protocol. If a device is set-up to communicate via a certain type of communication protocol, and the receiving device is set-up to interpret a different communication protocol then the data will not transmit correctly, and incoherent data will be received. It is much akin to requiring each device to "speak the same language" in order to talk to each other. Stated below are some of the different types of microcontroller communication standards, or "protocols".

4.2.1. UART

The Universal Asynchronous Receiver Transmitter (UART) protocol is a two-wire serial communication protocol. One of the main benefits of the UART protocol is this need for only 2 wires. UART communication involves two UART devices communicating directly with each other. The transmitting UART takes parallel data from a CPU at its data bus and converts it into serial form then adds a stop bit, start bit, and parity bit before sending it. Serial data means that the information is sent bit by bit and not multiple bits at a time. Therefore, the UART system only needs two wires, one for the serial transmitting data, and one for the serial receiving data. Once the data reaches the receiving UART it is converted back to parallel data at the receiving UART's data bus and the stop bit, start bit, and parity bit are removed. Each UART device has a Tx and Rx pin. The Tx pin transmits data over one line to the Rx pin of the receiving device. Unlike SPI UART transmits data asynchronously, which means there is no need for a clock signal line or master and slave relationship. To avoid the need for a clock line UART uses stop and start bits at the beginning and end of data to break it into packets. This way the UART knows when to start sampling and stop sampling data without a clock. To keep data transmission coordinated and avoid errors in received data each UART device is set to a certain Baud rate, a measure of data transfer speed. Once the receiving UART detects a start bit it begins to sample the bits at a specific frequency. If the receiving and transmitting baud rate are not aligned, then the data will be sampled at incorrect times and produce nonsense data. Fortunately, there is about a 10% maximum difference in baud rates between devices while still maintaining okay data transfer which gives a reasonable margin for real world manufacturing error. (4)

The packets transferred by UART have a very standardized structure. The data transmission line is usually held at the high voltage level when no data transmission is occurring. To signal the beginning of data transmission the transmitting UART pulls the line to low voltage for one cycle; this is known as the start bit. After reading the start bit the receiving UART begins to sample data at its set baud rate. The data being read by the receiving UART is called the data frame and can be from 5-9 bits long, or 8 max bits if a parity bit is used. In most scenarios this data is sent by least significant bit first. The parity bit is a way for the receiving UART to check for potential data change during transmission, almost like an insurance policy. The parity bit describes whether a data frame has an even or odd amount of 1 bit. A 0 parity bit means there is an even number of 1 bit in the data and a 1 parity bit means there is an odd number of 1 bit in the data frame. When a receiving UART reads a data frame it counts the number of 1 bit and checks if the value is even or odd before comparing it to the parity bit. This means if a bit has changed from a 1 to a 0 or vice-versa the parity bit will not match the count for the receiving UART and the UART knows that data errors have occurred via electromagnetic radiation or misaligned baud rates. Unfortunately, the parity bit only signals if the number of 1 bit is even or odd, not the exact number of 1 bits. This means that theoretically if two different bits were to change the UART would not register an error in data transmission. Finally, to signal the end of a piece of data the sending UART returns the data transmission line from low to high voltage for a minimum of two bit durations. (4)

4.2.2. SPI

Serial Peripheral Interface, or SPI, is one of the common standards for microcontroller communication. Common attachments that tend to communicate through microcontrollers through SPI include SD card modules, RFID card reader modules, and 2.4 GHz wireless transmitters and receivers like Bluetooth modules. Unlike the other two communication standards we researched SPI has the unique benefit of being a continuous stream of data and not breaking the data into packets. In other words, it can be transferred without interruption because it does not use the same stop and start bit data broken into packet use. Devices communicating with SPI have a master slave relationship. The master, like a CPU, has full control of communication and the slave device, like a sensor or memory chip, takes full instruction from the master. The communication takes place over four wires/nodes; Master Out/Slave In (MOSI), Master In/Slave Out (MISO), Signal Clock (SCLK), and Slave Select/Chip Select (SS/CS). While one master can control more than one slave, daisy chaining of the slave select nodes may be required if the master device only has one slave select pin. The SCLK wire synchronizes the data output bits to the receiving bits of the slave. One bit is transferred per clock cycle and the maximum flow of data is therefore capped by the clock rate. Because the master generates and determines the clock signal communication must always be initiated by the master. This synchronized clock signal makes SPI what's called a synchronous communication. The clock signal can be edited by configuring the clock polarity and clock phase. Clock polarity determines if the output bit is sampled on either the rising or falling edge of a clock cycle, while the clock phase determines whether the bit is sampled on the first or second edge of the clock cycle. Slave Select is self-explanatory, it provides a way for the master to tell which slave to sample the output. It is configured to be at the high voltage level normally with the master setting whichever slave it wants to talk to a low voltage level when it's ready to communicate. The MOSI and MISO lines are used to move the actual data. The MOSI line sends data from the master to the slave and MISO line does the opposite, each bit by bit. Unlike the MOSI line, for the MISO line the least significant bit is usually sent first. (3)

4.2.3. I^{2}C

Inter-Integrated Circuit communication or I2C communication is special in that it allows for both a single master controlling multiple slaves or multiple master controlling single or multiple slaves. This is useful for things such as having multiple microcontrollers access data from the same memory card. Like UART, only two wires are required to transmit data between the communicating devices. One line is the Serial Data line (SDA) which allows the master to send and receive data. The other line is the Serial Clock line (SCL) which carries the clock instructions. One thing to notice is the difference in acronym between this standard and SPI communication.

I2C is a serial communication protocol which means that data must be transmitted bit by bit. The clock line is required because I2C is a synchronous communication protocol like SPI. The master is in full control over the clock signal. I2C communication uses “messages” that are broken into “frames” of data. Each message contains an address frame signaling the binary address of the slave it wants to communicate with, and one or more data frames that contain the desired data to be read. The messages also contain start and stop conditions along with read write bits. ACK/NACK bits are located between each data frame. The start condition is represented by the SDA line switching from a high voltage to a low voltage before the SCL line switches from high to low. The stop condition is represented by a low to high switch on the SDA line after a low to high switch on the SCL line. The Address Frame is a 7 or 10 bit sequence unique to each slave that identifies them. Due to this restriction, the maximum number of slaves is a little less than (2^10)/4, 1008. The read/write bit is a single bit that instructs the slave on whether it is receiving data (low voltage) or transmitting data (high voltage). Each frame is followed by an acknowledge/no-acknowledge (ACK/NACK) bit. An ACK bit is returned to the transmitting device if a data or address frame was received successfully. The address frame is always the first frame after the start bit. Each slave connected to the master receives this address frame, and each slave compares their own address to the received frame. Every slave that does not match keeps the SDA line high while the slave that does sends a low voltage ACK bit back to the master. The last bit of each address frame is always the read/write bit. A data frame is always 8 bits long and sent the most significant bit first. Every data frame is followed by an ACK/NACK bit to verify that the frame was received correctly. The next data frame will not be sent until an ACK bit is returned. After all data frames are transmitted the master can halt flow by sending a stop condition to the slave.

Single master with multiple slave operations is not overly complicated. As mentioned before, the maximum number of unique addresses a master can send is dependent on the size of the address bit. The normal size is 7 bits providing 128 unique addresses. Multiple masters controlling multiple slaves is slightly more complicated. In order to ensure that multiple masters do not access the same slave at the same time each master checks the SDA line before beginning transmission. As mentioned before, the SDA line is switched to low voltage as part of the start condition. Therefore, if a master notices the SDA line is set to low voltage then it knows to wait before accessing that slave.

4.2.4. Standard Comparison Chart

Here we are comparing the different standards for the microcontroller’s communication, with UART being different in that it is a standalone IC, while the SPI and I2C are communication protocols.

	UART	SPI	I2C
Max Transfer Speed	115200 bps max, normally 9600bps	10Mbps	5Mbps
Asynchronous?	Yes	No	No
Number of Wires	2	4	2
Number of Masters/Slaves allowed	1/1	1/Unlimited	Unlimited/1008
Parallel or Serial Communication?	Serial	Serial	Serial
Successful Data Transfer Check?	Yes, Parity Bit	No	Yes, ACK bit
Level of Hardware Complexity	Most	Least	Middle
Main Advantages	No clock signal needed, Checks for successful data transfer	Fastest transfer rate, Supports multiple slaves, No stop/start bit means uninterrupted transmission, Data can be sent and received at same time	Checks for successful data transfer, Supports multiple masters and slaves
Main Disadvantages	1 to 1 communication, Slowest transfer rate, Baud rates must match within 10%, Data frames limited to 9 bits	4 wires, No check for successful data transfer,	Data frame limited to 8 bits,

4.3. Heart Rate Module Standards

Heart rate indicates the soundness of our heart and helps assessing the condition of cardiovascular system. In clinical environment, heart rate is measured under controlled conditions like blood measurement, heart voice measurement, and Electrocardiogram (ECG), but this can also be measured in a home environment also. Our heart pounds to pump oxygen-rich blood to our muscles and to carry cell waste products away from our muscles. The more we use our muscles, the harder our heart works to perform these tasks such as how our heart must beat faster to deliver more blood. A heart rate monitor is simply a device that takes a sample of heartbeats and computes the Beats per Minute (BPM) so that the information can easily be used to track heart condition. There are two types of methods to develop heart monitors- electrical and optical methods. The electrical method has an average error of 1 percent and average cost of $150. The optical method has an accuracy rating of 15 percent and an average cost of $20, which is what we will using with our PetAid Harness due to the cheaper cost. The normal BPM of a dog's heartbeat ranges from 60 to 140 beats per minute in a resting dog.

Due to the human and animal testing constraints that are involved when demonstrating a project design such as the PetAid Harness, the heart rate monitor testing will be different when compared to testing of other components involved for the PetAid Harness. Universities do not allow such type of testing; thus, the testing of the heart rate monitor cannot be performed on humans or animals. The heart rate monitor requires a heartbeat to read and detect data. Since it requires a heartbeat, this limits the different methods that can be done to accomplish testing. The other sensors involved such as the GPS/GSM module, temperature sensor, accelerometer, and WiFi/Bluetooth module do not require data to be read necessarily from a human or an animal in order to test. Heart rate data can be really useful whether you're designing an exercise routine, studying your activity or anxiety levels. The issue when compared with a temperature sensor, is that heart rate can be difficult to measure. To solve this issue, that is why our group decided on using the SEN-11574 Pulse Sensor.

4.4. Accelerometer Module Standards

The accelerometer has a default sensor, which is the device's main accelerometer sensor. A latest reading for a sensor such as the accelerometer sensor, is that it includes three entries whose keys are "x", "y", "z" and whose values contain a device's acceleration about the corresponding axes. There are also other values that must be considered when applying a three-axis accelerometer which is used for the PetAid Harness. Values can contain a device's linear acceleration or gravity depending on which object was instantiated. Definition of acceleration must be set as a standard when utilizing accelerometers. The acceleration is the rate of change of velocity of a device with respect to time. Its unit is meter per second squared (m/s^2) in SI unit format. The frame of reference for the acceleration measurement must be inertial, such as, the device in free fall would provide 0(m/s^2) acceleration value for each axis. The sign of the acceleration values must be according to the right-hand convention in a local coordinate system.

The device coordinate system is defined as a three-dimensional Cartesian coordinate system (x, y, z), which is bound to the physical device. For devices with a display, the origin of the device coordinate system is the center of the device. If the device is held in its default position, the Y-axis points towards the top of the display, the X-axis points towards the right of the display and

Z-axis is the vector product of X and Y axes and it points outwards from the display, and towards the viewer. The device coordinate system remains stationary regardless of the screen orientation.

4.5. Temperature Sensor Standards

Given the importance of a temperature's sensor's absolute accuracy as the ambient environmental temperature reference in sensor-fused awareness system (not device temperature), it is very important to fully understand the primary contributors to absolute temperature error. The following parameters require thorough specification and are intended for digital sensors only. Note that these parameters should also be applied to temperature sensors that are integrated into other sensors, are often used for thermal compensation, and are exposed to the user.

- FSR (Full scale range)
- Digital bit depth
- Absolute temperature error
- Sensitivity
- Noise
- Current consumption
- Integral non-linearity
- Transition time
- Long-term drift

This standard presents a standard methodology for defining sensor performance parameters with the intent to easy system integration burden and accelerate time to market (TTM).

4.6. GPS and GSM Standards

Analysis of the current handset market reveals the growing role of connectivity features in mobile terminals, such as the PetAid Harness. Apart from calling and text-messaging functions, data transfer and localization are the most important services offered by network providers.

4.7. Alkaline Power Supply Standards

While standards do indeed increase the overall safety and efficiency of design and production, the standards an engineer needs are not always free. Unfortunately, such was the case for our lithium battery standards. We originally wanted to use either International Electrotechnical Commission (IEC) Standard 61960, American National Standards Institute (ANSI) standard C18.2M and C18.2M, Part 2, or Institute of Electrical and Electronics Engineers (IEE) Standard 1625. All of these standards cost almost double what we predicted our product prototypes overall cost would be. We therefore were unable to purchase them, and a brief summary of what little information we could find on these standards online will be provided below.

- "American National Standards Institute (ANSI) standard C18.2M and C18.2M, Part 2 is a comprehensive battery standard covering general information, battery specifications and safety standards for consumer batteries of all types (standard alkaline AA, 3A, C, D, etc; lithium coin cells; rechargeable NiMH; etc.)." - batterypoweronline.com
- "This standard establishes criteria for design analysis for qualification, quality, and reliability of rechargeable battery systems for multi-cell mobile computing devices. It also provides methods for quantifying the operational performance of these batteries and their associated management and control systems including considerations for end-user notification." - IEEE.org

4.8. Dog Care Standards

Our project has the added component of being used by a living animal and we therefore had to ensure we understood the proper standards for dog safety. We focused on adopting a safe standard for maximum carry weight. Unfortunately, we found no available standard for dog carry weight and therefore had to develop our own standards. We did this by weighing ourselves and then carrying increasing amounts of books in a backpack. After we had added a book, we rated our comfort from 1-5, with 5 being the highest, and weighed the backpack (results in Graph # and Table #). Finally, we calculated what percentage of our body weight each interval was and decided on a limit. To ensure safety we stopped the experiment when the person carrying the backpack felt a comfort level of 2. We wanted to ensure as conservative of a weight restriction as possible so that the pet can wear the collar over long periods of time without feeling much discomfort. Therefore, we decided as a group that we would take the first data point with a 3 for discomfort level and go a few data points before that to obtain our maximum carry capacity for the pet as a percentage of its total body weight. We decided that the 18 Ibs/10.91% data point was a good conservative limit and were finally ready to declare our new weight standard for the PetAid Harness.

<u>*Standard PAH-1.1*</u>*: Any product designed to be worn for extensive periods of time by a household pet, or any land based living animal, shall not exceed 10% of the total body weight of the designated carrier animal. This is to ensure the long term safety of the designated carrier based on a few experimental trials we conducted on our group members.*

Total Weight (Ibs)	Percentage of Carrier's Weight	Level of Comfort Rating (1-5)
3	1.82	5
5.5	3.33	5
8	4.85	5
11	6.67	4
15	9.09	4
18	10.91	4
20	12.12	4
25	15.15	3
27	16.36	3
30	18.81	3
33	20.00	2

Along with our newly developed weight standard, we wanted to ensure that our measurements were following standard veterinary procedures to provide the optimal readings and increase safety. For temperature readings we wanted to ensure an accurate reading to give prompt notifications if the dog had abnormal physiological response. Accurate readings also decrease the amount of false alarms. The current standard we found for the most accurate readings was to insert a lubricated thermometer an inch into the pet's anus and wait 60 seconds. We found this standard far too invasive for a device designed to be worn long term and therefore had to find a different standard for temperature readings. Luckily, we found a backup standard for pets that are uncomfortable with the normal invasive procedure. By ensuring direct skin contact we can get a moderately accurate measurement. We plan to use this standard of taking dog temperatures.

Finally, we also wanted to find a temperature standard for devices worn by house pets. Unfortunately, we yet again could not find firm temperature standards for pet smart devices. This meant we had to develop our own standard once again. After doing research online we found that normal body temperature for dogs and cats is 101-102.5 degrees (2). In order to ensure the safety of the pets we decided to pick a normal operating temperature well below the average dog temp which we decided was 90 degrees. With this research we were ready to declare our new temperature standard for pet smart devices.

Standard PAH-1.2: Any product designed to be worn for extensive periods of time by either a cat or a dog shall not function at a temperature higher than 90 degrees for 90% of its operating time and will never exceed 100 degrees at any point during operation.

4.9. PCB Standards

A significant amount of research was applied when understanding the Printed Circuit Board and how it is produced. This led to us producing a list of tips, standards, and practices for producing our own printed circuits boards while ensuring correct electrical operation. Those tips and practices are listed below. (Edwin Robledo and Mark Toth)

- Ensure that the correct grid spacing is applied. Extra thought applied to the beginning of the design can avoid spacing issues while maximizing board usage.
- Attempt to keep trace lengths as direct and short as possible. This should be prioritized over avoiding going back over parts of the layout again. Impedance and parasitic effect can be minimized this way, and many circuits, particularly high-speed digital circuitry, will perform much better.
- Use a power plane to manage distribution of power lines and ground. Using pours on the power plane allows a significant amount of copper to common connections and will have power flow working at maximum potential with reduced impedance, and therefore voltage drop. This also ensures that ground return paths are up to snuff.
- Try to keep related components and test points as close together as possible. If a device uses a capacitor or resistor it helps to ensure that those components are located as close to the device using it as possible. This limits impedance and voltage drop and makes testing and fault-finding much easier.
- Try to replicate the board you need several times if a specific producer only produces a specific size larger than required or if specific non-custom sizes will save a significant amount of production cost.

- Ensuring that component values has a small range of standards. This will make the BOM simpler and make the overall production less expensive.
- Use the DRC (Design Rule Check) function of the printed circuit board software as often as possible. Checking as you go can make all the difference on complicated designs and makes a good habit to form.
- Do not attempt to optimize your design at the cost of reliability or safety. This means always using decoupling capacitors. They are cheap and can handle a robust amount of perturbation so be sure to slide them in whenever possible. Be sure to use a range of standardized values to keep the inventory list concise.
- Whenever possible keep polarized components pointed in the same direction. This avoids unnecessary confusion and ensures that parts are not connected improperly.
- Generate your own printed circuit board manufacturing data and confirm it before shipping it to fabrication to be produced. Use a free viewer to ensure it looks as pictured using your own outputted Gerber files. This limits unnecessary confusion while catching errors inadvertently.

5. Design Constraints

During our time designing the PetAid harness we came across a myriad of significant design constraints. Our most important constraint was the safety and comfort of the animal using the device. Our design is always meant to be worn so that as much data about the dog's health can be collected as possible. Therefore, we were extremely cautious about any long-term effects our design could create and aimed our focus on three aspects we thought were most essential for safety; overall weight, heat generated, and potential shock risk. As for comfort, we opted for a harness over a collar so we could better distribute the weight of the electronics for the pet which came with its own challenges.

Along with the unique constraints of designing a product to be worn by a living animal, we also had realistic design constraints as college students. Manufacturing, environmental, economic, ethical, health and safety, and time constraints all limit our ability to bring our product idea to its maximum potential. These constraints molded many of the final design decisions we decided to pursue, and therefore accurately upholding them was our most important focus.
A final constraint was durability. We wanted to ensure a product that could last for years even with the intense environments being strapped to a house pet can bring while still ensuring an economical cost. Many options for material were available, but we decided to make the casing out of a combination of 3d printed plastic and rubber to prevent water leakage.

5.1. Economic Constraints

All members of our design team are currently in college with no full-time income. Therefore, we were unable to invest a large amount of capital into our design and manufacturing. For instance, we were unable to obtain a current market product that performs some of the same tasks as our design because the cost of the market product was more expensive than our entire planned design by far. If we had more economic power, we would have spent much more money on research and development by purchasing multiple products and analyzing their strengths and weaknesses. We would have also bought and made multiple prototypes if we had more capital. With more funds we would have been able to buy multiple different versions of each of our sensors, microcontrollers, and other manufactured materials. By running tests and data analysis on all of these parts individually we could narrow down what combination of them would produce the best product for our goals. We would have then built multiple different prototypes using both our preferred parts and a more economical version. Afterwards, we would have seen what performance measures the preferred prototype had over the economical one and work on which measures can be given a higher margin of error. The inability to make these analyses and comparisons was by far our biggest economic constraint.

Our next design constraint was a price point lower than the average of other products we researched online. After looking at other products that accomplished a similar goal, we found a reasonable price point to be 120$. Our goal was to provide a product that could be manufactured for a significant amount less than the 120$ price point in order to ensure as many consumers can provide the detailed health analysis for the pets they need. This led to many design trade-offs. Whenever we felt we could decrease performance a reasonable amount for economic benefit we took the opportunity.

Our current cost estimate for parts is:

- MCU - $11.42
- Thermometer - $6.40
- Heart rate monitor - $3.53
- GPS unit - $29.95
- Accelerometer - $7.22
- Wi-fi/Bluetooth module - $14.07
- Battery - $14.95
- Harness - $15.98

- Total Cost = $103.5

As can be seen from our cost estimate, we are below our price point for our prototype while providing the added stability of a full-on harness. While our prototype is not too far cheaper than current retail price points, we believe that if expanded to a full-scale manufacturing level the cost per harness would decrease even further. For instance, many of the small electronic sensors and devices can be bought in bulk at a bulk price. Unfortunately, we do not have the economic power to buy these devices in bulk and pay the required manufacturing fees to produce the units quickly. This inflates the total cost of our prototype which we believe is still a solid price point. Also, depending on consumer response we may want to develop a lower performance economical option in our product line which may increase these constraints further.

The economic constraints of maintenance are low as the harness is designed to be durable and the only electronics that need maintenance is the battery. This is easily alleviated by replacing the battery or charging the unit. All this maintenance would fall upon the consumer and the maintenance information would be conveyed in the packaging with the product via a small brochure. This means that we have basically no maintenance constraints economically and are free to design as we wish when it comes to electronic part durability. We aim to have our product functioning optimally for at least 4 years.

Another economic constraint falls with physical durability and waterproofing. Due to our low amount of economic capital we are forced to build with materials that are easily accessible and low in cost. Therefore, we decided to not 3d print our casing or use rubber for waterproofing. The 3D print lab and other 3D printing manufacturers were not available due to the closing of many vendors for the pandemic health factors. A small container found online through Amazon is used as recommended by our professor in order to provide the proper casing that is both tightly and functionally stored for our PetAid Harness which is constantly worn by the pet. We believe this is not an adequate amount of protection for how long we want our product to last and therefore limits our design. If we had less economic constraints, we would organize a waterproof durable case design with a third party. We would then obtain the necessary factories and industrial machinery to produce this design efficiently and economically which would once again allow us to lower our price point for the consumer.

Finally, our last economic constraint is we are limited to technology that has already been developed because we have no capital for research and development. If we did not have this economic constraint, we would begin our design by asking which electrical components in the device are hampering our performance the most. Then, we would fund research into those parts to see if we can find economical shortcuts or even performance increases via new production processes. This would open our options and allow us to design the product completely to our vision.

5.2. Time Constraints

All members of the design team do not work on the design as part of a full-time income. This means all members have significant commitments which they must battle while designing the PetAid Harness. If this was a company with full time employees, we would be able to focus 40 hours a week just on product development. Unfortunately, with our current college workload and job commitments we cannot commit near that amount of time weekly. This time constraint leads us to not being able to put as much time into our design as we would like to and decreases our overall product quality.
We also must ensure that our prototype can be functioning within the course of a single summer semester which greatly limits the amount of design and testing we can do. If we did not have such a short deadline, we would focus more time on product research. We would want to reach out to experts in the field and talk to consumers to see what functions they desire most. Then, we would put our product through multiple testing phases for long periods of time before bringing it for demonstration. Unfortunately, these luxuries are not possible and constrain the quality and functionality of the device.

Assembly is also another large time constraint. Due to the small nature of our design team we are required to assemble the entire PetAid smart harness by hand from its parts. This leads to long amounts of time just putting the product together. If we had an assembly plant for our product, we would no longer have this time constraint. This would allow us to focus all time spent on hand assembly towards engineering and design of the product. This would increase the overall quality of the product and decrease cost per device.

Finally, our project has a unique time constraint due to the COVID-19 coronavirus. During development of our product the entire state of Florida was put under a state of recommended social distancing. This meant that large gatherings were highly discouraged, and the UCF campus which was our primary meeting spot was mostly shut down. This limited our meetings greatly. We were only ever able to meet via online meeting apps which greatly slowed down our ability to convey technical information. Also, testing and building our prototype became much more time restricted as well as we wanted to minimize the amount of times we met together in person as a full group. This meant we had to plan our testing in advance and did not get to do as much hands on design together as a group as we would have liked. Unfortunately, our final deadline could not change, and we were therefore left with the same amount of design time but a much less efficient design process. If we did not have this constraint, we would prioritize meeting in person as much as possible to ensure that everyone is getting hands on design input and that technical information can be conveyed as swiftly as possible.

5.3. Ethical/Testing Constraints

We decided to build a device that could be used on live house pets specifically because we believe it fills an important need that will benefit all house pets and their owners' long term. Our dedication to this goal did come with some significant constraints do not present in other products. Not only is it illegal in the state of Florida to just test a product on a house pet with no regulation or expertise, we also found it very unethical. Risks of testing biometric scanning technology in a dog harness include having a catastrophic failure that could electrocute the dog, or otherwise harm them in ways we did not expect. Therefore, we had to find unique ways of testing the functionality of a product designed for a dog that cannot be attached to one. First, we decided that all sensor testing would be done by a group member that agreed explicitly to the tests. Second, we had to create representations for the physiological actions and responses a dog might have. For the temperature sensor we attached the sensor to a gallon jug of water heated to a predetermined temperature. For the accelerometer we recreated common dog motions like rolling by attaching the device to the end of a cylinder and rolling it. Light breathing as simulated by attaching the device to a group member for a short period of time. Another test we developed for both the GPS module and the accelerometer was to ring the device in the automobile of a design team member and drive around the same block five times at a near constant speed. We then compared the data on the device with our route and known speed.

5.4. Health and Safety Constraints

As stated previously our number one constraint and priority was safety of both the design/testing team and the safety of the pet. This meant constantly considering the comfort and health of the house pet during design, and how we would test the product safely.
Our first constraint came with temperature sensing. We quickly found that an accurate temperature reading was a moderately invasive procedure and had to compromise on our performance goals. Instead of having the temperature sensor wherever we wanted on the device we had to ensure it was somewhere on the harness that had the most direct contact with the pet. We believe this to be under the armpit of the harness so that it is pressed firmly against an area with minimal fur. This however greatly increases friction and heat on the sensor while still providing a sub-par

reading. If we did not have this comfort constraint due to the temperature being taken from a living creature, we would not have to worry about sensor placement in design and could create a better performing device. If we can find a digital thermometer that works comfortably, we will try to replicate that as closely as possible, and potentially improve on the design.

We also became very aware that our product would always work at its highest potential if attached to the pet . This yet again came with health and safety constraints. If the device was always to be worn , then we needed to minimize its physical presence on the dog. This meant that both weight and heat were a huge factor in product design. Instead of being able to freely design a tracking product we had to limit our design to ensure we held up to these health constraints. To hold to this constraint we developed our own standards for how much a device worn by a live animal should weigh and how much heat a device strapped to a cat or dog should generate and defined these in the Standards section (4.0.0) as Standard PAH-1.1 and Standard PAH-1.2. After defining these standards, we explicitly stated the constraints of our device. For weight we noticed that most pet smart devices were rated for pets that weighed 8 pounds and up which we used as our baseline. Using this number and Standard PAH-1.1 we defined our max permissible device weight, casing and all, as 10% of 8 pounds; .8 pounds or 12.8 ounces. For temperature we first researched normal healthy pet body temperatures and then developed our own standard in Standard PAH-1.2 listed in Section 4.0.0. We then declared our design constraint to be that the device casing shall not function at a temperature higher than 90 degrees for 90% of its operating time and will never exceed 100 degrees at any point during operation. This constraint also helps to prevent overheating of the electronics in the casing. Without this constraint we would be able to generate much more heat allowing for much higher electrical performance.

For both the safety of our testers and the potential designated carrier we also had to apply constraints to the physical material of the casing. To avoid shock hazard, we had to ensure the product casing was made of a non-conductive plastic. This led to durability and economic restrictions for the casing. If we did not have this requirement, we might strive for a material that is still lightweight but more durable while being conductive. We also have to waterproof the device heavily due to the strong possibility that the pet will wander into humid environments while wearing the harness. This means we had to spend extra economic and time resources towards assembling and designing rubber fillings to ensure shock safety. If we did not have this restriction, we could cut weight and cost from the device.

5.5. Manufacturing Constraints

Manufacturing and economic constraints are deeply intertwined. With unlimited economic capital we would also have theoretically no manufacturing constraints except the physical constraints of the manufacturing process itself. For instance, even with unlimited capital we would still be unable to design a perpetual motion machine.

As stated, before our prototype must be assembled by hand. As UCF students we have access only to the manufacturing processes that UCF allows us. This means that we do not have the capability to craft every part to our exact specifications. This constrains us to have to use materials and parts that are more modular and can be easily formed or assembled by hand. This constraint is one of the main reasons we decided to use 3d printed plastic for any electronic casing. UCF has access to multiple 3d printers which allows us to bring a design and produce a casing quickly and without complicated manufacturing processes.

If we had access to any manufacturing process, we would prioritize building a casing that is more lightweight and durable than 3d printed plastic. It would also allow us to potentially build the board as one unit instead of having to order individual parts and assemble the total product by hand. These manufacturing constraints also lead to a loss in reliability as we are not able to manufacture more accurate sensors.

Due to these manufacturing constraints we would most likely have to provide a limited warranty to a user who purchases the PetAid Harness. The lack of consistent manufacturing processes for each part of the device would lead to reliability uncertainty that may affect the device.

With access to more complicated manufacturing processes we would also have had more options when it came to research and development. For instance, if we had access to semiconductor manufacturing, we may be able to build a very custom board for our project that greatly improves performance.

Another component affected by a manufacturing constraint is the actual harness itself. If we had access to our own manufacturing processes, we could develop a lightweight durable harness designed to have an electronic casing slide easily and comfortably into it. Instead, because of our manufacturing constraint we must buy an existing collar that is not meant to hold electronics and then carefully attach the electronics ourselves. Electronics may become easily damaged or get inaccurate readings due to the harness not being designed to have electronic attachments. Pieces could fall off or vibrate excessively leading to long term failure. If we did not have this constraint, we would manufacture our own harness design with a padded pocket for the smart harness electronics and casing.

Finally, the last manufacturing constraint is also a time constraint. Due to being college students we may not have access to processes that accomplish what we would like in a faster manner. For instance, because we cannot assign our own manufacturing processes, we cannot have a part be instantly built when we want it. The third party manufacturer may have other parts with priority which may lead to a backlog causing a delay in shipment of our part. The manufacturer may even have a specific high end and efficient process that is only run for expensive parts or parts in bulk. Without this constraint we would ensure our parts are always manufactured as soon as we need them and with the highest quality materials and manufacturing processes.

6. Project Hardware and Software Design details

Our goal is to create a product that is future-proof; to use hardware that has room for expansion, and software that is as resource efficient as possible. Another goal we want to keep in mind is to create a product that is resilient; when dealing with pets, the concept of wear and tear is less of a question of ‘if’ than ‘when.’ That being said, we want to make the hardware as protected and as difficult to tamper with as possible. In regard to software, we are looking for a minimal amount of space usage and rigorous documentation in the code to ensure a clearly understood software for the team as well as any viewer.

6.1. Microcontroller Controller System Layout

With the Arduino Uno selected as our microcontroller, our next step is to find the accommodating hardware for the biometric readings, the location, and the communication with the phone. In Figure 12, we show the pin layout for the Arduino Uno, which shows the amount of space we have to work with in order to fit all of our peripherals onto the board comfortably. Ideally we want to have extra space for any add-ons we may decide are appropriate in the stretch goals, but this is not a necessity. We are prioritizing finishing a prototype that completes as many of our minimum goals as possible before we consider the stretch goals.

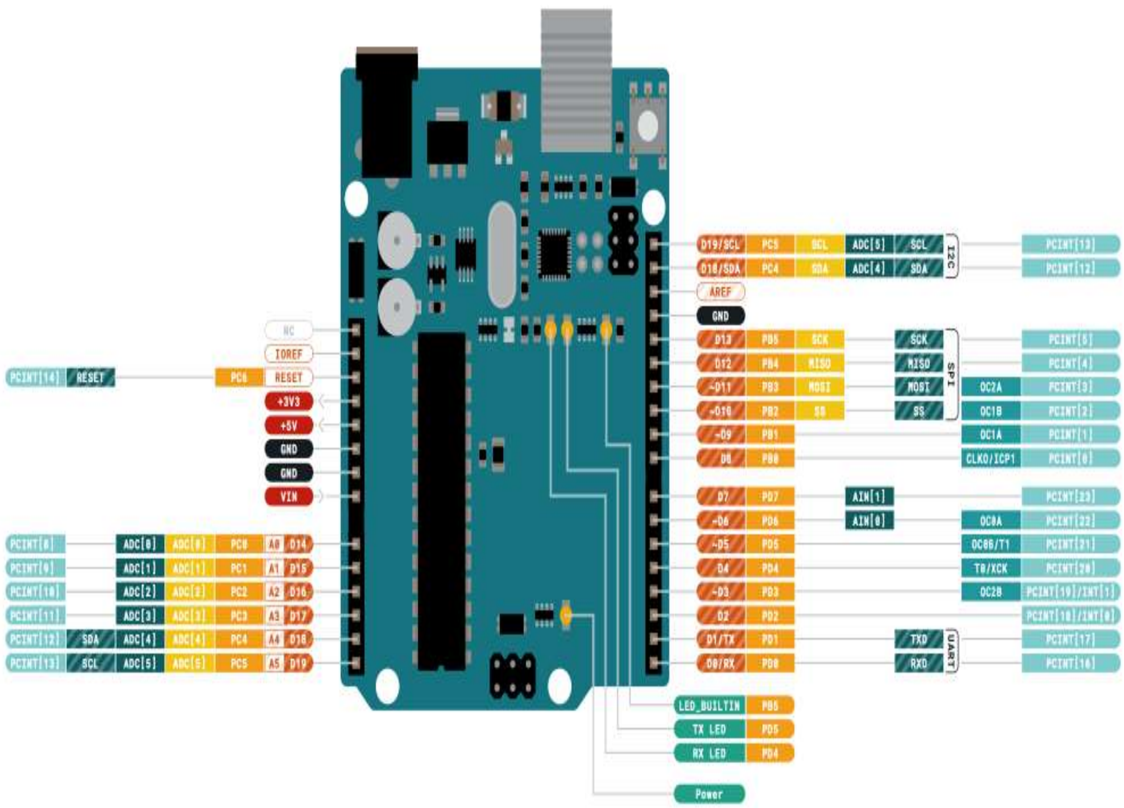


Figure 16: Arduino Board Pin Layout.

6.2. WiFi Subsystem

ESP8266 Wi-Fi Module is the leading component and most useful as defined by the group to meet our objectives and goals. Creating a compact system that can obtain both Wi-Fi & Bluetooth wireless technology into one single module, was essential in our design strategy. After much deliberation we agreed on not using the ESP32 development board and instead use the ESP8266-S01 module board to accomplish this goal. While the ESP32 has enough computing power to run as its own microcontroller, we will be connecting it to the Arduino Uno to organize and receive data to send. The Arduino and ESP32 are both microcontrollers and there is no need to utilize both of them, since only one microcontroller will facilitate a much easier, efficient design for the PetAid Harness. An ESP8266 WiFi module provides a superior integration with the Arduino board. In order to accomplish this, we need to make four connections: the TX pin, the RX pin, the Vin pin, and the GND pin. First, we connect the ground pin to the GND bus on the Arduino Uno. Then we connected the TX pin on the ESP8266 board to the Software Serial RX pin on the Arduino Uno, Pin 4 as shown in Figure 15. After that we connected the RX pin on the ESP8266 board to the Software Serial TX pin on the Arduino Uno, Pin 5. Although, pin 4,5 are not labeled as RX and TX pins, respectively on the Arduino board pin layout, these two pins are configured within the Arduino IDE code environment using Software Serial. Finally, we connect our +9V battery or the alternate 6-12V breadboard power supply source to the Vin pin on the ESP8266 WiFi Module.

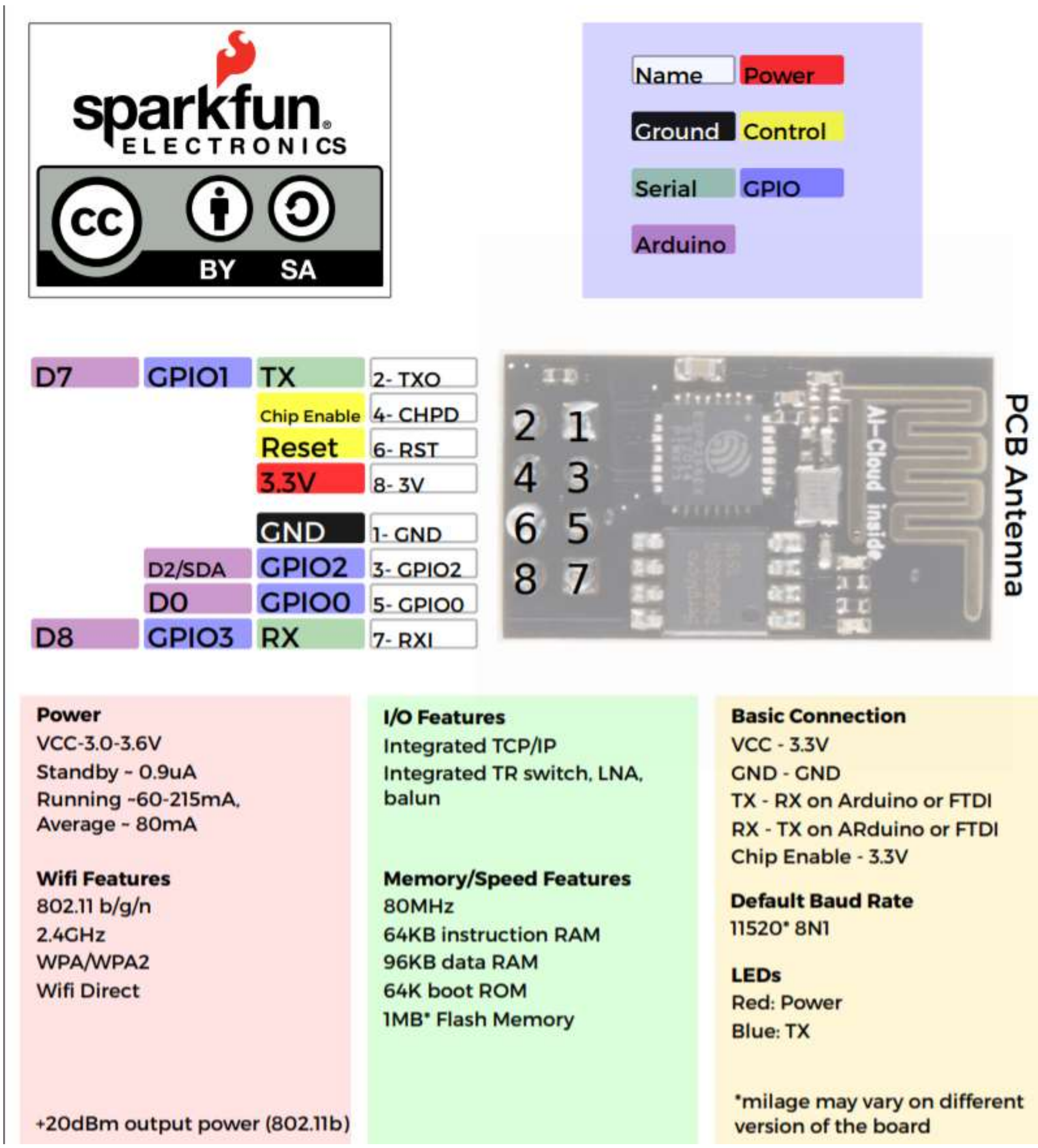
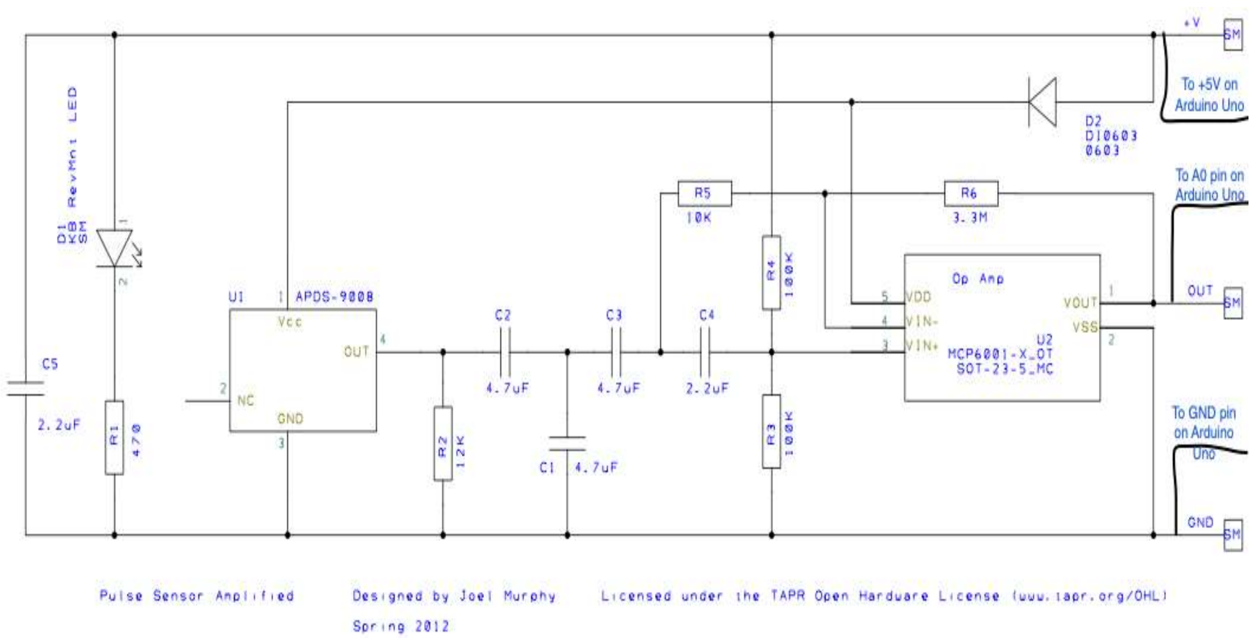


Figure 17: ESP8266 Pin Layout.

6.3. Heart Rate Module Subsystem

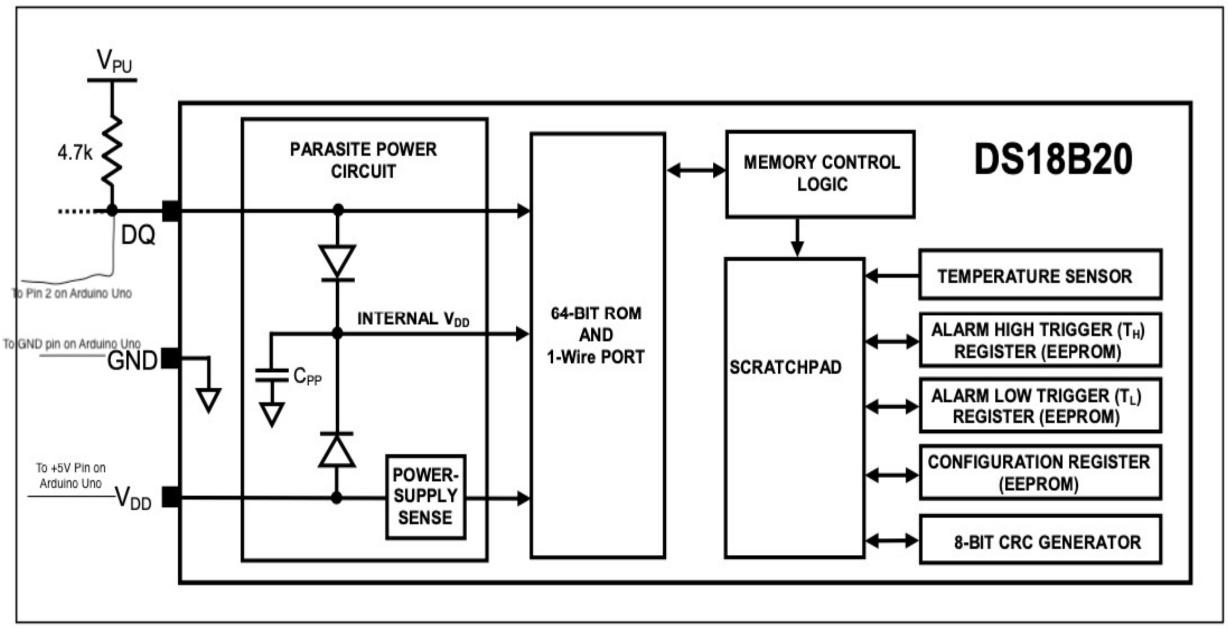
Heart rate data is extremely useful for measuring activity or anxiety levels. For that reason we found that making a heartbeat sensor a part of our product was a must. As a group after much deliberation we decided that the best choice by far for our development scenario was the SEN 11574 pulse sensor. Fortunately for us, the heart rate sensor we settled on is easy to integrate into our Arduino Uno microcontroller. The sensor has three wires connected to their own individual pins. One pin sends data from the pulse sensor and sends it to the Arduino Uno via any digital input pin. We connect this wire to A0 on the Arduino Uno. While this may seem incorrect at first because A0 is an analog pin, the A0 pin can also be configured to be a digital input pin. We then connect the ground pin to the GND bus on the Arduino Uno to ensure proper electrical functionality. Finally, we connect the +5V power pin to the +5V bus on the Arduino Uno to make sure that all of the electrical components have the right amount of power to function. This piece of the design will not be inside any casing and will be attached directly to the harness under where the dogs armpit would be. The sensor uses serial UART communication standard and must therefore be set to the right settings, like baud rate, in order to correctly transmit data.



6.4. Temperature Sensor Subsystem

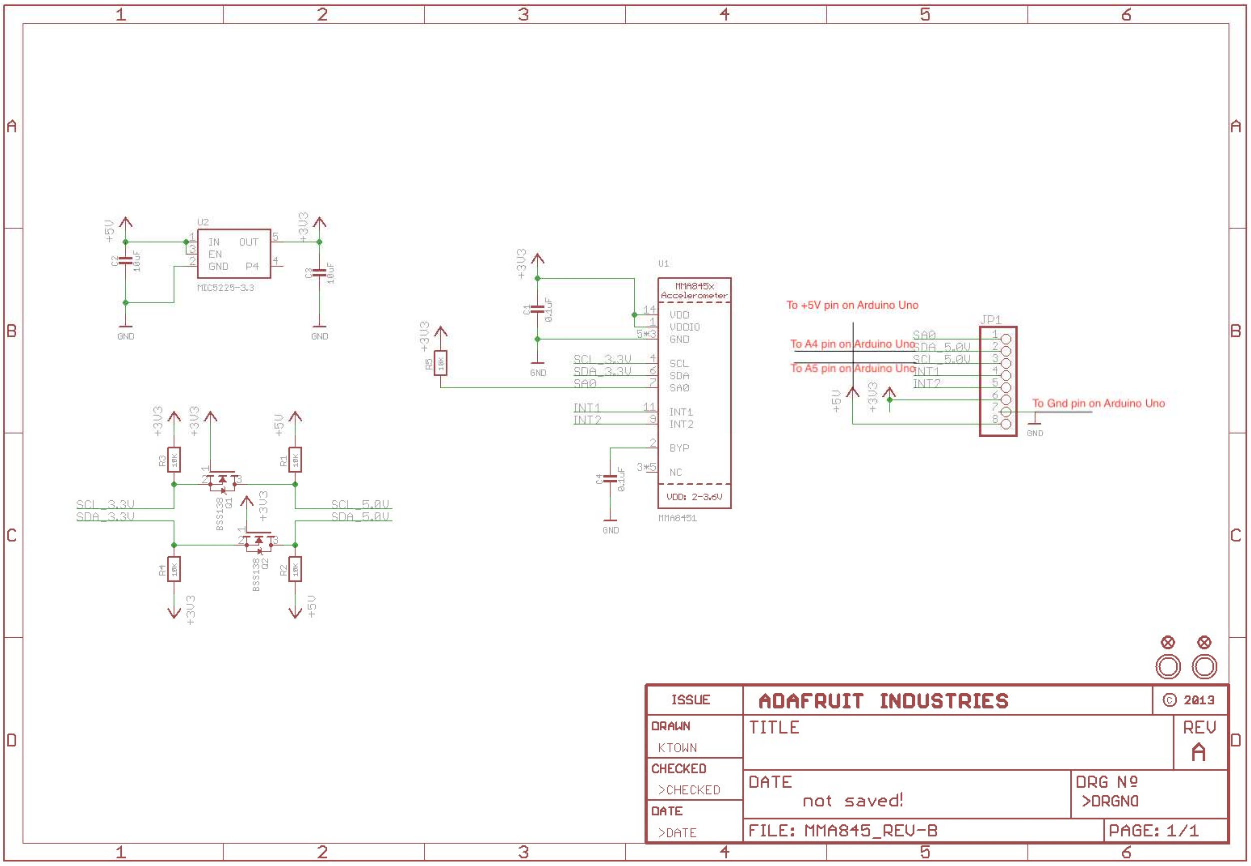
As a group the temperature sensing capabilities of our product were discussed as a top priority. We decided collectively that our best option was the DS18B20 digital thermometer from Maxim IC. This temperature sensor was most easily integrated with the Arduino Uno which we valued greatly. The digital thermometer comes with three pins, Vdd, DQ, and GND. We start by connecting the ground pin of the DS18B20 digital thermometer to the GND bus on the Arduino Uno microcontroller. Next, we connect the DQ line to any digital input, pin P2 in our case on the

Arduino Uno. This line transfers data from the thermometer to the microcontroller over serial UART communication. This means we must follow specific standards, like baud rate standards, to ensure that we transmit accurate data at the correct time. After connecting the data line, we connect the Vdd line to our +5V bus on the Arduino Uno microcontroller. Finally, we connect a 4.7k ohm pullup resistor between the Vdd and data line to ensure that the line is at high voltage when idle.



6.5. Accelerometer Module Subsystem

We decided as a group to order the MMA8451 Accelerometer. One of the main reasons for this group decision was that the module was extremely easy to connect to the Arduino Uno. The module uses the I2C communication standard above and therefore has a SCL, SDA, Vin, and GND pins. These pins must be accurately and correctly connected to the Arduino uno board or the whole design will collapse. Our wiring design is included in our final schematic in Figure #. By following the I2C standard and the schematic of the Accelerometer in Figure # and the schematic of the Arduino uno board, we found the correct pins and connected them. The accelerometer runs on 5V supply so the 5V output of the Arduino board was connected to the accelerometer Vin pin. This meant connecting I2C communication needs a SDA, or serial data pin, and SCL, or serial clock pin, so we connected the A4 pin which was labeled as SDA on the Arduino pinout to the SDA pin on the accelerometer. Next, we connect the A5 pin on the Arduino uno which is labeled as the SCL pin in the pinout to the SCL pin on the MMA8451 accelerometer. We also connect the ground pin on the accelerometer to the ground pin on the Arduino Uno. If we want to use the alternate I2C address for the MMA8451 accelerometer we will also tie the A pin to ground in the future.



6.6. GPS and GSM Subsystem

The GPS and GSM Subsystem is not utilized withing our final PetAid Harness project. Although the SIM808 GSM/GPS module is selected and purchased, there were time constraints during the implementation due to the COVID-19 pandemic. Research and development were still performed with an outcome of the following results. First the group had to understand the purpose and what functions can be provided by the GSM+GPS module. The purpose of the GPS+GSM is to send and receive GPRS data, this type of data consists of TCP/IP, HTTP, etc. communication protocols. The GPS+GSM module receives GPS data and A-GPS data which allows for monitoring and tracking of the pet at the command of the pet owner or user. Another main function of the GPS+GSM module is that it can send and receive SMS messages, as well as make and receive phone calls. In order to properly track activity and location of the pet wearing the smart harness we agreed that a module that had both GPS and GSM capabilities would be extremely helpful. By using the GPS combined with the accelerometer we can track activity and location easily, and by using the GSM we can send alerts and use the wireless phone network to send GPS location and data away from wifi. Being able to make and receive phone calls may not be the focus of the PetAid Harness project design, but the group had to ensure the PetAid Harness can send and receive SMS messages for alerts through notification messages. How the GSM works is shown in Figure #. The sim card in the device talks to a base transceiver station and that info is eventually received by the mobile service switching center which sends the data out. As a group we decided that the SIM 808 GPS/GPRS/GSM module was the best choice for the features we want to implement. This module is about the size of the board and needs its own 9V lithium-ion battery power supply to function. While the large size is a detriment to the portability of our design, we believe the features it brings greatly enhances the experience of the user of the product. In order to connect the module to the Arduino Uno hardware we need only 3 connections due to the module having its own power supply. The first connection is the basic ground connection to the GND bus on the Arduino Uno. The next connection is to pick a digital pin for the RX, or data receive, pin on the SIM808 module to connect to. For our case we picked Pin 11 as shown in Figure #. The final connection is for the module to send data to the Arduino Uno. The pin that sends data on the module is the TX, or transmit data pin, and we must pick another digital pin to connect to. For our case we picked Pin 10 as shown in Figure 17.

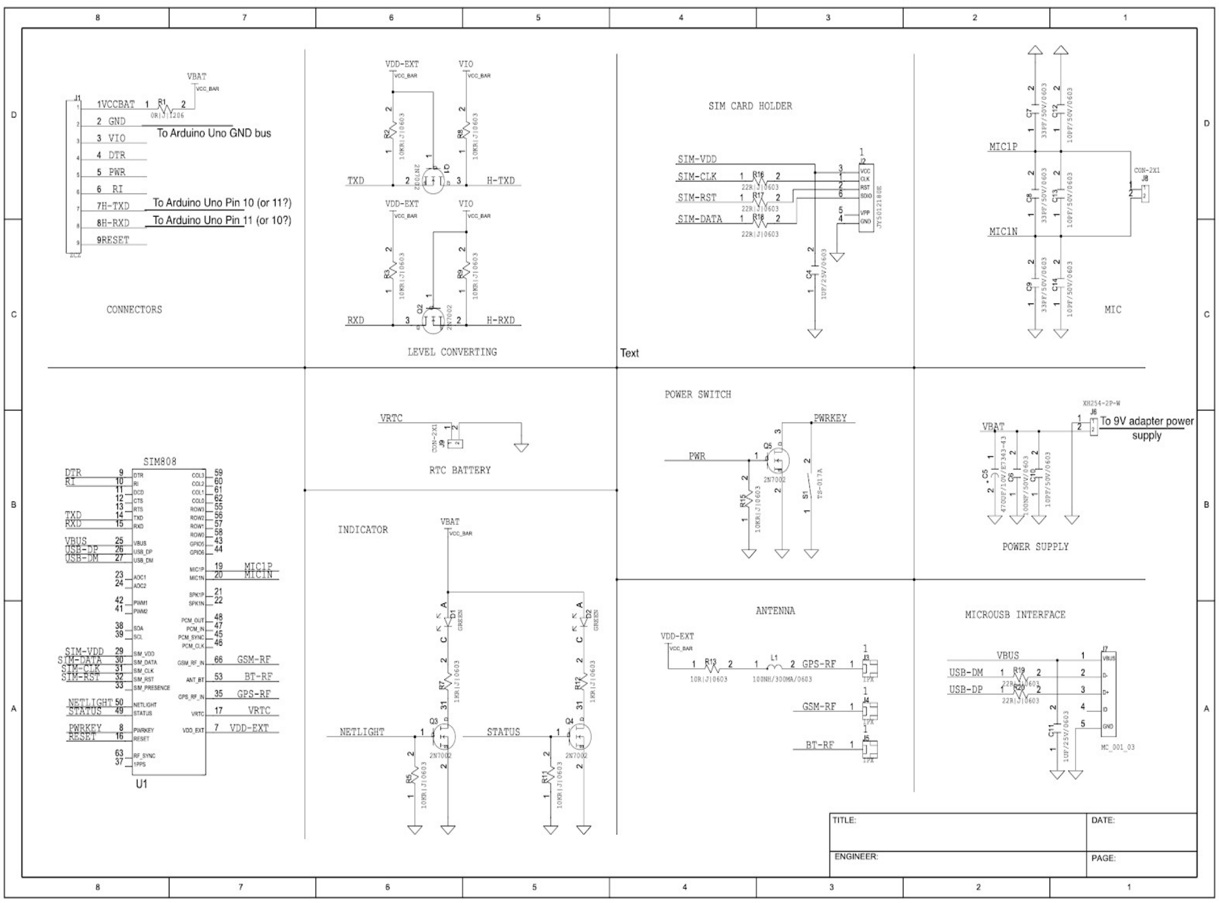


Figure 18: Schematic for SIM808 GSM+GPS module.

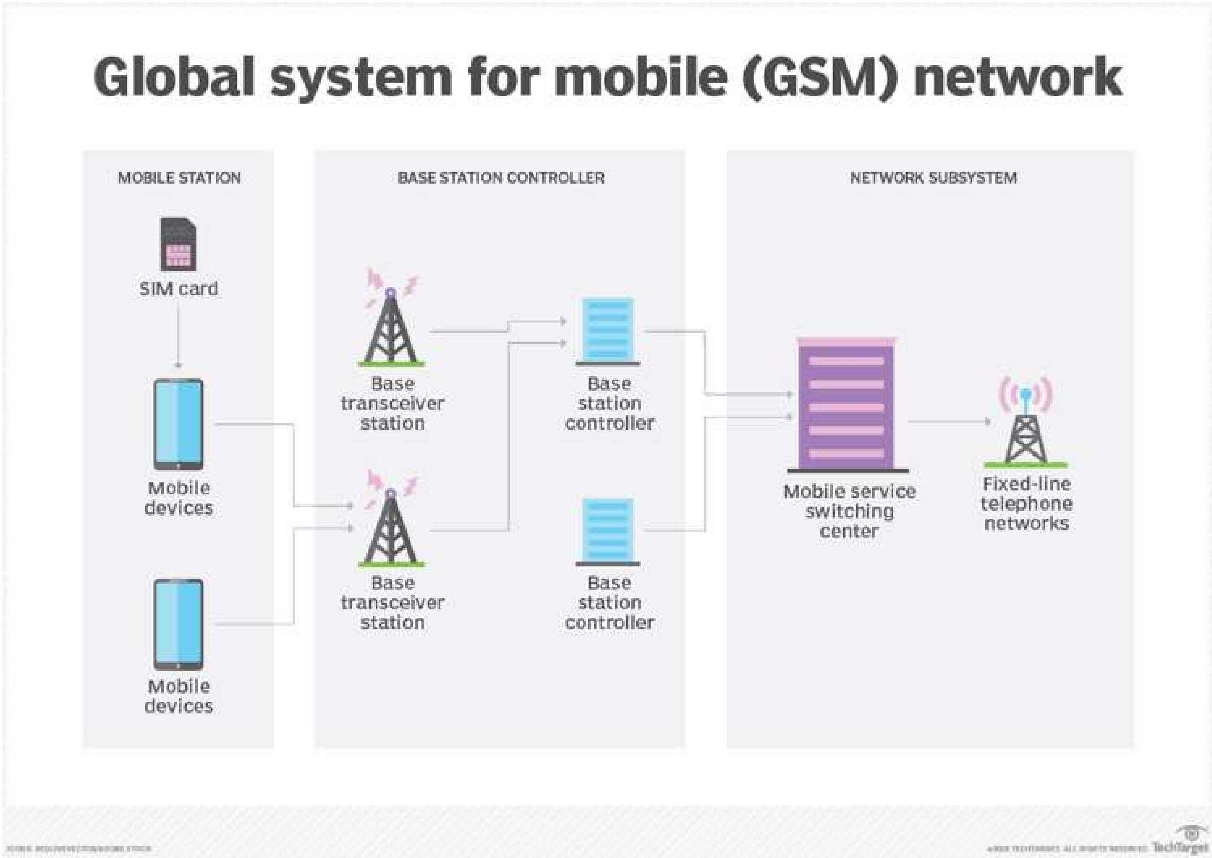
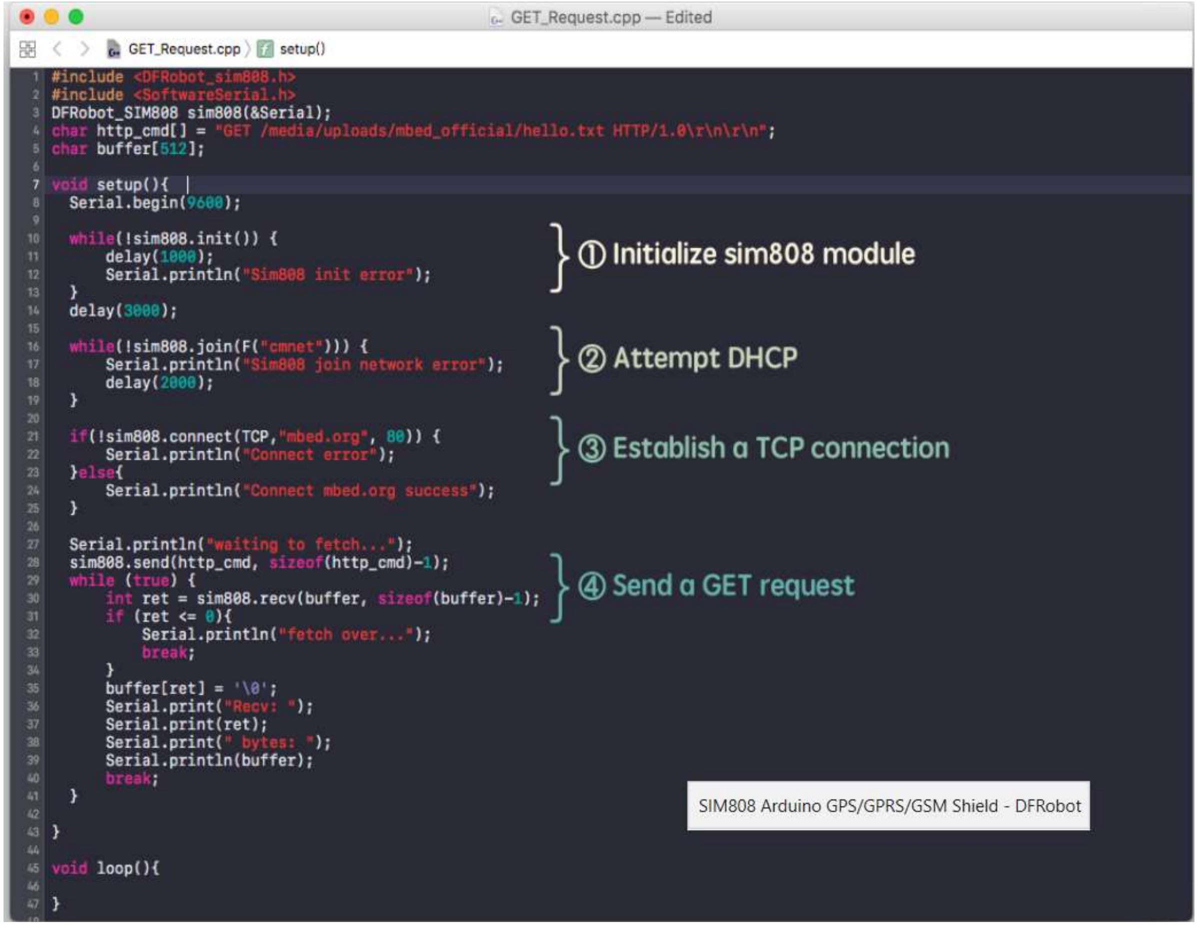


Figure 19: Global System Network System.

The SIM808 GPS/GPRS/GSM module has an Arduino UNO compatibility that helps use of the SIM808 Arduino shield as quick and easy during the implementation of the PetAid Harness. With intuitive functions packaged into a single library, the group can focus on the project and not waste time studying complex AT commands. There are four simple steps that can be explained on how a GET request can be successfully performed. First initialize SIM808 module, then attempt the DHCP network configuration to join the network. Third step involves establishing a TCP connection, and finally sending a GET request. Example of a GET request is shown in the Figure () below.



```
#include <DFRobot_sim808.h>
#include <SoftwareSerial.h>
DFRobot_SIM808 sim808(&Serial);
char http_cmd[] = "GET /media/uploads/mbed_official/hello.txt HTTP/1.0\r\n\r\n";
char buffer[512];

void setup(){
  Serial.begin(9600);

  while(!sim808.init()) {
      delay(1000);
      Serial.println("Sim808 init error");
  }
  delay(3000);

  while(!sim808.join(F("cmnet"))) {
      Serial.println("Sim808 join network error");
      delay(2000);
  }

  if(!sim808.connect(TCP,"mbed.org", 80)) {
      Serial.println("Connect error");
  }else{
      Serial.println("Connect mbed.org success");
  }

  Serial.println("waiting to fetch...");
  sim808.send(http_cmd, sizeof(http_cmd)-1);
  while (true) {
      int ret = sim808.recv(buffer, sizeof(buffer)-1);
      if (ret <= 0){
          Serial.println("fetch over...");
          break;
      }
      buffer[ret] = '\0';
      Serial.print("Recv: ");
      Serial.print(ret);
      Serial.print(" bytes: ");
      Serial.println(buffer);
      break;
  }
}

void loop(){

}
```


Figure 20: GSM Communication with GET request.

Availability of the SIM808 Arduino GPS/GPRS/GSM Shield from DFRobot resulted in being out of stock and not available during the integration and implementation process of the PetAid Harness. The SIM808 GSM+GPS module by iTead is purchased and utilized during the development stage, but resulted in difficulties and time contraints.

6.7. Power Supply Subsystem

There will be two separate components that provide power supply to the PetAid Harness. The first 9V battery will provide power to the Arduino Uno microcontroller and as well all the sensors except for the ESP8266 WiFi module. The second power supply provides power to the ESP8266 WiFi module. The decision to use two power sources is due to the ESP8266 WiFi module current consumption that is too large at 240mA and must have its own power supply.
Some common terminology that is used when talking about batteries are the following:

1. Capacity
2. Nominal Cell Voltage
3. Shape

4. Primary vs. Secondary
5. Energy Density
6. Internal Discharge Rate
7. Safety

Capacity- Batteries have different ratings for the amount of power a given battery can store. When a battery is fully charged, the capacity is the amount of power it contains. Batteries of the same type will often be rated by the amount of current they can output over time. For example, there are 1000mAh (milli-amp-hour) and 2000mAh batteries.

Nominal Cell Voltage - It is defined as the average voltage a cell outputs when charged. The nominal voltage of a battery depends on the chemical reaction behind it. A lead-acid car battery will output 12V. A lithium coin cell battery will output 3V. There is a keyword that is outlined in importance which is "nominal", the actual measured voltage on a battery will decrease as it discharges. An example of a demonstration of this decrease of voltage is a LiPo battery will produce about 4.23V, while when discharged its voltage may be closer to 2.7V.

Shape - Batteries can come in a variety of sizes and shapes. The common term is "AA" that references a specific shape and style of a cell.

Primary vs. Secondary - The corresponding definition for primary batteries is being disposable. Once fully-drained, primary cells can't be recharged in a safe or reliable manner. Secondary batteries are better recognized as rechargeable. These require another power source to fully charge back up, but they can fully charge/discharge many times over their life. In general, primary batteries have a lower discharge rate, which provides a long-lasting battery life, but they can be less economical than rechargeable batteries.

Energy Density - Combining capacity with shape and size of a battery, the energy density of a battery can be calculated. Different technologies allow different densities. For example, lithium batteries typically pack more juice into a given volume than alkaline or coin cell batteries.

Internal Discharge Rate - There are scenarios that can explain the process of how the internal discharge rate affects power supply from a battery. One possible situation is when trying to start a car that has been sitting for 6 months. Batteries discharge when sitting on the shelf or when unused. The rate at which the battery discharges itself over time is called internal discharge rate.

Safety - Because batteries store power, they are basically very tiny explosives. To prevent harm, batteries are designed to be as safe as possible. Most battery technologies are designed to discharge safely in the event of misuse. If you hook up an alkaline battery incorrectly, it may get hot to the touch but should not catch fire. Most Lithium Polymer batteries have safety circuits built-in to prevent damage to the battery and prevent it from unsafe usage.

The Table(31) shown below represents a few similarities and differences between some common primary batteries. These batteries are not rechargeable but can maintain an easy access source for the user or pet owner to purchase a new battery from any local grocery store or retailer. For this reason, a Li-Polymer battery is not taken into consideration during the breakdown of which battery type has the best specifications for the purpose of our PetAid Harness.

Battery Shape	Chemistry	Nominal Voltage	Rechargeable?
AA, AAA, C, and D	Alkaline or Zinc-carbon	1.5V	No
9V	Alkaline or Zinc-carbon	9V	No

Table 31: Common batteries, their chemistry and nominal voltage.

There are different methods that can be applied to regulate voltage so that the proper nominal voltage of 3.3V is applied to the voltage operation specifications of the sensors and also a higher voltage applied to the Arduino Uno Microcontroller for operation. A voltage regulator, otherwise known as a DC-DC converter provides a desired voltage level to have a boost or buck configuration. DC-DC converters are widely used to efficiently produce a regulated voltage from a source that may or may not be well controlled to a load that may or may not be constant. Four common topologies that makers might find useful are the buck, boost, buck-boost, and SEPIC converters. A buck converter steps a voltage down, producing a voltage lower than the input voltage. A buck converter could be used to charge a lithium ion battery to 4.2V, from a 5V USB source. A boost converter steps a voltage up, producing a voltage higher than the input voltage. This boost converter could be used to drive 9V output from two or three AA batteries, to supply towards the Arduino UNO microcontroller.

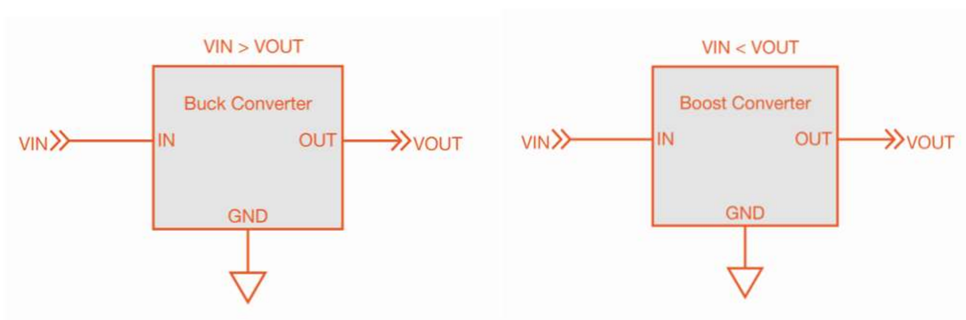


Figure 21: Buck Converter and Boost Converter.

There are some important performance characteristics that are considered when deciding what is the best possible DC-DC converter for the PetAid Harness. DC-DC converter datasheets have key parameters that describe their important characteristics to consider in designs.

Characteristics	Information
Efficiency	Fraction of input power that reaches the load. Some DC-DC converters are +90% efficient. It is good practice to ensure a DC-DC converter is 80% efficient.
Current rating	Maximum amount of current that a user should try to have the DC-DC converter provide to a load.
Temperature rating	If pushed past this safety limit, the DC-DC converter may overheat and be damaged or may shut down as a protective measure.
Ripple Voltage	Measure of how much ripple voltage is on the output. Be sure that the buck converter's voltage ripple rating meets your needs.
Regulation	Figure of merit related to how tightly the output is controlled over input voltage and load current. If a DC-DC converter has regulation rating of 1%, then the output voltage will not deviate more than 1% from the nominal value over the specified input voltage range and output current range.
Transient Response	DC-DC converters use closed feedback loops to provide a regulated output. Speed of control loop response gives an idea of how long it takes for the buck converter to respond to changing conditions.
Voltage rating	DC-DC converters have limits on how far up or how far down they can transform a voltage.
Size and Weight	DC-DC converters can be operated at very high frequencies, enabling them to be made small. Loss mechanisms increase with frequency, causing them to be less efficient, there is somewhat of a trade-off between size and efficiency.

Table 32: Characteristics of a DC-DC converter.

Another way to have a regulated voltage is presented as well. In order to increase the voltage between a battery's terminals, we can place the cells in series. Series means stacking the cells end-to-end, connecting the anode of one to the cathode of the next. By connecting the batteries in series, there is an increase in total voltage. Calculation of this voltage is done by adding the voltage of all cells to determine the operating voltage. The battery capacity for this series method maintains the capacity within the same amount. A battery holder with lead wires is the most reasonable task when utilizing this method since it will keep a 9V DC battery with an on/off switch instead of all two or three AA batteries needed for our PetAid Harness.

Figure 22: 9V DC barrel jack plug and battery pack.

6.8. Software Design

The way that the software will be created for the PetAid Harness is as follows: there will be the list of functions for the microcontroller that collect, analyze, and send the data to the phone, and there will be processes in the phone application that deduce the condition of the pet from the given data, the email verification and sign in process, the push and email notifications, and the data logging for future observation. The goal is to prioritize all the essential calculations on the microcontroller of the PetAid Harness, and all the record keeping and user-visible actions on the processor of the phone. That way the microcontroller minimizes the risk of bugs that could only be fixed by the developers, and the phone uses its own space for the documentation the user sees as needed for presentation to a veterinarian.

7. PetAid Harness Prototype Subsystem and Code

The initial prototype we will have for the PetAid harness will be based on the DIY form of the ThunderShirt, which we've seen to be a specially wrapped scarf. We will attach Velcro to the underside of the microcontroller to attach it for the purpose of the initial testing. All of the code will be tested with a series of edge case batches of data, and the harness's prototype will be tested by seeing how comfortable the scarf DIY ThunderShirt is on the pet that it's tested with.

Figure 23: The Humane Society of Navarro County DIY ThunderShirt Guide

7.1. PCB Schematic and BOM

Before we could combine our parts to make a prototype and test our design, we had to build a schematic and designate the correct connections for our Arduino Uno. After spending time viewing multiple schematic building programs online, we decided to go with the Scheme-it on program by Digikey because it has access to parts and part numbers directly through Digikey and allows us to export our final Bill of Materials (BOM) to Digikey to automatically make an order. It also allows us to build custom parts with custom pinouts and directly apply part properties and prices based on their Digikey part number.

Figure # shows the final Sheme-it schematic for our prototype PetAid Harness. There are 4 key types of lines; a high voltage line shown in red which indicates that the line is carrying a positive non-zero voltage, a clock line shown in blue which indicates that the line is carrying a clock signal from one device to another, a data line shown in purple which indicates data being transferred from one device to another whether it is digital or analog, and finally a ground line shown in black which indicates a 0 voltage reference.

To build our schematic we first built a custom part on Scheme-it for the Arduino Uno. We then attached lines to the 5V output and the ground of the Arduino Uno as we knew these would be important busses for all of the peripheral devices we are attaching. We then decided to add the peripheral parts one-by-one starting with the pulse sensor (SEN 1). After creating another custom part, we attached the power input of the sensor to the 5V output bus of the Arduino Uno. Then we attached the ground line to the ground bus. To transmit the pulse data we connected the signal line to pin A0. Next, we built a custom part for the accelerometer (ACC1) and added it to the design. We attached the power input of the accelerometer to the 5V output bus of the Arduino Uno as before and the ground pin to the common ground bus. The accelerometer uses Inter-integrated circuit (I2C) communication so two more pins require connections. The serial clock pin on the Arduino Uno is pin A5 so we connect this to the SCL pin on the accelerometer, and the serial data pin on the Arduino Uno is pin A4 so we connect this to the SDA pin on the accelerometer. After the accelerometer we made a custom part for the temperature sensor (SEN2). We yet again connect the power input for our temperature peripheral to the 5V output bus of the Arduino Uno and connect the ground pin. For data transfer we connected the DQ pin of the temperature sensor to the D2 digital input pin of the Arduino Uno. We need the data line at DQ to be high voltage when idle, so we also add a pull up resistor of 4.7K between the ground bus and the DQ data line. Next, we made a custom part for our GPS/GSM module (MOD2). We then connected the power input pin Vio to the 5V output bus of the Arduino Uno and the ground pin to the common ground bus of the Arduino Uno. The GSM/GPS module communicates using UART communication which means there are two data lines. The first data line TX goes into the RX pin of the Arduino Uno which is pin D5. The second data line RX goes into the TX pin of the Arduino Uno which is pin D4. We also attach the Reset pin to D6 so the Arduino Uno can control resetting of the module. To power the GSM/GPS module we also attach a Lithium Polymer battery to the battery pins of the module. We attach the V+ of the LiPo battery to the BAT+ pin on the module and the GND pin of the LiPo to the BAT- pin on the module. Finally, to receive and transmit GSM/GPS data we need two antennas. Therefore, we attach a GSM antenna to the GSM ANT port and a GPS antenna to the GPS ANT port. The last peripheral is our Bluetooth/wifi module (MOD 1), but it requires two parts to connect to the Arduino Uno. We decided to use the ESP-WROOM-32 module for Bluetooth and wifi, but it functions at a different digital logic voltage than the Arduino Uno. Therefore, in order for them to communicate we require a digital logic converter so that the Arduino Uno receives a sufficiently high voltage to register a high voltage level for its 5V logic. After creating custom parts for the Bluetooth/wifi module and the logic level converter, we begin by wiring the 5V output of the Arduino Uno to the high voltage side of the logic converter (HV) and the 3.3V output of the Bluetooth/wifi module to the low voltage side of the logic level converter (LV). Finally, we connect the TX and RX pins of the bluetooth/wifi module (GPIO 17 +16) to the low voltage side of the converter inputs and connect pins D0 and D1 of the Arduino

Uno to the high voltage side of the converter inputs so that one can serve as a digital input and one as a digital output.

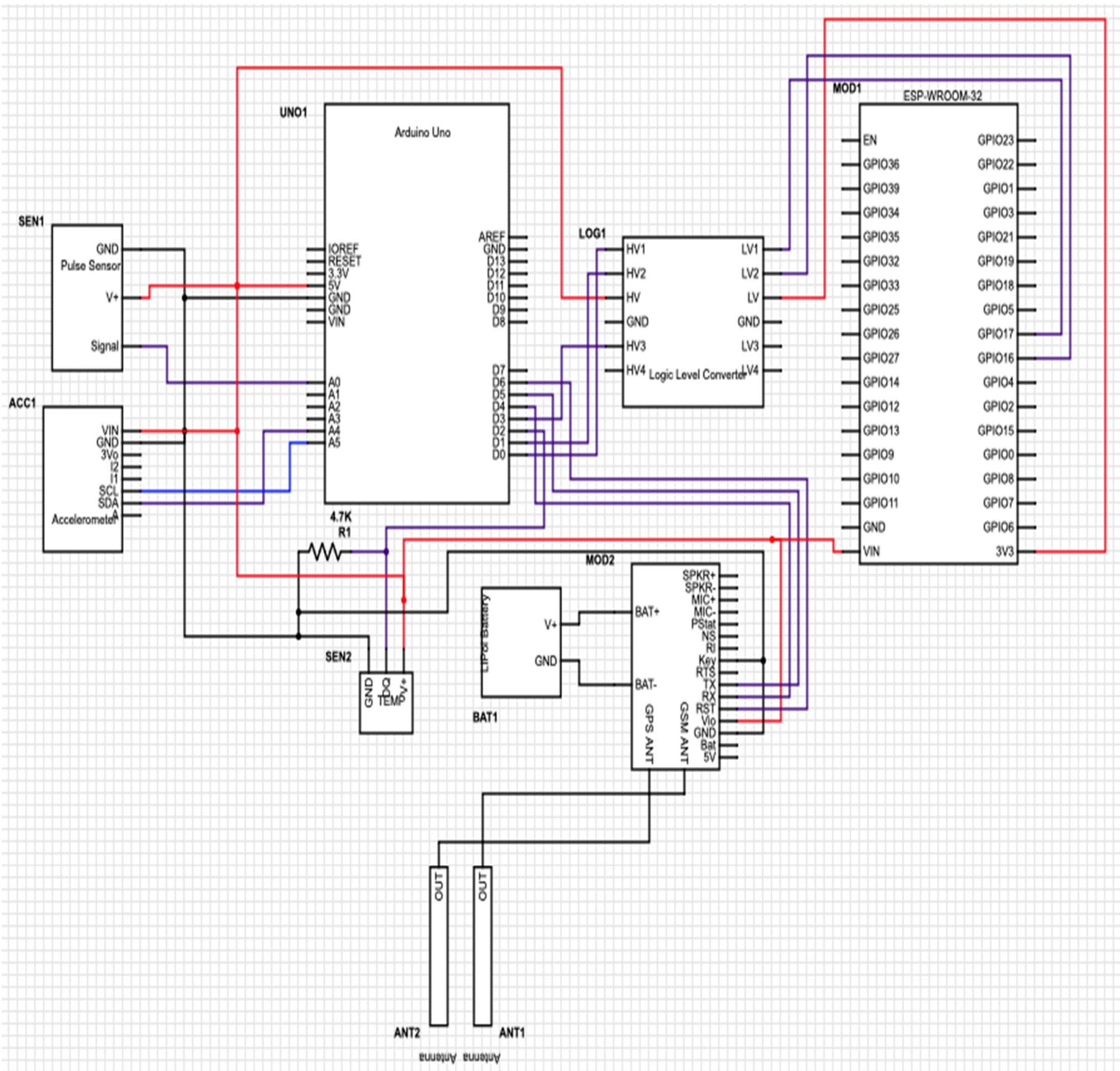


Figure 24: Initial Schematic made in Scheme-it by Digikey (Data Line, Clock Line, High Voltage Line, Ground Line)

Scheme it by Digikey was our preferred schematic program specifically because of how easily a BOM can be integrated into our design. Below in Figure # is a comprehensive list of all the electronic materials required to build the PetAid Harness. Our first part was the Arduino Uno listed as part number 1669-1000-ND, could be ordered for $22 and would ship immediately. Our second part was the SEN-11574 pulse sensor listed as part number 1568-1247-ND, costs $24.95, and would ship immediately. Our MMA8541 3-axis accelerometer and breakout board are listed under part number 1528-1041-ND, costs $7.95 and can ship immediately. For our temperature sensor we picked the DS18B20+ listed under part number DS18B20+-ND which costs $4.30 and can ship immediately. We decided to order 4 extra resistors in case of emergency and decided to go with 4.7K QBK-ND resistor listed under part number CFR-25JB-52-4K7 which costs a total of $.50 and can ship immediately. Our desired Bluetooth/Wi-Fi module is the ESP-WROOM-32 listed under part number 1965-1000-ND, and it costs $10 and can ship immediately. We also require a logic-level converter, so we went with the BOB-12009 listed under part number 1568-1209-ND which costs $2.95 and can ship immediately. We went with the 258 3.7V 1.2Ah Lithium-Ion Rechargeable battery listed under part number 1528-1838-ND, and it costs $9.95 and can ship immediately. Two antennas are needed for the GSM and GPS module. Antenna gain is considered since it describes how strong a signal an antenna can send out or receive in a specified direction. For our GSM antenna we went with the ANT-GSMMQB-UFL listed under part number ANT-GSMMQB-UFL-ND which costs $6.17, 18dB gain and can ship immediately. We also needed a GPS antenna and decided on the ECHO5/0.1M/UFL/S/S/15 under part number ECHO5/0.1M/UFL/S/S/15-ND which costs $14.48 and can ship immediately. Our last piece was the GPS/GSM module. We decided on the mini FONA 808 Cellular + GPS Breakout listed under part number 1528-1428-ND which costs $49.95. However, this part cannot ship immediately as it is on backorder, so we may need to substitute it with its bulkier counterpart. This alternate GPS/GSM module is the SIM808 GPS, GPRS/GSM RF Arduino Platform Evaluation Expansion Board listed under part number 1528-1707-ND which costs the same as the mini FONA 808 at $49.95 but can ship immediately. If this substitution is made the new GPS/GSM module would be wired the same as the initial GPS/GSM module we decided on. Table # contains an initial cost analysis.

RefDes	Name	Value	Manufacturer	Part Number	Digi-Key Part Number	Description
UNO1	Arduino Uno		Arduino	A000066	1659-1000-ND	ARDUINO UNO REV3
SEN1	SEN-11574		SparkFun Electronics	SEN-11574	1568-1247-ND	PULSE SENSOR FOR A...
ACC1	ACC Flip		Adafruit Industries LLC	2019	1528-1041-ND	MMA8451 3AXIS ACCEL ...
SEN2	DS18B20 Temp Sens		Maxim Integrated	DS18B20+	DS18B20+-ND	SENSOR DIGITAL -55C-1...
R1	RESISTOR	4.7K	Yageo	CFR-25JB-52-4K7	4.7KQBK-ND	RES 4.7K OHM 1/4W 5% ...
MOD1	ESP-WROOM-32		Espressif Systems	ESP32-DEVKITC-32D	1965-1000-ND	EVAL BOARD FOR ESP-...
LOG1	Logic Level Converter 2		SparkFun Electronics	BOB-12009	1568-1209-ND	LOGIC LEVEL CONVERT...
BAT1	Lithium Polymer Battery		Adafruit Industries LLC	258	1528-1838-ND	BATTERY LITHIUM 3.7V ...
ANT1	Antenna		RF Solutions	ANT-GSMMQB-UFL	ANT-GSMMQB-UFL-ND	RF ANT 850MHZ/900MH...
ANT2	Antenna		Siretta Ltd	ECHO5/0.1M/UFL/S/S/15	ECHO5/0.1M/UFL/S/S/15...	RF ANT 1.575GHZ CER ...
MOD2	GSM GPS Mini FONA 808		Adafruit Industries LLC	2542	1528-1428-ND	BREAKOUT BOARD GS...

Figure 25: Initial Bill of Materials (BOM) for the prototype PetAid Harness.

Schematic Name	Full Part Name	Required Quantity	Cost	Shipment Time
UNO1	Arduino Uno	1	$22	Immediate
SEN1	SEN-11574	1	$24.95	Immediate
ACC1	MMA8451 - Accelerometer, 3 Axis Sensor Evaluation Board	1	$7.95	Immediate
SEN2	DS18B20+	1	$4.30	Immediate
R1	CFR-25JB-52-4K7	5	$.50	Immediate
MOD1	ESP32-DEVKITC-32D	1	$10	Immediate
LOG1	BSS138 Logic-Level Translator	1	$2.95	Immediate
BAT1	3.7V Lithium-Ion Battery Rechargeable 1.2Ah	1	$9.95	Immediate
ANT1	ANT-GSMMQB-UFL	1	$6.17	Immediate
ANT2	ECHO5/0.1M/UFL/S/S/15	1	$14.48	Immediate

MOD2	FONA 800 GSM For Use With SIM808	1/0	$49.95	Backorder
MOD2	SIM808 GPS, GPRS/GSM RF Arduino Platform Evaluation Expansion Board	0/1	$49.95	Immediate
Total		15	$153.20	

Table 33: Initial Cost Analysis.

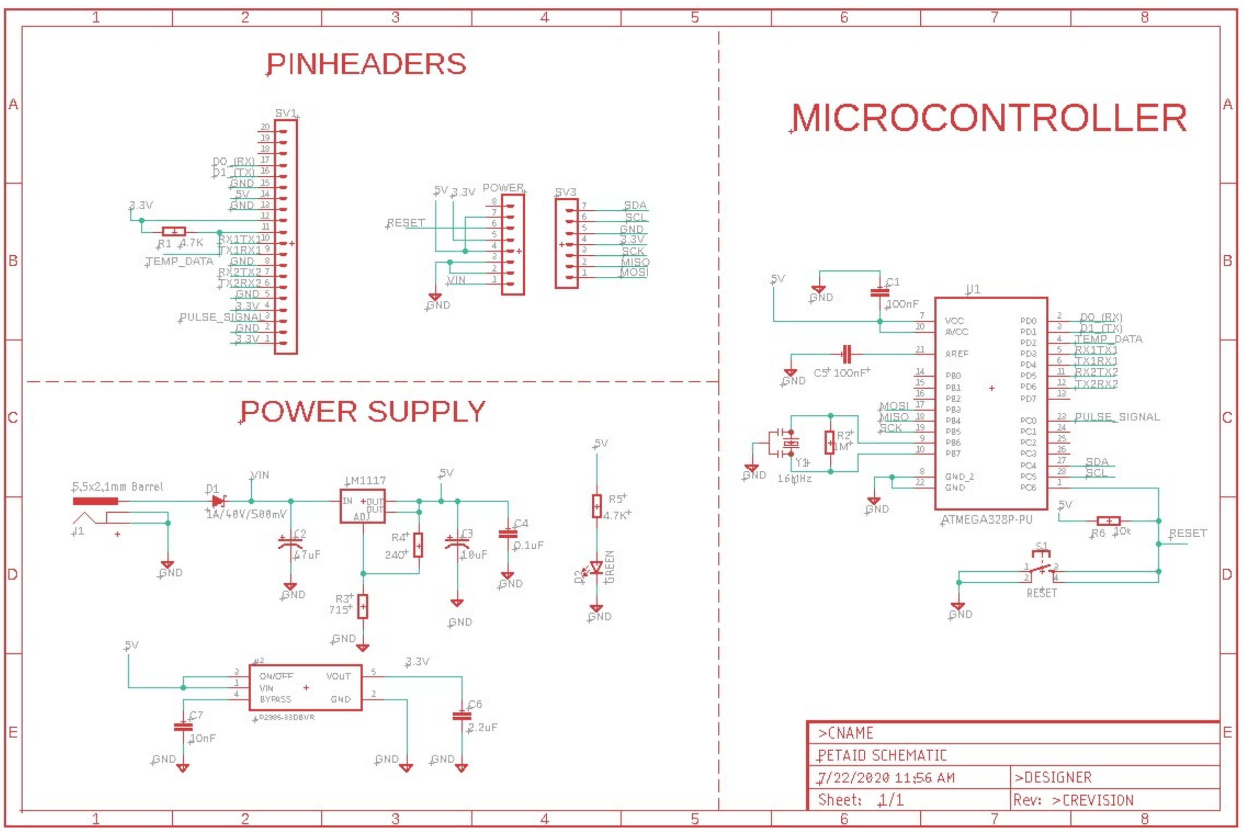


Figure 26: Final PCB Schematic.

The final PCB Schematic is displayed in Figure 26. There were different sections that were performed throughout our PCB design. Before the PCB is created a schematic is created on EagleCAD. This schematic includes components required for pin headers, power supply and microcontroller to fully operate for the overall design of the PetAid Harness. After this schematic is complete, then a board layout is created from this design and is explained further in the next section of PCB design.

7.2. PCB Design

The PCB design of the PetAid Harness will involve creating Printed Circuit Board (PCB). The PCB design brings our electronic circuits to life in the physical form. Using layout software, the PCB design process combines component placement and routing to define electrical connectivity on a manufactured circuit board. There is various PCB design software available for us to utilize for our project design, these can include Eagle Autodesk, Altium Designer, CircuitMaker, and EasyEDA. It was not an issue on what design software to use for the PetAid Harness PCB due to already knowing the tools and process of a PCB layout using one of these design softwares. Eagle Autodesk is available and introduced to us through the junior design course taken at the University. Being able to have familiarity with Eagle Autodesk as well as being free in an educational version to students provides an easy integration of our components onto the board. The purpose of this PCB design software is that it will allow to seamlessly connect schematic diagrams, component placement, PCB routing, and comprehensive library content. Once all the prototype parts are decided by the group, all the CAD symbol/footprint can be downloaded online and inserted into a schematic in Eagle Autodesk with the library manager. Next, after adding all parts required then the schematic is converted into a board layout where trace routing is problem-solved like working within a puzzle. Connecting all the traces will cover most of time duration throughout the PCB design, since engineering problem-solving skills are needed to overlap and produce no short circuits between multiple layers of the PCB board. An important discussion is that the PCB design is based on the microcontroller that our group decided to utilize for the PetAid Harness. The microcontroller is Arduino UNO, which will be the basis of our design. One of the goals given for this Senior Design capstone course is to successfully achieve our own made PCB board.

HARDWARE DETAIL

A. Printed Circuit Board

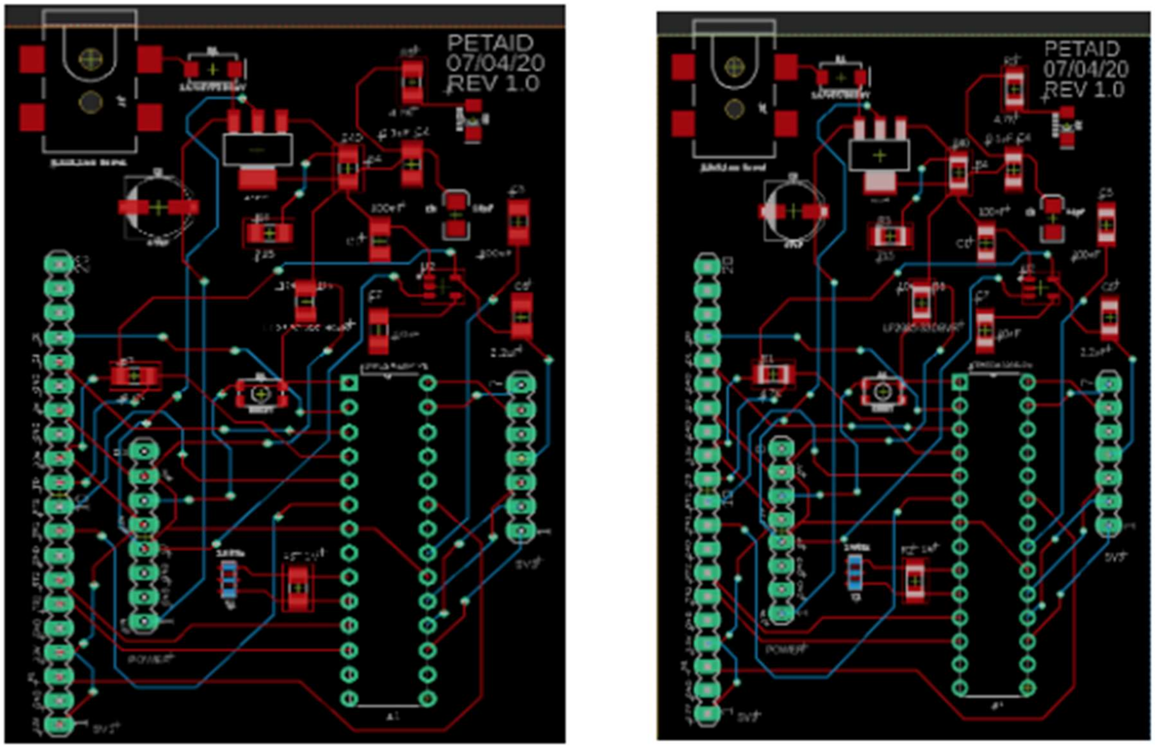


Figure 27. Printed circuit board top layer (left) and bottom layer (right).

The PCB was designed using EAGLE, which is one of many PCB CAD softwares out there on the internet. One 2-layer design was built for the PetAid Harness. The design is shown in Figure 4 and it features the ATmega328P-PU chip, LM1117, LP2985, MBRA140, 2 0.1 uF capacitor, 4.7 Kohm resistor, 47 uF capacitor, 10 uF capacitor, 2.2 uF capacitor, 10000 pF capacitor, 1 Mohm resistor, 715 ohm resistor, green LED, DC Barrel Power Jack, 16 MHz resonator, push button, and several pin headers for connections. This PCB board will be the most important part of the PetAid harness because it will connect everything and be the component to run the PetAid harness as it was designed for. Our group selected JLCPCB, which is a PCB manufacturer located in Hong Kong. Five boards were ordered and only two were assembled together and the other three were left available and open for us to solder. We soldered two boards, just to be safe because our team was limited on equipment to help solder the boards.

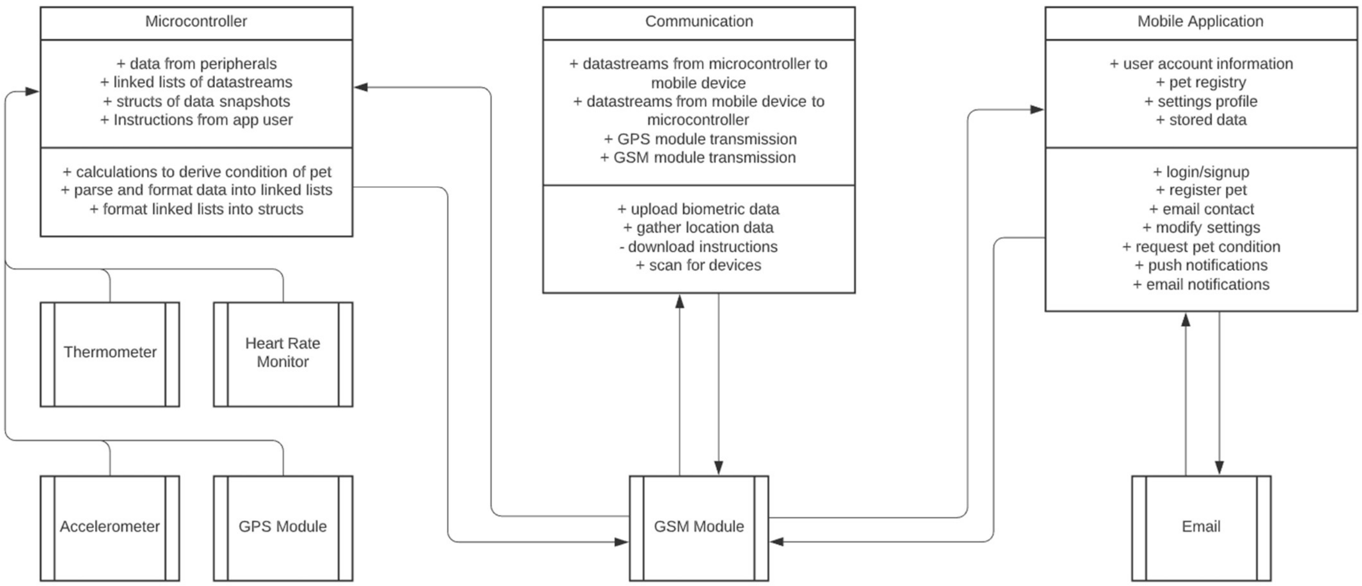


Figure 28: PetAid Harness Code Layout

The programming for the PetAid Harness has the design objective of collecting data from several peripheral devices measuring the body temperature, the heart rate, and the step count of the wearer of the harness, and perform analysis on those feeds of information into something a pet owner can benefit from in a mobile application display. When designing the layout for the code, we start in broad strokes, and get more specific as the code is fleshed out.

For us, the first distinction is hardware, software, the ThingSpeak server and the communication between them. There is a section of code dedicated to collecting the functions of the peripherals and running data analysis on the harness, there is a section dedicated to operations on ThingSpeak involving the GUI and operations in the application, and there is a section dedicated to communication between the phone and the harness, as well as code inside of the ThingSpeak site that graphs our output into a trackable format for further use by the owner. Within each of these sections there will be further distinction. For the hardware section, there are the sections dedicated to each of the peripherals; the thermometer, the HR monitor, and the accelerometer. Each of these focuses on parsing the data that is relevant to their respective subsystems and formatting them into something that can be analyzed by the board. For the thermometer, we use the DallasTemperature library which has a function for retrieving the current temperature for a given number of processing cycles that will approximate a more common measurement of time—seconds, minutes, hours—and a linked list variable dedicated to storing those integer values. A similar format will be set up for the heart rate monitoring; a cycle-based retrieval of a float value for the heart rate and a linked list variable to store the heart rate for analysis. With the accelerometer subsystem code, there is code to retrieve the current velocity of the harness, and a linked list of structs containing the speed and direction of the harness. This code

These codes handling the peripherals will be encompassed by a function that performs analysis on the values in the linked lists of each of the subclasses and will also be connected to the communication area of programming, as well as the display shown in the phone application. In addition to sending the raw linked list values for the peripherals for the purposes of logging the history of the pet's conditions and displaying the current conditions to the app user, but there will be the interactions between some of the variables in the linked lists, compiled into snapshots of the pet's current status. The relation of the heart rate and the body temperature of the pet cangive more insight into the status of the pet than only having those readings in the several combinations that each reading can have with another. For example, having a high body temperature with a low heart rate could mean that the pet is outdoors during a sunny day, and to have

The communication section of code will have functions that have the purpose of uploading packets of data to the mobile device and downloading instructions for which calculations to perform on the microcontroller. the code here will be simple considering that the microcontroller and the code for the mobile application will handle the formatting of the data into something usable by the recipient. As shown in the figure below, the code for communication will be responsible for the transmission of the biometric and location data from the microcontroller to the mobile device, and the transmission of the instructions

The mobile application's side of the code will involve the registry of an account for a PetAidharness owner, the desired settings for the harness that are set by the user, the email functionality that is used for notifications

8. PetAid Harness Prototype Testing

This section will now focus on testing the hardware and the software for the PetAid Harness prototype. We split the project testing into two sections, the hardware testing and the software testing. Our group decided to test each component individually before connecting everything together to have a smooth process. In addition. The software testing was done individually with each component we used to ensure that everything worked. That way we could identify an early problem with our components or code, and more easily discern errors in the future after those.

As for the harness itself, the testing would be no different than giving a pet a ThunderShirt to wear, which would suggest that it would be ethical to test for wear and tear by having a dog wear the harness without the microcontroller board or peripherals. Our intended purpose for the PetAid harness is to use with dogs over long periods of time, and dogs can vary in activity from being a lapdog to helping hunt, herd, or sport otherwise. Other similarly sized pets such as cats or certain pigs are less likely to be as active, so we want to use dogs to test the extremes of the harness's durability. We want to avoid making the vest uncomfortable for the pet, especially to the point of causing the pet to attempt to get out of the vest on their own. Ways to avoid this occurrence is by making the harness breathable for most levels of fur thickness and lightweight, so the pet can forget the harness is there.

8.1. Hardware Components Testing

The hardware testing will include the MCU, Wi-Fi + Bluetooth Module, Heart Rate Module, Temperature Sensor, Accelerometer Module, GPS and GSM module, Battery, and the Harness. Each test excluding the Harness will involve using the MCU board, A Breadboard, Jumping Wires, Resistors, Capacitors, Digital Multimeter and a power source. All the hardware components we choose for the project have online open source codes we can utilize to go with the testing we need. To obtain the best results each hardware component will be tested individually with the provided test code provided by either the manufacturer manual or online open source. After, each hardware component is tested and considered serviceable, then we will connect all the components together and test the final prototype with all the hardware components included.

8.1.1. Microcontroller Testing

The Microcontroller will undergo constant testing while the code is being written for the data analysis. This will be in the form of testing edge cases for the functions that we have, including gathering data in extreme temperatures, consistently getting an accurate heart rate, being able to reliably discern the velocity of the harness, and getting an accurate reading on the location of the harness. Should we decide we are able to add the haptic feedback functionality, we would need to ensure that it is a distinct and significant enough sensation for a pet to recognize. Aside from the testing of the functionality of the peripherals, we will test that the microcontroller is reliably supplying power to and taking information from the peripherals, and that there is no case where the microcontroller gets "bad" readings: inaccurate or inappropriate data formats, error messages, and unwanted interrupts. To test this, we will feed the microcontroller exaggerated amounts of information in order to define a point where the microcontroller can recognize that it is getting an incorrect reading on its own. For example, there is no possible way a dog is going to have a heart rate of 500 beats per minute, so if the microcontroller is picking that heart rate up from the monitor peripheral, there will be an issue with the microcontroller.

8.1.2. Wi-Fi + Bluetooth Module Testing

The Wi-Fi and Bluetooth module are both going to be tested to measure the range of connectivity between the microcontroller and a Wi-Fi signal and the mobile device the harness is synced to, respectively. Seeing as how there are different effects that the matter between the devices can have, we want to find a way to reduce the likeliness of interference reducing the product's effective range, or at the very least, gain the information of what environment affects the effective range for the product so it can be presented to the user of the application.

8.1.3. Heart Rate Module Testing

There are human and animal testing constraints that are involved when demonstrating a project design such as the PetAid Harness, the heart rate monitor testing will be different when compared to testing of other components involved for the PetAid Harness. Universities do not allow such type of testing, thus the testing of the heart rate monitor cannot be performed on humans or animals. The heart rate monitor requires a heartbeat to read and detect data. Since it requires a heartbeat, this limits the different methods that can be done to accomplish testing. The other sensors involved such as the GPS/GSM module, temperature sensor, accelerometer, and WiFi/Bluetooth module do not require data to be read necessarily from a human or an animal in order to test. Heart rate data can be really useful whether you're designing an exercise routine, studying your activity or anxiety levels. The issue when compared with a temperature sensor, is that heart rate can be difficult to measure. To solve this issue, that is why our group decided on using the SEN-11574 Pulse Sensor.

The Pulse Sensor Amped is a plug-and-play heart-rate sensor for Arduino. This helps simplify the usage for our project design and testing. It can easily incorporate live heart-rate data into the PetAid Harness using the MCU Arduino Uno. The testing process is provided by PulseSensor.com and is included into three parts. After ordering the pulse sensor, the pulse sensor kit contents involve a pulse sensor, color coded 24 inch cable with male header pins, matching ear clip, velcro finger strap, velcro dots, vinyl protective dots, and antistatic bag. These parts and tools are supplied for testing such as shown in the Figure() below.

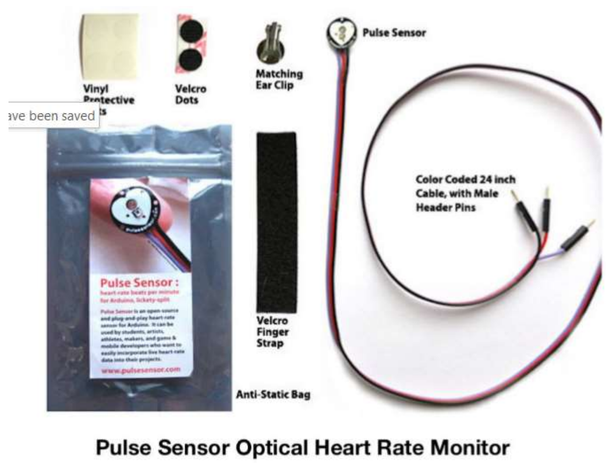


Figure 29: Heart rate pulse sensor kit for testing (courtesy of PulseSensor.com).

The front of the sensor is the heart logo, and this is the side that makes contact with the skin. On the front you see a small round hole, which is where the LED shines through from the back, there is also a little square just under the LED. The square is an ambient light sensor, exactly like the one used in cellphones, tablets, and laptops, to adjust the screen brightness in different light conditions. The initial setup before getting the pulse sensor up and running, is to protect the exposed circuitry so we can get a reliable heartbeat signal from any (naturally) sweaty/oily fingers by insulating the board. The Pulse Sensor is an exposed circuit board, and if the solder points are touched, the board could have a short, or introduce unwanted signal noise. Our group will use a thin film of vinyl to seal the sensor side. Located in the kit, there should be a small page of four clear round stickers, and one is peeled off to be centered on the Pulse sensor side with the heart shape logo. It should fit easily and once it is lined up, stick it onto the sensor all at once so that it will sort of stretch over the sensor and give it a nice close fit. This process takes care of the front side. The purpose for this preparation is so that the vinyl sticker offers a substantial amount of good protection for the underlying circuit, thus it can even be rated “water-resistant” so that it can be accidently splashed on with water and still be in operation.

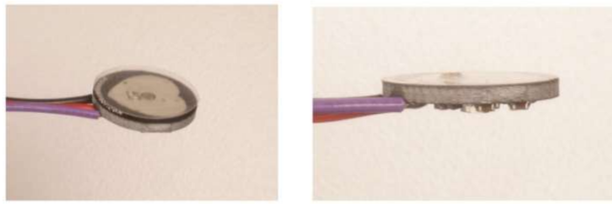


Figure 30: Thin film of vinyl stick onto front side. (Courtesy of PulseSensor.com)

The back of Pulse Sensor has even more exposed contacts than what is encountered in the front side, thus we need to make sure that it does not touch anything conductive or wet. There is a way of preventing this issue from happening is also done through protection. The easiest and quickest way for the testing that our group will perform is to protect the back side from undesirable and unacceptable shorts or noise is to simply stick a Velcro dot that is also included within the kit. The Velcro dot will keep our other sensor components for the PetAid Harness away from the Pulse Sensor parts just enough to get a good feel for the sensor and for our group to decide how the Pulse Sensor should be mounted for the best position located under the armpit of the pet dog. During our testing the Velcro dot comes off easily, so this is considered a temporary setup so that we can perform our testing of the Heart rate monitor sensor. It is noticed that the electrical connections are still exposed. After the testing results are obtained by our group for the PetAid Harness prototype and if the testing produces successful results, there is another way to better seal the Pulse Sensor by sealing the back side of the Pulse Sensor with a blob of hot glue over the circuit.

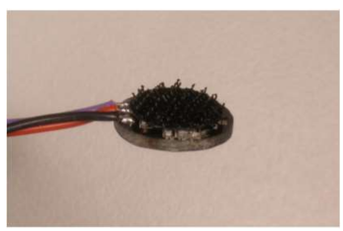
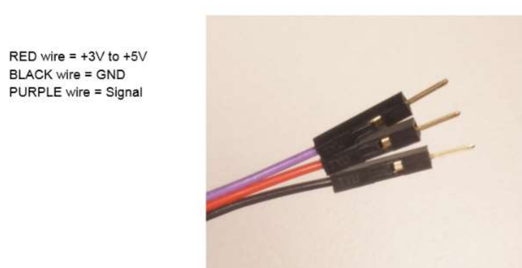


Figure 31: Velcro dot on back side of the Pulse Sensor. (Courtesy of Pulse Sensor.com)

In the kit a 24-inch color-coded cable, with 3 (male) header connectors should be found. This makes it easy to embed the sensor into our project and connect to an Arduino. No soldering is required. The Pulse Sensor can be connected to Arduino or plugged into a breadboard.



32: Male header pins to connect with Arduino.

As can be seen in the Figure() above, the connection points are displayed with the following: Red wire = +3V to +5V, Black wire = GROUND, Purple wire = Heartbeat signal. Before performing any connections from the SEN-11574 heartrate sensor to the Arduino Uno. It is strongly advised that we make sure to not connect the Pulse Sensor to our body while our computer or Arduino is being powered from the main AC line. Examples are charging laptops and DC power supplies. According to the data sheet of the SEN-11574, we should be aware of these precautions and isolate the sensor from the power grid by working under battery power. The following instructions are the connections that must be achieved for operation of the Pulse Sensor with the microcontroller selected for our project design, which is the Arduino UNO. Connect the Pulse Sensor to: +V (red), Ground (black), and Analog Pin “A0” (purple) on the Arduino UNO.

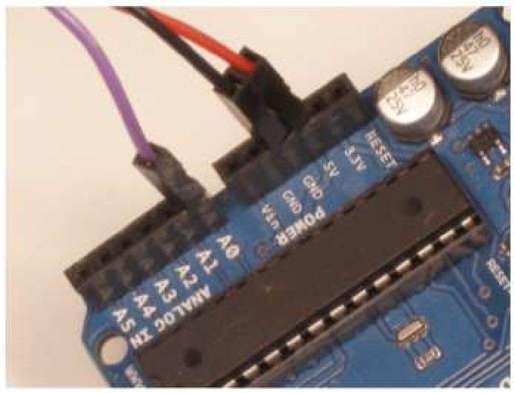


Figure 33: Arduino Uno +V, GND, A0 pin connectors.

The Arduino UNO microcontroller has GPIO pins located on board. These pins can include digital pins, analog pins, and power pin connectors for easy integration of the SEN-11574 pulse sensor. The last part for the testing of the heart rate sensor is running the pulse sensor code which is helpfully provided by pulsesensor.com. At this same location, the latest Arduino and Processing Pulse Sensor software can be found. The Arduino code is obtained and is called “PulseSensorAmped_Arduino-xx”, while the processing code is called “PulseSensorAmped_Processing-xx”. After the code is done uploading, the Arduino pin 13 blinks according to the time of our heartbeat when holding the sensor on our fingertip. During this testing process, the heart rate sensor should be handled carefully since in order to properly read an accurate reading from the sensor we must not grip the sensor too hard and if it is hold too lightly, noise interference and ambient light occur from movement. A correct reading on the Pulse Sensor will occur once a nice clean signal is displayed. This nice clean signal is discovered after an intermittent blink or no blink stop occurring. In order to view the heartbeat waveform and check heart rate, the Processing sketch previously uploaded allow us to view these results. Start up Processing on a computer and run the “PulseSensorAmped_Processing-xx” sketch. This is the visualization software that looks like the following Figure(34).

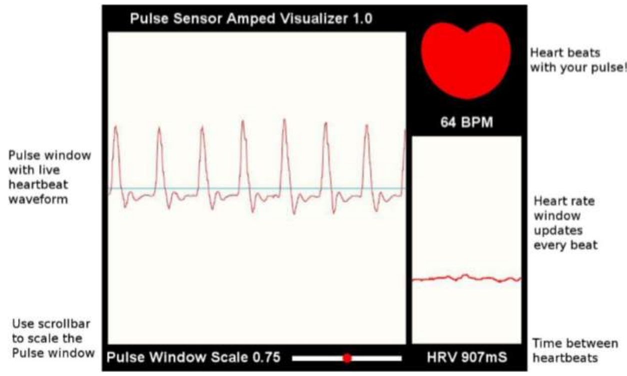


Figure 34: Graph of sensor data over time.

It is updated at every pulse with running average of the previous ten pulses. The big red heart in the upper right also pulses to the time of the dog's heartbeat. Testing involves using different parts of the body that has capillary tissue in order to ensure that when a pet wears the PetAid Harness with Heart rate monitor attached, a nice heartbeat waveform is seen as shown in the Figure(34). Along with this information, a schematic is created using EAGLE Autodesk CAD software in order to better represent how the wiring of the components and PCB are designed. A board layout is also shown in Figure(35) to represent the layers of the Arduino Uno microcontroller to be integrated with the SEN-11574 Pulse Sensor which is useful when determining a design rule check (DRC) for any errors during the PCB design.

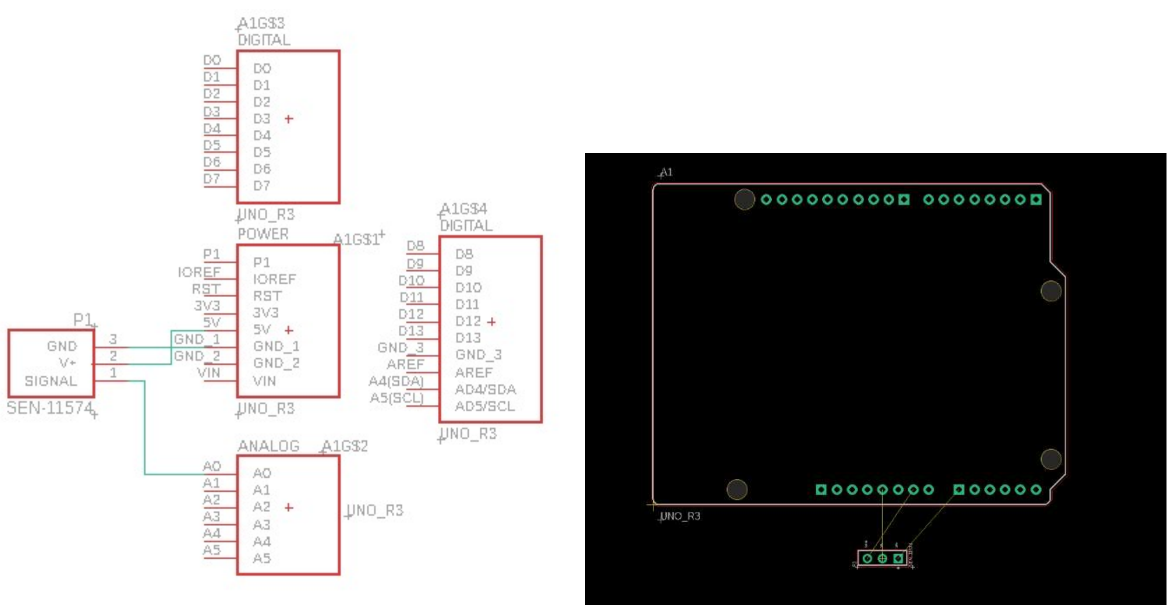


Figure 35: Heart Rate monitor schematic and board layout with Arduino Uno.

8.1.4. Temperature Sensor Testing

DS18B20 digital temperature sensor with Arduino will be the fourth hardware component tested. To test the digital temperature sensor, we need an Arduino board, DS18B20 sensor with a 4.7 KΩ resistor, a Breadboard, and jumper wires to connect everything. To use the digital temperature sensor with the Arduino board, our group used the Arduino Project Hub website that provided program files that made all the calculations we needed for our temperature testing. Figure () displays a simple circuit network we will use to test the digital temperature sensors. Because we are not allowed to measure the temperature of a dog, instead we will measure the temperature of the air, liquids, and ground to meet the specifications for the project. The main characteristics of concern for the temperature sensor are the precision, range, and consistency. To obtain the best results, each test will be repeated five times to ensure that the temperature sensor works well and meets the needs for the PetAid Harness.

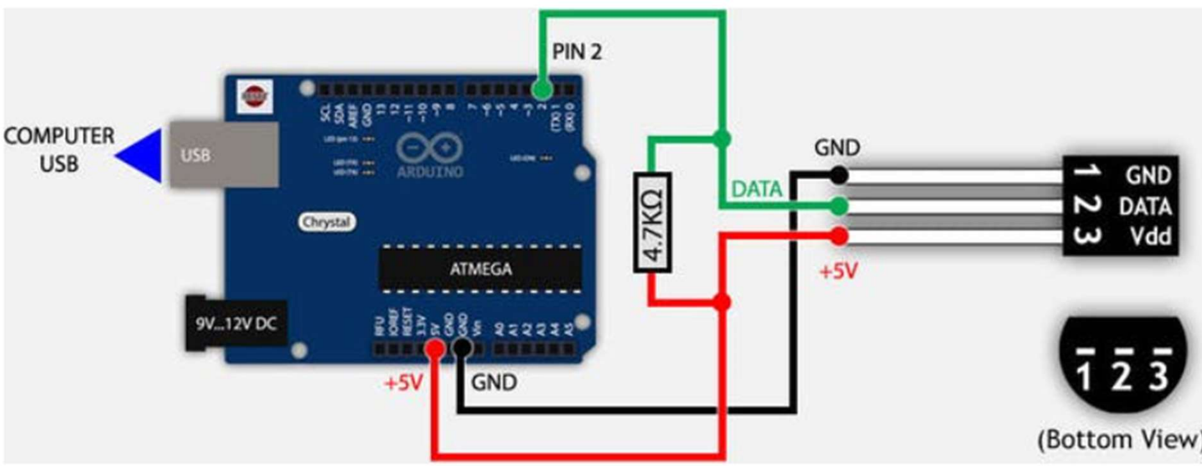


Figure 36. Circuit for the Temperature sensor testing (Courtesy of Arduino.com)

In order to test the precision of the DS18B20 digital temperature sensor, our group will perform different type of temperature measurements for different objects. We will record measurements for Ice Cubes, Boiling Water, different solid objects at room temperature, and clothing that has been left outside for 30 minutes. We will record and observe five different temperature measurements for each object to make comparisons and make the final calibrations if required.

To test the range of the DS18B20 digital temperature sensor, we will use objects with known temperature measurements and record different temperature measurements at different distances. Because the PetAid Harness is designed to be attached to the dog, we will only use short distances such as 1 centimeter, 2 centimeter, and 3 centimeters. For the best results we will record 5 different temperature measurements at 3 different distances. The recorded results will then be compared to the known temperatures for each object to verify that the recorded measurements are precise at each distance tested.

To ensure the consistency of the DS18B20 digital temperature sensor, we will analyze the recorded measurements for precision and range. If the precision measurements and the range measurements meet the expected temperature readings, then the consistency of the DS18B20 will meet the satisfactory characteristic.

8.1.5. Accelerometer Module Testing

In order to produce a practical product that brings peace of mind to our users we must ensure that our accelerometer peripheral produces accurate measurements. This means individually testing each axis of motion for the accelerometer, including other things. The first step of our testing process involves soldering the breakout pins to the board and onto the breadboard. Our first test is the y-axis of the accelerometer. To accomplish this, we attach the accelerometer to a two-foot cord and drop the module multiple times. We then compare this to the nominal gravitational acceleration to see if the device is operating properly. For our x-axis test we built a small ramp with a predetermined incline and attached the accelerometer to a small Hot Wheels car. We then will allow the car to roll down the ramp. After reading the data given, we then compare the acceleration to our calculated gravitational acceleration at our given incline. To test the y-axis, we do the same process as stated above but rotate the accelerometer ninety degrees before re attaching it to the small Hot Wheels car.

Our testing ended up largely being with ourselves; holding the accelerometer and moving around. Once we decided to add the step counter feature, we walked around or otherwise simulated steps while holding the accelerometer.

8.1.6. GPS and GSM Module Testing

Testing the reliability of the GPS and GSM modules was be done by monitoring the location of the harness and finding the maximum range of reception between the mobile device and the harness, respectively. Obstructions, such as concrete, metal girders, and trees, may all affect the maximum range of the GSM module, and was be checked to make sure the harness has as little a hindrance as possible to its range in environments like the inside of a house or the middle of a heavily wooded area. The GPS module's connectivity with its satellites and its communication with the microcontroller were the areas that we focused on in particular. While we were able to get some success in sending notifications with the GSM module, due to time constraints, and focusing on getting the most essential function of the PetAid harness completed first, we decided to postpone the testing of this area indefinitely.

8.1.7. Body Harness Stress Testing

Peace of mind is an essential goal for our product. We want to guarantee the safety of the pet attached to the device and the reliability of the equipment attached to the device at the same time. To do this we must ensure that the harness is up to our specifications with durability. To test our body harness, we decided to buy a second harness to perform testing on.

The first test to be performed will involve stretching. We will tie one end of the harness to a loop fastened to the wall on one end, and weights hooked on to the other end slowly one by one. We will carefully document how much each added weght weighs and how much the harness stretches per weight. We will continue to push the harness until the harness fails to hold together, or we run out of weights. We will then measure the total stretching documented in centimeters. We will then calculate the total force applied on the harness using the gravitational constant and that will be our standard of force required by the pet to stretch or snap the harness.

Our second test will be a tearing test. This test will serve purely as anecdotal input for the group to understand the basic limits of the harness and will produce no numerical data. The team will gather together with our test harness and bring various tools for cutting, such as a saw. We will test all of these different tools with varying amounts of force to see what the maximum endurance of the harness is.

Finally, we will test the harnesses ability to reliably hold our electronic material. After configuring our test harness with our prototype electronics, we will walk while shaking the harness much like a pet would. We will then log any loose connections or weaknesses in the electronic sturdiness that may be an issue. Afterwards the group will get together to discuss if we believe it is up to our specifications.

8.2. Software Testing

Testing the software for the PetAid Harness involves testing the accuracy and precision of the code that is dedicated to collecting and analyzing the data from the peripherals, testing the reliability of the communication between the microcontroller and the mobile device, and the fluidity and consistency of the ThingSpeak server when relaying data from the board.

In order to test the software dedicated to the microcontroller, gave the peripherals give a normal reading by using a small bowl of ice water, and the warmth of our hands. We were able to get the proper readings consistently regardless of the extremes the temperature, heart rate, or velocity, and the microcontroller was able to perform analysis on these and return to normal after the extremes are removed, so we deemed the reliability of the microcontroller a success. The communication area of the software involves the reliable transmission of packets between the ESP8266 of the PetAid harness and ThingSpeak's channel. Testing around this took the form of giving all our calibrating tests for every other aspect of the code at different distances between the PetAid harness and the ThingSpeak channel it's connected to. We were able to get consistent communication with the ThingSpeak server and send data every 15 seconds with the free version of the channel, so we deemed it a success. Should there be any errors that come up during or after the initial testing, we will change the status of that software in that area accordingly, and it will no longer be considered reliable, or bug-free.

9. Project Plan

Administrative content is important when creating a major project design such as the PetAid Harness. The PetAid Harness involves working with others on a long-term project, so obtaining a schedule is a very useful and oftentimes necessary responsibility that can result in great benefits. The senior design course does not only involve engineering work but is also about learning how to manage a project on a larger scale than any other previously tackled projects. Agreement between all group members on organizing schedules will keep us on track and organized. There are two discussions that are covered within this report to outline the project plan and project scope, these two discussions contain milestones and budget/finance. Successful completion of the PetAid Harness can be essentially achieved with an established set of deadlines assigned for the group team members when designing a project of this magnitude. This allows for each team member to know what his/her tasks are, and when they are expected to be done. These deadlines are converted into a schedule that is usually allocated on a weekly basis to remain within the project scope and not be overwhelmed by the countless information due to lack of preparation and procrastination. Team members can adjust their own personal schedule to ensure enough time and resources can be accommodated to make sure they can accomplish the tasks given to them. Overall extent length of the project design is designated to be incorporated over the last two semesters as capstone requirements for senior level undergraduate students.

Division of work responsibility is employed alongside the schedule plan as previously discussed. Deploying each member with their responsibility based on parts that must be attached on the PetAid Harness, such as the sensors, mobile application, and microcontroller provides with work being divided between each group member for the project design. As listed in Figure 1 and Figure 2 within this report, hardware and software block diagrams are also created as another method of ensuring tasks are completed successfully. A designated "leader" is assigned for each block that can include research and testing throughout the development process of the PetAid Harness. For example, the microcontroller is an important aspect for handling the inputs and outputs of the various sensors and modules. Different microcontrollers are available, and the assigned "leader" has the role of listing and comparing important specifications, however all group members have a role of assisting in information and deciding what microcontroller is going to be utilized for the PetAid Harness.

9.1. Milestone Discussion

Milestones are discussed as administrative content in order to mark a significant change or stage in development of the PetAid Harness. The project milestones further describe the importance of project planning with an integration of setting a schedule. These milestones relate to completion dates or weeks of completion throughout both the Senior Design 1 and Senior Design 2 semesters. Milestones have a timeline that are described from initially at the start of the design process until the completion of the project. Senior Design 1 is the initial process duration of the PetAid Harness consisting of research and developing stages of the design process. If appropriate milestones are set, the acquiring for most of the main project parts within the first semester will allow testing in the following semester or sometime before. Time allocated beforehand for testing purposes, allows for easier test preparation since it is anticipated that there will be issues or design failures initially. Utilizing the first semester for testing should permit enough time for the team to reflect on a redesign due to unpredicted factors and concerns of design obstacles or issues. The entire team has to work together during the prototype testing of both the overall PetAid Harness and individual testing of the ordered components to build the PetAid Harness. There is also a time constraint that was established during the divide & conquer document submitted to the professor when forming our project design decision. The PetAid Harness testing to ensure it operates correctly when worn on a dog is a challenge. Animal and Human testing are not allowed per the University requirements; thus, another course of action needs to be brainstormed by the team to work around it. This time constraint is considered within the milestones during prototype testing and demos.

Senior Design II will occur during the summer of 2020, which will gather all information obtained in Senior Design I and utilize it to test and build a prototype design. This prototype design must get rid of errors through demos and then build a final design that is intended to be presented at the end of the semester. Final development stages should be established once the final project is assembled and tested. A plan is set to perform individual testing of the sensors and module components that are integrated for the design of the PetAid Harness. Being able to test all the components individually to ensure they read the correct measurements within a basic environment without all components connected into one system, allows for simplicity when troubleshooting and therefore reduces a delay of time for issues. Once all the components are tested individually, then the team transitions into implementing all components into one single system for the PetAid Harness. The project idea decision, divide and conquer document approval, research, and project building are a couple of examples that are further discussed in the milestone tables listed below.

Senior Design 1

Milestone	Completion Week
Project Idea Decided	1/22/2020
Divide and Conquer Rough Draft	1/31/2020
Divide and Conquer Approved	2/5/2020
Research- Components	3/12/2020
Research- Operating Systems	3/19/2020
60 page Senior Design I Draft	3/20/2020
Hardware/Software Agreement	4/3/2020
100 page Senior Design I Draft	4/3/2020

PCB software tutorial complete	4/18/2020
PCB Simulated	4/20/2020
PCB Constructed and Tested and Final Report Due	4/21/2020

Senior Design 2

Milestone	Completion Week
Project Building- Casing/Non-sensor Hardware	5/11/2020
Project Building- Software for sensors completed	5/25/2020
Project Building- Motor system and sensors	6/8/2020
Project Building- All hardware combined	6/15/2020
Project Building- Software interface completed	6/22/2020
Project Building- Initial testing w/ adjustment	6/29/2020
Project Testing- Fine-Tune testing	7/6/2020
Project Testing- Prototype test completed	7/10/2020
Project Testing- Finished Product	7/17/2020
Project Presentation	TBD

Stretch Goals Provided by Dr. Richie

- Create our own phone application
- Upgraded physical design appearance: Final design should have surface mount parts such as a surface mount Bluetooth module to reduce all the wiring done for prototype testing, Infrared red Thermometer to be able to test on a stuffed puppet.
- Make project case as light as possible
- Use GSM module instead of GPS module itself, so that phone can communicate data both close and far away at different location by using sim card SMS services.
- Reference products: fit paws, “smart harness for dogs”.

9.2. Budget/Finance Discussion

The total cost of the project must be observed or predicted beforehand, to ensure it meets the maximum budget requirements initially decided as being affordable and reasonable by the whole group. As one of our goals of the project design, the PetAid Harness had to be kept at a budget as low as possible. Cost of the PetAid Harness is an obstacle for our group, since we are college students with no full-time job. This consideration is considered for UCF students which will be a determining factor for overall quality design of the PetAid Harness. Superior quality materials and highest performing component parts should not be decided, since the investment is to be within capstone requirements along with experiencing a short project deadline. A harness is the appropriate fitting and desired classification to be worn by the pet, therefore cost is a subject to be covered. The choice of selecting to utilize a harness instead of a collar was determined to be not much of a difference in terms of increasing budget cost. Decision of the budget scope for the PetAid Harness is discussed by all group team members within a group meeting. The discussion involved how important and demanding the PetAid Harness is desired by the pet animal lovers. The conclusion was resolved as integrating the PetAid Harness with good quality and multiple features that could be both low cost and within reach of the pet owners who love their pets so much as to invest in a smart harness to keep their dog safe and in healthy condition on a daily basis, along with maintain a low cost for the average pet owner who may not want to spend a substantial amount of money that could not be necessary. Components that are described as the prototype items to be used for the initial prototype testing process, will be used as part of the cost of the project design. These components that are accounted for are the various sensors, microcontroller, modules, and battery power supply. Multiple features are obtained from these components that includes heart rate monitoring, position movement, temperature sensing, and long battery life replacement. Features that are included for the final product design are GPS tracking + GSM module, and mobile app for long distance display of all this information onto the user's mobile phone. Once the components are selected based on their best specification, different vendors who supply the parts needed are researched to realize the cheapest and low-cost option for the budget desired.

Since the finances for the PetAid harness will be split between the members of the team, the decided set budget that is both an efficient cost between the team and many pet lovers throughout the U.S is determined to be around $100. Of course, there is a budget of $100 that was set during the initial process of the PetAid Harness while discovering and as well brainstorming throughout weekly meetings. The initial total cost needed to build and design the project, excluding the 3D-printed case, as shown below is $113.56. This cost will of course be subject to change since additional components might be needed. The additional components that are involved after the prototype is complete resulted in a much higher cost for the group. This cost is $454.55 which is including the purchase every possible component, equipment, accessories, and tools needed to complete the initial prototype of the PetAid Harness at a timely and error-free manner. Prototype testing should provide results that can be observed and further improve the PetAid harness design.

Component	Prototype Items	Vendor	Cost

Thermometer	DS18B20+	Digi-Key	$4.30
Heart rate monitor	SEN-11574	Digi-Key	$24.95
GPS+GSM	SIM808	DFRobot	$39.99
Accelerometer	MMA8451	Digi-Key	$7.95
Wi-Fi/Bluetooth	ESP32-Devkit-32D	Digi-Key	$10.00
Battery	Lithium Ion Polymer Battery 3.7V 1200mAh	Digi-Key	$9.95
Microcontroller	Arduino Uno	Digi-Key	$22.00
Harness	Rabbitgoo Dog Harness Adjustable Outdoor	Amazon	$15.98
Wi-Fi Module	ESP8266	Sparksfun	$6.95
Total			$135.12

Table 34: Initial Prototype Items Cost.

After carrying out performance testing, the final project budget increased. This is due to purchasing additional components for the replacement of components that were not implemented. This includes disregarding the GSM+GPS module (SIM808) for mobile network connectivity with the PetAid Harness. Some parts are replaced for improved functionality of our features, such as utilizing the ESP8266 instead of the ESP32WROOM.

Components	Website	Quantity	Price	Per Unit Cost
DS18B20+ and Waterproof type	https://www.digikey.com/product-detail/en/maxim-integrated/DS18B20-/DS18B20--ND/956983	1+1	$4.36 + $6.97	$4.36
SEN-11574	https://www.digikey.com/product-detail/en/sparkfun-electronics/SEN-11574/1568-1247-ND/5762397	2	$24.95 each	$24.95
MMA8451	https://www.digikey.com/product-detail/en/adafruit-industries-llc/2019/1528-1041-ND/4990790	2	$7.95 each	$7.95

WIFI Module ESP8266	https://www.sparkfun.com/products/13678	2	$6.95 each	$6.95
WIFI Module ESP32WROOM	https://www.digikey.com/product-detail/en/espressif-systems/ESP32-DEVKITC-32D/1965-1000-ND/9356990	1	$10.00	$10.00
Logic-Level Converter	https://www.digikey.com/product-detail/en/sparkfun-electronics/BOB-12009/1568-1209-ND/5673795	1	$2.95	$2.95
Battery 3.7V 1.2AH	https://www.digikey.com/product-detail/en/adafruit-industries-llc/258/1528-1838-ND/5054544	1	$9.95	$9.95
GSMMQB-UFL 850MHz/900MHz	https://www.digikey.com/product-detail/en/rf-solutions/ANT-GSMMQB-UFL/ANT-GSMMQB-UFL-ND/2781747	1	$6.17	$6.17
ECHO/5.0 GPS Antenna	https://www.digikey.com/product-detail/en/siretta-ltd/ECHO5-0.1M-UFL-S-S-15/ECHO5-0.1M-UFL-S-S-15-ND/6096346	1	$14.48	$14.48
Arduino UNO R3 ATMEGA328P	https://www.digikey.com/product-detail/en/arduino/A000066/1050-1024-ND/2784006	2	$22.00 each	$22.00
SIM808 GPS+GSM	https://www.itead.cc/sim808-gsm-gprs-gps-module.html	1	$40.00	$40.00
Breadboard Power Supply Stick 5V/3.3V	https://www.sparkfun.com/products/13032	2	$11.95 each	$11.95
PCB (5+2 SMT Assembly)	https://jlcpcb.com/	5+2(SMT Assembly)	$48.54	$23.00
9V DC Alkaline Battery	https://www.sparkfun.com/products/10218	1	$1.95	$1.95
PCB SMD Components	https://www.digikey.com/	1	$41.65	$41.65
Other			$116.93	$38.60
Total			$454.55	$266.91

Table 35: Final Prototype Cost.

should take advantage of in the future to keep the finances at the very minimal. Careful selection of the parts needed for the PetAid Harness should be performed because then there will be a long list of wasted project parts that could be affecting our team's finances and budget. Another reasoning behind this is that for many engineering students, this kind of project design is of such substantial magnitude that has never been exercised or experienced beforehand. The conceptual theory and material learned throughout the Electrical and Computer Engineering department from the University of Central Florida, must be utilized to have success when ordering and building these electronic components. This kind of situation is the factor introduced into increasing our cost finances by running various trial and error trials with different components to ensure they contain excellent performance operation for the PetAid Harness. Utilizing the proper and useful component parts, provides greater success for the best operation of the PetAid Harness within the allocated time frame during Senior Design I and Senior Design II semesters. After the PetAid Harness the total cost to develop the prototype is much higher since it involved being in the development process. Once the final design was complete the per unit cost to build one unit is $266.91. This cost can further be reduced if it is introduced into the production stage.

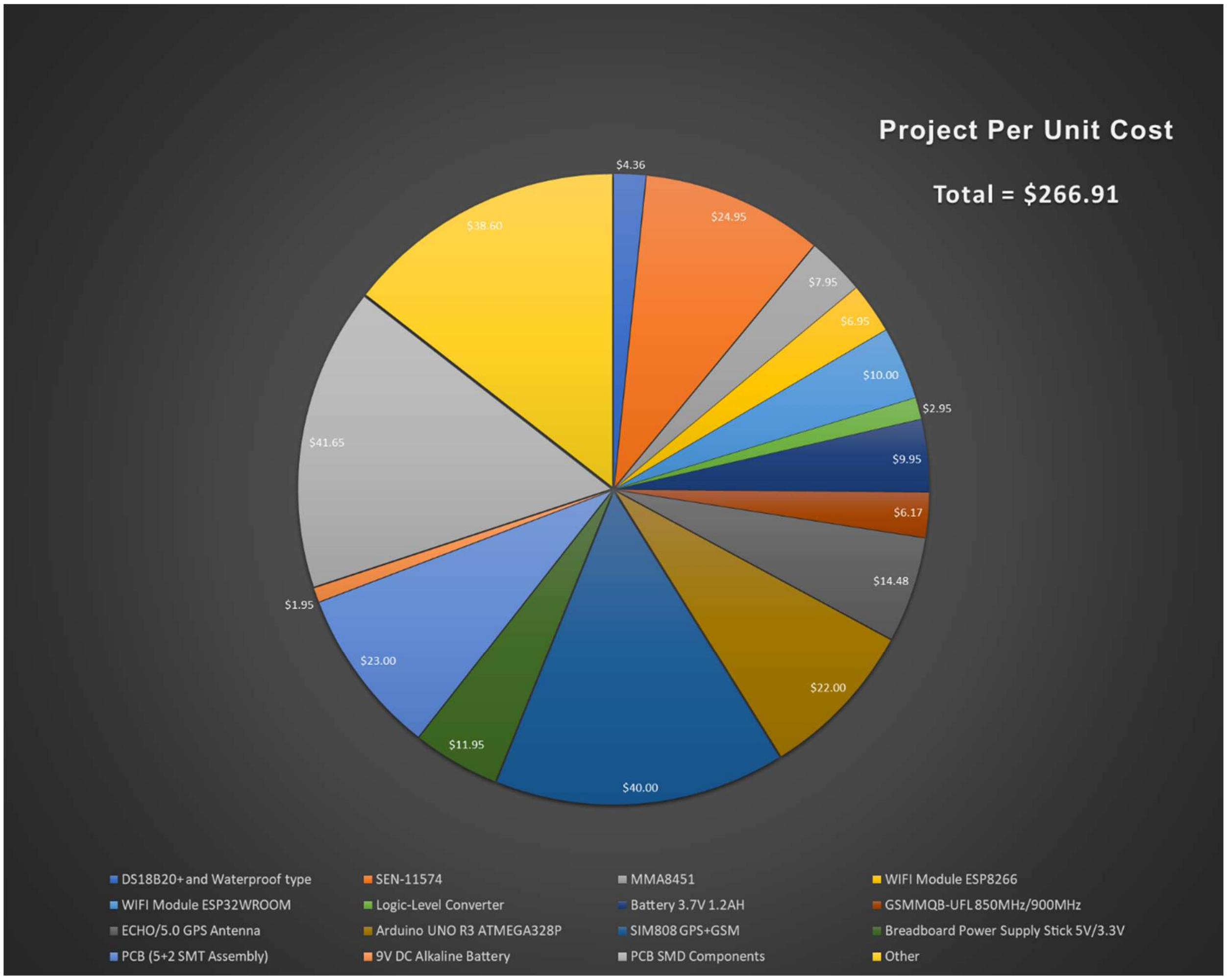


Figure 37: Prototype Project Cost.

10. Administrative Content

10.1. Works Cited

(Heart rate monitor):

"Design and Development of a Heart Rate Measuring Device using Fingertip", Khunla University of Engineering & Technology (KUET), M.M.A Hashem, Rushdi Shams, Md.Abdul Kader, and Md. Abu Sayed, 2013.

(Accelerometer):

"Accelerometer", W3C Candidate Recommendation, Anssi Kostiainen, Alexander Shalamov, 12.Dec.2019. https://www.w3.org/TR/accelerometer/

(Temperature Sensor):

IEEE Standard for Sensor Performance Parameter Definitions," in IEEE Std 2700-2017 (Revision of IEEE Std 2700-2014) , vol., no., pp.1-64, 31 Jan. 2018

(Wi-Fi/Bluetooth Module):

"ESP32(ESP-WROOM-32) Wi-Fi & Bluetooth Dual-Core MCU Module", dfrobot, Hugatry's HackVlog, 23.Oct.2018. https://www.dfrobot.com/product-1559.html

(GPS + GSM):

"What is GPS, how GPS module is used for base station applications, how to choose GPS module?", Techplayon, RF Basics, 4.Oct.2017. http://www.techplayon.com/gps-gps-module-used-base-station-applications-choose-gps-module/

"Android A to Z: What is GSM?", Android Central, Phil Nickinson, 17.Jan.2012. https://www.androidcentral.com/android-z-what-gsm

"GSM (Global System for Mobile communication), Search Mobile Computing, Margaret Rouse, March.2019. https://searchmobilecomputing.techtarget.com/definition/GSM

"SIM808 GPS/GPRS/GSM Shield for Arduino", dfrobot- product wiki.

https://wiki.dfrobot.com/SIM808_GPS_GPRS_GSM_Shield_SKU__TEL0097

(Power Supply):

"What is a Battery?", Sparkfun: Tutorials, Shawn Hymel, 3.May.2016.

https://learn.sparkfun.com/tutorials/what-is-a-battery/all

"Battery Technologies", Sparkfun: Tutorials, Nate, 6.Feb.2013.

https://learn.sparkfun.com/tutorials/battery-technologies?_ga=2.90668506.1712385747.1586311501-1763906661.1530135374

"Introduction to DC-DC Converters", Maker.io, Maker.IO Staff, 18.Feb.2016.

https://www.digikey.com/en/maker/blogs/introduction-to-dc-dc-converters

"Navigating the Regulatory Maze of Lithium Battery Saftey", Intertek.com, Tom O'Hara, 20. August. 2018

http://batterypoweronline.com/wp-content/uploads/2013/11/Intertek_Regulatory-Maze-WP.pdf

"IEEE Standard for Rechargeable Batteries for Multi-Cell Mobile Computing Devices", IEEE, IEEE Staff, 20. Oct. 2008

https://ieeexplore.ieee.org/document/4657368

"10 Best Practices of PCB Design", EDN.com, Edwin Robledo and Mark Toth, 7. March. 2014 https://www.edn.com/ten-best-practices-of-pcb-design/

(EDGAR) I need to add mine

11. Appendix

This is a section to go over notes to consider while the document is being created. Currently, it is addressing the issues that have come up with recent events occurring, as a large portion of our expected activities for the class have been made either difficult or impossible by what has been happening in recent months.

11.1. Progress Hindrance with Recent Events

Given the change of the class from on-campus lecture to online and the required social distancing, the entire group has been affected in ways that neither we nor the professors could predict or control. Adjusting with these changes has put us into a position of reactivity as opposed to proactivity; ensuring the well-being of family, of ourselves, preparation for lengthy self-isolation time, among other things to avoid as much trouble as possible. This has made the ability to meet in person impossible, which we believe to be far more valuable than our backup meeting method through either WhatsApp or Discord, and ordering the parts we've chosen for the PetAid Harness in order to build it and document our findings with the hardware, ideas for testing ethically, and discovery of unexpected factors have all been made nearly impossible with the current climate of affairs. While this is true, it is happening for the entire school, and as a result we are not likely to be uniquely affected by the situation at hand.

11.2. Datasheets

Heart Rate Monitor:

SEN-11574:
https://media.digikey.com/pdf/Data%20Sheets/Pulse%20Sensor%20PDFs/Pulse_Sensor_GSG.pdf

Temperature Sensor:

https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

Accelerometer:

https://cdn-shop.adafruit.com/datasheets/Freescale_MMA8451QR1.pdf

GSM/GPS Module:

https://cdn-shop.adafruit.com/datasheets/SIM808_Hardware+Design_V1.00.pdf

Wifi/Bluetooth Module:

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Logic Level Converter:

https://www.onsemi.com/pub/Collateral/BSS138-D.PDF

Arduino Uno:

https://media.digikey.com/pdf/Data%20Sheets/Arduino%20PDFs/A000066_Web.pdf

Power Jack Supply:

https://learn.adafruit.com/ladyadas-learn-arduino-lesson-number-0/power-jack-and-supply

9 Volt Battery Case:

https://www.digikey.com/product-detail/en/67/1528-1116-ND/5353576

DC Power Jack:

https://www.sparkfun.com/products/13975

DC Barrel Jack for PCB:

https://www.sparkfun.com/products/12748

Microcontroller tips:

https://www.microcontrollertips.com/key-factors-consider-choosing-microcontroller/

MSP430FR6989:

http://www.ti.com/lit/ug/slau627a/slau627a.pdf

Prepaid Energy System:

http://www.eecs.ucf.edu/seniordesign/fa2015sp2016/g21/doc/prepaid%20energy%20sysem.pdf

Wifi vs Bluetooth:

https://www.autodesk.com/products/eagle/blog/wifi-vs-bluetooth-wireless-electronics-basics/

LBEE5KL1DX module:

https://www.digikey.com/product-detail/en/murata-electronics/LBEE5KL1DX-883/490-13461-1-ND/6043961

Smart Lights:

http://www.eecs.ucf.edu/seniordesign/sp2017su2017/g02/documents/Senior%20Design%20Two%20Documentation.pdf

CC3100 wifi:

http://www.ti.com/lit/ds/symlink/cc3100.pdf

Health and Fitness Tracker for canine pets:

http://www.eecs.ucf.edu/seniordesign/su2017fa2017/g15/Group15_Conference_Paper.pdf

Pulse Sensor Getting Started Guide:

https://media.digikey.com/pdf/Data%20Sheets/Pulse%20Sensor%20PDFs/Pulse_Sensor_GSG.pdf

Pulse Sensor Datasheet:

https://media.digikey.com/pdf/Data%20Sheets/Pulse%20Sensor%20PDFs/Pulse_Sensor.pdf

AD8232 Heart Rate Monitor:

https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/AD8232_HeartRateMonitor_HookupGuide.pdf

Heart rate sensor we choose or likely : Link below to explain why we did choose this heart rate sensor

http://www.eecs.ucf.edu/seniordesign/su2017fa2017/g15/Group15_Conference_Paper.pdf

https://www.intorobotics.com/pick-best-temperature-sensor-arduino-project/

https://www.bmmagazine.co.uk/business/how-do-temperature-sensors-work/

https://www.omega.co.uk/temperature/z/thermocouple-rtd.html

Accelerometer: What they are and how they work:

https://www.livescience.com/40102-accelerometers.html

Helmet Tracking System:

http://www.eecs.ucf.edu/seniordesign/sp2013su2013/g08/HTS_ConferencePaperLi-Ion.pdf

ADXL345 Datasheet:

https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf

ADXL345 DIGIKEY:

https://www.digikey.com/product-detail/en/ADXL345BCCZ/ADXL345BCCZ-ND/2034829/?itemSeq=320620722

MOU-9250 DataSheet:

https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf

Selecting MCU:

file:///C:/Users/18632/Downloads/capstone.pdf

HC-05:

http://www.electronicaestudio.com/docs/istd016A.pdf

Nrf51822 DATASHEET:

https://www.nordicsemi.com/-/media/Software-and-other-downloads/Product-Briefs/nRF51822-product-brief.pdf?la=en&hash=A4B5A9AA6675A58F7B779AF81C860CD69EB867FD

esp8266 DFRobot:

https://botland.com.pl/en/wi-fi-modules/7453-dfrobot-wt8266-s1-wifi-module-based-on-esp8266.html

MAX30100 Datasheet:

https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf

AD8232 Datasheet:

https://cdn.sparkfun.com/datasheets/Sensors/Biometric/AD8232.pdf

TMP35 datasheet:

http://www.farnell.com/datasheets/1793789.pdf?_ga=2.1246449.1521236456.1585611839-2144125958.1585611839

IC sensors:

https://www.omega.com/en-us/resources/integrated-circuit-sensors

MMA8451 datasheet:

https://cdn-shop.adafruit.com/datasheets/MMA8451Q-1.pdf

Accelerometer/compass sensor:

https://www.adafruit.com/product/1247

All about batteries:

https://learn.adafruit.com/all-about-batteries/alkaline

ProjectHub:

https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806