SMART HARNESS

University of Central Florida
Department of Electrical Engineering and Computer Science
Dr. Lei Wei

Group 15 Members:
Dominic Vu - *Computer Engineering*
Hai Nguyen - *Computer / Electrical Engineering*
Matt Horton - *Electrical Engineering*

Table of Contents

Figure Index.

Table Index.

1.0 Executive Summary

The problem our team has decided to tackle for our senior design project is the seemingly small amount of information that is readily available when it concerns a dog's health and/or fitness. This project will be to create a harness that will contain an electronic device. This device will monitor certain aspects of a dog's health and well-being using a few well-placed sensors. The harness and electronic device will be half of the design, the other half being a mobile application. This application will be used to interface between the harness and the pet owner and will then be displayed on a mobile device.

To create this electronic circuit, many modules will need to be selected in order to implement our design. Included in this list of items will be modules such as a Bluetooth module, battery, accelerometer, etc. Throughout the semester we will thoroughly research all of these modules so that we can select the ones that will best for our needs for a low cost. Once all these modules have been selected, we will design a PCB (Printed Circuit Board) that will be the basis of our design. Attached to this PCB will be multiple sensors that will read various vitals and will report back to the PCB, which in turn sends the information to the mobile device via Bluetooth.

Most technologies in our modern age are accompanied by a mobile application. This project will be no different. The mobile device will be what the user will interact with. This application will be where all the data from the harness will be stored along with other information the user will input themselves. The idea is that all of the dogs' information will be easily accessed. The application will let you see trends in the health and activity of the dog so that if they are not getting enough exercise you can make sure to take them on an extra long walk.

Our team decided to tackle this problem because two out of the three team members do have dogs at their homes. With all the technology that exists out in the world, it seems almost every part of our lives is integrated with our electronics. Although this is true for humans, very little smart technology seemed to exist for our pets. Many pet owners would likely be interested to know more about the health of their furry friend, so the applications for a device such as this could keep getting more innovative as the years progress and technology advances.

The purpose of this report is to outline the research, testing, implementation, and budget as well as numerous other aspects involved with creating a project of this scale. Throughout this document, all of the information will be explained in detail and will complete our senior design capstone requirements.

2.0 Project Description

One problem with owning a dog is not having easily accessible health and fitness information that concerns your pet. In the age of smart technology it seems our four-legged friends have been forgotten. Knowing when your pet may be ill or helping keep him at a healthy weight is important, and a device that can assist in these tasks can be powerful for any dog owner. The motivation for this design is to provide a tool for your basic consumer that has a concern for their pet’s fitness and well being as well as the professional that needs an easy and effective way to record a dog’s health information.

To start, this harness will be adjustable, lightweight, portable, and very comfortable so that the dog can wear it around its body at all times without the need to constantly remove it. This could possibly eliminate the need for a collar. It should also be low cost and easy to use as this is meant for all consumers. Many features that would be included are similar to a human fitness device. It would contain a heart rate monitor using a sensor placed on the dogs chest. The device will also contain an accelerometer to be used as a pedometer (step counter) so you can monitor the activity level of the dog throughout the day and even when you are away. There will also be a thermometer/sensor placed in the front armpit to take temperature readings. While these are not as accurate as the invasive rectal thermometer, they can be used as a supplemental information source. The device will also contain Bluetooth capabilities for easy pairing with mobile devices.

The harness will also work in conjunction with a smartphone application. A feature that will be contained in the app is the ability to monitor your dog's activity level throughout the day. If a vet informs an owner that their dog is at an unhealthy weight, the user can input the dog's weight into the application and set up reminders to make sure that he or she is getting enough exercise. While the average consumer can get a lot of benefit out of this design, a veterinarian office could find many uses as well.

Whether this is for the housewife that cares dearly about their furry family member, the child who is caring for their first pet, or the veterinarian that is caring for a handful of puppies, the object of this smart harness is to provide a tool that anyone can use to keep track of their dog’s health and have a sense of reassurance that the dog’s health is well.

2.1 Project Goals

The goal of this project is to create a device successfully monitor the health of the dog during their normal condition such as when they sleeping, standing still, or running with heartbeat, temperature and accelerometer. Let say that when the dog was sleeping the monitor on the phone device application will be able to show their temperature and heart beating during sleeping of the dog. Meanwhile the dog is stand, the application will be able to show the normal heartbeat and temperature of the dog. Finally when they walking or running, the application will show the increasing the step counter from accelerometer and the increasing heartbeat.

The next goal of this project is the successful connection between sensor, MCU, and the Bluetooth with the application on the mobile phone. The application will show the current temperature, heartbeat, and the increasing of step counter on the application format. The software will save inside of the device with the list of heartbeat, temperature based on every minutes. The step counter will be able to display on the application of the mobile apps.

The device has to be able design with the cost is much cheaper compare to all the existing device which currently sale on the market. This section will be mention on the 2.3 outside world comparison.

The last goal is successfully to have small design from one of our engineering requirement and be able to fully function with the battery as the supply. The design for MCU size case is really important which will decide whether the device will need to be able install inside of the harness belt. Next, the battery need to be able supply the hardware and also last enough during our demo time on senior design 2.

2.2 Functionality

The functionality of this project will divide into two parts including hardware and software.

Hardware Function.

- Power on by the battery.
- The connection between an android phone or tablet and the MCU through bluetooth.
- The temperature, heartbeat, and accelerometer need to be able to sense the dog's temperature, heartbeat, and steps and send those information to the application via bluetooth connection.
- The device will be able anti water coming from outside.
- The bluetooth and the accelerometer will be integrated inside the MCU.
- The temperature, and heartbeat sensor will be on the different location of the belt.

Software Function.

- Display the heartbeat per minute.
- Display the temperature per second.
- Display the step counter when every time the dog make the movement.
- Have the history tab where the user can track all the data.
- Showing the successful connection with the bluetooth.

2.3 Outside world comparison

Our project is not only the first thinking about the idea dog harness. There are many product outside world was made in order to measures the dog activity.

The first product is collar for dog which is called Link AKC at https://shop.linkakc.com/. This product is the winner of CES 2017 Best Of Innovation. It has many function such as fast tracking dog through GPS, remote turn on and off LED light to help the user find the dog in the dark, remote sound to help training, digital scrapbook, and temperature alerts. In addition, this product has four sizes which is small 9.5” - 14”, medium 14” - 17.5”, large 17” - 21”, and extra large 20.5” - 25”. Finally the cost of this product is $199 dollars.

The second product is petpace which is sale on petpace.com. This product is not only for dog but also for cat. It is same as Link AKC the collar type. According to the website, “The PetPace collar uses non-invasive sensors to track temperature, activity, pulse, respiration, positions, calories consumed and burned, and heart-rate variations — all the elements underlining your pet’s overall health.” also their application is compatible is IOS and android device. The product weight 1.5 ounce and it can send the data rate at 2, 15, and 30 minutes interval. The product is cost $150 dollars.

Not only with the product that exist outside but also there are many idea current developed to help the people understand and monitor their pet’s wellness. This product currently under testing and developed in Japan called Inupathy. According to the website (http://inupathy.com/), the Inupathy is the product which will show the emotional of the dog during the time whether he/she relaxed, excited, happiness, interested, and stressed. The product divide into five color based on five emotional from the dog such as green (relaxed), orange (excited), rainbow (happiness), white (interested), and purple (stressed). Those emotional color are based on the calculation from the heartbeat of the dog, and it will showed the LED. With those light, the user will be able to tell the emotion of the dog. In addition, this product is connect with the mobile application, and it show the chart graph of the dog based on time like second, minute, day. The initial cost for this device is $169 dollars

In the conclusion, the market for the dog health monitor product is average above $150 dollars, and those are all the collars. Currently, there are not the products with the belt harness for dog in our group opinions. Our project might be the first one to make the belt harness for dog with the initial cost might much more cheaper compare to the price tags on the market.

2.4 Requirements

Every designs or project always need to have the requirements. It is critical to the student which direction they need to be taken in order to follow and create it. This will be a good practice for student because of outside world. First of all, the productor will using this the requirement to create the exact product before testing and sending to the market. Secondly the consumer will buy the product based on those criteria. To sum up, the harness project from group 15 will have both hardware requirement as well as software requirement.

2.4.1 Hardware Requirement

- Microcontroller
- Compact sensors to detect heart rate and temperature
- Total weight < 5 Lbs
- Accelerometer to record steps
- Bluetooth connectivity to mobile device
- Battery/Power Supply
- Safe for all dogs

2.4.2 Software Requirement

- User friendly
- Notifications for specific events (high temperature)
- Record steps (pedometer)
- Activity/Weight tracker
- Bluetooth connectivity to hardware
- Receive data from user and harness
- Display data from sensors
- Access/display heart rate
- Temperature reader
- Temperature records
- Pet health history records
- Android device compatible
- Java SE

2.4.3 Harness Requirement

- Durable
- Adjustable (fits dogs of many sizes)
- Lightweight, breathable fabric
- Comfortable
- Low Cost

2.5 User Manual

One of the main goals of this project was to make this harness as easy to use as possible. When the harness is initially utilized by the dog owner, some information will be needed and the user will be prompted to enter this data accordingly. Some of this data will include weight, age, known health issues, and other factors. Once completed, the user should be able to access information about their pet easily and quickly.

Step 1: Pair the Device

Before starting the application, the user will attempt to pair their mobile device with the Smart Harness. The Smart Harness will be Bluetooth enabled and should pair easily with an Android device easily (Apple devices may pair, but the app. may not work properly). The way this will work is there will be a button on the screen that will say "Please pair with the device". At this point, the user would click this button and press a button on the harness to act as an endpoint and start the communication between the phone and the harness. After 15-30 seconds, the two devices should be paired. The user will be notified of the success or failure of pairing and at that point the user would be directed to the setup of the application or prompted to retry to connect.

Step 2: Input Data

Once the connection is established, the application will prompt the user to input specific data about their dog. The information that will need to be provided will include but will not be limited to: the dog's name, weight, sex, age, color, breed and other notes that would be beneficial such as health issues or medication that the dog is on. All this information will be added to and displayed on the dog's profile. You will also have the ability to add a photograph of the dog if desired. Once the the dog's information has been updated, the user will be prompt with a list of slides that will walk through the application and how to use it. It will show you how to initialize the heart rate reading, the temperature reading, it will show you what kind of data is displayed and what it means, it will show you how to ping your veterinarian and send them critical or concerning information if needed. It will also show you how to do the initial pairing with your veterinarian so that that information can be transferred.

Step 3: Safely Strap on Harness

The next thing that the user must do is to make sure that the harness has been fitted to the dog properly and is powered on properly even after bluetooth pairing. The instructions will include detailed steps and illustrations on where to put the harness, where to put the dogs feet through, what buckles are used for what, and the suggested fitment for optimal use and comfortability. The instructions will also clarify where the heart reader and temperature readers are and how they work so that the placement is clearly defined and understood why it should be where it is. This important for the optimal function of the harness and the more the user understands, the easier it is them to diagnose issues if the readings do not seem appropriate or correct. If the dog is in danger of being shocked or harmed in any way or if any of the electronic equipment is exposed in an unsafe manner, REMOVE IMMEDIATELY. We all care for our pets and do not want them in any kind of danger.

Step 4: Sensor Placement

Once it has been fitted safely, make sure the sensors are placed in the correct locations so that the information will be detected properly. If the information from the sensors is not read correctly, you may get inaccurate information about the dog's health. For instance, if the temperature sensor is improperly attached, you may not know if your canine has a fever. These instructions will be displayed with detailed illustrations and steps required for calibration. Proper calibration is pertinent to accurate readings.

Step 5: Enjoy

After you have input the data, equipped the dog with the harness, and placed the sensors properly, the data should now begin to read. This data will include the heart rate, temperature, and the amount of steps the dog has taken. Now you can track the health of your dog! The end of the user manual should display a place for problems and solutions or questions and answers. We will most likely provide a link to a website with more information and a forum for discussion amongst company personnel for assistance in issues as well as peer to peer interaction. Facebook, instagram and yelp will also be included as this day and age, social media can contribute immensely to the success of a product/company.

2.6 User Manual Flowchart

This gives a simple breakdown of the steps involved with using the software. The block diagram is basic and easily understood because the use of the software was designed not to be complicated. The application was designed to be as easy to use as possible, once the initial information of the pet is input, the data is transmitted via Bluetooth to the application easily and will allow for the user to monitor their pet with ease.

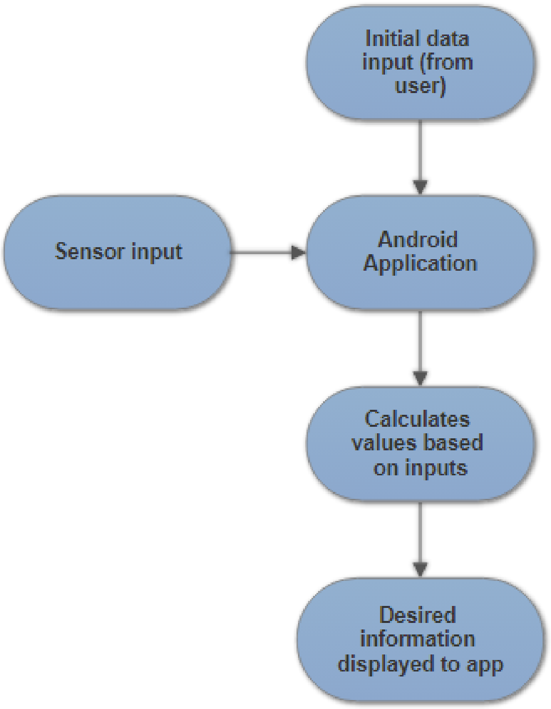

Initial data input (from user)
Sensor input
Android Application
Calculates values based on inputs
Desired information displayed to app

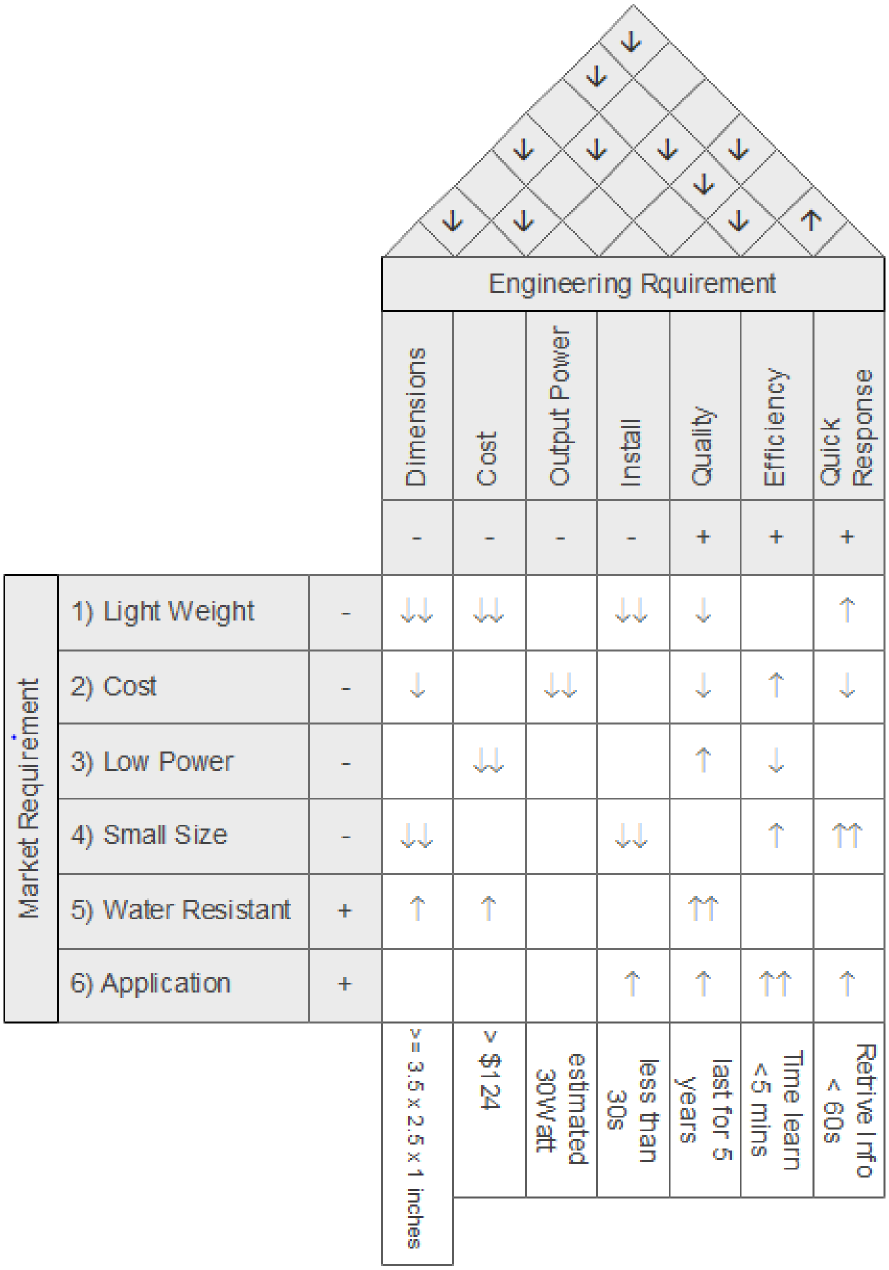
Figure 1 User flowchart

2.7 Safety Warnings and Hazards

Our user manual will consist of a “Safety Warnings and Hazards” section because it would be one of our constraints to include one as well as an ethical and political decision to avoid future lawsuits and legality. This section will include instruction and direction to best avoid any safety impingements or hazards. First off the device should not be utilized or operated by children under the age of 13 unless there is parental guidance or supervision.We have chosen this number because we do include electrical components that may very well cause physical harm to an individual if not used correctly. Also, it is a harness for an animal which has lots of holes that are large enough to fit a small childs head into and can cause accidental strangling. There are also a lot of small buckles that can fall off and a child can choke on it. Another warning or hazard would be to not completely submerge the harness in water while the harness is on the animal or when it is off. Right now we will not offer a product that is completely waterproof or submersible in water but only splash proof so to submerge it in water could cause damage to its internal components as well as the risk of electrical shock. We will also include general safety and hazard precautions such as disposing of the harness properly. If a user were to ever throw away the harness, they would need to do so by recycling of it properly or taking it to a designated place that disposes of electronics as it does contain those components. Lastly, we would add general precautions such as not using the harness in any way that it is not intended to be used.

2.8 House of Quality

The house of quality is necessary for our to define the project requirement so that we can based on those requirement to do research and build our hardware component and software applicatio.



		Dimensions	Cost	Output Power	Install	Quality	Efficiency	Quick Response
		-	-	-	-	+	+	+
1) Light Weight	-	↓↓	↓↓		↓↓	↓		↑
2) Cost	-	↓		↓↓		↓	↑	↓
3) Low Power	-		↓↓			↑	↓	
4) Small Size	-	↓↓			↓↓		↑	↑↑
5) Water Resistant	+	↑	↑			↑↑		
6) Application	+				↑	↑	↑↑	↑
		>= 3.5 x 2.5 x 1 inches	> $124	estimated 30Watt	less than 30s	last for 5 years	Time learn <5 mins	Retrive Info < 60s

Figure 2 The complete House of Quality for the dog smart Harness.

2.8.1 Notation of House of Quality

- + Positive polarity
- - Negative polarity
- ↑ Positive correlation

- ⇈ Strong positive correlation
- ⇊ Strong negative correlation
- ↓ Negative correlation

2.9 Hardware diagram.

The hardware design is very important for the dog harness project. The Hardware design below will show how our circuit work in general and the responsible of each group member.

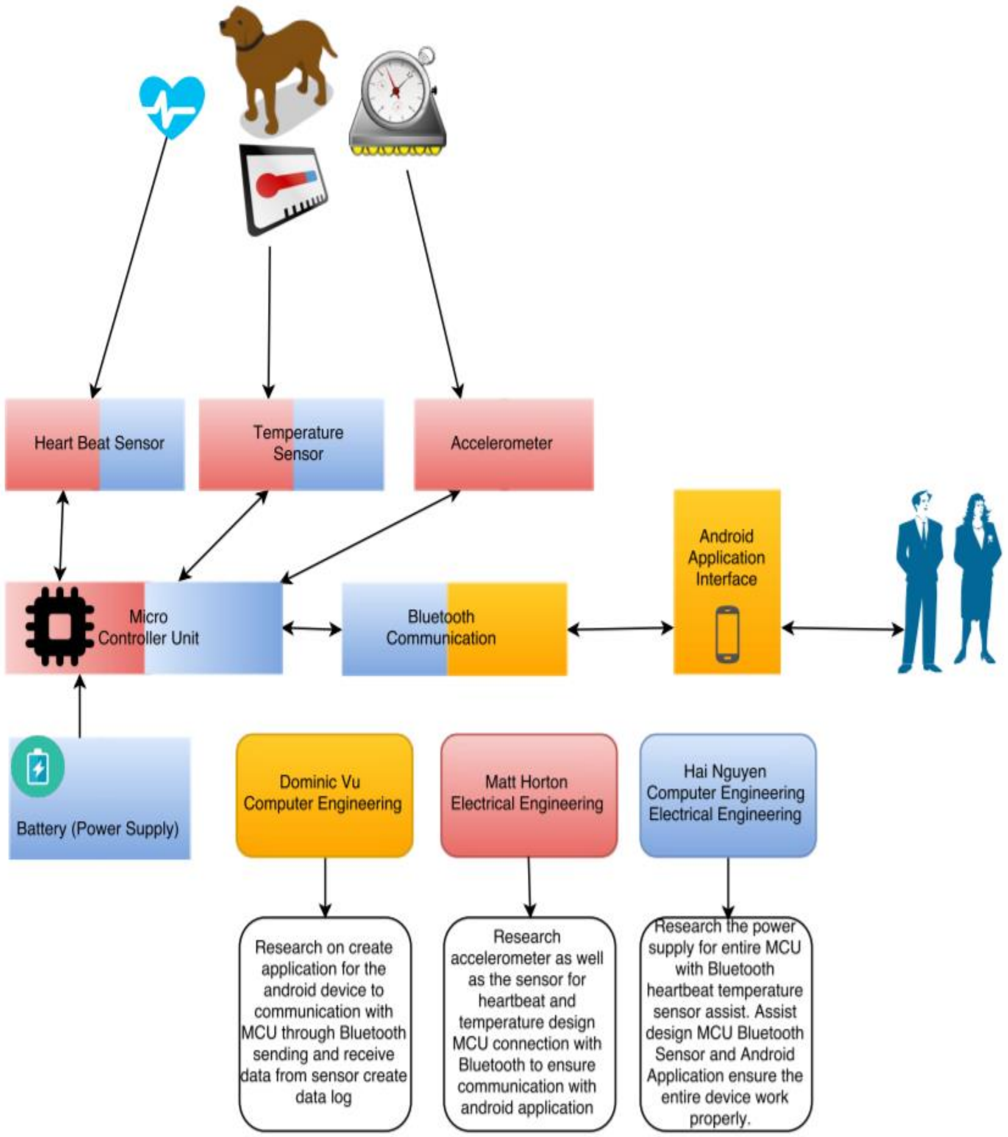

Heart Beat Sensor
Temperature Sensor
Accelerometer
Micro Controller Unit
Bluetooth Communication
Android Application Interface
Battery (Power Supply)
Dominic Vu Computer Engineering
Matt Horton Electrical Engineering
Hai Nguyen Computer Engineering Electrical Engineering
Research on create application for the android device to communication with MCU through Bluetooth sending and receive data from sensor create data log
Research accelerometer as well as the sensor for heartbeat and temperature design MCU connection with Bluetooth to ensure communication with android application
Research the power supply for entire MCU with Bluetooth heartbeat temperature sensor assist. Assist design MCU Bluetooth Sensor and Android Application ensure the entire device work properly.

Figure 3 Completion of Hardware Diagram.

2.10 Software diagram.

Beside the hardware, the software is also used for our project. This is the brief picture about how our software application will flow.

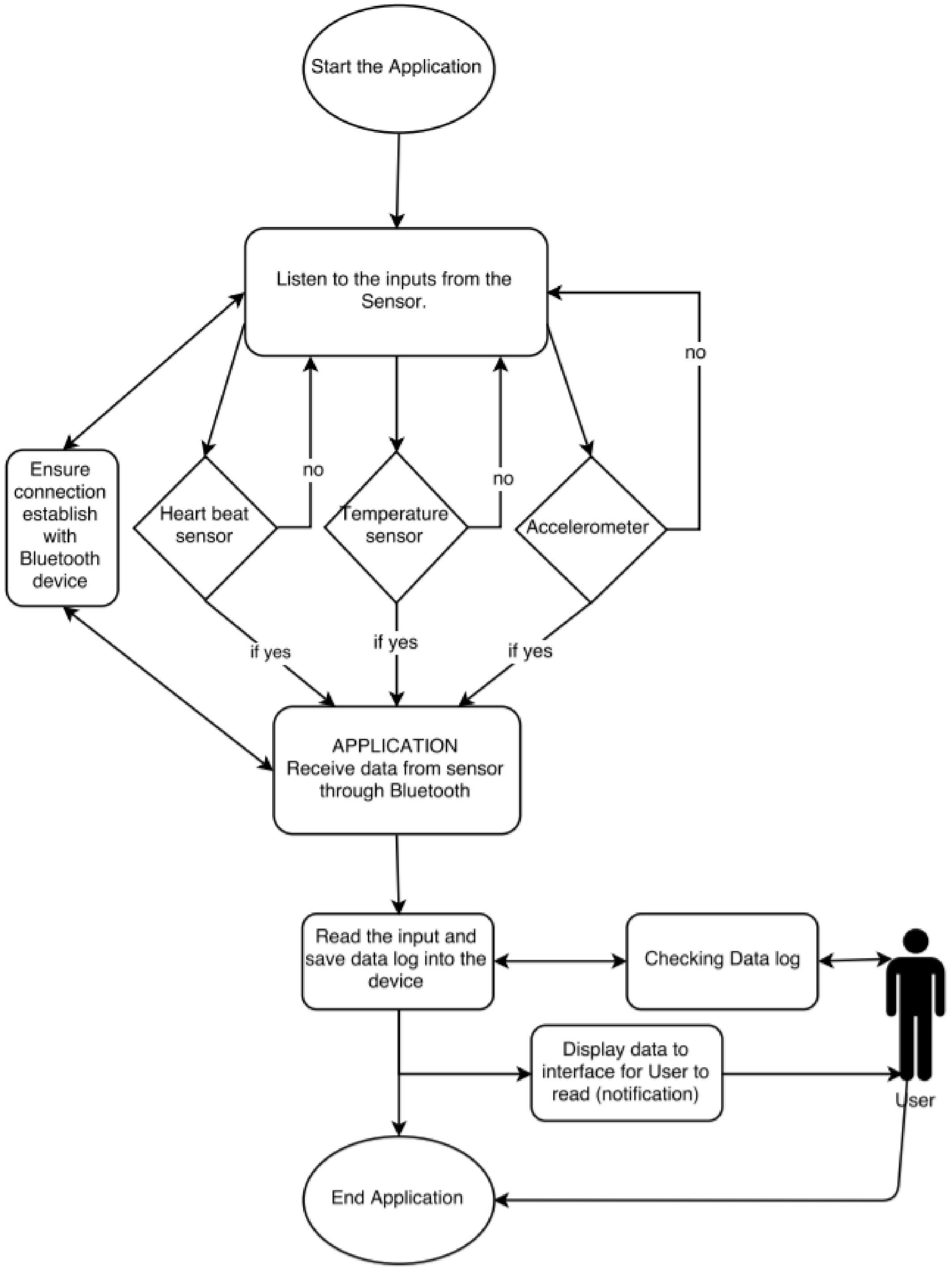

Start the Application
Listen to the inputs from the Sensor.
Ensure connection establish with Bluetooth device
Heart beat sensor
Temperature sensor
Accelerometer
no
no
no
if yes
if yes
if yes
APPLICATION Receive data from sensor through Bluetooth
Read the input and save data log into the device
Checking Data log
Display data to interface for User to read (notification)
User
End Application

Figure 4 Completion of Software Diagram.

3.0 Project Research

All research that was conducted for this design project will be included in this section. Extensive research has been performed for both hardware and software. The hardware components that were researched for this project include a microcontroller, accelerometer module, bluetooth module, temperature sensor, and a battery source. For the software research, many different things will be considered. Among them, will be to decide between using Android or Apple devices as well as deciding which high-level programming language that will ultimately be used for our design.

3.1 Microcontroller Module

It is necessary to have a an MCU for the smart harness as it is the brain of the hardware design. Everything centers on it. The battery will be powering the MCU and the MCU itself will distribute that power to the rest of the modules. We are not entirely sure what the memory requirement may be when it comes to the function so having an abundance of available memory is a definite plus. While cost is always a limiting factor, having an MCU that meets all our requirement while being easy to test and implement is of vital importance. So if any hardware component is going to be higher in cost, the MCU should be that part. An MCU with an easily removable chip (the only piece we can include in our design) makes it much easier to test, and being able to extract the chip without dealing with a soldered on chip is a huge advantage, as we are less likely to damage the chip in the removal process.

In addition to a removable chip, having an MCU that is accompanied by a development board is also important as it facilitates the initial coding and testing. If you acquire just the chip and no developmental board the testing process can be a bit more difficult. Another factor is our preference is to use an Arduino as they have a large library of code examples which can be utilized. This can make the project proceed smoothly as many issues we may run into are already known and solved. In the next section all of the microcontrollers considered are listed and each have distinct advantages and disadvantages. For the Smart Harness one of the following MCU's will be chosen based numerous criteria so that we can have a good fit for the project needs while making the process go as smoothly as possible.

3.1.1 Arduino Uno

This microcontroller board is built to house the ATmega328P. The ATmega328P chip is an 8-bit MCU that comes with 32 general purpose registers. It has an operating voltage of 5V, 14 digital I/O pins, 6 PWM digital I/O pins, and 6 analog input pins. Other features are a 16 MHz clock speed, 20 mA DC current per I/O pin, 10 ADC channels and 32 KB of flash memory. The ability for the chip to be removed allows for us to be able to test on a breadboard much easier than having to remove a soldered on chip.

Positive Factors

- Arduino Uno has a large library of code examples to utilize.
- Easily removable chip.
- Small, compact, and lightweight

- Operates at a low voltage range (1.8V – 5.5V)
- Operates with a low current per I/O pin (20 mA)
- Built in clock sources
- 32 KB of memory (0.5 utilized by bootloader)

Negative Factors

- Higher cost board ($24.95)

3.1.2 TI MSP430 Launchpad

The MSP430 Development board is created by Texas Instruments. This board houses the MSP430G2553 MCU. This is a 16-bit MCU with 16KB of memory, a 16 KHz clock speed and very low current: Active mode is 230 µA, Standby Mode is 0.5 µA, and Off mode at 0.1 µA. The voltage supply range is low as well, between 1.8V and 3.6V. This MCU is like the Arduino Uno in that the chip can be removed from the board, which allows for the ability to test on a breadboard much more easily. However, TI's Code Composer Studio must be used.

Positive Factors

- Easily removable chip.
- Small, compact, and lightweight
- Operates at a low voltage range (1.8V – 3.6V)
- Operates with a low current (230 µA)
- Onboard clock source
- 24 I/O pins

Negative Factors

- Average Memory - only 16 KB
- Restricted to using TI's Code Composer Studio
- Moderate cost ($9.99)

3.1.3 Adafruit Trinket – Mini Microcontroller

The Adafruit Trinket is a very small microcontroller board. This board houses the Atmel ATiny85 chip which contains 8 KB of flash memory, 8 MHz clock that can be doubled in the software and 5 I/O pins. There are two versions, one is a 3V, and the other is a 5V. Durable bootloader process that does not fail in the middle of a project. Although this chip has a lot of promise, the fact that it is soldered to the board makes it unlikely to be used.

Positive Factors

- Very small
- Compact and lightweight
- Operates at a low voltage (3V or 5V)
- Operates with a low current (230 µA)
- Onboard clock source
- Very low cost ($6.95)

Negative Factors

- Very low memory - only 8 KB
- Chip is soldered to board, not easily removed
- Only 5 I/O pins

3.1.4 PICAXE 08M2 Microcontroller

The PICAXE 08M2 microcontroller is a very small and very low cost chip. It is a small and lightweight chip which contains 2 KB of flash memory, a max frequency of 32 MHz and 6 I/O pins. It operates at voltages between 4.5V and 5V. Additionally it operates at low currents as well, around 20 mA at each pin. However, a drawback is the lack of large community software support that surrounds other Arduino.

Positive Factors

- Very small, compact, and lightweight
- Operates at a low voltage (4.5V or 5V)
- Operates with a low pin current (20 mA)
- Onboard clock source
- Extremely low cost ($2.95)

Negative Factors

- Extremely limited memory - only 2 KB
- Restricted to PICAXE specific software packages
- Less software community resources compared to Arduino
- 6 I/O pins
- No included development board, harder to code

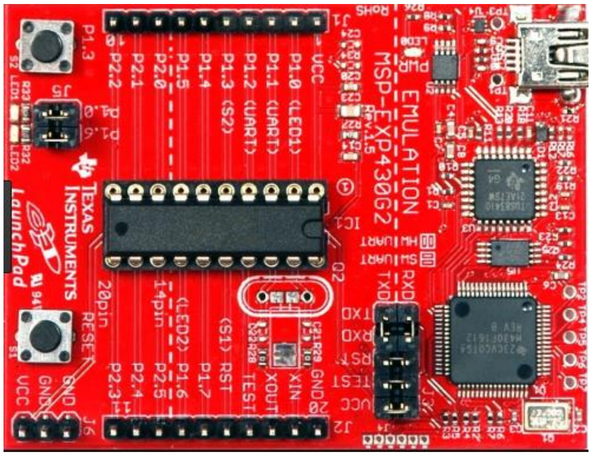


Figure 5 MSP430 Launchpad

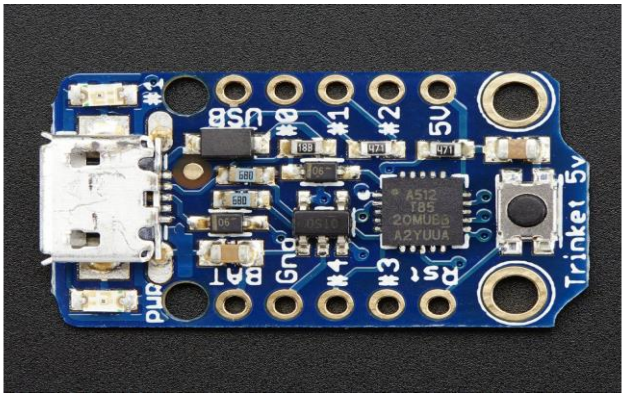


Figure 6 Adafruit Trinket

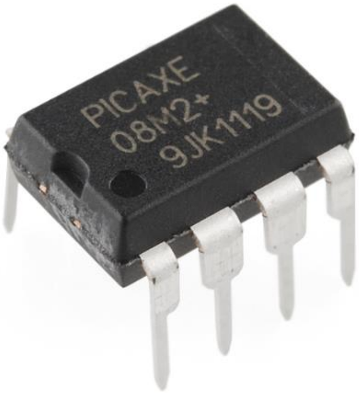


Figure 7 PICAXE 08M2 Microcontroller

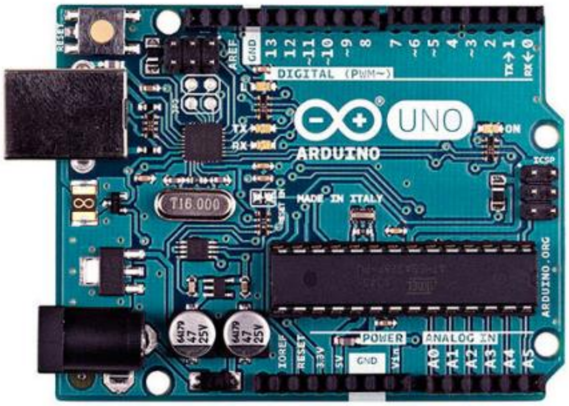


Figure 8 Arduino Uno (Selected MCU)

3.1.5 Microcontroller Parts Comparison

All of the microcontrollers chosen to be included in this comparison tended to be small in size and weight as well as having low energy consumption. Each of the microcontrollers have many advantages and disadvantages. While they all tend to be low in voltage and current, other factors vary greatly. Memory needs may be greater than some of the components can provide, such as the PICAXE 08M2 which only has 2 KB. Cost is another factor which should be considered carefully, as the Arduino Uno can be a good choice but it comes with the highest cost at $24.95. The MSP430 Launchpad may prove to be a good choice as it has the most I/O pins but it does have the drawback of a moderate cost. A very important factor is the ability to remove the chip from the development board that may or may not be provided. The Adafruit Trinket is not a good fit for this project as the chip would have to be carefully removed as it is soldered onto its circuit board. The board being provided is useful as it allows coding to be tested and implemented onto the chip before the PCB is ordered.

	Arduino Uno	MSP430 Launchpad	Adafruit Trinket	PICAXE 08M2
Chip	ATmega328P	MSP430G2553	Atmel ATiny85	PICAXE 08M2
Cost	$24.95	$9.99	$6.95	$2.95
Removable Chip	Yes	Yes	No	N/A
Testing Board Included	Yes	Yes	Yes	No
Onboard Clock	Yes	Yes	Yes	Yes
Clock Speed	16 MHz	16 MHz	8MHz - 16 MHz	32 MHz
Memory	32 KB	16 KB	8 KB	2 KB
Voltage	1.8V - 5.5V	1.8V - 3.6V	3V or 5V	4.5V - 5V
Number of Pins	28	24	5	6

Table 1 Microcontroller Unit Comparison Table

3.1.6 Microcontroller Module Selection

The criteria that are important to the design of our project helped us narrow down which MCU we chose for our Smart Harness. The main factors to consider were the ease of removal of the chip itself, low voltage, ample memory, and inclusion of a board for testing purposes.Taking into account all these reasons we have decided to develop our Smart Harness using the Arduino Uno that uses the ATMega328P chip. While being of higher cost it accommodates all of the factors we were looking for in addition to being low size, and low weight. Seeing as though this chip is the central hub of the hardware design it is acceptable to have this particular component cost a little more as the quality that comes with it is of high importance.

3.2 Temperature Sensor

Temperature Measurement been made since 1592 which is thermometer by Galileo according to Capgo Website. The temperature measurement is call as thermometer, and it has been widely used for many application such as measure human temperature, outside and inside measurement. The thermometer has grow along side with Human development until 20th century and It was adapted by modern science and develop into temperature sensor. Nowadays, there are many type of temperature sensors and they are widely used in many field such as medical, military, heavy industrial, computer, electronics, etc.

What is the temperature sensor? According to Trerice “ A temperature sensor is a device, typically, a thermocouple or RTD, that provides for temperature measurement through an electrical signal.” RTD stand for Resistance Temperature Detector which use a variable resistor that will change its electrical resistance into direct proportion into temperature measurement in a precise linear manner.

Temperature sensor has 3 pin to connected to the MCU including 1 for Vcc, 1 for ground, and 1 for analog voltage out

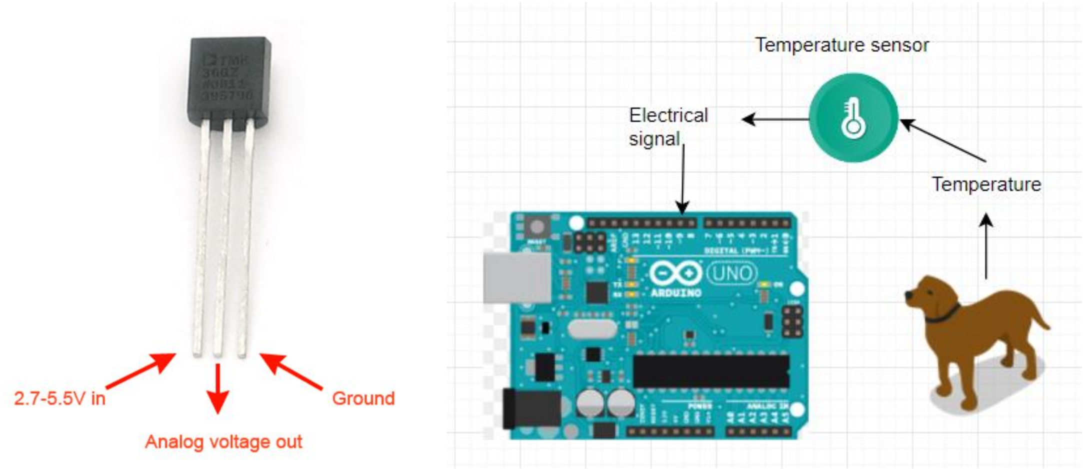


Figure 9 Temperature Sensor Connector Pin and Dog Temperature Sensor Example.

3.2.1 Research

There are many type of temperature sensor, our team Dog Smart Harness first choice is DS18B20 because this device has the water resistance and it also has the connection by long wire, so that our team can install it anywhere on the dog body. This device is extremely popular through the website and it commonly used on any MCU which including Arduino Uno. DS18B20 also can be easily integrated into the dog belt. The research not only stop at DS18B20 but also including the other two promising devices which is MCP9700 and MCP 9800. They all have great price as well as their performance.

3.2.2 DS18B20

DS18B20 is made by MAXIM company. It can give 9 bits to 12 bits Celsius temperature, also it has the range of -55°C to +125°C with error ±0.5°C over the range of -10°C to +85°C. This device come with long cable with connection with Vcc, ground, and DQ which make easy to integrate with the circuit and the dog’s belt. The voltage this device is around 3.0 - 5.5v, and the amp is 9 µA, so the total power is 27µW - 45µW. This product also sale on adafruit.com and it is support by the engineering team who live help customer every Wednesday night at 8pm ET. So that there is no worry about this device.

3.2.3 MCP9700

The Device MCP9700 also a temperature sensor which made by Microchip company. It has tiny analog temperature sensor according to the datasheet. MCP9700 have wide range temperature measurement range from -40°C to +125°C with error ±2°C same as for DS18B20 it has 3 pin for Vdd Vout and ground. The voltage consume for this device is from 2.3 - 5.5v, and the amp is 6µA which the total consumption power is 13.8 - 33 µW. The device is sale on www.sparkfun.com. This device is certified by RoHS which trusted and been tested. There are also 211 items in stock according to website; so that, our group will not worry about discontinue device until the project complete. However, our group don't make this device as our first choice because the device is actual small circuit which will difficult to wired and integrate into the dog belts.

3.2.4 MCP9800

The next device in our research is MCP9808 which is the next generation of MCP9700. The device is also manufacture by Microchip Inc. This device have the same range temperature with MCP9700 from -40°C to +125°C, but the accuracy is 4 times much smaller which is ±0.25°C, so this device is much more accurate compare to DS18B20 and MCP9700. The sensor also have another function called arlet which will let the circuit know when a certain temperature need to be watched. However, there is trade off for this device. The power consumption of this device is much larger. The voltage is 2.7 - 5.5v, and the amp during the operation is 200 µA. The total power of MCP9808 is 540 - 1100µW which is not an ideal for our design requirement that the low power consumption. There is another element about this device which make it not our choice for our group design the product is only bare circuit just like MCP9700 that hard to integrate into the dog belt to be able to read the temperature of the dog. The product sale on www.adafruit.com with 84 items in stock.

3.2.5 Temperature Sensor Parts Comparison

This table is created in order for our group to be able easy compare all the data. First of all, the price of MCP9700 and MCP9808 is much more cheaper compare to DS18B20. MCP9700 and MCP9808 is about 5 dollars compare to DS18B20 10 dollars. Next, the temperature range of DS18B20 is just a little larger about -55°C compare to -40°C for MCP9700 and MCP9808. MCP9808 is have the best out of three device which is ±0.25°C. However the power consumption of MCP9808 is largest compare to the other two devices. In addition,out of three device the DS18B20 having the waterproof option; meanwhile, the other two device do not have. To sum up DS18B20, MCP9700, and MCP9808 each of them have good and bad traits. They are all ideal to build in our circuit design.

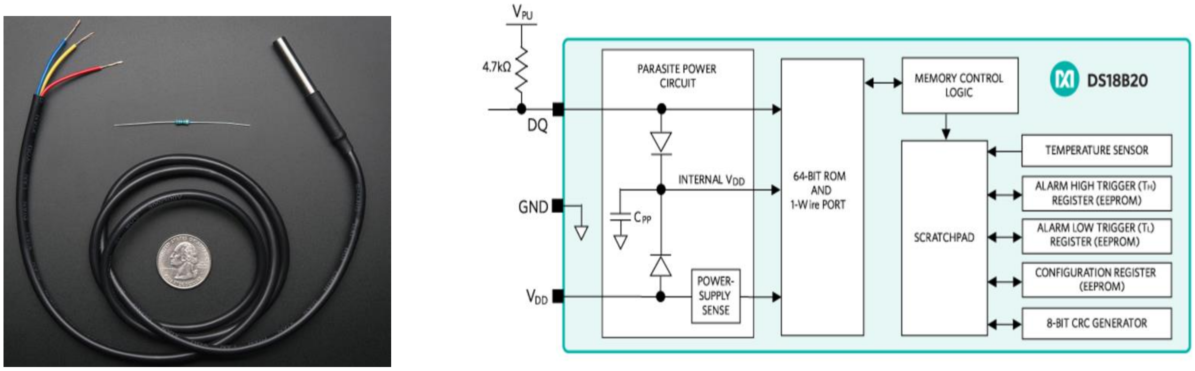
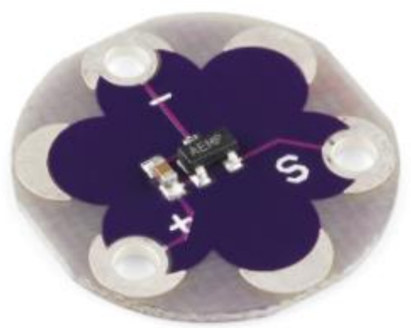


Figure 10 Actual Device and Circuit Diagram of DS18B20 (Selected Temperature Sensor)



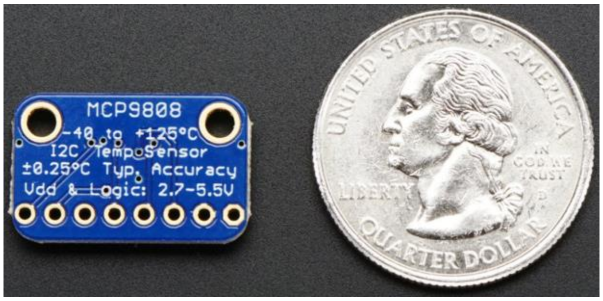


Figure 11 Actual Device MCP9700

Figure 12 Actual Device MCP9808.

	DS18B20	MCP9700	MCP9808
Price	$ 9.95	$ 4.95	$ 4.95
Usable temperature	-55°C to 125°C	-40°C to 125°C	-40°C to 125°C
Accuracy	±5°C	±2°C	±0.25°C
Cable	Yes	No	No
Voltage	3.3 – 5.5 V	2.3 – 5.5 V	2.7 – 5.5 V
Current	9 µA	6 µA	200 µA
Water Proof	yes	No	No

Size	Long cable	20mm x 0.8mm	21mm x 13mm

Table 2 Temperature Sensor Data Comparison.

3.2.6 Temperature Sensor Choice

Based on our group requirements and the data comparison between MCP9700, MCP9808, and DS18B20, we decided to choose DS18B20 by follow reasons. The first reason is the device DS18B20 have waterproof which match with our requirement. The design need to be able waterproof. The next reason is the device has the long cable which make easy for us to design and implement the sensor into the dog's belt. It can help us to locate the sensor to anywhere inside the dog's belt. The last reasons is the power consumption of this device is not too large which is from 27µW - 45µW, so that it is ideally we can use the battery as the power supply for our circuit.

3.3 Heartbeat Sensor

Heartbeat sensor came out since the Greek ancient time by placing the hand on the human's neck or wrist the physician be able to measure the heartbeat of a person. Until 1733, the Physician's Pulse Watch is made and it accurately measured according to George E. Billman. This is also the first idea for heartbeat sensor is make which later in nineteenth century. In 1977 by Seppo Säynäjäkangas invented the wireless heart rate monitor which has the sensor for the heartbeat module inside of his design.

According to elprocus.com Heartbeat sensor is " The basic heartbeat sensor consists of a light emitting diode and a detector like a light detecting resistor or a photodiode." when the tissue of the body such as finger or earlobe emitted, reflect the led light. The light is received by the light detector and based on the light reflection the detector output the electrical signal with the proportional to the heat beat rate. The signal is usually a DC signal relate to the blood volume of the body tissues. With those concept our group will use this theory to apply to a dog body in order to measure his or her heartbeat.

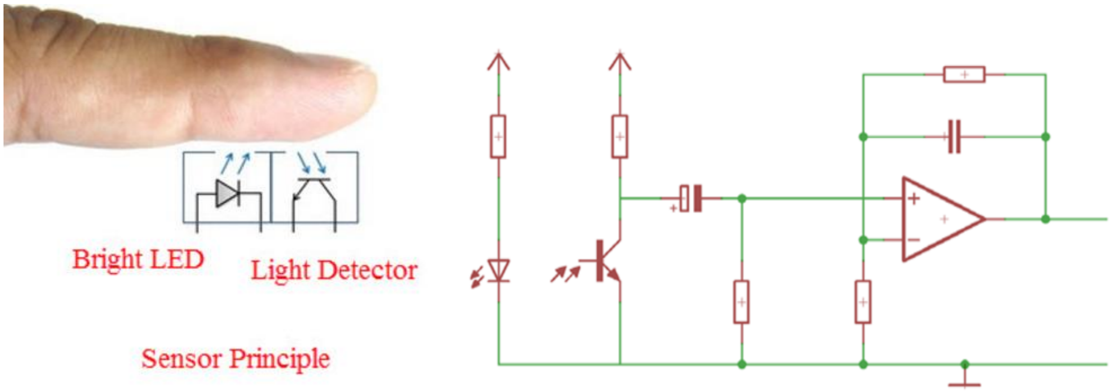


Figure 13 The Principle of Heartbeat Sensor and Its Circuit.

3.3.1 Heartbeat Sensor Research

Nowadays, there are many type of heartbeat sensor on the market. Some using the sound mean while the other using the optical, the design which our group choice is going the the optical heartbeat sensor. By using light emitted, our group will put those into the dog body which location where there is no or less dog’s fur in order to make the sensor work. There are 3 choice heartbeat sensor which our group want is, MAX 30102, SI1143, and APDS-9008, . The chip SEN-11573 is our first choice because of the price and the compatible with our choice MCU; meanwhile the other chips are also good candidate for our design.

3.3.2 MAX 30102

Max 30102 is the high sensitivity pulse oximeter which was make by Maxim Integrated in september 2015. It has low power heart rate moninot < 1mW and ultra low shutdown current at 0.7µA the dimension is 5.6mm x 3.3mm x 1.55mm with 14 pin optical module. The device can be operate at -40°C to +85°C, and with high sample rates. The power input is range from 1.8V to 5.0V. This device can be our candidate for our design because the power consumption is really low. However, our group don't choose this device because it is really advance chip for our project. In addition, the chip is only sale on ebay.com with $10. The website that our do not trusted.

3.3.3 SI1143

SI1143 is made by Silicon Labs. According to the description of mordendevice.com “the chip wasn’t exactly designed for a pulse sensor but it has all the necessary ingredients. These include variable LED control for three LEDs and two photodiode detectors.” However, our group still consider this device because it still be able to measure the heart pulse rate. the chip dimension is 32mm x 22mm which is bigger than MAX 30102, and it also compatible with our MCU. the power consumption is from 1.8V to 3.6V, the current is 9µA, the LED pulse is 25.6µs for every 800ms at 180mA, and it has 500nA standby current. The communicate of this sensor is 3.4 Mbps data rate, it can operate in the temperature range from -40°C to +85°C, and the device also have the power-on reset controller. To sum up, this is a really nice chip because of the specification above. However, our group do not choose this chip because it is really expensive which is about $20 on moderndevice.com.

3.3.4 SEN 11574

The heartbeat sensor SEN 11574 has the design based with the chip APDS 9008 light photonic inside. APDS is a miniature surface-mount ambient light photo sensor is made by Avago Technology. It is a light photo sensor is integrated into the circuit called SEN 11574. The device SEN 11574 was design hardware by Joel Murphy and Yury Gitman. It also widely used for the student project. The SEN 11574 is called Pulse Sensor Amped. It has the dimension 16mm x 3mm with the cable length 609mm (24”). The voltage input is 3V to 5V, and the current consumption is ~4mA at 5V. The operation temperature is from -40°C to +85°C.this device is best choice for our group because of its simple function for our project. Moreover, this device is compatible with our MCU, the price is only $5 at walmart.com.

3.3.5 Heartbeat Sensor Parts Comparison

This table is made which is make ourgroup easy to compare and choose between the device. Firstly, the voltage input of all the device is almost the same. The next thing is the current input of the sensor the MAX 30102 have the lowest current; meanwhile, the SEN 11574 have the highest input current 4mA. The dimension of all the device is almost the same but SI1143 is bigger compared to the other two. There only SEN 11574 have cable with 3 pin input GND, VCC, and signal. All the device operation temperature is from -40°C to +85°C. Lastly SEN 11574 is the cheapest.

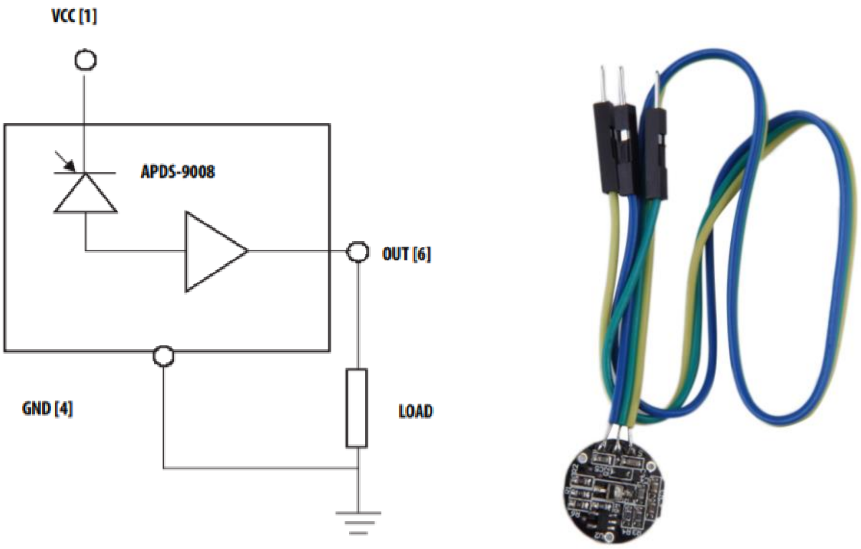
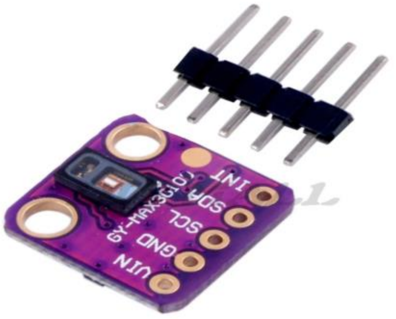


Figure 14 Photodiode of APDS 9008 and Actual Device SEN 11574 (selected)



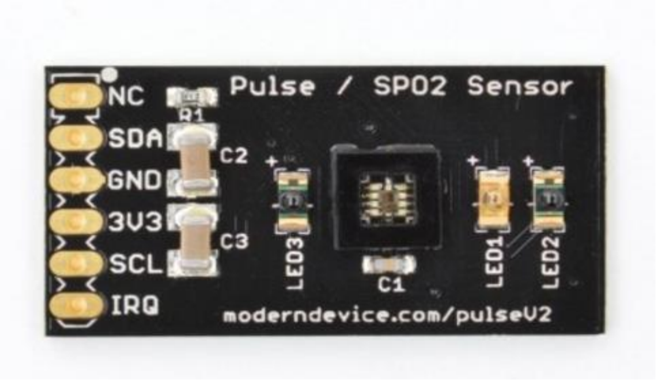


Figure 15 Device MAX 30102

Figure 16 SI 1134 Device.

	MAX 30102	SI 1143	SEN 11574
Voltage input	1.8V – 5.0V	1.8V – 3.6V	3V-5V

Current input	0.7 µA	9 µA	4mA
Dimension	5.6mm x 3.3mm	32mm x 22mm	16mm x 3mm
Cable	No	No	yes
Temp. operation	-40°C to +85°C	-40°C to +85°C	-40°C to +85°C
Price	$10	$20	$5

Table 3 Heartbeat Sensor Data Comparison.

3.3.6 Heartbeat Sensor Choice

Based on the comparison, our group make the priority choice is SEN 11574 by following reasons. The first reason is the SEN 11574 come with a long cable so that we can locate the heart any everywhere in the dog's belt. The second reason is the price. The SEN 11574 is only cost $5 which is much cheaper when compared to MAX 30102 and SI 1143. Also the SEN 11574 meet requirement of our project.

3.4 Bluetooth Communication

The Bluetooth was invented by Ericsson in 1994. Based on ABI research, it got adapted by five companies and form to bluetooth special interest group (SIG). it has been developed from 1.0 in 1998 to 5 generation in 2016. There are many application with bluetooth in our society such as speaker, computer, phone, car, and so on. With those advantage, our group will use the bluetooth communication get the measurement data from the sensor and send those information to our mobile application.

How the bluetooth device communication with the device? This question have been asked by our family and friends. The only thing they know about bluetooth is the Iphone or Android phone by simply pair with the car or speaker, and it will be able to play the music they want. However, in the reality, the bluetooth application is much more complex. According to Wiki, Bluetooth is another step of wireless technology, and it is used for data exchange by using the short radio wavelength UHF from 2.4 to 2.485GHz from a fixed mobile device to another. It is used to transfer data not only about the music data but also it used to transfer the signal from the sensor and received it back from the mobile application to control the system.

In our project, we using the bluetooth communication in order to sense the activity inside of the dog such as temperature, heartbeat, and accelerometer. We will be able to track the daily activity of the dog, based on those information which we obtain, the user (in the future) or we will know how healthy of their or our dog is, how much they work out a day. Moreover, the Vet doctor will be able to track down with the activity recorded on our database inside the mobile application.

3.4.1 Research.

With the advance of bluetooth developing, there are many chips sensor was made in the market. Our group is able to narrow down to three devices which is possible for our dog's harness project. The first device is Breakout LE BT BLE4.0 nRF8001 V1, the next one is BGM 113 V2, and the last one is RedBearLab BLE Nano nRF51822. In conclusion, the BLE Nano nRF51822 is our first choice for this project.

3.4.2 nRF8001 V1

nRF8001 V1 is manufactured by Adafruit Industries LLC. it can link to both IOS and Android operating system with (4.3+) device. It also simulating a UART device underneath of the circuit surface. It is call low energy because of 3V to 5Vdc and in input current is 100mA at 2.4 GHz with 10m range in order for connection to work. the dimension is 29mm x 28mm. However, “The nRF8001 library is not compatible with the Arduino Due at this time” according to the datasheet. The unit cost of this device is $20 on digikey.com. Due to the price and the library not compatible with Arduino, our group will not make this as our first choice for this project..

3.4.3 BGM 113

BGM 113 Blue Gecko Bluetooth is made by Silicon Labs. It has low energy application just like nRF8001. Moreover, this device has 32-bit ARM Cortex M4 core at 38.4 MHz with flash memory 256kB and ram 32kB. It has many application such as IoT sensor, End device, commercial, retail, health, wellness, industrial, home, building automation, smart phone, tablet, and PC accessories according to datasheet. It can connect with IOS and android device with 4.2 and up. BGM 113 has wide operating range power from 1.85V to 3.8V single power supply or 2.4V to 3.8V when using DC-DC, the current consumption is 16.9mA for TX and RX at 2.4GHz, and its dimension is 15.73 mm x 9.15 mm. The range of operation is 10m. The temperature range is -40°C to +85°C. However, this device is not compatible with Arduino. Lastly the cost for this device about $6 on mouser.com. To sum up, this device can not be good choice for our project.

3.4.4 nRF51822

nRF51822 is a bluetooth device which was manufactured by Nordic Semiconductor. It uses ARM Cortex MO 32 bit processor with 128kB flash memory and 32kB RAM. this device is used for computer I/O, interactive entertainment, beacons, personal area network, and remote devices. The voltage input range from 1.8V - 3.6V, and the current for TX and RX is 17.7 mA peak with DC/DC for 2.4Ghz communication. The device is using Bluetooth 4.0 Low Energy. It support for many device such as IOS 7/8, Android 4.3, and windows phone 8.1. The dimension of nRF51822 is 21mm x 18.5mm. The temperature operating is from -25°C to +75°C. The distance of this device is 10m. Finally, robotshop.com sale this device for $7.

3.4.5 SH-HC-08

According to Module Forces blogspot, SH-HC-08 was made by Module Forces a chinese company who only focus bluetooth module. The SH-HC-08 is the generation class 2 Bluetooth 4.0 ble model. It using the CC2541 chip to support the AT command. The size of this bluetooth is 26.7mm x 13mm x 2mm. It can perform under range temperature from 40°C to +85°C which is similar to BGM 113. It has the operate frequency is 2.4 Ghz. Unlike SH-HC-Sx series from previous model. This Bluetooth component is using the standby current ~ 1.5 mA. It can run the power input is 3.3 Vcc. the strong point about this bluetooth module is the dual communication with the application software which is needed for our project. After research we decided to use this bluetooth for our dog's harness project.

3.4.6 Bluetooth Communication Parts Comparison

Those bluetooth device is really great for our projects. For example, the nRF8001 and BGM 113 is used by all the previous groups in senior design at UCF and the other school, for their project, and the existing code been out there. However, our group will not use it for our project because of non challenging for our project. As a result, we make a decision to choose nRF51822 as a challenge for our project in both embedded coding installation inside the PCB. we will do the comparison to be clear about our selection.

First of all the voltage input, out of all 3 device the nRF51822 have the lowest voltage require. Secondly, the BGM 113 only use 16.9mm; however nRF51822 is only different by 0.8mA. The BGM 113's temperature is almost the same as nRF51822 based on both datasheet. BGM have smallest dimension with 15.73mm x 9.15mm. They all also have the same range of distance, and the BGM have higher flash memory compare to nRF51822, but they have same RAM which is 32kB. The nRF8001 has bluetooth version 4.3+, BGM 113 has 4.2+, and nRF51822 has 4.0+. All of them have 2.4GHz operational. Moreover, the nRF series are compatible with Arduino; meanwhile, BGM 113 is not. The price BGM should be the cheapest since it only cost $6.

Figure 17 Breakout LE BT BLE4.0 nRF8001 V1

Figure 18 BGM 113

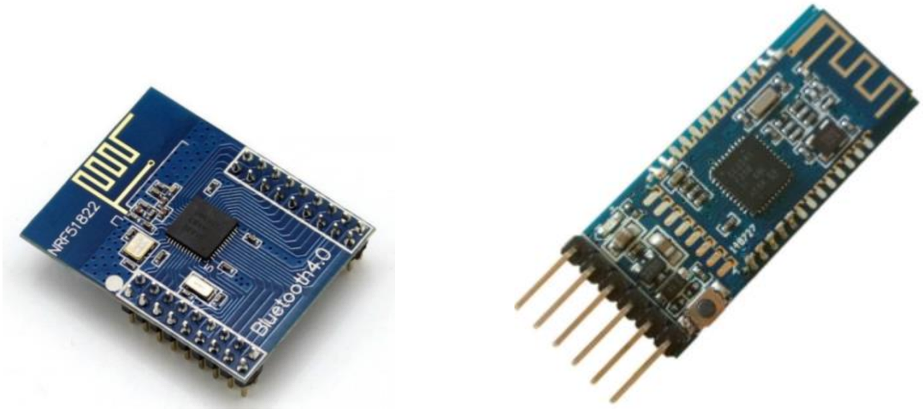


Figure 19 NRF51822 and SH-HC-08 (selected component)

	nRF8001	BGM 113	nRF51822	SH-HC-08
Voltage input	3V – 5V	1.85V – 3.8V	1.8V – 3.6V	3.3V - 5V
Current TX+RX	100mA	16.9mA	17.7mA	9mA
Temp. operation	unknown	-40°C to +85°C	-25°C to +75°C	40°C to +85°C
Dimension	29mm x 28mm	15.73mm x 9.15mm	21mm x 18.5mm	26.7mm x 13mm x 2mm
Distance	10m	10m	10m	10m
Flash memory	Unknown	256kB	128kB	Unknown
RAM	Unknown	32kB	32kB	Unknow
Bluetooth version	4.3+	4.2+	4.0+	4.0
Radio Wave	2.4GHz	2.4Ghz	2.4Ghz	2.4Ghz
MCU(Arduino) compatible	Yes	No	yes	yes
Price	$20	$6	$7	$8

Table 4 Bluetooth Data Comparison.

3.4.7 Bluetooth Communication Selection

After all the comparision, our group choose SH-HC-08 because it's compatible with our MCU Arduino. Even though, the SH-HC-08 have bigger dimension compared to the other bluetooth module, and the price by $1 dollar. However this bluetooth have longest range of distance 110m based on the database for this reason it would be for our project. Also, The SH-HC-08 should be the good choice because it has 4.0 bluetooth version compatible, and we can use it to communicate with our mobile phone application. The test will be conduct with the coding provide their company from github.com. This SH-HC-08 is fairly new product so that we won't worry about the discontinue for the senior design 2.

3.5 Accelerometer Module Research

The Accelerometer is a module that can detect motion in 3D space. In the real world, things can move in any direction on an x, y, and z imaginary grid. For our purposes we may only need 2D motion as the accelerometer will be used mainly as a pedometer or step counter, however most of the modules are already designed for 3D motion so these features will be present but most likely unused. This module can measure the distance and speed at which the dog will move and through the software we can determine the number of steps that the dog has taken. What is defined as a “step” will be a measure of the gait of the dog itself. This can be determined at a later time but may require a measurement of the length of the dog and other factors.

Factors for determining which module will be chosen depend upon cost, precision, voltage, compatibility with Arduino, and size. With the exception of the FLORA Accelerometer/Compass Sensor, most of the step counters are between $5-$8 so the costs for the first part are expected to be low.

3.5.1 Adafruit Triple-Axis Accelerometer MMA8451

The Adafruit Triple-Axis Accelerometer MMA8451 comes at a price of $7.95 and detects motion, tilt, and orientation because it is a digital accelerometer. The design allows for real-time activity analysis which is necessary to be used as a step counter. This design also comes with a 14-bit ADC, which allows it to be very precise. The voltage range requirement is between 3 V and 5V which will be a work well with our voltage specifications. This particular module is well suited to be used in conjunction with an Arduino microcontroller which may make some of our tasks a little easier to implement. The Adafruit Triple-Axis Accelerometer could be a good choice for our Smart Harness but may have extra unnecessary attributes.

Positive Factors

- Compatible with Arduino microcontrollers
- Compact, and lightweight
- I^{2}C Communication Interface
- Small size (3 mm x 3 mm x 1 mm)
- Operates at a low voltage range (3 V – 5 V)
- 14-bit ADC for higher precision
- Low current consumption (6 µA - 165 µA)
- Will operate with a 1.95 V – 3.6 V supply voltage

Negative Factors

- Moderate cost accelerometer ($7.95)
- May have excessive functionality
- 16-bit option is available

3.5.2 GY-27 3-Axis Compass Accelerometer

The GY-27 3-Axis Compass Accelerometer comes at a price of $5.82 and detects motion, tilt, and orientation because it is a digital accelerometer. The design allows for real-time activity analysis which is necessary to be used as a step counter. The voltage range requirement is between 3V and 5V which will be a work well with our voltage specifications. This particular module is well suited to be used in conjunction with an Arduino microcontroller which may make some of our tasks a little easier to implement. The GY-27 3-Axis Compass Accelerometer may not be a good fit for our project as it is larger than other options and may only work for specific functions and may not be able to be purposed as a step counter.

Positive Factors

- Low cost ($5.82)
- Arduino compatible
- Low weight
- Operates at a low voltage range (3 V – 5 V)

Negative Factors

- Larger size than other options (3.2 cm x 1.5 cm x 1 cm)
- Compass features unnecessary

3.5.3 FLORA Accelerometer/Compass Sensor LSM303

The FLORA Accelerometer/Compass Sensor comes at a price of $14.95 and detects motion, tilt, and orientation because it is a digital accelerometer. The design allows for real-time activity analysis which is necessary to be used as a step counter. The voltage range requirement is between 2.16 V and 3.6 V which will be a work well with our voltage specifications. This particular module is well suited to be used in conjunction with an Arduino microcontroller which may make some of our tasks a little easier to implement. The FLORA Accelerometer/Compass Sensor could be a good choice for our Smart Harness but may have extra unnecessary attributes.

Positive Factors

- Compatible with Arduino microcontrollers
- Compact, and lightweight
- I^{2}C Communication Interface
- Small size (14 mm diameter and 1.8 mm thick)
- Will operate with a 2.16 V – 3.6 V supply voltage

Negative Factors

- Very high cost accelerometer ($14.95)
- Compass features unnecessary

3.5.4 ASXL345-BB 3-Axis Accelerometer Module

The ASXL345-BB 3-Axis Accelerometer comes at a price of $5.01 and detects motion, tilt, and orientation because it is a digital accelerometer. The design allows for real-time activity analysis which is necessary to be used as a step counter. The voltage range requirement is between 2.0 V and 3.6V which will be a work well with our voltage specifications. This particular module is well suited to be used in conjunction with an Arduino microcontroller which may make some of our tasks a little easier to implement. The FLORA Accelerometer/Compass Sensor could be a good choice for our Smart Harness but may have extra unnecessary attributes.

Positive Factors

- Compatible with Arduino microcontrollers
- I^{2}C Communication Interface
- 13-bit for higher precision
- Low current consumption (0.1 μA - 23 μA)
- Moderate size (25mm x 25mm)
- Will operate with a 2.0 V – 3.6 V supply voltage
- Low cost ($5.01)

Negative Factors

- Compass features unnecessary

3.5.5 Accelerometer Module Comparison

Choosing the best accelerometer for our design was a difficult process. Many of the different modules researched had similar voltages and other than the FLORA LSM303 were all close in price. Size played a factor as well in choosing this module as they were all relatively small in size, with the Adafruit Triple-Axis accelerometer being the smallest at 3mm x 3mm.

The number of bits each module had to use will help determine the precision or accuracy of the pedometer when it is actually completed. The values given for three of them were 13-bit, 14-bit, and 16-bit. The GY-27 datasheet was unavailable/incomplete when researching this part and as a result disqualified itself from our project.

We concluded that we should find a middle ground when deciding which module to proceed with. The Adafruit MMA8451 is the smallest, has a good voltage range, good precision using 14-bits, and is still relatively cheap and may prove to be a good option for our project.

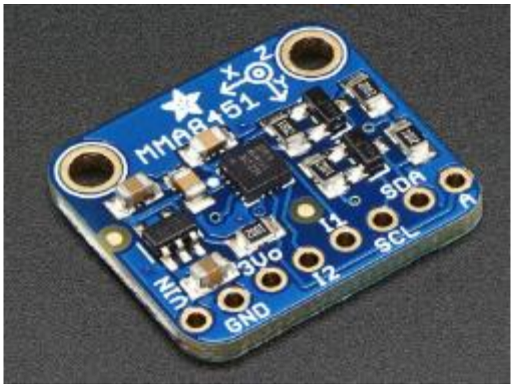


Figure 20 Adafruit Triple-Axis Accelerometer Accelerometer

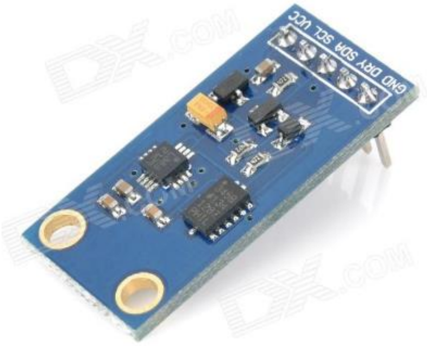


Figure 21 GY-27 3-Axis Comp.

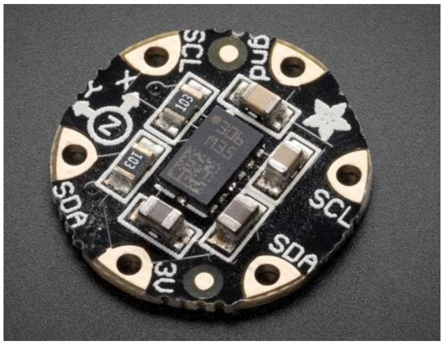


Figure 22 FLORA Acc. Sensor LSM303

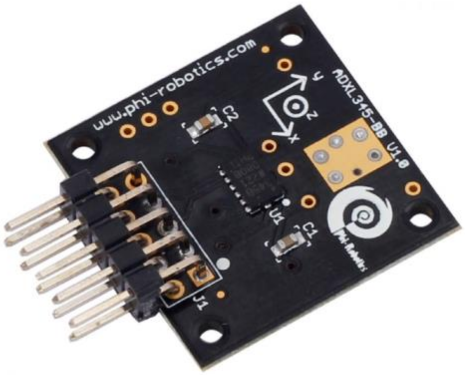


Figure 23 ASXL345-BB 3-Axis Accelerometer

	Adafruit MMA8451	GY-27	FLORA LSM303	ADXL345-BB
Size	3mm x 3mm (small)	3.2 cm x 1.5 cm (large)	14 mm diameter (small)	25mm x 25mm (moderate)
Cost	$7.95	$5.82	$14.95	$5.01
Precision	14-bit (High)	Unknown (datasheet unavailable)	16-bit (Very High)	13-bit (Moderate)
Supply Voltage	1.95 V - 3.6V	3 V - 5 V	2.16 V - 3.6 V	2.0 V - 3.6 V

Table 5 Accelerometer Unit Comparison

3.5.6 Accelerometer Module Selection

For the purposes of our project we have selected the Adafruit Triple-Axis Accelerometer MMA8451 module. Cost is a major factor here as well as size. Many of the accelerometers have very similar functionality and precision as well as voltage ranges that were fairly similar. The GY-27 is a very poor fit for the project due its larger size and the lack of an easily accessible datasheet made it more likely to encounter problems in the future. The group chose this module for many reasons. While not the cheapest, it was the the best fit due to its higher precision coupled with a moderate cost as well as meeting our requirements. It also has abundant information sources available due to its compatibility with Arduino.

3.6 Battery Research

Nowadays, everybody knows about the definition of battery "It is a container consisting of one or more cells, in which chemical energy is converted into electricity and used as a source of power" on google definition. The battery history has been started from late seventeenth century according to energizer.com. An Italian physicist started with his first "Voltaic pile" in 1798. It is a stack copper zinc disks that got separated by the cardboard disk contain with salt and acid solution. Later It was developed into many different types of batteries. At the present, the battery is used for many purpose such as car, mobile, mp3, light, speaker, and so on.

The battery carry two charge plus and minus either place opposite side of one another, or it was put on the same side such as 9V, car, mobile, or tablet battery. The battery hold the electrolyte charge and transfer electron into the circuit when the loop is closed according to Northwestern University. It flow from the negative charge to the positive charge.

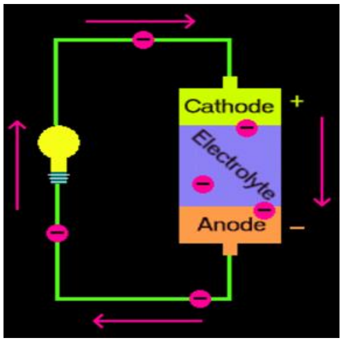


Figure 24 How a Battery Works

There are two principal of battery like the rechargeable and the non rechargeable. Those fall into the different type of battery such as Nickel Cadmium (NiCd), Nickel-Metal Hydride (NiMH), the Lead Acid, lithium Ion, and Lithium Polymer batteries. As the big selections for the battery, our group 15 decides to use non recharge battery Alkaline for our dog harness project.

According to Fred, The Alkaline battery is divided into different type of volts such as D, C, AA, AAA, 9 Volts. Depending how much voltage on our MCU, sensors, and bluetooth, we will decide how we will combine all the battery together in order to supply our MCU.

3.6.1 D Type

Battery D cell is also called ICE 20 according to Wikipedia. It has a cylinder shape, and the positive is on the top and negative is at the bottom. D battery type is used in high current consume device such as flashlight, radio receiver, boomboxes, mega phone and so on. In 2007 D cell was really popular sale in US. it was introduced by National Carbon Company in 1898, and it even developed before in world war II. In Navy, the military referred it as C cell battery aka (BA-42). It weight 135g.

This Alkaline battery has 1.5 V and also depend on type of battery so it can be recharged, and the current it can hold is 12000-18000 mAh. In walmart, it was sale for $12.96 for pack of 8, so the average for one battery is $1.62.

3.6.2 C Type

Not same as D type battery. According to Wikipedia.com C battery is slightly smaller than D type, and it was also call R14 battery. However, the C type battery is the medium drain current compare to D type. It has dimensions “50 millimetres (1.97 in) length and 26.2 millimetres (1.03 in) diameter.” it was made in 1920s. C type battery is using based on the chemistry which have different design compare to D type which is discharge condition. The reason it was made is mainly using for flash light application as well as toy, and music instruments. It weight 65g.

C type battery has 1.5 V, and depend on the type of batteries so it can be rechargeable. Not like D type of battery C type is can only hold up to 8000 mAh. In walmart, it was sale for 7.47 for pack of 4, so the average for one battery is $1.86.

3.6.3 AA Type

AA battery has a definition as double A (Mignon) in term of French “dainty.” it is the most popular battery on the market right now. According to Wikipedia.com, IEC 60086 call it size R6, and in British, it is called “double A battery” as in common term. Same as D and C type, the AA has top as positive charge and bottom have the negative charge. It was introduced in 1907, and was accepted by American National Standards Institute (ANSI) in 1947. The battery have much smaller dimension compared to D and C type which is "49.2–50.5 mm (1.94–1.99 in) in length, including the button terminal—and 13.5–14.5 mm (0.53–0.57 in) in diameter. The positive terminal button should be a minimum 1 mm high and a maximum 5.5 mm in diameter, the flat negative terminal should be a minimum diameter of 7 mm" according to Wiki. AA type is used for many application such as small flashlight, and portable electronic devices. It weight 24g.

The alkaline AA type battery has 1.5V, and it has 1800-2600mAh which is much smaller compare to the other two type battery. However due the standard and the size of this battery is much a good option to our project. On the market, this battery was cost around $9.88 for a pack 24 which is cost only $0.42 per battery.

3.6.4 AAA Type

AAA cell type battery is much smaller compare to the other battery above. It is used in low drain portable electronic device such as remote television, MP3 player, or digital camera. According to Wikipedia.com, the battery size is "10.5mm in diameter and 44.5 mm in length, including the positive terminal button, which is a minimum 0.8 mm high. The positive terminal has a maximum diameter of 3.8 mm; the flat negative terminal has a minimum diameter of 4.3 mm." ANSI called it as C18.1 or 24. This type of small battery only weight around 11.5g. Due to the small design this battery has been used to replace AA type batteries in many application such as remote control.

This alkaline AAA battery has 1.25 volt, and the power capacity is 860-1200mAh. It should be the smallest battery in our project research. On the market, AAA battery cost $13.97 for pack of 24 batteries which is $0.58 per unit. This battery might be not the best for our project because the power capacity is not really that great for us to use in our project.

3.6.5 9 Volt Type PP3

The last battery type in our research is 9 volt battery. Introducing in 1954 according wikipedia.com, the 9 volt battery is combined between carbon-zinc and alkaline with mainly lithium iron disulfide. The interesting thing about this battery is that it is combined with six individual 1.5V LR16 cells and enclose with a moisture-resistant. LR16 cells are slightly smaller than LR8D425 AAAA cells. The 9 volt battery have different design compared to the rest of regular battery is that it has two terminal positive and negative at the same side. Because of this design, this type of battery easy generating heat possibly catch on fire. The advantage of this battery is it has high voltage which make it easy to used in the radio transmitter or receiver. On wiki stated that "The Eveready company claims that it introduced this battery type in 1956." it say also say that the battery can be last for 10 years under no using condition. The dimension of this battery is "height 48.5 mm, width 26.5 mm, depth 17.5 mm." This battery weight 45g.

The 9 volt battery has 9 volt like its name. The power capacity is 550mAh. It cost $16.79 for pack of 12 batteries which is $1.40 for each battery. Due to the cost and the low in power capacity, this 9 volt battery type is not our first choice for this project.

3.6.6 Battery Comparison

After the research in some battery D, C, AA, AAA, 9 volt, those batteries have their own specific and the power capacity. Each of them is specifically used for different application. However, with the table below our group make a table to show the specification of each batteries in order, we can make the appropriate choice the battery for this project.

	D type	C type	AA type	AAA type	9 volt
Voltage	1.5V	1.5V	1.5V	1.25V	9V
Power capacity (mAh)	12000-18000	3800	1800-2600	860-1200	550
Size (HxD) in mm	61.5 x 33.2	50 x 26.2	50 x 14.5	44.5x10.5	48.5x26.5
Weight (g)	135	65	23	11.5	45
Price (per unit)	$1.62	$1.86	$0.42	$0.58	$1.4

Table 6 D, C, AA, AAA, 9 Volt Batteries Comparison.

Figure 25 Battery list AAA, AA, C, D, 9V

3.6.7 Battery Choice

Based on the design which we are supposed to used for our circuit Arduino. Our group decided to using six AA types batteries instead of the other batteries due to following reason. First of all, Arduino need to have the power supply from the battery between 7 - 12 voltages based on the datasheet. The C and D type are to big and heavy which is make we eliminate those two and then for AAA is too small and it only have 1.25V according to the table so that it also is not a good option for us to using AAA. the only left is AA and 9 Volt. the second reason is the energy consumption of the circuit. The more higher joules is the better for our circuit. According James Levis from baldengineer.com, the energy from 9V battery is 19440 J compare to 6 AA is 89100 J for 3600s. Those number are used based on Energy formula.

E = V x I x t

Where V is voltage, I is amps, and t is time in second. The time for our demo and presentation is 10 mins which is 600s, the voltage require for our circuit is 9V, and the current need is 50mA according to the data sheet. Plus the current from bluetooth, and sensor to the total current for our circuit 50mA (arduino) + 0.009mA (temperature sensor) + 4mA (heartbeat sensor) + 17.7mA(bluetooth) + 0.165mA (accelerometer) = 71.88mA total current drain by our circuit. That mean during our presentation we need at least E = 388.16 J.

Meanwhile the battery 9 volt have 19440 J for 1 hour, and the 6 AA is 89100 J. As we know the more we draw out the battery the Voltage get reduce smaller as well as for the energy. So that the option for our group is go with the biggest energy for our circuit so that we can have the circuit continues running during our presentation.

In this case for the six AA battery type we will doing the series application so that our power supply will have the total 9 voltage, and the current for our circuit is going to be 1800 - 2600 mAh. The battery will be place near the circuit where it will have the shortest distance. Also it won't be harm for the dog during the operation. The figure below is our design for the six AA battery and the part we get from amazon.

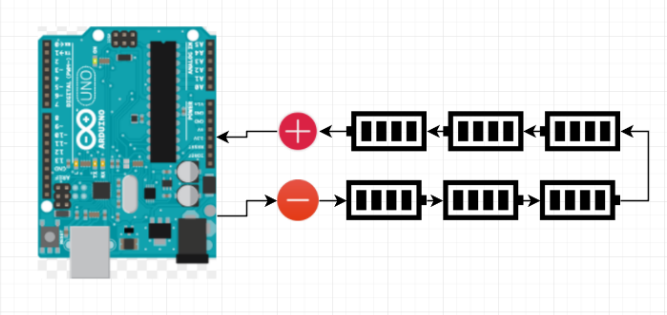


Figure 26 Design Battery in Series. **Figure 27** The Actual Battery Case

3.7 Software Research.

Along with the development of Iphone, Windows phone, and android phone, the software are developed and installed inside of the smartphone in order to control the hardware aka devices. from afar such as controlling the light by turn on and off, watching the home security camera, changing the house’s temperature with smart phone and so on.

Software is a program which the instruction was stored inside of a memory and it will be executed by the processor conditionally. According to wikipedia, it was invented by Ada Lovelace born in 1815. He was the first person who proposed the programmable computer which is called Analytical

Engine. With the developing of industrial, the program is developed into theory by Alan Turing in late 1940s and become more popular in 1980s, and in later years it become so much popular until nowadays. With the developing software, the human can do so many thing such as banking, credit cards, gaming, and so on. Not long after the smartphone was invented, the software become a useful tool for us to make the applications. In our project we are going to make a application where it will connect with our circuit in order to monitor the dog's condition.

3.7.1 The Reason development for a personal device

A major functionality and feature of our class project is our application that will link directly to the harness for ease of use and increase in usability of the actual product. The smart-harness for the canine is a great idea in the grand scheme of things but would actually render useless if we could not access the valuable information that the harness is supposed to provide. So the question that comes to mind now is how do we display this information to the consumer. As a variety of options come to mind, we could display the information to the consumer via lcd screen on the actual harness, we can record the information and display it later via website, we can record information to a usb drive and access all the information at a later time, or we can link an application to the hardware for live interaction. There are many factors that we have to consider when picking our final solution. When creating a product for a client we need to consider ease of usability, usefulness, durability, longevity, and user to product interaction. Let's take wireless charging capabilities for an example. In terms of the main points we are trying to hit, the wireless charger hits all of them. The wireless charger and the wireless charging capabilities was create to make charging hassle free by taking out the factor that you have to connect a charger every time you charge. It makes it easy and as simple as placing your phone down on the table. Its durable and should last long with the little to no adjustments and configuration you have to do with the charger. It is small and portable and extremely useful. So following that example we took these main points into consideration. Problems with putting an LCD screen on the harness would lower many of our main points including durability, usability and longevity. We would have to consider conserving the integrity of the screen as the dog may be rolling around in the dirt, jumping in water, and just regular wear and tear from being on the dog at all times. Also, to view information that the harness provides, the user would have to get the canine to behave and stay in one place while viewing the data or just simply removing the harness every time they needed to view data and then put it back on. Another option was to store the data somehow whether that be on the harness itself and access it later via website or simply store all the data on a USB drive so that the user could access the data at a later time. Again we run into the issue where there is a constraint for the client as to when and where they can view the data which takes away from the usability of the product as a whole. So the final conclusion we came to was to actually have the harness link to an application on the user's phone which would then the user can interact with the harness in live time. This makes it usable for the user as they can record and view data at any time without having any constraints, it makes it durable because it is one less mechanism that we would have to add to the harness, and it makes the user to product interaction seamless and easy. Another benefit that we thought would come from choosing an application is that not only can the user access all records that the harness has recorded but the information can be stored in a way that the user can share the information easily. For example, they can send information to their vet at any given time, they can share and compare with friends and see whose pet is healthier and make it a fun interaction like how Fitbit does with their step counter. So as to hitting all of our main core pillars for the user, an application for the user on a personal device was the best route to go. Now is the hard part which is deciding what platform we are going to support, what language we are going to use to develop the app, and so on and so forth. This will be discussed in the next section.

One of the requirements and decisions of this class project is to decide what platform we are going to support. As a time constraint and as a prototype for the product, we need to decide what platform we are going to support because we obviously do not have the time nor the resources to provide support for all platforms. Some options that we have gathered are Android, Apple, and Windows. The first and most obvious constraint we face when choosing a platform is who can use the application if we select this specific platform. So let us start with Android.

3.7.1.1 Android

From 2014 to 2016, the number of android users in the U.S. grew from 87.7 million people to 107.7 million people. The numbers that ran for Canada for last year's quarter was 352 million Android based phones sold. Android also makes up 76.4% of the Chinese mobile phone market. In terms of how easily accessible Android is, Android is available for 500 carriers worldwide. These numbers are very important in terms scalability. Let's say if we were to stick with Android and Android only, knowing these numbers not only helps us in the short term but it can give us a good gauge on the future. Another statistic we can look as is the market for apps on the Android platform. In 2015 the average Android phone generated about $5.70 for the Google Play Store. In 2016 there was an estimate of 224.8 billion app downloads from the google play store and in 2017 it is projected at a whopping 270 billion. If we look at these numbers we can see the growth of the market and how many people are actually engaging in the Play Store and downloading apps (whether it is free or paid apps) rather than just using their phones for its basic functionality. We can also see from the growth of the market which is now at 2.8 million apps which means the Play Store is growing at a very fast pace and it is not going down anytime soon. This is a good sign if we are to pick android as the platform for our app. A major benefit I also see from using android is that the upfront cost to get your app on the app store is a mere $25 dollars. This is a very reasonable number when we have to calculate it with everything else we have to take into account when building our product. Also, when we look at this from a developer standpoint, it is said that developing an android app cost significantly higher than most apps because of the difficulty and the time involved in creating one. Statistics show that android costs 2-3 times as much as iOS and this is due to many factors such as less sophisticated tools, generally more cumbersome APIs, fewer exposed advanced features, enormous QA issues brought on by fragmentation, etc. So in the app world, the general rule of thumb is to have two Android app developers for every Apple app developer. This is definitely one of the most important things to take into consideration due to the fact that we are facing both a time and resource constraint.

3.7.1.2 APPLE

For Apple, there are 85.8 million users in the U.S. as of 2016 and it grew from a significantly smaller number of 44.53 million in 2012. So as of 2016, apple makes up a staggering 42.9% of the U.S. smartphone market. It also makes up 8.2% of the Chinese mobile phone market. One thing that you have to take into consideration is the iPads that they sell which is a whopping 360 million where 26.04 million of them were sold in 2015 and the number is still rising. That is a lot of different devices that people can still access the app store for. Another considerable statistic is the average price of apps people spend on which is $35 dollars which is very high when comparing to a cost to a single app which is only in the lower single digits. These numbers are very impressive but one thing we have keep in mind is that it cost more money to use the Apple development tool which is 99 dollars annual fee for the Apple Developer Program. Even though the price of the development tool is almost doubled that of Android and even more than Microsoft, there are many other statistics that we need to take a look at and understand. Though android has just as much of the market as Apple does (varying a few percentages) one thing we have to understand is that the android platform is spread across so many phones, so many companies, so many tiers for smart phone where iOS is fed through one pipeline, Apple. When we look at the statistics we are not accounting for the lower end phones that the android platform is on. These $50 - $100 phones are sold worldwide in the millions but that does not mean that these low end users even access the app store! Most of these phones are the basic phone calls, SMS messaging and no data plan phones. Studies show that revenue from the Apple app store almost doubled that of Google Play store. What does that mean? It means that more apple users are buy apps than Android users and that is because the android platform is diluted through many channels. So if we look at the market now and see what other people are doing, they are choosing iOS first leaving Android as an afterthought. Why? Not only is it because development is cheaper and easier, but also because money for in-app purchases and advertising is overwhelmingly skewed toward iOS. A study a few years back showed that Facebook ads shows ads were 1,790% more profitable on iOS. So far it looks like Apple’s iOS App store is the way to go.

3.7.1.3 Windows

Windows platform is quite unimpressive when it comes to their sales and how many users there are in the world that actually have their phones and user their platforms. It is really bad when your number of users worldwide becomes a rounding error of 0.0%. In the middle of 2016 they sold nearly 2.4 Windows phones which was a decrease from back in 2015. These numbers are miniscule compared to the giants that run this industry. The registration cost is about 50 dollars or 100 annually to get your app on the store and keep the membership which is very expensive when you analyze how many people actually use the phone and how many apps are available on the market. Windows app store host around 700,000 apps where Apple and Android sport an attractive number of 2.2 million and 2.8 million respectively. Window's statistics are far from subpar numbers but it was worth a mention. It is important to always explore all avenues and leave no stone unturned especially when starting your own business but unfortunately there was just nothing under this stone.

3.7.2 Tools For Developing Software

Another variable that we have to take into consideration is the environment that we will be developing in and the IDE we need to use. This makes things extremely complex for a app developer now days because instead of using one development tool for all platforms, we have to learn different languages and different tools to develop on each specific platform. In this instance android developers would have to use any IDE such as Eclipse, iTellij, or Android Studios. For Apple developers, the IDE's that are provided are AppCode and Xcode. So the downfall to this is that there is no one tool for a developer to learn to build any app but would have to learn different programming languages, install a number of different IDEs, get familiar with different APIs and so on and so forth. Another aspect of choosing an environment to develop in is how well is the documentation assuming there is any. Is the documentation well written and thought out so that we can just pick it up and start developing? How well is the support assuming there is any at all? If we run into any issues like malfunction, crashes, or even just where to find a specific setting, can we find solutions online in a forum or can we call in and speak to a specialist? So, the premise of this section is to narrow down the most effective and efficient solution for our development as we have many obstacles to overcome and choosing the right tools will be crucial on how we overcome our obstacles.

3.7.2.1 Android Studio vs Eclipse vs IntelliJ

From the previous information provide you can easily conclude that more that 76% of the Smartphone market (including Samsung, LG, and HTC just to name a few) have Android as their main operating system. So naturally it attracted many people to app development and choosing Android as the perfect place to start. Android Studios, Eclipse and IntelliJ all provide the ability to build an android app but which one to use is the real question. Sure, we can close our eyes and pick one at random but that would not be very wise as we know some programs perform significantly better than others and hashing out those details in the beginning may save us a lot of time and headache.

3.7.2.1.1 Eclipse

As common courtesy, since Eclipse is the oldest, we'll start with this integrated development environment. Eclipse is a strong and mature development tool that has many feature and capabilities. This integrated development environment is used mostly to develop Java applications but can support many other languages such as Ada, ABAP, C C++, COBOL, D, Fortran, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby, Rust, Scala, Clojure, Groovy, Scheme, and Erlang. IBM was the main front runner of creating this tool and had invested nearly $40 million by 2001 where they formed with a board of stewards to further develop Eclipse as an open-source software. The name "Eclipse" may be a wordplay on their competitor "Sun Microsystems" and even though the CTO of IBM denies that ever being the reasoning behind the name, there is much speculation. To get the part we are most interested in is can it develop android apps; it can. ADT also knows as Android Development Tools is a Google-provided plug-in for the Eclipse IDE that is designed to provide an integrated environment in which to build android applications. Android Development Tools extends the abilities of Eclipse to let developers setup new Android projects, create an application UI, add packages based on the Android Framework API, debug their applications using the Android SDK tools, and export signed (or unsigned) .apk files in order to distribute their application. Some positives to using Eclipse are beneficial aspect such as it being free, it has a large and active community, which has resulted In a wide variety of plugins in turn makes it very customizable, Android Development Tools provides GUI access to many of the command line SDK, it has full support for both Java and XML so you do not have to download yet another application, and because Eclipse is based on SWT, it uses the native font rendering and thus looks better than other integrated development tools one some Linux systems where the Java font rendering is not optimal. Though there are a lot of positives, there are some negatives to consider. For example, even though the plugins for Eclipse are plentiful, it does not mean that they are reliable. A lot of them are not maintained, bug fixes can be slow, and you may need to download plugins from multiples sources. This can lead to disorganization and application crashes and bugs with you resolutions. This is definitely concerning for an Android application developer because even though the capabilities are there, Google themselves recommend moving away from Eclipse since they no longer support Android Development Tool and are fully committed to Android Studios and other IDE's which brings us to the next chain of topic.

3.7.2.1.2 IntelliJ IDEA

Next on the list of possible integrated development environment for developing software is IntelliJ IDEA. This IDE is developed by JetBrains who makes a slew of other programs along with IDEA which is offered in both a community version and a commercial licensed version but in this particular instance we are only interested in the community edition. An advantage to this IDE is that is supports Java versions all the way up to Java 8 and also supports a long list of other computing languages such as Erlang, Scala, Haskell, Python, and many more. Also, IDEA supports software versioning and revision control systems which makes development across a team much more streamlined and less troublesome. A few options they offer are Git, GitHub, and Subversion to name some of the more popular ones but they do offer more than that. What this also means is that we can develop across different operating systems because IDEA can run on Windows, OS X, and Linux as well. The most important aspect of this integrated development environment is that it allows for app development. For Android specifically, IntelliJ's IDEA seems to be the frontrunner and the main competitor to Android Studio which we will cover later. Back then when Android application development began, Eclipse was the main integrated development environment and tool kit to use but gradually over the years IDEA outgrew and overshadowed Eclipse immensely especially due to the fact that Google ditched Eclipse all together. A similar fact when comparing to Android Studio is that if we use IntelliJ's IDEA, we can get some of the same features that are supported in Android Studio in IDEA. For example, IntelliJ IDEA 13 includes all of the Android Studio features except for the redesigned new project wizard and the Application Engine cloud endpoints of integration. One of the appealing factors of IDEA is code development insight. It has many integrated tools that assist when coding which is extremely helpful as opposed to writing your code in Notepad. For example IDEA offers smart completion, chain completion, static members completion, data flow analysis, language injection, cross-language refactorings, detection duplicates, inspections and quick-fixes. Not a lot of IDE's have all of these capabilities and if they do they aren't as mature and intuitive as IDEA. This could ease the pain of issue you run into when developing an application which takes to the next point of ergonomics. In my previous experiences, when you have to deal with the IDE's complexity when you need to be developing code. So IDEA offers a editor-centric environment, shortcuts for everything, ergonomic user interface, and an inline debugger. You shouldn't need to leave your editor to do something that is not code related so quick popups are helpful for checking information without leaving the context you are in. It has a dedicated keyboard shortcuts for almost everything which enables you to code rather than spending most of your time looking for things and clicking around. One of the most important things is debugging your application when you run into issues you cannot solve right away and the debugger that IDEA offers is very powerful. For example, when stepping through the code you do not have to hover over variables to see what they contain because it displays it on the screen for you! As aforementioned, the toolsets that are offered are plentiful. There is version control, build tools, test runner and coverage, de-compiler, terminal, database tools, applications servers, docker, etc. A lot of these tools if not all of them are not offered in other IDE's and you have to waste your time looking for plugins and downloading external software. These are all the things that make IntelliJ IDEA extremely powerful and probably the integrated development environment tool that we are going to use but the biggest thing that I see that is beneficial when developing a brand-new application to put on the market is that not only is it compatible with Android development but you can develop Apple application as well. This is imperative when it comes to the growth of your company. If your application is doing well, you are going to want to make your application cross platform compatible and you can do this with IntelliJ IDEA.

3.7.2.1.3 Android Studios

Android Studio is the official integrated development environment for the Android platform which had its first stable build released in December in 2014. Obviously, this integrated development environment would be one of the options for us to look at if we decided to go with developing an Android application as opposed to developing an Apple application. It is based off of JetBrains IntelliJ IDEA which we talked about earlier but it is designed specifically for Android development. Similar to IDEA, it is compatible across platforms which include Windows, OS X, and Linux. Android Studio offers robust and flexible build system, it is designed for teams, and is optimized for all Android devices. You can configure your project to include local and hosted libraries, and define build variants that include different code and resources. It also integrates with version control tools such as GitHub and Subversion to keep your team in sync with the project. These may need to be installed separately when downloading the integrated development environment but at least it does provide those capabilities. Android Studio also supports a variety of languages. For example, you can develop in C and C++ using the Android Native Development Kit (NDK) but it isn't something that google promotes because they say "the NDK will not benefit most apps. As a developer, you need to balance its benefits against its drawbacks. Notably, using native code on Android generally does not result in a noticeable performance improvement, but it always increases your app complexity". So in this case, we will probably be using Java and JavaScript which is most optimal for us at the time. Other options may include Corona, Phonegap, C# and much more. One strong and powerful advantage that Android Studio brings to the table is its fast and feature-rich emulator. The Android Emulator simulates a device and displays it on the computer which enables you to pick the device you want to emulate and then it lets you interact with the device through button clicks, button hovers, and navigation. The Android Emulator installs and starts the application you are developing faster than a real device and then you can test and interact with it on various configurations and Android devices such as tablets, phones, TVs, etc. This is a great addition since you can do testing and demonstrations very quickly and effortlessly which is important in development. It is also important in testing in terms of devices. Say if you use a different integration development environment that does not provide an emulator but lets you compile and run the application in real time. You are able to see your application but how will look on a Samsung s8 compare to an older version LG phone? How will it display on a tablet when the aspect ratio is different? These things need to be taken into consideration when trying to optimize your application and building a versatile and capable product. The emulator is useful and robust as it is but we need to keep in mind that there are a few things that it cannot emulate. For instance, it does not include virtual hardware for WiFi, Bluetooth, NFC, SD card insert/eject, Device-attached headphones, and USB to name a few. There are a few things though that Android Studio is lacking that IntelliJ offers that we mentioned before. Android Studio does not offer JSON support. If the Android application needs to connect to a JSON endpoint, Android Studio does not offer it but IntelliJ IDEA does. If the application uses a relational database, IntelliJ IDEA will provide a very useful editor and also includes a connector which Android Studio does not offer. Something that IntelliJ IDEA offers that Android Studio does not is a REST API client which can be used to execute URL request to test any backend services we may implement. Since Android Studio IS built based off of IntelliJ, there are a few things they share in common which is useful in application tools such as code completion, style and formatting, version control basics, Gradle build system, etc. It may seem that Android Studio can be a useful environment to choose to develop an Android app if that is the route we choose to go.

3.7.2.2 Xcode vs AppCode

After researching and investigating on all the integrated development environments for Android, the next step is to look into all the different options for developing an Apple application. The deciding factor between choosing whether to develop an Android application or an Apple application may not lie in the actual platform but in the developing environment for that platform. The developing environment is just as important if not more. So the first integrated development environment we will look at is Xcode.

3.7.2.2.1 Xcode

Xcode is an integrated development environment that was specifically designed for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS and tvOS. It was first released in 2003 and is currently free for macOS Sierra users but Apple also made a version of the software that is available to the public with an Apple Developer account. SDo the software itself is completely free. Though anyone can download it and use it, since it is developed and created by Apple, you must have a computer on a Mac OS X platform. There are ways to get around this like using virtualization software like VMWare or Virtual where it takes out a lot of the complications. The only downside to this is that virtualization is a very demanding process and need more, if not a lot more, power to operate successfully. It is recommended that the minimum requirements are a dual core intel processor, 4GB of RAM, and hardware virtualization. So let’s say we get the integrated development environment up and running, what is next? What makes this a possible candidate for developing an Apple Application. Well it is the complete toolset for developing Mac applications but also a interface builder, testing application, and asset management toolkit. Xcode is supposed to hold a feature over everyone else’s head which is storyboarding. Most high-end iPhone apps involve extensive use of storyboards which are a quick and easy way to start building a app. The biggest benefit of using storyboards being able to see and follow the flow through the application and zoom in and out from a low level to a high level overview of what your application is doing. This is useful when jumping into an existing project or just following your current project. Another great benefit is that you can see your changes to the interface immediately instead of having to wait to compile and run your code. The only issue with storyboards are merge conflicts which occur when two people on the same team edit the storyboard at the same time and try to merge. It would be quite painful to sort our those merge conflicts because the format of the storyboard file is in XML and viewing a diff between an XML is not a small task. Another benefit that Xcode has it that its user interface is easy to work with and efficient. The build settings featureset is great and including app entitlements is a matter of minutes. A bonus with Xcode is that since it was developed by Apple, its process to upload the application to the App store is seamless. It can easily sign and organize the application and submit it via iTunes connect. The downside I see with using Xcode, and this is not its fault, is that it only supports two languages at this time which are Objective-C and Swift. Objective-C is not something we are not familiar with but is definitely not something we use every day so that would be something to get used to. As for Swift, it is a compiled language specific to developing Apple applications and that is something that we are definitely not familiar with. It would be a learning curve so we would have to factor that into our time resource, do we want to spend time learning a new language? On the bright side of things there is a lot of documentation on both languages and the internet is filled with information. Overall if this is the route to go it is a very versatile and capable software for MAC development.

3.7.2.2.2 AppCode

AppCode is an integrated development environment for developing Apple applications that was built on the JetBrains IntelliJ IDEA platform. So, this is their Apple application development environment to their IDEA Android application development environment. The first stable version of AppCode was released to public preview in April 2011. It supports Swift, Objective-C, C, C++, JavaScript, HTML, CSS, XPath, and XML. Similar to its sibling IDEA platform it is feature packed and full of functionality for the user. It comes with coding assistance, code completion, live templates, code analysis, iOS simulator and much more. One of the great things that I see so far is that is supports many more languages than Xcode and that is a big plus. Another key feature that AppCode has over Xcode is the task of refactoring. AppCode developers have a simple two key stroke and enter the name to refactor without any interruptions where in Xcode there is a “Rename” command. This command is used for renaming the names of classes globally and is said to be slow, uncertain, and can cause some changes that you did not necessary want to change to begin with. Some disadvantages though, may outweigh all the positives, are many to name which is the lack of storyboards, glitches, and the user interface. The lack of storyboards may not seem important but it can be a very useful tool when starting out or if you are just designing a simple application. Storyboarding is something that Xcode clearly has the advantage over AppCode. It is also known that when using AppCode, there are many instances where program crashes are frequent, scrolling becomes and issues, flashing and freezing of the application screen occurs, and the entire system can get locked up. Unfortunately, the only way to get out of some of these issues is simply closing the application or sometimes shutting down your entire box. The user interface is also spoken about a lot in terms of its features and settings. It has a long way to go when compared to the Apple native Xcode application. Application entitlements is not possible, build settings are substandard and when you edit them, only certain things get updated. Another constraint that AppCode may cause is that AppCode require Xcode because we would need to have Xcode installed on our computer before we even decide to use AppCode. It is meant to be used with Xcode but not actually replace it. Another advantage Xcode has over AppCode is that AppCode cannot push or submit apps to the App Store like Xcode can for obvious reason. What it can do is integrate with CocoaPods which is a dependency manager for Object-C projects or Swift. It has over 24 thousand libraries and is used in over 2.2 million apps. Of course, CocoaPods is not a requirement to build a project and not at all mandatory for AppCode but it is great that AppCode can integrate it to make it easy to manage all your dependencies and keep them up to date. The dependencies are specified in a text file called a Podfile and then CocoaPods resolve those dependencies and puts it together with the source code and links it with Xcode to build the project. Aside from these nuances that this developing environment may cause, there are still good things about it that are worth mentioning. Just like all the other integrated development environments, it seems that the standard or usual components it comes with are the Unit testing support, run/debug on device, code formation, ability to add plugins. Another aspect to take into account is that AppCode does cost money for its integrated development environment. Right now AppCode is available for a free 30-day trial period but then after that the cost of using AppCode would be $89 for the first year and &71 for the following years. It cost even more for app companies at $199 for the first year and $159 for the following year. It should be made a point that since AppCode cannot be used without Xcode, the strategic way to approach this is to thoroughly learn Xcode and get familiar with it and then get the 30-day trial of AppCode and then at that point make a better decision on whether or not AppCode is worth the extra money.

3.7.2.3 Java Development

Java is a general-purpose computer programming language that first appeared twenty-two years ago on May 23, 1995 and was designed by James Gosling. The developer company used to be Sun Microsystems who is now owned by Oracle. This computer programming language is concurrent, class-based, object-oriented and is meant for application developers to ‘WORA’ which means write once, run anywhere. James Gosling also designed this programming language off of the C/C++ syntax style so that developers would recognize and be at least somewhat familiar with so learning the language would be a breeze. It is also similar to C++ in the aspect that they are both object oriented programming languages. Java is designed to have as few implementation dependencies as it possibly can which makes it easy to run on all platforms that support Java. The Java platform was designed with portability in mind. This means that programs that are written for the Java platform must run similarly on any hardware and/or operating system or any combination thereof with substantial runtime support. This is done by Java bytecode which is pretty much the same thing as machine code and its purpose is the same thing but it is meant for a virtual machine that is written specifically for the host hardware to execute. People who use java usually either have a Java Runtime Environment installed on their local machine or on their web browsers for any Java applets. So what all this means is the benefit of this is that if we were to develop an Android application, we can use either Windows, Linux, or OS S and what the Java compiler does is converts the source code into bytecode. To provide some benefits that Java provides as a programming language and why it would be a good idea to choose it for this project is because Java is easy to learn, it is an object oriented programming language, it has a loaded API, large amount of open source libraries, and have plentiful documentation support. Java is easy to read, easy to write and easy to learn. Its has minimum magic characters like angle brackets, and fluent English like syntax. Java is also a object oriented programming language also known as OOP which means you have the ability to define not only the data type of a data structure but also the different types of operations or functions that can be applied to a data structure, hence the object part. Everything becomes an object that contains both the data and the functions. Another characteristic of the programming language is that objects can also inherit characteristics from other objects and use them as their own. This leaves a lot of room for optimization and the ability to add to existing code which important for a growing product or company. Java's impressive API is available for many things such as I/O, utilities, XML parsing, database connection, networking, and just about everything else. Whatever you cannot find, Google, Apache Commons, and others like it have it covered by their open source libraries. Also, if you run into any issues are need help figuring out something new, there is a humongous documentation called Javadoc. Javadoc tells you about syntax, methods, API, anything and everything you want to know about java. One of the biggest advantage of Java that I personally can agree with is that there is no need for ridiculous and frustrating pointers. Pointers in C are variables which contain the address in memory of another variable. Which means a pointer and point to another pointer which points to another pointer and you get the gist of it; it can be a huge mess and a giant headache. But since we are talking about app development it is important to understand that Android heavily relies on Java fundamentals and includes a lot of the standard Java libraries such as the data structure libraries, graphic libraries, etc. So when it comes down to it Java would seem like it would be the best option as a programming language to develop an Android app but one of the downfalls is that you can only develop Android applications as Apple applications do not support Java. If we want to develop an Apple application we would have to use Swift or Objective-C.

3.7.2.4 Objective-C Development

Objective see is a class-based, reflective, object-oriented programming language that came out in 1984, thirty-three years ago and was created by two men by the names of Brad Cox and Tom Love. Objective-C programming language was the main language used by NeXTSTEP operating system by the company NeXT which is where OS X and iOS were built upon. This explains why Apple applications can only be written in Objective-C or in Swift programming languages. Objective-C uses Cocoa and Cocoa Touch libraries as their standard interface or framework. The cool thing about Objective-C is that is acts like a top layer to C which means that you can compile any C program with and Objective-C compiler. Aside from that though, Objective-C is still a very object-oriented style kind of programming which consists of message passing, interfaces, classes, and much more. Message passing, which is the model of Objective-C's object-oriented programming, is how the code is referenced by a method. In more common object-oriented programming most time you call and method or you call on a class or object but in Objective-C you send a message and the receiving object itself will interpret it and is resolved at runtime. So in more conventional object-oriented programming calling a method would be illustrated as such "obj -> method(argument)" as where in Objective-C is written as so: " [obj method:argument]; ". This is not something that is difficult to follow but it would certainly take a learning curve and getting used to and that has to be taken into consideration when we are trying to develop a brand new application and learning a new programming language. Also, in Objective-C they require that interfaces and implementation of a class be in separately declared code blocks in which other object-oriented programming languages would call this class declaration. Overall it looks like this object-oriented programming language takes a little bit of C language and C++ and puts it together into its own language. May not be difficult to learn but do we want to spend the time and invest into this language is the real question.

3.7.3 Decision on Operating System and IDE for Applications.

To conclude this research on software and software development, we need to come to a decision on what components we are going to choose. To do that, we have come up with a point system to fairly and unbiasedly rank the options that we have and the one that comes out on top will be the one that should be the most beneficial to us and our project. The point system that we are going to use is the one where we take each component and rate it and from that forego onto the component and rate it accordingly until we put them side by side and choose the one with the most points. The criteria on which we are going to point these components are five imperative pillars that we believe are good points-of-references to use when defining our success in the future. These points are availability, usability, popularity, project size, cost, accessibility. Availability in a component defines how readily available is support for the product. How easy is it to do a google search and find results if we run into any issues? How strong is the documentation for that particular product? If the issues escalate, does the product even offer support? Popularity determines the reach a certain product has. Is it used all over the world? By Based on the table below, our group make the decision that Android IDE will be suitable choice for project because of Availability, Usability, Popularity, Project Size, Cost, and Accessibility.

	AVAILABILITY	USABILITY	POPULARITY	PROJECT SIZE	COST	ACCESSIBILITY
ANDROID	✓	✓	✓	✓	✓	✓
APPLE	✓	✓	✓	✓		✓
WINDOWS					✓	
ANDROID STUDIO	✓	✓	✓	✓	✓	✓
ECLIPSE	✓		✓	✓	✓	✓
INTELLIJ	✓	✓		✓		✓
XCODE	✓		✓	✓		
APPCODE	✓		✓	✓		
JAVA	✓	✓	✓	✓	✓	✓
OBJECTIVE-C	✓	✓	✓	✓	✓	

Table 7 Six Pillars for Decision

used overseas? Is there enough people using it that we can easily access resources or grow or even just be a talking point? Project size is imperative to consider even in the early birthing stages of a product because even though it might not matter now in the beginning stages, planning for the future growth of a product and its project size can prevent and account for roadblocks and constraints that you would normally run into if you did not consider the size of your team, or the complexity your product grew to. Cost which is an obviously important key to whether or not we would decide to go with an appropriate product because finance always needs to be accounted for. The last pillar of importance is accessibility. Accessibility is defined to us by how easily this product and be utilized and learned. Is it plug-n-play or does it require more of a learning curve? This is important because considering that we have so many other aspects of the product to develop including the physical components and hardware, time is a huge resource we can not be wasteful of. So with those pillars as the driving force for our decision, we concluded on an Android platform, the Android Studio's integrated development environment, and Java programming languages as our tools. We chose the Android platform because it was superior when it came to its popularity and accessibility. Android is known all over the world and sold on millions of phones every year. We can definitely scale out the product if it grew to what we think it would. Also, it is free to upload apps to the Play Store and we are able to price it at whatever price we choose. We chose Android Studio’s because it is the specific developing tool for Android so how can you go wrong with that? Android is also based off of Intellij’s IDE and they were our second choice so it all works out. As for our programming language, choosing Android obviously narrowed down our choices to just a couple which made it easier because we are more familiar with Java and worked with it in the pass so it is the obvious selection.

3.7.4 Software Selection Conclusion

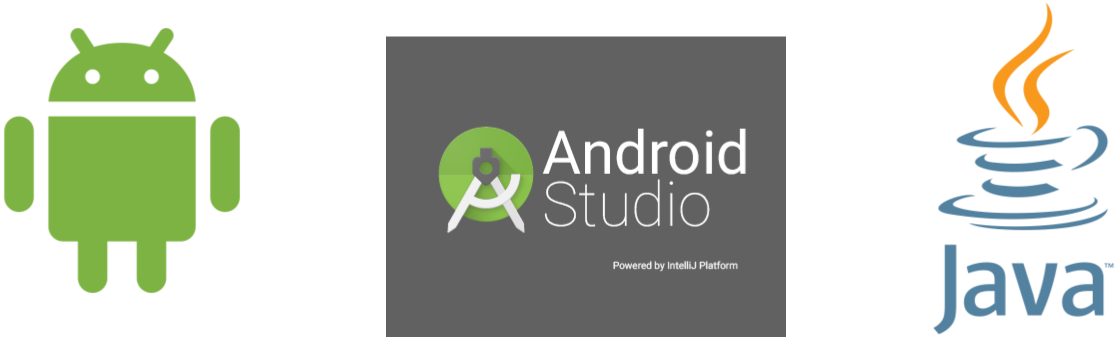


Figure 28 Android, Android Studio, and Java 8.0

Now that we have the software requirements in place, we can now drill down and start scoping out and designing our interface as well as our application flow. Basically the user interface/ user experience should now be all planned out and ready to be implemented. The more detailed we are now the easier it will be when the implementation process begins.

3.8 Harness Research

The harness is a vital piece of this design project. If it is not large enough or tough enough our electronic device may run into problems or even cause harm to the dog. For the purposes of this project we will purchase a completed harness and then alter it so that it suits the needs that are required of it. Ideally the harness will meet most or all of our needs and the alterations will be minimal.

But why a harness and not a collar? The harness allows us to have a location to place the PCB. The parts associated will be larger than those used on a collar, but will also most likely be cheaper. Also using a collar would make taking certain readings more difficult, such as temperature. The harness is also more comfortable for a dog when on leash. This is true because when a dog pulls when on a collar he gets choked and may hurt his throat, but a harness distributes the force to many different parts which ultimately makes it much less likely to injure the dog.

These are the requirements that will be needed for a selected harness to be chosen:

- Adjustable
- Sufficient surface area (back)

- Sufficient surface area (stomach)
- Thick straps around front legs
- Moderate to low in cost
- Waterproof/water-resistant fabric
- Tough/durable fabric
- Stylish if possible

While style is not a primary concern, it should play a factor as no one wants their dog harness to look odd, bulky, or unattractive. To be able to make this design project work we also need to select a harness that allows for our electronics to be safely secured to the dog. There are an endless amount of harness examples that exist. In this section we will go over some specific examples of the different types of harnesses that are for sale on the market and the type that would most benefit our project.

While our final harness may or may not exist in this section, it will be added to this section upon selection. Our design will depend on the size of our PCB. The harness needs to be able to contain the PCB and its case securely on the back of the dog. The alterations we make will be important but the surface area has to be large enough so that the case is not resting directly on the dogs back. In addition to this, the harness must be able to conceal and help protect the wires and connections of the sensors that will be places at various points on the dog's body. For this reason we need the straps that go around the front legs and around the body to be thick so that the wires can be sewn inside the fabric, or have fabric laid over the wires and sewn to the existing straps.

The size of the harness is what makes capable of fitting to all dogs of various sizes. The only dogs who may have difficulty with our design are very small dogs. This issue will only arise if our PCB is too large to place on the back of a tiny dog. Due to these criteria we can narrow down the type of harness we will ultimately need, but will not select until our PCB is designed.

We will look at numerous harnesses and the ones that have similar traits will be categorized into groups, such as “Harness Type I”. This will help us better explain what makes a good harness and what does not. Once we define what makes a good harness, we will select one in that category once the PCB size is known. We will also look at why collars will not be used for this project. Each of the different harness categories as well as the collar are accompanied by diagrams and pictures. These are important to understand why we will choose or will not choose a harness.

3.8.1 Collar Issues

Collars tend to be more popular than harnesses. This is most likely due to the fact that they are cheaper and can be replaced easily. Also, it could be due to the fact that owners like the way they look. For our project we considered collars but eventually decided against them for a number of reasons. While we list some of the potential issues with collars below, there may have been compounding problems as we moved forward with the project. The size of the collar could account for many, many of the problems we would have encountered.

- No surface area, electronics may be too large
- May be too difficult to place sensors

- Pressure on dog's throat/neck

The picture referenced in 3.8.1 is a basic example of a dog collar. It would have been quite difficult and more expensive to mount an electronic device to the small surfaces of the collar. Additional problems would have arisen with our sensors. Detecting temperature may have been very difficult using a collar. Another factor that limits the usefulness of the collar is that the electronic device may experience stress or pressure from the dog pulling. If the device was on the back of a harness it may feel little to no stress from a tugging dog.

While these collars tend to be popular they are a poor fit for our project and come with many potential problems that we have chosen to avoid. Collars are not impossible to implement for this project, in fact they can be found by a simple internet search as some exist already as kickstarters or are already for sale. But for the reasons stated above we have decided to move forward with the concept of the Smart Harness.

3.8.2 Harness Type I

These harnesses tend to be very cheap but they are also very small in every regard. They are also most likely to be uncomfortable on the dog. This is due to the amount of surface area. Less area means that certain points on the dog are likely to be very uncomfortable because all of the pressure from tugging on the leash will be places in certain locations on the dog.

These are poor choices when taking our project needs into account. The surface area for placing our electronic device is non-existent. This means that our device would sit practically on the back of the dog, which we are trying to avoid. Also the straps and underside fabric surface area are very small which makes concealing the sensors and wires very difficult. These harnesses are a bad fit for the purposes of our project. While they tend to be small they do have strong fabric. But for our purposes it is not enough.

By looking at the diagram (Figure 3.8.1) it is easy to see why a harness of this type would be a poor fit for the purposes of this project. The lack of any surface area on the back of the dog makes it near impossible to place our electronic device anywhere on the dog. Also, there is no stomach surface area either which makes placing a sensor there likely to be damaged by the dog or for various other reasons. The straps are thin and make the protection and concealment of any wires leading to sensors difficult if not impossible.

For the reasons stated above this type of harness will not be selected for our project. The cost range for harnesses of this type can vary by company but generally fall into a range of $13.00 to $25.00.

3.8.3 Harness Type II

This type of harness is a possible choice for our design. While they tend to have better surface area than Harness Type I, it may only be on one side (back or underside but not both). The fabric in this category tends to be a little weaker than Type I, some are of high quality and some of poor quality. This is most likely where we will get our harness from as long the PCB will fit, as it will most likely perform all other tasks we require reasonably well and stay low in cost.

In the diagram (Figure 3.8.3) many good traits are pointed out. This harness type would have larger surface area and stronger straps than Type I. Without knowing the PCB size this may or may not be enough surface area. If the PCB is small enough to fit this harness, than this would be an excellent option for our design.

The costs may be higher than Harness Type I but overall should still be relatively low. The cost range for harnesses of this type can vary by company but generally fall into a range of $20.00 to $30.00. This brings us to our third and final harness type.

3.8.4 Harness Type III

These are heavy duty harnesses. The durability and strength are of the highest quality and they also contain plenty of surface area to comfortably place our electronic device. The price for these harnesses tend to be higher, but if our PCB is large and can only be placed on harnesses of this type, then we will have no choice but to select this type of harness.

Many of these are also known as “Tactical” dog harnesses and are used for military dogs or for service dogs. For the purposes of our project this we will only choose this harness if the device makes it necessary.

Figure 3.8.4 shows a variety of harnesses that can be classified as Harness Type III. Not all of the harnesses in this type are created equal, some may have smaller straps or the harness may stretch all the way to the rear of the dog's body, but in general they meet and exceed all our criteria. However some may a bit too large, bulky and expensive, so this is only likely to be chosen if the Type II prove to be too small for the device or if the prices tend to be similar. The cost range for harnesses of this type can vary by company but generally fall into a range of $20.00 to $100.00.

3.8.5 Harness Comparison

Comparing the different harness types (as well as basic collars) was important in deciding how to proceed with our project. Of the types discussed Type I was a poor fit for our Smart Harness because there was little to no surface area for the device to be placed, as well as having no real way to conceal the various wires that will attach to the sensors.

	Harness Type I	Harness Type II	Harness Type III
Cost Range	From $13.00 to $25.00	From $20.00 to $30.00	From $20.00 to $100.00
Sufficient Back Area	No	Yes	Yes
Sufficient	No	Yes	Yes

Underside Area			
Sufficient Strap Thickness	No	Yes	Yes
Sufficient Durability	Yes	Varies	Yes

Table 8 Harness Comparison Table

3.8.6 Harness Selection

Based on the above table the best harness for our projects purposes will most likely be a selection from the Harness Type II category. As you can see from the table that is forementioned, the Harness Type II would be an optimal choice as it fits all the criteria we would need to house the hardware components and be able to sustain longevity and durability. It also fits better into our budgeting and finance requirements because there is a possibility that we may need to order more than just one harness due to testing purposes. We would need to test its durability by doing some stress test and that would require putting the harness in environments that will make the harness not presentable. We have not decided on a particular harness yet and will decide once the PCB board has been designed and ordered. Only then will we know our size constraints and we will be able to move forward in selecting our harness for the project.

Moreover the harness is one of the most important component for our project. Without the harness, our project can not be done. Also, the harness will be modified once we get it due to our group members will integrate the circuit inside the harness.

Chances are high that this harness will not be selected and ordered until the second semester of our senior design course. Although the selection comes later it is important to plan and organize a strategy when it comes to the foundation of our hardware components. This matter will be discussed more during the prototype section. However the prototype is what we intend to do, it might also be modified later because of the unknown reality issues.

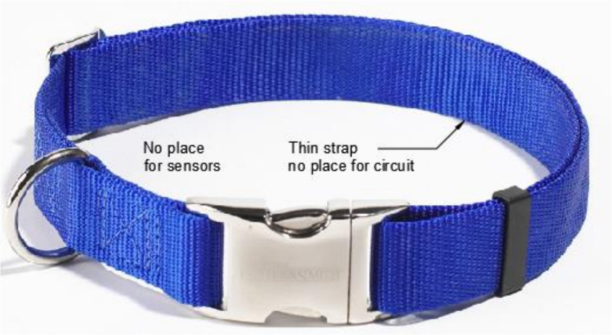


Figure 29 A basic dog collar

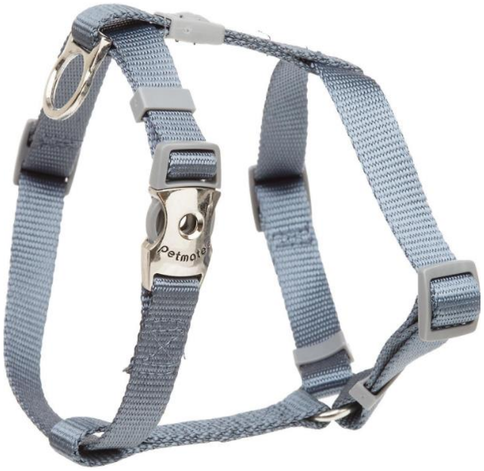


Figure 30 Example of Harness Type I

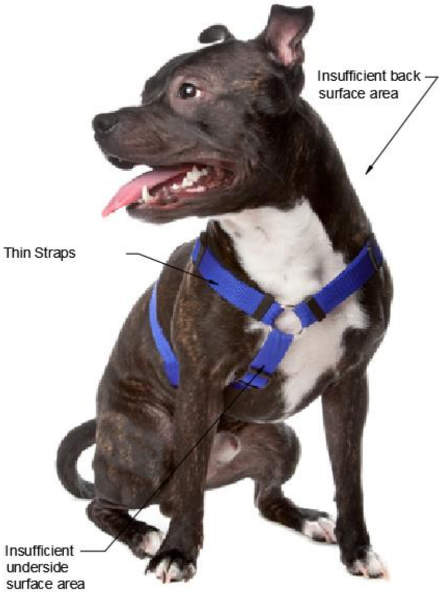


Figure 31 Example of Harness Type I.a

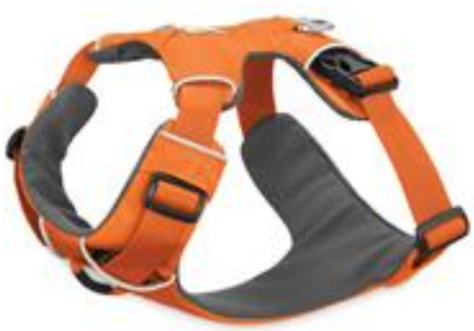


Figure 32 Examples of Harness Type II

Figure 33 Examples of Harness Type II.a

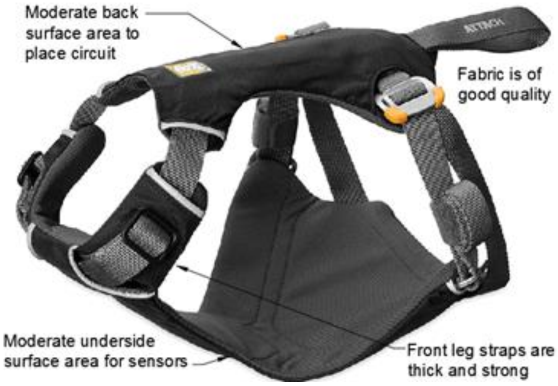
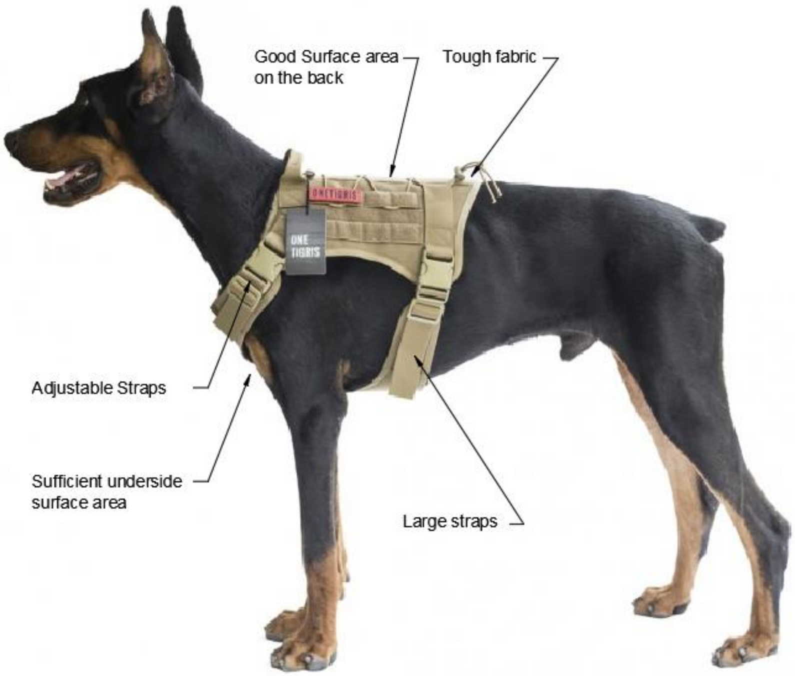


Figure 34 Examples of Harness Type II.b

Figure 35 Examples of Harness Type III

Figure 36 Example of Harness Type III.a



Good Surface area on the back
Tough fabric
ONETIGRIS
Adjustable Straps
Sufficient underside surface area
Large straps

Figure 37 Example of Harness Type III.b

4.0 Design Constraints and Standards

This section is to highlight any standards and/or constraints that apply to the Smart Harness. The constraints that are relevant to this project are listed first, and there may be some constraints that we will come across in the future as the project progresses. This is followed by addressing any standards that might be applicable to our design. Any and all standards that are pertinent to the project are here discussed at length. We will follow all standards that are listed here for the entirety of the project. Some might have big impact on our design, and some might affect a little on our design. However, our group will list all as much as we can in order to understand more about the constraints and standard which we learn in class.

4.1 Project Constraints

Uncontrollable factors for dog: We do not yet know the dog which we will test the harness on which means we are unsure of the temperament of the dog, the size, the hair type, the hair length, the breed. We can have all the specifications but if we do not narrow down the details to the correct breed type if not all breed types, then we might have an issue when testing on the dog.
Not only with the unknown about the dog's condition but also the hardware and software design of this project will need to be consider with the realistic constraints during the development process.

4.1.1 Economic Constraints

The designing cost is a most major constraint due to the couple reasons. The first reason is the cost of the component. After the researching part for our project, we can see that if we want the newest sensor for our project such as temperature, heart beat, accelerometer, and bluetooth. It would cost a lot money, and the different would be $20. The next reason is the funding. As we know that senior design at University of Central Florida, some group is getting funding from the big company such as Boeing, Harris Corporation, or Texas Instrument. However in our case, we have to get our own fund; as a result, the expanding function for dog harness is limit.

Another economy constraint is the manufacturing cost which has two reasons which keep our design limited. First of all, in order to be able to sale the product, the quality of harness has to be above the standard quality. However, to achieve this we need will need the high quality of component inside of our hardware. This reason will get contrast with our engineering requirement which is the cost of harness has to be low under $125. The next reason is out the market there many type of harness such as the collars which as mentioned in our market research above. The minimum of the cost of harness is above $150 which higher than our initial cost so far for the component. The total cost of our component is already $70. Due to the price system and quality of product, the research of balance need to be achieved.

4.1.2 Environmental Constraints

Nowadays, environment is the one of the most important element in our society. There are many question has been raised that will your design or product will be friendly to the earth? or will the product can be recycled after used? The dog's harness belt will not associate with the earth such as pollution or damaged environment. However the recycle will be the problem for our project design. As nowadays, we know very little about the circuit board recycling. After the using, where they will go? This is the question that our group been conducting research. As a result, we find out that the circuit board with technology as now, we could not recycle. In addition, all of the will be shipping to the third world country such as China, India, or South Africa. Recycling is not one of possible solution for our group during the time of developing project.

4.1.3 Social Constraints

A main social constraint that we see for our product is how many people we can reach and affect with our product. The highest and most imperative goal for us as a team and as a product above all else is how we can change the world. If that means providing the tools for consumers to have the ability to better their pet's life, then that is a goal well met. Almost everyone has a phone all over the world, even children at a very young age own a phone (which may be a good demographic to take into consideration). 42.5 million households in the United States alone own one or more dogs and the total number of dogs in the country is in excess of 73 million. The question is how can we reach these people. The harness should should be versatile enough to fit as many dogs as possible. The harness should be easy to put on so that your average person can slip it on their pet with little to no hassle. The harness should not be detrimental to the animal but instead beneficial to its life. We need to provide a platform where information and data is readily available for both the user and the practitioner with full transparency. And eventually we should provide a network where everyone without product can socialize their pets health and data in a way that it will motivate others to better their pet's life and have a good time while doing it. As they say, there is strength in numbers. And those that do not use the product should feel the need or want for the product because they see such benefits that come from it. This should also open up a new realm for the veterinarian and provide them with tools that were never accessible before. If we meet these constraints then I think we can easily say we've done a good job.

4.1.4 Political Constraints

The harness and software application on a basic and micro level should not violate any United States law or regulations. Any work that is done should be of our own and patented so that others can not steal our ideas and hard work but that also means we should not steal other ideas and hard work as well. It is also said that we cannot violate any human rights nor animal rights by the use or development of our product. Any testing that is done will be done in a safe and controlled environment that will not cause harm to any human nor animal. Just like when kids toys come with toxic paint, or a recall on a motor vehicle that can cause death or injury, we need to take into full consideration of the effects our harness may have on an animal. We need to provide clear and concise warning signs and documentations of any hazards. Politically we should keep the production of our product and software development in the United States to build our homelands economy but there is a lot of controversy on this on whether importing goods is actually beneficial so this will have to be addressed later. Lastly, our product as a whole cannot discriminate against any gender, race, or ethnicity. The product will not be offensive in any way or have the use of profanity, vulgarness, or explicit content. This is not only to prevent future lawsuits but also from just an ethical standpoint is important to follow these political constraints.

4.1.5 Ethical Constraints

As far as we concern is that while the hardness be harm for the user or for the dog? The priority of our project is make sure that this device will need to be safe and friendly for both dogs and its owners. The dog harness belt project will not using any toxication product inside such as battery with mercury or the cheap sensor in order to lower the cost of product. The harness circuit will using the considered material in order to make sure the safety of the user. Moreover, our group also will not cut off any functions of the projects or anything solely for financial reason.

When it come to copy and patent protection, during our work will make sure to give the credit to the people who has made the patent such as Arduino MCU, software IDE, and the Sensors design.

4.1.6 Animal Health and Safety Constraints

Animal health and safety constraints include but are not limited to the design of the dog harness, the hardware components of the dog harness, the testing of the dog harness, and the use of the dog harness. The design of the dog harness should not impede on daily functions of a dog. It should not restrict them from doing dog and dog related activities. The harness should not deteriorate the dog's health, comfort, nor life expectancy in any way. The hardware components should be placed in a way that does not harm the animal. For example there should not be any stray wires that could electrocute the animal, the battery should not explode on the dog, or the hardware should not get so hot that it causes discomfort or even to an extreme situation where it burns the animal. Testing should be never put an animal in any harm, it should be done in a controlled and safe environment, and it should be done responsibly. The user manual should clearly define how the harness is meant be used and should deter against irresponsible uses. The last and maybe the most important constraint is to make sure the information/data that we provide through the harness and its hardware components are as accurate as possible. We cannot produce a product that is misleading to the public that could put the dog's life at risk.

4.1.7 Manufacturability Constraints

When making the dog harness, the manufacturing need to be taken in account. One of the element is the material aka part. For instance, when we choosing the component all the group members will consider by two things. One is the time line of that component's technology. The second one is how much is the component will be expected available. Two is the available quantity of the components. How many they have left in stock? If there are more than five hundred components left, it will not be the issued for the project; however, if it only have one hundred components left that where we don't want to choosing for that component because of discontinued component problems.

The next thing in manufacturability is the blueprint of the dog hardness, the prototype, and the water splash proof. First of all the blueprint, as the definition of blue print the details about of dog harness need to layout every details from circuit design, the resistors component, or the capacitor. To accomplish this our group will use the Eagle cat Software and draw out everything for the circuit design. Secondly, the prototype is need to be makes. Due to the lacking of funding, our group could not order from any company to modify the dog belt or integrated the circuit inside of the belt. To work on this problem, we will put everything together by our own in order to obtain the belt with the circuit inside. Lastly, the water splash proof need to be made by flowing one of our requirement. Again due to lack of funding we could not request the manufacturer to help us the water splash proof for the dog harness. To make this work, our group will work on the design for a case where we can put the circuit inside as well as connection of the wire also the research online to do the waterproof will need to be conduct by google.com or youtube.com.

4.1.8 Sustainability Constraints

The sustain goal of the dog harness will need to be guarantee working for least of five years under assume dog activity conditions such as run jump or rolling around. The sustainability constraint is connect with the environment where the dog live in, and it also the most important factor of the product age during the time line.

First of all the circuit which inside the box need to be able firmly attach inside of the belt during the normal day of the dog. For example, the user will take the dog out to the park and let it run around. If the circuit not firmly attach to the belt there will be a risk that the circuit will be fall out or two sensors temperature sensor and the heart beat sensor will be dislocate and it will give the application wrong reading which is that what our group do not want it to be happened. To fix this problem, our group will make a tight pocket with the seal button and fit tight completely for the box circuit inside the so that the chance the box circuit fall out will not occur. As for the sensor, our group will integrate it firmly along with the belt so that it will not be dislocate during the normal activity of the dog.

Secondly, the pressure from the dog body will need to be look into in order to achieve the survival of the harness. In the situation where the dog will press its own body into the belt during the time it lay down. For the reference let say that the dog is weight 150lb. The box circuit need to be able withstand the light pressure from the dog body. However there is another solution is that we can put the circuit in front of dog chest so that the problem will not be occur.

Thirdly, it is a situation where is the dog rolling around on the ground. The box of circuit need to be able to withstand the impact of the momentum force from the rolling of the dog. The plastic case and the metal case will need to be consider when we are making the design for this project. The plastic case is softer compare to the metal case for the circuit. Each of them have advantage and disadvantage. For the plastic case, the advantage is light, soft, and durable, and the disadvantage is once the momentum force impact to the case it will lose the durability and easy to break. For the metal case, the advantage is hard, and the disadvantage is made from the metal it will be easy conduct the charge from electrical and make the damage to the circuit inside. In order to counter this constraint, we need to carefully choose the case so that it will make the safety to the circuit.

Fourthly, the temperature is one thing it need to be consider. The circuit sensor need to be withstand the hot temperature outside during the summer, and the cold temperature during the winter. Every component inside the harness always have the operational temperature. If the component temperature is exceeded the limit, the sensor or the MCU will not be able to function. To fix this problem, carefully pick the components and research location for the circuit inside the dog belt and position of the circuit on the dog body need to be made.

Lastly, the water splash proof is the most sustainability constraint factor which will affect to our design the most. In the situation where the dog will jump into the water, if the case of circuit get the water leak in, the dog harness will cease to function. As for the old people say, the water is enemy of the electrical. Many device until this day, all of the will stop function unless they have the waterproof option. However, in order to obtain this solution, our group will need work on the design for the circuit case, the seal of the circuit and the connection to make sure that there is no leakage water into the circuit.

4.1.9 Testing Constraints

Testing constraints pertain to both hardware and software. Testing constraints for the hardware will define the success and sustainability of the product. When testing the hardware, we need to pay attention to the environment that we are testing in. We need to remember that we are building a product for a creature that have unknown characteristics and behaviors that may or may not be sporadic. So we to account for this we need to test the harness and its hardware components in the most extreme and uncontrolled environments as well and noncontrolled. For example the sturdiness of the harness and the components need to be tested in rain, water, mud, wind, physical destruction, etc. We also need make sure that the heart rate monitor, the temperature reader, and the speedometer all have accurate readings. In the software aspect of testing, we need to make sure that there are no crashes or glitches. We need to make sure the data that comes over from the hardware are accurate. The most important part of testing the software aside from accurate data though is security and censorship. There should never be a case or scenario where users or veterinarians having data or information that they are not supposed to. We need to make sure that there is no leak of data or the ability to get data that is not yours. Privacy and patient confidentiality should be one of if not the most important if not the most important requirement and constraint that we have. Any breach of security or privacy is not only unethical but leaves a lot of pain for lawsuits and court time.

4.2 Project Standards

What are standards? These are the ideas that shape some of the boundaries of a product or design. A standard is defined as “something established by authority, custom, or general consent as a model or example” (Merriam-Webster). For the engineering world, there are numerous standards that exist for all sorts of devices and technologies. While not mandatory, standards are encouraged to be followed when designing or creating any product. In this section we will describe any hardware standards or software standards that are relevant in the design of our smart harness.

4.2.1 IEEE 802.15.1 WPAN/Bluetooth

Bluetooth gets its name from a danish king who united Denmark and Norway. Through communication he was able to combine two separate countries into one. He connected them, just as Bluetooth is used to connect various things today which is why it was given such an appropriate name. While not all our hardware are subject to standards there does exist a standard within IEEE that is meant for any device that contains Bluetooth technology.

The institute of Electrical and Electronics Engineers (IEEE) contains an 802 standards committee. Within IEEE there is a group with the designation 802.15 that mainly deals with wireless personal area network, or WPAN. Within 802.15 there are multiple standards in place. Each of the different standards are allocated a frequency band. 802.15.1 is the task group that is based on Bluetooth technology. This standard states that the Bluetooth device must operate in the 2.4 GHz frequency range. It also must use spread spectrum transmission technology which helps reduce interference and it must support transmission of voice and data at the same time using only 1 Mbit/s rate.

4.2.2 Alkaline Battery Power Supply Standards.

The battery standard was provided by ANSI American National Standard Institute to prevent the cell explosion under misused or abused cases, or the potential damage which can be caused from the battery. The standard was stated from the electropaedia.com.

The Temperature.	**Problem.**	**Consequence**
-60°C to 10°C	Low temperature	Lithium plating
	Low voltage	Anode dissolution
	Over Voltage	Overcharge/ Overheating
	Over Current	Overheating
60°C to 80°C	Over temperature.	SEI Layer breakdown.
120°C to 130°C	Separator Melt.	Short circuit or overheating.
	Electrolyte Breakdown.	Flammable gas Released.
180°C to 190°C	Pressure Build up	Possible Rupture
	Cathode Breakdown	Oxygen Release
	Oxidation of Cell Components	Thermal Runaway(Can Cause Fire)

Table 9 Part of Battery Cell Protection Mechanisms from ELECTROPAEDIA

The battery supply for dog harnesses circuit will fall under the -10°C to 60°C category. During the summer daytime in Florida as reference, the temperature of the sun is around maximum 40°C. The battery will fall into the low voltage problem which will cause the anode dissolution which will not cause any damage to the battery supply. As a result, the power supply which is the battery will not make any damage to the PCB system.

Those standard for the battery is the important factors for our group to consider when we decided to have the power supply as the battery. Also it is necessary for both user and their dog during the time using of the harness.

4.2.3 Heart Rate Monitor Standards.

The heart rate of a dog is a vital sign for a dog's health so it is imperative that we abide by the standards in order to provide a safe product for the end user. The heart rate and rhythm of a dog can be monitored or measured by directly applying a stethoscope or monitor to the chest wall so this is where we will need to place the component. Another standard is to take measurement while the dog has the most minimal movement to provide the most accurate reading possible. A canine's normal heart rate for dogs is between 60 and 140 beats per minute so if the measurement of the harness provides something out of the range, it is important to be able to notify the end user that their dog's heart rate is in a critical zone and should seek medical or professional help. Keeping track of these heart rate measurements is also a standard we must follow because diagnosing an issue can start by seeing a trend in decline of heart rate. This is critical information that we must provide through functionality of the harness and heart rate monitor.

4.2.4 Android Application Standards.

This section will cover the Standards that Android has defined themselves as well as the standards we defined that need to be in place for a successful and well thought out product.

Standard Design

- The app does not redefine the expected function of the system icon (such as the Back button).
- The app does not replace a system icon with a completely different icon if it triggers the standard UI behavior.
- If the app provides a custom system icon, the icon strongly resembles the system icon and triggers the standard system behavior.
- The app does not redefine or misuse Android UI patterns, such that icons or behaviors could be misleading or confusing to users.

Navigation

- The app supports standard system Back button navigation and does not make use of any custom, on-screen "Back button" prompts.
- All dialogs are dismissible using the Back button.
- Pressing the Home button at any point navigates to the Home screen of the device.

Permissions

- The app requests only the absolute minimum permissions that it needs to support core functionality.
- The app does not request permissions to access sensitive data (such as Contracts or the System Log) or services that can cost the user money (such as the Dialer or SMS), unless

related to a core capability of the app.

UI and Graphics

- The app supports both landscape and portrait orientations (if possible).
- Orientations expose largely the same features and actions and preserve functional parity. Minor changes in content or views are acceptable.
- - The app users the whole screen in both orientations and does not letterbox to account for orientation changes.
- Minor letterboxing to compensate for small variations in screen geometry is acceptable.
- The app correctly handles rapid transitions between display orientations without rendering problems.
-

User/App state

- The app should not leave any services running when the app is in the background, unless related to a core capability of the app.
- For example, the app should not leave services running to maintain a network connection for notifications, to maintain a Bluetooth connection, or to keep the GPS powered-on.
- The app correctly preserves and restores user or app state.
- The app preserves user jor app state when leaving the foreground and prevents accidental data loss due to back-navigation and other state changes. When returning to the foreground, the app must restore the preserved state and significant stateful transaction that was pending, such as changes to editable fields, game progress, menus, videos, and other sections of the app or game.
- When the app is resumed from the Recents app switcher, the app returns the user to the exact state in which it was last used.
- When the app is resumed after the device wakes from sleep (locked) state, the app returns the user to the exact state in which it was last used.
- When the app is relaunched from Home or All Apps, the app restores the app state as closely as possible to the previous state.
- On back keypresses, the app gives the user the option of saving any app or user state that would otherwise be lost on back-navigation.

Stability

- The app does not crash, force close, freeze, or otherwise function abnormally on any targeted device.

Performance

- The app loads quickly or provides onscreen feedback to the user (a progress indicator or similar cue) if the app takes longer than two seconds to load.
- With StrictMode enabled, no red flashes are visible when exercising the app, including during game play, animations and UI transitions, and any other part of the app.

SDK

- The app runs on the latest public version of the of Android Platform without crashing or loss of core function.
- The app targets the latest SDK by setting the targetSdk value to minimize the use of any platform-provided compatibility fallbacks.
- The app is built with the latest SDK by setting the simpleSdk value.

Battery

- The app supports power management features in Android 6.0+ properly. In the case where core functionality is disrupted by power management, only qualified apps may request an exemption.

Visual Quality

- The app displays graphics, text, images, and other UI elements without noticeable distortion, blurring, or pixelation.
- The app provides high-quality graphics for all targeted screen sizes and form factors.
- No aliasing the edges of menus, buttons, and other UI elements is visible.
- The app displays text and text blocks in an acceptable manner.
- Composition is acceptable in all supported form factors.
- No cut-off letters or words are visible.
- No cut-off letters or words are visible.
- No improper word wraps within buttons or icons are visible.
- Sufficient spacing between text and surrounding elements.

Data

- All private data is stored in the app's internal storage.
- All data from external storage is verified before being accessed.
- All intents and broadcasts follow secure best practices.
- Intents are explicit if the destination application is known.
- Intents enforce the user appropriate permissions.
- Intents that contain data and payload are verified before use.
- No personal or sensitive user data is logged to the system or app-specific log.

App Components

- Only application components that share data with other apps, or components that should be invoked by other apps are exported.
- This includes activities, services, broadcast receivers, and especially content providers.
- Always set the android:exported attribute explicitly, regardless of whether or not you export any of your application's components.
- All application components that share content with other apps define (and enforce)

appropriate permissions. This includes activities, services, broadcast receivers, and especially content providers.

Networking

- All network traffic is sent of SSL.
- Application declares a network security configuration.
- If the application users Google Play services, the security provider is initialized at application startup.

Libraries

All libraries, SDKs, and dependencies are up to date.

Policies

- The app strictly adheres to the terms of the Google Play Developer Content Policy and does not offer inappropriate content, does not use the intellectual property or brand of others, and so on.
- The app maturity level is set appropriately, based on the Content Rating Guidelines.
- The app supports power management features in Android 6.0+ properly. In the case where core functionality is disrupted by power management, only qualified apps may request an exemption.

App Details Page

- The App’s feature graphic follows the guidelines outlined in this blog post. Make sure that:
- The app listing includes a high-quality feature graphic.
- The feature graphic does not contain device images, screenshots, or small tet that will be illegible when scaled down and displayed on the smallest screen size that your app is targeting.
- The feature graphic does not resemble an advertisement.
- The app’s screenshots and videos do not show or reference non-Android devices.
- The app’s screenshots or videos do not represent the content and experience of your app in a misleading way.

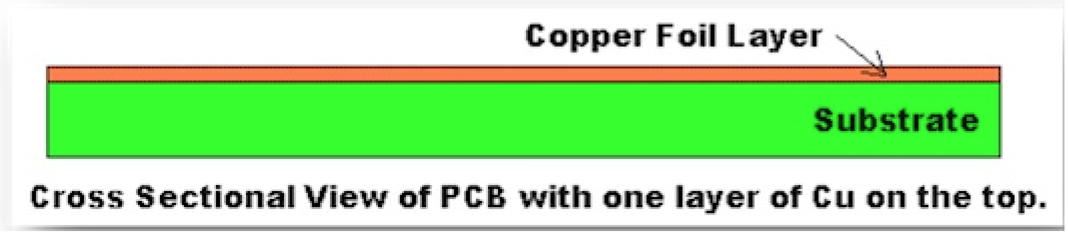
User Support

- Common user-reported bugs in the Reviews tab of the Google Play page are addressed if they are reproducible and occur on many different devices. If a bug occurs on only a few devices, you should still address it if those devices are particularly popular or new.

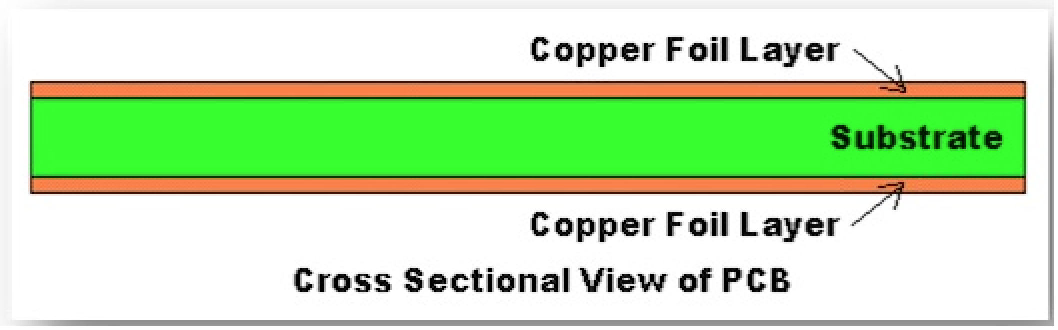
These are the software standards that we have found for developing an app specifically for the Android platform. We choose from the list of standards located on their android development standards website and selected only the ones that would apply to us. It is important that we follow these standards to ensure a quality software product and also if we ever want to make it onto their Play Store as these are their specific requirements.

4.2.5 PCB Standards

Every PCB they all have the standard they have 2 basic levels including copper foil layer and substrate layer according to Mahmoud Wahby from EDN.com. He also stated that " The main portion is a non-conductive (insulative) material (substrate) usually made from fiberglass, and epoxy. The substrate material used to separate layers comes in different thicknesses, from 0.005" to 0.038". Conducting layers consist of copper (Cu) foils" And those PCB they are divided into three type. First of all single layer it only have the copper foil layer on 1 side and the substrate (electrical insulator on the other side. This type is only using for 1 side this pcb which is really common nowadays. It is used for the home made PCB for example or non complex PCB layout.


Copper Foil Layer
Substrate
Cross Sectional View of PCB with one layer of Cu on the top.

Next is the double-layer PCB where the top and bottom have the copper foil layer and the between is the electrical insulator. This type of PCB was made due to the increase the complex of the PCB schematic structure.



The last type is the Multilayer PCB as same for the name. The Multilayer have many layer as much as it can go according to the source, it can be stack up to 32 layers which depend of the design of the circuit. The layer Stack up. And it go by other

Copper top
Inner 1
Inner 2
Copper Bottom.

Where the top and bottom of PCB is reverse for the signal meanwhile the 2 inner is using for the ground, and power planes. Each PCB design they have their own thickness standard including the copper layer.

- .031" (also .039" is common)
- .062" (most commonly used size)

- .093"
- .125"

As for our project we will using the pcb based on this standards which is from an article from Mahmoud Wahby EDN.com.

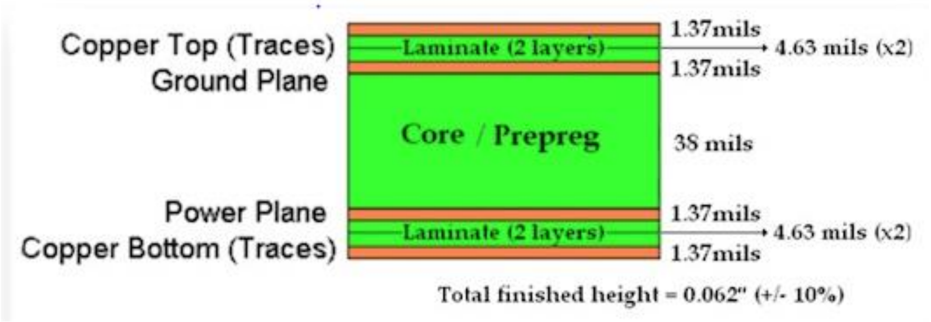


Figure 4 Layer PCB Example from EDN.com

4.2.6 Dog Body Temperature Standards.

The traditional method of reading a temperature of an animal is very invasive which is through the anus with a thermometer. The purpose of this temperature reader is to minimize the pain/discomfort of the animal and make it as easy as a click of a button without any hassle. As research shows, another accurate way to measure a dog's temperature which is non-invasive is at the armpit. Although it may not be as accurate as the rectum of the animal, it can be as long as a few variables are eliminated. The area in which the temperature is being taken needs to be clear of fur to allow for accurate results. The temperature needs to have full skin contact without the interference of fur and debris. For most dogs, the armpit should have the least amount of hair and with the harness, we should be able to have the temperature reader secured in a single place. Another variable that needs to be eliminated is disturbance. When utilizing the temperature reader, it should be suggested that the harness be in its most snug and fit position optimized for accurate results as well as the least movement as possible. With minimal movement, it allows for less error. With all these discretions directed to the end user, the final standard should be provide accurate results. As defined in the ethical constraints, we should always provide accurate information for the user as any faulty results may lead to extreme harm to the animal.

5.0 Project Design

The following sections and subsections contain all the information that is relevant to the design of the Smart Harness. The hardware is made up of an ATMega328P MCU, an MMA8451 Accelerometer, the SEN 11574 heart rate sensor, an SH-HC-08 Bluetooth module, the DS18B20 temperature sensor, and the battery. For the hardware all the schematics, pin layouts, and block diagrams to clearly explain our design will be shown and discussed. The software design and processes will also be contained in this section.

5.1 Hardware Design

All of the hardware design that went into the creation of the Smart Harness is described in this section. We tested each module and sensor individually, then connected everything for a final test. aside from the accelerometer. During our tests with the MMA8451, we ran into connectivity issues. These issues are most likely caused by a loose connection. Our first attempt at soldering proved unsuccessful. Future attempts will most likely solve this problem because as we move forward our soldering skills will improve.

5.1.1 Hardware Block Diagram

The hardware block diagram will show the details as much as possible about how the PCB will operate once it get turn on. Here is the brief description when the PCB was turn on. First of all, after the power is turn on, the current will flow into the MCU. Then, it will activate the Chip ATMega328P inside the MCU, then it will send the current and the voltage into the sensor to activate the devices including the temperature sensors (DS18B20) , heartbeat sensors (SEN 11574), bluetooth device (SH-HC-08), and the Accelerometer (MMA8451). After it get active DS18B20 will send the current temperature back to the MCU, as same as for the heartbeat sensor (SEN 11574), and the accelerometer will also send the signal of the location to the MCU by that the ATMega328P will process those signal. It will communicate with the bluetooth (SH-HC-08) and send the data to the application (software). And the process will repeatedly rapidly until the power was shut down

we also plan to developed for the the switch on and off from the application. By using two ways bluetooth control from SH-HC-08. This feature will allow us to control the MCU power without have to turn off the circuit by hand. Instead of one way transfer information, now we can actual command the the circuit to turn off or on. Moreover, this function is rather popular nowadays. There are many device switch on and off by bluetooth such as light, music, or speaker. However, this is another function we might brought up if we need more function for our project depending on the situation during our Senior design 2. We might actually develop this plan. This one can be really challenge for us to do since no one in our group members have experience with this function. The research in order to achieve this will need to be done properly.

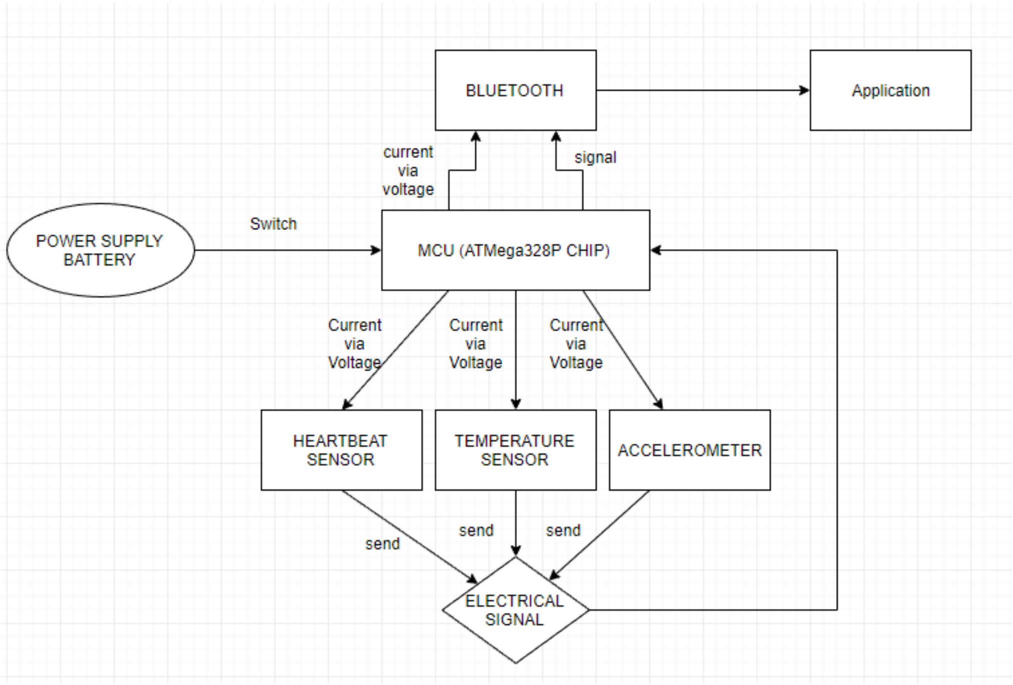


Figure: Dog's Harness hardware block diagram.

5.1.2 Design Overview

Currently, our hardware design consists of about six main components. These include the MCU ATMega328P chip, a heart rate sensor, temperature sensor, accelerometer module, bluetooth module, and the battery power supply. We will use the supply 3.3V and 5V from the MCU as well as 3 analog pins and 3 port pins.

Once we have acquired all the parts, we will design the connections between the modules, sensors, and MCU. After we have discovered the best design, we will test each of the modules and sensors independently. If all goes well with each component we will attach them all and run a final test. Those test will be discussed in a later chapter, but in this section all of the modules and how they are connected to one another will be discussed.

5.1.3 Microcontroller

The pin connections for the ATMega328P microcontroller are shown in this section. The figure below shows the pin layout of the ATMega328P microcontroller. The following table shows each of the pins and their usage in our project. We do have many unused pins, but originally we had intended our project to be just the starting point, and the unused pins give the smart harness the ability to add numerous future features such as a global positioning system. The most used pins were the analog and digital inputs that were used for our sensors.

One of the many reasons we chose this MCU was for the extra memory that it contained. The ATMega328 has 32 KB of memory onboard. This allows for numerous tasks to be done at once and gives us peace of mind concerning our memory allocation. Although this is a nice luxury to have, another benefit gained by selecting this microcontroller is the access to the Arduino support library. Connecting to the different modules was not difficult at all due to the abundance of open source code. This allowed us to connect and test very easily. When the final PCB is completed the testing should be just as easy. Overall the selection of this MCU has proven to be a good decision for our project.

Table of Pin Connections

Pin #	Connection	Pin #	Connection
1	N/A	15	N/A
2	N/A	16	N/A
3	DQ (DS19B20)	17	N/A
4	N/A	18	N/A
5	N/A	19	N/A
6	N/A	20	N/A
7 (VCC)	Vin	21	N/A
8 (GND)	GND	22	GND
9	N/A	23	DQ (SEN 11574)
10	TXD (SH-HC-08)	24	N/A
11	RXD (SH-HC-08)	25	N/A
12	N/A	26	N/A
13	N/A	27	SDA (MMA8451)
14	N/A	28	SCL (MMA8451)

Table List of Pin Will Be Used and Not.

5.1.4 Temperature Sensor Module

This describes the connection of the DS18B20 temperature sensor to the MCU. This was not particularly complicated as it was only necessary to connect it to the voltage, a digital input pin, and then all that was needed was to ground the sensor. A pull-up resistor of 4.7K□ was used to facilitate the connection.

The purpose of this is module is to take readings, and that information is being transmitted to the MCU to be stored and ultimately displayed on the mobile device. The temperature reading will be numbers, and these numbers are only expected to be as accurate as one decimal place. As a result, this information can be sent through digital means.

The module was then connected to the MCU by using a digital input pin. The pin used in the test, and the pin used in the schematic are different due to the fact that testing was done on a development board that accompanied the ATMega328P microcontroller. During the test pin # 2 was used. For the final design a digital input must be used, and for the ATMega328, the pin # 2 seems to be used for the bluetooth receiver RXD pin. As long as the pin chosen in our final design is a digital pin, than the temperature sensor should work properly.

5.1.5 Heartbeat Sensor Module

The connection of the SEN 11574 Heart Rate sensor module is similar to that of the temperature sensor. This module was only required to be connected to the voltage, ground and a pin. However, this sensor was connected via an analog pin.

The output of a heart beat is described by a graph. Unlike the temperature sensor, the input of a heart beat is “messier”. For this reason an analog pin is needed for the heart beat sensor to work properly.

This module was then connected to the MCU by using an analog input pin. The pin used in the test, and the pin used in the schematic are different due to the fact that testing was done on a development board that accompanied the ATMega328P microcontroller. During the test pin A0 was used. For the final design an analog input must be used, and for the ATMega328, the pin # 23 is pin A0. As long as the pin chosen in our final design is a analog pin, than the heart rate sensor should work properly.

5.1.6 Accelerometer Module

The MMA8451 accelerometer module has connectors that, for it to work properly, need to be soldered on. Once this has been done, there are four connections to be make. VCC must be connected to the power source and the circuit must also be grounded.

The other two connections are the ones that will be used to transmit the information. These are the SCA and SCL pins located on the accelerometer module. The information that is to be transmitted are X, Y, and Z coordinates. By using this change in these coordinates, we will be able to use this module to create a step counter for the smart harness.

5.1.7 Bluetooth Module

The SH-HC-08 Bluetooth module has only six pins placed on it. Of those six pins, four are needed to connect with the MCU.

VCC is connected to the voltage supply, GND is sent to ground. The other two pins are RXD and TXD. TXD is the transmitter pin, while RXD is the receiver pin.

The MCU has pins with the same designation. Thus the connection is TXD (accelerometer)-RXD (MCU) and RXD (accelerometer)-TXD (MCU). The information that this module transmits will be any data that travels from the mobile device to the MCU. This includes the heart rate, temperature, and pedometer data.

5.1.8 Hardware Schematics

The schematic below contains all the elements that will be included in the Smart Harness. This includes all the sensors, the bluetooth, the accelerometer and the MCU. All of the schematics were created using the Autodesk EagleCAD software. This will not be our true final design as there are still many issues to work out. For now this is a good representation of our progress and this will continue to evolve as we continue our project development.

Each of the components shown connected to the MCU. These may not be the same pins shown during our test phase, but this is due to the fact that we used the development board for testing. The schematic below is not using the development board that came with the ATMega328P MCU. Also, as the connection of the temperature sensor and heart rate sensor were near identical, the same schematic was used for both as the ability to find locate the schematic of the heart rate sensor was proving problematic.

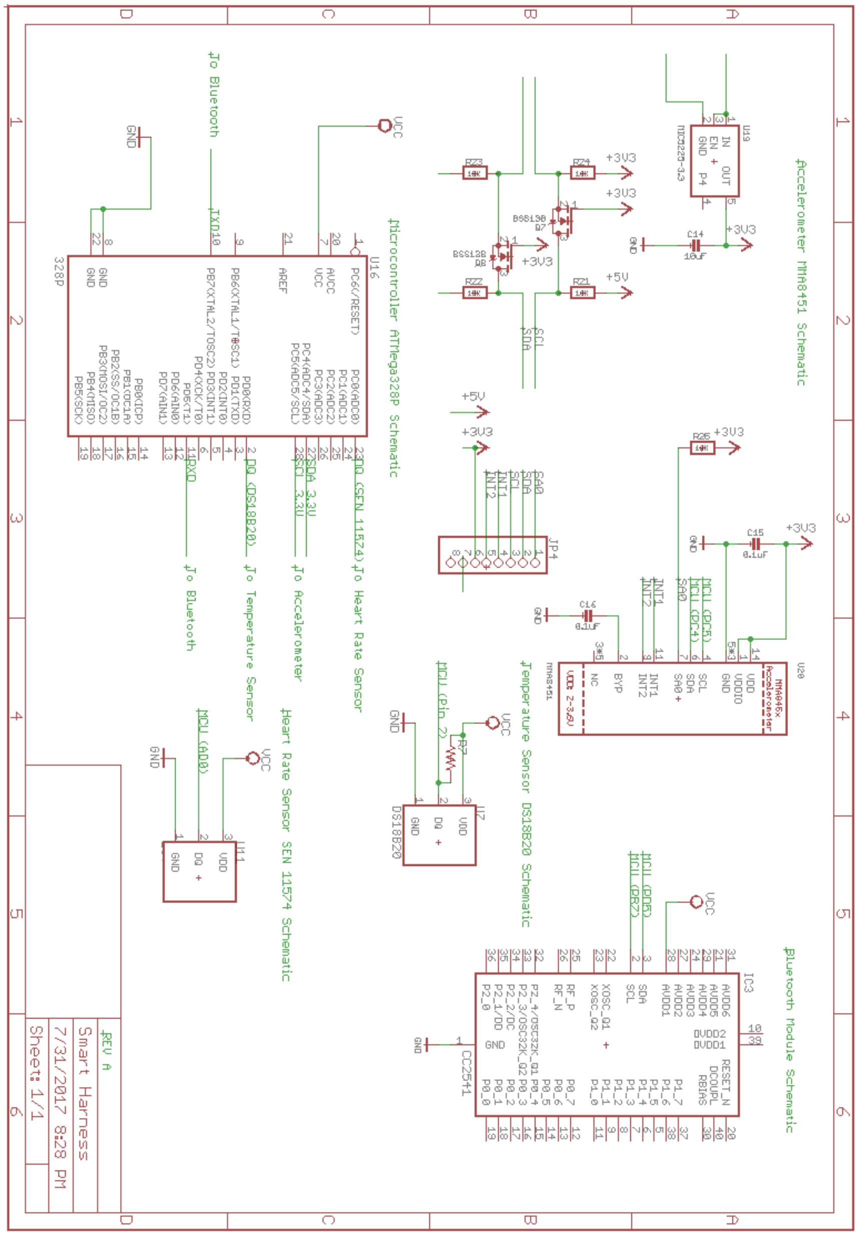

Accelerometer MMA8451 Schematic
Microcontroller ATMega328P Schematic
Temperature Sensor DS18B20 Schematic
Heart Rate Sensor SEN 11574 Schematic
Bluetooth Module Schematic
To Bluetooth
To Temperature Sensor
To Accelerometer
To Heart Rate Sensor
U16
328P
U19
U20
U7
U11
IC3
CC2541
DS18B20
JP4
VCC
GND
+3V3
+5V
REV A
Smart Harness
7/31/2017 8:28 PM
Sheet: 1/1

Figure: Dog’s harness schematic.

5.2 Software Design

All of the software design that has gone into the creation of the Smart Harness Android application is described in this section. We needed to design the blueprint of how the application should work and the user experience / user interface that needed to be designed. The flow of the the application and how it should work is described as well as mock ups of the look and feel of the application. The software design is a big part of defining the product as a symbol and company.

5.2.1 Software operation flow

During the software, one of the first goals is design the flow of the application so that there is a general idea of how the application is to be coded and it gives a good outline for the developer to keep certain things in mind as he/she implements functionality and interfaces. To give a brief description of what this flow represents, we will go through each step. First you will launch the application by clicking on the application icon. Next, if this is the first time that the user is opening the application, we need to configure the bluetooth to connect with the harness but if not, we can go ahead and move directly to the dashboard of the application. If it is the user’s first time and decides not to configure the bluetooth as this time, go ahead and close the application or if configuration is successful, move directly to the dashboard. From the dashboard, there will be some data displayed but furthermore, the user will have the option to take measurements or view records. If the user decides to take measurements, they will be directed to the next screen which they can choose to either take the critters heart rate or temperature. If they decided to view records instead, they will be directed to a screen that will give the option to either view heart rate history or temperature history. Anywhere there is a double edged arrow, it represents the ability to either to backwards or forwards from one screen to another.

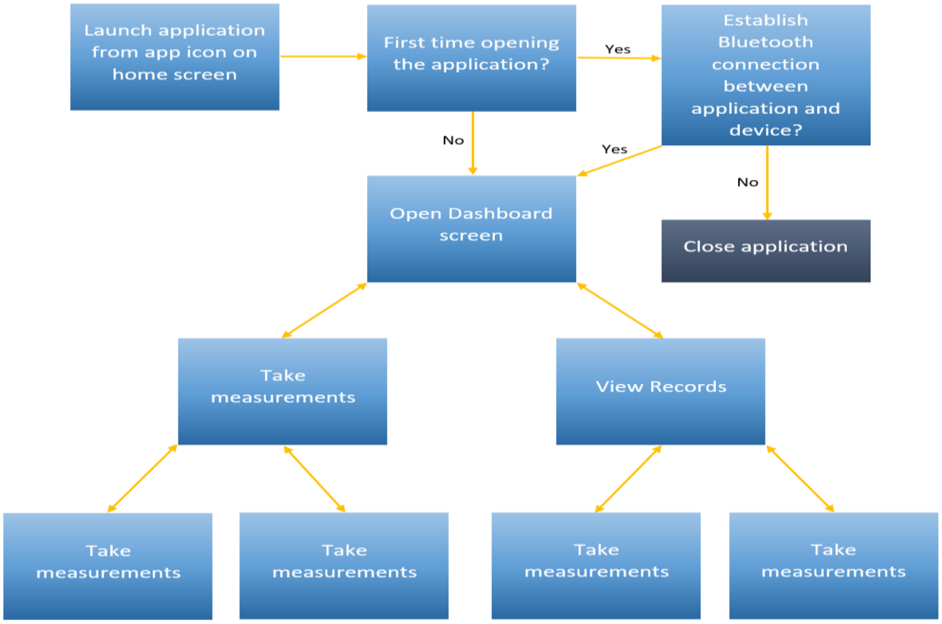


Figure 38 Diagram of UI flow

5.2.2 User Experience / User Interface

The purpose of the user experience / user interface design is to define the requirements for the applications look and feel and how the user will interact with the application. This part is of the design process not only includes the ability to design the applications interfaces to be appealing, desirable and aesthetic, but also a user friendly experience when inside the application which requires market research to collect data and input on how other companies are doing it or what is favorable to the consumer. This part we will go over what those requirements are and why the requirements are defined the way that they are.

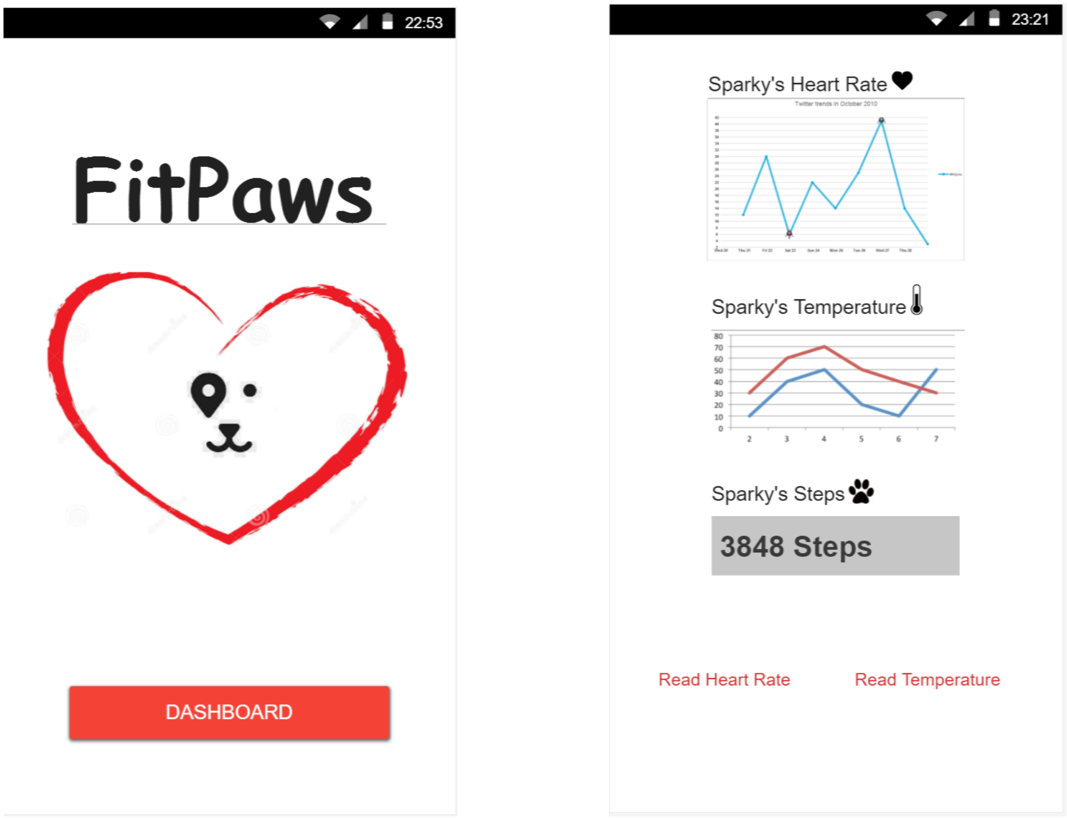


Figure 39 Software Front Page and Dashboard

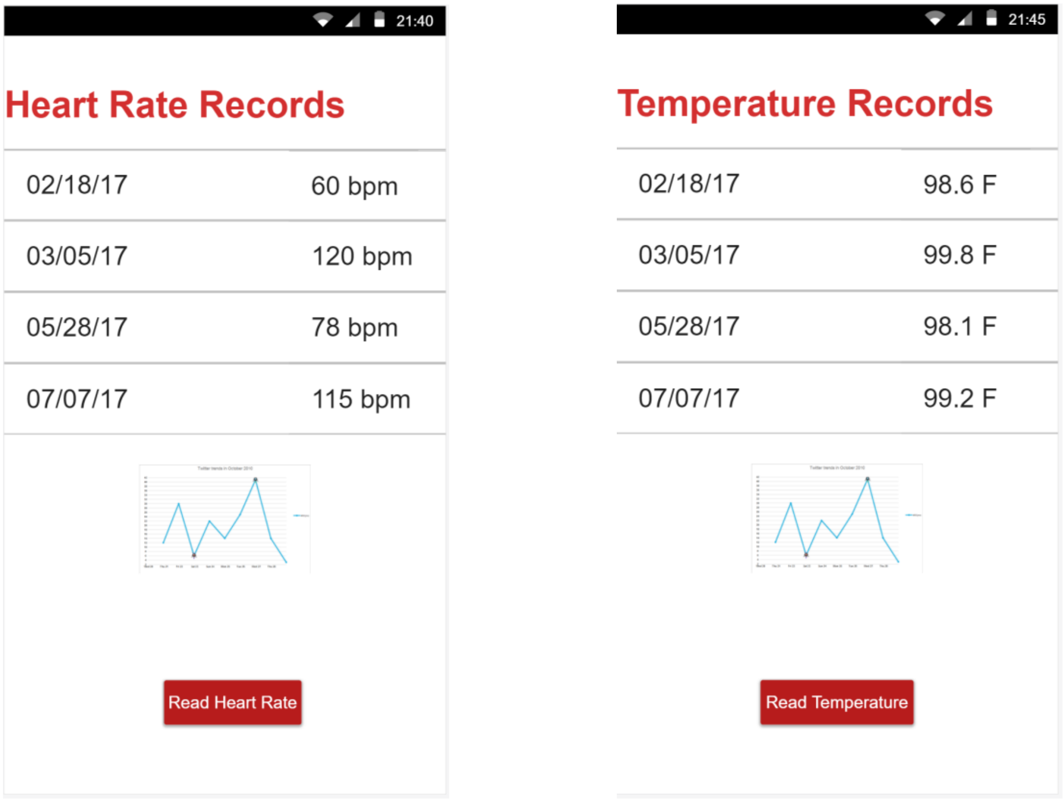
5.2.2.1 Front Page

The objective here is was to create a logo and brand name for the product. We needed to appeal to the masses with something that is simplistic and and welcoming. This was done by the cartoon logo of the dogs face with the heart around it. If we were to throw the image on an ad by itself, someone would look at it and take a pretty good guess on what it was for and that was the objective; to make the logo easy to understand and be appealing. The name so far is not set in stone because there some concern with it not telling the whole story. The part of the name “fit” comes from the word fitness. This application and harness is designed to be a health tracker for a canine and provides the ability to take certain vital measurements that would be useful in tracking a dog’s overall health. Putting the word “fit” into the name is a grey area because are we are not necessarily a fitness tracking application but a lot of our key components can attribute to a canine’s fitness.

Regardless, this is a talking point that can be addressed later on. Front screen is displaying our logo and name for obvious reasons and gives you a quick easy access to the dashboard.

5.2.2.2 Dashboard

The dashboard will and should be the largest traffic point for a user. They open up the app and they will go to the dashboard, they take measurements and come back to the dashboard, they view past records they come back to the dashboard. So the dashboard must have optimal functionality integrated into the screen so that the end user can have a hassle free interaction. The less clicks the better! The Dashboard needs to display all pertinent data and should be able to access anywhere in the system from here. It will display the last recorded heart rate and temperature, it will display line graphs for both heart rate and temperature, and it will give access to reading heart rate, reading temperature, and settings/profile.



Heart Rate Records

02/18/17	60 bpm
03/05/17	120 bpm
05/28/17	78 bpm
07/07/17	115 bpm

Temperature Records

02/18/17	98.6 F
03/05/17	99.8 F
05/28/17	98.1 F
07/07/17	99.2 F

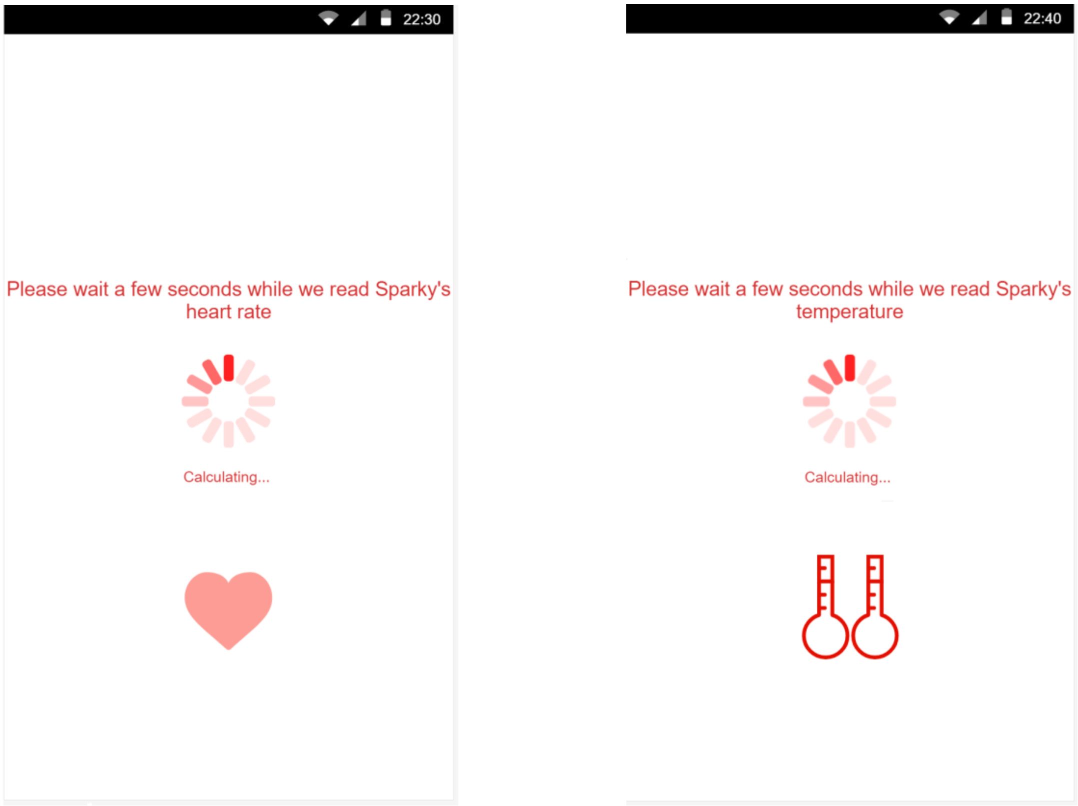
Figure 40 Reading Heart Rate and Temperature

5.2.2.3 Read Heart Rate

From the dashboard the user will readily be able to access the read heart rate functionality. On command, you can press this button and a new screen will come up showing you that the read heart rate functionality is actively in process. This screen can be found below. The purpose of this the loading screen below is to prevent the user from interrupting the process because once the process is initiated, it will not be able to be stopped. It also gives a good place to put some cartoon action or neat gif to display to the user to make it more physically appealing. One it is done reading, it will display on the read heart rate records screen as the latest with a date and time and bpm. It will also add itself to the line graph.

5.2.2.4 Read Temperature

As far as the user interface/user experience aspect of the functionality, it will be very similar to the read heart rate interface. It will display records with the latest reading displayed on top with a line graph of all the the past readings. The read temperature button will also take you to a loading screen that you can view below that will also prevent the user from further interaction. The key difference here is the time it takes and because reading the temperature may or may not take longer, there might not be a need for a loading screen but for consistency sake, we will add it now until we have the full functioning reader to determine the need for the loading screen.



Please wait a few seconds while we read Sparky's heart rate

Calculating...

Please wait a few seconds while we read Sparky's temperature

Calculating...

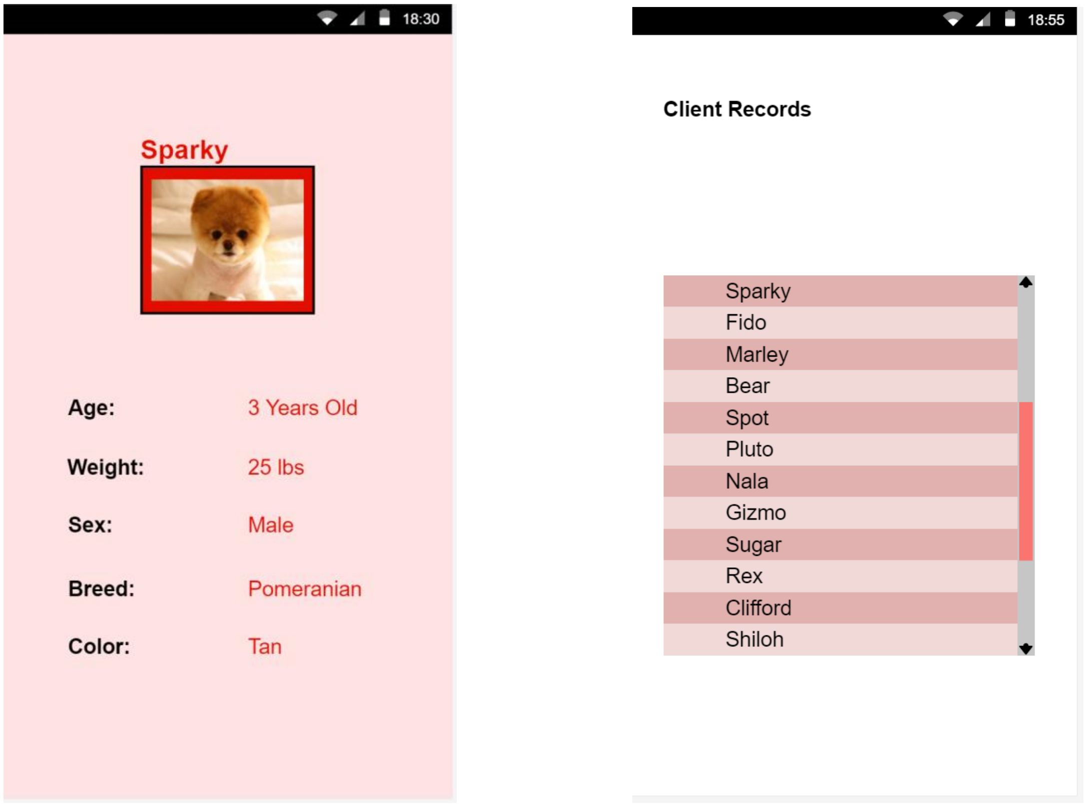
Figure 41 Heart Rate and Temperature Load Screen

5.2.2.5 Dog Profile

The dog profile is important more for the veterinarian than it is for the regular user. The way this works is the user will be able to set up a profile for their canine by adding a picture, entering the name, date of birth, weight, sex, breed and color. On the system settings page you will also be able to add a name for the owner of the pet, the contact information of the owner such as address, email and phone number, and any additional notes. This information can be updated at any time. How it is important for the veterinarian is that when there is interaction between doctor and patient, the veterinarian can add the user's pet information to their repository through a click of a button. This will help the use case of when the patient sends a temperature reading to the veterinarian with concerns, the veterinarian will have all the information readily available to contact or make notes on the patient. You can see how the profile will look like below. Another reason why the profile of the pet would be important is for future implementation of the product. If we ever decide to expand and have a user to user interaction and be able to add "pet friends" and basically turn it into a social media platform for pets and pet owners.

5.2.2.6 Doctor's Cient Records

Client records is the pinnacle of the doctors functionality. It will act as a repository for the veterinarian and a communication front to directly interact with the patient. For example if the user shoots over a heart rate reading, there will be a notification that appears on the screen from the app as well as an icon next to the pets name similar to how messaging or emails work. From there the veterinarian can then click into the animal's profile and access the message to see what the issue is. This should be a seamless and effortless interface to not only improve efficiency but also not interfere with the veterinarian's daily responsibilities. There will also be the ability to silence notification for a period of time due to after hours or being in surgery and anything else that requires no disturbance. The doctor's interface is clearly more robust but there is a general mock-up of the interface below.



Sparky

Age: 3 Years Old

Weight: 25 lbs

Sex: Male

Breed: Pomeranian

Color: Tan

Client Records

Sparky
Fido
Marley
Bear
Spot
Pluto
Nala
Gizmo
Sugar
Rex
Clifford
Shiloh

Figure 42 Dog Profile and Doctor's Client Records

First are foremost we need to emphasize that these are general mock-ups that are used to take ideas in our head and drop them into something physical so we can clearly point out the negatives and positives and things we need to change, add, or remove. These mock-ups are by no means the end all be all to our user experience/user interface. So to clarify what is going on in these images, the first thing we need to do is determine a theme and something that will give people the ability to identify us from everyone else, basically a brand. Target has a red color scheme, Walmart has blue, Publix has green, and this is what they use to uniquely separate themselves from other people so we are going with red. Also, we want to point out and emphasize simplicity in our design. In our mock up we have the most minimal of data, information, and “busy-ness” which may need some additional modifications but the main concept is to have a modern and simple look. This is an application for a user’s companion and it is best to keep it as simple as possible. If we bombard the user with too much statistics and graphs and data it may be a deterrent to many users. So to accommodate this we basically put all the necessary information and nothing more. The user has the ability to read the heart rate and temperature of the animal on command and then we will display all previous readings up until the most current one. We will also provide a line graph for the user to be able to point out any oddities or sudden changes so that they can quickly notify their veterinarian. The daily steps of the canine will be simply displayed on the dashboard and the user can view them directly as there is no need to read steps since it does it on its own. Another additional feature that the software will provide is the ability to send their veterinarian any recent reading or all records if there is a need. There are a few situations where this will occur which are, when a patient meets at veterinarian for the first time and needs to send the veterinarian all past records, or if there is an oddity or concern and the patient wants to immediately ping the veterinarian for questions. These are just a few reasons on how and why this could come in handy. This functionality though, is a perfect segway into the next user interface/user experience which is that there will be two different accounts that will be available, one for the patient user and one for the veterinarian user. The two users would have different needs and uses from the product so it would only make sense to make two different interfaces. The users side will provide the ability to perform all the functions it needs where the veterinarian side will be more like a repository and communication point. Also, this sets up for a great marketing and financial scheme for the product. You can price point it differently for each user.

5.3 Design Conclusion

In the conclusion, the design will including 2 parts hardware and software design. The hardware will design based on the 6 main component including ATMega328P chip, Temperature sensor DS18B20, Heartbeat sensor SEN 11574, Accelerometer (MMA8451), Bluetooth module (SH-HC-08), Battery 6 AA Alkalines. Secondly, the design is Software. The software (Android Studio) will build an application for android phone.

6.0 Prototype Harness.

Our group build the prototype using all the component which is on the breadboard like the picture below. The breadboard for the the prototype will be too many wire. However, once we get to the senior design 2 we will eliminate all unnecessary component on the arduino such as extra port, some LED, and the reset button. After that, we will put everything together including the chip, VCC, ground, analog, connection ports, sensors, and bluetooth module.

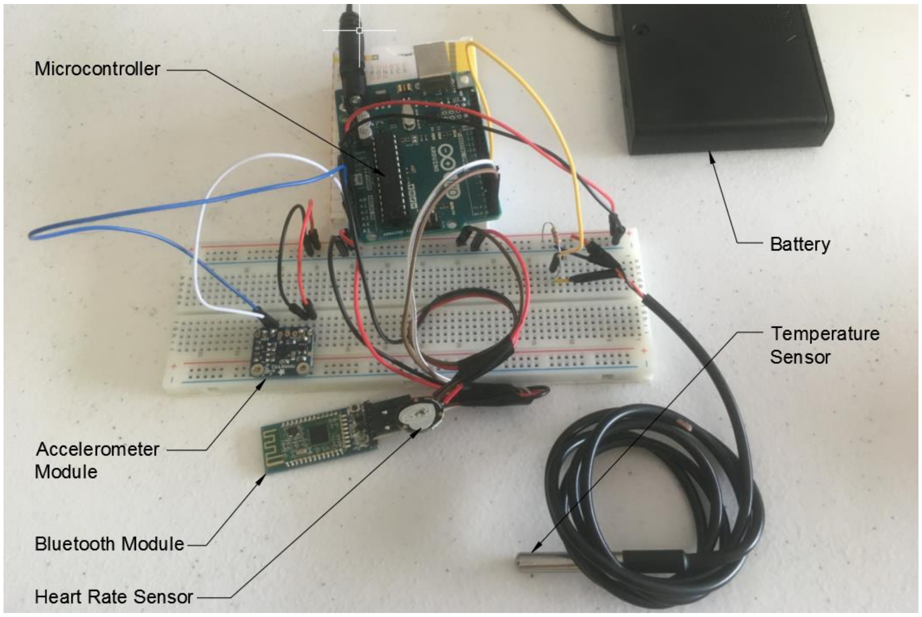
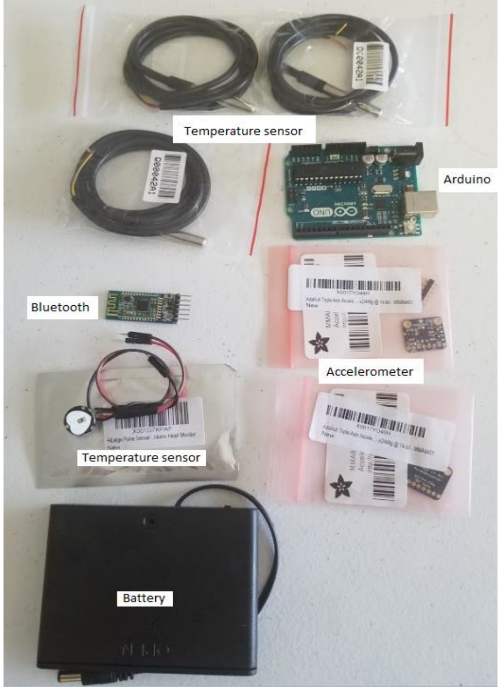


Figure Prototype of Dog's Harness Circuit

6.1 PCB Design.



The design for the PCB will divided into 2 parts including the component will be integrated inside of PCB and the component which require the connection. The microcontroller is the chip ATMega328P, the the MMA8451 accelerometer, and the bluetooth SH-HC-08. Those three component will be integrated on the PCB for senior design 2. However, the Temperature sensor and the heartbeat sensors will be relocated to the different location of the dog body in order to get the accurate reading.

Figure : Components for the PCB design

6.2 PCB, Battery, Sensor Location.

The location of the PCB battery, and the Sensor is very important for the design. It will determine the effectiveness reading will depend on the location of the PCB, sensor battery for the harness belt. As for our group's decision, The circuit and the battery will be place on top of the belt, the temperature and the heartbeat will run the cable to the chest of the dog where is less dog's fur there so that we will more accuracy reading for the sensor.

6.3 Potential Prototype Design Issue.

Even though our design seem like good. However, there are couple potential design Issue. First of all the design for PCB due to the theory of building the PCB, we still not know how the acutal current and voltage will be drain out of the battery. Secondly, the bluetooth connection might not work up to 110m like the datasheet statement. Next, the connection of heartbeat and temperature need to be connected very secure. If it is not, the sensor will send the wrong reading to the MCU. Lastly the most biggest problem is the integration for the PCB. The location of PCB sensor device, battery might not work like we think due to the activity of the dog like running, and standstill. To sum up all the design potential potential, they will need to be test in the physical condition after our PCB is done during our the senior design 2.

6.4 Actual Cost.

The initial cost and the actual cost are very different. During this time, our group had ordered all the component We are estimate the actual cost after the prototype is build up. The table below will display the cost unit by 1 for our hardware

Actual Cost for testing.		
Component	**Price**	**Vendor**
MCU.	$ 20	Arduino.com
Heartbeat sensor.	$ 2.40	Amazon.com
Temperature Sensor.	$ 6.99	Amazon.com
Bluetooth.	$ 7.99	Amazon.com
Accelerometer.	$ 8.98	Amazon.com
Battery.	$ 6.99	Amazon.com
Total	$ 53.35	

Table Actual Cost for Hardware Component.

7.0 Testing.

The testing is very important before we will actually build the PCB and the software for this project. The testing will divided into two parts hardware and software testing. The hardware testing will involve with breadboard and Arduino software from online which is open source and the guideline for the component. All the hardware's code for pcb is provided by the Vendor, Manufacturer, or some outstanding Individual. All the credit for our test will go to them. As for the software testing we will doing the simulation test and physical testing.

7.1 Hardware Testing.

The hardware testing will break down into 4 section temperature, heartbeat, bluetooth, and the accelerometer. Each test will have the code open source which is provided either by github.com, manufacturer manual, or some individual website, and the code will be installed inside of Arduino IDE.

7.1.1 Temperature Sensor Testing.

The temperature testing will involve with three parts involve with hardware coding in C, temperature component installation, and the display temperature. The testing example was made on www.naylampmechatronics.com. It is a website store where they sale all the electronic component.

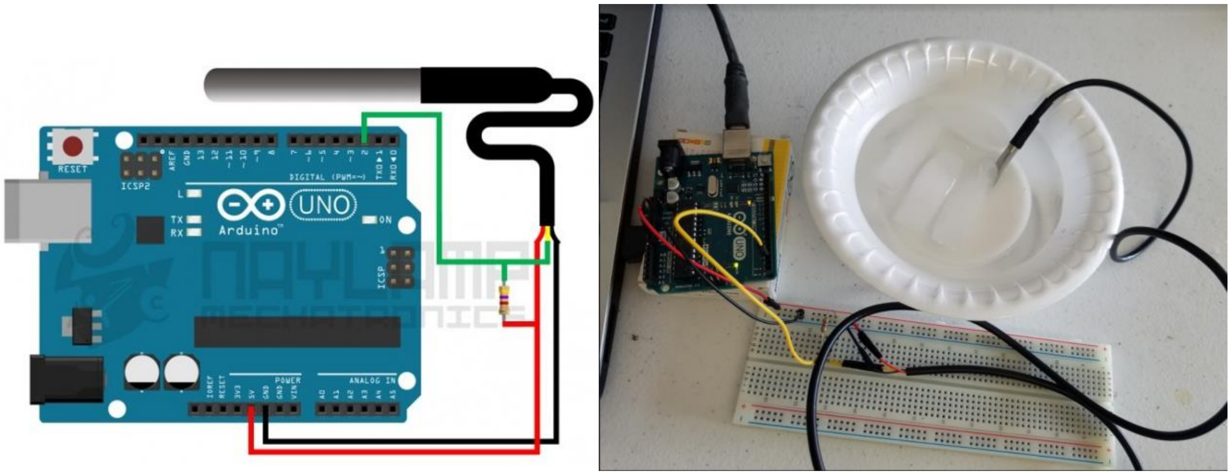


Figure Temperature Example Testing (Left) and Our Testing (Right).

7.1.2 Heartbeat Sensor Testing.

The heartbeat Sensor testing is provided by pulsesensor.com. It also divided into two parts hardware installation and C code. We followed the instruction by the manual from the website by Joel and Jury. the C-code they also provided on their github website. During the testing, we make and follow the connection between Arduino Uno and the heartbeat sensor. As a result, we are success in the testing.

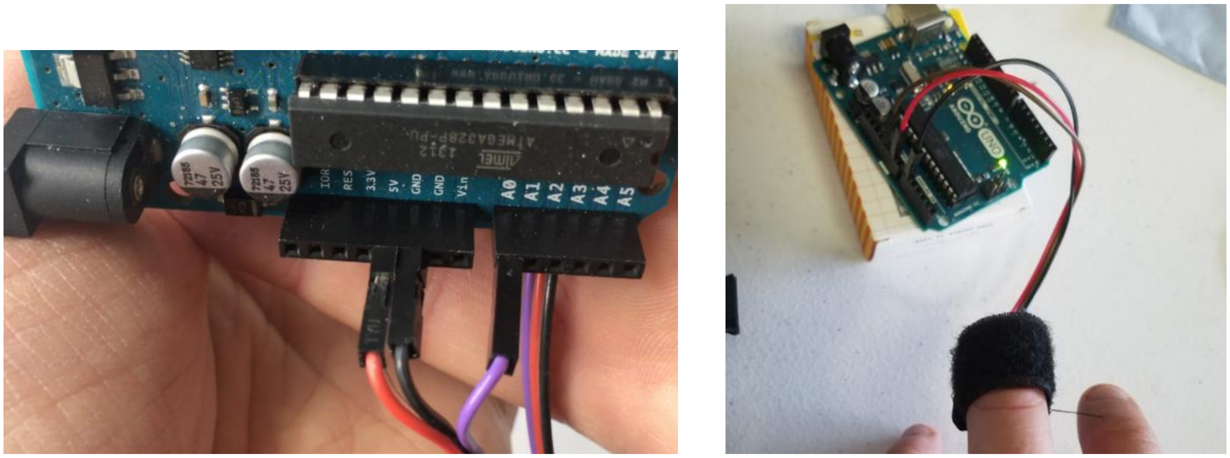


Figure Joel and Jury Heartbeat Connection (Left) and Our group heartbeat connection (Right)

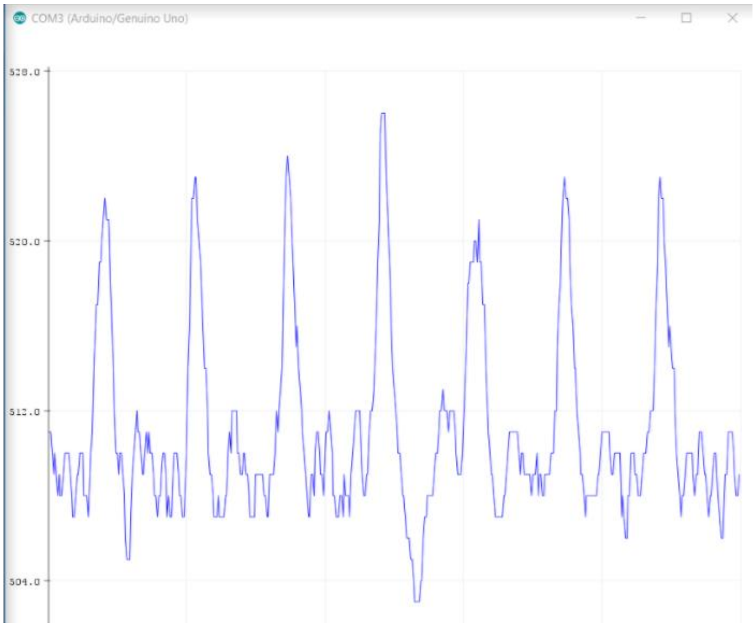


Figure The Heart Rate Reading of Hai Nguyen One of Our Group Member.

7.1.3 Accelerometer Sensor Testing.

This test is also same as the heartbeat testing. The test was provided by Lady Ada. it is including three parts of test. First of all it is the component adafruit MMA8451 need to get soldering, and it was shown inside of the manual from cdn-lean.adafruit.com. The second part is the c code which provided also by Lady Ada on github.com. Lastly, it is the hardware wire installation.

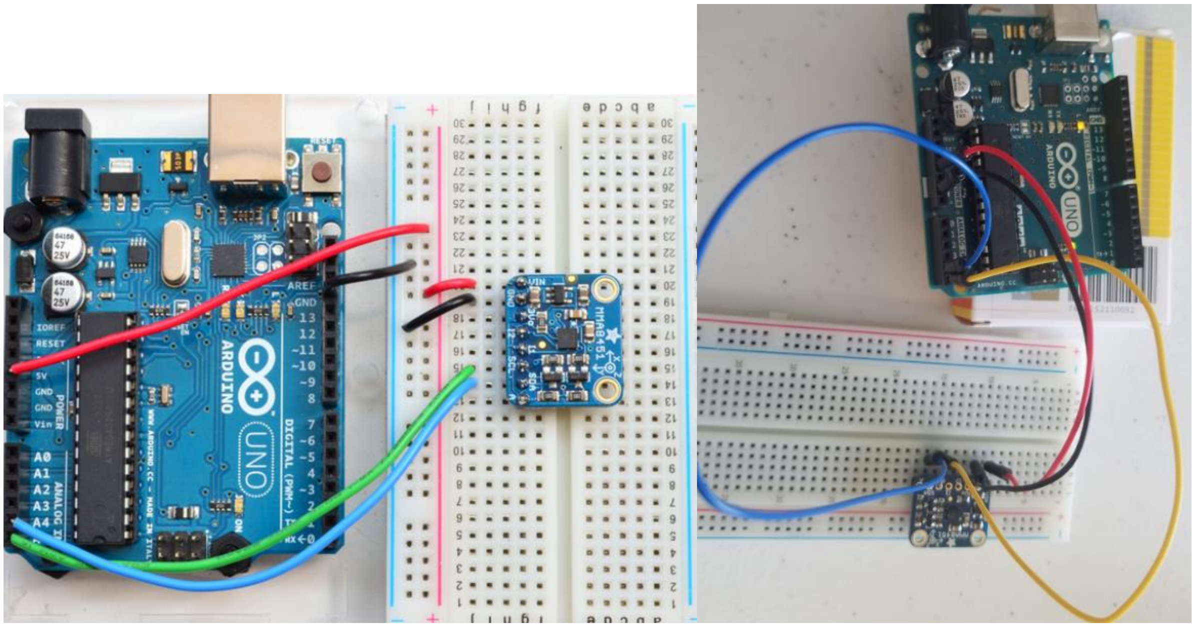
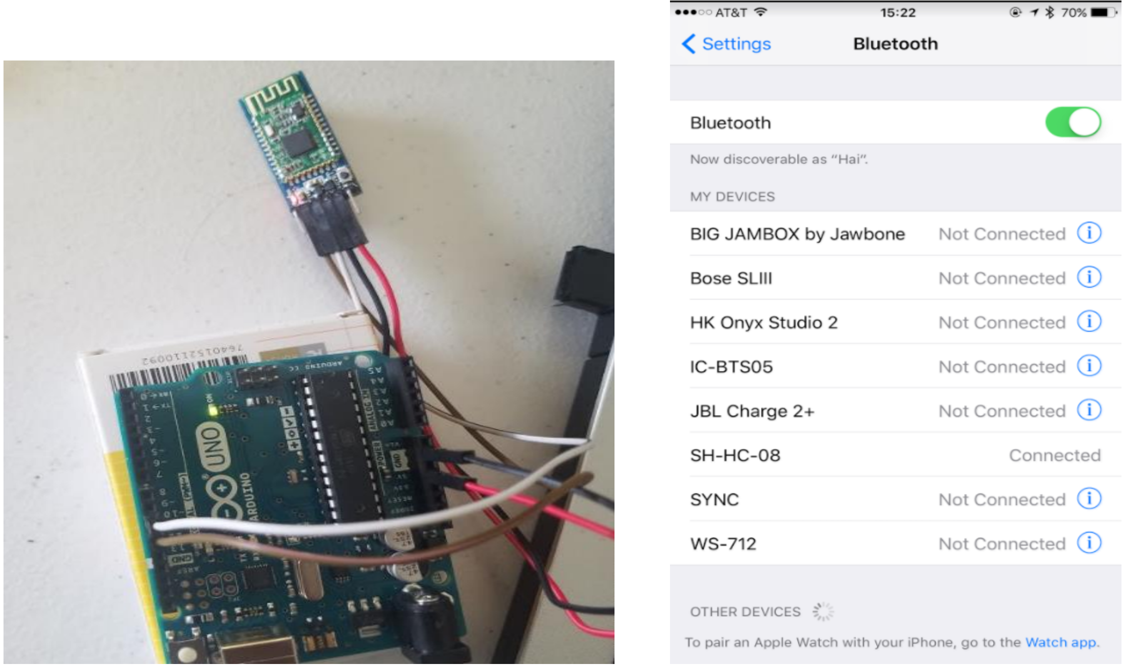


Figure Ladyada's Testing (Left) and Our Testing (Right)

7.1.4 Bluetooth Testing.

The bluetooth testing involve 3 parts. Hardware installation which is wire connection for the RX, TX, GND, and the Vin. this test is provided by Module Force at http://moduleforces.blogspot.com the company provided the code. Also, they sale the product. The c code can be found inside of their manual. The special about this test is that we need to download the BLE Scanner for Android devices and Light Blue for IOS devices.



Bluetooth

Bluetooth

Now discoverable as “Hai”.

MY DEVICES

BIG JAMBOX by Jawbone	Not Connected
Bose SLIII	Not Connected
HK Onyx Studio 2	Not Connected
IC-BTS05	Not Connected
JBL Charge 2+	Not Connected
SH-HC-08	Connected
SYNC	Not Connected
WS-712	Not Connected

OTHER DEVICES

To pair an Apple Watch with your iPhone, go to the Watch app.

Figure Group 15 Bluetooth Hardware Connection and Iphone connection.

7.2 Software Testing.

After the hardware testing, our group also doing the software testing. However, the testing is involved with all the simulation based on all the components and time. It also conducted on the functionality of each component. However it is only for the simulation but with no the real application yet because of developing the actual application will take long time.

7.2.1 Simulation Testing.

Simulation testing will involve interaction between the phone application and the physical components of the smart harness. We will have to run through a series of test and experiments under numerous conditions to ensure a consistent and accurate result. These test will need to be conducted both before and after full development of the product. The first testing done before is to ensure that the application is capable and compatible with our hardware components and our testing after full development is to run through the test as a end user and how they would use the product. All these test will need to be recorded and documented properly. Each test will be ran at a minimum of five times.

7.2.1.1 Bluetooth Simulation Testing

For the first round of testing, we will need to make sure make sure we build an interface to be able to interact with the Bluetooth. As mentioned before in the flow of the application interface, the first time the user is greeted it will be with an welcome screen and then immediately asked to connect to the bluetooth. The first test to be performed is to to make sure the button that we display to sync and connect to the Bluetooth is functional and actually makes a connection with our Bluetooth component. From that point on the user may or may not keep their Bluetooth on at all times but we know for sure that the user will not always be next to the smart harness so the hardware component as well as the software application needs to be able to sync automatically when within range or display another button under settings to be able to connect again if need be. We will need to test the range of the bluetooth and be sure to document it for user knowledge. After the connection to the Bluetooth component is secure and working properly, we will able to continue and test the rest of the components.

7.2.1.2 Heartbeat Sensor Simulation Testing

After the connection to the Bluetooth we will be able to communicate to the hardware and be able to channel all the data we need to present it to the interface. The first component we would start with is the heartbeat sensor. There will be a button on the screen to manually measure and acquire the heartbeat reading at any giving time. So the first test will need to be to make sure that when we click that button, accurate readings will come through and be presented and recorded to the application. This will be in a controlled environment on a human without the harness and then on a canine with the harness on. This way we can make sure that we can get expected results and ensure that the readings are accurate. After that we will be able to perform testing with the harness on the dog while it is active (ie. jumping, running, rolling, walking around) to see the expected results from an end user’s point of view. What we should is expect is that the heartbeat will be higher than that of when the dog is not active and moving around but another variable we are testing for is movement and making sure that it is even possible to get accurate readings when there is movement.

7.2.1.3 Temperature Sensor Simulation Testing

Just like the heartbeat sensor, we need to perform readings in a controlled environment to ensure expected as well as accurate results. The first round of testing will also be on a human without the harness as well as a canine with the harness on. The variable we have to account for that we found out through hardware testing is that the temperature sensor takes the temperature of the metal casing surrounding the sensor. So if there are temperature changes, we have to wait for the whole casing to reach the expected temperature to get an accurate reading. After those test are performed, testing of the harness on the dog while it is active will need to be conducted. This will include it running around and jumping up and doing what dogs do. These test are important because the button on the interface will allow manual reading of temperature at any given time so if they are able to do it, we need to test for it. The application will also need to display this information and record it accurately.

7.2.1.4 Accelerometer Simulation Testing

The accelerometer testing may be easiest or the hardest test of them all. The way the interface as been designed is that the step counter will always be displayed on the dashboard with the animal's current step count. This means that the accelerometer will always have to provide data or at least store it for the application to access this information at all times. One thing we need to consider is that the phone will not always present or in proximity of the harness so when it is, the step count needs to be updated as soon as there is a connection made. The first series of test will obviously be to use the accelerometer while the application is connected through Bluetooth and make sure that it is providing an accurate step count. We can do this by attaching this to a human and a dog with the harness on and taking steps and seeing the results on the application. We also have to take into account that a human's steps will be different than a dog's steps so there will need to be some calibration and algorithm generation to differentiate the two. Then the second series of test will be to put the harness on a dog while it is active and calibrate the step counter when the dog is jogging, running, or jumping around. All these steps will need to be accurately displayed to the application interface.

7.2.1.5 Battery Simulation Testing

There will need to be some testing wrapped around the battery and how it is displayed to the application. As most applications, there will be a status bar to show you the battery life of the power source. This is a time constraint as we will not see changes in the battery life for a decent amount of time because there is no way to immediately see a difference in power. One way around this testing to start off is to put in batteries that we know are half full or getting low and make sure it displays correctly in the application and if the batteries are indeed getting low we will need to display some type of warning to the screen notifying the user that the battery life is dwindling and is in need of changing or recharge. Although this is a workout around to get immediate results, still need to make sure that the life of the battery is displayed accurately in the application and the proper notifications are presented.

7.2.2 Physical Testing

The physical testing will be conduct in the senior design 2 before our group will do the demo at the end of the semester. The physical testing will work on all the combination together including hardware and software which will be install on an android phone. It also involved with the belt with the PCB, battery, and sensors will be installed inside of the dog's harness.

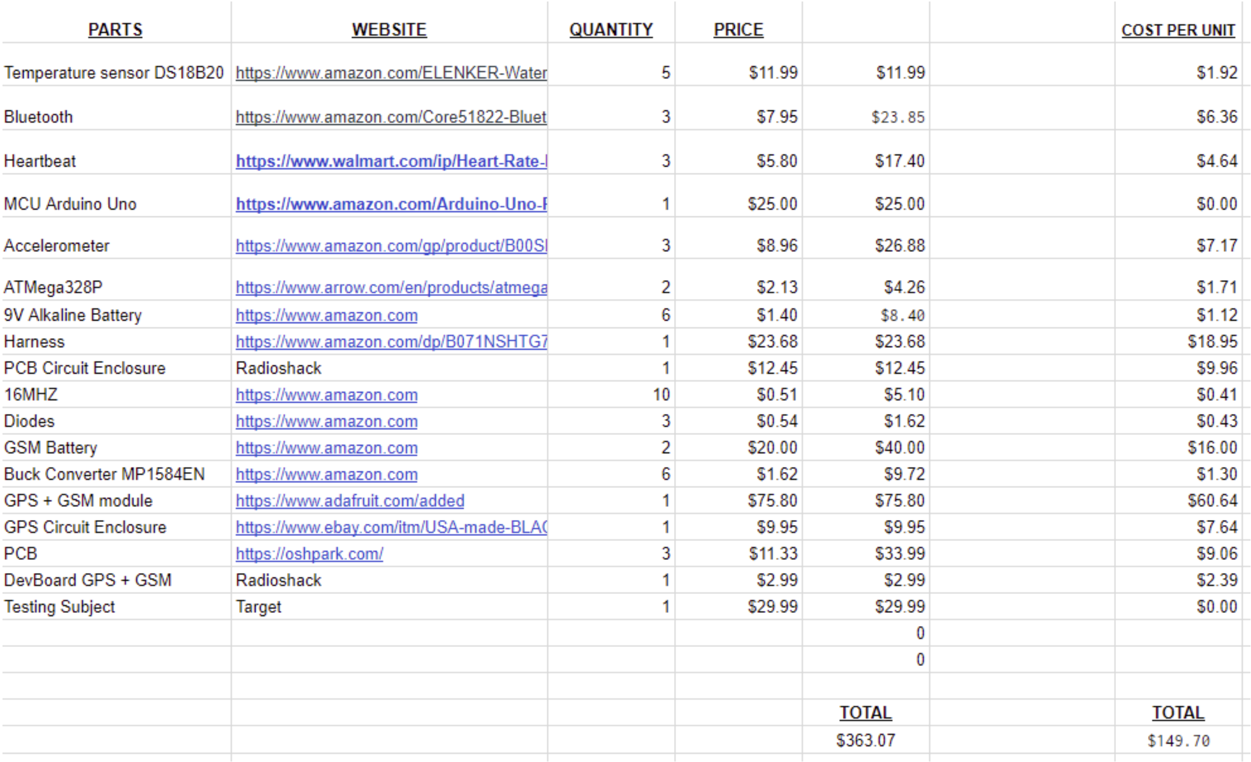
8.0 Administration

This senior design course is not just about engineering, it is also about learning how to manage a project on a larger scale than anything we have done so far in our college career. During this semester we learned how to balance a budget based on our project funding and expenditures. We also had to set a schedule and follow it. These are tasks any real business or employer would expect an employee to be aware of. In this section all of the budget and scheduling information is laid out in detail.

8.1 Estimated Project Budget and Financing

Below is a table of the budget we have decided upon for the senior design project. The total costs of this budget will be divided equally between the three members of the team. Once all the parts have been ordered as well as the final PCB layout is decided we will have a better idea of what the total costs will be. As of now, this is just an estimate of the total costs to be incurred and is a good measure of what each team member will need to contribute.

Keeping the costs as low as possible while acquiring good quality components is vital for this project. as one of the market specifications for the Smart Harness was to keep our costs below $175.00 for a single harness. Ideally we will exceed this expectation and keep the costs even lower than that, but for estimate purposes we decided to stick with this number for now. The most expensive component may turn out to be the microcontroller chip, as it is the control center for the hardware design. This is acceptable as most, if not all, of the information will pass through this component. The harness itself may prove to be an expensive piece of the puzzle as well, but hopefully research will lead us to an affordable solution.



PARTS	WEBSITE	QUANTITY	PRICE			COST PER UNIT
Temperature sensor DS18B20	https://www.amazon.com/ELENKER-Water	5	$11.99	$11.99		$1.92
Bluetooth	https://www.amazon.com/Core51822-Bluet	3	$7.95	$23.85		$6.36
Heartbeat	**https://www.walmart.com/ip/Heart-Rate-**	3	$5.80	$17.40		$4.64
MCU Arduino Uno	**https://www.amazon.com/Arduino-Uno-F**	1	$25.00	$25.00		$0.00
Accelerometer	https://www.amazon.com/gp/product/B00SI	3	$8.96	$26.88		$7.17
ATMega328P	https://www.arrow.com/en/products/atmega	2	$2.13	$4.26		$1.71
9V Alkaline Battery	https://www.amazon.com	6	$1.40	$8.40		$1.12
Harness	https://www.amazon.com/dp/B071NSHTG7	1	$23.68	$23.68		$18.95
PCB Circuit Enclosure	Radioshack	1	$12.45	$12.45		$9.96
16MHZ	https://www.amazon.com	10	$0.51	$5.10		$0.41
Diodes	https://www.amazon.com	3	$0.54	$1.62		$0.43
GSM Battery	https://www.amazon.com	2	$20.00	$40.00		$16.00
Buck Converter MP1584EN	https://www.amazon.com	6	$1.62	$9.72		$1.30
GPS + GSM module	https://www.adafruit.com/added	1	$75.80	$75.80		$60.64
GPS Circuit Enclosure	https://www.ebay.com/itm/USA-made-BLAC	1	$9.95	$9.95		$7.64
PCB	https://oshpark.com/	3	$11.33	$33.99		$9.06
DevBoard GPS + GSM	Radioshack	1	$2.99	$2.99		$2.39
Testing Subject	Target	1	$29.99	$29.99		$0.00
				0		
				0		
				TOTAL		TOTAL
				$363.07		$149.70

Table 11 Estimated Budget for Project.

8.2 Project Milestones

When working with others on a long term project, a schedule is a very useful and often times necessary task that has great benefits. This allows for each team member to know what his/her tasks are, and when they are expected to be done. A rigorous schedule combined with weekly meetings that allow each member to know where they stand and where the whole project stands are a powerful tool. This allows team members to adjust their own personal schedules to make sure they have enough time and resources to make sure they can accomplish the tasks given to them.

Schedules will greatly raise the chances of a successful and on time project. This is because of the organization that accompanies the timeline. Below this introduction, our team has agreed upon this schedule to keep us on task and organized. These project milestones may shift and change throughout the course of our project. If that does occur, it is important for the group to address and agree to any changes so that no one is uniformed of the new schedule. While this schedule is made up mainly of deliverables and milestones, they are a good reference when proceeding forward with our project.

This timeline encompasses both senior design one, and senior design two. Senior design one which will take place in the summer of 2017, will be mainly focused on the research, design and testing of our Smart Harness. Senior design two which will take place in the fall of 2017 and will be used to test and build our prototype along with demos and presentations. If we follow our tentative schedule we will have a successful senior design project and will be prepared to join the workforce upon our completion our college programs.

Senior Design I	**Tasks**	**Start**	**Status**
	Group Formation	05/28/2017	Completed
	Collaborate Ideas	05/28/2017	Completed
	Divide & Conquer RD	06/02/2017	In Progress
	Finalize Idea with professor	06/06/2017	In Progress
	MILESTONE 1 -- Finalize Idea		
	Review notes from professor's advice	06/07/2017	Pending
	Divide & Conquer Final Documentation	06/09/2017	Pending
	MILESTONE 2 -- Full Requirements & Specifications		
	Research, document and design PCB	06/16/2017	Pending
	Research, document and design Microcontroller	06/16/2017	Pending
	Research, document and design Heart sensors	06/16/2017	Pending
	Research, document and design Accelerometer	06/23/2017	Pending
	Research, document and design Harness	06/23/2017	Pending
	Research, document and design pedometer	06/23/2017	Pending
	Research, document and design compact sensors	06/23/2017	Pending
	Research, document and design dog anatomy	06/30/2017	Pending
	Research, document and design Bluetooth	06/30/2017	Pending
	Research, document and design Phone Application	06/30/2017	Pending
	Senior Design I Documentation RD (60p)	07/07/2017	Pending
	Senior Design I Documentation RD grade review	07/14/2017	Pending
	Senior Design I Documentation RD Adjustments	07/14/2017	Pending
	Senior Design I Documentation RD (100p)	07/21/2017	Pending
	Senior Design I Documentation RD grade review	07/28/2017	Pending
	Senior Design I Documentation RD Adjustments	07/28/2017	Pending
	Senior Design I Final Documentation	08/01/2017	Pending
	MILESTONE 3 -- Research & Design Phase Complete		
Senior Design II			
	Build Prototype	08/21/2017	Pending
	Testing	TBA	Pending
	Redesign	TBA	Pending
	Testing	TBA	Pending
	Redesign	TBA	Pending
	Testing	TBA	Pending
	Finalize	TBA	Pending
	Peer Presentation	TBA	Pending
	Final Report	TBA	Pending
	Final Presentation	TBA	Pending

Table 12 Table of Project Milestones.

8.3 Team Members

Our group 15 consist 3 members including Matthew Horton, Dominic Vu, and Hai Nguyen. Dominic Vu and Hai Nguyen are Computer Engineering. Meanwhile Matthew is Electrical Engineering. Matthew Horton will take care the hardware and Hai Nguyen will assist Matthew in hardware. Dominic Vu will work on the application and Hai Nguyen will also assist Dominic in software.

Matthew Horton

Matthew is a student at UCF studying for his Bachelor's of Science in Electrical Engineering. He was born in the small city of Springfield, Ohio and moved to Florida at the age of 15. His interest in electronics began when he built his first computer around the age of 17 and has always been fascinated with understanding how things work. Matthew is currently employed as an engineering intern producing CAD drawings for engineers at R.L. Plowfield & Associates. Upon graduation, Matthew is eager to join the workforce and eventually return to obtain his Masters in Electrical Engineering.

Hai Nguyen

Hai Nguyen is a student at UCF studying for Bachelor's of Science in Computer Engineering. Also, he has intent to go for Electrical Engineering once he complete his degree in Computer Engineering. He was born at Ho Chi Minh City in Vietnam, and he moved to U.S in 2009. He has been interest in computer at 14 years old. He also wants to obtain the knowledge from school to build his own device. After graduation, Hai want to join U.S army and become an engineer for them. Later, He can pursue his Master Degree in Computer Engineering or Electrical Engineering.

Dominic Vu

Dominic is a Vietnamese American born in Layton, Utah on August 1991. His parents migrated over to the United States of America back in 1982 to escape communism and embark on a journey to a better life. Growing up he was always interested in the medical field (which was also heavily influenced by his parents) but after pursuing a degree in pharmacology, after the third year of undergrad he decided that was not what he wanted to do for the rest of his life. His father is an Electrical Engineer that has been a software developer for the past thirty years for the government. He works as a civilian for the navy and has held a developer position at NavAir on science drive for the past 15 years. Since Dominic was deterring from his initial plans to become a pharmacist, his father had suggested for him to embark on a path to engineering. The electrical engineering course curriculum includes computer programming classes and after taking one or two of those courses, he fell in love. So he switched once again to computer engineering as he found more interest in the programming field and will graduate in the fall with a Bachelor's of Science in Computer Engineering. Currently he is employed locally at a software company that specializes in contract management. He started there two years ago working as technical support and moved his way up to quality assurance until finally landing a position as junior developer. The main goal for him is to gain as much experience and knowledge as possible through work and school to eventually land a stable and fulfilling job. There are no plans in the future for returning to school but through past experience we can see that the future is truly unknown.

9.0 Global Positioning System Module (Additional Component)

The Global Positioning System(GPS) module was a late addition to our project. It was believed that a smart harness would seem incomplete without implementing this component. A GPS module will allow the user to be able to determine the location of their dog while they are wearing the smart harness. This module utilizes satellites that are in orbit to determine a single location on a sphere (Earth).

The GPS module we added to our project is the Adafruit Fona 808 Cellular + GPS Shield for Arduino. At the center of the module is a powerful GSM cellular module with an integrated GPS. This module is called the SIM808. The module is a two in one module that contains both cellular and GPS tracking. This allows the location of the GPS to be transmitted to a mobile device. The module allows the user to connect with any global GSM network with any 2G SIM. The module is a fully-integrated GPS using the MT3336 chipset with -165 dBm tracking sensitivity, that can be controlled and query over the same serial port and the accuracy of the device is 2.5 meters.

Some extra components were needed to make this module operate, it needs a passive GPS antenna uFL to connect to the GSM, as well as a 2G mini SIM card. The Fona 808 also requires a microcontroller that operates in the 3-5V range. The ATMega328P microcontroller that we chose for our design does operate in this range.

For our project we will use the passive GPS. The device will give its location upon it being requested. It will show its exact location using Google Maps. It will not be dynamic like active GPS but will cut down on memory usage. If the subject is in motion, multiple location requests will have to be sent to enable the user to track the GPS device.

This device is newer technology. A big reason this module was chosen was because of its simplicity. Combining GSM and GPS in the same device makes this module much easier to utilize.

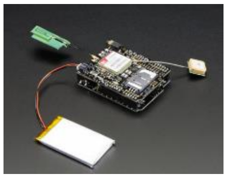


Figure 9.0 Adafruit Fona 808 Cellular + GPS Shield for Arduino

10.0 Final Design Information

The final design was a working Smart Harness paired with a mobile application that was created for the purpose of utilizing the Smart Harness. Below, in Figure 10.1, is a photograph of the final PCB upon completion.

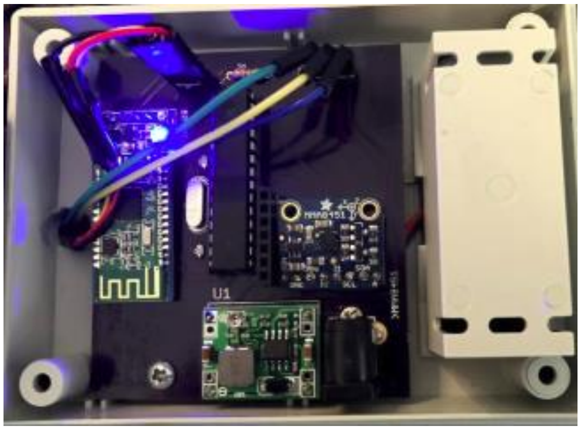


Figure 10.1 Final PCB

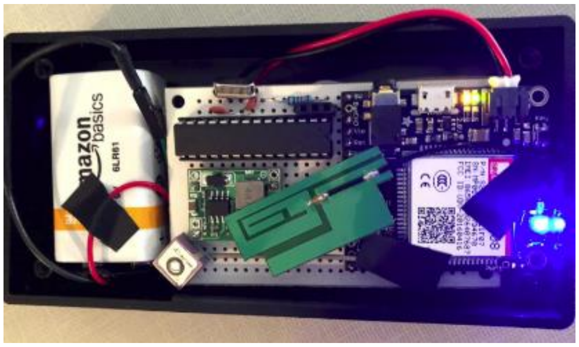


Figure 10.2 Final GPS + Cellular Unit

This PCB included a bluetooth, accelerometer, a buck converter, the heart rate sensor and temperature sensor, as well as a 16MHz crystal and an MCU. The entire PCB was powered by a 9 V battery and was all placed in an ABS plastic circuit enclosure. This enclosure was placed on a tactical harness that was fitted with the sensors as well. When presented at the end of our semester for our final review, all of the functionality worked as intended. This PCB sent the information to a mobile application which then displayed, in real time, readings from the harness. However, many of the proposed software functions were not included in the final design due to time constraints. Features such as alerts, and graphs could be added to the application in the future if the members of the group chose to pursue such an endeavor.

Also included with the PCB was an additional GPS unit. Above, in Figure 10.2, is a photograph of the GPS and Cellular module upon completion. The GPS unit is a passive GPS that will only show the location in an instant, like a pin, on google maps. Unlike navigation, the GPS is not dynamic and will not move actively with the dog, but will take the location at any instant the location is requested. Both of these units, the GPS + Cellular, and the main PCB, were placed on the harness and demonstrated to success on Tues. Nov. 28th 2017, at approximately 6pm.

11.0 Issues

Just like any engineering endeavor, or project ran into some issues during development. Bluetooth connectivity to the Samsung device that was originally intended to be used. This was solved using a different android device, which proved to be a great solution. Time was a limiting factor, as much of it was spent trying to resolve the bluetooth issues. As a result many of the software features were not included in the final design. The GPS implementation ran into issues as well. It was a late addition and as such much of our design was already set in stone. Our MCU did not have enough memory to handle the new part and a 2nd MCU was needed to allow for the functionality of the GPS to be successful.

12.0 Citations

1. "History of Temperature." *CAPGO*. Capgo Prt Ltd, n.d. Web. 20 June 2017.
2. 2017, N. (2017). *Number of Google Play Store apps 2017 | Statistic*. [online] Statista. Available at:https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/ [Accessed 7 Jul. 2017].
3. *2-Wire High-Accuracy Temperature Sensor*[PDF]. (2010). Microchip Technology Inc.
4. A. (2016, September).*ATmega328/P*[PDF]. San Jose: Atmel Corporation.
5. A. (2015, December 07). Adafruit/Adafruit-Eagle-Library. Retrieved August 01, 2017, from https://github.com/adafruit/Adafruit-Eagle-Library
6. Adafruit MMA8451 Accelerometer Breakout. (2017). 1st ed. [ebook] Adafruit. Available at: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-mma8451-accelerometer-breakout.pdf [Accessed 1 Aug. 2017].
7. *APDS-9008 Miniature Surface-Mount Ambient Light Photo Sensor*[PDF]. (2008, November 7). Avago Technologies.
8. Alpharubicon.com. (2017). *Batteries*. [online] Available at: http://www.alpharubicon.com/altenergy/batteryfred.htm [Accessed 20 Jul. 2017].
9. Android Authority. (2017). *Google Play Store vs the Apple App Store: by the numbers (2015)*. [online] Available at: http://www.androidauthority.com/google-play-store-vs-the-apple-app-store-601836/ [Accessed 7 Jul. 2017].
10. Android Authority. (2017). *I want to develop Android Apps - What languages should I learn?*. [online] Available at: http://www.androidauthority.com/want-develop-android-apps-languages-learn-391008/ [Accessed 7 Jul. 2017].
11. AppsFlyer. (2017). *New Report on Global In-App Spending Habits Finds That Asian Consumers Spend 40% More In Apps Than the Rest of the World - AppsFlyer*. [online] Available at: https://www.appsflyer.com/pr/new-report-global-app-spending-habits-finds-asian-consumers-spend-40-apps-rest-world/ [Accessed 7 Jul. 2017].

12. Agarwal, T. (2017). *Heart Beat Sensor - How to Measure Heart Beat: Working and Application*. [online] ElProCus - Electronic Projects for Engineering Students. Available at: https://www.elprocus.com/heartbeat-sensor-working-application/ [Accessed 6 Jul. 2017].
13. Arduino.cc. (2017). *Arduino Uno Rev3*. [online] Available at: https://www.arduino.cc/en/main/arduinoBoardUno [Accessed 7 Jul. 2017].
14. B. (2016, March 23). Energy Density of 9V battery vs AA batteries. Retrieved July 20, 2017, from https://www.baldengineer.com/9v-battery-energy-density.html
15. *BGM113 Blue Gecko Bluetooth ® Module Data Sheet*[PDF]. (n.d.). Austin: Silicon Labs.
16. Billman, G. (2017). *Heart Rate Variability ? A Historical Perspective*.
17. Boards), G. (2017). *GY-27 3-Axis Compass Accelerometer Module for Arduino (Works with Official Arduino Boards)*. [online] DX.com. Available at: http://www.dx.com/p/gy-27-3-axis-compass-accelerometer-module-for-arduino-147763#.WVqOKojyu71 [Accessed 7 Jul. 2017].
18. Bolton, D., Handy, A. and Handy, A. (2017). *A Scoring Comparison of Android and iOS Development - The New Stack*. [online] The New Stack. Available at: https://thenewstack.io/scoring-comparison-android-ios-development/ [Accessed 7 Jul. 2017].
19. *DS18B20*[PDF]. (2015). Maxim Integrated Products, Inc.
20. dzone.com. (2017). *Why Android Studio Is Better For Android Developers Instead Of Eclipse - DZone Mobile*. [online] Available at: https://dzone.com/articles/why-android-studio-better [Accessed 7 Jul. 2017].
21. En.wikipedia.org. (2017). *Android (operating system)*. [online] Available at: https://en.wikipedia.org/wiki/Android_(operating_system) [Accessed 7 Jul. 2017].
22. En.wikipedia.org. (2017). *Android Studio*. [online] Available at: https://en.wikipedia.org/wiki/Android_Studio [Accessed 7 Jul. 2017].
23. En.wikipedia.org. (2017). *Eclipse (software)*. [online] Available at: https://en.wikipedia.org/wiki/Eclipse_(software) [Accessed 7 Jul. 2017].
24. En.wikipedia.org. (2017). *IntelliJ IDEA*. [online] Available at: https://en.wikipedia.org/wiki/IntelliJ_IDEA [Accessed 7 Jul. 2017].
25. En.wikipedia.org. (2017). *Java (programming language)*. [online] Available at: https://en.wikipedia.org/wiki/Java_(programming_language) [Accessed 7 Jul. 2017].
26. En.wikipedia.org. (2017). *Objective-C*. [online] Available at: https://en.wikipedia.org/wiki/Objective-C#Interface [Accessed 7 Jul. 2017].
27. En.wikipedia.org. (2017). *Bluetooth*. [online] Available at: https://en.wikipedia.org/wiki/Bluetooth [Accessed 6 Jul. 2017].
28. En.wikipedia.org. (2017). *Heart rate monitor*. [online] Available at: https://en.wikipedia.org/wiki/Heart_rate_monitor [Accessed 6 Jul. 2017].
29. En.wikipedia.org. (2017). *AA battery*. [online] Available at: https://en.wikipedia.org/wiki/AA_battery [Accessed 20 Jul. 2017].
30. En.wikipedia.org. (2017). *AAA battery*. [online] Available at: https://en.wikipedia.org/wiki/AAA_battery [Accessed 20 Jul. 2017].

31. En.wikipedia.org. (2017). *C battery*. [online] Available at: https://en.wikipedia.org/wiki/C_battery [Accessed 20 Jul. 2017].
32. En.wikipedia.org. (2017). *D battery*. [online] Available at: https://en.wikipedia.org/wiki/D_battery [Accessed 20 Jul. 2017].
33. En.wikipedia.org. (2017). *Nine-volt battery*. [online] Available at: https://en.wikipedia.org/wiki/Nine-volt_battery [Accessed 20 Jul. 2017].
34. Qrg.northwestern.edu. (2017). *How do batteries work?*. [online] Available at: http://www.qrg.northwestern.edu/projects/vss/docs/power/2-how-do-batteries-work.html [Accessed 20 Jul. 2017].
35. Energizer.com. (2017). *Battery History | Dry and Wet Cell Battery History | Energizer*. [online] Available at: http://www.energizer.com/about-batteries/battery-history [Accessed 20 Jul. 2017].
36. F. (2013, October). *Xtrinsic MMA8451Q 3-Axis, 14-bit/8-bit Digital Accelerometer*[PDF]. Freescale Semiconductor, Inc.
37. Facts, A. (2017). *Topic: App stores*. [online] www.statista.com. Available at: https://www.statista.com/topics/1729/app-stores/ [Accessed 7 Jul. 2017].
38. Fortune.com. (2017). *Apple's users spend 4X as much as Google's*. [online] Available at: http://fortune.com/2014/06/27/apples-users-spend-4x-as-much-as-googles/ [Accessed 7 Jul. 2017].
39. *High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health*[PDF]. (2015, September). Maxim Integrated Products, Inc.
40. Home, H. and First?, A. (2017). *Android vs. iOS: Which Should I Learn First?*. [online] Hiring | Upwork. Available at: https://www.upwork.com/hiring/mobile/android-vs-ios-which-to-learn-first/ [Accessed 7 Jul. 2017].
41. Industries, A. (2017). *Adafruit Trinket - Mini Microcontroller - 5V Logic ID: 1501 - $6.95 : Adafruit Industries, Unique & fun DIY electronics and kits*. [online] Adafruit.com. Available at: https://www.adafruit.com/product/1501 [Accessed 7 Jul. 2017].
42. Ieeexplore.ieee.org. (2017). *IEEE Xplore Search Results*. [online] Available at: http://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=802.15.1 [Accessed 20 Jul. 2017].
43. INUPATHY - the world’s first dogs mental visualizer. (2017). *INUPATHY - the world’s first dogs mental visualizer*. [online] Available at: http://inupathy.com/ [Accessed 14 Jul. 2017].
44. Javarevisited.blogspot.com. (2017). *10 Reasons to Learn Java Programming Language and Why Java is Best*. [online] Available at: http://javarevisited.blogspot.com/2013/04/10-reasons-to-learn-java-programming.html [Accessed 7 Jul. 2017].
45. JetBrains. (2017). *Features - IntelliJ IDEA | JetBrains*. [online] Available at: https://www.jetbrains.com/idea/features/ [Accessed 7 Jul. 2017].
46. JetBrains. (2017). *JetBrains AppCode: Swift Execution of Your Bright Ideas*. [online] Available at: https://www.jetbrains.com/objc/ [Accessed 7 Jul. 2017].

47. Leswing, K. (2017). *Apple is pulling further ahead of Google in this one key area*. [online] Business Insider. Available at: http://www.businessinsider.com/apple-app-store-revenue-per-user-versus-android-2016-7 [Accessed 7 Jul. 2017].
48. LINK AKC Smart Collar. (n.d.). Retrieved July 14, 2017, from https://shop.linkakc.com/products/link-akc-smart-dog-collar?gclid=CjwKEAjw16HLBRDF9L2UmOCH7U8SJAASVESoTk_L1ttTlwckr8S1ksDiLvjIS7KUlke9jeq2OoQG8xoC82Hw_wcB
49. M. (n.d.). ModuleForces. Retrieved August 01, 2017, from http://moduleforces.blogspot.com/2017/02/bluetooth-40-ble-module-sh-hc-08.html
50. Modern Device. (2017). *Pulse/Heartbeat Sensor | Modern Device*. [online] Available at: https://moderndevice.com/product/pulse-heartbeat-sensor/ [Accessed 6 Jul. 2017].
51. *MSP-EXP430G2 LaunchPad™ Development Kit*[PDF]. (2016, March). Dallas: Texas Instruments Incorporated .
52. *NRF51822 Multiprotocol Bluetooth® low energy/2.4 GHz RF System on Chip*[PDF]. (2014). Nordic Semiconductor ASA.
53. *LSM303DLHC*[PDF]. (2011). STMicroelectronics.
54. *Low-Power Linear Active Thermistor™ ICs*[PDF]. (2005). Microchip Technology Inc.
55. Picaxe.com. (2017). *PICAXE-08M2 microcontroller (AXE007M2) - Hardware - PICAXE*. [online] Available at: http://www.picaxe.com/Hardware/PICAXE-Chips/PICAXE-08M2-microcontroller/ [Accessed 7 Jul. 2017].
56. *Pulse Sensor Getting Started Guide*[PDF]. (n.d.). Pulsesensor.com.
57. Robotshop.com. (2017). *ADXL345-BB 3-Axis Accelerometer Module*. [online] Available at: http://www.robotshop.com/en/adxl345-bb-3-axis-accelerometer-module.html?gclid=Cj0KCQjws-LKBRDCARIsAAOTNd7J-k7BmlKrFW5OacFyj3470K84nvk9hFerps1RczAcxzRPT5MBflMaAkSpEALw_wcB [Accessed 7 Jul. 2017].
58. SMART-SENSING COLLAR. (n.d.). Retrieved July 14, 2017, from http://petpace.com/smart-sensing-collar/
59. steve cheney – technology, business & strategy. (2017). *Why Android First is a Myth*. [online] Available at: http://stevecheney.com/why-android-first-is-a-myth/ [Accessed 7 Jul. 2017].
60. *Si1143-AAGX Data Short*[PDF]. (n.d.). Austin: Silicon Labs.
61. Studio, M. (2017). *Meet Android Studio | Android Studio*. [online] Developer.android.com. Available at: https://developer.android.com/studio/intro/index.html [Accessed 7 Jul. 2017].
62. T. (2013, May). *MIXED SIGNAL MICROCONTROLLER*[PDF]. Dallas: Texas Instruments Incorporated .
63. Tutorial sensor digital de temperatura DS18B20. (n.d.). Retrieved August 01, 2017, from http://www.naylampmechatronics.com/blog/46_Tutorial-sensor-de-temperatura-DS18B20.html

64. Townsend, K. (2016, September 15). *Getting Started with the nRF8001 Bluefruit LE Breakout*[PDF]. Adafruit Industries .
65. Wahby, M. (n.d.). PCB design basics. Retrieved August 01, 2017, from http://www.edn.com/design/pc-board/4424239/PCB-design-basics
66. *What is a Temperature Sensor?* [PDF]. (2001, January 1). Oak Park: Trerice.
67. World Famous Electronics llc. (2017). *Open Hardware*. [online] Available at: https://pulsesensor.com/pages/open-hardware [Accessed 6 Jul. 2017]
68. *Adafruit Fona 808 Cellular + GPS Shield for Arduino* [PDF]. (Updated 2017, January 1) Lady Fona. Available at: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-fona-808-cellular-plus-gps-shield-for-arduino.pdf