Monitoring Device and Software for Equine Colic Symptoms Detection

500-TT-1-HorseBit

Design Roadmap Document

[redacted]

Academic Advisor: Kartikeya Murari, kmurari@ucalgary.ca

Sponsoring Company: HorseBit

Sponsor Representative: Dylan Rae, dylan.rae@ucalgary.ca

Table of Contents

List of Figures

List of Tables

Glossary

Accelerometer: Electromechanical device used to measure acceleration forces.

Analog Signal: Continuous signal for which the time-varying feature of the signal is a representation of some other time varying quantity.

Cloud Controller: Storage appliance that automatically moves data from on-premises storage to cloud storage.

Colic: Severe abdominal discomfort characterized by pawing, rolling, and sometimes the inability to defecate.

Electrode: Device that detects the electrical activity of the heart.

GANTT Chart: Chart which shows certain work planned for certain periods of time in relation to the time allotted for the entire project.

Gyroscope: Device used for measuring or maintaining orientation and angular velocity.

Heart Rate: Speed of the heartbeat measured by the number of contractions (beats) of the heart per minute (bpm).

Hspa+: Mobile telephony technology that allows for data transmission speeds up to 42 Mbps.

IDE: Integrated Development Environment which is an application that gives the programmer the tools they need to program.

Impedance Pneumography: Commonly-used technique to monitor a person's respiration rate, or breathing rate.

IMU (Inertial Measurement Unit): Electronic device that measures and reports a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, sometimes also magnetometers.

Lame (Equine): Mild to severe loss of ability to move normally that can be caused by problems in a horse's bones, muscles, nerves, tendons, ligaments, brain, circulation, and metabolism.

Microcontroller: Compact integrated circuit designed to govern a specific operation in an embedded system.

Respiratory Rate: Rate at which breathing occurs.

Server: Computer or computer program that manages access to a centralized resource or service in a network.

Thorax: Part of the body of a mammal between the neck and the abdomen.

Transceiver: Combination of transmitter/receiver in a single package.

Transmitter: Set of equipment used to generate and transmit electromagnetic waves carrying messages or signals.

UI: User Interface referring to space where humans interact with computers.

Vital Signs: Group of the most important signs that indicate the status of the body's vital functions.

Project Motivation and Objectives

Motivation

There exists a large demand for performance horses, but unfortunately most are lame by the end of their careers. Our sponsor has conducted multiple customer interviews and discovered that these valuable horses are at an increased risk of becoming lame due to a general lack of knowledge about horse health, or knowledge of the health of an individual horse. Key symptoms are not able to be identified due to the unfamiliarity with the horse or not being able to monitor the horse at certain times throughout the day.

Although many diseases exist, for the sake of simplicity the project is focused on equine colic, a relatively common disorder of the digestive system in horses [1]. Colic has been described to us by veterinarian Camille Adams, of the Westlock Veterinary Center, as the primary disease inflicting horses in Alberta. Colic is potentially fatal, and the severity increases proportional to the time left untreated. Once the symptoms are severe enough, the horses have to go in for a costly surgery where more than one in ten do not survive [2]. Of those that survive the procedure, another one in ten are not able to return to their previous performance levels.

Objectives

The goal of the project is to develop a device that is able to track important vitals in a horse and act as a first alert system for the potential presence of equine colic. The device will monitor the horse for an extended period of time and indicate to the user, through the use of an app and website, whether the horse is healthy or showing signs of disease. With the implementation of this device a customer is able to get data that they personally may not have the expertise to obtain due to a lack knowledge. It also allows detection at a point where colic is not threatening and can be fought without surgery.

Project Scope and Deliverables

The purpose of this section is to outline the project scope and deliverables. This will be done at a high level, and will not include details on how the implementation will take place. For example, there will be discussion about what vitals are measured but the method of measurement will not be discussed.

The vision of HorseBit as a company is to develop a microchip that can be implemented within the skin of a horse to measure essential vitals which can then be used to detect the presence of equine colic and other disease. The scope has been simplified for the purpose of our project to the creation of an external device that is able to measure specific vitals, use those vitals to determine the health of a horse, and transmit selective data to display to a website and app created for that purpose.

Specifically, our goal is to create a proof of concept by the end of the course that is able to detect various vitals from a horse. Sensors will be used to receive these vitals, which will be attached to a microcontroller unit. This microcontroller unit will process these signals and use the data to decide if the horse is healthy, at risk, or in danger which will be denoted by a green-yellow-red system. The processed data and outcome will be sent wirelessly to a data server. An app and website will be created to display this data in a clear and concise manner to users.

The following is a simplified list of deliverables for our project:

- Proof of Concept Device and Hardware Implementation
  - Sensor Selection and Design
  - Hardware Implementation
  - Signal and Data Processing
- Data Transmission and Storage
- User Software
- User Manual
- Detailed Design Package

Detailed information for each of these can be found below under their respective headings.

Proof of Concept Device and Hardware Implementation

An enclosure will be created that will house a microcontroller, all the sensors and the communication device used in the project. The sponsors have specified that the visual of the device is not a priority of the project. Therefore, focus will be placed on the functionality of the device and the enclosure will be made to fulfill two primary purposes; to protect the electrical equipment and the ability to strap on to a horse in a way that allows for the sensors to collect their necessary information.

The hardware implementation portion of the project is being able to combine all the different hardware components that will be discussed below to make a cohesive device. This will involve a single microcontroller wired to the four different sensors, data transmission hardware and the batteries to power the device.

Sensor Selection and Design

The most crucial portion of the device will be the sensors. Camille Adams noted that for equine colic the two most indicative vitals are respiratory rate and heart rate. She identified these are the two important vitals due to the fact that a notable rise in both of them indicates the presence of only a few diseases, of which the most common is equine colic. It will be necessary that our device incorporates devices that are able to provide accurate measurements for both of these vitals.

An accelerometer and a sensor to detect temperature may also be implemented in the device depending on viability and work priority as the project progresses. The purpose of an accelerometer is to help us better filter the data received from respiratory and heart rates. Depending on how fast a horse is moving, it's respiratory and heart rates will naturally increase, and will incorrectly trigger our device to identify the horse as unhealthy. With the addition of an accelerometer we will be able to create different conditions based on a horse's movement.

Temperature is another vital that is indicative of equine colic, but is a vital that Camille identified as potentially problematic. In the early phases of equine colic, it will cause a rise in internal temperature but if enough time passes the effects will reverse and the horse will have an internal temperature that is lower than usual. Additionally, increased and reduced internal temperature is a symptom of many more equine diseases than colic. Despite this, temperature is a vital we are hoping to include in our measurements as it will provide extra and useful information to the user.

Signal and Data Processing

Signal and Data processing is another important part of the device. The data processing will be done through the use of a microcontroller that will receive inputs from the various sensors listed above. The microcontroller will be coded to convert the inputs of the sensor into usable and readable data, and then use that data to determine how healthy the horse is on a green-yellow-red system. This data will then be sent through means of wireless communication to an external server for use with our user software.

The deliverables from this portion of the project is the post-processed data and the code for the processing.

Data Transmission and Storage

Data transmission is identified by the sponsors as a welcomed, but not necessary, addition to the project. Wireless data transmission will allow real-time data to be displayed on the user software. If data transmission is implemented there will also be a need to implement a method of storing the data that is being transmitted. While an app or a website can host a certain amount of data, it will not be able to do so for hours of data, especially if the data is taken at intervals of a few seconds. A data server will be used as the target of our data transmission. If wireless data transmission is not implemented, then an SD card attached directly to the microcontroller will be used to collect and store data instead.

User Software

User software is identified as a major priority for the sponsors. The user software will include either a website, an app or both. The purpose of the user software is to provide information about the product, health vital information about the horse the device is attached to, and indicates the relative health of the horse. Vitals will be displayed in graphs over a period of time, and the health of the horse will be displayed as either green (healthy), yellow (at risk) and red (in danger) for easy visualization for the user. A priority for the website and app is a clean UI.

User Manual

A user manual will need to be provided to the sponsors detailing the use and function of the proof of concept device, as well as the user software. As the project progresses, the details required for this document will be discussed with the sponsor.

Detailed Design Package

A detailed design package will be provided for every single portion of the project to the sponsor. This is an important deliverable as it details all the work that has gone in to the project, and gives the sponsor the information they need to progress their product past the timeline of the capstone. This package will include all technical documents required by the ENEL 500 course, as well as the code used for the user software and data processing within the microcontroller.

Preliminary Design of the Proposed Solution and Alternative Solutions/Methodologies

This section will go into detail about our final preliminary design based on our extensive research. This section will also include the alternative solutions and methodologies we had for each of our project components. The following block diagram is a high level overview of our objectives to display the horse's vitals in real time on a website and Android app.

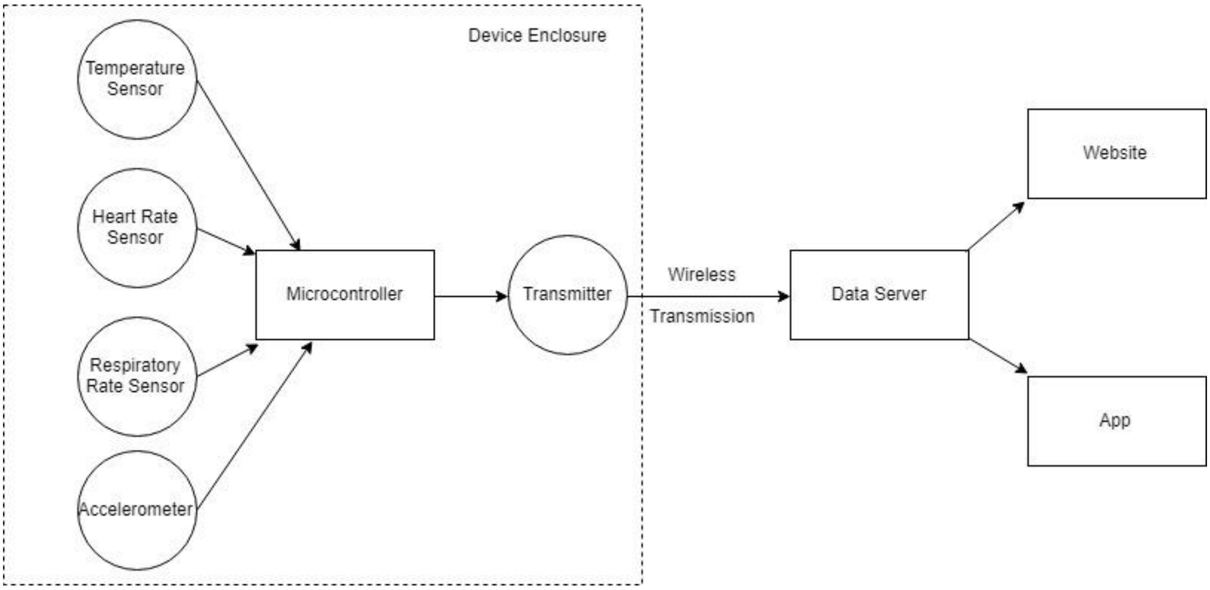

Figure 1: Block Diagram of Proposed Solution

This project will be divided into the following parts

1. The sensors placed on the horse
2. The enclosure/harness to hold the sensors
3. The signal processing at the output of the sensors
4. The wireless transmission of the output of the processed signals
5. The graphical display using the transmitted data on an App or Website

These parts are explained in more detail below.

Sensors

Multiple sensors will be used to measure a horse's vitals; the data acquired by these sensors will be used to determine the presence of colic in the horse. As discussed with a veterinarian (Camille Adams) the vitals that are most indicative of Colic in the horse are abnormal respiratory and heart rate. Other vitals can be false positives, but temperature and movement are interesting vitals to measure as they will allow us to identify many other problems in a horse.

When choosing sensors the most important aspects to be considered for our purposes are size and data reliability. The following section will discuss a more detailed selection process for the sensors required to detect the following vitals:

- Heart Rate
- Respiratory Rate
- Temperature
- Movement

Respiratory Rate and Heart Rate

The sensor chosen for detecting heart rate and respiratory rate is called the ADS1292R Analog Front End IC. This sensor uses impedance pneumography to capture the respiration rate; this is done by placing two electrodes on the horse to observe changes in the electrical impedance of the horse's thorax caused by respiration. These electrodes also measure the heart rate by capturing a series of coordinated electrical signals that are sent to the heart by the autonomic nervous system.[3]

A third electrode will also be used for common mode noise reduction to improve the reliability of the data. This data can further be optimized by adjusting the data rate and PGA setting. As the averaging is increased by reducing the data rate, the noise drops correspondingly. By increasing the programmable gain amplifier (PGA) value reduces the input-referred noise, enhancing signal accuracy. The noise reduction aspects of this device as mentioned above are the reasons that it was chosen over the device mentioned below; accurate information is crucial in our application.

The device is 5 mm x 5 mm, the size is relatively small so it would not pose any issues fitting in our enclosure. The operating temperature is –40°C to +85°C, meaning it would not pose any issue in Alberta's tough climate. This sensor also provides compatibility with the Arduino Uno; our microcontroller of choice.

An alternative sensor found for detecting heart rate is called "Pulse Sensor". This sensor needed to be able to gather reliable data, it was chosen because of its amplification and noise cancellation circuitry which would allow for somewhat reliable pulse readings. This sensor is 0.625" in diameter and 0.125" thick, so it would not pose any issues fitting in our enclosure. This sensor also provides compatibility with the Arduino Uno; our microcontroller of choice. A schematic of the circuitry of the sensor is below: the APDS-9008 detects concentration of light bouncing back through the body, the D1 LED gives light source to do this and the op-amp amplifies the signal so it can be read by a microcontroller.

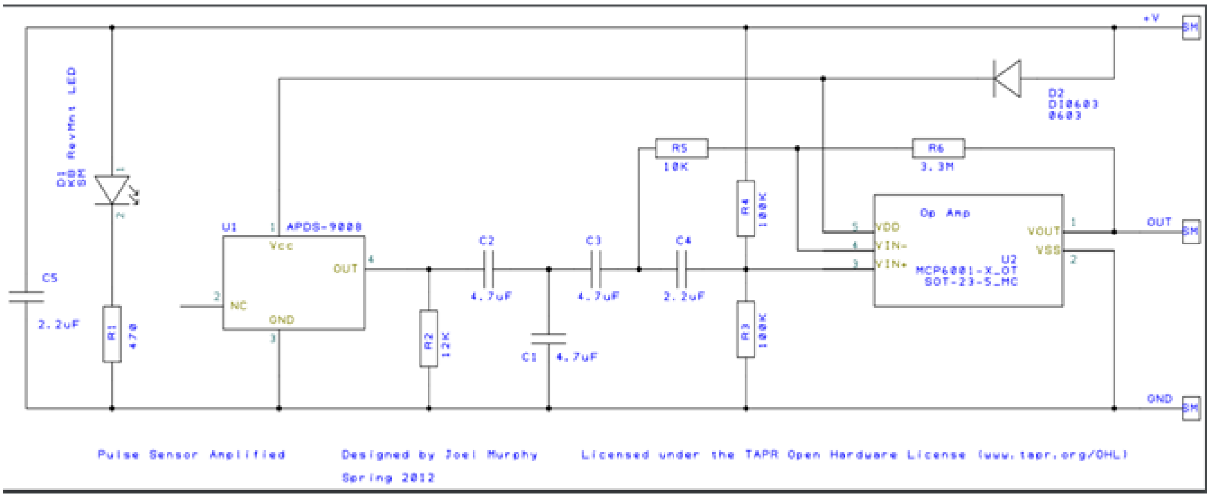

Figure 2: "Pulse Sensor" Schematic [4]

Temperature

The sensor chosen to gather body temperature of the horse is the DS18B20. This sensor is very precise (±0.5°C) which is because it doesn't rely on the accuracy of the microcontroller to measure the analog signal and this sensor has a digital output, meaning the signal will not degrade. This sensor has a 1-wire interface, meaning it will only take up one pin on the microcontroller. The temperature range that this sensor will sense is -55 to 125°C; this is acceptable for our application.

An alternative sensor that was looked at was the LM35DZ, this sensor was not chosen because it provides lower precision (±1°C) and would take up 3 pins on the microcontroller, and microcontroller real estate is very valuable for other devices to be attached to.

Motion

To determine the speed and positioning of the horse an IMU (Inertial Measurement Unit) has been chosen as the main method. An IMU has been chosen over separate accelerometers and gyroscopes is because of the IMU's multi functionality to be able to detect both movement and positioning of the horse. The IMU chosen is the MinIMU-9 v5, this IMU consists of an LSM6DS33 3-axis gyro and 3-axis accelerometer and a LIS3MDL 3-axis magnetometer [5]. These sensors are compact in size at 0.8″ × 0.5″, therefor they won't pose any issues fitting in our enclosure. This sensor also provides compatibility with the Arduino Uno; our microcontroller of choice.

Enclosure

To design the enclosure we first have to define all of our parts that are going to be on our horse, and the size and dimensions of those parts.

- Particle Electron 3G Transceiver: Dimensions are 5.08 x 2.032 x 1.27 cm [7]
- Particle Electron Battery: Dimensions are 1.08cm x 3.40cm x 4.98cm [14]
- Arduino Uno with all relevant sensors: Dimensions are 7.50 cm × 5.50 cm x 2 cm

Enclosure to encase modules

To encase the modules, we will use a 3D printed enclosure and glue all the modules we need firmly into place. We will leave a space open to be able to let 3 sensor wires through the pouch to the horse belt. The dimensions of the enclosure will be 10cm x 9cm x 4cm and will have a hole in the back to allow the 3 sensor wires to pass through.

Strap Around Horse's Leg

We will be measuring the horse's vitals near the hoof of the horse's leg. The general idea of the enclosure will be a hollow leather belt (to allow wires to run through) with a pouch attached containing all of our modules with a slight opening to allow for sensor wires to run through the belt. The belt will have a velcro strap in order to provide the perfect fit for the horse so that it is not uncomfortable. The pouch will also have a velcro strap to fit the 3D enclosure more tightly. It will look very much like a fanny pack. We can have this enclosure made at a leather maker such as R-N Leather Works.

Figure 3: Visualization for Horse Enclosure [15]

Microcontroller

The diagram below shows the overview of our microcontroller design. Four biomedical sensors are used to measure the health of a horse, and they are connected to the microcontroller by pairs of wires. These sensors measure different biomedical information of a horse as analog signals, and then transmit these signals to the analog pins located on the microcontroller board. The microcontroller receives the signals and gives a simple data processing on-site to ensure the data is reliable and accurate. The microcontroller uploads the data to a cloud server for further processing by using an external wireless communication module.

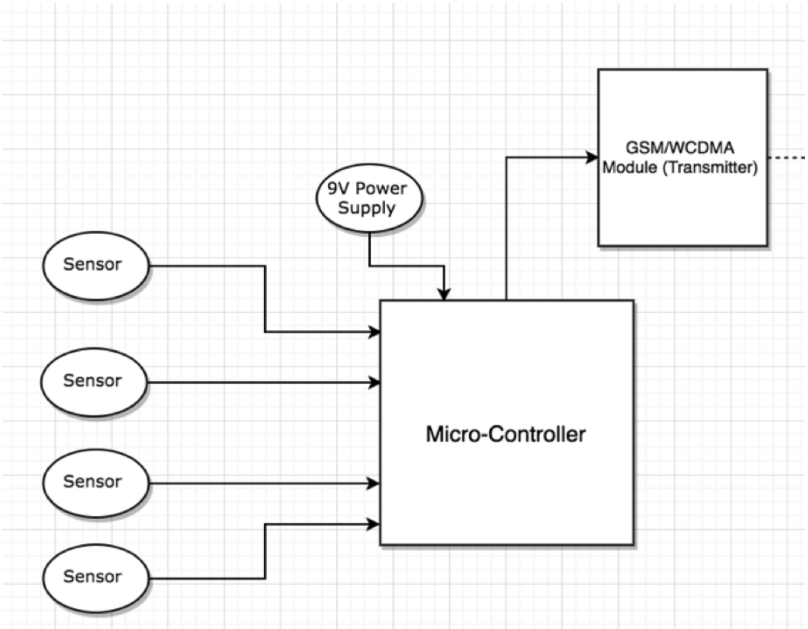

Figure 4: Microcontroller Overview

There are many options for microcontroller available for this project. Based on our research, *Arduino Uno Rev 3* becomes our primary option, while *Raspberry Pi 3* is an alternative solution in case the project requires a more powerful processor and multi-tasking.

Arduino Uno vs. Raspberry Pi 3

Arduino Uno Rev 3 is a microcontroller while *Raspberry Pi 3* is described as a Single Board Computer [18]. Compared to Arduino, *Raspberry Pi* is more powerful with larger memory and a stronger processor. Moreover, *Raspberry Pi* can have an operating system installed on the board which supports multitasking and intense computations like a computer does, such as audio and graphics. *Arduino*, on the other, is much lighter than a *Single Board Computer*. It has most of the functionalities that *Raspberry Pi* has besides multitasking. *Arduino* can have only one program running at a time, meaning that multithreading will not be supported.

These two boards by themselves do not have enough functionalities for our project, but they both support the addition of external modules and there are various modules available that allow for different functionalities. For example, a communication module can be added onto a board in order for the board to communicate through a WCDMA/GSM network.

Figure 5: Raspberry 3 (left) and Arduino Uno (right)

Primary Microcontroller

According to the scope of our project, multithreading is not necessary to have in a microcontroller. The main task of the microcontroller in our project is to read multiple analog signals from the sensors and transmit the data to a server through a WCDMA/GSM wireless network for further processing.

Arduino Uno is selected to be the primary design mainly due to its simplicity of setting up and the ready-for-use analog pins. It is perfect for a project with a small amount of analog inputs. We currently plan to use four (4) sensors to measure different biomedical features of a horse. *Arduino Uno* has a maximum of six (6) analog pins on the board which means it’s able to hold the sensors. Additional analog pins are also available as the project evolves and requires more than six (6) analog inputs.

Alternative Micro-Controller

The major concern for our primary microcontroller is the power of its processor and memory. The processor has to be able to handle a large amount of incoming analog signals as well as communicate to the cloud server via WCDMA/GSM. Referring to the datasheet (Appendix), Arduino’s flash memory sometimes becomes very limited when the libraries to drive the communication module take up a large portion of the flash memory. If the Arduino processor slows down dramatically with all the tasks running in one program, we will abandon the primary microcontroller and use Raspberry Pi 3 instead. Reading sensor inputs and transmitting signals will be then separated into two different threads so that we can speed up the whole process.

Data Processing

The diagram below shows an overview of how the sensor data is processed. The sensor outputs received by the cloud server, historical data and baseline values for different biomedical features are stored in memory, and an algorithm uses all this information to determine the health of a horse.

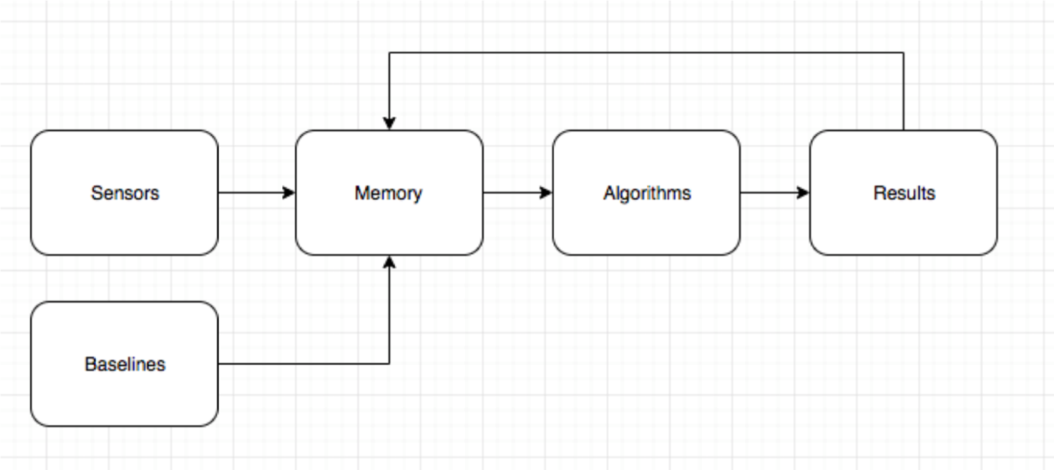

Figure 6: Overview of Sensor Data Processing

Sensor Outputs

Due to the scope of this project which is to monitor the health of a horse, instant data is not necessary and a little bit data delay in seconds can be acceptable as continuous sampling costs huge amount of power. Therefore, in order to save power for a longer battery life, we have decided to sample all the sensors at a period of five (5) seconds.

In other words, the microcontroller controls all the sensors and samples them every five (5) seconds. After receiving data from the sensors, the microcontroller will upload the data to the cloud server in a very short delay via GSM/WCDMA.

Baselines

Baselines are determined by consulting a professional veterinarian and other sources. These values are stored in the cloud server as the reference of a healthy horse, they are used by an algorithm to determine the result as constant parameters.

Memory

The microcontroller needs to have enough flash memory to store the data from the sensors and the code, which includes libraries. With a small amount of incoming data at a time, Arduino should be able to process the data although it has limited flash memory according to the datasheet (Appendix).

Hard drive memory in the cloud server, however, can vary as more and more historical data needs to be stored. For a less expensive cost, we have initially chosen one (1) GB hard drive memory for our cloud server and we may increase this in the future.

Algorithms

Primary Algorithms – Mean Value Comparison

The basic algorithm for processing the data is to compare the mean values within a 5-minute time frame with the baseline values and historical data. The reason why we choose such a time period is that a longer time window gives more accurate results, while a small portion of data from the sensors can vary due to interference and communication failure, etc. By doing this, we can eliminate the impact from faulty data points which can cause a wrong signal.

If the mean values see a variation away from an expected level, the controller will then generate ABNORMAL signal to draw users' attention to the horse. For the mean value of a dangerous setting level, DANGEROUS signal will be generated.

Alternative Solutions – Machine Learning Algorithm

Machine Learning algorithms seem to be an optimal solution over our chosen method. If our primary solution produces results with fairly poor accuracy, we will switch to use a machine learning algorithm to compute the results. However, the cost of building and training such an algorithm is tremendously high. Using this solution requires a huge amount of time and efforts. Machine Learning solution requires numerous data for training at the beginning, and afterwards maintenance of the model is also an issue as more and more data feed in the algorithm may require change of the structure and parameters.

Results

Results are generated by the algorithms described above. Results are divided into three (3) categories: HEALTHY, ABNORMAL and DANGEROUS. HEALTHY means that the measured data is within the range; ABNORMAL means that the data is just a little bit out of the range; DANGEROUS means that the data has large deviations from the range. All the results are stored in the cloud server with a timestamp as historical data. The server can use the recent historical data along with the baselines to compute the next result.

Communication/Transmission Method

For this project, we will need to transmit the data collected from the sensor strapped onto the horse to the internet where the app will extract the data from the horse to display real time

information about the horse's vitals. Our ideal situation is to transmit data from the horse and get it to a cloud server database where it can be used by our website to display the data.
We went through 3 different wireless techniques that we could use in order to transmit data to the internet while minimizing cost and maximizing data rates and efficiency. Here are the 3 techniques and their pros and cons.

1. **3G Wireless:** This is a common method of communication for cell phones. This method would mean buying a 3G transceiver, mounting it on the horse and using the sensor data to transmit data directly to the internet [12].

<u>Pros</u>:

- Direct connection to the internet
- Great coverage due to cell phone service providers providing 3G service
- High data rate
- Small and durable

<u>Cons</u>**:**

- High battery usage
- There is a monthly bill and data limit.

2. **LoRa (Long Range):** This service was designed to support the early Internet of Things (IoT) network and can support extremely long range distances of upto 15 - 20 km. To use this service, we would have to buy a LoRa transceiver as well as buy the Lora Gateway if one does not exist in our area. Then we would have to connect the transceiver to the horse, send the data to the gateway which then send the information to a cloud based network controller. Here are the strength and weaknesses of LoRa [11].

<u>Pros</u>:

- Extremely low power usage (battery can last up to 10 years)
- Extremely long range of connectivity
- Costs for using network is free

<u>Cons</u>:

- Little infrastructure built (network in its early phases) which causes low coverage even with the big distance range
- High costs of building our own gateway
- Low online support due to less usage and early stage of development.

3. **WiFi**: This is the common internet service we use at home. We can buy a WiFi transceiver and put it on the horse, which would be connected to the internet through a local router from where we can use the data [13].

<u>Pros</u>:

- Very common
- Can find in every single household and in public places
- High data rate

- Small and durable transceiver
- No monthly payment, low cost of device

Cons:

- Very low range, 10 - 15 meters.

After careful consideration of all 3 wireless mechanisms, we have come to an agreement to use 3G to transmit our data wirelessly. The reason is that we had to have something long range so that connection isn't lost when the horse is more than 10 - 15 meters away. That left 3G and LoRa as our only options, however the LoRa network infrastructure has only 3 gateways in Calgary, and it would require a lot of cost in order to build our own LoRa gateway. Although 3G has monthly cost, it is only $3 per month and has a much higher data rate than the LoRa network. Although it has high power consumption, for the purpose of this project, we don't need something to stay on the horse for years.

Product Selection and Reasoning

The product that we intend to use is the Particle Electron for the 3G wireless transfer of our data. It is 3G transceiver with its own sim card that connects directly to the internet just like a cell phone. This device is about the size of a thumb, and so it's small, light, durable and can easily fit within our enclosure. It comes with a battery and a very thin antenna as well. The cost is $69 for this module [7].

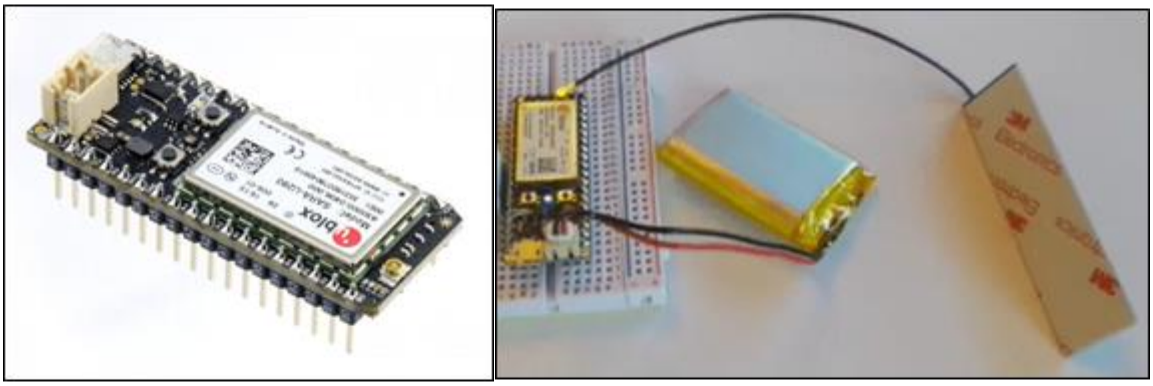

Figure 7: Particle Electron

It is easy to setup, each device shipped will have a unique identifier and can be registered upon receive on the particle.io website. To use the service, it costs $3 per month for 3Mb of data and $0.60 extra for every additional 1MB of data. In our case however, the 1Mb should be good enough [6].

Of course, like a cellphone, the 3G transceiver must also have a data provider, and the one that Particle Electron uses in Canada is Rogers Wireless, according to their website [6].

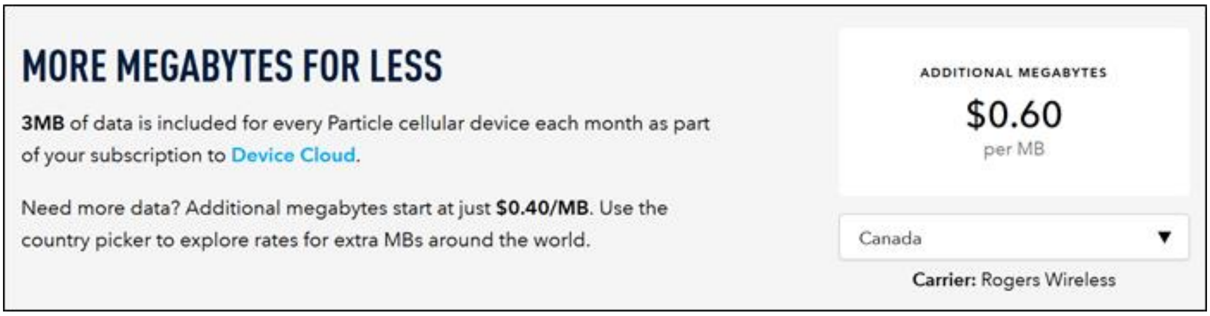

Figure 8: Particle Electron 3G Plan Details

However, we were still concerned with the coverage on the Rogers network, and so we checked the network coverage of its HSPA+ network (that is the name of its 3G network) and it seems Rogers wireless has very good coverage around the Calgary area [8].

Figure 9: Rogers Network Coverage Around Calgary Area

For each Particle Electron we are going to have to create an account on the particle.io website. We can program the Particle Electron online as well as send and receive data from it. The programming interface we use is on the particle website and uses the same syntax as Arduino, and we can program our device as well as send/receive data online since our device has a unique ID [9].

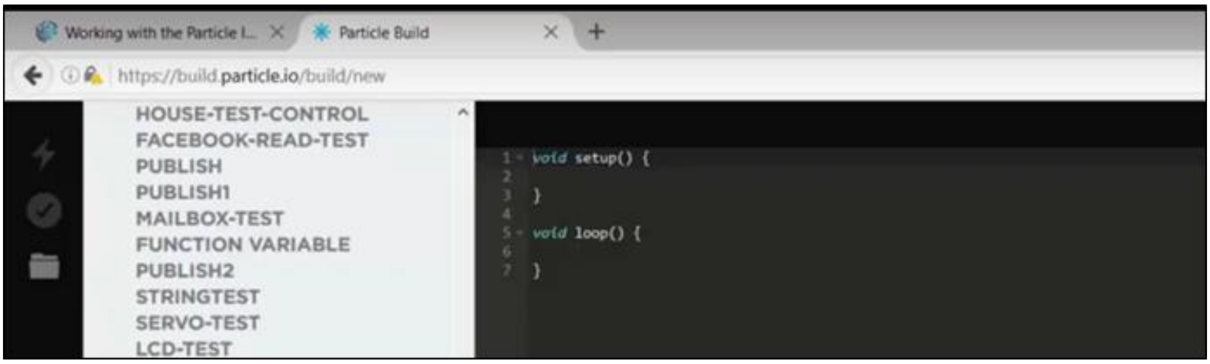

Figure 10: Particle Electron IDE

Here are the technical specifications of the Rogers HSPA+ (3G) network [10].

- The frequency used is 850MHz/1900Mhz
- Theoretical download speed of 21 Mbits/second

Alternative Product Considered

The *SIM5215A* module supports FTP/HTTP/HTTPS with maximum download and upload speed of 384 Kbps. An external memory SD card is also included in this module so that data can be stored in an SD card for redundancy. The module is compatible to our alternative microcontroller *Raspberry Pi 3* too. [17]

Figure 11: SIM5215A Module for Arduino[17]

The reason that this product was not considered was the lack of support online as opposed to the Particle Electron which has many resources. It was hard to find which network this module connects to in Canada and the monthly cost, which makes it difficult to include in our design.

Software

There are various software components that we need to implement for the project. The majority of the software components will be on the application level. However there will still be some software and data processing in the microcontroller. We will transmit the data that we gather from all the sensors to an online database. Furthermore, we will create a website that communicates with this database to display real time data from the sensors. The website could also potentially have a marketplace where the sponsor is able to sell and advertise the final product to customers from all around the world. Additionally, we will also create an app that also communicates with the online database and display the sensor data to the user's phone. The figure below explains the high level process of the application layer.

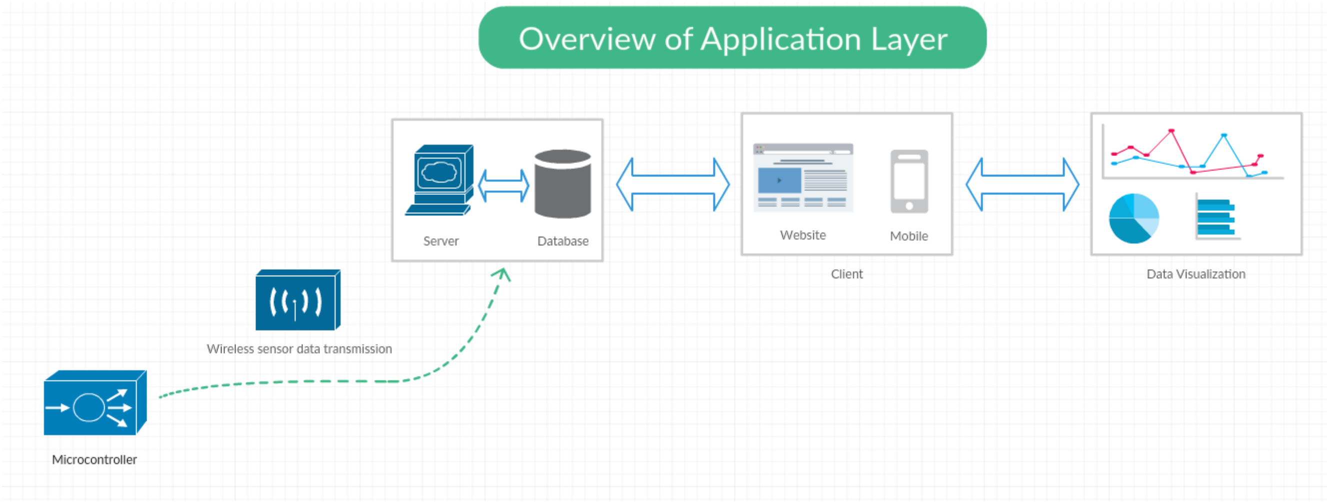

Figure 12: Overview of Application Layer

Database Structure

For the scope of this project, we will only need to create and use one table to store the sensor`s data. However on the long run, there will be the need to create additional tables. A customer table, for example, will be needed in the future to separate and differentiate the data from different horses. The table structure we are going to use to hold the data is shown in the figure below.

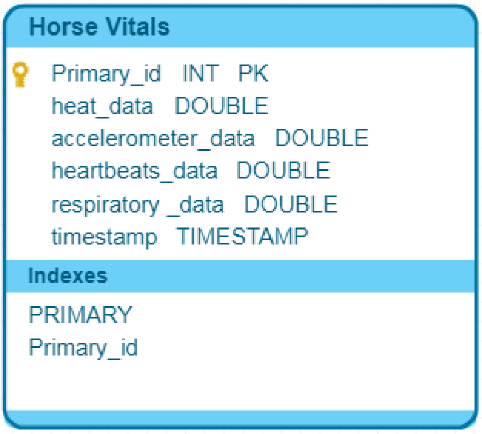

Figure 13: Table Structure to Hold Data

The primary id is used to keep track of the amount of data incoming. The sensor`s data are stored in the data fields of the table. Finally, a timestamp is used to keep track of the data`s arrival date and time. Eventually, a website and an app will be communicating with this database in order to visualize and process the data.

Website

The website is a major component of the overall deliverable. It will ultimately show live and processed data to the user. Therefore, it was important that we analyze the various options to build the website. The following list summarizes our options for creating the website

- Use a website builder to create the website
  - Wix
  - WordPress
  - Squarespace
- Create the website from scratch

Each of the options in the list above has its pros and cons. Initially, we analyzed the first level of options comparing creating the website from scratch against using a website builder. The following table summarized the advantages and the disadvantages of each option.

Options	Advantages	Disadvantages
Create website from scratch	More functionality Lots of support	Hard to develop Hard to maintain Hard to update
Create website using a builder	Flexible Lots of templates Easy to develop Easy to maintain Easy to update	Closed system Limited support

Table 1: Advantages and Disadvantages of Website Creation from Scratch vs Using a Builder

After analyzing the table above, we decided to create the website using a website builder. Subsequently we analyzed the following table to determine which website builder we were going to use to develop the website.

Options	Advantages	Disadvantages
Wix	Popular Flexible Lots of templates Easy to learn Easy to maintain Good Interface Lots of support	Forced to use Wix code for advanced functionality
WordPress	Popular Lots template Flexible	Blog based system Relatively hard to learn Limited support
Squarespace	Flexible	Relatively hard to learn

		Limited support

Table 2: Advantages and Disadvantages of Various Website Builders

After analyzing the table above we decided to use Wix to create the website. Wix had a lot of advantages and seemed perfect for the purpose and scope of this project. The website will communicate with the database to get, process and display the data.

Mobile Platform

In addition to creating a website, we will also create a mobile app. The app will complement the website and will display various data visualization. We decided to create an Android app over OS and Windows for the following reasons

- We have previous experience creating Android apps
- Android is the most popular mobile platform
- Cheapest developer cost among the three

In order to develop the Android app, we will need to use an integrated development environment (IDE). The two options we have are either Android Studio or Eclipse. We decided to use the Android Studio IDE over Eclipse for the following reasons

- Android Studio is the official IDE supported by Google
- Android Studio has way more functionality over Eclipse
- Android Studio has *modern* android virtual device (AVD) to test the app

The mobile app will also communicate with the database to get, process and display the data. We will try as best as we can to follow material design, since this is the recommended design for android developers and supported by Google.

Data Visualization

Regardless of the website and app, the data visualization will be consistent. We will have a dashboard that consists of

- Line graphs
- Donut graphs
- Bar graphs

The line graph will be used to show the real live sensor data. The donut graph will be used to show the averages during a 24 hour timeframe. Finally, the bar graph will be used to compare weekly performance and averages. We will consider other types of graphs and charts once we get into the testing and reliability phase of the project.

IP and Legal Agreements

The primary discussion about intellectual property was held between the capstone project team and Dylan Rae, the designated representative of HorseBit. All intellectual property rights will remain with HorseBit due to the extent of the project being only a proof of concept. All hardware, and software created for this project will be under the ownership of HorseBit. Additionally, the capstone group is not entitled to any future profit if HorseBit develops a device based on the work done in these two semesters.

Technical Specifications

The table below details technical specifications of our components and project:

Maximum Budget	**$5000**		
Function	**Name**	**Dimensions**	**Specifications**
Respiratory Rate & Heartbeat Sensor	ADS1292R Analog Front End IC	5mm x 5mm	Sample Rate (Max): 8 kSPS Resolution : 24 Bits Input Voltage Range (Max): 5.25 V –40°C to +85°C
Temperature	DS18B20	2mm x 2mm	Input Voltage (Range): 0.5V to 6.0V -55°C to 125°C
Motion Sensor	MinIMU-9 v5	20.32mm × 12.7mm x 2.54 mm	Input Voltage (Range): 2.5V to 5.5V Supply current: 5 mA
3D Enclosure	N/A	100mm x 90mm x 40mm	N/A
Leather Belt and Pouch	N/A	N/A	N/A
Microcontroller	Arduino Uno	68.6mm x 53.4mm [15]	Input Voltage (Range): 6V to 20V Supply current: 20 mA -40°C to 85°C [16]
Wireless Transmission	Particle Electron	50.8mm x 20.32mm x 12.7mm	Input Voltage (Range): 3.88V to 12V

			Supply current: 250 mA -30°C to 75°C [14]
Online Dropbox	Dropbox	N/A	2GB Storage [17]
Mobile Application	HorseBit App	N/A	OS: Android Mobile Application
Web Application	HorseBit Site	N/A	Website Building Software: Wix

Table 3: Technical Specifications

Materials, Supplies, Tools and Cost Estimates

Materials/Supplies/Tools	Function	Cost
ADS1292R Analog Front End IC	Respiratory Rate & Heartbeat Sensor	$70
DS18B20	Temperature Sensor	$5
MinIMU-9 v5	Motion Sensor	$30
3D Printed Enclosure	3D Enclosure	Free
Leather Belt and Leather Pouch	Leather Enclosure	≈$50
Arduino Uno	Microcontroller	$25
Particle Electron	Wireless Transmission	$69 + ($3/month per 3 MB + $0.60 per extra MB)
Dropbox	Cloud Server	Free
Wix Unlimited Package	Website	$18 per month
Android Studio IDE	App	Free
Total Cost:	--	$249 + $21/month

Table 4: Materials/Supplies/Tools Function and Cost

The total cost of our project with our current components comes out to be around $249 with an additional $21 per month for the use of Wix and Rogers services. All prices are accurate for the date of this document, and are taken from the line items respective websites. The leather belt and leather pouch has been approximated based on Amazon searches of the two items.

Additional costs for shipping and sped up shipping have not been included in these totals, as they are dependent on the final shopping carts. We have also not added any costs for the ordering of additional components if we receive defective or un-usable products, which is a likely case for our sensors. These two variables can increase the final costs estimates by a significant amount.

Some of the specialized tools we will use throughout project include a 3D printer, and soldering equipment. Both of these are free through the University of Calgary.

Risks and Risk Mitigation Plans

There are various risks involved in very stage of the project that could alter the initial schedule set out. For the foreseeable risks, the project team has come up with mitigation plans so that project timelines are not impacted majorly. The risks for this project can be broken down into three primary categories; management failure, component failure, and integration failure.

Management Failure

This category outlines those risks that are schedule impacting but not as a result of a component of a project not being able to be solved or a component not working. This includes the risk of timely incompletion because of unavailability of team members, delayed delivery of hardware, and inaccessibility to resources.

Unavailability of Team Members:

Although the project schedule has been created with the basis that there will be weekly meeting and that a certain amount of work will be completed every week, it is unreasonable to expect that team members will always be available. Unavailability of group members can be due to factors such as vacation, sickness, and burden from other courses. This may cause an individual to not complete his weekly workload, cause him to miss a deadline or be unavailable for meetings.

To combat this we have created secondary leads that are there to assist the primary lead. This helps spread out the workload and one person can pick up a greater load during a certain week if the other is missing or is busy with heavy course load such as midterms.

Additionally, we have also agreed to meet during holidays and weekends if needed to catch up with the schedule. Longer meetings will take place during the weekend and holidays close to the deadline if we have yet not completed the work. By creating extra dedicated time for the project, it allows the team to place their entire focus on that work during that time.

Delayed Delivery of Hardware:

Sensors and hardware are an essential portion of the project that are dependent on the delivery of those components to us in Calgary. Based on online estimates, the maximum amount of time that it will take our components to arrive is around 4 weeks, but it is irrational to believe that no delays can take place within the delivery.

To mitigate this problem our group will place greater emphasis on the portions of the project that don’t require this components so that even if one part of the project is delayed, we are ahead on another part. By doing this we will have additional resources and time available to dedicate to the hardware portions of the project and the overall schedule of the project will not be impacted greatly.

Our large budget also provides us an extra step of mitigation in this. If a piece of hardware is absolutely crucial, we can pay for faster delivery. This is also useful if we need to order extra parts due to malfunction.

Inaccessibility of Resources:

A part of our project is dependent on other people and resources being available for use to us. Specifically this includes resources through our sponsors, such as access to veterinarians and horses for testing, which may be unavailable to us exactly at the time that we require to remain on schedule. To mitigate the unavailability of veterinary resources we will use text resources provided to us by HorseBit for our research. We will place extra time in our testing schedule to account for testing horses not being available immediately.

Component Failure

A significant risk to the project is the delay or failure of one of its various components. These components each have their own portions that are sources of risk. The following table helps outline and summarize the risks, their outcome and probability, and the mitigation plan:

Risk	Risk Summary	Mitigation Plan
Enclosure		
3D Printing	As an object is 3D printed and cooled, there is a chance for the material to become bent and out of shape.	3D printing is free at the university so we can get our object 3D printed early to check for flaws.
Leather Pouch	Leather pouch may not be of suitable dimension to integrate 3D printed enclosure.	Use 3D printed enclosure to size for the leather pouch and have adjusting straps within pouch.
Sensors and Processing		
Unexpected Behavior	Although there has been research done to determine our sensors, there is a risk that they will not	Create a list of alternative sensors that can be used as potential replacements.

	behave as expected and give us garbage results.	
Inconsistent Readings	Another risk factor is getting inconsistent readings that do not accurately reflect what is being measured. For example readings being consistent and then having a single spike.	Processing algorithm detects and isolates inconsistent readings. The software will not include these readings in the calculation but they will be noted down in a log file for extra information.
Microcontroller Processing Limits	There is a risk that our Arduino microcontroller may not be able to handle the amount of inputs and outputs that we require of it.	We will purchase a replacement Raspberry Pi 3 microcontroller which has much higher capabilities in terms of processing power due to its use of multithreading. We may also increase the amount of seconds between samples from 5 to 10 seconds.
Wireless Transmission		
Coverage and Data Transfer Speed	Our choice of 3G is limited by the coverage provided by it, as well as the data rates that it provides which could prove to be too slow for our application.	The sponsor has confirmed that this is not an essential part of the project, so an SD card directly attached to the microcontroller can be used for data storage instead.
User Software		
Flaw in UI	There's a chance that the sponsor may not be happy with the design and UI of the app or website.	This can easily be mitigated by sending regular updates to the sponsor with the current UI and functionality of our user software.

Table 5: Component Failure Risks, Risk Summary and Mitigation Plan

Integration Failure

Integration failure is one of the greater risks to our project schedule. The reason for this is because integration of our components happens far later into the project and any delays could push us very close to or over the project deadline.

To mitigate the risk of failure we will do testing in each phase of our project for compatibility and also test individual portions with sample data. The following table is a summarization of our plans:

Integrating Components	Risk	Mitigation Plan
Sensors -> Signal/Data Processing	Sensors may not give readable or usable data to the microcontroller.	Testing of our sensors, attached to the microcontroller on their arrival. Sensor data can be taken from humans if horses are unavailable. Processing of data will be done with sample data obtained from our veterinary contacts so that it is not dependent on sensor availability.
Data Processing -> Data Transmission	Processed Data is not transmitted in its entirety or not as quickly as intended. Data may not be able to be transmitted at the place of testing on a horse, which is located outside city limits.	Wireless transmission will be regularly tested with junk data similar to that which will be supplied by the final processed data. Once the transmission portion of the project is complete, we will test it outside city limits with junk data.
Data Transmission -> Data Storage	Data is stored into cloud server in a manner which makes it hard for website/app to read.	Sample data will be created to use in place of real data to test the behavior and form of our stored data.
Data Storage -> User Software	Data Storage must be constantly available to the user software and both user	Sample data will be injected into storage for use of user software to test this functionality.

	software must be able to access data from different devices.	
Complete Device -> Testing	A risk exists that the device, when put together, may not function when placed on a horse.	To mitigate this we will do testing as each individual component is completed, rather than depending on a final test.

Table 6: Integration Failure Risks, Risk Summary and Mitigation Plan

Major Technical Tasks and Milestones

The project technical tasks and milestones are tracked through a GANTT chart created through the use of the GanttProject tool. This is not meant to act as an exhaustive list of every single deadline that we have set for the project, but instead an overview of the most important deadlines. This chart will be updated weekly to reflect the state of the project.

The chart starts with the completion of the initial design which is marked with the Design Roadmap Document as the milestone. The chart is presented below:

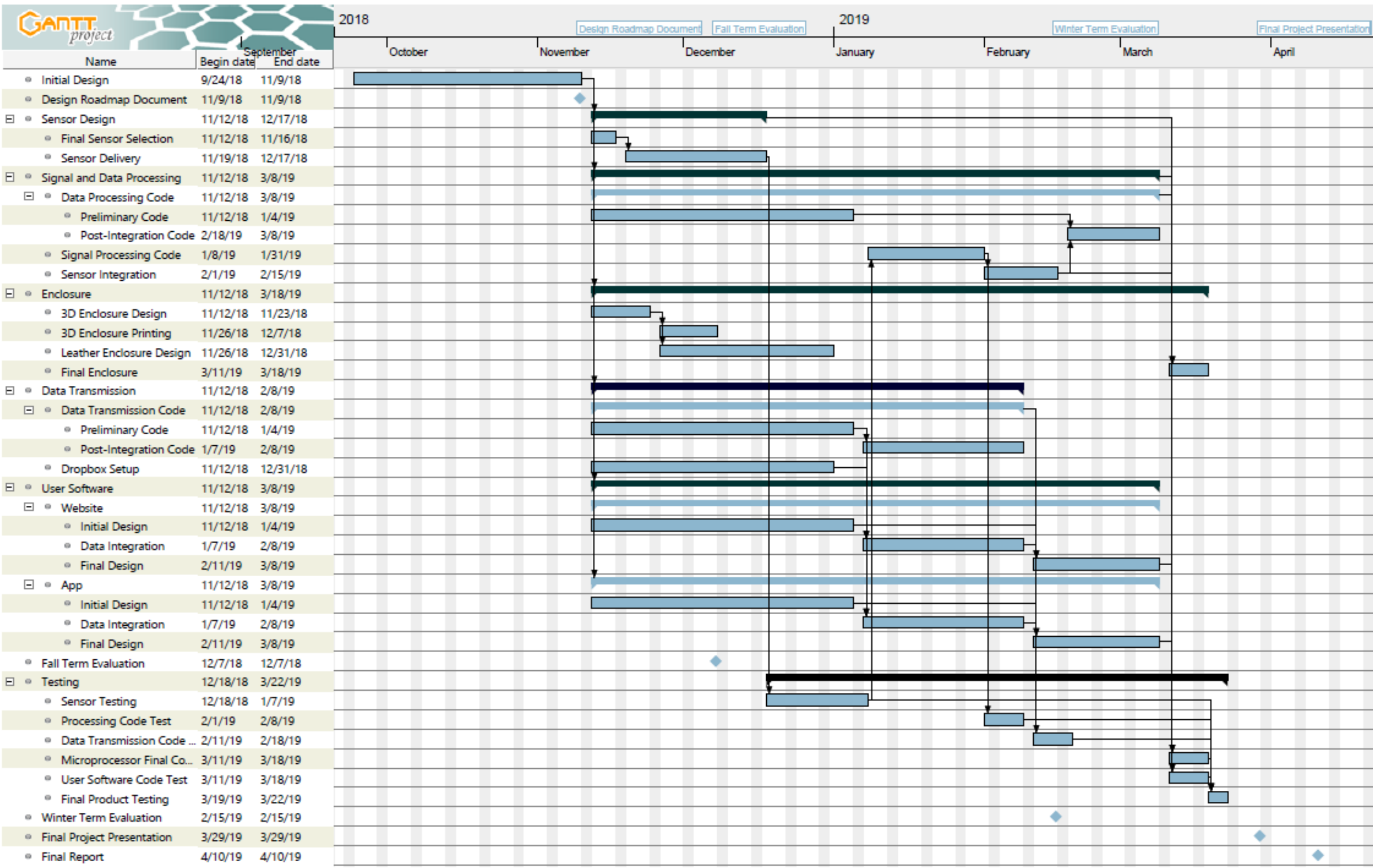

Figure 14: GANTT Chart (Larger Image in Appendix)

This table summarizes each of the tasks:

Task	Task Summary and Deliverables
Sensor Design	
Final Sensor Selection	This tasks involves selection of final sensors to order following advice sessions with the TA, Academic Advisor and Sponsor. The deliverables will be a list of sensors to be ordered by the sponsor for use in our project.
Sensor Delivery	This task is included to identify the expected amount of time for delivery of our sensors. We have indicated a month for them to arrive.

Signal and Data Processing	
Data Processing Code	The data processing code portion of the project is split into a preliminary design and post-integration design portion. Ideally we'd have signals to work with, but in lieu of that, code will be developed using baseline metrics obtained from online studies and veterinary resources. The deliverable will be the code itself and the outputs indicate the health of the horse for use by the transmission portion of the project.
Signal Processing Code	This code can only be written once the sensors have arrived and have been tested to work with the microcontroller. The deliverables are the code itself and outputs for the use of the data processing code.
Sensor Integration	Sensors will be integrated completely with the completion of the signal processing code. There is no deliverable.
Enclosure	
3D Enclosure Design	Design of our 3D enclosure based on our initial design for the project. Deliverable is an AUTOCAD file with the specifications and design ready for 3D printing.
3D Enclosure Printing	Amount of time given for printing of our 3D enclosure design. Deliverable is the 3D enclosure.
Leather Enclosure Design	Design of our leather enclosure, including dimensions. Deliverable is the leather enclosure.
Final Enclosure	Final design of the enclosure once sensor design is complete. The deliverable is the integration of the 3D printed enclosure with the final device present and the leather enclosure.
Data Transmission	
Data Transmission Code	The data processing code portion of the project is split into a preliminary design and post-integration design portion. Preliminary code will be created with sample data, and will then be updated to work with the outputs of our signal and data processing code. Output is the code and data to the Dropbox.
Dropbox Setup	Simple setup of Dropbox for use as storage.
User Software	
Initial Design	Initial design of UI for both of our user software's. Output is a website and app to present to the sponsors for review.

Data Integration	Data integration involves using sample and finalized data to create graphs of measured vitals and indicators of health for our user software. Deliverables are an updated website and app for the review of our sponsors.
Final Design	Final design of website and app with sponsor feedback reflected.
Testing	
Sensor Testing	Testing of our sensors on arrival to see the data that they provide for our use. Will determine if our initial design is ready to implement or if new sensors need to be ordered.
Processing Code Test	Testing of our Signal and Data Processing portion of our project. Inputs to the test will be our sensor data and outputs will be viewable vitals data and an indication of the horse's health.
Data Transmission Code	Testing of our data transmission code. Inputs to the test will be our processing code data and outputs will be a viewable list of that data within our Dropbox.
Microcontroller Final Code Test	The microcontroller final code is a test of the integration between our processing code and transmission code. A test will be run to see if the microcontroller can read, process, and then send the code without failure.
User Software Code Test	Final test of the user software. Inputs will be the data present in the Dropbox and outputs will be the graphs on the website and app, as well as the identifiers of health included.
Final Product Test	A test of our final product: the proof of concept device. Device will be strapped on to the horse and we will check to see if the app and website are displaying that data in real time.

Table 7: Tasks and Task Summaries and Deliverables

Roles and Responsibilities

The basic hierarchy of this team is set up with a single long-term project manager, and every member of the team having a role as a technical lead in different portions of the project. These roles have been decided based on the previous experience that the group has developed over their education and internships, as well as based on the interest shown by an individual in that role. The following table summarizes the expectation of these roles and who has been assigned each of those roles:

Role	Role Expectation	Assigned to
Project Lead	This individual is responsible for creating and updating the project schedule and deadlines, and making sure that the entire team adheres those deadlines. The project lead is also tasked with communication with external resources (e.g. sponsor, teaching assistant, professor) and is the point of contact for internal issues. The project lead will make sure that all team members adhere to the team contract and is responsible for reviewing and submitting all project documentation.	Muhammad Hussain
Technical Lead	This individual will be responsible for all deliverables under the specific technical component of the project they are assigned to. They will be responsible for documentation involving the technical portions of the project that they are assigned.	Every Member
Secondary Lead	Each technical lead will be assigned as a secondary lead for a different technical component of a project. Their job will be to provide aid and a secondary source of knowledge and ideas for that component.	Every Member
Meeting Minute Recorder	This individual is responsible for the upkeep of the Google Docs Team Logbook. The recorder must make sure that all meeting discussions, design decisions, role and responsibility decisions and any other significant decisions are accurately recorded in the logbook.	Hussain Qureshi

Mediator	This individual is tasked with running the weekly project update meetings. He must make sure that that meetings remain structure and give every individual an opportunity to communicate on each topic. This individual is also responsible for mediating discussions between members if issues arise.	Hussain Majeed

Table 8: Roles and Role Expectations for Project Team

Each technical lead is given a portion of the project as their responsibility based on the deliverables of the project. Additionally, each member of the team will act as a secondary lead for a different technical component of the project to help avoid overburdening and stress. The following table highlights how we have decided to split up the project and the members who are responsible for each portion:

Component	Technical Lead	Secondary Lead
Sensor Design	Muhammad Hussain	Hussain Qureshi
Hardware Implementation	Hussain Qureshi	Haoxian Zhang
Signal and Data Processing	Haoxian Zhang	Muhammad Hussain
Data Transmission	Hussain Majeed	Moaz Barakat
User Software	Moaz Barakat	Hussain Majeed

Table 9: Technical and Secondary Leads for Project Components

A RASCI (Responsible, Accountable, Supportive, Consulted, Informed) Chart has been created to help outline the responsibilities that each member and role will have assigned to them.

	Muhammad Hussain	Hussain Qureshi	Hussain Majeed	Haoxian Zhang	Moaz Barakat
Documentation	R,A	S,C,I	S,C,I	S,I	S,I
Meeting Moderation	S,C		R		
Logbook	S,C	R,A	S	S	S
Enforce Team Contract	R	I	R	I	I
Sensor Design	R,A	R,S,C	I	I	I
Hardware Implementation	I	R,A	I	R,S,C	I
Signal and Data Processing	R,S,C	I	I	R,A	I
Data Transmission	I	I	R,A	C,I	R,S,C
User Software	I	I	R,S,C	I	R,A
Testing	R,A	R	R	R	R

Table 10: RASCI Chart

Measures of Success and Validation Tests

The measures of success and validations tests will be broken down into categories based on our project components.

Sensors

Successful Product: Sensors are able to get readings from the horse and this data is collected by the Arduino Uno.

Validation Test: Sensors will be tested on objects with a set of known values. The overall performance of the measured values needs to stay within 5% tolerance, otherwise the sensors will not pass the test and will be marked as defective.

The complete device will be placed on the horse and the sensor data will be analyzed. The data will be checked while the horse is standing still and then again after it has warmed up and moved around for 10 minutes. This will help us see if the data is accurate; if the heart and respiratory rate go up, if the accelerometer reads an increase in acceleration and if temperature rises.

Enclosure/Harness

Successful Product: The leather belt and pouch is comfortable for the horse to wear and straps on well enough to detect vitals. All modules have their wires soldered and are glued to the inside of the 3D enclosure (placed in the pouch). The horse's movements should not displace any of the parts inside the enclosure. The sensors must be able to get their readings when placed inside the 3D printed and leather enclosure.

Validation Test: The device will be tested by being placed on a horse for a duration of 30 minutes to see if provides any discomfort to the horse. The horse will be allowed to move freely in this time, and we will check to see if the device works at the end of the period.

Signal and Data Processing

Successful Product: The microcontroller is able to receive data from the sensors every five (5) seconds. Received data should reflect a horse's vitals accurately within 5% error. The received data and historical data is able to store in the flash memory of the microcontroller. The algorithm has access to the data in its memory and produces accurate results directly associated with the health of a horse.

Validation Test: The validation tests for this is similar to that for testing the sensors, as we need their data to determine if the signal and data processing is correct. Following the test that we perform for signal processing, we can see the output to determine if the horse is healthy.

Since the horses we will be testing on will likely be healthy, it provides a challenge to test for the yellow and red ranges of our product. To “induce” colic symptoms, we will change the code to hold the accelerometer at 0 and let the horse move around. Due to the rising heart rate, temperate and respiratory rate this will cause, the microcontroller should determine that there are colic symptoms in the horse and alert the user.

Wireless Transmission

Successful Product: The Particle Electron 3G transceiver is able to read processed signal data from the microcontroller and can send it to the online IDE, from where it can be sent to the Dropbox.

Validation Testing: To test the transmission outside the system, we will apply a 5 volt signal to one of the IDE’s digital inputs and see if it can detect it and display it on its output console.

For testing within the system we will feed sensor data to the microcontroller and see if we get the correct outputs within the Dropbox storage that we set up.

User Software

Successful Product: The app and website will be considered successful based on two metric. The first is the functionality of its graphs and alert systems for horse health. If the app and website are able to display the data we collect from a horse and also show if it’s healthy or not, it’ll be considered successful. The second metric is its UI. The app and website must look professional and must be deemed acceptable by the sponsors.

Validation Testing: To test the graph and health alert functionality, we can feed simulated data into the app and website through Dropbox and check the display. The app and website will be under constant review by the sponsor, who will provide us feedback to improve it.

Entire Project

The entire project will be deemed successful based on the results of each individual component, and the sponsor’s opinion on the project as a whole. The project will be deemed extraordinary if we are able to pass all the validation tests listed above as an entire system. For an acceptable project, we would require the respiratory rate and heart rate sensors to be functioning, the microcontroller be able to process their data and the user software be able to display that data within a clean UI.

Appendix

Respiratory Rate & Heartbeat Sensor

ADS1292R

ELECTRICAL CHARACTERISTICS

Minimum and maximum specifications apply from –40°C to +85°C. Typical specifications are at +25°C. All specifications are at DVDD = 1.8 V, AVDD – AVSS = 3 V(1), V_{REF} = 2.42 V, external f_{CLK} = 512 kHz, data rate = 500 SPS, C_{FILTER} = 4.7 nF(2), and gain = 6, unless otherwise noted.

	PARAMETER	TEST CONDITIONS	ADS1291, ADS1292, ADS1292R MIN	TYP	MAX	UNIT
ANALOG INPUTS						
	Full-scale differential input voltage (AINP – AINN)			$\pm V_{REF}$ / gain		V
	Input common-mode range			See the *Input Common-Mode Range* subsection of the *PGA Settings and Input Range* section		
	Input capacitance			20		pF
	Input bias current (PGA chop = 8 kHz)	T_A = +25°C, input = 1.5 V			±200	pA
		T_A = –40°C to +85°C, input = 1.5 V		±1		nA
		Chop rates other than 8 kHz		See *Pace Detect* section		
	DC input impedance	No pull-up or pull-down current source	1000			MΩ
		Current source lead-off detection (nA), AVSS + 0.3 V < AIN < AVDD – 0.3 V		500		MΩ
		Current source lead-off detection (µA), AVSS + 0.6 V < AIN < AVDD – 0.6 V		100		MΩ
PGA PERFORMANCE						
	Gain settings			1, 2, 3, 4, 6, 8, 12		
	Bandwidth	With a 4.7-nF capacitor on PGA output (see *PGA Settings and Input Range* section for details)		8.5		kHz
ADC PERFORMANCE						
	Resolution		24			Bits
	Data rate	f_{CLK} = 512 kHz	125		8000	SPS
CHANNEL PERFORMANCE (DC Performance)						
	Input-referred noise	Gain = 6(3), 10 seconds of data		8		μV_{PP}
		Gain = 6, 256 points, 0.5 seconds of data		8	11	μV_{PP}
		Gain settings other than 6, data rates other than 500 SPS		See *Noise Measurements* section		
	Integral nonlinearity	Full-scale with gain = 6, best fit		2		ppm
	Offset error			±100		µV
	Offset error drift			2		µV/°C
	Offset error with calibration			15		µV
	Gain error	Excluding voltage reference error		±0.1	±0.2	% of FS
	Gain drift	Excluding voltage reference drift		2		ppm/°C
	Gain match between channels			0.2		% of FS
CHANNEL PERFORMANCE (AC performance)						
CMRR	Common-mode rejection ratio	f_{CM} = 50 Hz and 60 Hz(4)	–105	–120		dB
PSRR	Power-supply rejection ratio	f_{PS} = 50 Hz and 60 Hz		90		dB
	Crosstalk	f_{IN} = 50 Hz and 60 Hz		–120		dB
SNR	Signal-to-noise ratio	f_{IN} = 10 Hz input, gain = 6		107		dB
THD	Total harmonic distortion	10 Hz, –0.5 dBFs, C_{FILTER} = 4.7nF		–104		dB
		100 Hz, –0.5 dBFs, C_{FILTER} = 4.7nF		–95		dB
		ADS1292R channel 1, 10 Hz, –0.5 dBFS, C_{FILTER} = 47 nF		–82		dB

(1) Performance is applicable for 5-V operation as well. Production testing for limits is performed at 3 V.
(2) C_{FILTER} is the capacitor accross the PGA outputs; see the *PGA Settings and Input Range* section for details.
(3) Noise data measured in a 10-second interval. Test not performed in production. Input-referred noise is calculated with input shorted (without electrode resistance) over a 10-second interval.
(4) CMRR is measured with a common-mode signal of AVSS + 0.3 V to AVDD – 0.3 V. The values indicated are the minimum of the two channels.

Table 11: ADS1292R Ratings

Temperature Sensor

DS18B20

DC Electrical Characteristics

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Supply Voltage	V_{DD}	Local power (Note 1)		+3.0		+5.5	V
Pullup Supply Voltage	V_{PU}	Parasite power	(Notes 1, 2)	+3.0		+5.5	V
		Local power		+3.0		V_{DD}	
Thermometer Error	t_{ERR}	-10°C to +85°C	(Note 3)			±0.5	°C
		-30°C to +100°C				±1	
		-55°C to +125°C				±2	
Input Logic-Low	V_{IL}	(Notes 1, 4, 5)		-0.3		+0.8	V
Input Logic-High	V_{IH}	Local power	(Notes 1,6)	+2.2		The lower of 5.5 or V_{DD} + 0.3	V
		Parasite power		+3.0			
Sink Current	I_L	$V_{I/O}$ = 0.4V		4.0			mA
Standby Current	I_{DDS}	(Notes 7, 8)			750	1000	nA
Active Current	I_{DD}	V_{DD} = 5V (Note 9)			1	1.5	mA
DQ Input Current	I_{DQ}	(Note 10)			5		µA
Drift		(Note 11)			±0.2		°C

Note 1: All voltages are referenced to ground.
Note 2: The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU}. In order to meet the V_{IH} spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$.
Note 3: See typical performance curve in Figure 1. Thermometer Error limits are 3-sigma values.
Note 4: Logic-low voltages are specified at a sink current of 4mA.
Note 5: To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
Note 6: Logic-high voltages are specified at a source current of 1mA.
Note 7: Standby current specified up to +70°C. Standby current typically is 3µA at +125°C.
Note 8: To minimize I_{DDS}, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or V_{DD} – 0.3V ≤ DQ ≤ V_{DD}.
Note 9: Active current refers to supply current during active temperature conversions or EEPROM writes.
Note 10: DQ line is high ("high-Z" state).
Note 11: Drift data is based on a 1000-hour stress test at +125°C with V_{DD} = 5.5V.

Table 12: DS18B20 Ratings

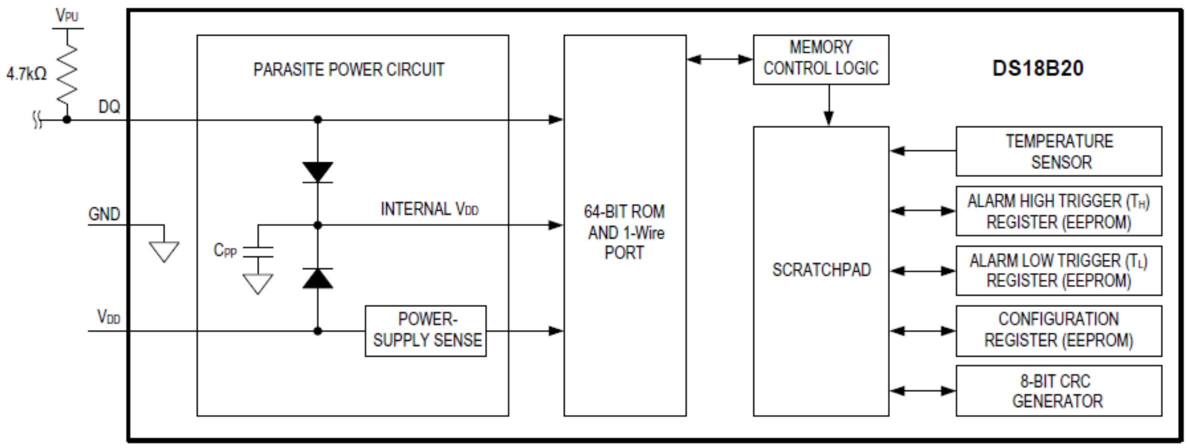

Figure 15: DS18B20 Block Diagram

Motion Sensor

MinIMU-9 v5

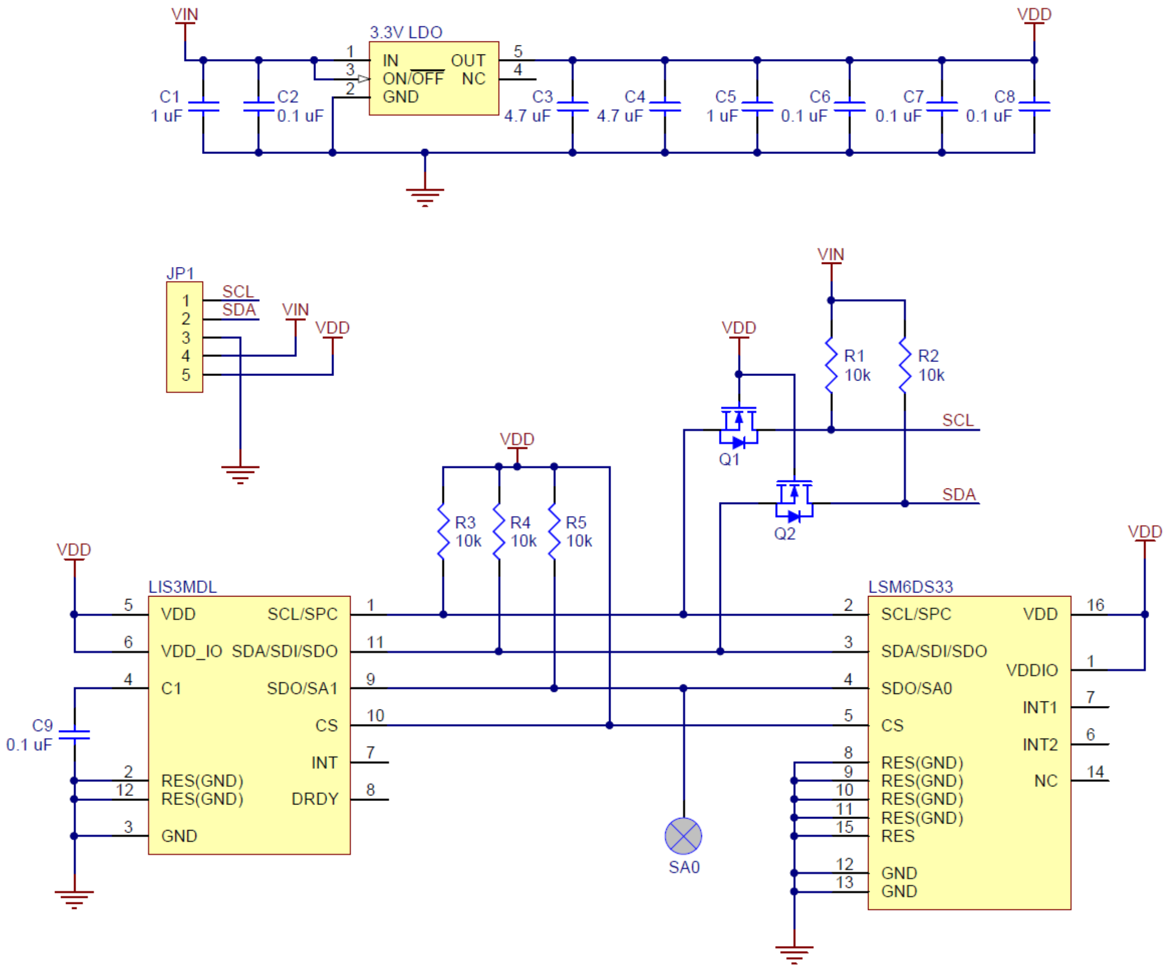

Figure 16: MinIMU-9 v5 Electrical Schematic

Microcontroller

Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 13: Arduino Uno Ratings

GANTT Chart

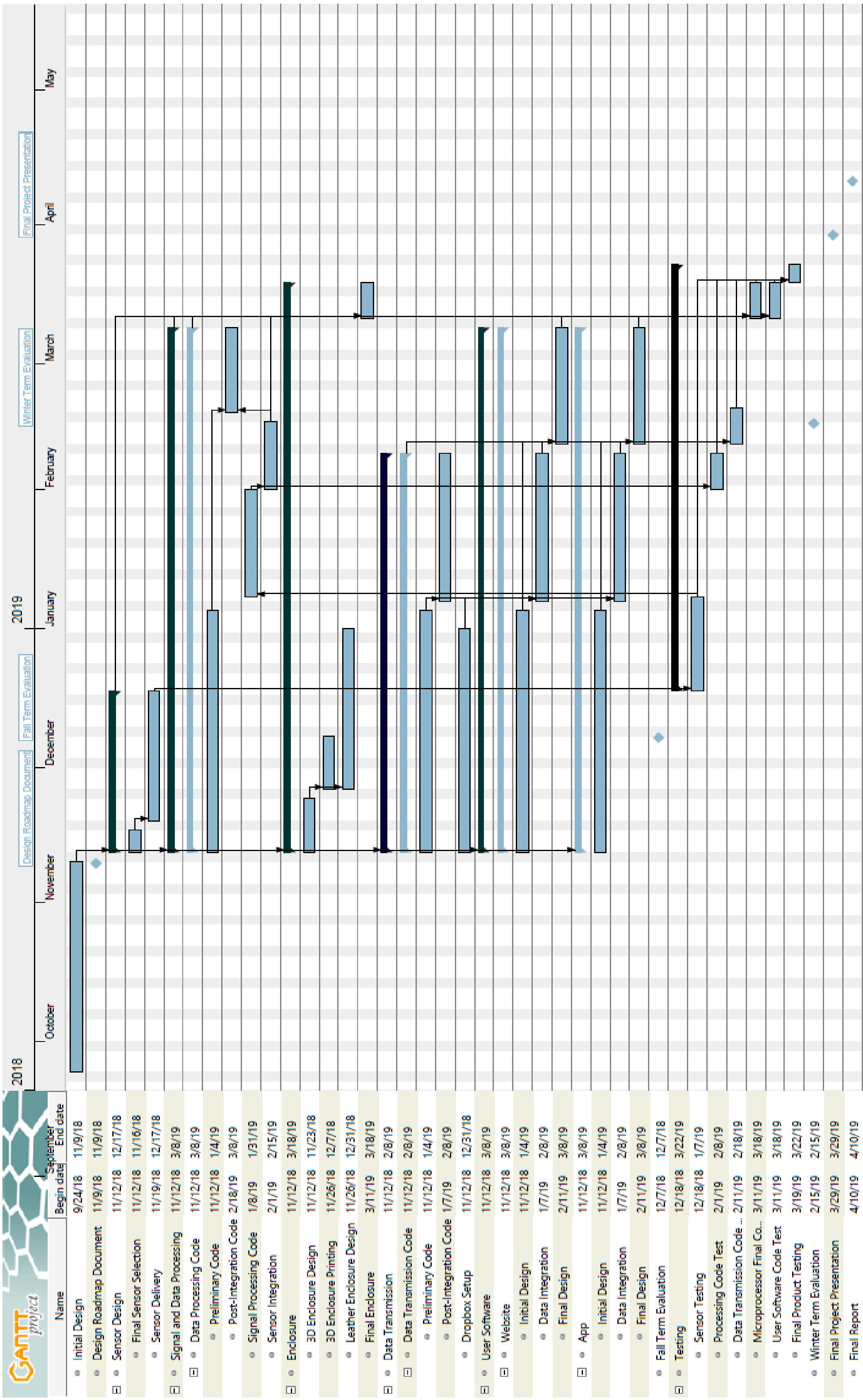

Figure 17: Enlarged GANTT Chart

References

[1] Petmd.com. (2018). Colic in Horses | petMD. [online] Available at: https://www.petmd.com/horse/conditions/digestive/c_hr_equine_colic?page=2 [Accessed 3 Oct. 2018].

[2] Jurga, F. (2017). Will we compete again? Statistics favor survival, return to form for horses following colic surgery. [online] The Horse Owner's Resource. Available at: https://equusmagazine.com/blog-equus/research-statistics-survival-horses-colic-surgery-55350 [Accessed 3 Oct. 2018].

[3] A. Soehn, "Measuring the Heart – How do ECG and PPG Work?", Mar 21, 2017. [Online]. Available: https://imotions.com/blog/measuring-the-heart-how-does-ecg-and-ppg-work/. [Accessed Oct. 29, 2018].

[4] "Pulse Sensor". [Online]. Available:https://pulsesensor.com/products/pulse-sensor-amped. [Accessed Oct. 29, 2018].

[5] "MinIMU-9 v5 Gyro, Accelerometer, and Compass (LSM6DS33 and LIS3MDL Carrier)". [Online]. Available:https://www.pololu.com/product/2738. [Accessed Oct. 30, 2018].

[6] Global Cellular IoT. (n.d.). Retrieved November 12, 02, from https://www.particle.io/cellular/

[7] Particle Retail. (n.d.). Electron. Retrieved November 2, 2018, from https://store.particle.io/collections/electron

[8] Our network. (n.d.). Retrieved November 2, 2018, from https://www.rogers.com/consumer/wireless/network-coverage

[9] Electronics, C. (2016, October 14). Retrieved November 02, 2018, from https://www.youtube.com/watch?v=b6sUP16HWKM

[10] Matrix Mobile. (n.d.). Canadian Cellular Frequencies. Retrieved November 02, 2018, from https://matrixmobile.ca/frequencies/#1471801246119-d4d3c0f2-a6b2

[11] Electronicsnotes. (n.d.). LoRa Wireless for M2M & IoT. Retrieved November 02, 2018, from https://www.electronics-notes.com/articles/connectivity/lora/what-is-lora-basics-m2m-iot.php

[12] Unuth, N. (2018, October 14). What Is 3G Wireless Technology? Retrieved November 02, 2018, from https://www.lifewire.com/what-is-3g-3426465

[13] CableFree. (n.d.). Data Rates in 802.11n. Retrieved November 02, 2018, from https://www.cablefree.net/wirelesstechnology/wireless-lan/data-rates-in-802-11n/

[14] BiPOWER. (n.d.). LITHIUM-ION RECHARGEABLE BATTERIES LP103450. Retrieved November 03, 2018, from http://www.bipowerusa.com/products/bipower-li-ion-lp103450-1750.pdf

[15] GoodHiding. (n.d.). Strong Pouch and Matching Belt [Digital image]. Retrieved November 03, 2018, from http://www.goodhiding.co.uk/images/keypouch1.jpg

[16] Matrix Mobile. (n.d.). Canadian Cellular Frequencies. Retrieved November 04, 2018, from https://matrixmobile.ca/frequencies/#1471801245339-44fd6b9f-0d33

[17] Cooking Hacks. (n.d.). 3G/GPRS shield for Arduino, Raspberry Pi and Intel Galileo. Retrieved November 04, 2018, from https://www.cooking-hacks.com/3g-gprs-shield-for-arduino-3g-gps

[18] Gudino, M. (2018, September 10). Arduino Uno or Raspberry Pi 3 - How to Decide. Retrieved November 04, 2018, from https://www.arrow.com/en/research-and-events/articles/comparing-arduino-uno-and-raspberry-pi-3

[19] Particle Docs. (n.d.). Particle. Retrieved November 04, 2018, from https://docs.particle.io/datasheets/cellular/electron-datasheet/

[20] Arduino. (n.d.). ARDUINO UNO REV3. Retrieved November 04, 2018, from https://store.arduino.cc/usa/arduino-uno-rev3

[21] MicroChip. (n.d.). ATmega328P. Retrieved November 04, 2018, from https://www.microchip.com/wwwproducts/en/ATmega328p

[22] DropBox. (n.d.). How do I create a Dropbox account? Retrieved November 04, 2018, from https://www.dropbox.com/help/account/create-account