End of Project Documentation

Electrical Assisted Bicycle

April 2019
Team 8
Pranay Chaudhary
Isaac Parrish
Harjot Sidhu
Veronica Gonzalez
Prof. Russ Tatro

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

EXECUTIVE SUMMARY

Our project is an electric-assist bicycle to help reduce the risk of obesity. In this report we provide a detailed explanation of the work performed and our experience with this project over our timeline from Fall 2018 to Spring 2019. A significant number of people are obese or morbidly obese with the availability and ease of access to unhealthy foods such as those from fast food restaurants. Our project focused on providing those who are obese an opportunity to exercise with our electric assisted bicycle. It keeps the rider's heart rate within a target range of their maximum heart rate and attempts to improve the exertion of their body while riding. It will provide a physically beneficial riding experience. Our bicycle uses a hub motor attached to a bicycle wheel to provide the motor assistance necessary to either cause pedaling or to reduce the pedaling necessary by the rider. The motor will receive the power necessary to adjust through the measurements of the rider's heart rate from the pulse sensor attached to the handlebar. Once the rider places the pulse sensor on their finger it will send measurements to the Arduino Uno through which our software programing done in C will determine the range the heart rate falls within. For fifty percent of the maximum or lower, the motor will receive no power and encourage the user to pedal more in order to increase physical exertion. For a range of sixty-five percent to eighty-five percent, the motor will receive incremental increases or decreases in power. If the rider's heart rate is higher than eighty-five percent of the maximum heart rate, the motor will receive full power and provide complete pedal assistance in order to reduce the strain on the rider. The temperature sensor in our project will provide an ambient temperature of the environment to the user in order to determine whether riding is beneficial, too hot, or too cold in current temperatures. There is also a set speed option which the rider can set as being the maximum possible speed they will ride at. The algorithm will then adjust the increments and decrements in power to the hub motor so that it does not surpass this speed. Our Raspberry Pi board will communicate with the Arduino Uno in order to display the heart rate, speed, set speed, and temperature measurements on a screen for rider's to easily be able to see data. This ease of access to the data collected can allow for riders to determine whether their physical exertion has improved over time and if they are truly benefitting from the product. Throughout this report we detail our progress, testing, research, and results.

Abstract* - Our team worked on providing an engineering solution to the societal problem of obesity. We sought to find a solution which would allow obese people to be able to lose weight. Our solution was to improve physical fitness with bicycling which would attempt to keep the user's heart rate in a range that would lead to the optimal amount of fitness for the user without overexerting their body. There can be a much higher prevalence in obesity due to the high-density diet available from modern foods such as those found in fast food restaurants. The use of biking as a solution and providing useful information for personal health when biking can increase an interest in controlling obesity and maintaining a healthier lifestyle. Not all obese people are capable of riding a bicycle long distances but many who are capable have a hard time doing so because they are somewhat out of shape and would benefit from an electric assist to help them ride longer distances without burning out. Our prototype consisted of a brushless DC motor, a pulse sensor, a temperature sensor, and three lead acid batteries. The boards used to program our project were an Arduino Uno and a Raspberry Pi. The electric-assist bicycle will determine the amount of assistance the hub motor provides based on how close to their maximum heart rate the user is exerting themselves. Our prototype will also provide a data log of the user's heart rate for the duration of the riding session, as well as, their speed, the temperature, and the set speed. Throughout this report we will provide the details of the design, funding, risk assessment, marketability, and current status of our prototype.

Keyword index:

Application, Arduino, Battery, Bicycle, Body Mass Index, Communication, Consumer, Data, Data Logging, Design, Electric, Heart Rate, Hub Motor, Marketability, Motor, Obesity, Pulse Sensor, Raspberry Pi, Risk

I. INTRODUCTION

Our product of an electric-assisted bicycle is targeted to overweight collegiate students who have little time to exercise. With our product they will be able to extend their exercising level, completing more fitness activity than they were able to complete beforehand. Our targeted consumer of our work is overweight college-level students. The reason for this is that many collegiate students often end up gaining more weight than expected their first year of college, starting a trend that keeps going as they continue onto their years in school. The official yet non-official term for this is the Freshman 15. This can be contributed to new eating habits, eating while studying, eating late at night, eating snack food, emotional eating, alcoholic drinking, and not enough exercise. While our electric-assisted bicycle may not be able to fix all of these issues to eliminate the Freshman 15, we can deliver help for an essential part of the solution. This crucial part being the ability to exercise. With most college campuses made with large traveling distances from class to class or class to dorms/facilities, having a bike is extremely handy. However, since our targeted consumer is overweight collegiate students, even riding a traditional bike may seem hard at times and difficult to continue riding due to low endurance levels for this specific exercise. With an electric-assisted bicycle, the user will be able to perform commutes on or off campus further without giving more required energy. Ultimately, this will encourage exercise by having the ability to go further and longer than with a traditional bicycle. Along with being able to combat the Freshman 15, collegiate students have a budget, and due to them being students the budget is astronomically high. Our product being at relative lower prices than other electric bicycles makes our product a more ideal fit to students within the college discourse community. These points of analysis make our product have a higher revenue attractiveness. Due to colleges with students ranging in the thousands to the ten thousand, our product has large variety of consumers. More students can equate to more revenue intake, which will help lower the price of our product so more and more students can afford the electric-assisted bicycle

To make this project successful, we laid out a set of features our electric-assisted bicycle would have. Our project consisted of six main features: the electric motor will provide 100% of its drive assistance capacity when the user's heart rate is within 15% of their maximum heart rate. The electric motor will provide 0% of its drive assistance capacity when the user is exercising below 60% of their maximum heart rate. The heart rate monitor for exercise level determination monitor's the user's heart rate and establishes a baseline heart rate. Sets the resting heart rate as the recorded value within 30 seconds of initial contact. Acquires the maximum heart rate as the highest recorded beats per minute value during the user's exercise period. Max heart rate from previous session is used for the start of new riding sessions. Initial max heart rate is asked to be set by the user. Provides 15 second averaged heart rate measurements as exercise continues. The ambient temperature sensor, LM35, measures 0-120 Degrees Fahrenheit environmental temperature with 1-degree resolution. The memory log of riding activity and heart rate will record acquired data with a timecode of the hour, minute, and second. For multiple uses within the same day, the log file will specify rides (e.g. ride 1, ride 2). This data will be written once every 15 seconds and output into a file, when the file reaches 20 MB a new file will be opened. The user feedback shows the sensor and calculation data on a display, easily readable for users with average vision, within 2 seconds of acquisition to inform user of riding progress and health levels. This is all done through the Node Red application on Raspberry Pi using the Raspberry Pi screen to display the application. And finally, for controlling the effort of the user's physical response to exercise: the system will keep the effort of the exercise in the target rate set by the user by delivering energy to the motor until target speed is reached in increments, or decrements, of 10%. Effort is determined by the percentage in maximum heart rate, over a 15 second interval, the user will be assisted in the amount of assistance they require in increments of 10% as stated above.

There are a number of different electronic microcontrollers and sensors we used for our project that we will dive more into depth later on in the paper as well. These included the microcontrollers and sensors this project that are implemented as the LM35 temperature sensor and the Pulse Sensor which is our heart rate sensor. As for microcontrollers, we used the Arduino and the Raspberry Pi 3. The use of many sensors that provide input data for temperature and heart rate/pulse. These sensors will be connected to an Arduino microcontroller where the results will be obtained and then sent serially to our Raspberry Pi 3, which will have the output data displayed on our Node Red Raspberry Pi application. The characteristics of these sensors will be coded in C language on an Arduino IDE to obtain the output data. The results obtained from these sensors will then be shared with a Raspberry Pi 3 and the results can then be displayed on a Node Red application through a Raspberry Pi screen display that the user can see live feedback for their vitals. The Raspberry Pi is a mini computer that can be used for many different technical projects such as Internet of things projects.

Our project was not funded by any private parties, just by the team members in this group. It is a little tough when we do not have the funding, we would need to make an astute electric bicycle, especially with all of us being full time students that are not working career type of jobs. However, we did what we could with the allocated resources we had. We had our electric bicycle hub motor with the controller, bicycle, bike rear rack, bike chain tool kit, pulse sensor, Shimano 6 speed sprocket, DROK Buck-boost Converter, Raspberry Pi Screen, Raspberry Pi Screen Mount parts, KNACKRO Boost converter, wires, voltage booster, and three 12V batteries. The rest of the parts and tools we had, whether it was mechanical, hardware, or software, we supplied from our own selection. The project total came out to around $600 for the pieces that we had to purchase. This total is not counting what we put into this electric-assisted bicycle project from our own personal collection.

We had some important project milestones and major events that we had to get checked off our class list and our personal list to make sure the project keeps on flowing and going on time to meet the final deadline in the spring semester. These milestones and events included: societal problem choice, design idea, pulse sensor output to pulse width modulation(pwm) to hub motor, finalizing our technical review for the end of the fall semester, fixing any mistakes we had from the fall semester and rewriting them into our contract for the spring semester, ethics quiz which tested our ethical behavior, and final integration for the deployable prototype. As for the deployable prototype, our project was left off in the fall semester as a collection of separate parts which operated normally individually. Which what we needed for the final deadline of the fall semester. Throughout the spring semester we had to work to integrate all of these components to work with each other, which was a bit harder than we had anticipated. Whether it was from different wiring situations or different setups worked better with other setups. Fast forwarding to the spring semester, the other major portion of work this semester was in putting together the hardware which made up our final prototype design. In the fall semester we had been working with just the bicycle wheel with the hub motor attached to it. In the spring semester we purchased a bicycle and installed the hub motor along with the wheel on the bicycle. Next, we worked on attaching the motor controller to the bike’s frame and grouping together the wires. The Raspberry pi board to be used for a major portion of our project was mounted to the center of the handlebars with the Raspberry Pi screen placed on top of the board. The mount was made using a rectangular block of wood, a pipe strap, a couple of wood screws, and velcro strips. The velcro strips were glued to the bottom of the Raspberry Pi’s case and then used to stick the velcro to the wood block. The pipe strap was placed around the handlebar and screwed into the wood block. This was integral for our display on the user feedback feature we had in our punch list. With this setup our Raspberry Pi and the screen attached to it were held to the handlebar and it was very stable. For the wiring to hold better during operation of the bicycle we had to solder the connections and then wrap heat shrink around them. This prevented our wiring and connections from working improperly or becoming disconnected. On the software side of our project we had to clean up the structuring of our code along with new additions to it. With cleaning up our code we had to create sections based on each block of code such as for the speedometer, the PID function, and the main loop to be run for the program. In our code for the data logger our work was cut out for us due to the complexity of having to find ways to acquire our data over a Wi-Fi connection from our Raspberry Pi board.

In the design documentation and design overview, we talked about the exercise regulation and how important for people who have not been doing a lot of exercise will need the assistance of pacing themselves to achieve a longer duration of exercise. This is what we decided our electric-assisted bicycle to do. We wanted to solve the societal problem that was required to choose, while designing the electric-assisted bicycle. That was more so challenging in the way that each individual had their own set of skills that we could use for the project. This project is educational, we wanted to maximize the application of the skills we have acquired in our educational careers. The components we ultimately decided to use were: A brushless DC motor, an Arduino, a Raspberry pi, a pulse oximeter, a display, and a gear shifting bicycle. We decided to use the brushless DC motor because they are efficient, they do not cause the bike to drag when the battery dies, and speed control is built into the driving function. We decided to use an Arduino because it has 6 ADC channels with 10-bit resolution. Even though the resolution is a bit overkill for what we were doing, we needed at least 2 ADC as well as a microcontroller to do computations, the Arduino turned out to be a good choice for us because we were doing rapid prototyping. We used a Raspberry Pi for the display and the data logging, we chose the Raspberry Pi 3 because it is open source, has capabilities like Wi-Fi connectivity and serial line communication, has easy to use display available, and it’s functions could be replaced with other devices if the project demanded so. The pulse oximeter was chosen because it would be easy to use, give an accurate pulse, and be relatively easy to implement. The Arduino performed as expected, however it needed a little bit of extra circuitry to meet the needs of our project.

When we purchased a brushless DC hub motor, it came with a controller that would drive it based on a DC signal given from a thumb throttle potentiometer. The motor itself turned out to be an excellent choice and we were able to use the built-in hall sensors to measure the speed of the wheel, giving us a speedometer. The Arduino turned out to be a great choice for the speedometer implementation because it has interrupts that allow us to keep a constant calculation on the speed. When we built the actual control function, we used the controller that came with the wheel and programmed a PID function based on the speedometer function, that we built, and a DC output signal. The Arduino has a PWM output ability and we built an extra circuit to get the average DC output of the PWM and pass it through a buffer based on a one-to-one amplifier using an OP-Amp. The Pulse oximeter sensor feeds into one of the ADCs on the Arduino, while this should have been one of the simpler aspects of our project, it turned out to be one of the more troublesome parts. When we hooked up the sensor in the spring, we could only get accurate readings from it once in a while. We recognized that the sensor had some deficiencies and started trying to replace it with smart watches or other sensors but those came with a headache of their own. What we found was that the pulse oximeter sensor itself had issues with noise introduced by holding it. When we used a Velcro strap, most of the issues went away and it operates reasonably now, provided it is mounted correctly. The bike frame turned out to be alright, it looks nice and that means something, but it has a shock in it that gets in the way of mounting the electronics. There were some things that we would do differently if we could go back in time. Of course, if we had our choice in it, we would prefer to have a better budget for more expensive pieces of equipment, as we found out that the cheaper the equipment, the harder it is to implement that device or tool into our project. We do have a user's manual for using our project: The Electric-Assisted Bicycle. There are four initial steps that we will cover later on in this paper. From there, we have riding the actual bicycle, whether the user feedback comes into play and shows what and where we should be focusing our time on and what to slow down on. We have different types of parts on our electric assisted bicycle as well. These types could be broken down into three different sections: hardware, software, and mechanical. For the hardware section: we have the boards that we used: Raspberry Pi 3 and Arduino, sensors that we used: ambient temperature LM35 sensor and the Pulse Sensor for the heart rate, different kinds of wires and tools to attach the bicycle together, the hub motor with the controller, and the batteries used to control the entire bicycle. For the software component of the electric-assisted bicycle, we used the Arduino IDE, Python, Node Red, logging files, and serially reading the data. For the final third section, the mechanical portion, we have the physical bicycle itself with its attachments of the Raspberry Pi Screen and different clips/nuts/bolts we used to implement our devices and equipment onto the electric-assisted bicycle. There will be more so explained in the later sections of the paper. We did not have any Vendor Contacts, therefore there will be nothing explained for that section of our paper.

II. SOCIETAL PROBLEM

There is a common phrase among schools nationwide called the Freshman 15. This phenomenon is based off of students gaining around 15 pounds during their freshman year of college. Sometimes this is more than 15 pounds and sometimes it is less than 15 pounds. There are several root causes for this Freshman 15: eating late at night, eating unhealthy cafeteria food, keeping unhealthy snacks and food on hand in the dorm room, drinking excessive amounts of alcohol, energy drinks, cheap food coupons and offers, fast food delivery to dorm rooms, skipping meals, lack of exercise, poor nutritional skills and education, poor sleep habits and sleep deprivation, and not understanding what their bodies need nutritionally to be healthy.

There are many key factors that play into obesity. Diet, gender, culture, genetics, lifestyle, economy are several to name. These key factors also have sub categories which further help examine where the issue may have started and arrived. For example, the culture aspect of it also relies on one's ethnicity and religion. Digging further into diet and genetics, it is known that high density foods are a cause for obesity. These high-density foods are ones that are high in fat and sugars. They are heavily destructive to the body and serve to be completely unhealthy. There is a way to have a balance, however it is a path that is not really followed in our modern lifestyles. The "Western Diet" is found to be very high in unhealthy sugars and fats. The diet portion of obesity is controllable. However, the genetic aspect of obesity is not controllable. Just because something is controllable or not, does not mean you can help enhance that aspect and still make a change for your own body. It all starts with having a want to change. Age and lifestyle are key factors, as well. Obesity is more prevalent after college, varying from the ages thirty to forty. With a lifestyle surrounded by obesity, there is an increased risk of diseases, whether they are cardiovascular, or hyperextension based. It is known that you would have to burn 1,000 to 2,000 extra calories a week if an obese lifestyle is the one you have. It is possible to reduce this by exercising for ten to twenty minutes a day, which will help reduce risks of diseases for all ages. An example of this is riding a bike to work instead of driving if you work close by to your house. Obesity is different for everyone; males and females have different obesity aspects. Maternal obesity is a very real deal. Women who are considered obese prior to pregnancy are at risk of the following diseases; gestational diabetes, perinatal morbidity, mortality, preeclampsia, cesarean section, and thromboembolism. These are just some of the many issues that arise with obesity. Taking preventative steps to help reduce and completely cure one of obesity are essential in the world we live in today. We are slowly, but surely, stepping away from the right way of living and need to take a stand in getting back to the way that helped us live on for so long. Obesity starts from within, but it can also end from within.

III. DESIGN IDEA

Our design project is an electric-assisted bike which will make use of multiple sensors and a pedal system to provide obese users the ability to effectively work out their bodies. The product will be usable by other people as well and will take into consideration the necessities of the average bike rider. With the use of our bike we expect to bring to the user a bike riding experience which can help to understand more about their physical health and how they may improve their exercising methods. The Raspberry Pi application will provide more ease in seeing how they can change their diet or what they may be doing wrong. The app will provide the user with such information by taking in data from the sensors of the project and the pedal system of the bike, such as the miles per hour of the electric-assisted bicycle. Our team has a design idea that will help support the user in reducing the risk of obesity. The design consists of an electric-assisted bicycle that monitors the user's vitals through the different sensors used and provides health information through an application. The idea came to us when we realized that everyone can go out and ride a bicycle, however, most people that are overweight or obese have trouble being able to ride the bicycle for extended periods of time. These periods of time are where there can be realistic results in reducing obesity and increasing the health of an individual. The concept behind our electric-assisted bike is to give a push to those who are winded and cannot possibly go any longer. For example, instead of being able to ride five miles on a bike, our goal is to make the user be able to ride ten miles with the electric-assist on our bicycle. Now imagine riding the bike to work or out to a public outing, there may be concerns that you will end up profusely sweating as you arrive to your destination. Our team has focused on combating this issue with monitoring the user's vitals through the application on a Raspberry Pi of the electric assisted bike. The ability to read heart rate, see the ambient temperature listed for the riding environment, and how much energy is expended in miles per hour before there is sweating which can become intolerable is all available within our application of the e-bike. Having the capability to reduce the risk of obesity and all the while remaining comfortable is exactly what our electric-assisted bicycle will provide to the user. Many components of our design and project are not all that different from what other people have done. An electric bike can already be purchased and used to commute with varying levels of intensity, many of them even have pedal battery charging and the ability to switch between electric and human drive. Sensors for vital signs already exist in smart watches, wearable pulse monitors, pedometers, and other fitness gadgets. What makes our project different is that we are going to integrate these components together in a way that allows the user to have the electric bike drive respond to fitness sensors and adjust the bike response accordingly and flexibly to meet the broad needs of an individual in present day America.

The key components in the our electric-assisted bicycle are as follows: battery, motor, chassis/wheel set, pedal electric-assist system, pressure grip throttle, switch on panel of throttle, battery - motor - throttle cable system, readable display on handlebar panel. To complement our electric-assisted bicycle, there are several key components in our Raspberry Pi Node Red application, which include; the ability to read vitals such as a pulse through heartbeat sensor, ambient temperature sensor, and the ability to see a definitive biking time and miles ridden per hour which will be displayed all through the Raspberry Pi display placed onto the electric-assisted bicycle. Our Raspberry Pi display will be run through our Raspberry Pi. In the future, we plan to use the information gathered with our sensors to output to a mobile app, however that is time lets us, as we only have a gathered amount of time for Senior Design. Our Node Red Raspberry Pi application seems to be working wonderfully for now, as we are pleased with the results and readability of our display with the frames and labels created in Node Red. In considering our design idea we are looking to build a product that meets the demands of the problem statement previously outlined. For our design idea to properly arrive at the expected outcome of helping users who are obese we require a few resources. Some components we may require are a motor assist system for the bike which would include a controller, and a power supply to charge the battery when cycling does not keep up which can be equated to a battery. These are just components for the electric assist portion of our design which would be on the bike. The components we require for the second part of our design consists of sensors for the user's vital readings. The sensors will give meaningful feedback to us about the user's bodily condition. These sensors will be the ambient temperature sensor and pulse/heart rate sensor. We will also need a microcontroller to process the data such as the Raspberry Pi. There is also a plan to use a display to give feedback to the user in the form of easy to read metrics, but we could instead just create an application for the user's phone later in the design stage. For now, Node Red's Raspberry Pi application works just fine.

For our electrically assisted bike we will be using a combination of multiple features from both hardware and software. The six main feature sets of our project will be the Electric Drive Assistance for exercise effectiveness, the heart rate monitor for exercise level determination, monitoring the environmental/ambient temperature, the memory log of riding activity (rpm, time spent riding and estimated distance traveled in miles) and heart rate, the user feedback through a screen attached to the handlebars, and controlling the effort of the user's physical response to exercise. First, the parts of the bike we will be working on will be the sensors, pedal assistance, the motor delivering enough power, a display holder for our application, and placement of each individual item. The bike will be our main focus since it is the larger part of our project and will be the mode of transportation which will assist the user in exercise. With our battery we intend to be able to power the motor to provide enough of an assist for the pedaling of the user. The screen/phone holder will be necessary in order to provide a comfortable accessibility to the user's phone. For the purposes of using our app, which we integrated into this project, this will be a beneficial feature. Next, the sensors will take readings on the ambient temperature and heart rate of the user when riding before, during, and after. The ambient temperature sensor will provide the temperature of the environment around the user while biking. The mobile app which we implemented into our project will be the third and last major feature set of our design. It will take the data provided through the sensors and will have to be able to provide diet and exercise information for the user based on their performance. Adding onto this, the electric motor will provide 100% of its drive assistance capacity when the user's heart rate is within 15% of their maximum heart rate. The electric motor will provide 0% of its drive assistance capacity when the user is exercising below 60% of their maximum heart rate. The heart rate monitor for exercise level determination monitor's the user's heart rate and establishes a baseline heart rate. Sets the resting heart rate as the recorded value within 30 seconds of initial contact. Acquires the maximum heart rate as the highest recorded beats per minute value during the user's exercise period. Max heart rate from previous session is used for the start of new riding sessions. Initial max heart rate is asked to be set by the user. Provides 15 second averaged heart rate measurements as exercise continues. The ambient temperature sensor, LM35, measures 0-120 Degrees Fahrenheit environmental temperature with 1-degree resolution. The memory log of riding activity and heart rate will record acquired data with a timecode of the hour, minute, and second. For multiple uses within the same day, the log file will specify rides (e.g. ride 1, ride 2). This data will be written once every 15 seconds and output into a file, when the file reaches 20 MB a new file will be opened. The user feedback shows the sensor and calculation data on a display, easily readable for users with average vision, within 2 seconds of acquisition to inform user of riding progress and health levels. This is all done through the Node Red application on Raspberry Pi using the Raspberry Pi screen to display the application. And finally, for controlling the effort of the user's physical response to exercise: the system will keep the effort of the exercise in the target rate set by the user by delivering energy to the motor until target speed is reached in increments, or decrements, of 10%. Effort is determined by the percentage in maximum heart rate, over a 15 second interval, the user will be assisted in the amount of assistance they require in increments of 10% as stated above.

The microcontrollers and sensors this project implemented are the LM35 temperature sensor and the Pulse Sensor which is our heart rate sensor.

As for microcontrollers, we used the Arduino and the Raspberry Pi 3. The use of many sensors that provide input data for temperature and heart rate/pulse. These sensors will be connected to an Arduino microcontroller where the results will be obtained and then sent serially to our Raspberry Pi 3, which will have the output data displayed on our Node Red Raspberry Pi application. The characteristics of these sensors will be coded in C language on an Arduino IDE to obtain the output data. The results obtained from these sensors will then be shared with a Raspberry Pi 3 and the results can then be displayed on a Node Red application through a Raspberry Pi screen display that the user can see live feedback for their vitals. The Raspberry Pi is a mini computer that can be used for many different technical projects such as Internet of things projects. Raspberry pi is an advanced RISC Machine (ARM) that runs on a GNU/Linux operating system called Debian. It has a 64-bit quad-core processor that runs at 1.4 GHz, 5 GHz wireless LAN, 1 GB LPDDR2 SDRAM, 4.2 Bluetooth, CSI camera port, DSI display port for touch screen display, and a Micro SD port for storing data. It operates at a voltage of 5V. It also contains 4 USB ports, 40 General-Purpose Input Output (GPIO) pins, an Ethernet port, HDMI port, 3.5mm audio jack, and a VideoCore IV 3D graphics core. As for the communication of the Raspberry Pi 3: Raspberry pi can be configured to use I2C communication with the chipKit MAX32. Similar to the MAX32 microcontroller, SPI and UART are other forms of communication using GPIO pins. TCP/UDP is also another form of communication except it uses network protocols to control sensors connected to GPIO pins over the internet. The data from those sensors can be sent wirelessly to a computer. Another way to get data information from raspberry pi is to create a web server on the raspberry pi to send data to a website. Network communication will be essential for this project because of the data needed to be shared between the Raspberry Pi and application purposes to display feedback from the sensors to the user.

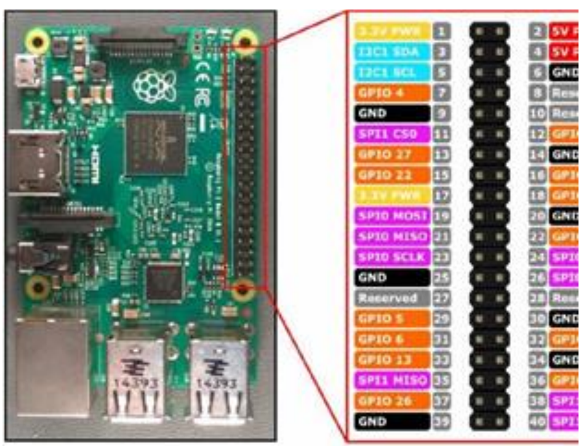

Fig. 1. Raspberry pi 3 pinout diagram for I/O and I2C [2]

As for the Pulse Sensor, we used the DFRobot Gravity pulse sensor. Reviewed many pulse sensors and the price ranges between $12 to $25. The cheaper pulse sensors didn't have great reviews because they didn't tend to be sensitive and didn't give results based off of changes in heart rate. The DFRobot pulse sensor is the best option, even though it is priced at $21.00 it is the most reliable. This sensor is made by PhotoPlethysmoGraphy (PPG) techniques and senses by detecting changes in blood volume in the microvascular bed of tissues. The signal output is interchangeable between digital square wave mode and analog pulse mode by a switch on the sensor. It works by contact to the skin. To use and operate this sensor, it will need the connections as shown in figure 5. A ground, VCC, and a signal connection. The voltage needed to operate this sensor is 5V.

The next feature we have for our project is monitoring the environmental temperature from our temperature sensor, the LM35. Our temp sensor will measure from 0-120 Degrees Fahrenheit of the environmental temperature with 1- degree resolution. Environmental temperature will play a role in monitoring the user's health levels throughout each exercise-inflicted ride. Temperature plays a role due to the fact that if it is too hot on a specific day, we will inform the user of this risk of riding in said temperature. A very hot day may have negative impacts onto the user, which is against what we are planning to utilize our device for. The way we will be monitoring the environmental temperature is that we will be measuring a targeted set range. This range of ours measures of 0-120 degrees Fahrenheit. There will be intervals of 1-degree resolutions from our environmental temperature sensor. We will be testing the temperature with real life applications, such as having our sensor in extreme cold and hot climates.

We will have self-made climates: a cold climate by using compressed air turned upside down, which has the effects of a simulated cold environment surrounding the temperature sensor. We will simulate a hot climate as well, this is done by using a hot air gun to make the surrounding temperature for the sensor an extreme of hot. These self-made climates will indicate if our sensor is capable of reading our set target range of temperature in degrees. In our eyes this will reduce the negative impacts that temperature can have on the user's riding experience.

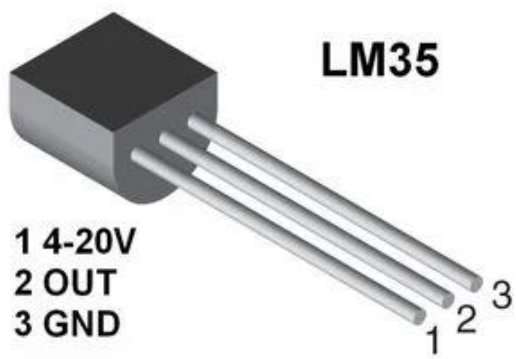

Fig. 2. LM 35 Temperature Sensor [3]

IV. FUNDING

This project's costs were completely covered by the team members. After researching the necessary devices and components to purchase, one of the team members would purchase the item and the cost would be split amongst the four team members evenly. Shown in the project cost list table below, the name and price of each item bought is listed.

Table I:

Project Cost List [4]

Murtisol Electric Bicycle Hub Motor w/ Controller	$164.69
Bicycle	$105
Bike Rear rack	$26.50
Bike Chain Tool Kit	$10.80
Pulse sensor	$24.89
Shimano 6 speed sprocket	$17.56
DROK Buck-boost Converter	$11.39
Raspberry Pi Screen	$24.98
Raspberry Pi Screen Mount Parts	$10.90
KNACKRO Boost converter	$26.50
Wires	$9.31
Batteries (3)	$132.45
Total Project Cost	$564.97
Cost Per Team Member	$141.24

The cost divided amongst each team member evenly was made easier using organization and planning mobile applications. Initially, the team did not have a set budget for spending, however, an approximate maximum spending limit of $200 to $250 was the maximum most team members were willing to contribute towards the project.

V. PROJECT MILESTONES

For the work done over the two semesters of this project's timeline, it consisted of multiple major milestones which marked our progress. These milestones included our determination of a societal problem, our design idea, the feature set, and system integration. Each milestone had its own obstacles which we had to overcome before being able to make further progress on our project.

Arriving at a specific societal problem we wanted to focus on for our timeline of two semesters was our first major obstacle when starting. Our team was unable to arrive at a direct answer for what our focus should be. We either focused too deep on a specific problem or were not descriptive enough in choosing the problem. We arrived at the societal problem of obesity as we were looking at the major problems most students face in college and the Freshman 15 was one of them. We determined that obesity would be the societal problem we wanted to focus on.

To create a design idea for our societal problem we thought of what the most logical solution for solving the problem of obesity would be. The immediate answer was fitness. Arriving at possible ways to provide fitness solutions was the difficult part because our options were either mobile fitness such as through biking or fitness through exercising machines and weight lifting. Biking was our choice because it would allow us to stretch our engineering abilities in determining how to make changes to a regular bicycle so rider's were able to exert their bodies more efficiently than just pedaling.

The major part of this project consisted of having a feature set we wanted our project to meet by the end of our timeline. This would be a list of features we were to aim to have by the end. We first wanted to have a heart rate monitor because it would allow us to measure the rider's heart rate and adjust the bicycle's speed based on the rider's performance. Next, we wanted to make sure the rider was able to receive feedback on their riding activity through a screen. We used a Raspberry Pi Screen to provide this feedback by displaying data of their heart rate, speed, set speed, and the temperature. The data displayed on this screen was also logged to a separate file which is accessible to the rider for further detailed looks at their progress. The temperature was also measured using a temperature sensor which provided the current environmental temperature. All of these features were properly achieved over the course of our progress on the project.

System integration was an integral and most difficult part of the work required for our project. We had to make sure first that all parts worked independently and then to integrate them into one overall properly working system. System integration consisted of constant troubleshooting in order to determine whether all components and both the Raspberry Pi and the Arduino Uno worked properly together. This was achieved through a lot of testing.

VI. WORK BREAKDOWN STRUCTURE

For the distribution of work amongst all four team members, the table below can be referenced. It shows the hours each member spent in the work necessary to complete work packages. Each team member was assigned work packages which would most easily be completed by them and help to reduce the total time put into the overall project.

Table II:

Team 8 Work Breakdown Structure [5]

Level 1	Level 2	Level 3	Work Packages	Hours	Owner
Electric drive assistance for exercise effectiveness	Motor system for providing pedal assistance	Speed/power controller	Power delivered to the motor	9	Isaac P.
			Speed of the motor	8	Isaac P.
			Pulse width modulation	9	Pranay C.
		Power system	Input to the controller	7	Isaac P.
		Filter circuit	Filter out the ripples in pwm output	11	Isaac and Pranay
		Battery Voltage regulation	Connection between all electronic devices to the battery	Spring 2019 10	Pranay C. and Isaac P.
		Mounting	Wheel Assembly	Spring 2019 5	Harjot S.
	Generator system	Voltage regulation system	Implement an IC for proper voltage regulation	Spring 2019 9	Isaac P.
		Battery Charge interface	Determine connections necessary for charging the battery	Spring 2019 8	Harjot S.
		Calculation of power	Derive equations for calculating the power generated	Spring 2019 7	Pranay C.
	Battery charging methods SPRING 2019 Task	Integration of generator to the battery charger	Arrive at methods for charging the battery using the motor as a generator when not providing electric-assistance	Spring 2019 10	Isaac P.
		Wall plug charger	Electrical plug for charging the battery from a power source	Spring 2019 6	Harjot S.
Heart rate monitor for exercise level determination	Maximum heart rate measurement	Set up programmable code for the pulse sensor on the Max32	Programming for the maximum heart rate to be determined between different riding sessions	7	Pranay C.
	Average heart rate measurement	Programming done for finding the average heart rate using C code on the Max32	Create a programming file for the code which will be used to measure, calculate, and send the heart rate data to the Raspberry Pi from the Max32	8	Veronica G. and Pranay C.
Control the effort of the user's physical response to exercise	Motor interaction with bicycle hardware	Determine the method through which incremental motor assistance can be provided	Calculate the total possible power that can be put into the motor	6	Isaac P.
			Create increments in the power the motor can output	9	Isaac P.
		Programming for converting pulse sensor ranges to pwm output	Pwm value adjustments for motor speed increments	9	Pranay C.
	Speedometer for determining speed of the wheel	Programming to measure the speed of the wheel	Rpm measurement to determine the mph speed at which the wheel is spinning	8	Isaac P. and Harjot S.
			Error correction for the expected pwm value output converting to speed and the actual value	11	Pranay C. and Veronica G.
	Programming testing for overall system of sensors, board, and motor	Test the programming for pulse and temperature sensor readings and measure alongside other forms of measurement for accuracy	Adjustments to be made based on the accuracy of the readings as compared to other measurements	17	Whole team task
		Testing the sensor readings being sent from the Max32 to the Raspberry Pi	Board communication testing using both wired and wireless communication for deployable prototype	Spring 2019 15	Veronica G.
		Testing of the accuracy for the pwm ranges and values sent for motor power	Adjust values associated with motor assistance provided with the pulse sensor reading ranges	17	Isaac P. and Pranay C. and Harjot S.
		Motor testing for correct speeds as measured on the speedometer	Calibration of the correct speed being displayed on the speedometer	7	Isaac P.
	Design finalization SPRING 2019 TASK	Putting together the deployable prototype	Setup of the wiring and circuits to the bike frame	Spring 2019 11	Harjot S.
			Attachment of the hub motor and wheel to the bike frame	Spring 2019 7	Harjot S.
			Finalizing the wireless connectivity of the prototype and use of the phone application	Spring 2019 11	Veronica G. and Pranay C.
User feedback to a screen attached to the handlebar	Interaction of screen with the Raspberry Pi	Data output from the Raspberry Pi to the screen	Calculations and measurements from the Max 32 are sent to the Raspberry Pi for proper collection of data	12	Veronica G.
	Android application for	Wireless communication	Create database using server software	Spring 2019 8	Veronica G.

	bike display(Spring 2019)	between Raspberry Pi and the app	PhpMyAdmin		
			Server side scripting to access database	Spring 2019 8	Veronica G.
		Feature implementation using Android studio in Java	Create heart rate/pulse object	Spring 2019 7	Pranay C.
			Create speed of motor object	Spring 2019 7	Pranay C.
			Create temperature object	Spring 2019 7	Harjot S.
		Interface design	Create the design we want for the app to look neat and organized and properly display data	Spring 2019 7	Pranay C.
Monitor environmental temperature	Environmental temperature	Temperature sensor reading output to the Max32	Create an organized and labeled list of the data collected for temperature sensing.	11	Harjot S.
Memory log of riding activity, heart rate, and temperature	Riding activity data gathering	Heart rate data gathering	Find a way for delivering the data gathered from the pulse sensor to the Raspberry Pi for storage in the memory log file	8	Pranay C.
		Temperature sensors data gathering	The temperature sensor data gathered from the Max32 has to be programmed to be sent to the memory log file	9	Veronica G. and Harjot S.
Lecture Class Assignments			**Problem Statement Individual and Team Report**	8	Isaac P.
				10	Pranay C.
				10	Veronica G.
				10	Harjot S.
			Design Idea Report	7	Harjot S.
				6	Isaac P.
				8	Veronica G.
				10	Pranay C.
			Work Breakdown Structure and Report	14	Pranay C.
				10	Isaac P.
				12	Veronica G.
				10	Harjot S.
			Project Timeline	15	Pranay C.
				12	Isaac P.
				13	Veronica G.
				12	Harjot S.
			Risk Assessment	14	Pranay C.
				13	Isaac P.
				13	Veronica G.
				13	Harjot S.
			FALL 2018 HOUR TOTAL:		
			Isaac Parrish	148	
			Pranay Chaudhary	160	
			Veronica Gonzalez	113	
			Harjot Sidhu	105	

The weekly written assignments for each semester were included alongside the work packages necessary for us to complete. This breakdown of the work done by each team member is an accurate representation of team member contributions. Our work breakdown structure consists of six features totaling 41 Work Packages, and an anticipated total of 683 hours for the Fall and Spring semester. The work packages and anticipated hours were split between our team of four, Isaac Parrish with 177 hours, Pranay Chaudhary with 192 hours, Veronica Gonzalez with 155 hours, and Harjot Sidhu with 159 hours. The hours reflected from our team activity reports for the hours spent for Fall and Spring semesters are a total of 1503.25 hours. From each team member, Harjot Sidhu spent a total of 439.5 hours, Pranay Chaudhary spent a total of 421 hours, Isaac Parrish spent a total of 296.75 hours, and Veronica Gonzalez spent a total of 307 hours. This is the complete total hours spent to complete the project. More hours have been spent after the last activity report has been submitted. These hours are not included. The total hours spent compared to the anticipated hours projected for both Fall and Spring semesters are not in close range. The numbers of hours spent to complete the project are twice as much as we anticipated. This shows that teams should double their work load anticipation when planning work packages and hours to complete those packages.

VII. RISK ASSESSMENT AND MITIGATION

With any project, there are bound to be risks involved in the process. Even the best of best designs may have risks floating in their plans. Our senior design project has been focused on solving the societal problem of lowering the risk of obesity. We have designed a solution to this societal problem, being an electric-assisted bicycle. Having an electric machine poses many risks itself, as maintaining electricity in a controlled effort takes knowledge and experience on the user's part. Our electric-assisted bicycle has hardware features and software features. Within the hardware features, such as a heart rate monitor, a temperature sensor, a hub motor, a generator circuit, and a battery. All of these hardware features make it very possible for there to be risks in our electric-assisted bicycle. Each of these hardware features also have a potential failure that results in risks and are dependable on one another to function as a whole electric-assisted bicycle. For our electric-assisted bicycle, we have a

Probability	5- 81- 100%	Sensor burns out					Lack of communication
	4- 61- 80%			Programming errors	Battery overload	Time management	
	3 41- 60%		Motor controller failure	Sensor output to display		Heart Rate Sensor failure	Hub Motor Malfunction
					Microcontroller failure		
	2 21- 40%	Generator circuit failure		Max32 Communication with Raspberry Pi			Burning out the hub motor
	1 0- 20%	Display screen malfunction	Temperature sensor malfunction		Low Battery Capacity		
		1-Very Low	2- Low	3- Medium	4- High	5- Very High	
	Impact						

Fig. 3. Risk Assessment [6]

hardware side of the machine and a software side of the machine. For the software side, there are not as many potential risks and failures to the hardware side. The reason for this being that the software side puts in a lot of hardware features and implements these hardware features into software use for the user of our electric-assisted bicycle. The risks involved with the software side of our project is the motor-controlled sensor feedback. Our electric-assisted bicycle is based on the fact that we will have a pedal-assist system implemented into our e-bike that helps support the user on rides. It is used when the user's heart rate is elevated due to physical exercise and fatigue to a level and range that is considered high for that specific user. Once this heart rate range is reached, the pedal assist system configures for this and sends an electric assist to help the user not exert as much energy. The risk in this is that the pedal-assist system will send too much electric-assist which can unbalance a person riding the electric-assisted bicycle. Another risk we have for software is having our display controlled via sensors and motor controlled, for similar reasons as above, not having the correct sensor reading. For our last risk, we have our two microcontrollers, a Raspberry Pi and a Max32, not communicating with one another, having the failure being a Bluetooth connection. There are risk mitigations to each of these risks listed below. Whether it is hardware or software, there are risk mitigations and ways to solve each risk.

In order for our project to provide a product that people can use we have to provide motor control based on sensor input as well as feedback to the end user in the form of an LCD, stored measurements, and data about effort level. The risks to our project are heart rate sensor failure, temperature sensor failure, motor failure, motor controller failure, generation circuit failure, battery problems, microcontroller failure, programming failure, and any others listed in this article. If any one of them fails it is a big deal, however in each case we have ways of fixing the problem quickly and effectively. For the risks associated with the hardware of our project, one of the most important things we can do is to have extra parts ordered in case one of the sensors burns out. The same cannot be said for the hub motor, although, the risk associated with that can be mitigated by properly studying the motor and troubleshooting whether or not a malfunction has occurred and can be fixed. In the case that it cannot be fixed we can order a spare motor and the impact on our project would be the time required for the motor to arrive. To mitigate risks involving our battery we will purchase a lithium battery with a higher capacity which will be properly integrated in our circuitry to reduce the possibility of overloading a battery without enough power to support our project. For the problems associated with the programming of our pulse sensor and temperature sensor, the time required to fix them can vary because of the trial and error method which might be required. Proper analysis of our programming can help to prevent this and understanding how the programming is communicating with the microcontroller boards would be very beneficial in reducing the time put into troubleshooting. One of the most important aspect of our project will be time. Proper time allotment for the activities and tasks associated with our project can help to greatly increase our work efficiency on the project as well as increase the quality of the final product. Risks involving time can be reduced when team members communicate and provide each other assistance if one of us is stuck on something. Speaking of communication, the highest risk in the project can be reduced by making sure that team members can be comfortable enough to tell each other their progress and any problems they may be dealing with regarding the project. If a problem arises from time conflicts, a broken part, or delays in getting a work package completed, communicating that you are having trouble can be the most beneficial thing to do as it will inform the team of your difficulties and allow for adjustments in the project timeline.

VIII. DESIGN PHILOSOPHY

To solve the societal problem of increasing obesity in America, we chose to build an electric bike that would not only encourage people to exercise but would help people exercise effectively by regulating their exercise. Exercise regulation is especially important for people who have not been doing a lot of exercise and need help pacing themselves to achieve a longer duration of exercise. Solving the societal problem required designing a bike that was easy to use, practical to use, and encouraging to use. In the beginning of our project we each had a set of skills that we could use and a project that didn't necessarily require all of them. Because this project is educational, we wanted to maximize the application of the skills we have acquired in our educational careers. We also wanted to stretch those skills and acquire new ones.

The components we ultimately decided to use were: A brushless DC motor, an Arduino, a Raspberry pi, a pulse oximeter, an LCD display, and a regular bicycle. We decided to use the brushless DC motor because they are efficient, they do not cause the bike to drag when the battery dies, and speed control is built into the driving function. We decided to use an Arduino because it has 6 ADC channels with 10-bit resolution. Even though the resolution is a bit overkill for what we were doing, we needed at least 2 ADC as well as a microcontroller to do computations, the Arduino turned out to be a good choice for us because we were doing rapid prototyping. We used a Raspberry Pi for the display and the data logging, we chose the Raspberry Pi 3 because it is open source, has capabilities like wifi connectivity and serial line communication, has easy to use LCD displays available, and it's functions could be replaced with other devices if the project demanded so. The pulse oximeter was chosen because it would be easy to use, give an accurate pulse, and be relatively easy to implement. Not everything turned out the way we wanted it to, and we had some hiccups along the way.

When we started using our choice of components, we found that some of the things exceeded our expectations, while others were a letdown. The Arduino performed as expected, however it needed a little bit of extra circuitry to meet the needs of our project. When we purchased a brushless DC hub motor, it came with a controller that would drive it based on a DC signal given from a thumb throttle potentiometer. The motor itself turned out to be an excellent choice and we were able to use the built-in hall sensors to measure the speed of the wheel, giving us a speedometer. The Arduino turned out to be a great choice for the speedometer implementation because it has interrupts that allow us to keep a constant calculation on the speed. When we built the actual control function, we used the controller that came with the wheel and programmed a PID function based on the speedometer function, that we built, and a DC output signal. The Arduino has a PWM output ability and we built an extra circuit to get the average DC output of the PWM and pass it through a buffer based on a one-to-one amplifier using an OP-Amp. The Pulse oximeter sensor feeds into one of the ADCs on the Arduino, while this should have been one of the simpler aspects of our project, it turned out to be on of the more troublesome parts. When we hooked up the sensor in the spring, we could only get accurate readings from it once in a while. We recognized that the sensor had some deficiencies and started trying to replace it with smart watches or other sensors but those came with a headache of their own. What we found was that the pulse oximeter sensor itself had issues with noise introduced by holding it. When we used a Velcro strap, most of the issues went away and it operates reasonably now, provided it is mounted correctly. The bike frame turned out to be alright, it looks nice and that means something, but it has a shock in it that gets in the way of mounting the electronics. There were some things that we would do differently if we could go back in time.

If we were doing it again and we had a larger budget, we would still choose the Arduino, because it has the features we need. But we would get a Bluetooth module for the Arduino and use a cell phone app for the display because that would be less expensive to produce and would allow people to integrate better with their other fitness items. We would choose a more robust heart rate monitoring system that was wireless and worn on the wrist, like a smart watch. If we had a better sensor, we would have had more time to implement features such as battery charging and increased exercise resistance for the hard-core exerciser. We would also choose a different bike frame that we could mount our electronics to a little bit easier. And we would choose a lighter weight battery system such as lithium ion or nickel metal hydride. While our design choices weren’t the most optimum looking in hindsight, some of them couldn’t be better, some could, but they were sufficient for what we are doing.

IX. DEPLOYABLE PROTOTYPE STATUS

Our project was left off in the fall semester as a collection of separate parts which operated normally individually. Throughout the spring semester we had to work to integrate all of these components to work with each other. The other major portion of work this semester was in putting together the hardware which made up our final prototype design. In the fall semester we had been working with just the bicycle wheel with the hub motor attached to it. In the spring semester we purchased a bicycle and installed the hub motor along with the wheel on the bicycle. Next, we worked on attaching the motor controller to the bike's frame and grouping together the wires. The Raspberry pi board to be used for a major portion of our project was mounted to the center of the handlebars with the Raspberry Pi screen placed on top of the board. The mount was made using a rectangular block of wood, a pipe strap, a couple of wood screws, and velcro strips. The velcro strips were glued to the bottom of the Raspberry Pi's case and then used to stick the velcro to the wood block. The pipe strap was placed around the handlebar and screwed into the wood block. With this setup our Raspberry Pi and the screen attached to it were held to the handlebar and it was very stable. For the wiring to hold better during operation of the bicycle we had to solder the connections and then wrap heat shrink around them. This prevented our wiring and connections from working improperly or becoming disconnected. On the software side of our project we had to clean up the structuring of our code along with new additions to it. With cleaning up our code we had to create sections based on each block of code such as for the speedometer, the PID function, and the main loop to be run for the program. In our code for the data logger our work was cut out for us due to the complexity of having to find ways to acquire our data over a wifi connection from our Raspberry Pi board.

X. MARKETABILITY FORECAST

A. Market Size and Competition

The electrical bike market size was 15.7 Billion U.S. dollars in 2016 and expected to reach 24.3 Billion U.S. dollars in 2025 [7]. This shows that there is a demanding market for electrical bikes worldwide. Specifically, in United States, the market sales for E-bikes in 2017 were $77.1 Billion U.S. dollars [7].

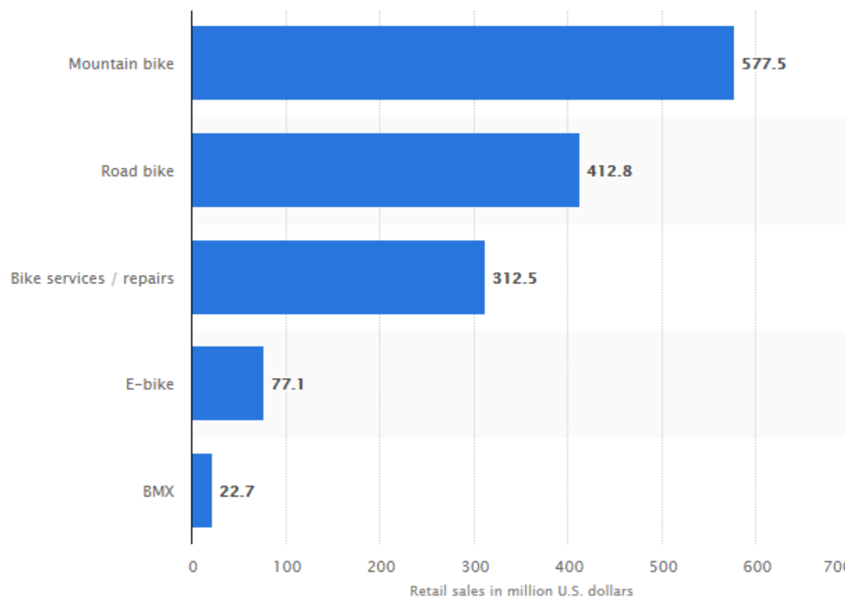

Fig. 4. Bicycle sales by category in the U.S. [7].

As show in figure 4 [7], the E-bike market is extremely smaller than the larger competitors of Mountain bikes and Road bikes. This is an issue to face for market competition because there are eight times more sales in Mountain bikes. Other competition we will face is against other E-bike companies themselves. E- bike manufacturers such as Pedego Electric Bikes, Yamaha Motors, M1 Sporttechnik. Also, Sony, Robert Bosch GmbH, and Samsung for component suppliers that supply batteries and motors [8]. Since E-bikes are new and trending, our product will be in the ferment phase of the technology cycle. Over time the market sales are expected to rise when our product enters the growth phase of the technology cycle.

What differentiates our E-bike from the rest of the E-bikes are the features provided and particularly our motor control assistance that is determined by a heart rate. Our features provide convenience to the customer which makes it a positive opportunity for our product to be successful.

B. Market Segments/ Target Customer

An electric bicycle can bring together a combination of customers from many market areas such as the bicycle market, a portion of gym equipment market, athletics market, the green transportation market, health, fitness, and electric bike markets. There are a wide range of customers that our product would appeal to. We will focus on health and fitness of our customers and from this area narrow down to the overweight and specifically, college students as our target customer. We can meet the needs of health and fitness with our product by helping people become fit while fighting the societal problem of reducing obesity.

C. Switching Costs

The estimated costs of our E-bike are $1200, based off of the $400 base cost for development and using the least profit price of 3 times our base price. Our competitors of Mountain bikes and road bikes range between $100 to $1000 depending on the quality. The average E-bike ranges between $700 to $3,500. College students would choose the lowest prices considering the low budget most students have. Considering the features and electric assist capability that other E-bikes do not have makes our price ideal for an E-bike. Considering the lowest price in Mountain and Road bikes compared to our E-bike, this is a threat to our product.

There would be a major cost difference for customers to switch from their current bike suppliers to ours if the features are not desired would be a difficulty for our customers. Given the features not provided by any other E-bike supplier, the cost would be $500 increase from a basic E-bike. The cost may be difficult if the student is on a strict budget.

D. Work Required to Achieve Marketability

The deployable prototype would need to undergo some hardware changes to achieve marketability. Excellent electric bikes have a smooth ride and have instant feedback with the user. To achieve this, the lag time between hardware devices needs to be refined. A fast microcontroller that communicates instantly with other hardware removes the lag time will be needed. The heart rate sensor is sensitive and doesn't give precise readings in heart rate. An excellent and maybe more expensive heart rate sensor that gives accurate readings will also be needed. The battery size that the deployable electric bike has sitting on the back may be too big and not safe for riders. A smaller more compact power source will be needed, specifically one that does not take up so much space and that is light in weight. Once these changes are made to the current deployable prototype then it will be ready to be manufactured to achieve marketability.

XI. CONCLUSION

To close, our team developed an electric-assisted bicycle to help reduce the risk of obesity in collegiate level students through steady and progressive exercise of riding a bicycle. There were bumps and obstacles to overcome in our project, as there is in most senior design projects. We as a team tackled these as best as we could with the knowledge we had at the time. We have found numerous studies, shown previously in the paper, that collegiate students are at a downfall type of risk when it comes to weight gain. Whether that is minimal or maximal is up to each individual student and their choices of lifestyle, however this does not take away from the fact that college students would benefit from exercise. A relatively easy way for this exercise would be using our electric-assisted bicycle which can give the user feedback on their heart rate and ambient temperature to ensure better and safer riding conditions. To reiterate what was said before in this paper, we tackled a societal problem while producing a project to help solve a bit of this problem. We had fun as a team and overall learned a great bunch in engineering, problem solving, and soft skills to carry into our professional career and lives. More so on our project, we had a feature set which comprised of sixth features. These features were the electric motor will provide 100% of its drive assistance capacity when the user's heart rate is within 15% of their maximum heart rate. The electric motor will provide 0% of its drive assistance capacity when the user is exercising below 60% of their maximum heart rate. The heart rate monitor for exercise level determination monitor's the user's heart rate and establishes a baseline heart rate. Sets the resting heart rate as the recorded value within 30 seconds of initial contact. Acquires the maximum heart rate as the highest recorded beats per minute value during the user's exercise period. Max heart rate from previous session is used for the start of new riding sessions. Initial max heart rate is asked to be set by the user. Provides 15 second averaged heart rate measurements as exercise continues. The ambient temperature sensor, LM35, measures 0-120 Degrees Fahrenheit environmental temperature with 1-degree resolution. The memory log of riding activity and heart rate will record acquired data with a timecode of the hour, minute, and second. For multiple uses within the same day, the log file will specify rides (e.g. ride 1, ride 2). This data will be written once every 15 seconds and output into a file, when the file reaches 20 MB a new file will be opened. The user feedback shows the sensor and calculation data on a display, easily readable for users with average vision, within 2 seconds of acquisition to inform user of riding progress and health levels. This is all done through the Node Red application on Raspberry Pi using the Raspberry Pi screen to display the application. And finally, for controlling the effort of the user's physical response to exercise: the system will keep the effort of the exercise in the target rate set by the user by delivering energy to the motor until target speed is reached in increments, or decrements, of 10%. Effort is determined by the percentage in maximum heart rate, over a 15 second interval, the user will be assisted in the amount of assistance they require in increments of 10% as stated above. The use of many sensors that provide input data for temperature and heart rate/pulse. These sensors will be connected to an Arduino microcontroller where the results will be obtained and then sent serially to our Raspberry Pi 3, which will have the output data displayed on our Node Red Raspberry Pi application. The characteristics of these sensors will be coded in C language on an Arduino IDE to obtain the output data. The results obtained from these sensors will then be shared with a Raspberry Pi 3 and the results can then be displayed on a Node Red application through a Raspberry Pi screen display that the user can see live feedback for their vitals. As we have said previously, our project had no funding other than the funding provided by each own team member. The project came to around $600 for the pieces that we purchased. These pieces were our electric bicycle hub motor with the controller, bicycle, bike rear rack, bike chain tool kit, pulse sensor, shimano 6 speed sprocket, DROK Buck-boost Converter, Raspberry Pi Screen, Raspberry Pi Screen Mount parts, KNACKRO Boost converter, wires, voltage booster, and three 12V batteries. We are hopeful that we can impress and win over some representatives of other companies while we display our product at the senior showcase. We believe we have put in a lot of hard-working and honest hours into our project, whether that was from writing the reports or to doing actual technical work or to doing more software related work. There was always something to work on and we allocated each spot for each team member to work on respectively. We had some important project milestones and major events that we had to get checked off our class list and our personal list to make sure the project keeps on flowing and going on time to meet the final deadline in the spring semester. These milestones and events included: societal problem choice, design idea, pulse sensor output to pwm to hub motor, finalizing our technical review for the end of the fall semester, fixing any mistakes we had from the fall semester and rewriting them into our contract for the spring semester, ethics quiz which tested our ethical behavior, and final integration for the deployable prototype. The Arduino turned out to be a good choice for us because we were doing rapid prototyping. We used a Raspberry Pi for the display and the data logging, we chose the Raspberry Pi 3 because it is open source, has capabilities like wifi connectivity and serial line communication, has easy to use display available, and it's functions could be replaced with other devices if the project demanded so. To explain what was said earlier in the paper, when we built the actual control function, we used the controller that came with the wheel and programmed a PID function based on the speedometer function, that we built, and a DC output signal. The Arduino has a PWM output ability and we built an extra circuit to get the average DC output of the PWM and pass it through a buffer. We recognized that the sensor had some deficiencies and started trying to replace it with smart watches or other sensors but those came with a headache of their own. What we found was that the pulse oximeter sensor itself had issues with noise introduced by holding it. When we used a Velcro strap, most of the issues went away and it operates reasonably now, provided it is mounted correctly. The bike frame turned out to be alright, it looks nice and that means something, but it has a shock in it that gets in the way of mounting the electronics. This goes to show that not everything was perfect with our project and its separate individual pieces. To start riding the bicycle there are four initial steps. Now from these steps, we have the actual riding that goes along with the bicycle. Now whether that is the user feedback coming play or seeing when and where to slow down and to speed up. This can be left to the user's discretion. We have different types of parts on our electric assisted bicycle as well. These types could be broken down into three different sections: hardware, software, and mechanical. For the hardware section: we have the boards that we used: Raspberry Pi 3 and Arduino, sensors that we used: ambient temperature LM35 sensor and the Pulse Sensor for the heart rate, different kinds of wires and tools to attach the bicycle together, the hub motor with the controller, and the batteries used to control the entire bicycle. For the software component of the electric-assisted bicycle, we used the Arduino IDE, Python, Node Red, logging files, and serially reading the data. More so on the software side of our project we had to clean up the structuring of our code along with new additions to it. With cleaning up our code we had to create sections based on each block of code such as for the speedometer, the PID function, and the main loop to be run for the program. For the final third section, the mechanical portion, we have the physical bicycle itself with its attachments of the Raspberry Pi Screen and different clips/nuts/bolts we used to implement our devices and equipment onto the electric-assisted bicycle. Overall, we do feel as if we did a good job on our project to implement all of our features from our punch list into the deployable prototype.

REFERENCES

[1] “Preventing the Freshman 5, 10, 15, 40,” Obesity Action Coalition. [Online]. Available: https://www.obesityaction.org/community/article-library/preventing-the-freshman-5-10-15-40/. [Accessed: 28-Apr-2019].

[2] diyprojects.io, “Activate the i2c bus of Raspberry Pi 3 or Pi Zero W • DIY Projects,” DIY Projects, 01-Jul-2017. [Online]. Available: https://diyprojects.io/activatei2c-bus-raspberry-pi-3zero/#.W7EaqGhKjIU. [Accessed: 30Sep-2018].

[3] Instructables, “LM35 Temperature Sensor,” Instructables, 15-Oct-2017. [Online]. Available: https://www.instructables.com/id/LM35-Temperature-Sensor/. [Accessed: 29-Apr-2019].

[4] Project Cost List. Created by Pranay Chaudhary. April 29, 2019.

[5] Team 8 Work Breakdown Structure. Created by Pranay Chaudhary. Fall 2018.

[6] Created by Pranay Chaudhary. Team 8. Risk Assessment Chart. 1-November-2018.

[7] “E-bikes - global market size 2025 | Statistic,” Statista. [Online]. Available: https://www.statista.com/statistics/674381/size-global-market-electric-bicycles/. [Accessed: 25-Feb-2019].

[8] “E-bike Market , Segmented by Propulsion Type, Usage, and Geography - Growth, Trends, and Forecast (2019 - 2024),” Analysis of UAE Telecom market segmented by major players, market share, revenues and regulations.[Online]. Available: https://www.mordorintelligence.com/industry-reports/e-bike-market. [Accessed: 25-Feb-2019].

[9] Electric Assist Bicycle Deployable Prototype. Team 8. Picture taken by Pranay Chaudhary. April 18, 2019.

[10] Hardware Block Diagram, Veronica Gonzalez. April 28, 2019.

[11] Arduino Uno Pinout. Y. Tawil, “Understanding Arduino UNO Hardware Design,” All About Circuits, 01-Jul-2016. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/. [Accessed: 28-Apr-2019].

[12] LM35 Temperature Sensor Schematic. T. Agarwal, T. A. T. Agarwal, Aneesh, T. Agarwal, and Edgefx Technologies Pvt Ltd, “LM35 Temperature Sensor Circuit and Its Working,” Buy Electronics & Electrical Projects in the United States, 10-Jan-2018. [Online]. Available: https://www.efxkits.us/lm35-temperature-sensor-circuit-working/. [Accessed: 28-Apr-2019].

[13] Pulse Sensor Schematic. M. #237238, M. #321089, M. #661774, OHoilett, M. #721982, M. #554862, M. #219477, GearJunkie88, M. #40211, M. #680503, and M. #1317812, “Pulse Sensor,” SEN-11574 - SparkFun Electronics. [Online]. Available: https://www.sparkfun.com/products/11574. [Accessed: 28-Apr-2019].

[14] Raspberry Pi 3 Pinout. diyprojects.io, “Activate the i2c bus of

Raspberry Pi 3 or Pi Zero W • DIY Projects,” DIY

Projects, 01-Jul-2017. [Online]. Available: https://diyprojects.io/activate-i2c-bus-raspberry-pi-3-zero/#.W7EaqGhKjIU. [Accessed: 30-Sep-2018].

[15] Raspberry Pi 3 Schematics. “Schematics,” Schematics - Raspberry Pi Documentation. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/README.md. [Accessed: 28-Apr-2019].

[16] Upper Level Software block diagram. Veronica Gonzalez. April 28, 2019

[17] Data Logger Flowchart. Veronica Gonzalez. April 28, 2019

[18] Motor Control Flowchart. Veronica Gonzalez. April 28, 2019

[19] PID Function Flowchart. Veronica Gonzalez. April 28, 2019

[20] Heart Rate Motor Control Flowchart. Veronica Gonzalez. April 28, 2019

[21] Speedometer Function Flowchart. Veronica Gonzalez. April 28, 2019

[22] Temperature Function Flowchart. Veronica Gonzalez. April 28, 2019

[23] Shimano Bicycle. Walmart. [Online] [Accessed 29-April-2019] Available: https://www.walmart.com/ip/Hyper-26-Shocker-Men-s-Dual-Suspension-Mountain-Bike-Black/54169167

[24] Hub Motor with Controller. Amazon. [Online]. [Accessed 29-April-2019] Available: https://images-na.ssl-images-amazon.com/images/I/717q3DPrvJL._SL1500_.jpg

GLOSSARY

Body Mass Index – A weight to height ratio use to determine whether a person has too much or too little body fat.

Consumer –A person who purchases goods and services for personal use

Market– An area or arena in which commercial dealings are conducted

Appendix A

Electric Assist Bicycle User's Manual

The goal is for an average person who is capable of riding a bike and using a computer to be able to achieve exercise in an automatic fashion, taking out the need for mental calculation of exercise level determination.

Fig. 5. Electric Assist Bicycle Deployable Prototype [9]

Powering the bike

This bike requires a 36V battery.

Step 1:

Connect the positive red (+) wire to the Positive terminal of the battery

Step 2:

Connect the Negative black (-) wire to the negative terminal of the battery

Powering the control module

Step 1:

Connect the control module to the display device with a USB device cable

Step 2:

Connect the mini USB cable from a battery pack to the display device

Step 3:

With the display turned on and fully booted open the “startupscript” icon, if prompted select “open in terminal”

Step 4:

Select one of the options: Low, Medium, or High.

Riding the bike

Display features below will help you exercise as you ride this electrical assist bike.

Select Menu: this allows the user to select an intensity of exercise.

BPM: This shows the heart rate of the user wearing the heart rate monitor in beats per minute.

Degrees: This will show you the temperature of the bike environment in degrees Fahrenheit.

MPH: This gauge will show how fast the bike wheel is travelling in MPH.

Set Speed: This will show you How fast the bike is set to travel

The select menu allows you to set the intensity of exercise, which means the average heart rate desired.

To set speed select the speed which you would like to travel by pedaling up to that speed and press the set button. The speed will be set, and as long heart rate is in the range of exercise, the bike will regulate to this speed. If the heart rate goes below the range of exercise, the set speed will decrement by 10% of the original speed that was set by pressing the button, every 20 second interval. If the heart rate increases to the range of exercise, the set speed will return to the original value.

Note: Proper use of the heart rate monitor is required in order to achieve exercise effectiveness.

Appendix B

Hardware

Hardware Block Diagram

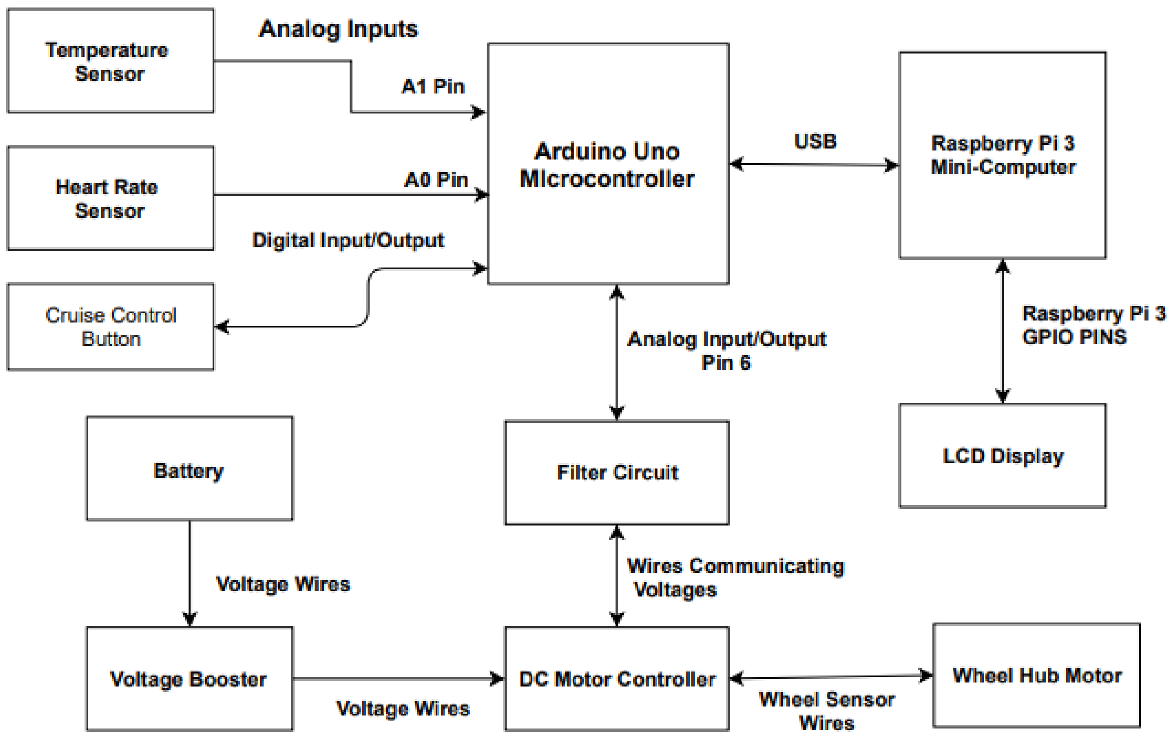

Figure B-1. Hardware Block Diagram [10]

Arduino Uno Schematic

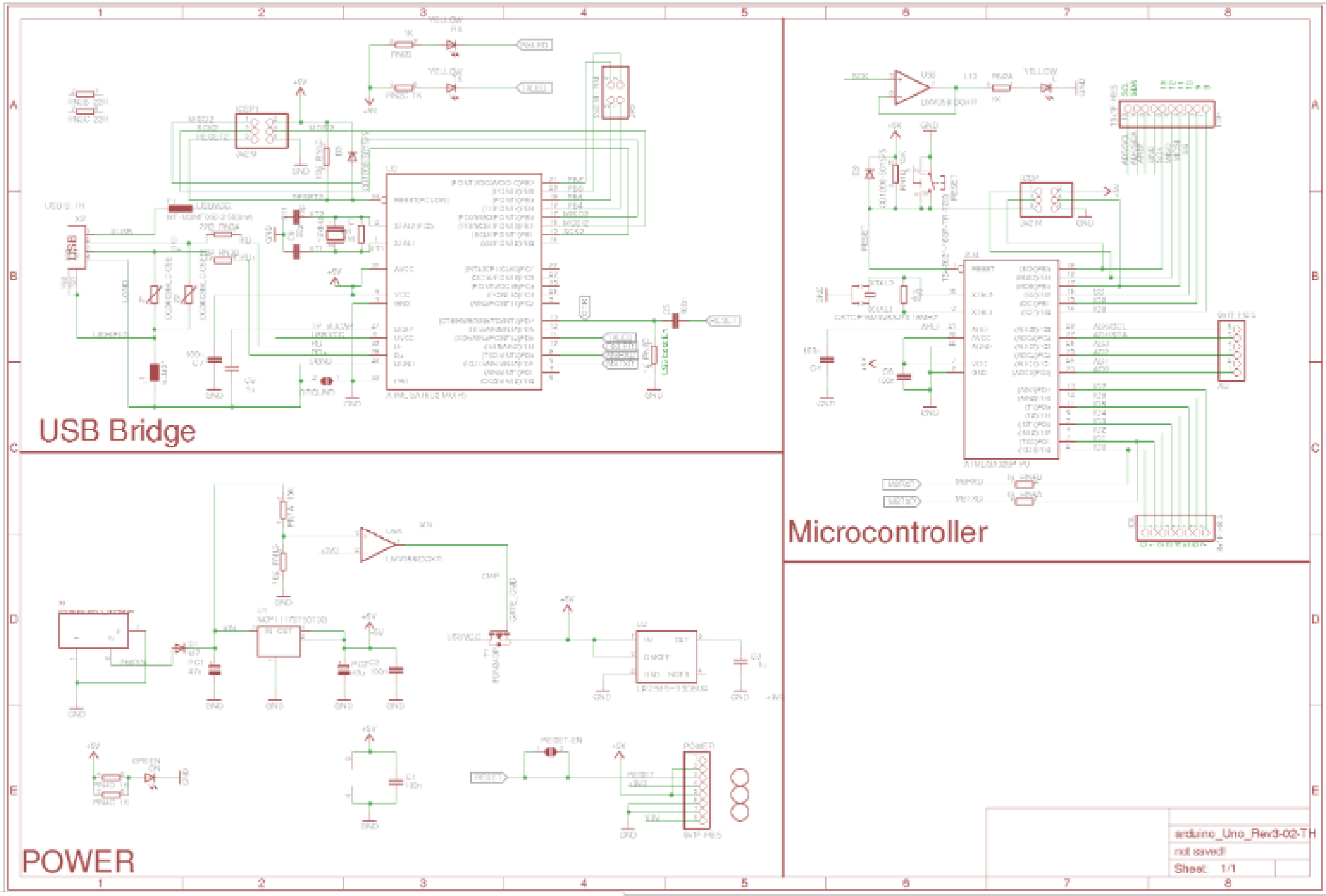

Figure B-2. Arduino Uno Schematic [11]

Arduino Uno Pinout

Figure B-3-1. Arduino Uno Pinout [11]

LM35 Temperature Sensor Schematic

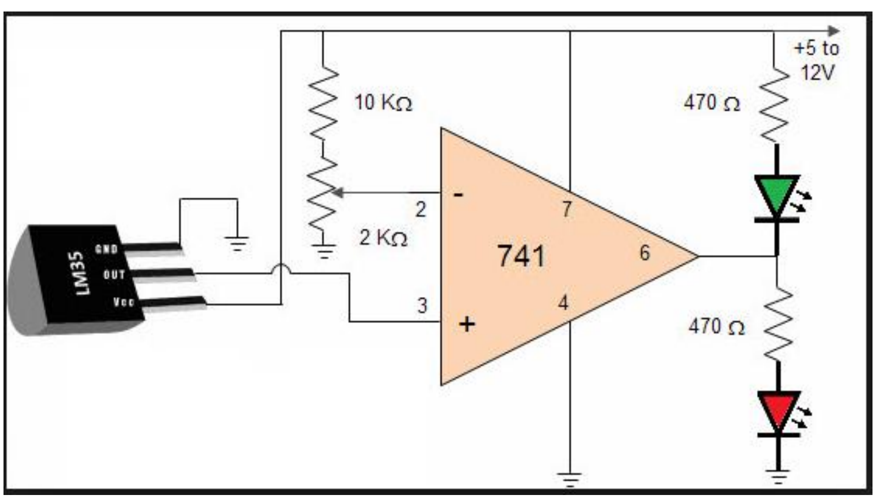

Figure B-3-2. LM35 Temperature Sensor Schematic [12]

Pulse Sensor Schematic

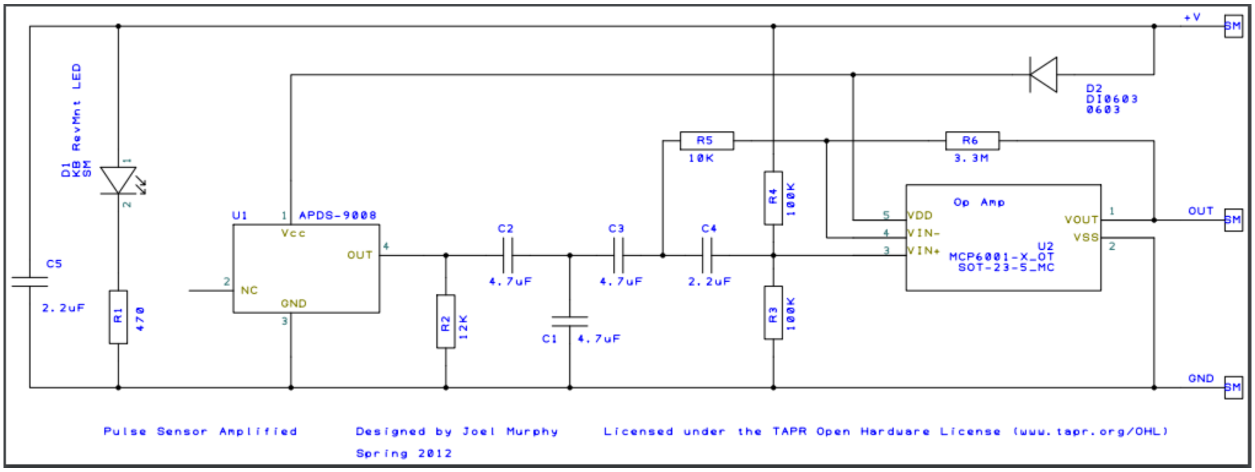

Figure B-4-1. Pulse Sensor Schematic [13]

Raspberry Pi 3 Pinout

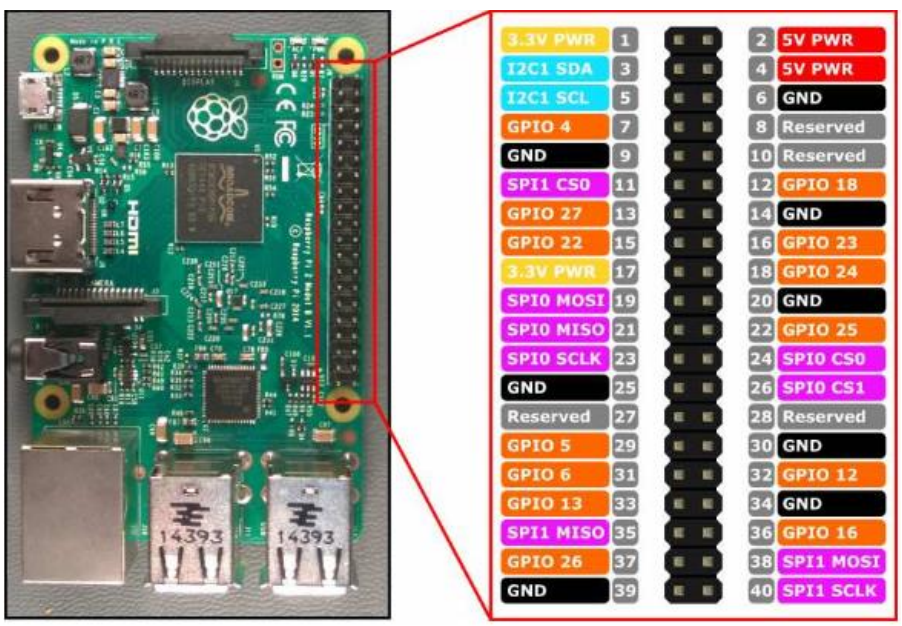

Figure B-4-2. Raspberry Pi 3 Pinout [14]

Raspberry Pi 3 Schematics

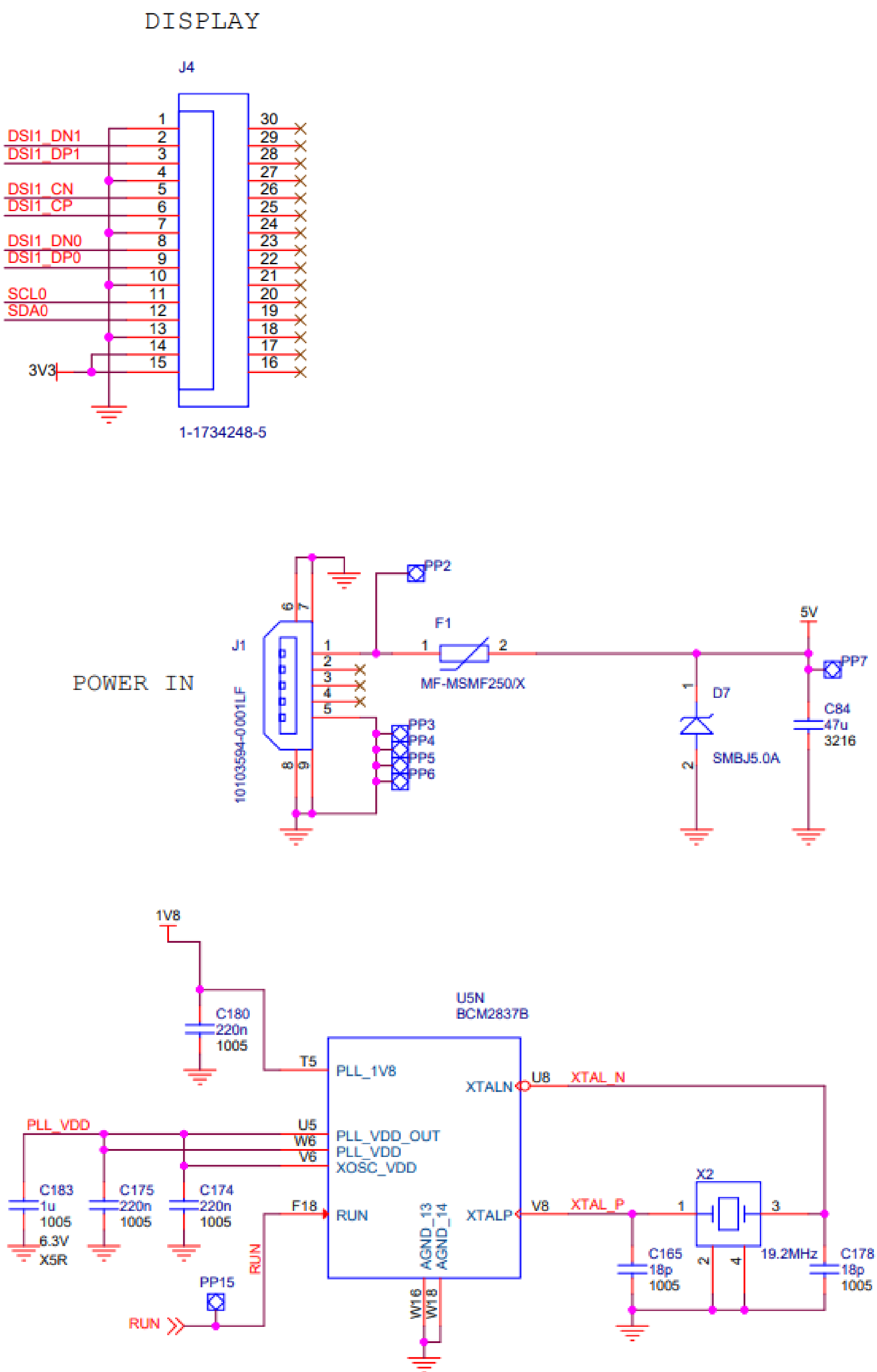

Figure B-5. Raspberry Pi 3 Schematic [15]

GPIO Pinouts Schematic

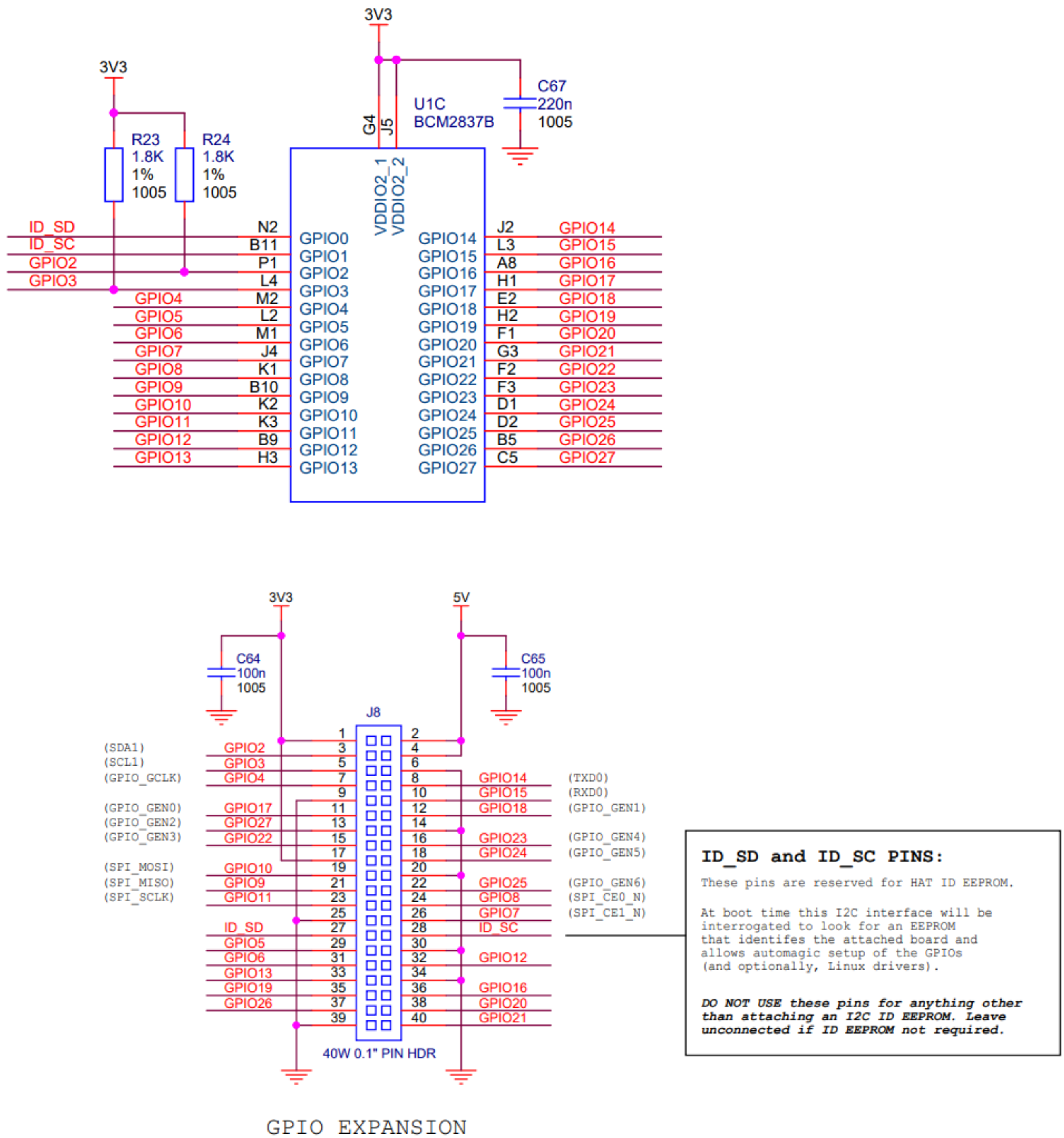

Figure B-6. Raspberry Pi 3 Schematic [15]

Appendix C

Software

Upper Level Software Block Diagram

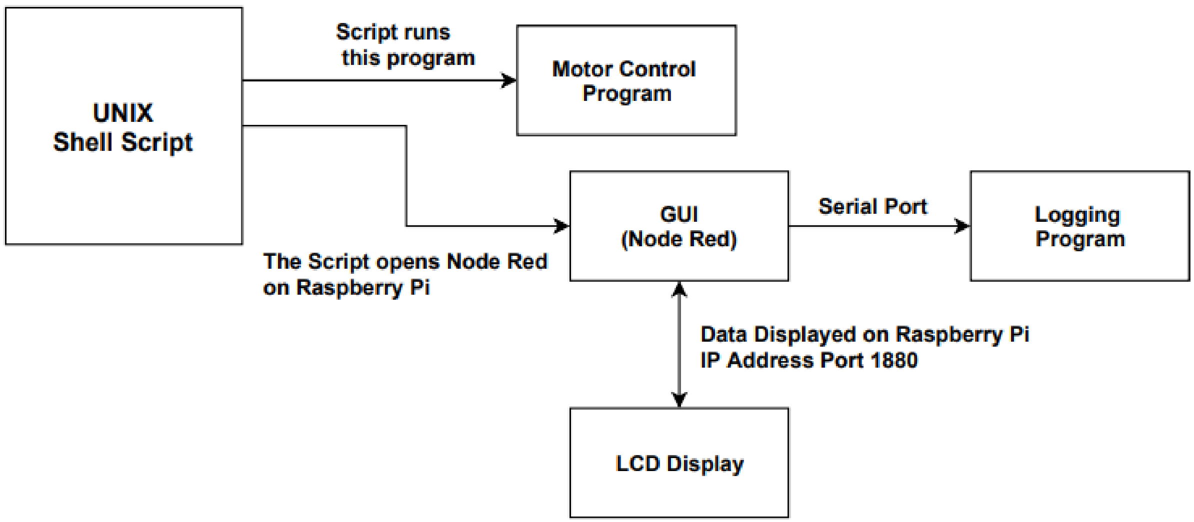

Figure C-1. Upper Level Software Block Diagram [16]

Upper Level Software Operation

On start-up of our system a Shell script is executed on the raspberry pi and runs the files we need to operate our system. The Shell script opens the commands prompt and runs the Motor Control Program and executes commands we need to open and Run our Node Red Program. The Node Red Program opens a serial port to communicate data with the Logging Program and is the same data being displayed on the LCD display. The data visual can also be accessed via laptop or cell phone by typing the IP address of the Raspberry Pi 3 and port 1880 in a browser.

Data Logger Software Flowchart

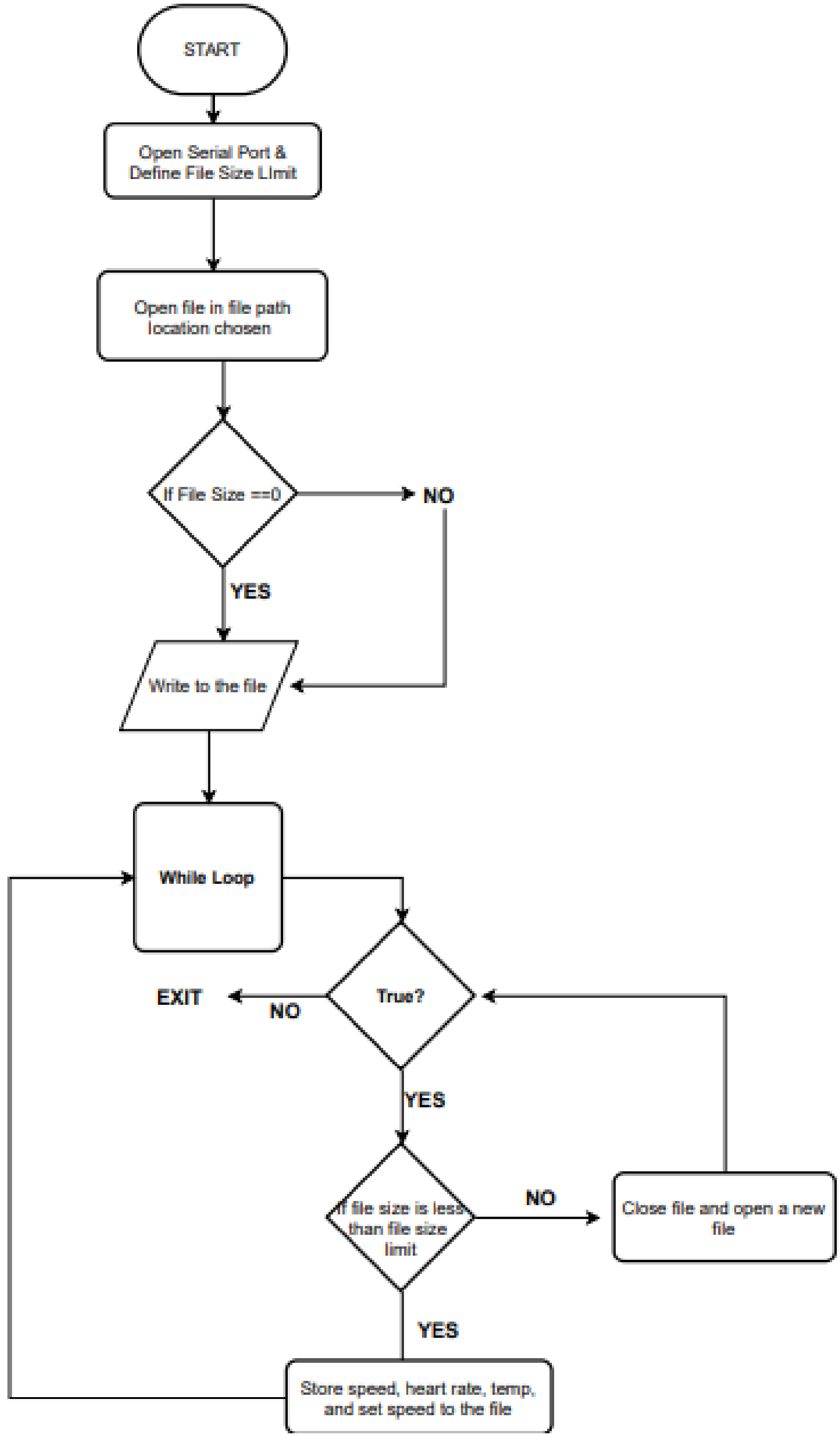

Figure C-2. Data Logger Flowchart [17]

Motor Control Software Flowchart

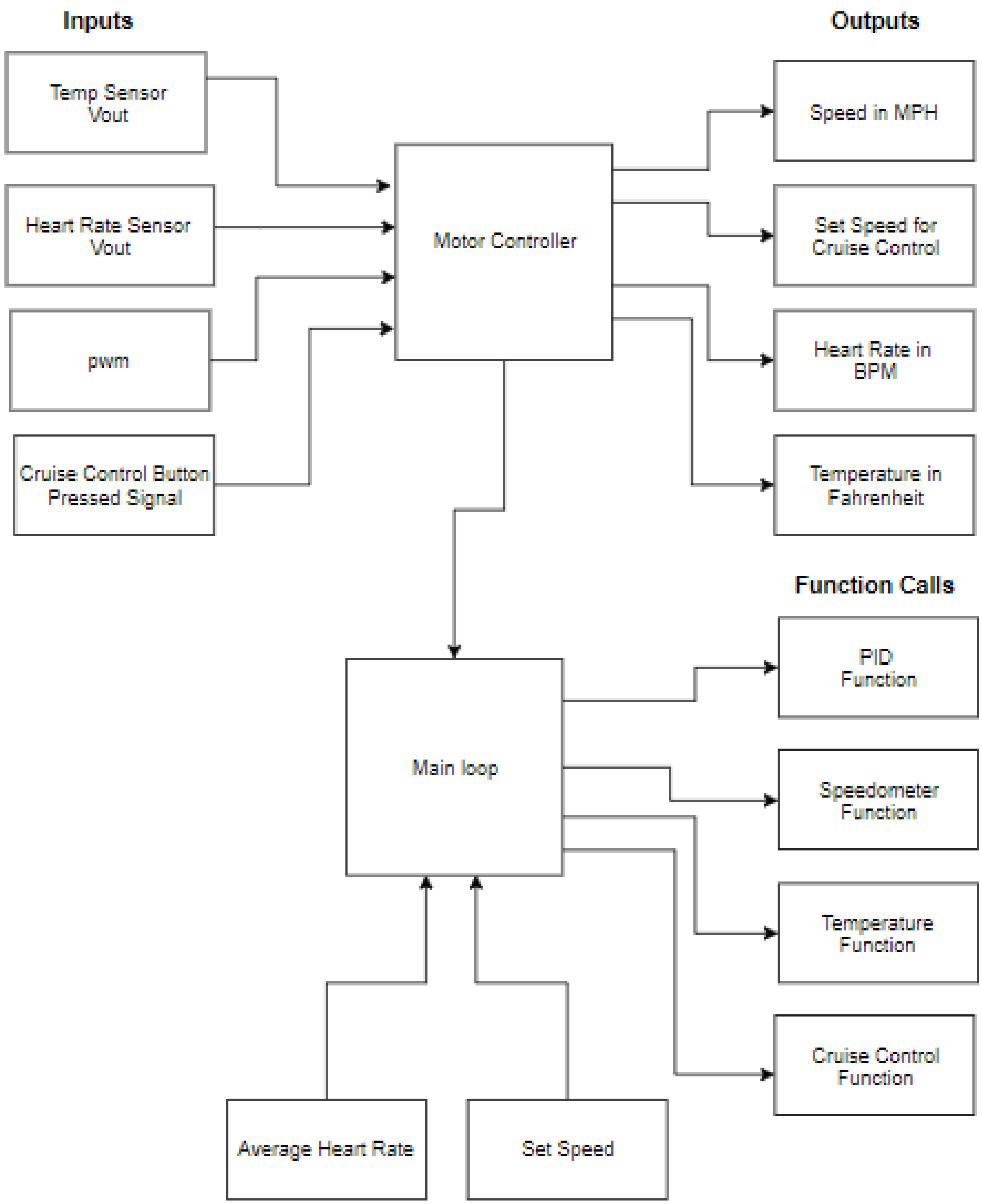

Figure C-3. Motor Control Flowchart [18]

PID Function Software Flowchart

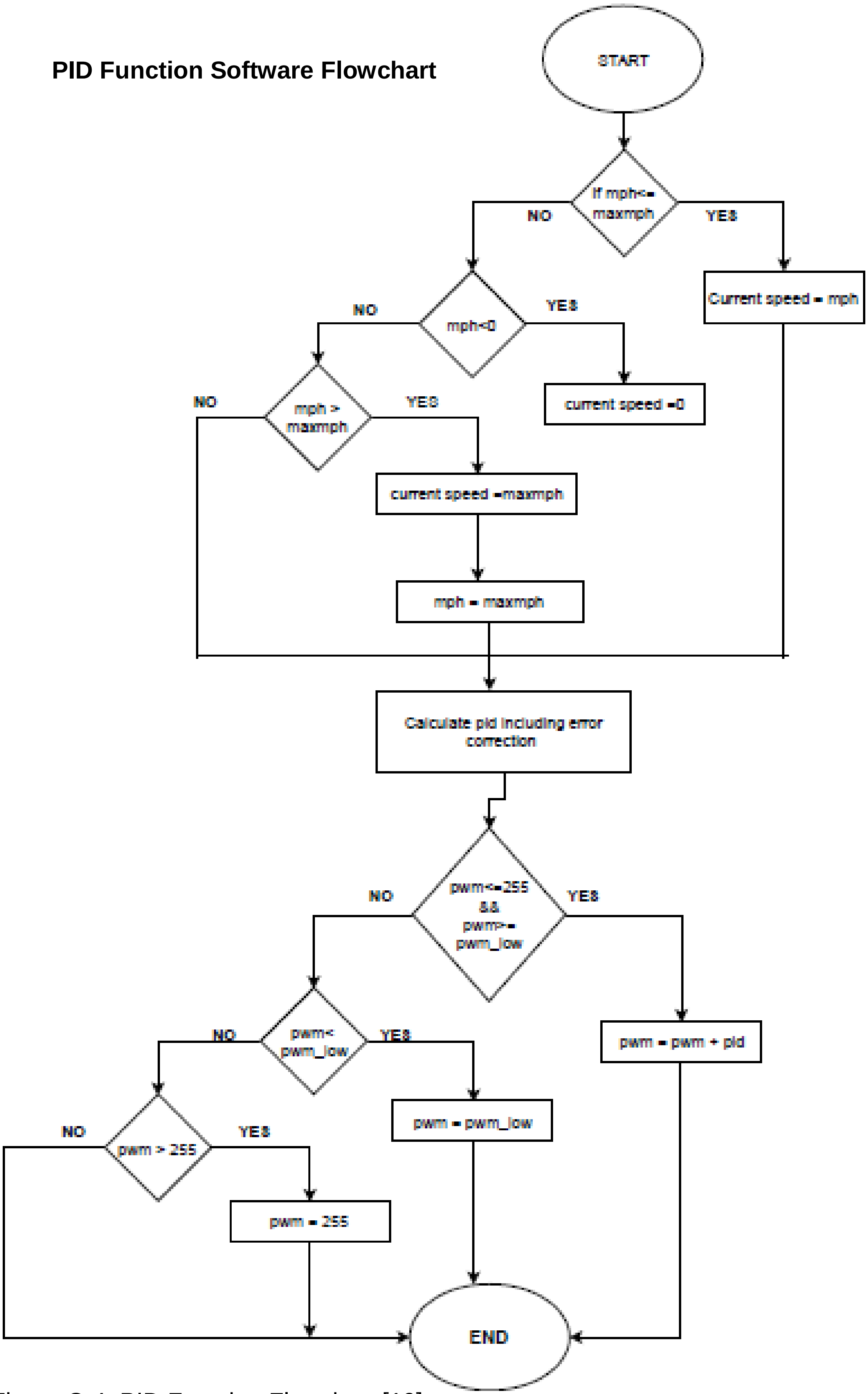

Figure C-4. PID Function Flowchart [19]

Heart Rate Motor Control Software Flowchart

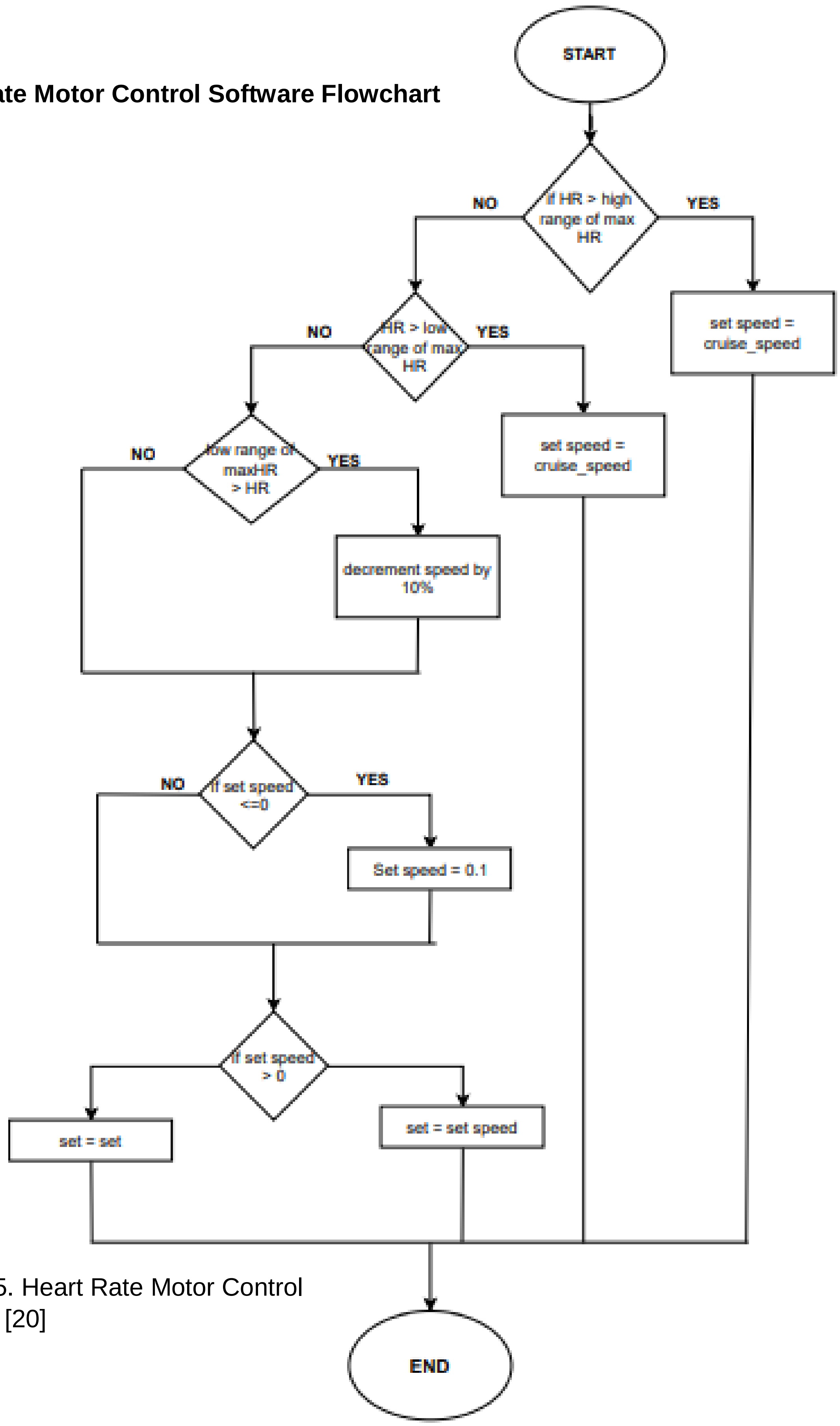

Figure C-5. Heart Rate Motor Control Flowchart [20]

Speedometer Function Software Flowchart

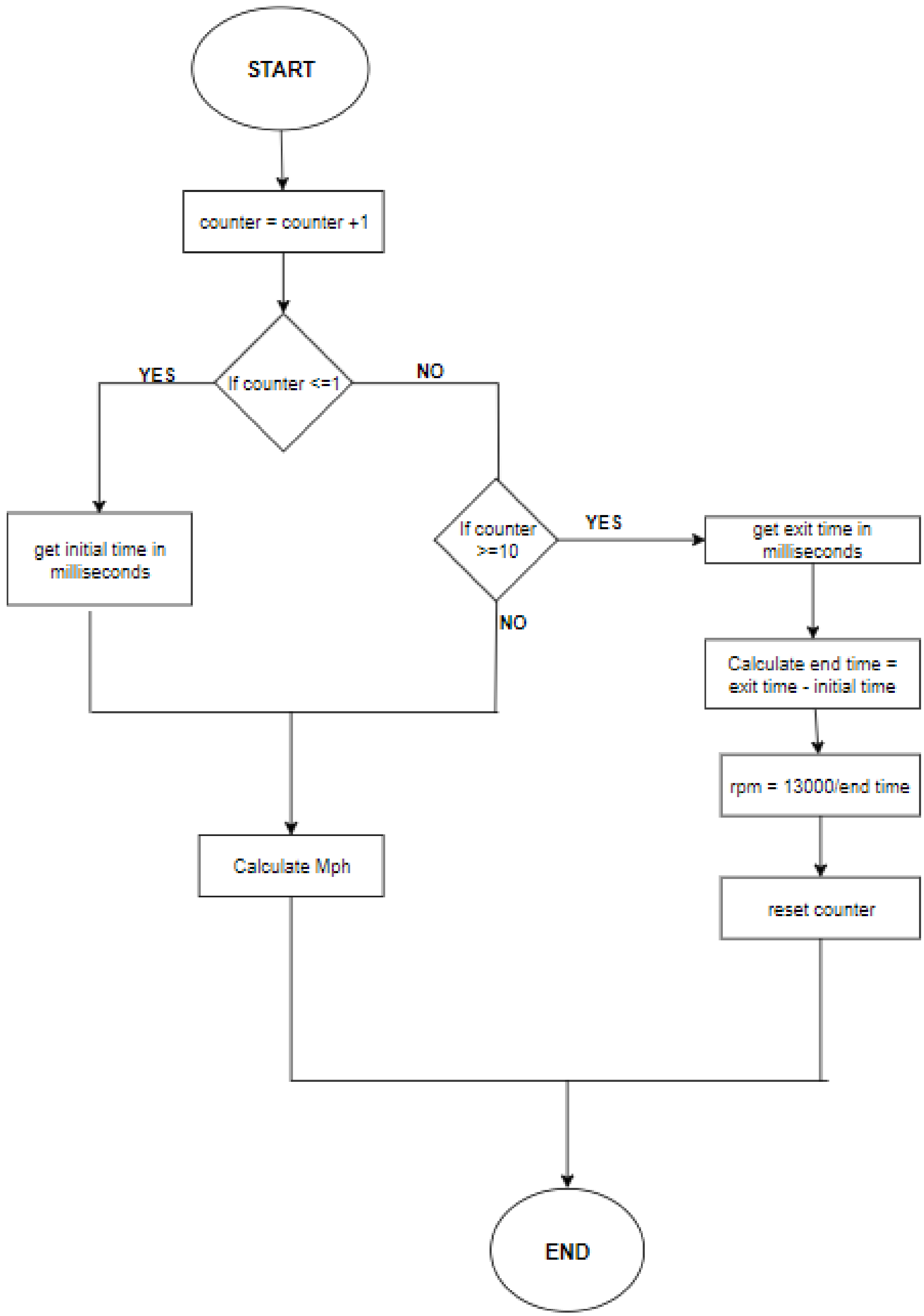

Figure C-6. Speedometer Function Flowchart [21]

Temperature Sensor Software Flowchart

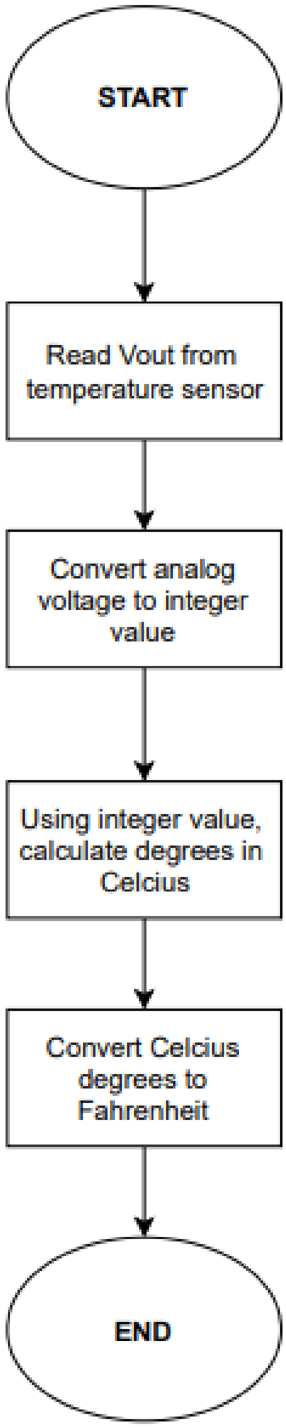

Figure C-7. Temperature Function Flowchart [22]

Appendix D

Mechanical

For our mechanical component of our bicycle we had a few pieces that we integrated together for our final deployable prototype. These pieces include the physical bicycle itself with its attachments of the Raspberry Pi Screen and different clips/nuts/bolts we used to implement our devices and equipment onto the electric-assisted bicycle.

Shimano Bicycle

Figure D-1. Shimano Bicycle [23]

Hub Motor w/ Wheel and Controller

Figure D-2. Hub Motor and Controller [24]

Appendix E

Vendor Contacts

No vendor contacts were used. All costs and purchases were evenly divided amongst all team members.

Appendix F

Resumes

The resumes for all team members are listed in the following order below:

1. Isaac Parrish
2. Pranay Chaudhary
3. Harjot Sidhu
4. Veronica Gonzalez

Isaac Parrish

Objective

• Employment in a company where I can gain long term experience in engineering real things to enhance the quality of life for people. • Find a team to work with long term.
• To build the future.

Education

BSEE | MAY 2019 | SACSTATE

· Major: Electrical and Electronics Engineering ·
Related coursework:

o Power Systems Analysis
o Robotics o Machine
Vision o Electronics II

Skills & Abilities

MANAGEMENT

· I have taken an active role in management of senior design and other projects at school

COMMUNICATION

· I have done many presentations in classes as well as articulated to team members what needs to happen.

LEADERSHIP

· Father and Husband

Pranay Chaudhary

Objective:
Seeking a position within your company where I can expand my knowledge and build on my hands-on experiences with my skillset.

Education:
California State University, Sacramento — **Expected Graduation:** May 2019
Bachelor of Science, Electrical and Electronics Engineering — GPA: 3.0

Engineer-in-Training Certification (EIT), — State of California, March 2019

Fluent in Hindi, English, and minimal Spanish

Related Course Work:

Network Analysis	Logic Design	Communication Systems
Signals and Systems	Circuit Analysis	Applied Electromagnetics
Electromechanic Conversion	Power System Analysis	Robotics
Microprocessors	C Programming	Machine Vision
Feedback Systems	Electronics (BJTS, FETS)	Power Electronics

Project Experience:

- Worked on a feedback systems project for an elevator. Made use of a Parallax Propeller Microcontroller board to control servo motors and push buttons. Programming consisted of using PID functions to improve elevator functionality.
- Senior design project with a team in which we created an electric-assisted bicycle to improve the physical fitness for obese people. Used an Arduino board to collect heart rate data and send it to a Raspberry Pi to determine how much power to provide to the hub motor to achieve a beneficial amount of pedaling by the user for exercise.

Software Knowledge:
Adobe Dreamweaver, Arduino IDE, C, Cadence, CSS, Digilent Waveforms, HTML, Matlab, Microsoft Visio, Microsoft Office Word, Excel, and Powerpoint, Multisim, MySQL, OrCad, PHP, PSPICE, SimpleIDE, Verilog.

Work Experience:
Orchard Supply Hardware — **Aug 2016 –Nov 2018**
Cashier/Customer Service Representative

- Maintained proper communication with coworkers and supervised cashiers
- Provided customer service in a professional manner
- Used basic accounting skills to handle money
- Interacted directly with customers and assisted with product information

California Department of Transportation — **Sept 2018 – Present**
Electrical Engineer Student Assistant

- Create and update Microsoft Excel spreadsheets for data collection
- Update and maintain Filemaker Pro database
- Maintain files for Traffic Signal Timing Sheets
- Programming for websites

Harjot Sidhu https://www.linkedin.com/in/harjotsidhu2

OBJECTIVE:

(916) 871 - 9172

Engineering position to apply my technical abilities and education.

sidhu.harjot96@gmail.com

EDUCATION:

Bachelor of Science

Major: *Electrical & Electronics Engineering emphasis in Controls, CSU Sacramento* Expected Graduation: **Spring 2019**

RELATED COURSES:

Electronics II	Robotics	Microprocessors
Machine Vision	Power Elec Control Drives	Computational Methods and Applications
Logic Design	Network Circuit Analysis	Modern Communication System
Feedback Systems	Applied Electromagnetics	Signals & Systems
CMOS and VLSI	Electromechanical Conversion	Embedded Systems*

TECHNICAL SKILLS:

Programming:	C/C++ • Python • Java • Ladder Logic • Verilog • Assembly
Software Applications:	MS Office 2003 – 2016 • Cadence PSpice • CAD* • Quartus II • Model Sim-Altera • NI Multisim • Visual Studio • ADS • IDE • MATLAB • PLC
Operating Systems:	Windows 7 – Windows 10 • Linux (Ubuntu)*
Devices:	Oscilloscope, Digital Multimeter, Function Generator, FPGA*, Arduino, Raspberry PI, Max32, Parallax Propeller
Verbal Languages:	Bilingual: English/Punjabi

PROJECT EXPERIENCE:

Senior Design Project – Electric-Assisted Bicycle Fall 2018 – Spring 2019

Objective of the project was to design and build an electric-assisted bicycle that helps reduce obesity through safe exercise:

- Motor was driven based on heart rate.
- Monitors environmental temperature.
- Stores all sensor data calculations into a memory log.
- Data acquisition is displayed for user feedback through a GUI, graphical user-interface.

Machine Vision – Thermal Recognition Spring 2019

Objective of the project is to design and build a thermal recognition system:

- Using a Raspberry Pi B, Raspberry Pi camera board, Arduino, IR Sensor, and LEDs.
- Application of this system can be used for facial recognition.
- Coding is done in Python and C

WORK EXPERIENCE:

Engineer Intern Dept of Public Works San Rafael 07/18 – 01/19

- Assisted application of projects using software services
- Provided project management support for engineering operations such as initializing permits, documentation, and process development
- Aid in conflict management between multiple consumers with precise decision making

Veronica Gonzalez

OBJECTIVES

- To obtain a position of employment and gain work experience in a team oriented environment by utilizing my job and educational skills.

EDUCATION & CERTIFICATES

Associate of Sciences in Mathematics — December 2014
CSU of Sacramento, Bachelor's in Computer Engineering. — Expected May 2019
California Guard Card — October 2017- Current

SKILLS

- Microsoft Office: Word, Excel, Power Point
- Good communication skills, experience in writing technical lab reports
- Problem solving and time management, attention to detail.
- Programming in C, Python, Verilog, and VHDL.
- Development Tools: Cadence, MPLAB IDE, Arduino IDE, PSpice, Xilinx ISE.

RELEVANT COURSE WORK

Introduction to Intel Computer Architecture	Matrix Screensaver
Programming Concepts and Methodology II, Java programming.	
Introductory to Circuit Analysis	
Intro to Logic Design	State Diagram Presentation
Intro System Program Unix, programming in C.	
Data Structure and Algorithm Analysis	
Circuit Network Analysis	
Advanced Logic Design using Verilog and VHDL.	
Computer Interfacing	Alarm System
Electronics I – Intro to ideal op amps, BJTs, FETs, etc.	Virtual Op Amp Design
Advanced Computer Organization using MIPS and Verilog	CPU Datapath
Computer Hardware Design	
Systems and Signal Processing	
Computer Network + Internet	

WORK EXPERIENCE

Ship Dock — **Amazon Fulfillment** — Aug. 2016 – March 2017

- Scan packages with handheld scanner and sort into different arc angel locations. Print gaylord labels after scanning and closing it out after it's filled up. Move gaylords with a pallet jack. Scan totes and stack on pallets and closing them out that go to different amazon locations. Re-induct packages from jackpot. Inducting on flat inductor.

PROJECTS

Electric Assist Bicycle — **Senior Design** — Aug. 2018-May 2019

- My team and I designed an electric assisted bike to help reduce obesity through exercise