CPE/EEE Senior Design

May 6, 2014

End of Project Documentation

Authors:
Michael FRITH
David LARRIBAS
Devin MOORE
Benjamin SMITH

Supervisors:
Fethi BELKHOUCHE
Russ TATRO

Abstract

In this document we describe the year long senior project undertaken by undergraduates at California State University, Sacramento. We chronicle the development of a smart electrically assisted bicycle. Discussed in this documentation is the feature set, scheduling, breakdown of tasks, and technological development of Project Forward. We discuss the integration of features and the use of the device.

Index Terms

Bicycles, occupational health, brushless motors, electric motors, current control, closed loop control systems

CONTENTS

List of Figures

LIST OF LISTINGS

LIST OF TABLES

I. Introduction

UNDERGRADUATES look to the Senior Design course as the final opportunity to display an aptitude for engineering before being launched into the professional world. The professors have described the desired experience to be a “resume moment”. This capstone experience encapsulates what it means to be a practicing engineer. The goal of the course is to deliver a technological solution for a societal issue; nothing contrived, but a real problem with a deliverable solution which can compete upon the world stage. Our group is intensely passionate about technology and the benefits it can provide for the community. This document will not only detail the development of our solution, but also records the human side of the project. We look to provide a faithful representation of our successes, failures, and most importantly, growth. Senior Design is a nine month process spanning a student’s entire senior year of engineering school. Growth always comes with growing pains; we’ve had our share of both.

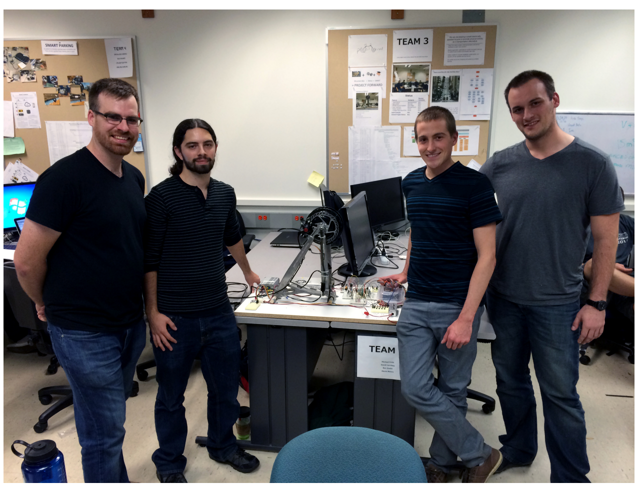


Fig. 1: Team 3 With Laboratory Prototype

Our team formed the summer preceding the start of the course. The first task was to decide on a topic. We needed a project that attacked a societal problem while still providing the group members a means to show their engineering ability. We have two electrical and two computer engineering students which makeup the project team.

Our solution to the societal problem needed to be designed to incorporate each individual team member’s strengths. We looked into the rapid transportation of medicines using UAVs, but we determined this may not be a viable solution for the problem. We looked to the issue of traffic safety and how a vehicle to vehicle mesh network could make highways safer. This idea was passed up because of the lack of work for the electrical engineers. As we discussed different topics, the summer came to a close and the class begun without the team settling on a project. On the first day of the course we settled on a project topic.

All of the group members are involved with cycling in some manner. Ben and Devin were both employed at bicycle shops, Mike is a mountain biker, and both Ben and David have cycle commuted for extended periods of time. Due to the common thread of interest we chose to do a cycling related project. All of us noticed the drop in physical activity as we worked through engineering school. This drop in activity level was particularly troubling. For this reason, we choose to take on the issue of inactivity as it was very close to us. The members of the group are preparing to enter a career focused on sedentary activity. Inactivity, as we’ll explore below, has a host of detrimental effects on health and wellness.

Looking at the nature of the problem we found a few means of addressing the issue of inactivity, and selected a technological solution aimed at enabling commuters. Electrically assisted bicycles are common around the world, but not yet in the United States. Seeing this disparity as opportunity to address the issues of American cycle commuters, we began to work on the design of our project. We wanted to provide a comprehensive solution for the cycle commuter. Through research, we found major obstacles for average people to consider cycling as a primary form of transportation were terrain, safety, and their ride being too daunting for their skill or physical ability. It is possible to provide a technological solution for these problems. Figure 2 shows the basic components of our system. The project includes a heart rate based control system for an electric motor to assist the rider though physically demanding routes. There is an integrated lighting system and horn for increased visibility and perceptibility while riding. There is also a convenient user interface in the form of a smart phone application for the user to interact with the bicycle.

All four members of the group are cyclists of

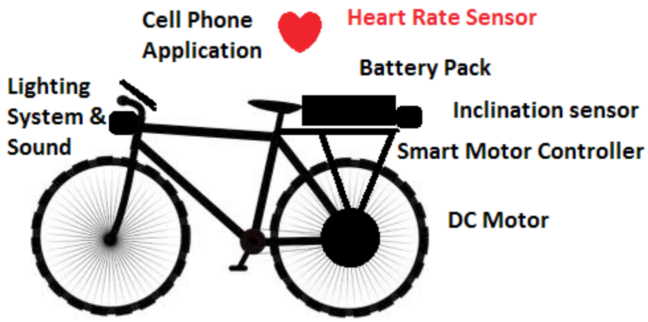


Fig. 2: Project Forward System Overview

varying skill levels. We found that we each represent a potential use case for an electrically assisted bicycle. In all of these use cases, the common factor to our group is the lack of time we can dedicate to exercise exclusively. We all agree that a system which allows us to incorporate cycling seamlessly into our daily lives, by using it as a commuting alternative, could be an answer to our increasingly sedentary behavior. Before the technical details of the project are introduced, let's take some time to analyze the societal problem of inactivity and its effects.

II. The Societal Problem

THE technological revolution of the previous half century has introduced a large number of jobs and leisure options which require little physical effort. This lack of activity contributes to large numbers of the population becoming unhealthy. Previously, technological limitations forced humanity into strenuous work. The advent of industrial machinery has removed the necessity of regular physical exertion. This has enabled many laborers to exchange their strenuous work for a less physically demanding job. This shift toward more sedentary work is having profound impact upon the worker's health.

Leisure and travel have undergone a similar transformations; both of which now favor more convenient means that require a smaller amount of physical effort. These societal shifts towards inactivity have affected the average health of people in a negative way. Societies all across the globe are facing increasing rates of diabetes, obesity, and cardiovascular disease. As technology advances, there is less need for physical exertion which contributes to the world becoming less healthy. Individuals can greatly improve their health with a small increase of cardiovascular activity on a regular basis. There exist opportunities to include exercise throughout daily routines with minimal impact on efficiency, one of which is choosing a bicycle as opposed to a motor vehicle for commuting.

Our understanding of this societal problem has grown over the nine month span we were dealing with the project. The first semester's work brought about a better understanding of the societal problem and its target demographic. During the first semester, we concentrated on the effects of a completely sedentary lifestyle. We felt this was too narrow of a focus which neglected the most intuitive use case for an electric bicycle, those who are already active. The local cycling community's interest was much greater than we had anticipated. This enthusiasm brought with it a concern for safety. This led to a refocus on a technological solution for safety due to the number of related concerns about sharing the road with motorists.

We had already included safety into our solution, but we failed to give it the weight we now feel it deserves. With additional feedback from commuters and other cyclists, we found that safety was among their biggest concerns. This knowledge reaffirmed the group's commitment to the safety system. The design of the safety system was made more robust in light of this information.

A. The Scope of the Problem

Inactivity and its health effects were once the exclusive blight of the developed world. This problem has changed its face over the past few decades to impact an increasing percentage of the world's populace. Seen in 3 and 4, the World Health Organization has found that low and middle income countries have overweight and obesity rates that are on the rise, especially in urban settings, due to the evolving nature of work, transportation, and leisure. [4]

Since the advent of accessible motorized transportation, traveling great or short distances requires only a short walk down the driveway. Stanford School of Medicine noted that our bodies are experiencing a significantly reduced demand for physical activity due to these changing modes of transportation as well as changing workplace environments and entertainment. [5] Nielsen has found

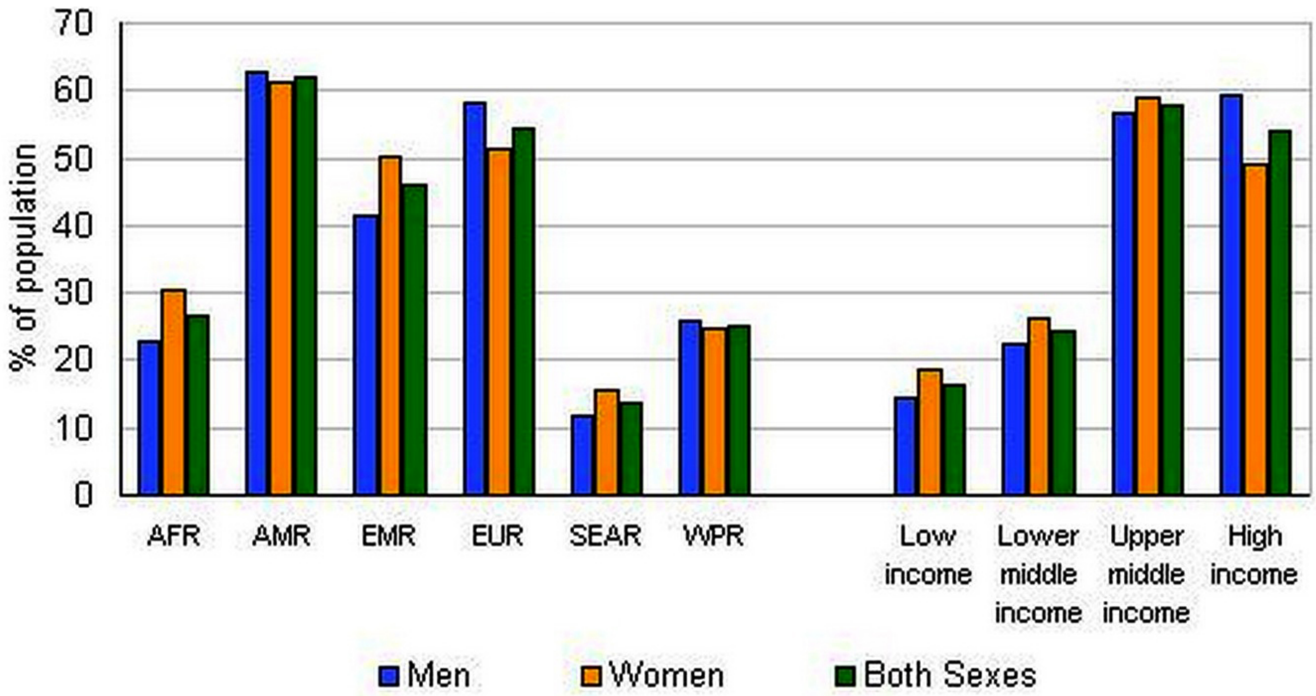


Fig. 3: Percentage Overweight (BMI 25+) Ages 20+ [1]

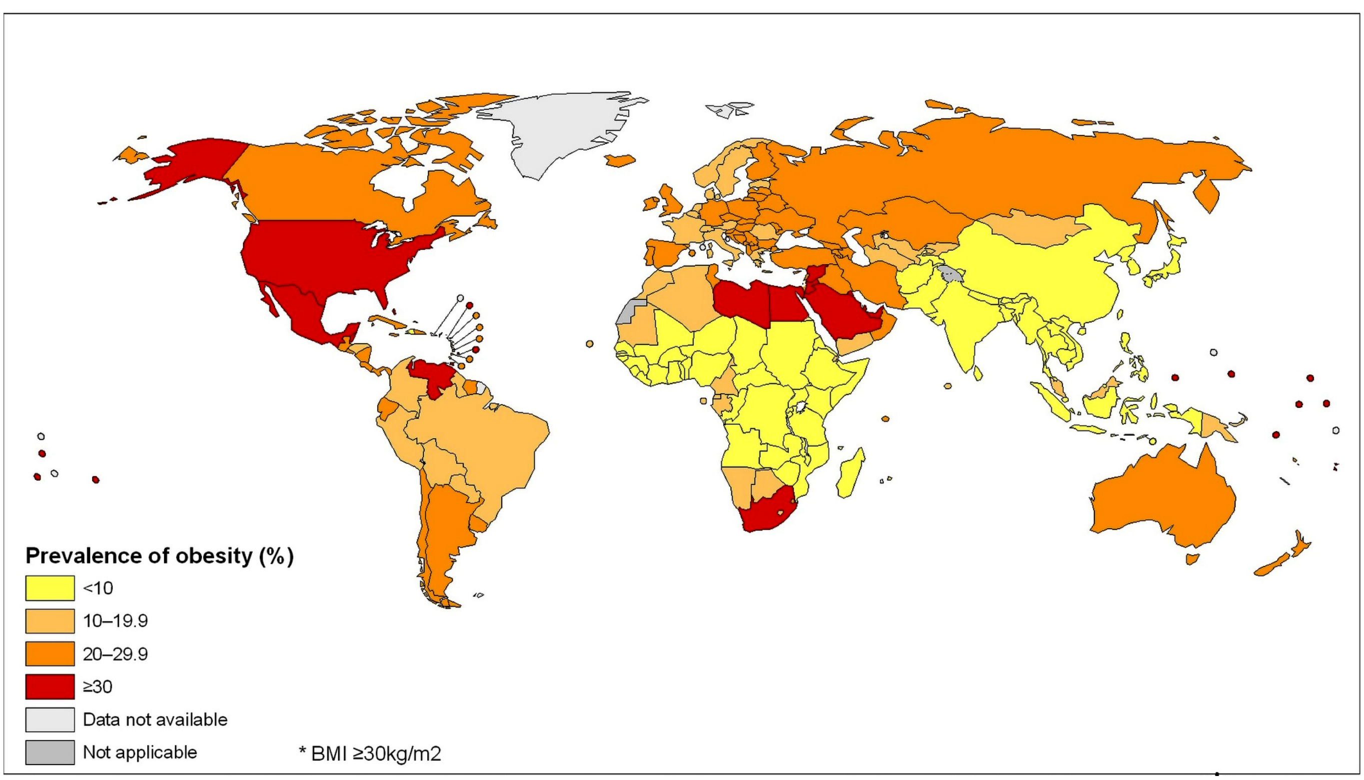


Fig. 4: Prevalence of Obesity, Ages 20+, Age Standardized, Both Sexes, 2008 [1]

the average American spent more than 33 hours a week watching television in 2012, indicative of sedentary entertainment at home. [6] This is not only an American phenomenon. Citizens of the UK may spend upwards of 28 hours a week watching television. As seen in 5, from 1970 to 2006, there has been a long-term decline in annual hours worked leading to more time available for this sedentary form of leisure.

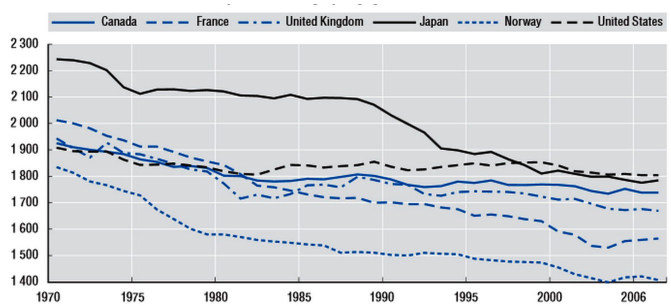


Fig. 5: 1970-2006: Long-term Decline in Annual Hours Worked [1]

B. Health Effects of Inactivity

Inactivity can carry with it major detriments to human health, both physical and mental. These physical and mental effects include stress, depression, obesity, and heart disease. These afflictions carry with them a reduced lifespan and a lower quality of life. Even if people are fairly healthy and somewhat active, their health can often still benefit from increased activity.

With the increase in sedentary lifestyles, there has also been in increase in physical ailments. Obesity has become a worldwide issue because of the great number of detrimental effects it has on the human body and the increase in the percent of population afflicted with obesity. Each year there are 2.8 million deaths worldwide caused by being overweight or obese. 35% of all adults twenty years of age or older were overweight, with many them being obese, according to a survey in 2008. [7] In addition to obesity, inactivity can lead to cardiovascular disease. Nearly half of all adults worldwide are affected by cardiovascular disease which contributes to stroke and kidney failure. There were an estimated 17.3 million deaths caused by cardiovascular disease accounting for 30% of all deaths worldwide, making it the number one cause of death. [8] Hypertension alone, a form of cardiovascular disease, is responsible for 9.4 million deaths per year worldwide. [9] Diabetes doubles the risk of death when compared to others within the same age group. Diabetes will damage the heart, kidney, nerves and even blood vessels. [10] Inactivity is not the only cause of the previous ailments, but physical activity can alleviate or even cure these effects.

Physical activity increases energy levels with as little as 30 minutes of exercise a day. [6] This can promote many benefits at home and in the workplace such as increased energy and productivity. Adequate physical activity can provide energy throughout the day which can increase focus on any task, physical or not. [11] Regular physical activity will increase cardio-respiratory and muscular fitness. [9] Individuals can increase wellness by engaging in regular physical activity 150 minutes a week for adults, which is less than 30 minutes per day. [4]

In addition to the physical conditions caused by physical inactivity, it also causes many negative mental effects. These include depression, anxiety, and low self-esteem. Depression can cause a decrease in work performance which results in a significant decrease in employee profitability. [12] More than 350 million people worldwide have depression making it one of the leading causes of disabilities in the world. [5] Anxiety, though it can be beneficial in small doses, large amounts can turn into a hindering disorder. These disorders include generalized anxiety disorder, obsessive-compulsive disorder, panic disorder and post-traumatic stress disorder. [6] Obviously, physical inactivity does not cause PTSD, but regular physical activity can help alleviate the symptoms of all these forms of anxiety. Additionally, with the increase in weight control that comes with physical activity, a better self-image may also be produced.

Increased physical activity can cause a person to look and feel more in shape. Physical activity can lead to an increase in performance and has been shown to lead to improved self-esteem. Depression will also improve with moderate levels of physical activity. [6] As one ages, moderate aerobic and strength training activities 3-5 times a week for 30-60 minutes can improve aspects of mental health, such as improved thinking, learning and judgment abilities. These activities can also help with movement, keeping joints, bones, and muscles healthy as well as slowing bone density loss. [4]

C. Existing Solutions and What Our Solution Offers

There are many ways in which one can become more active. Examples include running, bicycling, swimming, and working out at the gym. Most of these activities require specific time to be set aside for them. Many people have trouble allocating this time to activities they have to go out of the way to do. Time at the gym means less time for whatever is most important to that person, such as time with their loved ones or with their TV.

Some forms of exercise can be harmful to at-risk groups. Running or strenuous weight lifting are common examples due to their high impact nature. Cycling and swimming are known low impact exercises that do not have these drawbacks. Cycling, however, can be used in many cases to integrate exercise into their daily commute. Although half of all car trips made are under three miles, only 1.6% of Canadians and 0.6% of Americans commutes to work via bicycle. [13] In high traffic areas it may often be the case that a cycle commute will take less time than a car. Replacing short car trips a few times a per week with a bicycle could increase a person's activity level without having to set aside a large amount of time for exercise.

Cycling is by no means a panacea. Getting started can be a daunting task for the inexperienced as well as those with physical limitations. Large hills can also prove discouraging - if not incapacitating - or a cyclist may view their commute to be too long. Safety is a concern for any rider, whether on the street or on a trail. A solution which can address these concerns could be a viable option for a commuter, while providing an excellent form of physical activity. Our solution is a smart electrically-assisted bicycle that can provide a safe and tailored riding experience for users.

D. Experience From the First Semester

The process of developing our design in the first semester brought about some changes in our understanding of the societal problem. While the core of the design did not need to be adapted, it benefited from some additional thought in light of the experience. We were surprised to see that there is much more interest from cyclists that we had anticipated, and much more interest in safety than we had planned for. There is also a stronger-than-anticipated aversion to weight from the enthusiast community. These issued were addressed by increasing the visibility of the rider with large indicator lights and a significantly loud horn.

E. Interest From Cyclists

We spent the majority of the first semester focusing on the group of particularly inactive people. Talks with the cycling community have shown this to be too narrow of a focus. The intermediate/enthusiast level has shown the most interest. This demographic is particularly interested in the weight of the system. The lighter battery configurations would be of particular interest to this group. This group also considers themselves to be techno savvy and willing to adapt new technology that enables them to get the most of their ride. However, the elite-level cyclist did not show the same level of enthusiasm. Additionally, current commuters using gas-powered assist expressed interest in going green with an electric solution.

F. Safety Is Paramount

Many of the people spoken to about the project, and bike riding in general, confessed they don't feel safe riding their bicycles due to the possibility of being hit by a car. This is a fair concern considering in 2011 there were 677 cyclist fatalities and 48,000 injuries caused by motor vehicle collisions. [14] The leading cause of bicyclist fatalities is due to motorist misconduct. [15] We are not able to change how a driver behaves, but we can help make drivers aware of cyclists on the road. We do this through safety lighting. For the cases in which a driver is not paying attention visually, the rider will also have an audible alert. This will be the most useful when the rider notices a driver starting to turn in front of them. These types of accidents are the most common cause of fatality to bicyclists.

G. Weight

An informal survey resoundingly stated one of the big reasons people don't ride their bikes was that their bikes are too hard to move, especially up and down stairs in apartments. Our innovation isn't necessarily the bike itself, but the features we offer to electric bicycles. With that being stated, our test bike weighs 38 lbs with the motor installed. We found it a little difficult to lift the bike onto a roof mounted bike rack. With the batteries and the rest of the gear, this weight has ballooned to 75 pounds. Things to keep in mind as far as ways to help with this problem would be to start with a lighter platform. Road bikes can weigh as little as 15 lbs (minimum weight required by UCI, the international cycling authority, to ride in sanctioned races.) Additionally lighter batteries exist than our 30 pound SLA (sealed lead acid) battery, we have the opportunity, at much financial cost, to reduce the weight of the batteries to 8 lbs. With these very expensive solutions, we could have a full system as light as, or lighter than, 45 pounds, almost half of what our deployable prototype weighs in its current form.

H. Our Target

We are no longer focusing on the extremes of physical activity: the seriously at-risk groups and the super-athlete. Over the course of this last semester we came to realize that the four of us actually represent our best use-case. During the winter break some of us tried cycling again and noticed that we had suffered from sitting too much this past semester. The irony was not lost that this was due in large part to devoting as much time as we did to this very project. If we had cycle commuted regularly over the course of the previous semester, we believe that the benefits would enhance our personal wellness and keep us in better physical shape. We also believe it would have improved our performance in school by relieving anxiety and provided increased energy.

I. The Need for a Solution

The increasingly sedentary behavior of humans worldwide is a detrimental trend. People are turning towards television and computers for entertainment as a leisure standard and driving as their primary mode of transportation. This increases the tendency toward an inactive lifestyle; unfortunately, this lack of activity has adverse effect on peoples health. Detrimental health effects include an increased risk to cardiovascular disease, diabetes, as well as depression and anxiety disorders. There are many ways people can become active, however many of these are non-viable or inconvenient due to busy lifestyles or physical limitations. An activity that can easily be worked into daily life, while remaining accessible, would provide a practical solution to inactivity. Bicycling can be used for low impact exercise, travel, and leisure, and has the ability to be conveniently integrated into daily life. If bicycling became more accessible, an increase in its use could benefit the health and wellness of society.

We feel enabling a commuter to choose cycling as a viable form of transportation provides us a unique opportunity to create a technological solution to the problem of physical inactivity. Many solutions exist on the market that place an electric motor on a bicycle. Many of these systems allow the rider to use the bicycle as an electric motorcycle or trick the motor into an assistance mode by merely moving the pedals past rotational sensors without actually supplying any of their own effort into the system. Knowing that simply putting a motor on a bike would not suffice to provide any solution of substance to our chosen problem of inactivity, it became apparent that a multi-faceted approach would be necessary. Our solution would have to address why people are not utilizing cycling as well as ways we could provide assistance to a cyclist to perhaps entice them into choosing cycling. Additionally, we wish to ensure that our system has marketability.

Our research shows that electric bike date back to as early as the late 1800s [16]. Despite their introduction over two hundred years ago, they have not had widespread use in the United States. However, with the recent advances and cost reductions in battery storage and electric motor technologies, these bicycles have become much more attractive to consumers.

Large manufacturers are beginning to produce electrically-assisted bicycles but the market has yet to define itself. Ed Benjamin, chairman of the Light Electric Vehicle Association(LEVA), compared the American and European markets, “[In Europe] we have a nine-year period in which sales went up to 10 times what they were in 2004.” [17] The European market is booming. Figure 6 shows the dramatic increase.

Benjamin proceeds to describe the state of the US market, “We’re going to see something similar in the United States, but it’ll be a little bit slower. The United States is not a bicycle-as-transportation culture. We are a cars-are-transportation culture.”

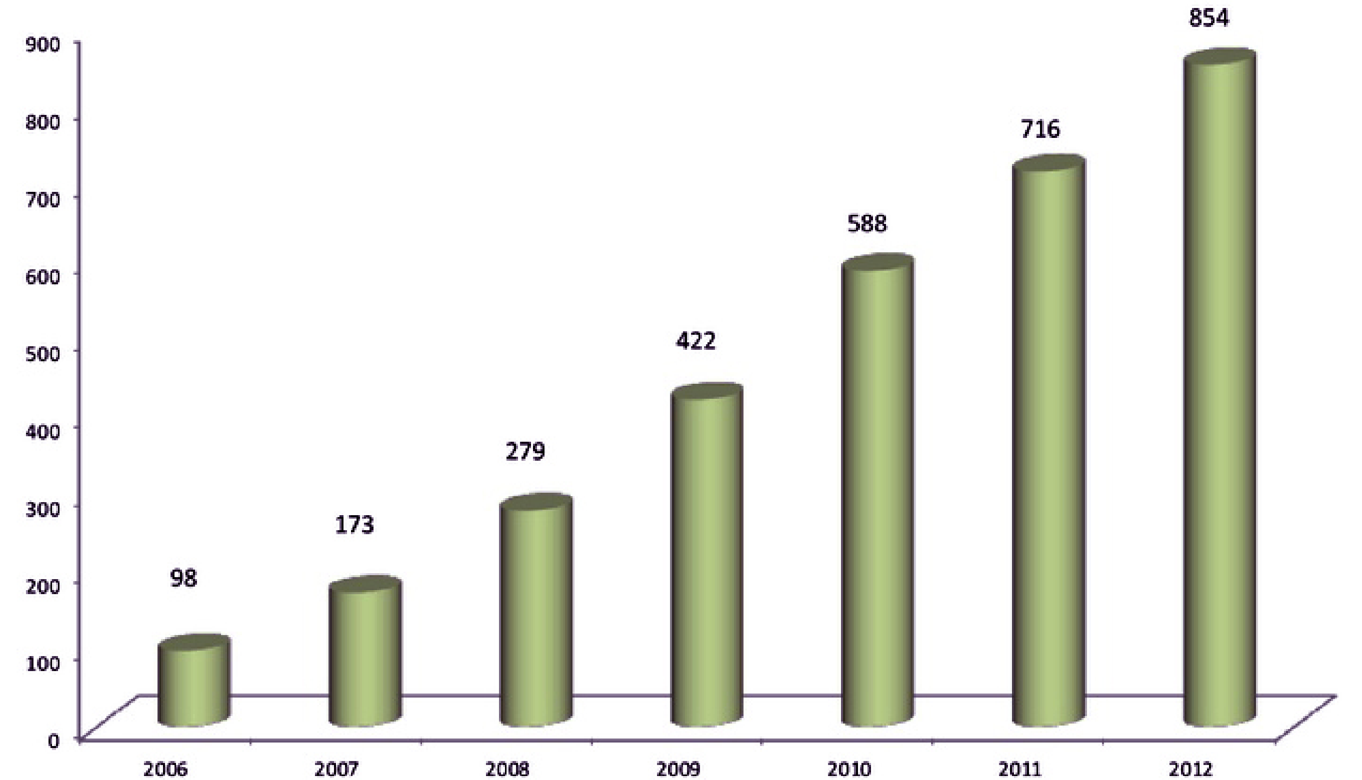


Fig. 6: European Electric Bicycle Sales by Year (Thousands of Units Sold) [2]

[17] This difference can be observed by comparing several geopolitical sales of electric bicycles. While North America and Western Europe have similar population sizes, North America purchases roughly half as many electric bicycles.

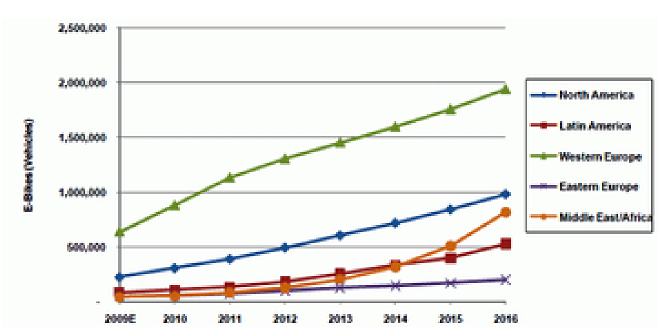


Fig. 7: Electric bicycle sales by year [3]

When looking at this data, it is our conclusion that the US market is not saturated, and is in fact at a tipping point to embrace electric bicycles if they could provide the same capabilities and assurances that a car could. We feel that we can have a successful and marketable solution if we address the problems introduced in our societal problem of why don't people choose bicycling. The reasons we feel most capable of addressing break down into two main categories: the lack of ease of cycling as well as a perceived lack of safety.

While we can address the lack of ease by simply placing a motor on a bicycle like most existing electric bikes in the market, we feel that does nothing to encourage a rider to exert effort and does nothing to tackle our goal of getting people active. To enable a cyclist, we need a smart system capable of knowing when a rider needs assistance and when they don't. This is handled through a smart deterministic control system that utilizes biometric feedback to measure a rider's level of exertion, and predicts when they will need assistance by reading the bicycle's current inclination.

Addressing the problem of safety, we wish to minimize car-cyclist interactions. We will do so by increasing the visibility and perceptibility of the rider, as well as making them more predictable to cars, pedestrians, and other cyclists. We will also help the rider see where they're riding at night or in the dark. We have accomplished this through the creation of a lighting system including a headlight, tail light/brake light, as well as blinkers. We have also included an audible alert device to provide an auditory awareness component to this system.

Lastly, our system will utilize a mobile device to display data to the rider and allow them to tailor their ride to their own level of need. This device comes in the form of an Android application that communicates wirelessly to our smart control system. We provide the capability of charging this device through a power distribution hub utilizing a USB connection.

J. Societal Problem and Its Relation to Design

The societal problem gives the design phase of the project its purpose. The specification for the design phase rises out of the needs identified in this section. The design needs to address the problems that keep people from using alternative commuting methods. The primary components are the safety issue and the lack of physical fitness. These are the core components of what our technological solution needs to address.

III. Project Design Phase

THE design phase of the project offered a great opportunity to exercise the creative aspect of engineering. After a thorough analysis of the societal problem we knew what we wanted to provide for the end user. Transforming this loose set of specifications into an engineering document is the challenge of a design engineer and what we faced in this section of development. The first step of this process was to define what the instructors referred to as a "punch list", or a design specification that could be easily verified at the end of the project to ensure we had done what we set out to do. The list is then expanded upon to define what each of the features required, what will a deliverable will be for each task, and how it will integrate into the final product.

A. Punch List

The "punch list" is a listing of features to be implemented in the final product. These core features represent the bulk of work to be done in the project. They fall into five basic categories. Their individual tasks will be expanded upon later in the hardware and software documentation sections. This

Fig. 8: Team 3 Discussing Design

was the feature list decided in the second week of the project during the design phase. While the actual implementations of the features changed throughout the nine month process, the intention of the features were implemented.

- Smart Assist
  - FPGA Based Control Mechanism
  - Inclination Based Assistance Control
  - Effort Based Assistance Control
- Motor and Motor Control
  - Commercial Motor Integrated Into Design
  - Safe Connections
- Power Distribution
  - 5V Subsystem for Control Components
  - Battery Charging Circuit
  - USB Charger for Cell Phone
- Safety and Lighting
  - Headlight and Taillight
  - Front and Rear Turn Signals
  - Horn
- User Interface and Controls
  - Bluetooth Android Control Application
  - Real-time Data Display
  - Data Logging

B. Punch List Feature Definitions

1) Smart Assist: The amount of motor assist is controlled in real time, based upon inclination of the system and the rider's heart rate. This is accomplished by interfacing with an accelerometer and gyroscope for the inclination, and a wireless heart rate sensor for the rider's heart rate.

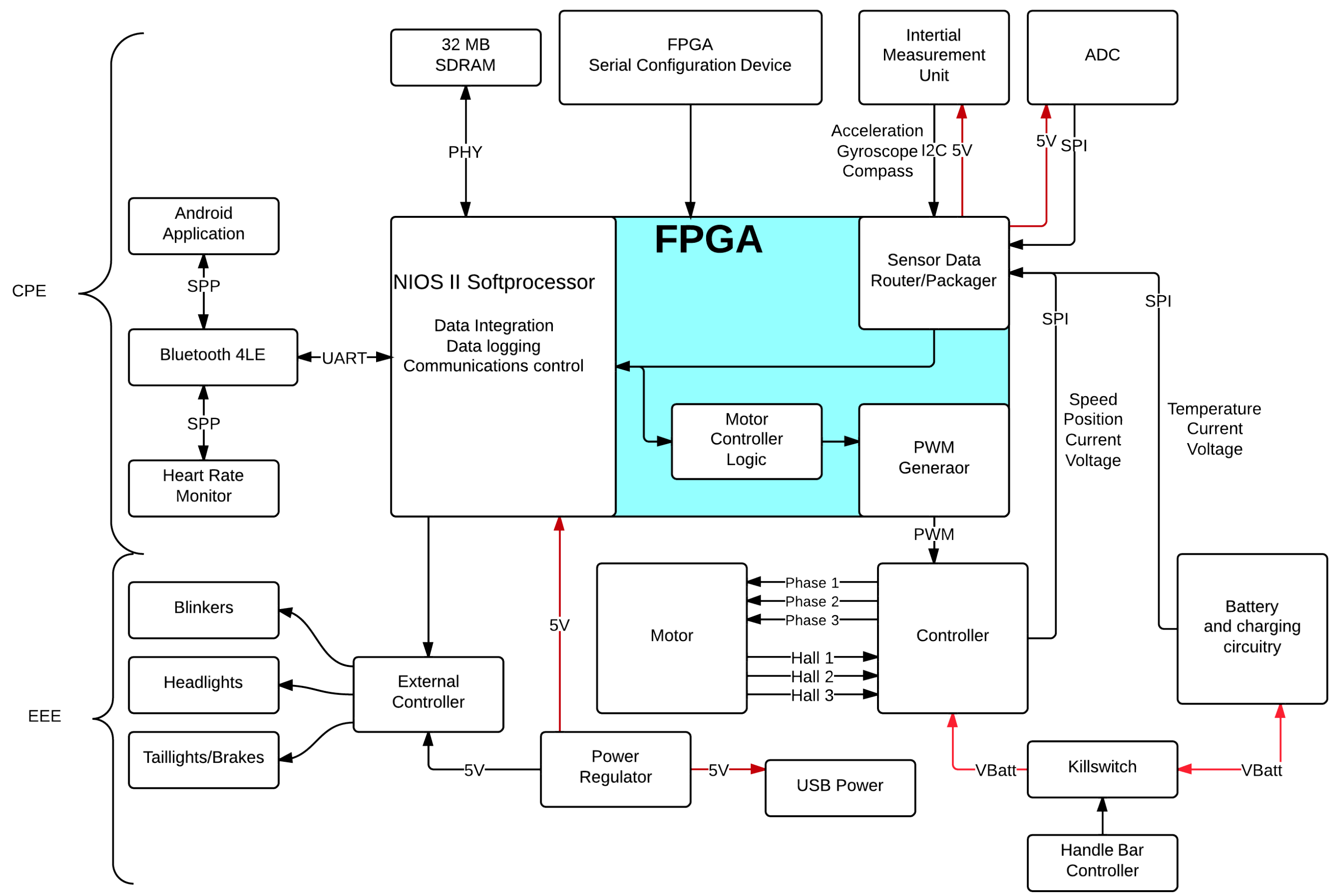


Fig. 9: Block Diagram of Control Hardware

a) Qualifications for Success: Smart control can be tested by isolating a control variable and manipulating it while recording the systems output data. The control algorithms are to be developed in a simulation environment and tested against generated data. A successful match between theoretical and experimental data will validate the systems performance. Once the assistance has been verified in the lab, the rest of the qualifications will come from the rider's experience with the fully integrated system.

2) Motor and Motor Control: The motor provides the assistance for the user. This is a very important part of the design, the user experience is directly affected by this system.

a) Qualifications for Success: The motor will power the wheel to variable speed and power depending on the incoming signal. The electronic speed controller shall detect wheel position and speed and control the motor to maintain a speed given an input signal.

3) Power Distribution Hub:

a) Buck Voltage Regulator: A buck regulator is provided to convert the 12 volts from one of the batteries to a 5V source to be used by the external components and the FPGA. The regulator supplies up to 3A of current which could be required at any given time.

b) USB Power: A female USB cable connected to the 5V power bus at the output of the buck regulator is available on the FPGA enclosure. This is intended to allow the user to charge their cell phone while riding. As a cell phone is required to operate the system, we felt it necessary include this feature.

4) Safety and Lighting:

a) Control Algorithm: The control algorithm also is a form of safety check in the unfortunate event that the user falls over while on the bicycle. The motor will turn off when the bicycle tilts beyond a certain angle. This is also applied to a backwards angle, the direction of a back flip, to prevent the user from flipping over backwards.

b) Power Safety: A hard kill switch is placed between the battery and ESC which completely severs the motor from its source of power. Because the motor is an inductive load, we have also included a flyback diode to ensure current has a path back to the battery. A physical handlebar switch is also attached to the ESC's "ignition" line as a redundancy to disable the motor. The benefit to the handlebar kill switch is, it allows the rider to disable the motor while on the bike without needing to turn around. To prevent too much current from being drawn from the battery, we have incorporated a fuse in series with the battery and the hard kill switch.

c) Safety Lighting: The safety lighting consists of a headlight, brake/taillight, and front and rear turn signals. The headlight illuminates the area in front of the bicycle in order to increase the riders visibility and perceptibly. The lighting must not interfere with a driver's ability to see and thus must not shine directly into a driver's eyes. LEDs are the chosen source of illumination because of their low energy consumption and durability. The headlight is activated manually by a switch on the headlight enclosure.

The brake and taillight are a red set of LEDs which will be active at all times the headlight is active, but at a reduced intensity. When the user applies the brakes, the intensity is increased to full to alert those behind the rider that the vehicle is slowing.

The turn signals consist of amber LEDs mounted on the rear and front of either side of the bicycle. The lights are controlled by a toggle switch to determine which set of lights to blink. Both lights remain off when the toggle is in the center position.

d) Safety Audible Alert: An audible alert device is a system composed of a differential output amplifier fed by an R2R DAC. The horn is activated through a momentary switch on the headlight box that activates a ramp signal generator on the FPGA to be fed to the R2R DAC input. The output of this system is fed to a 4-inch 4-watt speaker mounted on the front of the bike. The purpose of this system is to alert pedestrians, cars, and animals that a rider is present.

5) User Interface and Controls:

a) Cell Phone Application: For the system's user interface, we wanted something simple and familiar to the majority of our target market. From our research, we discovered some of the newer commercially available electric bike solutions, like the Copenhagen Wheel [18], provide smart phone applications. The cellphone application will primarily act as the rider's dashboard. It will display the rider's speed, heart rate, and inclination. The user is required to enter a desired heart rate cap and send it to the FPGA for use in the assistance algorithm. The application is meant to be used from a cell phone mounted to the handlebars much like commercially available GPS systems.

b) Handgrip Controls: We have placed several controls at the fingertips of the rider. They are a motor disable switch, a turn signal switch, a switch to turn on or off the headlight and taillight, as well as a button to press to activate the audible alert. They have been mounted on the headlight enclosure. A brake lever also has a switch that tells the FPGA it has been activated and to disable the assistance momentarily and activate the brake light while the lever is activated. Hard switches were chosen for these as we wanted them to operate with or without the assistance on.

C. Summary of Design Phase

The set of features we have included in Project Forward we feel successfully address a number of the reasons people choose or are forced to be inactive. We feel these features can enable an average person to try commuting on a bicycle as an alternative to more sedentary forms of transportation. Putting together these systems proved to be more expensive and labor intensive than anticipated, but the group was able to deliver the full design specification. The design feature set was then built into a prototype that would demonstrate the various functionalities of the project for the course instructors and members of industry. This is called the "Deployable Prototype" its creation and function is documented in the section to follow.

IV. Creation of Laboratory Prototype

To successfully complete all of our goals, a significant amount of work and time had to be put in to each of the subsystems. During our first semester together, we worked on creating a prototype that could demonstrate our feature set. This prototype's purpose was to prove the feasibility of our design idea and show significant technical progress. The prototype was presented as seen in Figure 10 and the demonstration satisfied the prototype requirements. The following section describes the financial cost and labor hours required to develop this prototype.

A. Funding

Our goal was to create a system that is less expensive than the premium electric bikes on the market, such as the Optibike r8HD that comes in at $13000. We initially budgeted $1500, which is similar to what a decent name brand bike goes for. Unfortunately we missed our target budget due to cost overruns by underestimating part costs. Other overruns occurred because of the hurried nature of the build. We also purchased multiples for some parts to ensure we could continue development in case of something breaking. The initial funding proposal can be seen below along with our expenditures for the laboratory prototype, deployable prototype, and total project spending. We obtained grant money from the school for two ESCs and an extra hub motor.

TABLE I
COST ESTIMATION FROM FALL

Part	Cost in Dollars($)	Purpose
DE0-Nano	100	Central Processor
9 DOF IMU	100	Orientation Sensors
Bluetooth	50	Communication
Electric Motor	150	Propulsion
Motor Controller	100	PWM to Motor
Battery	400	Power
BT4 Heart Rate	70	HR Monitoring
Bicycle	FREE	Framework
Cell Phone	FREE	UI/Controls
Miscellaneous	75	Framework
TOTAL	1050	N/A
TOTAL +35%	1417	Room For Error

This section tabulates the costs for building our laboratory prototype. All funding was provided internally by group members. The costs are compiled in Table II.

B. Schedule and Milestones

Our final goal of the first semester was to have a prototype that could easily demonstrate all of our features in a laboratory setting. To accomplish this, we had to reach certain milestones. The first of these milestone was the breadboard proof, where we would demonstrate the validity of our project. The second milestone was the midterm technical review where we would demonstrate all of our features working together as a system.

For the breadboard proof, we demonstrated the FPGA controlling the motor using the first revision of our assistance algorithm. This section is the core of our control system; therefore, its achievement would prove the validity of the feature. In order to accomplish this, our prototype platform needed to be completed, including the wiring and testing of our motor, ESC, and power source. More importantly, the first iteration of our IMU needed to be functioning. The IMU required the writing of an I^2C interface from scratch in Verilog on our FPGA. This would allow us to read and write to registers on a 9 degrees of freedom sensor stick, specifically the accelerometer and gyroscope. From these readings we were able to get a basic idea of inclination. The output of inclination fed a Verilog PWM module, which output a variable duty cycle 3.3V square wave to the throttle input of our ESC, allowing for demonstration of inclination based assistance.

The next milestone was the midterm technical review. The midterm technical review is where we demonstrated our features integrated as a laboratory prototype. Our various subsystems were completed, from safety to Bluetooth cell phone integration, as well as biometric feedback into our control algorithm. The inclination based portion of the control algorithm was enhanced by the use of filters and the implementation of a gyroscope to build a complementary filter, thus increasing the resolution of our determining angle. The tasks involved in reaching both of these milestones can be seen in our WBS diagram, Figure 12.

TABLE III
FALL SCHEDULE AND MILESTONES

Date	Milestone
September 17th	Design Idea Contract
October 8th	Breadboard Proof
November 12th	Midterm Technical Review
December 3rd	End of Term Documentation
December 10th	Lab Prototype Demonstrations

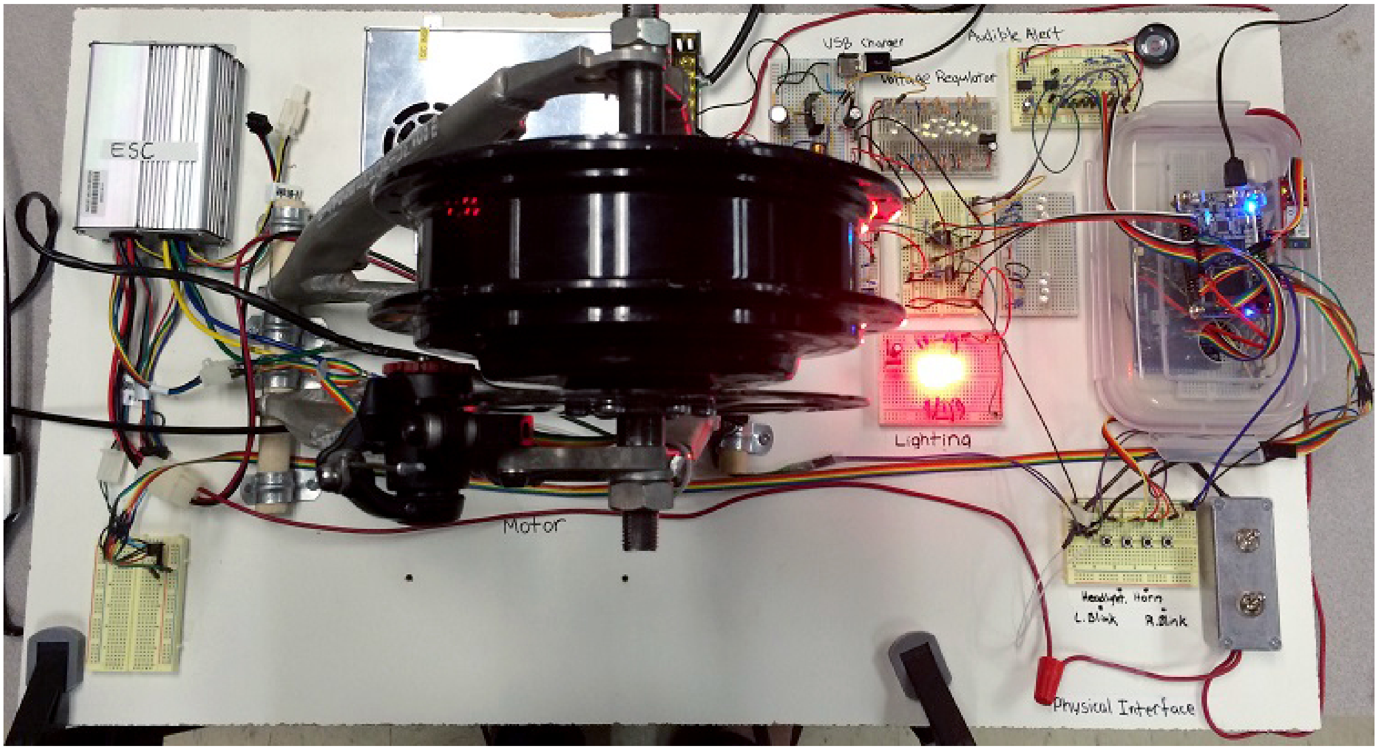


Fig. 10: Laboratory Prototype on Display

TABLE II
FALL SEMESTER INTERNAL FUNDING PROPOSAL

Purpose	Item	Supplier	Cost
Heart Rate Transceiver	PAN1325	Digikey	65
IMU Sensor Stick	9DOF Stick	Sparkfun	100
PCB Fabrication for PAN1325	PCB	OSH Park	21
Hub Motor	BMC V4-T	SF-eBikes	440
ESC	Lyen 9Fet Mark II	Lyen.Com	120
Brake Levers	Ebike Brake Levers	Lyen.Com	20
Wires	14 Gauge Wires	lowes	28
FPGA	DE0-Nano	Newark	0
Lighting	LEDs and Resistors	Digikey	20
Test Fixture	Brackets, Wood, and JBweld	Home Depot	20
Brakes	Rotor	Peak Adv.	6
Voltage Regulator	Various Components	Digikey	30
Kill Switch/Physical Interface	Various Components	Metro Electronics	28
DC Power Supply	S-350-36	Amazon.com	40
Heart Rate Transceiver	AP2	Digikey	65
FPGA Development Board	DE0-Nano	TerasIC	100
Bluetooth Transceiver	Bluetooth Mate Gold	Sparkfun.com	65
Misc Parts + IC's	Metro Electronics	Metro Electronics	176
		Totals	1344

C. Breakdown of Work Required

Managing a project of this scope as a whole would be difficult, so we broke the project down into deliverables, and more granularly into the tasks it would take to create these deliverables. The deliverables and their associated tasks for the laboratory prototype are discussed below. A summary of the number of work hours to build the laboratory prototype can be seen in Table IV.

Fig. 11: Team 3 Developing Lab Prototype

TABLE IV
LABORATORY PROTOTYPE WORK HOUR SUMMARY

Task Category	Ben	David	Mike	Devin	Group
Administrative Activity	60	48	32	35	159
Documentation	103	72	74	109	358
Research and Acquiring Hardware	5	16	16	0	44
Designing Assistance Algorithm(Verilog)	110	0	0	58	168
Developing Android Application(Java/XML)	0	0	0	37	37
Building Prototype Platform	9	23	37	7	76
Designing and Building Safety System	3	22	22	3	50
Creating Voltage Regulator and Power Distribution Hub	0	33	0	0	33
Designing and Building Physical Interface	0	0	12	0	12
Obtaining Motor/ESC Data	0	0	16	1	17
Testing Prototype Demonstration Platform	31	22	26	25	104
Total Hours	307	228	228	273	1154

D. Motor System

The electric drive system is the physical means responsible for providing assistance to a rider. It is made up of an electronic speed controller (ESC) and Brushless DC (BLDC) hub motor.

- Total Estimated Time: 20 Hours
- Assignee: Mike
- Deliverable: A Brushless DC motor working with a compatible ESC
- Mike's Actual Time: 32 Hours

1) Speed Controller: The ESC accepts a PWM signal from our FPGA to adjust the speed of the hub motor. Speed data is transferred back to our FPGA.

a) Selection: We selected a speed controller during our laboratory prototype development phase to be relatively affordable, durable, and have some data readouts we can utilize. We selected a Lyen 9 FET Mark-II based on these criteria.

b) Integration: We want to read speed of our motor from the ESC which has an output of 5V. This output is pulsed in relation to the rotational speed of the motor. In order for the FPGA to read this pulse, the pulse must be stepped down to a safe 3V. This was achieved through the implementation of a voltage divider. We mathematically deduced the relationship between the frequency of these pulses to the hub motor's rotation. Using an oscilloscope, we verified our findings that for every four pulses per section, three physical rotations occur per minute. We created a Verilog module to do this math and provide data to our mobile device.

2) Motor: The motor converts energy sent by the ESC to rotational power, utilized to propel the rider along.

3) Integration: With all the individual components selected, the next step was to integrate these components. Phase wires from the motor are connected to the ESC to provide power to the motor. Phase position wires are also connected to relate the position of the motor back to the ESC.

a) Wiring: During the laboratory prototype development, we wired the ESC to the motor by color matching from the controller to the motor. Our

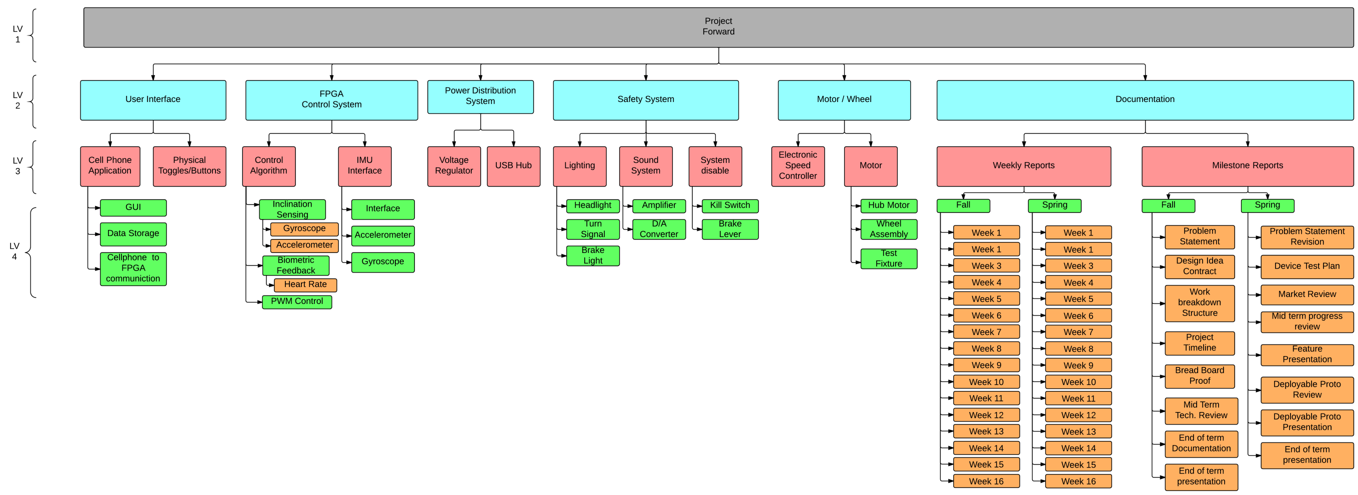


Fig. 12: Work Breakdown Structure

motor performed sporadically, which we attributed to the DC power supply we used to power the system.

E. FPGA Control System

The Altera Cyclone IV FPGA contains all of the control logic for the system. The control algorithms were written entirely in System Verilog and synthesized with the Altera Quartus development environment. The synthesized logic gathers the data required to determine the amount of assistance the rider requires and adjusts the setting for the motor at 4Hz. The purpose of this system is to be transparent to the user once it's parameters are set. The control system should provide assistance when required and not disturb the user's riding experience while in motion.

- Total Estimated Time:160 Hours
- Assignee: Ben, Devin
- Deliverable: Control system to provide assistance with electric motor, based on user heart rate and inclination.
- Ben's Actual Time: 110 Hours
- Devin's Actual Time: 58 Hours
- Total Actual Time: 168 Hours

1) IMU Interface: A communications protocol was developed to communicate with the sensors. Two way communication is required to set up the data formatting and state of the sensor modules. The two common options for this interface are the I^2C and SPI. I^2C is more complicated, but allows for fewer pins to accomplish the same task. The SPI protocol uses more communication lines but can operate at a much greater speed. We used the I^2C protocol to communicate with the sensors because we did not need the increased speed of the SPI configuration.

a) Accelerometer: The accelerometer is polled with the FPGA over a I^2C bus. We wrote a basic implementation of the I^2C protocol that does not support clock stretching or multi byte transfers. We left out the clock stretching from the specification because devices we communicate with are not spaced across clock domains and we will not access them faster than they can supply data. We created a large finite state machine that will initialize the accelerometer, then systematically poll all of the registers in the device for the X-Y-Z axis data. Completion means data from all three axis can be delivered by the sensor module. This was achieved in the fall semester for the laboratory prototype.

b) Gyroscope: The gyroscope is polled by the FPGA. The gyroscope driver is identical to the Accelerometer, with the exception of different addresses for the all of the data registers. The gyroscope driver was completed in time for the laboratory prototype.

2) Motor Control Algorithm: Sensor data will be filtered and put to work in this section of the controller. The actual control logic for the assistance algorithm is all located within this section. The result of the motor control algorithm will be directly translated into a PWM duty cycle.

a) Inclination Sensing: The control algorithm interprets the accelerometer and gyroscope data to determine the orientation and acceleration of the system. The orientation data is used to determine the slope of terrain the rider is traversing and increase the PWM input for steeper hills. The system will also turn off assist for downhills.

b) Biometric Feedback: Our design is utilizes heart rate as an input to our control algorithm. The data is obtained from a wireless off the shelf heart rate monitor. This data comes into the FPGA through an ANT+ transceiver module. To accomplish this, we have created an instance of the NIOS II soft processor and a C program to connect to and initialize the Garmin heart rate strap. The assistance algorithm contains a number of experimentally derived constants. These constants were found through the ride testing of our highly trained test pilots. Sarcasm intended.

F. Power Distribution Hub

This component of the project is the means to supply power to the low voltage components. These include the safety system, FPGA, and USB port. The power distribution hub is comprised of the buck regulator and the transistor switches. For the laboratory prototype, breadboard proofs of these components were developed and tested for viability.

- Total Estimated Time: 50 Hours
- Assignee: David
- Deliverable: A buck regulator circuit capable of safely and efficiently stepping down 36, or 12 volts to 5 volts, equipped with a USB charger for a phone. Also included are a set of transistor switches.
- David's Actual Time: 33 Hours

1) Buck Voltage Regulator: The buck regulator selected was the LM2676-5 as it was able to convert the 36V down to 5V and provide 3A at its load. This seemed to be a good match, but in actuality, it did not hold up. A drop of that magnitude was unrealistic in one step and the breadboard layout led to further issues which resulted in multiple burnouts of the regulator including a burnout during the End of Term Presentation. Fortunately, a contingency was in place and was able to quickly replace the switching regulator in a matter of minutes. The linear regulator ran off a 9V battery to avoid overheating.

There was difficulty in detecting what was wrong with the buck regulator because it would work for hours without issue. It provided a solid output at 5.02V under different loads and was able to be turned on and off without issue. One regulator lasted for over a month before burning out. This just emphasized the importance of testing.

To fulfill the requirements of the feature set, a female USB port was included on the regulator breadboard. Only the 5V and ground pins of the port were active as the remaining two data pins are not used to charge mobile devices.

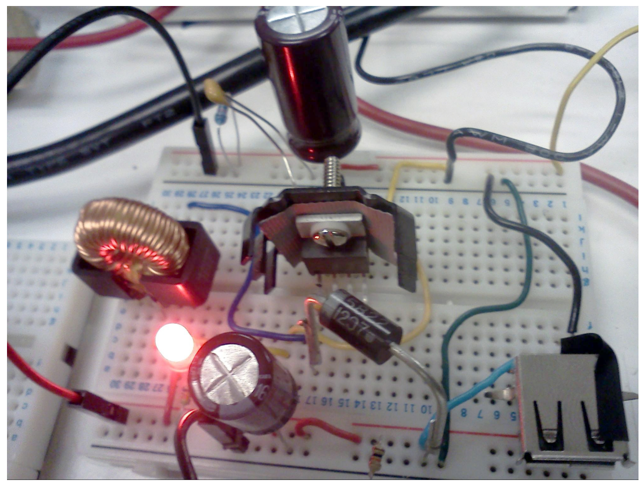


Fig. 13: Buck Regulator

2) Transistor Switches: At the time of the laboratory prototype, Darlington configuration transistor switches were selected as the means to provide power to the safety lighting while being controlled by the FPGA. TIP120s were selected because they could support the current required by the loads. The transistors were mounted on a breadboard with a voltage divider at the base. These transistors worked without issue for the laboratory prototype and would go on to be used in the deployable prototype. Extra Darlington transistors as well as other types of transistors, including MOSFETs, were held to the side as mitigation. The mitigation transistors were decided on during the initial research phase and took little time to set up.

G. Safety System

The safety systems consist of lighting and audible alert portions. These systems are intended to increase the rider's visibility and perceptibility.

- Total Estimated Time: 115 Hours
- Assignee: David, Mike

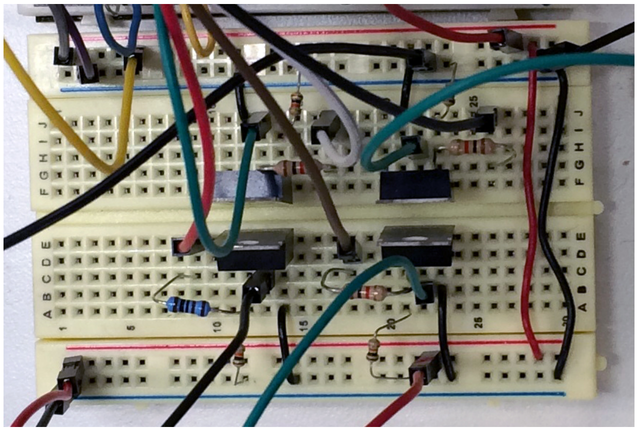


Fig. 14: Transistor Switches

- Deliverable: Functioning headlight, tail/brake light, blinkers, and horn.
- Mike's Actual Time: 22 Hours
- David's Actual Time: 22
- Ben's Actual Time: 3 Hours
- Devin's Actual Time: 3 Hours
- Total Actual Time: 50 Hours

1) Lighting: The safety lighting consists of a headlight, brake/taillight, and left and right turn signals. The brightness of these systems needed to be demonstrated in order to prove viability. The layout of each light was the same with the exception of different color of LEDs, the number of LEDs and the value of the resistors depending on the needs of the different LEDs. The lights can be viewed in Figure 15. LEDs are significantly robust under proper usage, but extras were purchased as a mitigation.

Each light received its own breadboard for the laboratory prototype to aid in the visualization of where the lights would be located on an actual bicycle. The lights could be tested by pressing some buttons which were located on a separate breadboard. These buttons will be discussed further in a section titled User Interface. By holding down any of these buttons the headlight or taillight would turn on or the left or right turn signals would blink. The brake light is activated when the brake lever is engaged. This was to show that the user could control the lighting system manually with physical switches.

The deployable prototype would later adopt more LEDs on the headlight and taillight and will transfer to a perfboard design to be encased in proper enclosures which will be mounted to the bicycle.

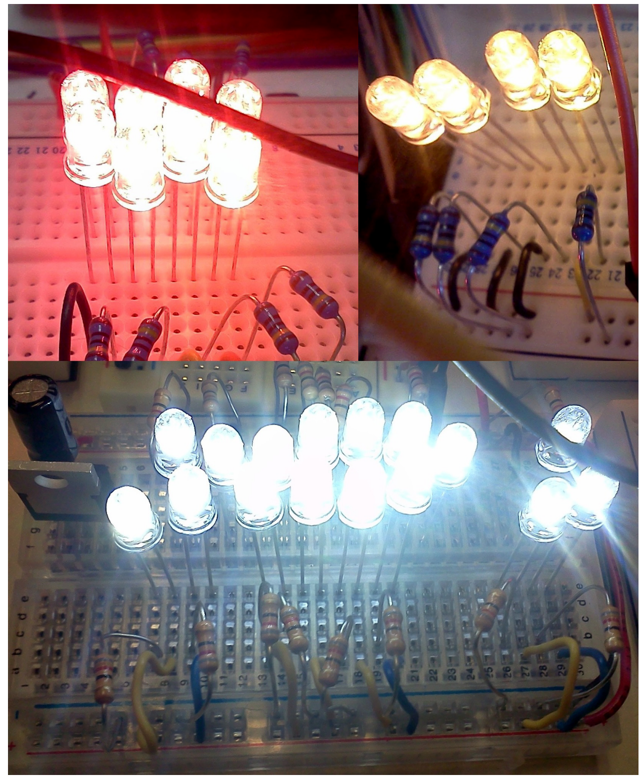


Fig. 15: Lighting Systems

2) Audible Alert: The audible alert consists of an R2R DAC and an audio amplifier IC. The DAC utilizes an R2R ladder network configuration into a summing amplifier. The summing amplifier chosen was a MC33171D due to its rail to rail operation. The output of the DAC is fed into a MC34119 audio amplifier, selected for its differential output. The differential output separates the output from ground without using extra components. The speaker selected was an 8 ohm Radioshack special, to prove the concept while developing the system indoors.

a) Audio Ramp Generator: We synthesized a Verilog module to generate a ramp signal to feed the input of the DAC module. The module increments a counter of an 8 bit register. Each of these bits is output in parallel to feed the inputs of the R2R DAC. The counter starts when a triggering signal is asserted high to the FPGA. This signal also enables the output of this module, otherwise the outputs are set to zero.

H. User Interface

The user interface for the laboratory prototype consists of the mobile application and brake levers.

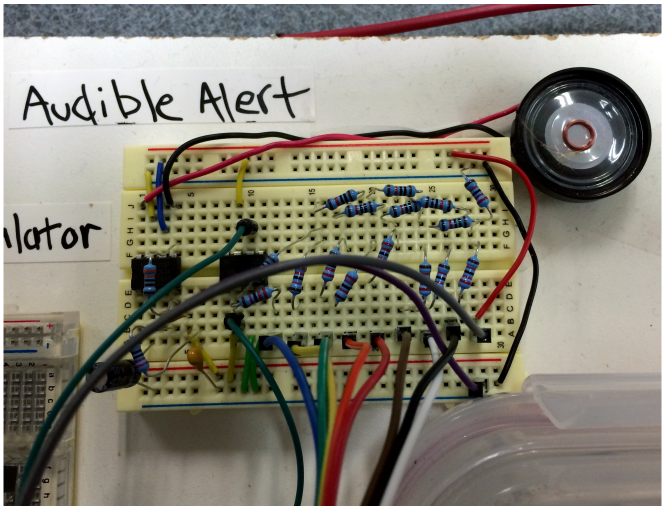


Fig. 16: Breadboard Version of Audible Alert

The mobile application is used to enter the heart rate cap and display real-time information.

- Total Estimated Time: 45 Hours
- Assignee: Ben, Devin
- Deliverable: Android Application that can communicate with the FPGA, and a brake lever that will send a signal to the FPGA when it is pressed.
- Devin's Actual Time: 37 Hours
- Mike's Actual Time: 12
- Total Actual Time: 49 Hours

1) Mobile Application: For the first semester of our project, it was important to have a way to enter a heart rate cap for our assistance algorithm. We decided to accomplish this with an Android application that can communicate with the FPGA over Bluetooth.

a) Researching Android Development: In order to create an Android application to communicate over Bluetooth and display data, we first had to do some research. We did this by reading a beginner's Android development book as well as going through the lessons on Android's developers site. Android training. This went fairly quickly, and gave us enough knowledge of Java and XML to get started on the application.

b) Creating Android Application: Creating the Android application started with downloading and installing the Android SDK. The next step was creating a new project using a Samsung Galaxy S3 smart-phone as the target device. Permissions had to be set for Bluetooth communication and data logging. Building the structure and layout of the application was done using XML, and the features of the buttons or "active areas" were created in the Java code.

c) Creating Verilog Bluetooth Interface: In order for the cell phone to be able to communicate with the FPGA, we interfaced to a Bluetooth transceiver on the FPGA. We had Bluetooth Mate Gold from Sparkfun which uses an RN-41 module for the transmissions. We had to connect with the cell phone and transfer data to and from the FPGA. We found an open source UART module on Open Cores written in Verilog. We altered it to work with the transceivers Baud rate and used it to send and receive data.

2) Physical Interface: A physical interface will be needed to communicate information to the user as well as gather user input.

a) Brake Lever: A signal is sent to the FPGA when the brakes are activated so the motor will cease to supply power to the wheel. We purchased a set of brake levers with a switch that opens when the break is activated. This was installed on the handlebars of the bike and to the front brake.

I. *Laboratory Demonstration Platform*

After we decided on a design idea, we needed to be able to safely demonstrate all of our features in a laboratory setting. Starting with the entire system implemented on a bicycle would not only be dangerous, but would make it difficult to create the individual features from scratch. We decided to create a platform that could be mounted on a desk and be able to demonstrate all of our features in the lab.

- Total Estimated Time: We completely overlooked these tasks.
- Assignee: Ben, Devin
- Deliverable: Table top platform that successfully, and demonstrates all of our features as safely as possible.
- Mike's Actual Time: 63 Hours
- David's Actual Time: 45 Hours
- Ben's Actual Time: 40 Hours
- Devin's Actual Time: 32 Hours
- Total Actual Time: 180 Hours

1) Building Prototype Platform: To create this safe demonstration platform, we first found a rear triangle of a bicycle that could hold our motor. It was broken, so we used a welding compound to increase the integrity of the frame and prevent any further damage. We also mounted all of the power systems, ESC, and safety systems to the board. After everything was mounted, we labeled all of the sub systems and made sure they were visible. We also organized all of the wires that were running across the board.

2) Testing Prototype Platform: Once the demo board was created, we needed to test all of the systems. This brought to our attention a few problems, such as signal wires being too close to each other and unprotected.

J. Administrative

Throughout the semester we had team meetings to keep the group's work efforts synchronized. We also submitted weekly reports to our project advisor detailing the work we had accomplished as well as our project's status. The team spent a very large amount of time creating documents, from our societal problem statement to the end of term documentation. The instructor commented in the beginning of the class that the administrative task is the most common to underestimate; we were not the exception to this rule.

- Assignee: Ben, Devin, Mike, David
- Deliverable: Weekly reports, presentations, and Documentation
- Mike's Time: 106 Hours
- David's Time: 120 Hours
- Ben's Time: 163 Hours
- Devin's Time: 144 Hours
- Total Actual Time: 517 Hours

K. Risk Assessment

No project comes without taking on risks, at least any projects worth doing. We had several risky elements during the development of our laboratory prototype, which we explore below.

ESC and Motor (High Risk, High Impact:

a) Risk: There was a risk that the motor would not interface with our electronic speed controller (ESC). There was also a risk that the ESC would not accept a PWM input from the FPGA as a throttle.

b) Mitigation: Different ESCs and motors would need to be tried in various combinations to find a working system. If this could not be achieved, this project would not be feasible and a new one would need to be selected.

I^2C Interface (High Risk, High Impact):

c) Risk: The I^2C interface was a risky undertaking due to our limited exposure with complex System Verilog applications. Due to the required rapid development schedule, there was also a major risk that we would not get it written in time.

d) Mitigation: We could use an Arduino for the polling of sensor data and parallel bus the data out into the FPGA; additionally, we could switch the project in entirety to Arduino.

Heart Rate (High Risk, High Impact):

e) Risk: Our Biometric feedback requires the user's heart rate. Not being able to obtain heart rate data from the user therefore is a high impact risk. We intended on using an off-the-shelf wireless heart rate monitor, so getting the correct data into the FPGA requires a wireless module compatible with the protocol of the heart-rate monitor, as well as a Verilog module to properly configure and obtain the data from the wireless module.

f) Mitigation: There are multiple wireless adapters in existence and a few different ones were ordered. An Arduino can also be used to obtain the wireless heart-rate data and process it, and then output parallel bytes to our FPGA for implementation. We utilized this mitigation during the development of our laboratory prototype to spread the Verilog development between design phases.

Voltage Regulator (Medium Risk, Low Impact):

g) Risk: The buck converters have more components to them than a linear regulator thus more possibilities for failure. This results in a decent chance for failure by burning out.

h) Mitigation: Having backup bucks on hand will allow for a quick and clean change out if one of the bucks burn out. Additionally, linear regulators were set up and put into place because they are much more robust than the bucks and will yield the same output.

Switching System (Low Risk, Low Impact):

i) Risk: There is a chance for switches to burn out if too much current is pushed through them or if too much voltage is dissipated by them. The switches may also draw too much current from the FPGA if the collector current is too large.

j) Mitigation: Backup switches were purchased for easy replacement in case of burn outs. If too much current is drawn, MOSFETs were also purchased as a backup because they draw much less current.

Power (Medium Risk, Low Impact):

k) Risk: A power supply was purchased from Amazon with little information regarding its reliability. It is also not uncommon for power supplies to burn out when under frequently changing outputs.

l) Mitigation: Batteries were purchased and a power supply from the school was provided and that could have been used as a temporary replacement.

Cell Phone (High Risk, Medium Impact):

m) Risk: There was a fairly large risk that the Android application would not work as intended. None of the team members have had much experience with Android development and the Bluetooth communication was a complex undertaking.

n) Mitigation: We would need to divert all of the user input to a physical interface in the case of a failure. We would remove any real time data display or implement an LCD screen.

Lighting (Low Risk, Low Impact):

o) Risk: There was a small possibility for the LEDs to burn out.

p) Mitigation: Extra sets of LEDs were purchased for each light in the event any of them burned out. If these still didn't work, a flashlight could have been used to demonstrate the switching and FPGA control.

Sound (Low Risk, Low Impact):

q) Risk: There was a small risk that we would not get the DAC to work with an FPGA, so a sound would not be able to be generated, or that the amplifier design would not drive a load sufficiently.

r) Mitigation: We had the Analog Discovery on hand in case the DAC wouldn't work. If the amplifier didn't drive our speaker, we could replace it with a store bought amplifier such as a miniature guitar amplifier. If everything failed, we could use a bell.

V. Creation of Deployable Prototype

Once we had proved that we were capable of implementing our features in the lab, we moved onto integrating all of the subsystems to create a deployable prototype that worked as a complete system to prove our design idea. Integration testing uncovered a substantial number of problems in most systems. Resolving the issues uncovered during testing was the largest expenditure of time for the creation of the deployable prototype. The experience gained by the group in the mitigation of these problems is invaluable, we were able to work as a team to overcome significant technical problems. The combined efforts of the group ensured we were able to deliver the prototype fully functional and on time. The sections to follow document the tasks to overcome these challenges, funding, and the sheer work effort required to deliver the prototype.

Fig. 17: Instructor Advisement

A. Funding

There were many times in which we were not able to correctly predict our total spending. A majority of this came from not having a clear understanding of what we actually needed. This came in part from not knowing what methods we would ultimately settle on to achieve each feature. For example, we were originally going to use USB as the method to connect the external components, but instead settled on connectors that were more robust. Additionally, we had a series of catastrophic failures including the motor falling out of the frame, an ESC smoking, as well as an FPGA development board failing due to being connected to greatly dissimilar grounds. These problems impacted our deployable prototype expenditures greatly.

Our expenses from Spring 2014 can be seen in Table V.

B. Schedule and Milestones

The spring semester was expected to consist of documentation and system integration. This would have been true, but we ran into an unforeseen issue involving the control of the motor.

TABLE V
SPRING SEMESTER INTERNAL FUNDING PROPOSAL

Purpose	Item	Supplier	Cost
Redundant Control System	Dynastream AP2	Digikey	50
Redundant Control System	IMU Digital Combo	Sparkfun	65
Redundant Control System	Bluetooth Mate Silver	Sparkfun	45
Get Parts from SF	Fuel	AMPM	40
Get Parts from SF	Toll	California	5
Get Parts from SF	Parking	Some Lady	10
Integrating Motor	Misc Bike Parts	Bike Emporium	40
Rewiring	Additional Wiring for ESC	Lyen	24
Bike Throttle	Hand Throttle for Testing	Lyen	20
Battery	Batteries	Amazon	94
Misc Parts	Misc Parts	Metro	28
Battery Box	Battery Box	CSUS Bookstore	22
Misc Parts	Misc Parts	Home Depot	16
Misc Parts	Misc Parts	Metro	84
Misc Parts	Misc Parts	Home Depot	12
Misc Parts	Misc Parts	Metro	24
Replacement FPGA	DE0Nano	Newark.com	94
Enclosures	2 Spare FPGA Enclosures	Adafruit	40
Misc Parts	Term Block	Digikey	12
Power Hub	(Buck+Components)x2	Digikey	25
Mirrored Rear Wheel	DT Swiss EX500 x2	Bicycle Emporium	120
Lacing Rim to Hub	DT Swiss Champion 2.0mm x32	Bicycle Emporium	25
Repair FPGAs	Voltage Regulators	Digikey	20
Holding Motor in Frame	Torque Arms	Ebikes SF	58.79
Misc Parts	Misc	Metro	104.47
Backup Motor	MAC High Speed Hub Motor	Lyen	325.91
Backup ESC	2x Lyen 9-FET Mark II	Lyen	200.56
	Deployable Prototype	Total	1604.73
	Total Project	Total	2948.73

TABLE VI
SPRING SEMESTER EXTERNAL FUNDING PROPOSAL

Purpose	Item	Paid By	Cost
Backup Motor	MAC High Speed Hub Motor	Grant	325.91
Backup ESC	2x Lyen 9-FET Mark II	Grant	200.56
Refreshments		Tatro	50
		Total	576.47

During the development of the laboratory prototype, we had two major milestones that coincided with hardware demonstrations that determined the build schedule. The first of these was the midterm project review, where we needed to demonstrate the functionality of our project as a device that is field deployable. The second demonstration is the deployable prototype demonstration where we demonstrated a complete system, fully integrated and tested.

The process of converting our laboratory prototype to a deployable system required removing the motor from the lab prototype and lacing it into a wheel that can be attached to a bicycle. We then needed a way to power our systems, so we ordered sealed lead acid batteries due to their robust nature and ease of use. To facilitate the testing process, we purchased a twist grip hand throttle from Edward Lyen of Lyen. Having this throttle would prove extremely useful in testing the bike while the control system was developed in parallel. Attaching the motor to the bike, laced into the wheel and with tire attached, we wired up the ESC to the motor as we had done all the previous semester. This configuration provided no power and no torque to the wheel, a finger could be used to stop it from spinning full speed as set by the hand throttle. After researching wiring configurations, we changed the phase wire pairings to green yellows matching, and blue with blue. This configuration has proven itself and the motor no longer has its squelching issue we throughout the laboratory prototype build. The ESC also no longer gets warm to the touch as it did the first semester.

Once the bike was in a testable configuration as seen in Figure 18 we set to prove the validity of our laboratory prototype's implementation of the control algorithm. We came into this section thinking the throttle input, varying the duty cycle of a PWM signal would work similarly to the way a gas pedal in a car does, more throttle = more assistance. We found out on our first set of ride testing that the throttle sets speed much in the way a cruise control system does in a car.

Once the bike was in a testable configuration as seen in Figure 18 we set to prove the validity of our laboratory prototype's implementation of the control algorithm. We came into this section thinking the throttle input, varying the duty cycle of a PWM signal would work similarly to the way a gas pedal in a car does, more throttle = more assistance. We found out on our first set of ride testing that the throttle sets speed much in the way a cruise control system does in a car.

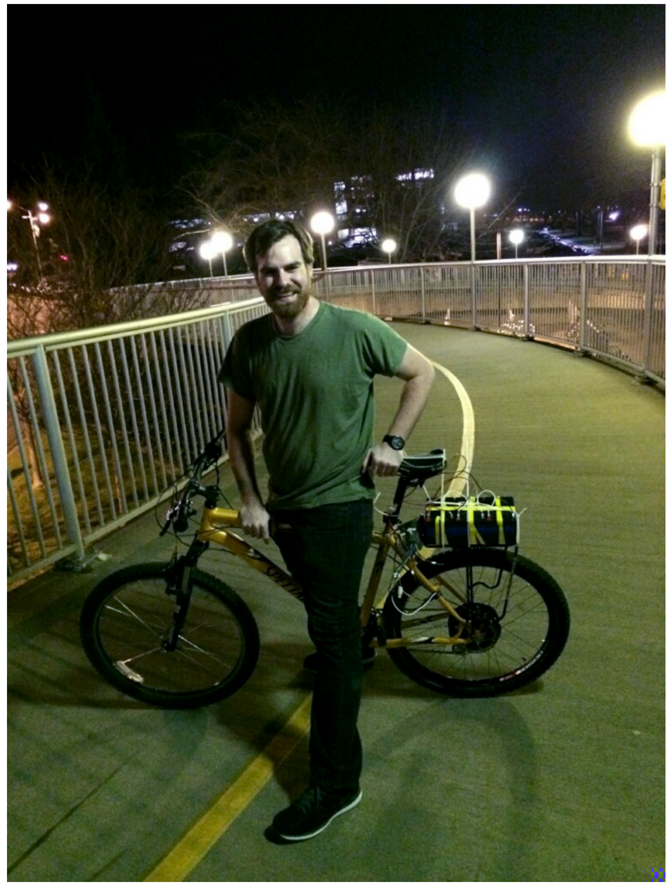


Fig. 18: Initial Laboratory Prototype

Through experimentation, we determined that current dictates assistance, and changed gears to develop an assistance system that controls current by throttle manipulation of our ESC. We utilized a shunt resistor already on board our ESC and built a differential amplifier with a low pass filter to reduce noise from the circuit. The output of this filter was fed into an ADC on our FGPA development board.

While developing the algorithm to control current, we utilized simulated load testing to try and simulate the bike under actual load. The load testing consisted of activating a disc brake attached to the motor and bicycle. While undergoing this testing, the motor spun out of the dropouts holding it in place to the bike. We purchased and installed torque arms to prevent this from happening again, and reinstalled the motor and reconnected it to the ESC. The ESC promptly smoked, indicating damage that had occurred during the previous incident.

Ben and Mike traveled to San Francisco to meet with our suppliers one on one, purchasing two new ESCs and a new motor to make sure we could complete the project. Ilia of E-Bikes San Francisco tested our motor to make sure it was still functional and not damaged. We reinstalled the motor and our new ESC. Systems were fully online again. With working systems, we could continue our development of the project. This phase of our prototype can be seen in Figure 19.

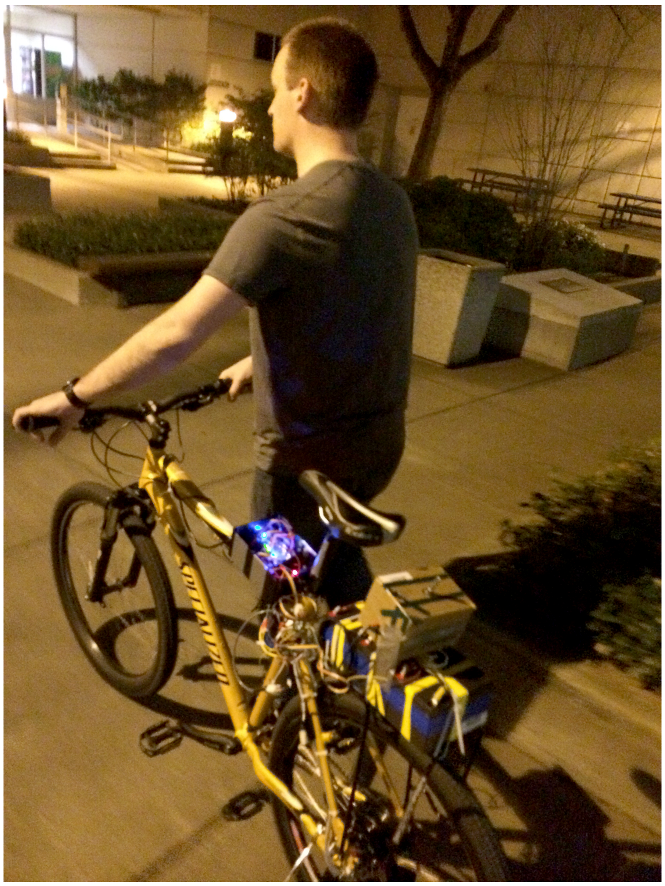


Fig. 19: Second Implementation of Laboratory Prototype Featuring FPGA

Ben and Devin completed a current control algorithm on the bike in time for the Midterm Project Review, proving the concept that we can control assistance to the rider. Having successfully demonstrated our concept of assistance, it was time to start integrating all these systems together in a deployable fashion.

David and Mike broke off to start working on encasements, creating a reinforced battery box to safely house the batteries, a control systems box to integrate all the systems together, a taillight box with brake light and blinkers, and a headlight box with headlight and blinkers, as well as the switches to activate the safety systems. These system can be seen in their installed form in Figure 20.

Fig. 20: Deployable Prototype Near Completion

During the integration phase the step down buck regulator was being tested, we tapped into the closest 12 volt cell of our battery pack. The power supply worked so we connected the power supply to our FPGA and then connected the grounds between the ESC and the power supply. The FPGA sparked violently. The ground that we pulled from the battery for the power supply was 24V higher than the ground for the rest of the system, ESC included. We corrected this issue by using the same ground for all systems.

While David and Mike were constructing the hardware for the deployable prototype Devin and Ben were chasing around a bug in the control firmware that would cause the system to oscillate in certain circumstances. We thought this oscillation was the result of incorrect tuning constants in the control system. After a month of experimentally testing different constants we found an error in our Verilog HDL which was causing a division to be done as an unsigned integer instead of signed. We changed the style of this code which fixed the problem. Once the control system was behaving properly we begun to integrate the FPGA based control system with the housings.

With all the systems integrated, the current control algorithm in place, our deployable prototype had been completed.

TABLE VIII
DEPLOYABLE PROTOTYPE SEMESTER HOURS

Feature	Task Category	Ben	David	Mike	Devin	Group
Smart Assistance	FPGA Enclosure		16	19		35
	Signal Processing/Data Acquiring	40				40
	Current Control	79			64	143
	EPCS/.jic Configuration	20				20
	Assistance Algorithm	18			16	34
	Heart Rate Communications	24			8	32
	Researching/Ordering Batteries			8		8
	Testing ESC			6		6
	ADC Driver	5				5
	ADC Amp/Filter			8		8
Power Hub and Regulator	Design and Obtain Parts		3			3
	Build		8			8
	Testing		10			10
Safety System	Lighting Testing		10	3		13
	Building Lights		34			34
	DAC/Sound			11		11
	Research/Acquiring Parts		9			9
Mobile Application	Research/Learning Android dev				6	6
	Android Application				18	18
	Data logging				8	8
	Verilog State Machine				4	4
	Testing	6			8	14
Deployable Prototype	Mounting Drive Train	48	37	100	37	222
	Riding Bicycle	8		14	13	35
	Watching us Ride the Bicycle		10			10
Administrative	Documentation	41	54	49	38	182
	Weekly Reports	12	15	18	15	60
	Bulletin Board/Hive Website/Logo		5			5
	Presentation	10	12	22	13	57
Totals		311	223	258	248	1040

TABLE VII
FALL SCHEDULE AND MILESTONES

Date	Milestone
January 27th	Beginning of Spring Semester
February 10th	Test Plan
March 3rd	Market Review
March 17th	Midterm Progress Review
April 7th	Feature Presentations
April 21st	Deployable Prototype Review
May 5th	End of Project Documentation
May 12th	Deployable Prototype Presentation

C. Breakdown of Work Required

This section contains the work packages that comprise our design of a smart electrically assisted bicycle. The bicycle consists of a motor system, user interface, FPGA control system, power system, and safety features. These elements are necessary for a complete commuting package, which will enable a user to consider cycling as a more viable means of transportation, thereby increasing the level of activity of the user. Each of the work packages has its own risks and work required to complete which will be detailed in the section to follow. Unfortunately, for most of these tasks, we never properly estimated the amount of time required so that statistic has been omitted from this section. The total hours for the deployable prototype can be seen in Table VIII.

D. Motor System

The motor and electronic speed controller are the heart of the project. In conjunction with the speed controller, the motor facilitates the transfer of power to a wheel of the bicycle providing assistance to the rider. The motor/wheel consists of a hub motor as well as the components of the rear wheel. This will be the system which actually provides the assistance for the user based on the output of the FPGA control system.

- Assignee: Mike
- Deliverable: Operational Hub Motor and Related Electronic Controller
- Mike's Actual Time:4 Hours
- Ben's Actual Time: 4 Hours
- Total Actual Time: 8 Hours

1) Hub Motor: The rear wheel will contain a motorized hub in order to assist with the propulsion of the bicycle. Getting the motor to turn is paramount to the operation of our project. Requirements for the hub motor to be successfully implemented are the accompanying wheel elements, an electronic speed controller controlling its power, and a bicycle to be implemented in. This is an incredibly high risk deliverable, if something here doesn't work as intended, it will be detrimental to our entire project.

a) Integration with ESC: The hub motor did not function as we had expected once we transfered systems to the bike from our stationary platform. We determined the wiring configuration of matching colors was incorrect and found the correct permutation to correct this issue. The wiring for both the Hall sensors as well as the power delivering phase wires needed to be reconfigured to the correct permutation. The motor performs as expected and as intended in this configuration, we haven't had any of the same sporadic instances where the motor wouldn't turn on. Additionally, the ESC no longer runs hot as it had during the laboratory prototype phase.

2) Wheel Assembly: Hub motor to be laced to a rim by experienced wheel builder. Rim drilling, spoke length and freewheel are to be determined by hub selection. This will be complete when the wheel spins correctly with the hub motor installed. Ben has built wheels before and expects no risk of access to materials or tooling.

E. Smart Assistance

The assistance algorithm is implemented within the FPGA and decides how much motor power the user requires at any given time. The algorithm uses the inclination of the system, the user's heart rate and their desired rate as set in the cell phone application. This task required a number of hours, particularly for tuning the response of the algorithm in the integration phase.

- Assignee: Ben, Devin
- Deliverable: Control system to provide assistance with electric motor, based on user heart rate and inclination. Thoroughly tested.
- Mike's Actual Time:14 Hours
- Ben's Actual Time: 240 Hours
- Devin's Actual Time:160 Hours
- Total Actual Time: 414 Hours

1) Current Control: This was an enormous task that was not expected during the first semester. Our entire project design is predicated on providing assistance to a rider based on biometric and terrain feedback. Going into this semester, we had a crude algorithm in place to supply an output the ESC that we though would control assistance. In fact, this controlled speed. So when we thought the rider needed more assistance, the bike would simply draw as much current as it could to speed up. We wanted to supply the rider with assistance, or torque, independent of their speed. We determined current through the motor to be closely related to the amount of assistance, or torque, supplied by the motor. We were able to infer the amount of current drawn by the motor and use that measurement to roughly provide the amount of torque required by the assistance algorithm. This is not a particularly precise method of torque control but any error is below the perception threshold of the rider. The completion of the CurrentControl module allowed us to proceed with the development of the assistance algorithm.

2) Tuning the Assistance Algorithm: The assistance algorithm is simple proportional controller. The set point from the cell phone is compared against the measured heart rate to produce an assistance signal that is scaled and added a number reported from the IMU angle resolution module. Because we did not have complete information on the motor's electrical characteristics nor felt the need to develop a quality model for the system, we experimentally derived the tuning constants for the system. The experiments took two forms: testing in the laboratory with a logic analyzer and simulated load, once we had a set of constants we felt would provide the experience we were looking for we took the bike out on the trail to experience the algorithm in action. The basic concerns of tuning were to provide the fastest transient response while not having perceptible oscillation for the user. These tasks were accomplished in time to deliver the deployable prototype with the user experience we envisioned.

3) Testing the Algorithms: This consisted of a number of real world trial runs to confirm the user experience of the system in operation. We would try a number of tuning parameters and observe the effect on the ride. This phase of testing ensured we had developed a system that provided the assist functionality we had envisioned during the design phase of the first semester.

4) Creating Configuration Image: The FPGA, along with the NIOSII program needed to be stored on non-volatile memory so the bike did not have to be reprogrammed after losing power. All of the storage inside the Cyclone IV device is volatile so a EEProm flash device is used to store a configuration image for the FPGA. The NIOSII soft processor contains a special module which allows it to store data on top of the FPGA image for it's configuration.

F. Power Distribution Hub

This component of project is the means to supply power to the low voltage components. These include the safety system, FPGA, and USB port. The power distribution hub is comprised of the buck regulator and the transistor switches. For the deployable prototype, perfboarded components were developed and tested for viability.

- Assignee: David
- Deliverable: A compact buck regulator circuit capable of safely and efficiently stepping down 36, or 12 volts to 5 volts, equipped with a USB charger for a cell phone. Includes a set of transistor switches. Fully tested.
- David's Actual Time: 21 Hours
- Total Actual Time: 21 Hours

1) Buck Voltage Regulator: The regulator is now solidly fixed on a perfboard. and has much greater efficiency since the laboratory prototype. The input has been changed from the 36 volts of the supply for the motor, to just 12 volts of one of the main batteries. Because of this change to a lower input voltage, the circuit had to be redesigned. The same LM2676-5 IC was used, but the inductor and capacitors were changed to adhere to the new specs.

Converting to a perfboarded unit provided many benefits. These include shorter leads, more solid contacts, and condensed space. The regulator now shows increased efficiency and produces little heat. The regulator resides within the FPGA enclosure which is fixed on the bicycle frame beneath the rider.

In the case of a failure, the regulator would have been exchanged for a more durable, but vastly less efficient linear regulator. Additionally, a 9V battery pack consisting of four 9V batteries in parallel was built as a backup source if the 12V source was too much for the regulator to handle. The larger the difference between the input and output voltage of a regulator, the harder the regulator will have to work. This would cause greater losses through heat and increase the chances of the regulator burning out.

2) Darlington Configuration Transistor Switches: The Darlington configuration transistor switches required no redesign and only needed to be converted to a perfboard. The layout was determined to prevent the emitters of the transistors to touch and to allow for easy access to all pins as the FPGA and I/O board will need to interact with this unit. This unit is contained within the FPGA enclosure.

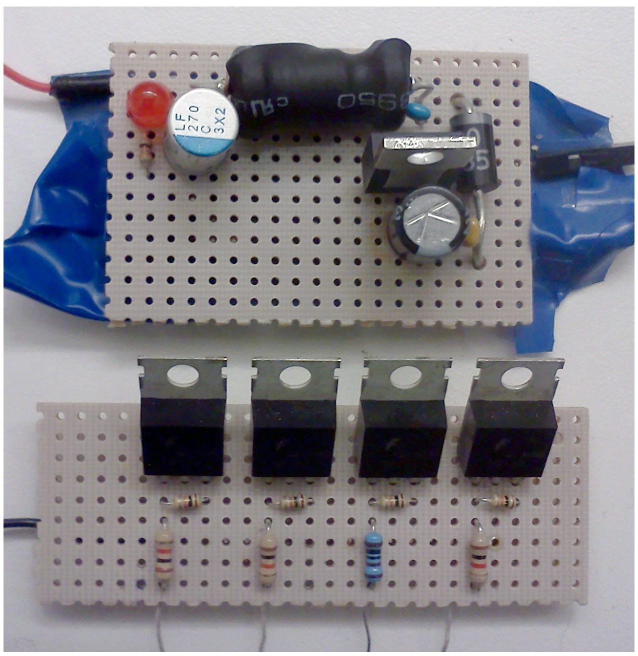


Fig. 21: Top: Buck Regulator, Bottom: Transistor Switches

G. Safety System

The purpose of the safety system is to aid in the prevention of collision through increased visibility and perceptibility. This system consists of lighting and an audible alert.

- Assignee: David
- Deliverable: Headlight, tail/brake light, blinkers and horn all completely packaged and tested.
- Mike's Actual Time: 18 Hours
- David's Actual Time: 53 Hours
- Total Actual Time: 71 Hours

1) Lighting: The lights have come a long way since the laboratory prototype. All the lights are now mounted onto the bicycle and enclosed within one of two lighting enclosures. There is an enclosure on the front of the bike and another on the rear.

The front lighting enclosure resides on the handlebars and contains the headlight, the front left turn signal, and the front right turn signal. This enclosure also houses all the switches which control the lighting systems. The rear enclosure is bolted to the battery box. This enclosure houses the taillight, rear left turn signal, and rear right turn signal.

All the lights are no longer controlled by buttons. The headlight and taillight are now activated by flipping a rocker switch on the front enclosure. The turn signals are activated by flipping a 3-way toggle switch. One direction activates the left turn signal, the other direction activates the right turn signal, and the center mode disables both turn signals. When the turn signals are active, they blink, as to be expected. The brake light is still engaged when the left brake lever is applied.

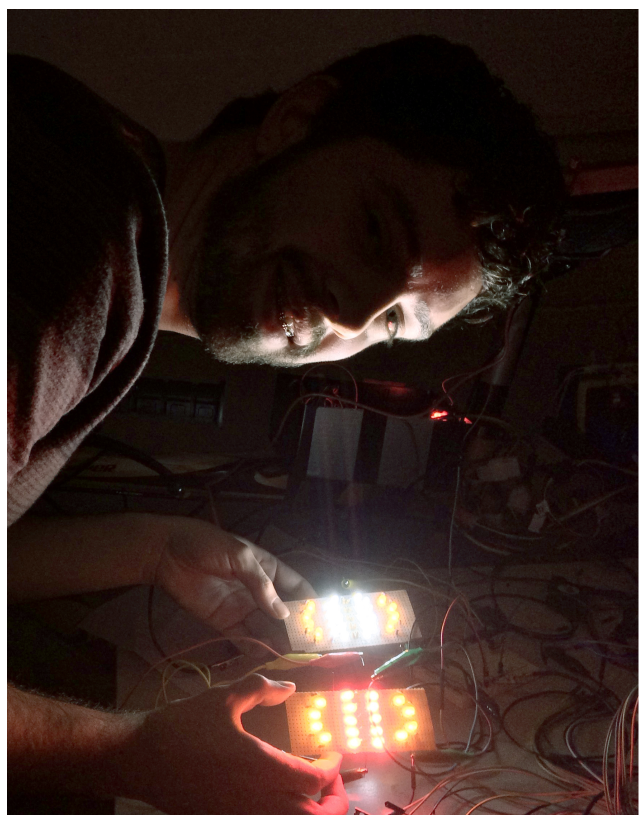


Fig. 22: Lighting Systems

2) Audible Alert: An audible safety system will alert drivers, pedestrians, and others of the presence of a rider.

a) Amplifier: Through testing it was determined that the horn would not be loud enough to alert others of the riders presence. The gain of the amplifier was increased though increasing the feedback resistance value. The value was increased until the waveform started to clip when viewed on an oscilloscope. The gain of the amplifier is sufficient now to drive the speaker to 94dBA from 3 feet.

b) Speaker: The audible alert's speaker would not be robust enough to ride with in the form it came into the deployable prototype phase. We purchased a new, larger speaker that had mounting brackets included so we could mount it on the bike. This speaker was utilized in the afore mentioned testing of the amplifier.

c) Protoboarding: The audible alert was housed on a breadboard though the development of the Laboratory prototype. To increased the ruggedness of the device we transfered it to perfboard and soldered all the components in place.

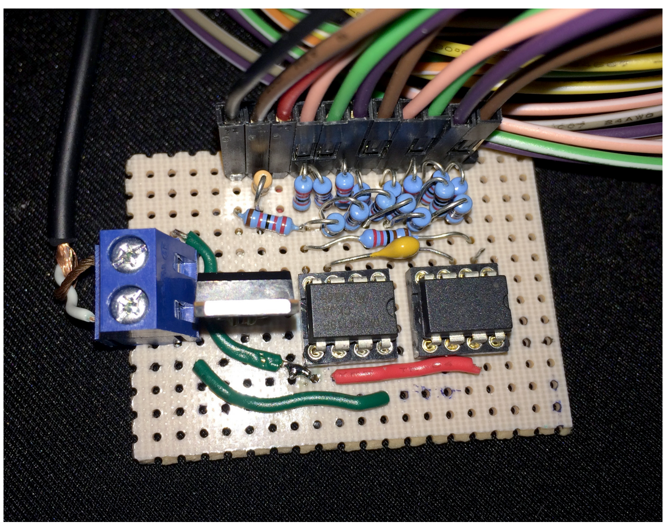


Fig. 23: Perfboard Version of Audible Alert

3) Casing and Mounting: All the lighting systems are enclosed within cases and mounted to the bicycle. The headlight enclosure is mounted on the handlebars of the bicycle. This enclosure consists of the front turns signals, the headlight, a killswitch, and the physical interface to the safety systems. The taillight enclosure is mounted to the rear of the battery box. This enclosure houses the taillight and rear turn signals. These can be viewed in Figures 24 and 25 respectively.

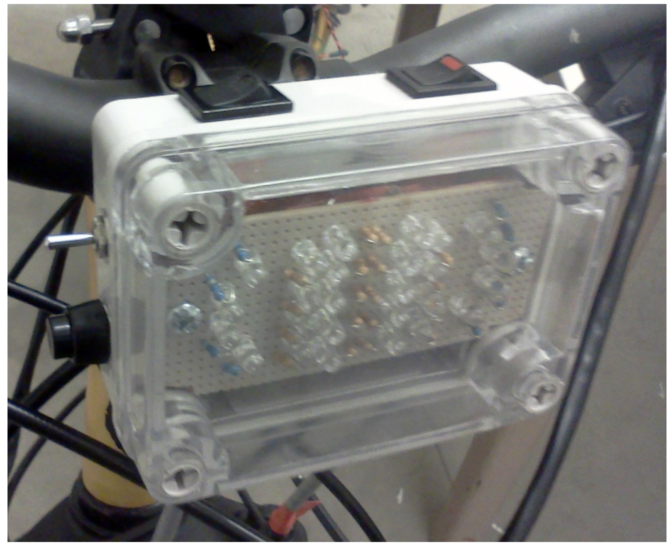


Fig. 24: Headlight Enclosure

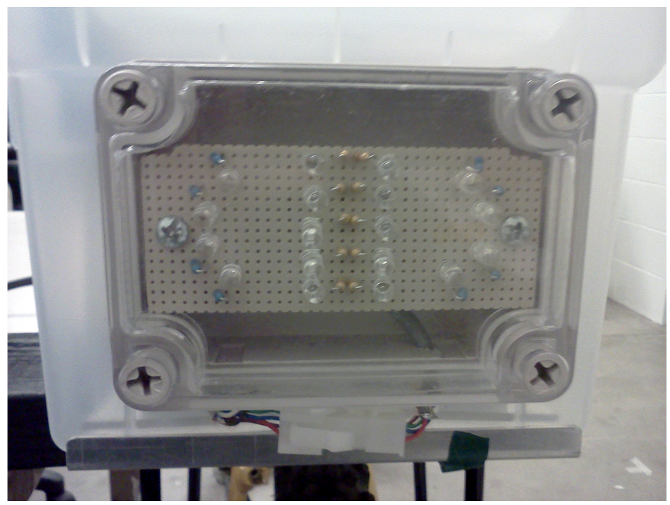


Fig. 25: Taillight Enclosure

The FPGA enclosure is the housing for the FPGA and various other components. This containts the FPGA, buck regulator, transistor switches, power bus, IO board, audio amplifier, and a killswitch for the 5V supply. The enclosure and a diagram of its layout can be seen in Figures 26 and 27 respectively.

H. User Interface

The interface will allow the rider to control different aspects of the system such as lights and turn signals. Additionally the user will be able to customize the amount of electric assist to better fit their needs and physical abilities. Physical toggles and buttons on the handlebars are used for these tasks. There will also be an option of using a han-

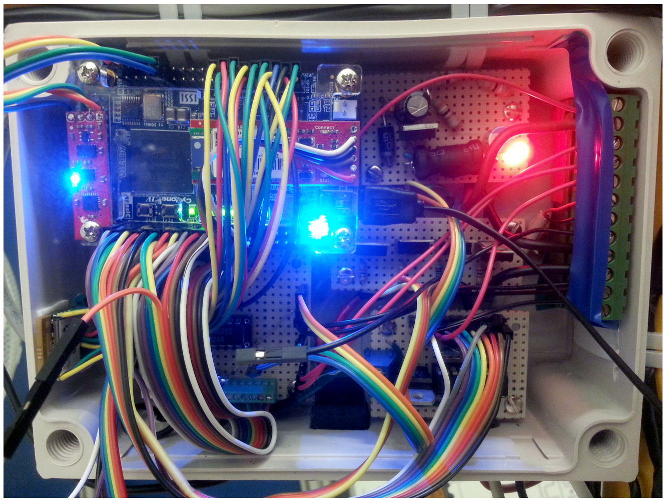


Fig. 26: FPGA Enclosure

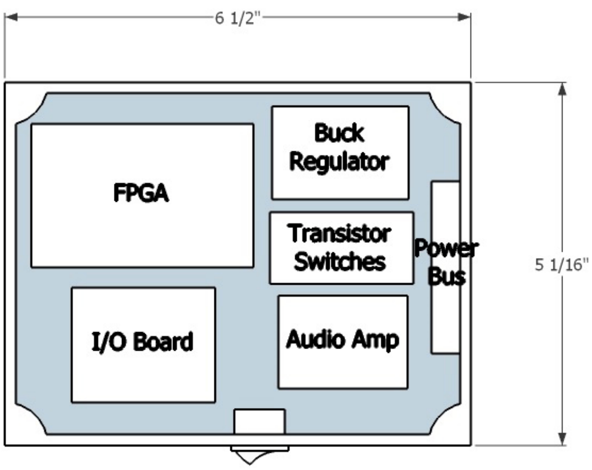


Fig. 27: Diagram of FPGA Enclosure

dlebar mounted cell phone through an application for a more detailed and intuitive experience.

- Assignee: Devin
- Deliverable: Android application capable of communicating with the FPGA and logging data received from the system.
- Ben's Actual Time: 6 Hours
- Devin's Actual Time: 44 Hours
- Total Actual Time: 50 Hours

1) Cell Phone Application: The application will allow the user to connect their smart phone to the system via Bluetooth to easily access information in real time. Through the application, the rider will be able to change settings as well as store and view collected data. The total hours required to implement the application can be seen in Table VIII

a) Troubleshooting: Since the mobile application was required to test any of our control algorithms, we were able to put it through extensive testing throughout the semester. We ran into many problems that had to fixed throughout the semester. This included altering the Verilog state machine, altering the open sourced UART module's state machines, changing the way the mobile application reads Bluetooth data, and adding other error checking for all of the buttons and entered data.

b) Data Logging: In the beginning of this project we were playing around with the idea of possibly logging the user's data for later use. We decided not to include it in our design contract because we weren't sure how difficult it would be to implement, or if it would even be useful. Once we started moving into to testing we changed our minds and decided to implement a data logging feature. To create this capability, we just had to do a little research into the Android file system, and add the functionality into the data display loop. When the loop is first started, to display the real time data to the user, it opens a new file named with a timestamps, and inserts the new data into the file, following CSV format, as the data comes to the phone.

c) Testing: Along with the normal use testing, we had to test the reliability of the connection between the cell phone and the FPGA, as well as the accuracy of the data being transmitted between the devices. This required the use of protractors for the angle, Android SDK debugger called LogCat, and Quartus' SignalTap to see inside of the FPGA. We also tested the effect of distance between the devices for the connection. The results of these tests can be seen in the testing section later in the document.

2) Physical Interface: In order to ensure the rider's safety, we have included a variety of physical switches. These switches will allow the user to control crucial functions of the bicycle in the event that the user's cell phone no longer functions. These include the electric brake lever and the switches for the safety systems.

a) Brake Lever: A signal is sent to the FPGA when the brakes are activated so the motor will cease to supply power to the wheel. We purchased a set of brake levers with a switch that opens when the break is activated. This was installed on the handlebars of the bike and to the front brake.

b) Safety System Switches: The headlight enclosure is mounted to the handlebars of the bicycle which puts it in a great location to house the switches the rider may need to use while riding. These switches consist of a rocker for the headlight/-taillight, a 3-way toggle for the turn signals, and a momentary button for the audible alert. A second rocker switch is also included on the enclosure to kill the motor.

- Assignee: Ben, Devin, Mike, David
- Deliverable: A complete and ridable smart electrically assisted bicycle.
- Mike's Actual Time: 119 Hours
- David's Actual Time: 53 Hours
- Ben's Actual Time: 48 Hours
- Devin's Actual Time: 37 Hours
- Total Actual Time: 257 Hours

3) Feature Integration: To integrate our systems together for the deployable prototype, we needed to develop robust connections and housings for the systems. We didn't want to connect wires between systems located remotely on the bike directly to the FPGA, so we developed an IO board that handled any connections to the main control systems box and the external systems.

a) Encasements: The FPGA and other control systems require a means to be housed on the bike. We selected an enclosure large enough to house these systems and devised a way to mount it to the bike using zip-ties. The systems themselves are mounted to the this box using standoffs and screws attached to a piece of plastic. This sheet of plastic is held in place with screws to the encasement itself. We drilled holes through the case to permit wires to enter it, as well as slot to be able to mount it to the bike.

b) IO board: The IO board was designed to handle connections entering the control systems box. We used screw terminals to hold these incoming wires in. The board houses all pull-down/pull-up resistors and the current control amplifier.

c) Wiring: To control the external systems, we created wiring harnesses using 6 conductor Belden cabling. We attached Molex Mini-Jr connectors to the end. These cables are routed to the lighting systems and the data header for the ESC.

4) Integration Testing: Once all of the wires had been organized, and the subsystems perfboarded, we began testing the functionality of the deployable prototype. Many of the pins on the FPGA had to be reassigned and reconnected to the correct inputs and outputs. The FPGA was now in a new position, so the IMU calculations needed to be adjusted to produce the correct inclination values. We also implemented the final safety checks in case the bicycle was on its side or flipping over backwards. We also took turns riding the system around to get a better feel for the amount of assistance our algorithm was providing. This allowed us to tune the algorithm further, ensuring the rider's experience was where we had envisioned it.

I. Administrative

Administrative tasks, particularly documentation, represent a large portion of the work hours required in the spring semester.

- Assignee: Ben, Devin, Mike David
- Deliverable: Weekly reports, presentations, and documentation.
- Mike's Actual Time: 103 Hours
- Ben's Actual Time: 71 Hours
- David's Actual Time: 96 Hours
- Devin's Actual Time: 74 Hours
- Total Actual Time: 344 Hours

1) Feature Integration: To integrate our systems together for the deployable prototype, we needed to develop robust connections and housings for the systems. We didn't want to connect wires between systems located remotely on the bike directly to the FPGA, so we developed an IO board that handled any connections to the main control systems box and the external systems.

Fig. 28: Integrated System

a) Encasements: The FPGA and other control systems require a means to be housed on the bike. We selected an enclosure large enough to house these systems and devised a way to mount it to the bike using zip-ties. The systems themselves are mounted to the this box using standoffs and screws attached to a piece of plastic. This sheet of plastic is held in place with screws to the encasement itself. We drilled holes through the case to permit wires to enter it, as well as slot to be able to mount it to the bike.

b) IO board: The IO board was designed to handle connections entering the control systems box. We used screw terminals to hold these incoming wires in. The board houses all pull-down/pull-up resistors and the current control amplifier.

c) Wiring: To control the external systems, we created wiring harnesses using 6 conductor Belden cabling. We attached Molex Mini-Jr connectors to the end. These cables are routed to the lighting systems and the data header for the ESC.

J. Risk Assessment and Mitigation

THE construction of the deployable prototype was plagued with risks the group had not foreseen. Even though these risks were not planned for, a mitigation strategy had to be implemented nonetheless. We had four large system failures that required the replacement of parts. The group was able to overcome these obstacles but at significant time and cost. The cause of the failures and the action taken to mitigate the issue are described below.

1) Current Control Issue: The electronic speed controller did not allow the FPGA to control the amount of torque provided by the hub motor. Our control system required the ability to control torque to provide the user experience we had envisioned during the design phase. We had to develop a secondary control system which converted the assistance requirement into a form that would make the ESC control torque. Further details about the implementation of the current control algorithm can be found in Section V-E1 on page 25.

2) Motor Axle Mount Failure: The manufacturer of the motor claimed torque arms are not required to mount this motor as a rear wheel. These torque arms connect to the hub motor's axle and are secured to the frame to prevent the axle from spinning. Despite the manufacturer's recommendation our motor managed to spin it's axle in the frame causing moderate damage. The frame damage was repairable and the installation of the aforementioned torque arms prevented problems as the project moved forward.

3) ESC Failure: Once we remounted the motor with the torque arms in place, we reconnected the ESC and tested the system to see if any of components were damaged. Unfortunately the motor would not spin. We could hear some stuttering, as if it were trying to move, but it was clearly not working correctly. After checking all of the connections and following some bad advice from Devin, we turned it back on only to watch the ESC let out some of its smoke. After some vulgar language, a few tears, and a baby monkey riding backwards on a pig, we decided we were in need of a new ESC and perhaps a new motor.

4) FPGA Regulator Failure: The FGPA development board's voltage regulation system was damaged by powering the board with an incorrect ground reference. The regulator was designed to handle a maximum of 16 volts and our wiring created 24 volts at the input of the regulator. Fortunately, the regulator failed spectacularly and left an open circuit, saving the FPGA. New regulators were purchased and the development board was repaired allowing the project to move forward.

K. Market Research

The market research of the second semester validated our biometric feedback as an innovative feature, and slightly changed our target market. We realized that many people were not commuting on bicycles because of the safety issues rather than the required physical effort. This lead to a stronger emphasis on our safety systems, but no real change in our design. Pricing became an obvious issue if we were to try and produce the final product ourselves, which lead to the idea of selling or leasing the design of our control system to a larger company. The US market for electric bicycle is growing rapidly, and having biometric feedback in the control system is novel idea that could prove to be lucrative.

VI. User Manual

THE user manual is intended to serve as a way for technically adept users to get a basic understanding of our system. While the system is designed to be relatively transparent for the user, the installation process has not been refined beyond the prototype level and will require significant technical skill to ensure proper if not safe operation. Topics of installation, the start up processes, and user use of the safety and cellphone systems are covered below.

A. Installation

The system's installation procedure can drastically change based on the system that it must be integrated into. There is only a single output and input required by the system to function properly. The system needs a way to control the motor, the current setup requires an analog voltage input. The system also needs a way to infer how much work the motor is actually doing, this is in the form of a shunt resistor for the ESC. Most ESCs use an analog voltage as an input for speed control, this can almost be assumed with a new controller. Few controllers will have the built in shunt resistor and might require a resistor to be installed. to do this, solder a very small resistor in series with the ESC ground, about 2 mΩ secure the measurement leads to each side of the resistor. If the user is uncomfortable working with electronics and high power circuits it is recommended that they seek the help of a professional.

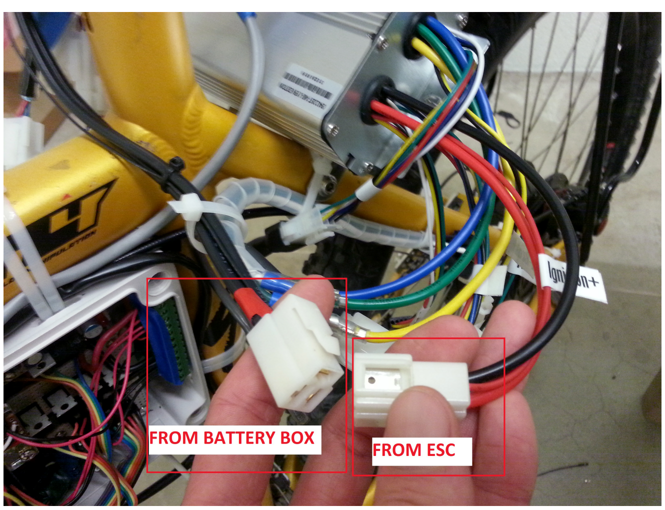


Fig. 29: Power for the Motor

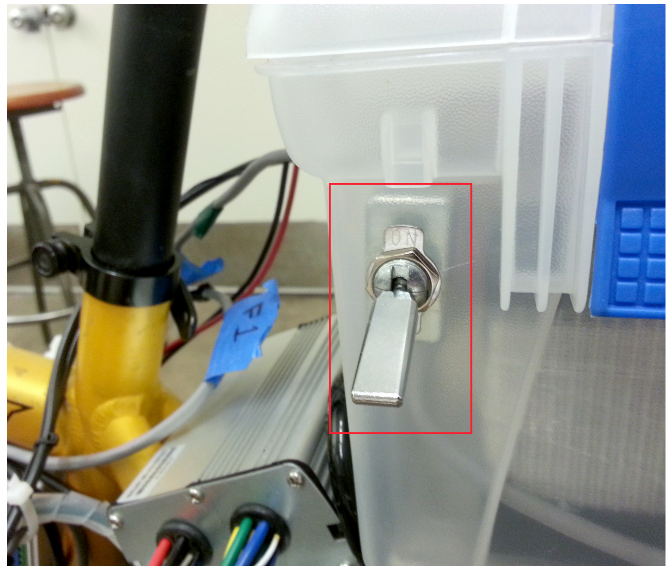


Fig. 30: Power Switch

B. Starting the System

To start Project Forward and to get yourself active, first you need to turn it on. Ensure the ESC switch is off on the top of the headlight box(#1 in Figure 32). Turn the power switch(Seen in Figure 30) on the battery box on, located on the rear of the bicycle on top of the bicycle rack. Turn the Smart Control box with the switch at the bottom of the box seen in Figure 31, located inside the main triangle of the bike. A red light should turn on inside the box as well a few other lights inside the box. The Smart Control system is now active and the motor is powered. Before starting off on your life changing journeys, make sure the safety system is fully functioning.

C. Safety System

Turn the headlight switch on the top of the headlight box. The headlight and taillight should be now be on. Squeezing the left brake lever will activate the brake light, which should appear as a more intense version of the taillight. Now test the left and right blinkers, which are activated by moving the switch located on the right side of the headlight box up, which will activate the left blinker, to the center position, and then to the bottom position, which will result in the right blinker activating. Lastly, press the button below the blinker switch to test the horn, which will be active an audible warning.

Switches and buttons seen in Figure 32

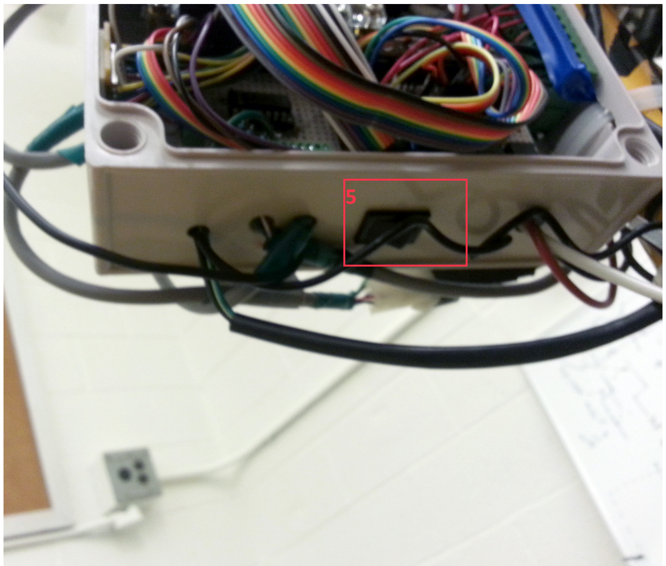


Fig. 31: FPGA Box Power Switch

1) ESC Power Enable
2) Headlight and Taillight Switch
3) Left/Right Blinker Toggle
4) Horn Button

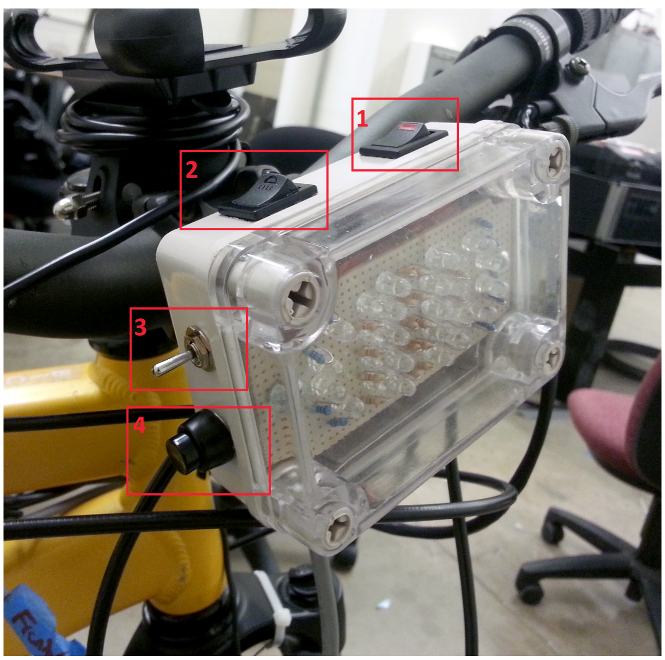


Fig. 32: Safety Switches

D. *Using Mobile Application*

You're almost ready to get out and get active. Install the Project Forward application for Android platforms. Open the software and press the connect to system button. This activates a Bluetooth link to the smart control system, allowing you to tailor the ride to your level. Enter in your desired heart rate and wheel size of your bike. Press the begin display button and this will activate the connection on the Android based mobile device. The interface can be seen in Figure 33.

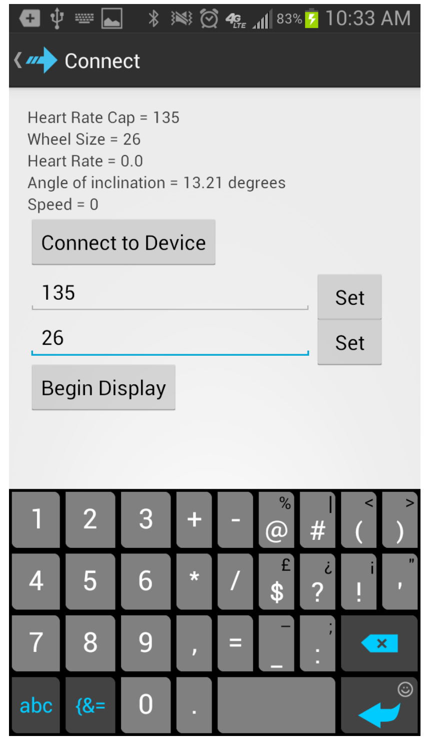


Fig. 33: Android Application Screen

E. *Riding the Bike*

Now that you're connected the phone, you're ready to ride Project Forward. Set the heart rate cap to your desired exertion cap. Attach the ANT+ heart rate strap to your chest. While holding the left brake lever, enable the motor by activating the switch on top of the headlight box. Let go of the brake and start pedaling! You should feel a boost when your heart rate exceeds the cap you set on the phone or when the bike starts going up a hill.

F. *Charging the System*

After every use, make sure to charge the system. Turn the motor switch off and the battery box off. Turn the smart control box off. Disconnect the three wire connector coming out of the battery box. Plug the charger into this cable leaving the battery box, which can be seen in Figure 34. Plug the charger into the wall. Turn the switch on the battery box on. The charger light should be blinking red while charging. The battery is fully charged when the light on the charger turns green.

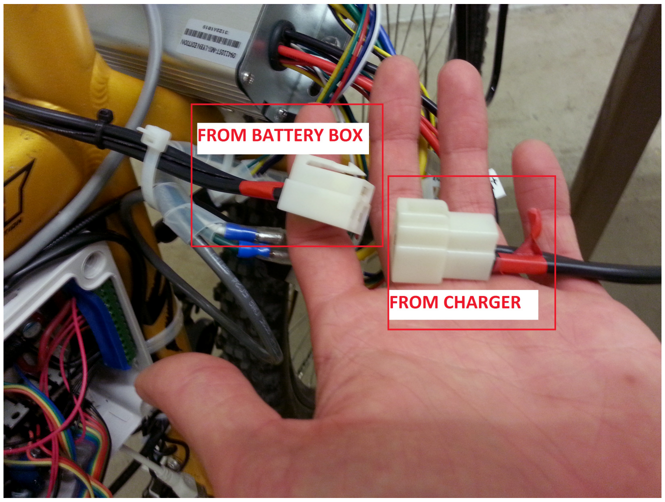


Fig. 34: Connections for Charging Battery

VII. Hardware Design

EVERY physical component of Project Forward had to be selected with due cause. From the bicycle and drive train to the lighting systems and the enclosures that housed them, every portion of the project needed to be selected, designed and implemented. This section of the document with present the hardware needed and the design process that went into the development of these parts. Some of these components were not designed from scratch, but there selection was still pertinent to the success of Project Forward and will be discussed. Schematics can be referenced in the Appendix.

A. Electric Drive System

The electric drive system is crucial for the operation of Project Forward, it is the means in which the system provides assistance to the rider. The motor supplies power from a battery through an electronic speed controller to decrease the amount of effort a rider needs to exert. This feature allows a rider to consider cycling as a feasible means of transportation by reducing the strain of cycling. The design of these systems focuses on the features of this system, as well as their integration to work in a seamless fashion. In order to have a smart electrically assisted bicycle, first you need a bicycle in which to apply our features.

In order to provide assistance, there needs to be a conversion from electrical energy to mechanical energy in a controllable fashion. To fit the bill we used a motor, a motor controller, and a portable form of electrical power storage. The system also needs to be seamless to the user while providing the necessary assistance. All the systems need to be mounted to the bike and connect to each other as necessary in a fashion that is as unobtrusive to the rider as possible, allowing them to ride the bicycle as if with no modifications.

1) Options for Electric Drive Train and Solutions Implemented: First, we needed a bicycle. We could have built our own, bought one special for the project, or used one that a group member had available. We chose the last option as Michael had a bicycle gathering dust in his apartment that met the criteria recommended for a platform to build an electric bicycle on. This saved the group a couple hundred dollars. The general criteria necessary for an electric bicycle is a bicycle with working drive train components, brakes, and a robust frame capable of handling the extra weight of a motor and related accessories. The bicycle also has mounting points for a rack to mount large loads to, as seen in figure 35. A used mountain bike is a good platform because the frames are generally designed to take the abuse of dirt trail riding. These characteristics allow them to generally take on more weight of the additional components, as long as you don't then try and take them off road again.

Fig. 35: Specialized Rockhopper with Systems Installed

Now that we have a bicycle, we need a means to propel it. An electric motor is necessary to achieve this goal. For electric bicycles, common motor choices fall into two main categories: hub-drive and mid-drive configurations. Modern electric bikes tend to use brushless DC motors exclusively. Mid drive configurations mount the motor in the center of the bicycle near the bottom bracket. They drive the chain of the bicycle connected to the rear wheel. They tend to be a bit more robust, as they are fixed in their mounting location and not subject to the same mechanical abuse that a motor mounted in a wheel is. Mid-drive systems are more complex to connect to the bicycle, requiring alterations or replacement of the bike's crank set and drive train.

The other option is a hub-motor. A hub motor is contained within a wheel of the bicycle and electromechanical force is applied directly to that wheel instead of through the bike's existing drive train. They come in two distinct flavors, direct drive and geared. A direct drive motor tends to be larger than the geared versions to maximize torque at a lower RPM. They are capable of regenerative braking and are mechanically simpler than their geared counterparts. However, as they are constantly engaged, they do inhibit pedaling when power is not applied. This causes the rider to work against the motor when not in use. For Project Forward, we want to promote the rider providing as much energy as possible when they can without our system adding more work, so this solution was deemed insufficient.

A geared hub motor has the same drive train simplicity of the direct drive version, however they are smaller in size and they have the benefit of a clutch allowing the wheel to freely spin when not powered. This ability minimizes the effect of adding a motor the bicycle. The smaller size is resultant of having a 5:1 internal gear reduction inside the hub motor. This allows the motor to spin at a higher speed than the outside wheel to get more torque at a lower external wheel speed. Hub motors come with different windings as well, optimized for either a higher wheel speed or for more torque. For Project Forward, we wish to diminish the effects of terrain on a rider, therefore we opted for a higher torque winding motor, as can be seen in figure 36.

To control the motor, we need an electronic speed controller(ESC). The ESC takes a throttle input, in many cases a 0-5V analog voltage, and converts that into a set motor speed. This analog voltage can be replaced with a PWM with a varying duty cycle, allowing it to be controlled from a discrete time standpoint. As we had a motor selected, we needed a controller that can properly control a BLDC motor equipped with a Hall effect sensor. There are a few types of controller capable of this. First would be the name branded controller. The big name in ESCs is Kelly Controls. Their controllers for E-bikes start at \$150 and work their way up. They are somewhat proprietary, using their own plugs and connectors for accessories such as brakes and the throttle input.

Fig. 36: BMC V4T Hub Motor

The next main category is cheap Chinese controllers. These start at \$20 and go up from there. They tend to have little to no documentation and limited connectivity as far as accessories. The controllers have limited current handling capabilities due to them relying on the traces of the board to carry the motor current.

The solution we chose was to use a modified Chinese controller, the Lyen 9 FET Mark II Sensored ESC, as seen in figure 37 The manufacturer is fairly local and has provided a large amount of technical support to us. He added connections for accessories that are easily modifiable for our purposes, including brake input, a throttle input and current sensing port. The current sensing port measures the voltage across a 2 mΩ resistor. The traces on the board have extra solder applied to them, increasing their current carrying ability.

Fig. 37: Lyen 9 FET Mark II Sensored ESC

Lastly, we need a means to power our systems that doesn't require us to be plugged in with a long extension cord. While developing our laboratory prototype, we selected a 36V 10A DC power supply to allow us to develop our systems without requiring a battery and charger. For our second semester, we are developing a deployable prototype that requires us to run cordless. A 36V battery capable of supplying 10+ amps of current was required.

There are two main options to supply this power: lithium and lead acid batteries. The lithium based chemistries are capable of supplying larger currents for longer than their lead acid counterparts, but come with self-oxygenating properties that require additional safety considerations. Sealed lead acid batteries are heavier than similar capacity lithium batteries. Due to their more robust nature and ease of use, we went with 12V 12AH sealed lead acid batteries from UPG. As seen in figure 38, we wired the batteries in series to meet our need for 36V. The total weight of the pack is 30 lbs.

2) Interdependence and Integration: Project Forward, in its entirety, is attached to the bicycle. This requires all of our systems, from the electronic drive system, batteries, deterministic control system, and safety systems to be mounted to our bicycle.

The electric drive system interfaces with our deterministic control system through some connections we have exploited in our ESC. The ESC has a header for a product called Cycle Analyst. This

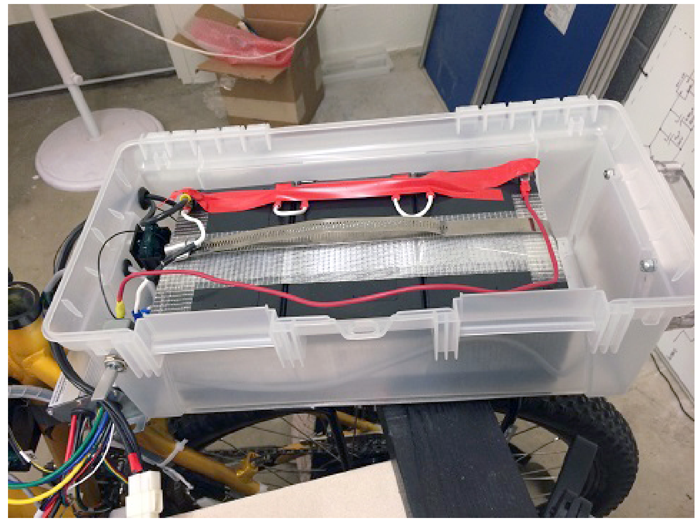


Fig. 38: Batteries Installed in Battery Box

header contains a throttle override connection, a current measurement, and a speed readout, all of which we exploit for our project.

The throttle override connection we use to control the speed of the motor from our control system. A PWM signal is sent by our controller with a varying duty cycle to control the wheel speed.

The ESC has a 2 mΩ internal shunt resistor we use to measure the current of the motor for our control system. We built a differential op-amp circuit to boost the voltage across the this resistor. The differential amplifier also contains a low pass filter to reduce high frequency noise picked up from inside the ESC casing. The gain of the amplifier is 57, the cutoff frequency is 10 Hz. The output from amplifier is fed into an ADC on our control system, which uses the current value to determine the amount of assistance the bicycle is currently giving the rider.

The speed readout provides a 0-5V pulse every time one of the motor's poles moves past one of its hall sensors. This happens 16 times per internal revolution, where there are 5 internal rotations per external rotation due to the internal gear reduction. This data is fed as a digital input into our system controller.

In order to utilize the batteries, we made a battery pack. The batteries are held together with clear strapping tape. The are wired in series to into a battery wiring harness that runs through a fuse and a kill switch. The kill switch has a flyback diode that provides a path for current to flow back into the battery if the switch is opened and the motor is under load. The battery, fuse, and switch is mounted to a battery box that we mounted on the rack on the rear of the bicycle frame. The battery is held in place with a hose-clamp style stainless steel strap. The box is reinforced with an aluminum plate on the bottom, which is used an anchor for the battery mounting strap. We pull 12 volts from the battery closest to the ground line, sharing the ground with the system. This 12 volt connection is stepped down using a buck regulator to provide 5 volts to our subsystems.

The motor is laced into a DT Swiss EX-500 rim. A robust rim is needed as the motor itself weighs 10 lbs, the bike weighs 25 lbs, and the battery pack adds an additional 30 lbs. The motor is mounted into the rear dropouts of the bicycle. We have reinforced the dropouts with the use of stainless steel torque arms. The torque arms prevent the wheel from turning out the dropouts of the bicycle.

B. Safety System

1) Lighting: To begin designing the lighting systems, we had to first select a type of light. It is common now for bicycle lights to be made from LEDs and for good reason. They are resistant to impact, bright, have long lifetimes, and draw little power which allows them to last for long durations without having to replace the battery.

After deciding on the type of bulb to use, the LEDs had to be selected. The color choices were determined by law. A headlight must be white, a brake light must be red, and turn signals must either match the color of the headlight or brake light, depending on whether they are in the front or back respectively, or they must be amber. [19] We decided to go with amber because it made them more distinct from the other lights.

We decided to use the C503B (red and amber) and LW514 (white) models of LEDs because of their 15 degree viewing angle, low energy consumption and cost. This was meant to represent a base state of lighting. We wanted to demonstrate to a potential user that they could have lighting that was inexpensive and wouldn't decrease the user's ride time by any significant amount. Of course, because of the modularity of the system, a user could replace these lights with brighter lights that would provide increased brightness at the cost of more power expended, depending on their needs.

Prototypes of the lighting systems were made on breadboards. This was the state of the lights at the time of the laboratory prototype. The lights all illuminated as expected and were controlled by the FPGA and transistor switches. With this setup, we were noticing some problems. When the brake light or headlight was held active and the turn signals were blinking, the brake light and headlight would blink inversely with the turn signal. This also occurred with the audible alert. We had speculated that it had been an issue with the programming but would later learn it was the fault of the resistance of the wires.

With over half an amp running through the wires to supply power to the lighting systems, the voltage at the input of the lights would drop significantly. This was not something we had previously encountered because we had never developed devices that draw that much current and up until now, the resistance of the wires had been negligible. Our solution to this problem was to use a power bus. This allowed the front lights, rear lights, USB, FPGA, and audible alert could draw power from their own wires rather than share a path. This reduced the max current in any line to less than 300mA. There is still some voltage drop, but it is not as significant.

The final process leading up to the deployable prototype was to convert the lights onto perfboards and properly encase and mount the lights onto the bicycle. We did this by first selecting suitable cases. We chose cases that would be large enough to house the components, had threaded holes for mounting the lights within, and were durable. The cases we chose were roughly 3 1/8 by 4 3/8 from Adafruit. The taillight enclosure was able to mount to the rear of the battery box without any addition modifications to the taillight enclosure. The headlight enclosure required two holes to be drilled into the back of the box and was then mounted to the handlebars with a reflector mount. Addition holes were drilled into the bottom of each enclosure to allow for the wiring. Originally, the lights were going to connect to the power hub by means of USB, but more robust connectors were used instead.

Further progress could be made on this feature by converting to PCB designs over the perfboard layout. This would increase efficiency. Other lights could also be produced to demonstrate the modularity of the system. What we have now is a low power/low brightness example, but a high power/high brightness alternative could be presented.

2) Audible Alert: The audible alert's design evolved over the course of the project. We anticipated the need for the horn when discussing safety at the beginning of the project.

The generation of the audio signal would be handled by the FPGA. We instantiated a Verilog module that starts counting on an input signal. The parallel output of this counter is fed to an R2R ladder. This R2R ladder was designed as a simple solution to converting a digital signal to an analog signal using the principle of voltage division.

As operational amplifiers ideally do not draw current, using one here as summing amplifier made sense. We selected a rail to rail op amp, as our input signal includes zero volts, and we have one 5V rail to use. For this purpose a MC33171-D was utilized. This amplifier is not capable of supplying enough current to power a speaker, creating the need for an audio amplifier.

We selected an MC34119 audio amplifier IC to drive our speaker due to its differential output. The differential output allows us to run the speaker directly to the output of the IC with no additional components in its signal path.

For the laboratory prototype we built this circuit on a solderless breadboard and had it driving a small PC speaker. We noticed that while running it with our other systems it would pick up noise and amplify it through the speaker, even without our input running.

It was clear the small speaker would not work for our deployable prototype, so we obtained a larger 4" speaker that was mountable to the bike. We tested the system with this load, the volume output was only 72dBA. We increased the gain of the MC34119 circuit by increasing the value of the low-pass filter and monitoring an oscilloscope connected to the output. We stopped increasing the gain once the output waveform started to clip.

The volume of the output was now measured at 95dBA at 3 feet. This was measured using Ultimate Ears dB measuring iPhone application.

Once the gain was set, it was time to finalize the design. We added a Darlington switch to control the power to the audible alert, disabling it when the system isn't activated. The circuit was transfered over to a perfboard, thus increasing the durability of the design and allowing it to mounted in our final control system box.

C. Power Distribution Hub

1) Voltage Regulator: The first step to designing the voltage regulator was to decide on the criteria. We estimated the current that would need to be provided by adding together the expected current draw of the FPGA, lighting, audible alert, and USB with a cell phone attached. We had estimated that we would need roughly 2A of current at 5V. We decided to design our regulator to provide 3A of current in case we drew more than we expected. The regulator also needed to be able to take in 36V as this was the source we had decided to use for our motor.

Two main types of voltage regulators exist. These are linear regulators and switching regulators. Linear regulators are much easier to design, but they are not efficient. The current in is the current out and the voltage get dropped. This results in a massive power loss. Additionally, the power lost is in the form of heat which would have to be dealt with. The switching regulator would be a more complex design, but in the end, would have a much greater efficiency.

We settled on the LM2676-5 buck step-down switching regulator. It has the capability to take in 8-40V and provide 5V at the output. A buck means it steps down the voltage, but steps up the current which is ideal for our project.

The first iteration of this design was on a breadboard. This is the design used for the laboratory prototype. The breadboard design was poorly implemented due to the limitation of a breadboard. The connections were not solid and it was difficult to place the parts where they could ideally be. This led to a working, but fragile, regulator. When tested, the regulator turned on and operated fine, sometimes for many weeks, but would eventually burn out and need to be replaced. Fortunately, we had linear regulators and 9V batteries on hand to replace the faulty buck in case it burnt out during the end of term demonstration.

Following the demonstration, changes needed to be made to the buck circuit to improve its durability and efficiency. The first change we made was to the input voltage to the buck from 36V to 12V. We decided it would be best to pull off of one of the main batteries instead of all of them in order to greatly increase the efficiency. This required a redesign of the buck circuit. Following the redesign, the buck was transferred to a perfboard. This would allow for a more ideal layout and solid connection. The buck was then tested and the results showed an increased efficiency.

Finally the buck needed to be mounted. The original idea was to have the buck and the transistor switches combined into their own enclosure, but we decided to combine it within the FPGA enclosure to conserve space and lessen the complexity of the system by eliminating one more case that needed to be mounted to the bike. Future improvements would be to convert the design to a PCB. This would increase the efficiency further.

2) Transistor Switches: We needed a way to have the FPGA control the lighting systems, but the FPGA can only supply up to 8mA on its pinouts. This is not nearly enough to power the lights. To combat this, we decided to use transistors as switches to be controlled by the FPGA. We decided to use BJTs because they are current amplifying transistors and that is exactly what we want.

A BJT by itself was having a hard time providing the current needed to the lights. For this reason, we decided to use a Darlington configuration with the transistors and this led to a large increase in current gain. We had attempted to use individual BJTs to build the Darlington configuration, but eventually settled on using prebuilt ICs to increase efficiency, simplicity, and durability. A pull-down resistor was included at the input as well as a resistor between the FPGA and base of the transistor in order to lower the 3.3V coming out of the FPGA to a more desirable voltage. The laboratory prototype was built using breadboards.

The design worked well and little needed to be done in the way of redesign. We converted the transistors to a perfboard to take up less space and provide more solid connections between components. The unit was then placed inside the FPGA enclosure along with the buck regulator.

Future modifications would include transferring over to a PCB design. The resistors at the input of the base could also be reselected to optimize efficiency, but as it stands, there is little loss through the resistors.

VIII. Software Design

THE software that controls this assistance feature is crucial to the user experience and function of the system. The software breaks into three large areas: The FPGA's HDL which controls the on bike systems and a cellphone application which allows the user to interact with the FPGA control system. The FPGA code is Verilog HDL and C for the soft processor, where the Android application is Java. This section will first describe the FGPA based system and then the Android application. The project's source code can be found on Github.

A. Design of Control System

Commuting by bicycle can be a daunting commitment. Many are deterred by the physical requirement and do not get the opportunity to build up enough fitness to enjoy the commute. The real time assistance feature reduces the amount of physical fitness required to start cycling. By lowering the barrier of entry we hope to encourage more people to choose cycling as an alternative transportation method which also has great health benefit.

The particular part of the project, or feature, highlighted in this section is a real time control system that controls the motor to provide intelligently adjusted assistance for a user. The output of the control system is generated by a number of physical and biometric factors. This control system could be adapted to any current motor and electronic speed controller pair. This means the control system is applicable to pre-made electric bike solutions for the hacker and custom builds for the DIY'er. In our case it is paired with a common Chinese ESC and motor which allows us to demonstrate the design.

The following sections the purpose of the real time assistance feature and its pertinence for the overall system's feature set will be discussed. The feature's testing plan and results of the testing will be reviewed to assure its fulfillment of the design requirement. Finally, the dependence on other systems will be discussed, including integration testing and results.

1) Design Purpose: The goal of Project Forward was to provide a way for people to get more exercise in their daily lives. One way we thought to address the situation was to make cycle commuting a reasonable alternative for a large number of people. Sacramento and its outlying regions are a great example of a cycle commute friendly area. The Bike trail from Folsom to Sacramento is a great example use case for this feature.

One of the greatest barriers to entry for new commuting cyclists is physical fitness requirement.

A rider of moderate fitness could not make the ride in addition to a full workday.

A review of the Folsom to Sacramento route tells us it is 20 miles one direction. A rider with a moderate fitness level might be able to ride 40 miles in a day without significant strain. This would take about three hours, not unreasonable for a ride. This effort can be too great a task when paired with an already exhausting workday and limited physical ability. The objective is to lessen the physical effort required to cycle commute. Addressing the issue of the physical exertion would allow more people to choose the alternative commuting method.

The system should be as transparent as possible. While the system should provide very tangible assistance the rider should not be distracted from the task at hand, enjoying their ride.

a) Fulfillment of Feature Set: The real time assistance feature solves this problem. The user needs to adjust the amount of exertion required by a particular route. This can be done with minimal user interaction by monitoring the rider's heart rate. When the user's heart rate goes up it follows that their cardiovascular system is under greater load and they require more assistance. This control system does just that. It acquires the heart rate though a commercial monitoring strap and adjusts the motor based on the current reading compared to a user entered setpoint.

2) Control Architecture: As described in the previous sections this system monitors a user's heart rate and makes a decision about how much assistance is required by monitoring a number of variables. Table IX and Figure 53 give an overview of our control systems topology. The system can be broken down into three basic parts: gathering data about the current state of the outside world, determining the amount of assistance to provide, and ensuring the motor provides an appropriate amount of torque for the assistance requirement.

3) FPGA Implementation: We chose to implement our design with a FPGA because of the devices innate parallelism and deterministic operation. The massive parallelism of the FPGA allowed us to write DSP modules for the IMU and ADC data streams. The direct control provided by Verilog hardware descriptive language means we could tune the system to our liking. During the project we took it upon ourselves to implement a custom I^2C and SPI driver to communicate with a few of our devices. Those bus implementations can be found in HDL files IMUInterface and ADC_CTRL. Table X shows the final metrics for device resource usage.

TABLE IX
OVERVIEW OF CONTROL HDL FILES

Filename	Description of purpose
ClockManagement.sv	Generates clock signals for modules
IMUCalculations.sv	Gathers the data from IMU module and processes the data
CellPhoneProtocal.sv	Retrieves user setting for heart rate cap from cell phone
ADC_CTRL.sv	Retrieves phase wire voltage from external ADC device
MotorControl.sv	computes current assistance requirement adjusts motor's speed setting accordingly
CPU.sv	NIOS II soft processor communicates with ANT module via UART and delivers heart rate data

TABLE X
FINAL SYNTHESIS REPORT

Family	Cyclone IV E
Device	EP4CE22F17C6
Total Logic Elements	10,295 of 22,320 (46 %)
Total Combinational Functions	8,734 of 22,320 (39 %)
Dedicated Logic Registers	4,795 of 22,320 (21 %)
Total Registers	4838
Total Pins	95 of 154 (62 %)
Total Memory Bits	396,296 of 608,256 (65 %)
Embedded Multiplier 9-bit Elements	8 of 132 (6 %)
Total PLLs	3 of 4 (75 %)

Figure 53 shows the six modules that are used to generate the motor control signal "PWMout" that is fed into the motor. This signal is synthesized by comparing a number of parameters that the user sets and the current state of the system as seen by a number of sensors. Table IX lists each file and gives a brief description of its purpose. The function of each module will be expanded upon in the next several subsections.

A National Semiconductor ADC128S022 analog to digital converter is used to sample the voltage

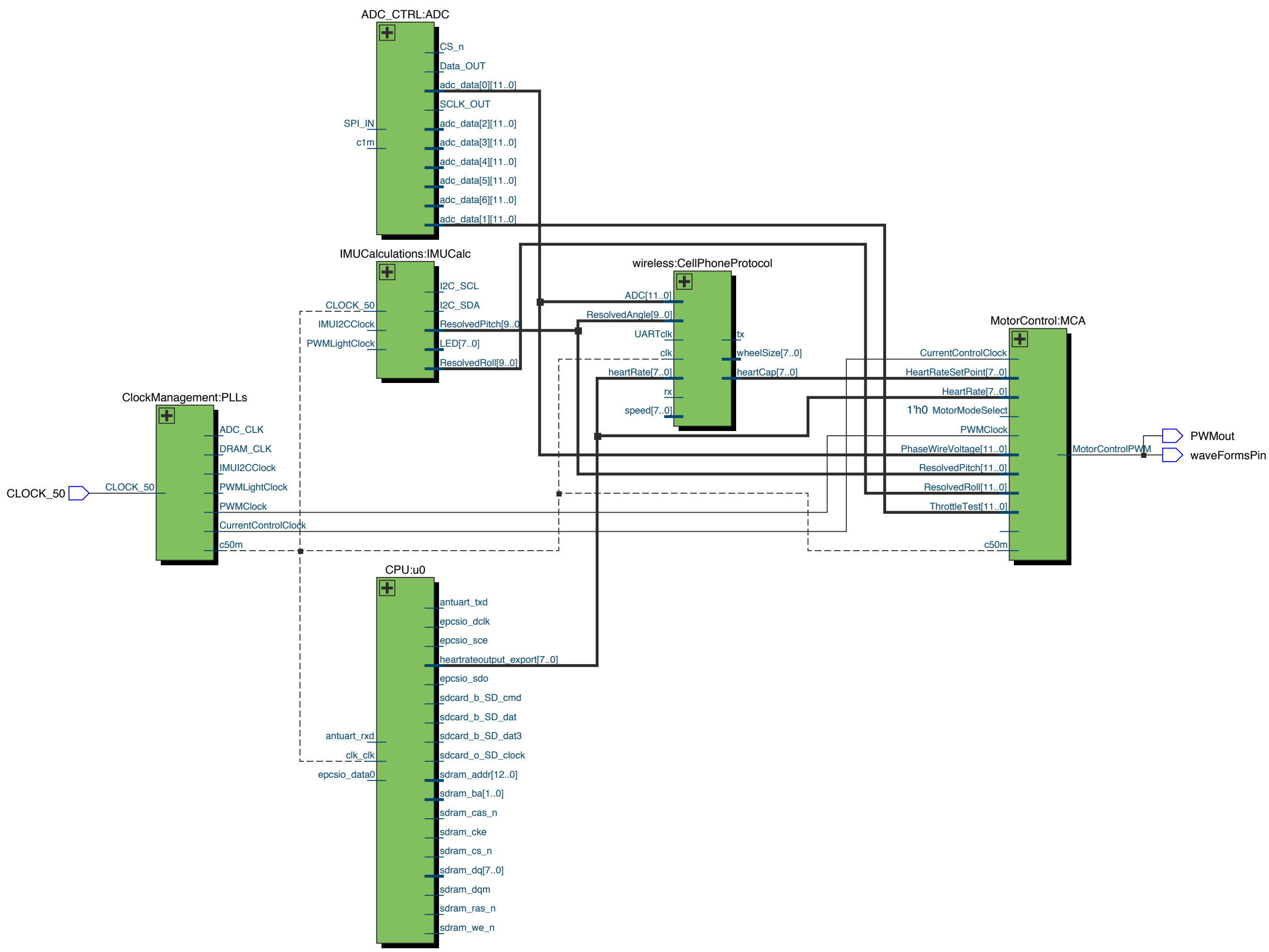


Fig. 39: Overall Control System Architecture

TABLE XI
REFRESH RATE OF IMPORTANT SYSTEMS

Filename	Data update rate
IMUCalculations	100Hz
CellPhoneProtocal	50Mhz
ADC_CTRL	400hz
MotorControl	.25Hz
Heartrate	.50hz

across a shunt resistor internal to the electronic speed controller. An analog front end was built by the electrical engineering team to take a differential measurement across the resistor that included a single pole low pass filter and amplified the signal until our maximum expected current generated a 3.3V signal at the ADC. 3.3V was chosen because it was the reference voltage for the ADC and would provide smallest voltage step per LSB and greatest accuracy. The current measurement is taken at 400Hz to provide sufficiently up to date information for the current control algorithm inside of the Motor Control module in order to make appropriate adjustments to the motor signal.

4) Heart Rate Data Gathering: Heart rate is the primary feedback component of our system. Measuring the user's heart rate allows the system to know the current exertion level of the rider. This allows our system to adjust the power output of the motor in real time to suit itself to the user's need. We chose heart rate as a control mechanism because the body will increase the heart rate as it sees fit based on the bodies demand. As user exertion increases the heart rate will increase, this gives the system a good indication of when the user is overexerted. Our system steps in when the exertion level of the user is too high and provides assistance with the electric motor.

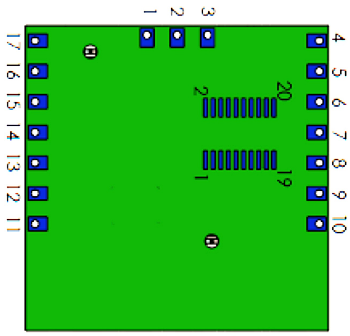


Fig. 40: FIT 2 Pin Map

TABLE XII
AP2 ANT+ TRANSCEIVER MODULE PIN MAP

Pad Number	Bank	Module Name	Module Description
1	TOP	TEST	Tie to Ground
2	TOP	RST	Reset the Device
3	TOP	VCC	Power Supply
4	Right	GND	
5	Right	NC	
6	Right	SUSPEND	Suspend Control
7	Right	Sleep	Enable Sleep Mode
8	Right	NC	
9	Right	PORTSEL	Tie to GND for async
10	Right	BR2	Baud Rate Slection
11	Left	TXD0	Transmit Data Signal
12	Left	RXD0	Receive Data Signal
13	Left	BR1 Baud	Rate Selection 1
14	Left	BR3 Baud	Rate Selection
15	Left	Reserved	Tie to GND
16	Left	Reserved	Tie to GND
17	Left	RTS	Request to Send

a) NIOS II Softprocessor: The NIOS II soft processor is used to communicate with the ANT+ module via UART. We chose to do this because of the availability of C code which parses ANT message packets and communicates with the Garmin ANT+ heart rate strap. Table XIII shows the different components used to build the soft processor system. These are individual component instantiations though Altera's SOPC builder. The soft processor uses 3200 Logic Elements and has a maximum frequency reported by the static timing analysis tool Timequest of 133MHz but is currently clocked at 100 MHz. The soft processor configures itself at boot by reading a image stored on the FPGA's EPCS nonvolatile configuration device. At boot the program image will be loaded into the SDRAM which contains the entire memory space of the processor. Table XIII contains information about the various modules that can be found instantiated in our project. The soft processor is programmed in C that was compiled with Altera's NIOSII EDS toolset. This code can be found with all of the project's code on Github. Figure 41 shows the basic execution flow of the program.

TABLE XIII
OVERVIEW OF NIOS II MODULES

Module name	Description of purpose
clk_0	Clock Source for RAM and NIOS, driven by FPGA PLLs
NIOS2_QSYS_0	NIOS II E 32-bit RISC soft processor
JTAG_UART	UART link over USB for communicating with development tools
SDRAM	Main memory for NIOS program, variables, constants
UART_0	UART for communication with ANT+ transceiver
TIMER_0	Timer to trigger polling of ANT+ transceiver and controls such as timeouts
HeartRateOut	Parallel IO to send heart rate back to FPGA for use by control system
EPCS_Flash_Controller	Allows NIOS II to use the FPGA's EPCS configuration device for non-volatile storage
WDT	Extra timer that resets the ANT+ link in when there hasn't been a valid data packet received in 10 seconds

b) AP2 ANT Transceiver: Our system uses a commercial transceiver that uses ANT+ transmission standard. This was chosen because of the availability of commercial heart rate sensors that use this standard to communicate data. Other transmitter options include Blue tooth and a variety of vendor specific RF standards. We wanted a standard that was compatible with a number of different sensors which restricted our choices to ANT or Bluetooth. ANT+ was selected over Bluetooth because of the more simple connection procedure, authentication system, and the standardized protocol for message transmission. ANT+ has a specific provision for heart rate monitors that specifies the rate in which heart rate data is sent to receiver. This standardized message protocol means the heart rate will be enclosed in a packet with a specific message ID which

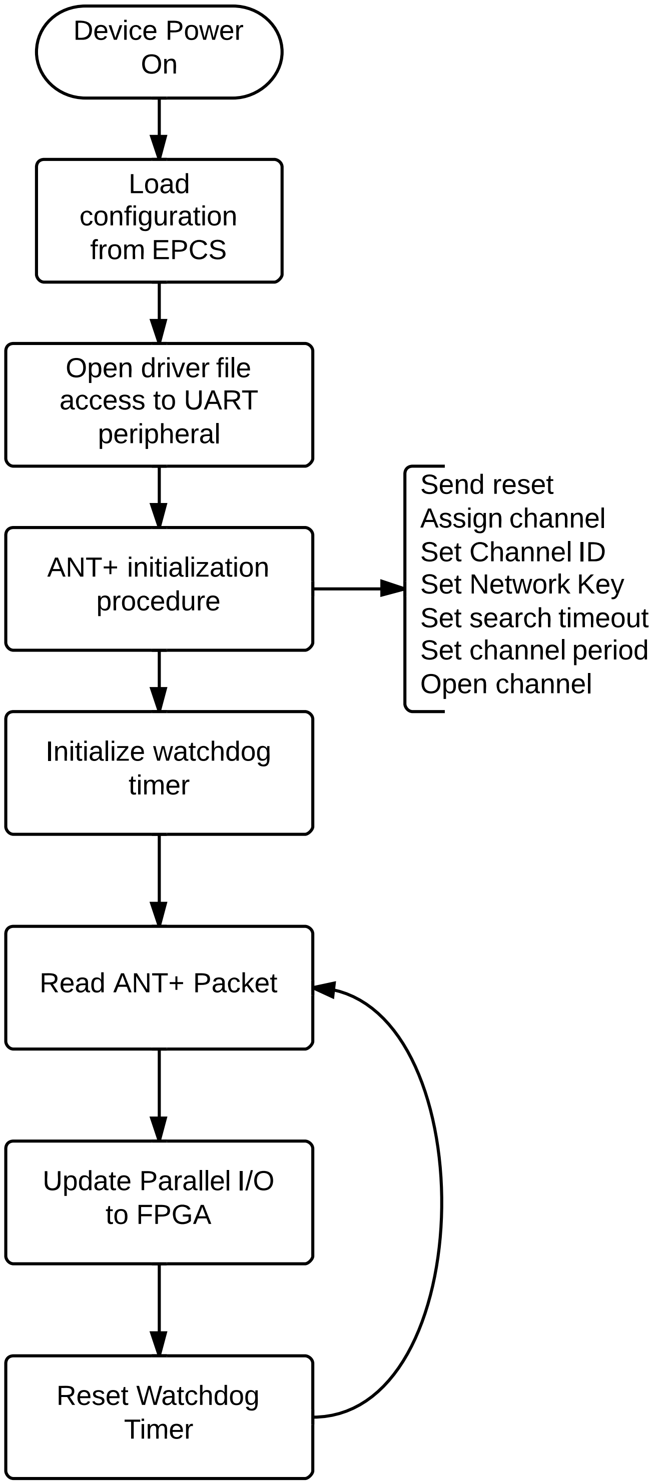


Fig. 41: C Program Flowchart

makes the information easy to extract. This device provides reliable data but the low power transmitters are limited by range, the specifics of how the range issue are mitigated are discussed within the testing section.

c) Garmin Heart Rate Strap: A standard commercial heart rate strap-sensor was used. It can be purchased from their website [20]. The current implementation of the system is only compatible with the Garmin device. ANT+ devices use a unique manufacturer identifier as part of the address. Expanding compatibility would mean small changes to the C code for the NIOS II soft processor.

5) IMU Data Gathering: Part of the feature set was to anticipate changes in the user's assistance needs. We thought to do this by sensing the inclination of the system. If the user is about to climb a hill it follows that they will increase physical exertion as well as heart rate. We did not want to allow this spike in heart rate before the system responded. This is accomplished by using an accelerometer and gyroscope whose data is processed using the FPGA and some DSP algorithms. This information is passed into the assistance algorithm which uses in its calculation of the current motor setting.

The internal structure of IMUCalculations.sv can be seen in Figure 42. IMUInterface.sv contains the low level I^2C driver written for the project that communicates with the accelerometer and gyroscope. LowPassFilterAverage.sv contains the IIR filtering mechanism that will be described below. SensorFusion.sv contains an algorithm which combines the high speed accuracy of the gyroscope with the accelerometer's absolute reading for reduction of drift. The data is then passed though a Taylor Series approximation for tangent, because transcendental functions are not supported by Verilog.

The HDL for the angle resolution algorithm can be seen in Listing 1. Note the division by 100 at the end of line 33; this was needed because the algorithm we used required the use of decimal constants which Verilog does not allow for. The entire equation was scaled for processing and then reduced.

a) Inertial Measurement Filtering Algorithms: An item of particular interest was the filtering algorithms used to process the accelerometer's data stream. The angle calculated by the IMU processing module is used in part to calculate the amount of assistance the motor gives to the user. A stable signal from the accelerometer allows the value to be used in the assistance algorithm without noise affecting the user's experience. Figure 43 shows the simple hardware synthesis of the Verilog code. This style of filter accomplishes the desired task

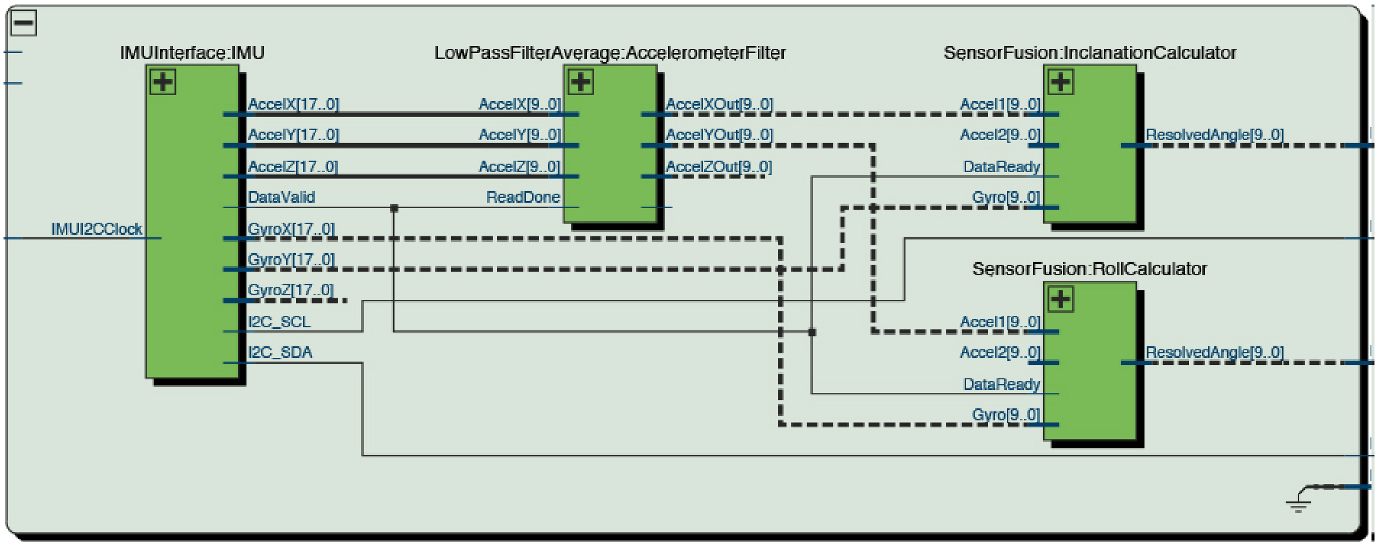


Fig. 42: IMU Processing Architecture

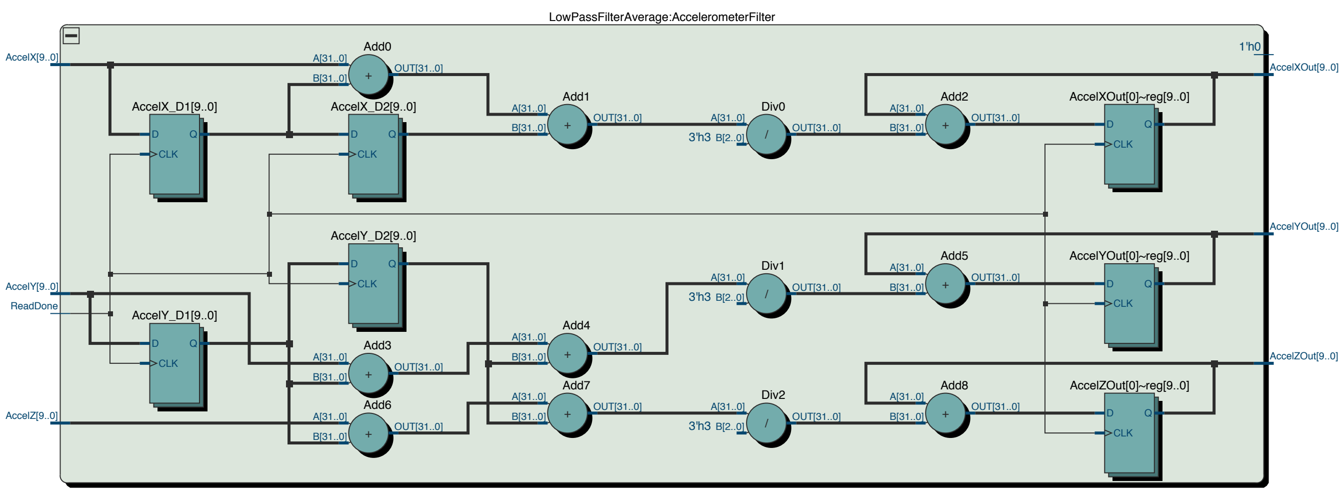


Fig. 43: IMU Windowed Average Filter for Accelerometer Data

of smoothing the data in the time domain. More information about the recursive moving average filter can be found in the textbook referenced for this project. [21]

6) Motor Control System: The motor control algorithms are the central components for this feature. The motor control system as a whole is divided into three parts internally. AssistanceAlgorithm takes in the heart rate cap setting from the cellphone, the IMU data as well as the current heart rate from the soft processor. It is a proportional control system at its core.

CurrentControl is a module that uses the ADC reading of a shunt resistor that the ESC contains internally on the ground line for the motor power phase lines. This is required because the ESC does not offer torque as a measurement. This problem and how we overcame it will be described in the testing section to follow.

The PWM generator is a high frequency, 2MHz output, PWM generator for the ESC. The ESC accepts an analog voltage for its set point. There is a small time constant RC filter with unknown values at the input of the ESC set point. We experimentally discovered that it required a very high frequency of output or it would oscillate though the high and low states of the PWM. Upon discovering this we pushed the PWM's clock up to the highest frequency the FPGA speed grade would allow.

B. Testing of the FPGA Control System

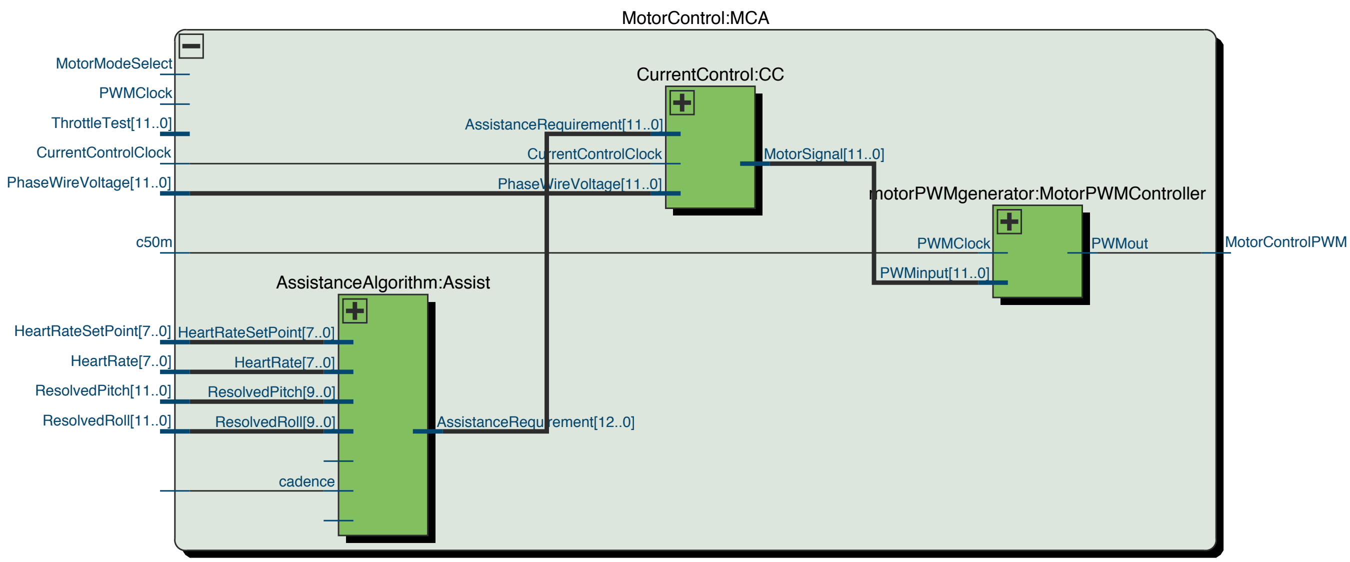


Fig. 44: Motor Control Algorithm Architecture

```
module SensorFusion(
  input                                 DataReady,

  input             signed  [9:0]   Accel,
  input             signed  [9:0]   Gyro,
  output    logic   signed  [9:0]   ResolvedAngle

    //|
    //| Constants for angle calculation and adjustment
    //|-------------------------
    localparam          SampleTime =  1;    //multiplied by 100
    localparam signed   GyroOffset = -7;
    localparam signed   AccelOffset = 8;


    //|
    //| Local registers and wires
    //|-------------------------
    logic  signed [9:0]  AngleNext = 0;
    logic  signed [9:0]  SampledAccel = 0;
    logic  signed [9:0]  SampledGyro = 0;

    //|
    //| Main logic
    //|-----------------
    always_ff@ (posedge DataReady) begin
        ResolvedAngle <= AngleNext;

        SampledAccel <= Accel;
        SampledGyro <= Gyro;
    end

    always_comb begin
        AngleNext = (94*((ResolvedAngle*99 -
      (SampledGyro+GyroOffset)/3)/100)+(SampledAccel+AccelOffset) *
      (100-94))/100;
    end
endmodule
```

Listing 1: IMU Angle Processing Algorithm

THE testing of the illustrated a number of design challenges. The greatest of them was our ESC's set point was for speed, not current, the solution is described below. The basic goal of testing was to ensure the control system provided the experience the group envisioned during the design phase. We used a number of tools in the testing of this feature. The primary method was SignalTap, Altera's embedded logic analyzer. SignalTap allowed us to gather a great deal of data at every cycle of the clock. We also used Altera's embedded sources and probes Megafunctions which gave us a way to change system parameters from the computer in real time. This allowed us to modify constants in our control systems and observe the change in response.

1) ESC Control Problem: The greatest challenge of the testing phase was adapting to the change in the definition of the ESC's control point. We originally thought that the analog voltage controlled the flow of current for the motor. This was incorrect, the voltage is mapped to a speed, the controller will increase current as much as required to meet this speed.

This was overcome by the addition of the Current Control module described in the Verilog section. This module measures the current flowing to the motor by way of a shunt resistor. The module compares this number against the product of the assistance algorithm to adjust the speed setting for the motor. This provides a mechanism around the speed setting problem and allows the control system to indirectly set torque via current.

2) ANT Distance Limitations: Another important limitation discovered in testing was the range limitation of the ANT+ standard. We originally planned to mount the ANT transceiver behind the user; this location did not yield enough valid data packets to be useful. We found the user's body to be the cause of the signal loss. The problem was mitigated by moving the transceiver in front of the user. Since the relocation of the ANT transceiver there has not been a problem with obtaining data

C. Mobile Application Design

Since the user requires an ability to control the amount of assistance, our system needed some form of input with many different levels. After thinking about multiple different possibilities for the input, we realized the rider most likely has a very familiar input device in their pocket, or possibly already mounted on their handlebars. A large majority of Americans now own a modern smart phone. They posses the hardware to be able to communicate wirelessly, the software to be able to display and process information, and a small enough form to be safely mounted on handlebars.

a) Feature Set: Once we decided to make a mobile application, we needed to create a list of features and design guidelines. Firstly, we wanted to keep the application simple and easy to use. Meaning no requirements for logging in to anything or Internet access, as well as minimal steps to reach full system operation. We also wanted an easily recognizable icon to launch the application(Seen in Figure 45).

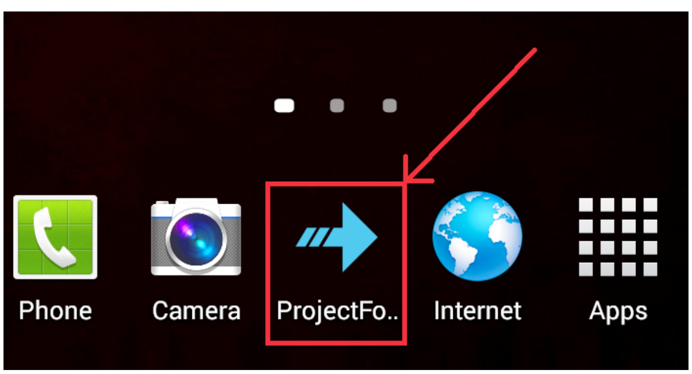


Fig. 45: Application Icon

1) Connect, send, and receive data wirelessly over Bluetooth with the FPGA
2) Be able to enter heart rate value for the assistance algorithm
3) Be able to enter wheel size for speed calculations
4) Display system data
   - Angle of Inclination
   - User Heart Rate
   - Speed
   - Current Draw of Motor(Assistance)
5) Log the displayed data in a CSV file

b) Fulfillment of Feature Set: The mobile application provides a robust user interface and satisfies our original goals of allowing the rider the control the level of assist and display system data. The rider is easily able to enter a desired heart rate cap that is sent to the control algorithm on the FPGA. They can also see pertinent real time data coming from the FPGA. Additionally, it is made for a small form factored device that fits nicely in a universal handlebar mounted cellphone holder. All of these features have been implemented and tested thoroughly which will be discussed later in this document.

1) Mobile Interface Architecture: Creating a mobile application that fulfills our intended features requires more than just a few files. We have a Verilog state machine on the FPGA to react to the cell phone's requests and gather the required data. Some data received from the phone, like heart rate cap and wheel size, are also distributed to and stored in the modules that require the data. It sends and receives the data to and from the phone using an RN-41 Bluetooth transceiver that works through UART.

The actual mobile application is written in XML and Java. There is a main XML file, AndroidManifest.xml, that states which versions of Android the application is compatible with, what permissions it needs from the user, as well as the general layout of the application. Each different screen also has its own XML file to control the layout and define what functions the "active areas"will call when they are pressed. The actual processing and Bluetooth communications are done in Java. The cell phone must find the RN-41 on the FPGA, request a connection, and set up input and output streams. It will then communicate with FPGA depending on which buttons have been pressed. Additionally, if the display button has been pressed the application will display and log all of the data from the FPGA into a CSV file on the phone. This file can easily be opened in excel on any computer.

Fig. 46: Cell Phone Mounted

2) Design Decisions: We decided to go with the Android operating system for the free development cost, compared to Apple's $99 per year subscription to put applications in their store. Additionally I already have an Android smart phone and do not own a Mac. We decided to use Bluetooth as the wireless communication because it will work as long as both ends have power, and does not require any external routers or Internet connectivity. The FPGA had already been chosen as our main controller, so I had no choices there but to create a Verilog module for the interface between the system and the RN-41 Bluetooth transceiver. I chose to have the data display on the same screen as the user input to simplify the process of changing assistance while riding. Riding a bicycle is dangerous enough without having to navigate a complicated application.

3) FPGA Implementation: In Verilog we needed a UART interface to communicate using the RN-41 transceiver. We obtained an open source UART module from Open Cores. It has an input pin, rx, connected to the tx pin of the RN-41. And the tx output is connected to the rx input of the Rn-41. They Bluetooth transceiver naturally runs at 115200 Baud, so we needed to change the clock divider in the UART module to run at the same speed. When the UART module and the RN-41 transceiver are both working correctly, the module holds a received

TABLE XIV
DATA TRANSFER PROTOCOL

10 Bit Hex Value	Description of response
001	Send heart rate reading
002	Send the sign of the angle
003	Send the value of the angle
004	Send speed value
005	Get ready to receive heart rate cap
006	Get ready to receive wheel size
007	Send low byte of shunt ADC value
008	Send high byte of shunt ADC value
1XX	Assign received value to heart rate cap register
2XX	Assign received value to wheel size register

bit high after receiving a full 8 bits and filling up the rx register. When data is to be sent to the cell phone, the tx register must be filled, then a tx bit can be driven high for the UART module to send to the data out through the tx pin.

After the UART module was working as intended we created another module to receive the cell phone's intent as well as send out the correct data. This is created in the Wireless.sv file and contains a case statement triggered on the received value from the phone(Seen in Table XIV). Once the FPGA receives a value, it loads the tx register with a value based on the received byte, then drives the transmit byte high for a clock cycle so the UART module transmits the tx byte over Bluetooth. A small sample of the Verilog code can be seen in Listing 2.

```
//|
//| Parameters for the case statements.
//| ----------------------------------------
localparam SEND_HR = 10'd1;
localparam SEND_ANGLE_SIGN = 10'd2;
localparam SEND_ANGLE_VALUE = 10'd3;

always@(posedge clk)
begin
  //| Once a byte has been received from the cell phone, the
      received bit will be high for one
  //| clock cycle. The Android application will be sending requests
      for specific data by
  //| sending a specific byte of data to the FPGA.
  if(received)
    begin
      //| This case statement controls what data will be sent to
     the cell phone depending
      //| on the byte most recently received from the cell phone.
      //if(!initialize_heart && !initialize_wheel)
      casex({initialize_heart,initialize_wheel,rx_byte})
        SEND_HR:         //Send heart rate to cell phone.
          begin
            tx_byte <= heartRate;
          end
        SEND_ANGLE_SIGN:    //Send just the top two bits of the
     resolved angle so the cell
          begin        //phone can interpret the sign.
            tx_byte <= 8'd0;
            tx_byte[1:0] <= ResolvedAngle[9:8];
          end
        SEND_ANGLE_VALUE:   //Send the bottoms 8 bits of the
     resolved angle. The Android app is in
          begin        //charge of interpreting it correctly with
     the previous byte.
            tx_byte <= ResolvedAngle[7:0];
          end
        default:    //If there is an error in the bluetooth
     transmission, just send back a 0.
```

```
            tx_byte <= 8'd0;
      endcase
      transmit = 1;   //This must be high for one clock cycle to
     send the data in the tx register
                 //to the cell phone.
    end
  else transmit = 0;       //This must remain low until a byte is
     loaded into tx register and is ready
end               //to be sent.
```

Listing 2: Sample of Wireless Verilog

4) Android Implementation: The format of the different screens are created in XML files. The top XML file also contains the required permissions such as allowing use of the cellphones Bluetooth module and reading and writing to files in memory. In the XML files, the buttons, or "active areas" are defined along with the functions or methods they will be calling.

The first screen the user sees after clicking the application icon, is a simple splash screen that gives info about us. After pressing the begin button, the user enters the main screen. From here there are a few different options as can be seen in 48.

The "Connect to Device" button initializes the Bluetooth connection as well as the input and output streams between the devices. As of now the user can pair the phone with the RN-41 through their normal Bluetooth menu when they enable their Bluetooth. The application will check the list of devices stored on the phone, and when it finds the RN-41 it assigns it to MyDevice pointer. It then attempts to connect to it, then open input and output streams if the connection was successful. This method is shown in Listing 3 on the next page.

The "Begin Display" buttons begins the display loop which runs off of a timer. This begins twice a second, and requests data from the FPGA, logs the received data, and displays it on the screen. After the mobile application sends the specific byte to requests data, it then waits and checks the input buffer to see if the FPGA has responded. If not, it sends the request again, otherwise it takes the byte and places it in the correct location. The flowchart diagram in Figure 47 illustrates the main display loop that follows the protocol outlined in Table XIV.

Since the main display loop is running on a timer, the set buttons will work independently of the display. The user can tap the empty text box labeled "Enter Heart Rate Cap" to bring up a virtual keyboard to enter the value they want. Then once the "Set" button is pressed, the value is sent to the FGPA and is routed to the correct register in the Wireless Verilog module. The ADC data, angle

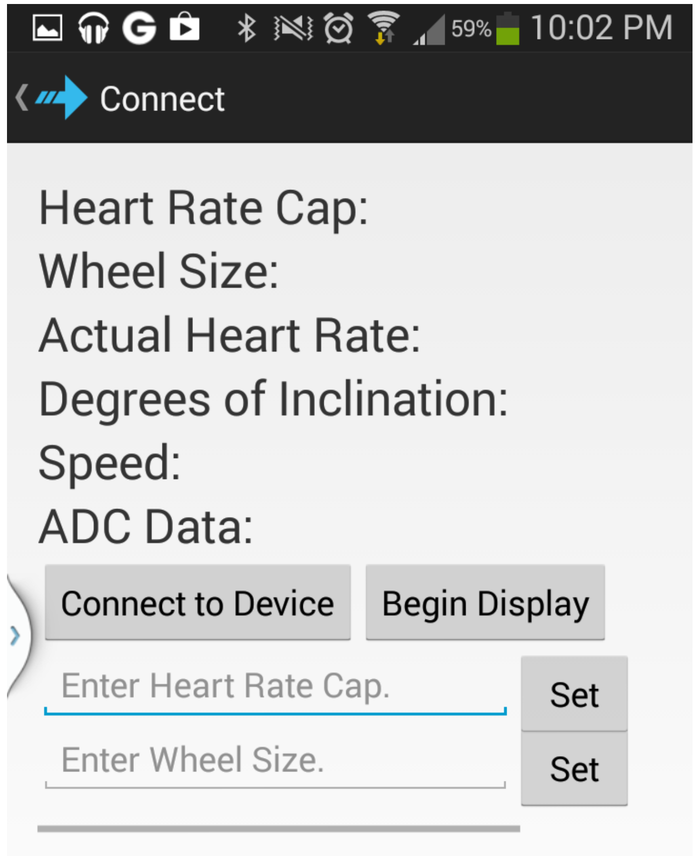


Fig. 48: Main Screen

data, and speed data are coming from the FGPA and a very raw format and must be processed to display a recognizable value to the user. If the inclination angel is negative, the value must go through Equation 1 to deal with the 2's compliment form, and a negative will be added in front of the value after it goes through Equation 2. If the angle is positive in just goes through Equation 2 which converts the raw number into an angle in degrees.

The speed had to be shifted right by 2 bits to fit into a byte to send over Bluetooth. Once the phone receives that value, it must be sent through Equation 3 to get MPH. Wheel size comes from the user input in inches.

The current in Amps is found with Equation 4. The divider is found by using Ohms law with the resistance of the shunt resistor in the ESC, and the voltage we are reading from it. It is also being amplified, so the number reverses that operation to display the smaller number to the user.

$$\text{input} = 255 - \text{input} \qquad (1)$$

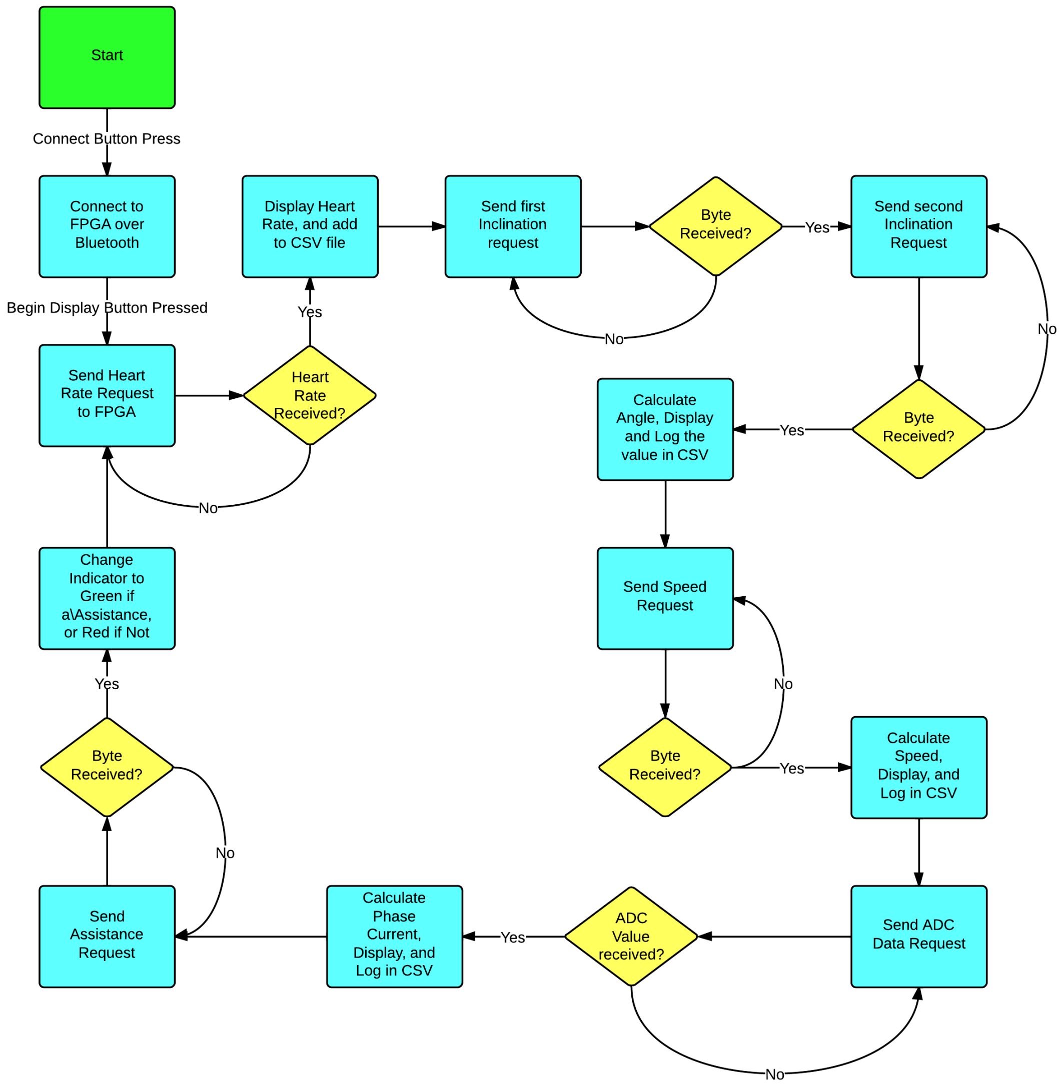


Fig. 47: Java Display Loop

$$\text{Angle} = \frac{\text{input}}{2.8} \tag{2}$$

$$\text{Speed} = \text{input} * \text{WheelSize} * \frac{60}{63360} \tag{3}$$

$$\text{Current} = \frac{\text{input}}{902} \tag{4}$$

```
//Set up the Bluetooth adapter on the phone, and use it to get a
     list of bonded devices.
//Bonded devices are not connected yet(no communication socket).
//------------------------------------------------
    myBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    myBondedDevices =
        (BluetoothDevice[])
      myBluetoothAdapter.getBondedDevices().toArray(
            new BluetoothDevice[0]);

    //If there are no devices bonded to the phone, notify the user.
    //-----------------------------------------------
    if (myBondedDevices==null){
      TextView t = (TextView)findViewById(R.id.top_connect);
        t.setText("bad dog");
    }
```

```
//Populate the array of Bluetooth devices that are bonded to
    the phone
//------------------------------------------------
if (myBondedDevices.length > 0) {
    int deviceCount = myBondedDevices.length;
      if (mDeviceIndex < deviceCount) myDevice =
    myBondedDevices[mDeviceIndex];
      else {
          mDeviceIndex = 0;
          myDevice = myBondedDevices[0];
      }
      String[] deviceNames = new String[deviceCount];
      int i = 0;
      for (BluetoothDevice device : myBondedDevices) {
          deviceNames[i++] = device.getName();
      }

      myDevice=myBondedDevices[0];
      TextView t = (TextView)findViewById(R.id.top_connect);
      t.setText(deviceNames[0]);
      TextView t3 = (TextView)findViewById(R.id.top_connect2);
      t3.setText(" ");
      TextView t8 = (TextView)findViewById(R.id.top_connect3);
      t8.setText(" ");
      TextView t9 =
    (TextView)findViewById(R.id.heart_ratecap_display);
      t9.setText("Connection Succesful!");
      TextView t10 =
    (TextView)findViewById(R.id.wheel_size_display);
```

```
        t10.setText("Connected to: ");

        //Create a Bluetooth socket and try to connect to the RN41.
      It has to be the
        //first bonded device in the Bluetooth options on the phone
      at this point.
        //----------------------------------------
        try {
       mySocket =myDevice.createRfcommSocketToServiceRecord(MY_UUID);
      } catch (IOException e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
      }
        try {
       mySocket.connect();
       connectedFlag = 1;
      } catch (IOException e) {
       e.printStackTrace();
     TextView t42 =
      (TextView)findViewById(R.id.heart_ratecap_display);
         t42.setText("Connection failed!!");
        connectedFlag = 0;
```

Listing 3: Establishing Bluetooth Connection

5) System Integration and Interdependence: The cell phone mounts very easily to the handlebar and interacting with it while riding the bike is not a problem. The application is very dependent on the state of thee FPGA. If there is no power to the system, then the application will do absolutely nothing. The data logging files are still accessible from the file system, but the application its self is useless without the FPGA. Also, there would be no point in setting a heart rate cap if the rider was not wearing a functioning Garmin heart rate monitor. So, for the mobile application to be of any use, the entire system must be working properly. We have also been running into some unforeseen dependence on the NIOS II soft processor. Sometimes when the heart rate monitor stops transmitting, the NIOS must be reset and occasionally causes problems with the UART module for some reason. We are still currently looking into this, but it is a rare occurrence.

6) Possibilities for Expansion: Having the application running on Android opens up all kinds of possibilities. There are many different ways to display the system data including graphs and images. The data can also be stored in cloud storage instead of directly on the phone. Adding new features from the FPGA would only require new routing and case statements, but could be done easily. There could also be additional features to be able to see the data and add favorite assistance profiles while not actually being connected to the bike.

The mobile application provides the user a simple way to adjust the assistance level and see real time data. It can also be user by the rider to evaluate their performance over time by keeping the data logging files. The application has proven to function correctly on my Samsung Galaxy S3 smart phone and provide full functionality to the rest of the system. Being able to send a user heart rate cap to the FPGA is a vital step in our smart assistance algorithm, and the mobile application is very capable of accomplishing this task.

D. System Integration and Interdependence

THE control system does not operate in isolation. It requires information from the cellphone for user data entry and the motor to provide assistance. These two systems required a great amount of integration testing when they were paired with the control system. A great deal of time was expended in the second semester to integrate the individual systems in which the control scheme relied upon.

1) Cell Phone Application: The most important data received from the cellphone is the heart rate threshold value for the assistance algorithm. This threshold is the number the assistance algorithm uses to compare the current heart rate against the dominant component of the assistance calculations.

2) ESC and Motor: As described in the testing section, how the ESC interprets the control signal is very important for the assistance control system. Finding a way to accommodate the ESC's mode of operation was one of the great technical hurdles of the project. The development of the algorithm contained within the CurrentControl module represented a significant portion of development time in the second semester.

E. Conclusions from Integration Testing

The real-time assistance feature is at the core of the project and how we're attacking America's problem of inactivity. The feature was defined during the design phase as real time assistance based on biometric feedback. The heart rate sensor provides the biometric feedback that is needed to control the motor by the user's bodies' requirements of the moment. This in conjunction with the CurrentControl module which allows the control system to adjust current, which approximates torque, meets the full requirement of the feature's definition. Moreover, this system operates in tight integration with the cellphone application and the Motor ESC pair to fully accomplish the goals of the project. We have provided a system that offers a number of conveniences to the cycle commuter, that can extend their range, and ease their burden. It is the hope of the designers that the real time assistance mechanism will allow more people to commute on their bicycle and enjoy the benefits of a healthy lifestyle.

F. Safety System Control

The Safety Systems are a set of buttons and switches for the user to interact with. These switches are used to create input signals for the FPGA that control the various safety systems. Figure 49 shows the inputs, outputs and System Verilog modules used to implement this functionality.

TABLE XV
OVERVIEW OF CONTROL HDL FILES

Filename	Description of purpose
ClockManagement.sv	Generates clock signals for modules
SafetyControls.sv	Safety System top level file
SoundRamp.sv	Generates audio waveform for horn
BlinkerControls.sv	Creates flashing effect for front and rear turn signals

IX. MECHANICAL DRAWINGS AND DOCUMENTATION.

THE second phase of Project Forward was transitioning the project from a stationary laboratory prototype to a rolling unit built an an existing bicycle. This required a more complicated mechanical build including integrating systems that were on breadboards to more reliable and permanent units on perfboards, mounting these systems in enclosures to reasonably protect them from the elements and instructor's boots, and attaching them to a moving bicycle.

A. Motor and Wheel

The hub motor was laced into a wheel using a single cross pattern. The wheel was bolted directly to the frame initially, however during simulated load testing using the added resistance of a brake on the wheel, the motor spun out of the frame, damaging the frame and the ESC in the process. The motor was reinstalled with stainless steel torque arms in place, securing the wheel to the frame using both the traditional bolt on the axle, as well as stainless steel hose clamps to keep the torque arm secured to the frame. The torque arms, now secured to the frame, will prevent the motor from spinning within the dropouts and falling out for the frame again. Detailed views of the installation of the drivetrain and torque arm can be seen in 50 and 51.

Fig. 50: Detailed View of Drivetrain Integration and Torque Arm Installation

Fig. 51: Detailed View of Disc Brake Installation of Hub Motor

B. Rack

To hold the batteries on the bike, we installed a standard bicycle rack on the rear of the bike. It is held in place utilizing existing hardpoints on the frame designed to mate with accessories such as racks and fenders. The rack's specification sheet states that it can handle loads up to 50 lbs. Our batteries weigh in at 30 lbs, plus an additional 5 lbs for the battery box and accessories attached to the battery box. The rear rack ensures that we have a safe place to hold our batteries that will not get in the way of normal operation of the bicycle.

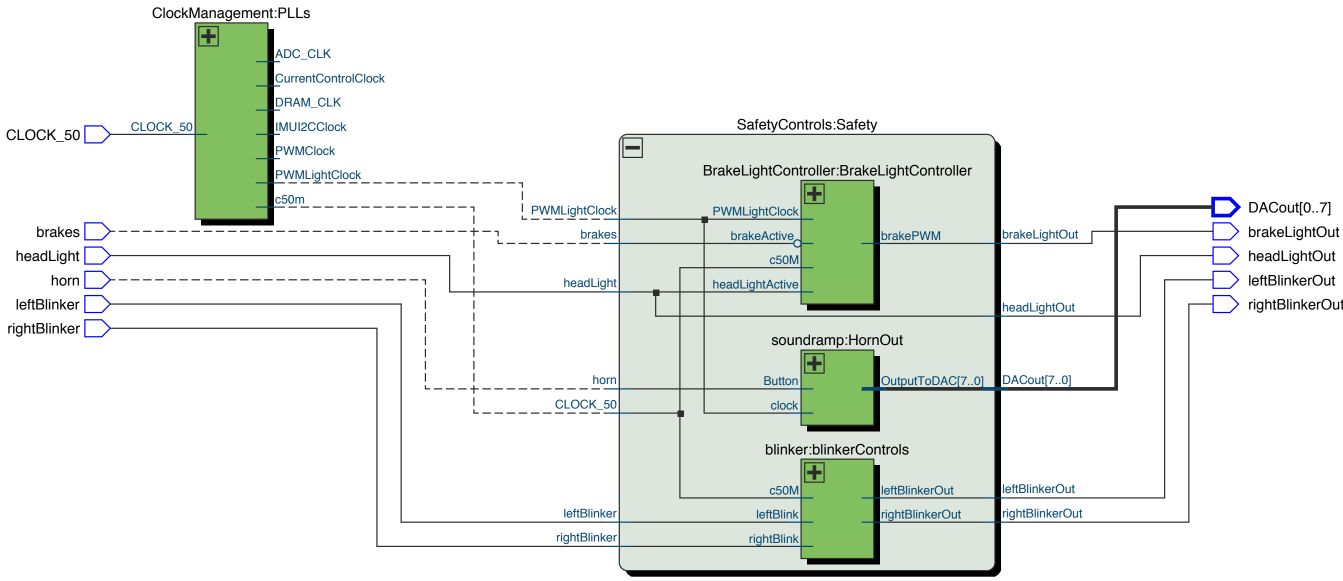


Fig. 49: Safety System Control RTL

C. Battery Pack

We needed a way to hold the batteries together in order to secure them to the bike. We created a battery pack by lining the three SLA batteries in a row and securing them together utilizing Scotch brand EXTREME strapping tape. Strapping tape has fibers embedded in it much the same way duct tape does, however the fibers are less flexible, as is the adhesive tape, creating a more rigid and secure pack. We wrapped the batteries in two directions, one over the center of the cells and again along the side of the cells. The batteries were then wired in series using short 14 gauge wires connecting the adjacent positive and negative terminals of the cells.

D. Battery Box

To protect the batteries from the elements and as a way to secure them to the bike, we determined we needed an enclosure for the batteries. The enclosure consists of an art supply box from the student campus bookstore. The box needed to be large enough to contain the battery and robust enough to hold the batteries in the box. It did not meet the second criteria on its own, so it was necessary to reinforce the box using a metal sheet. The metal sheet has holes drilled through it and the box itself so the box and metal sheet could be mounted to the rack on the bicycle. The box also contains a large metal hose clamp attached to the metal sheet to hold the batteries down. Between the batteries and the bottom of the box, we added a layer of plastic to avoid having direct contact between the cells of the batteries and the bolt heads protruding up from the bottom of the battery box.

The battery box also contains several safety features, including a fuse and main power switch for the system. They were drilled into the box and affixed using screws. On the rear of the box we have attached our taillight module, consisting of brake light and turn signals. Through the box we have routed the cabling for the taillight module, as well as the main power lines connecting to the ESC. We have also tapped into one of the cells of the battery to provide power to our FPGA box to power the goodies inside.

E. FPGA Box

The FPGA box includes all of the main electrical subsystems. They include the FPGA with IMU and ANT+ hardware modules, the audible alert system consisting of a digital to analog converter and differential output amplifier, as well as a power supply to power all of these systems and Darlington switches to power the safety lighting system from our power supply as opposed to the FPGA. To handle connections into and leaving the box we built an IO board screw terminals to provide a more robust connection to the bike than using standard male/female headers. It also allowed us to fine tune the wiring coming into the box after it was discovered the wires we made between boxes had mirrored connections between them.

F. ESC and Motor

We have created a new connector between the ESC's data header and FPGA using Molex connectors. The connector was created due to a fairly rare connector being used on the ESC that we could not order in small quantities. The connector is mated with its opposite gender and fed into the FPGA box and into the IO board.

The ESC is wired to the battery box through a three wire connector. The connector has a signal level 36 volt cable that we have routed from the battery to a switch on the headlight box and back to the ESC. This connection enables the ESC's output to the motor. Additionally, positive and negative power cables from the battery are connected to the ESC through the other two connectors present in this connector.

The ESC is connected to the motor using 14 gauge wire. The phase wires are color coded and do not necessarily connect to the matching color as reported in the laboratory prototype documentation. The correct permutation are the yellow wires are connected to green wires, and the blue connects to blue. The signal connectors for the hall sensor wires also follow this convention.

G. FPGA Box and External Systems

The FPGA box is connected to all external subsystems, and all sensors are also routed into it. The cadence sensor is zip tied to the chainstay on the non-drive side of the bike. The signal wires are fed into the IO board and are pulled low internally. The audible alert speaker is wired directly to the DAC/amplifier board, not to the IO board. The headlight box and its connections are fed into the IO board with the exception of the 36 volt line which runs from the Battery box and back. The taillight connectors are fed through the battery box and to the FPGA box. Two conductors unused for signals to the tail light module are used for the 36 volt enable signal that is routed to the headlight box and back.

X. Hardware Testing Results

TROUGH testing we find whether our selected components are capable of meeting the need for our project.

A. Bike

The motor on the bike spun out the rear dropouts holding it in place during load testing. This required us to add additional reinforcements in the form of stainless steel torque arms to hold it in place.

B. ESC

Once we repaired the damage cause by the motor spinning free of the bike, the ESC blew smoke after reconnecting it to the motor. The damage we believe was caused by the motor disconnecting while under load.

C. Battery

The internal resistance of the battery is going to result in a voltage drop of the output as a load is applied. Ideally, no internal resistance is desired. This is so the output voltage is independent of the current drawn by the load. The manufacturer specified $200 m\Omega$. Using a voltage divider consisting of two 20 watt 8 Ω resistors in series, we found the internal resistance of the batteries to be 210, 282, and 293mΩs respectively per battery.

Additionally, we found the range of the battery to be 7+ miles under full throttle conditions. We rode the bicycle with the throttle maxed out until the battery pack or one of the individual batteries dropped to 60% capacity of 37 volts for the pack or 12.2 for the individual. The initial voltage of the pack was 39.6 V. We checked voltages on cells using a multimeter at intervals of 1, 2.5, 3.5, 5, and 7.5 miles. Starting around the 3.5 mile marker, the bike began to slow down going up hills. At the 7.5 mile marker, a battery was depleted to 12.1 V. The pack's final voltage was 36.7. Extrapolated to a full charge of 40.3V fresh off the charger, this would add an extra mile or two to the range of our pack, totaling to a range under full throttle of 10 miles.

1) Current Control: As we discovered the throttle input ESC sets the speed of the bike, we determined through ride testing that this would not allow us to tailor the assistance level of our system. We decided to attempt to control current. In order to do so, we needed to treat a signal originating from the ESC that measures the current through the ESC to the motor. This was accomplished with an active filter with a gain of 60 to allow a full scale reading from the ADC on our FPGA.

D. Power Distribution Hub

1) Darlington Configuration Transistor Switches: The transistors accept the current from the FPGA and supply the required current to the lighting systems. At the base of the highest drawing load's transistor, there is a 1.4 volt difference from ground. This means there is some current flowing through the second resistor of the input voltage divider. This resistor is 10k which makes this loss negligible(About .140mW). The first resistor of the voltage divider causes much more loss equivalent to 1.9mW. This loss is acceptable and the system needs no more adjustment.

2) Buck Regulator: The testing for the buck regulator consisted of changing the loads and measuring the input/output current and the input/output voltage to determine the efficiency of the regulator and to determine if the regulator could supply the current needed.

An interesting note is the voltage measured at the loads. These voltages had a significant drop compared to the voltage measured at the output of the buck regulator. There were upwards of 5 feet of line during testing consisting of thin wire, gator clips, and a current meter. These connectors caused a drop in voltage between the output of the buck and the input of the lighting systems. Once more permanent wires are connected, the testing will need to be performed again. The efficiency is based upon the voltage measured at the end of the buck.

E. Safety System

1) Safety Lighting: The lighting modules operate at the expected 5V with some degree of tolerance to changing voltages. They were operating, albeit dimly, down to about 3.8V and possibly further. They will continue to illuminate at higher voltages, but continued overcurrent will lead to shorter life spans of the bulbs over time.

The testing for these systems was based upon visibility. I, along with Mike, took these modules outside at night and marked 100ft with increments of 25ft. I then walked out to these distances and viewed all the lighting modules from the front and from the side. All lights were easily viewable up to this distance. The turn signals became indistinguishable at around 150ft. No changes were needed.

TABLE XVII
LIGHTING VISIBILITY

Light	Distance (ft)	Visible at 0 °C	Visible at 90 °C
Headlight	25/50/75/100	all yes	all yes
Brake Light	25/50/75/100	all yes	all yes
Left Turn Signal	25/50/75/100	all yes	all yes
Right Turn Signal	25/50/75/100	all yes	all yes

2) Audible Alert: The audible alert was tested with a new speaker. At two feet, the output was at 80dB using an A-weight metering. I determined that this would not be enough, so I swapped the feedback resistor to turn it up to 11. The final result is a 95dB output indoors. Outdoors, at 25 feet it is around 76dB. At 100 feet, it was measured at 65dB. The ambient noise was measured at 56dB. It was perceivable on a quite night from Capistrano Hall with the speaker at Santa Clara, over 680 feet away.

XI. Software Testing Results

THE software testing was focused on the user experience while riding the bicycle. The primary way of ensuring this was experimental tuning of control system constants. The team iterated the adjust test cycle many times before settling on values which we feel matched the experience we had envisioned in the design phase.

A. Adapting to Speed Controller Input

Integration testing showed the commercial electronic speed controller used in our design does not operate as we thought. We interact with the controller through an analog Input signal voltage labeled "throttle." We thought this signal controlled the amount of power the controller would allow to flow through the motor. This control signal is actually mapped to a speed set point and current is a function of an internal control algorithm that maintains a given speed.

The motor's "cruise control" operation mode posed a significant problem. We needed to perform

TABLE XVI
BUCK REGULATOR EFFICIENCY

Load	Input Voltage (V)	Input Current (mA)	Output Current (mA)	Output Voltage at Load (V)	Output Voltage at Buck (V)	Efficiency (%)
Headlight	12.8	99	210	4.64	5.01	82.0
Brake Light	12.8	102	220	4.63	5.01	84.4
Left TS	12.8	64	120	4.80	5.01	73.4
Right TS	12.8	63	120	4.79	5.01	74.6
All Lights	12.8	205	460	4.16	5.01	87.8
Cell Phone	12.8	110	267	4.42	5.01	95.0
FPGA	12.8	110	240	4.75	5.01	85.4
AMP	12.8	24	50	4.90	5.01	81.5
All Systems	12.8	270	620	3.7	5.01	89.9
Avg Eff						83.78

the real-time assistance feature. The core intention of the real time assistance feature is to not significantly change the riding experience but to allow a rider to extend their range and ability. We needed to be able to control the amount of power the motor provided the rider, the amount of assistance.

The motor controller has a ground shunt resistor on the motor's driving transistors. The voltage across this known resistor value will tell us the current being drawn by the motor, which is directly related the torque output of the motor. Unfortunately, we did not have accurate values of their correlation constant nor the means to acquire them. We did have, however, a rough estimate of the amount of work the motor is doing.

We also know the user's heart rate, this tells us if the user needs more assistance or not. The current through the motor gave us a mode of control when combined with the heart rate. User's heart rate too high we need more current through the motor, too low less current. The control algorithm creates an indication of how much the motor output must be changed by subtracting the motor current signal, multiplied by a constant, from the heart rate, which is also scaled by a tuning parameter. This error value is then accumulated by a register which makes up that cycle's motor output value.

This control scheme implements the intent of the real time assistance feature. We can sense the user's exertion level and adjust the amount of work the motor does with enough accuracy that its actions are not apparent to the user.

B. Mobile Application

The mobile application was being unofficially tested throughout the semester. Any time we wanted to test our assistance or current control algorithm, we had to set the target heart rate from the cell phone. Consequently, we obtain many hours of valuable test data that lead to small changes to the application and the Verilog interface. We also dedicated time specifically to testing the application and Verilog state machine with the following procedures and results.

1) Bluetooth Data Transmission: I tested the connection between the FPGA and the cell phone by attempting to connect from the mobile application at different distances. I also left the display loop running and checked that the displayed numbers were still updating correctly. Outdoors we were able to connect and send/receive data reliably up to 150ft before losing data transmissions and being unable to connect. The distance is sufficient but if the connection is lost, the user has to reconnect before the display begins again.

a) Procedure: Using Sources and Probes on the inclination angle on the FPGA, the cell phone application, and a level to compare them against.

TABLE XVIII
MOBILE APP. INCLINATION VERIFICATION

Angle(Degrees)	Displayed Value	FPGA
-90	-76	-212
-60	-71	-198
-45	-50	-140
-25	-28	-78
-15	-15	-42
-10	-10	-28
-0	0	0
10	10	28
15	15	42
20	21	58
25	28	78
45	50	140
60	71	198
90	76	212

TABLE XIX
USER INPUT VALIDATION

Input	Received as Heart Rate	Received as Wheel Size
0	0	0
1	1	1
20	20	20
50	50	50
120	120	120
150	150	150
200	200	200
256	0	0
1000	232	232
-100	cannot enter symbol	cannot enter symbol
A	cannot enter symbol	cannot enter symbol
!	cannot enter symbol	cannot enter symbol
2.56	cannot enter symbol	cannot enter symbol
No value	Crash	Crash

b) Results: The results were good for the angles we will be concerned with. They are accurately measured for the angles less than 25 degrees. The FPGA data contains larger numbers simply because we wanted more resolution there. We used a Taylor series approximation for the tangent, and had to divide by 2.8 on the phone to get the units into degrees so the rider understands the value.

2) User Input Validation: I tested the user input by pressing all possible "active areas," or buttons, in different combinations. If the devices were not connected to each other, the application would crash when trying to begin the display loop, or attempting to sent the heart rate cap or wheel size to the FPGA. This was fixed by creating a connection flag in the Java to disable the buttons when not connected. I also tested the heart rate cap and wheel size inputs by sending different values to the FPGA and watching what it receives in the Sources and Probes tools. The only problem that occurs is when the user presses the send button when the fields are blank. The application freezes while waiting for a response from the FPGA which will never come because the FPGA didn't receive anything. This will be fixed by checking the user input before allowing the application to try to send. We will also only allow for numbers between 0 and 255 since it is only sending a single byte, and the neither the heart rate cap or the wheel size should be out of those bounds.

a) Procedure:

- Connect FPGA to computer.
- Open Quartus and Program the FPGA. We put probes on the received byte in the cell phone module as well as the heart rate cap and wheel size bytes.
- Open Sources and Probes. Once the probes have been activated, it will display the values in the registers.
- Open the cell phone application, press the connect button, type the number in the correct field, then press the set button.
- Write down what the FPGA receives in corresponding bytes.

b) Results: All of the numbers from 0-255 are sent correctly to the FPGA. If 256 is sent from the phone, then the FPGA only receives a 0. This is because the it is only sending 8 bits for these values. Both the wheel size and heart rate cap should never realistically receive values outside of 0-255.

3) Data Logging: After connecting the cell phone with the FPGA, the received data is logged into a CSV file for later referencing. We made sure the time stamps for each set of data were correct and the logged values matched up with the displayed values as well as the internal values on the FPGA. While testing this we noticed that it will log a "0" if some error in communication has occurred. This happens rarely, but is very apparent when graphing the results. I fixed it by only logging a zero if two zeros are received in a row, otherwise log the last correct value. Also, since the current to the

Fig. 52: Team 3 with the Completed Prototype

motor is constantly fluctuating very rapidly and the cellphone is only receiving new data twice a second, the ADC reading is very jumpy. I want to implement a half second moving average of the current from the FPGA to display to the user

4) Stability of Display Loop: There have been times throughout testing when we made changes to separate parts of the Verilog, and somehow the FPGA would stop replying to the cell phone. This causes the display function to stay in a while loop constantly writing a request to the FPGA for data. We have not found a definite solution for this problem, but it seems to be timing related. So we will be spending time constraining all of our paths and further testing this.

XII. Conclusion

Once the testing was finally complete, we finished our adjustments and repairs, demonstrated our final product, and began the final documentation. Here we are, at the end of this incredible experience, looking back at all that we've accomplished and trying to transform the memories into words.

Project Forward represents the result of a nine month build process where we, a group of engineering students, took hold of the societal problem of inactivity and explored the potential for a technological solution to this problem. We came up with a smart electrically assisted bicycle that utilizes biometric and terrain feedback to provide assistance to a rider, enabling them to consider cycling based modes of transport as an effective means of commuting while simultaneously incorporating physical activity into their daily lives. The project addresses not just the physical aspects of the commute, but also addresses safety concerns of riders with the inclusion of a rider awareness system, incorporating lighting and sound.

The build process was planned and broken down into tasks which comprise the laboratory and deployable prototypes. The laboratory prototype proved the concept, functionality and interoperability of these systems. The deployable prototype demonstrated the system as a product that can be used in the field. During the first phase, we struggled to adapt to this new type of work progression and create the individual subsystems for our project.

For the laboratory prototype we designed and built all of our systems to satisfy the feature set we had laid out. Once we knew all the systems operated properly on their own, we combined tem to demonstrate system integration. This allowed us to demonstrate that we had viable system.

Leading up to the deployable prototype, a test plan was devised. After testing, modifications needed to be made for some of the systems. One such modification was made after the discovery of our motor controller setting speed and not assistance. Our solution was to design and implement an electrical current based control using both hardware and software.

The successful implementation of our design has led us to a fork in the road where we consider whether or not to market our design and monetize the effort we put in. Communication with our peers, friends, and family have been supportive of this undertaking, and our market research has shown that the E-Bike market of the United States has not saturated and may be looking for the right spark to launch it into the mainstream, such as has already taken place in markets such as China or Western Europe.

Moreover, the project has helped in the transformation of our team members from green students to budding engineers. This development of skill sets was not limited to technical tasks relating to hardware and software, but enhanced our problem solving, and communications skills, preparing us for the transition into professional engineering careers. During these nine months together we have had so many memorable experiences, both the good and the bad. This has been an invaluable step towards applying our education, and each one of us is extremely proud to be a part of Project Forward.

References

[1] W. H. Organization. (2013) Overweight: Situation and trends. [Online]. Available: http://www.who.int/gho/ncd/risk_factors/overweight_text/en/index.html

[2] A. of the European Two-Wheeler Parts' and A. Industry, "Industry and market profile," *Industry and Market Profile*, 2013.

[3] P. Research. (2013) Global market opportunities, barriers, technology issues, and demand forecasts for e-bicycles, pedal-assist bicycles, and e-bicycle batteries and motors. [Online]. Available: http://www.navigantresearch.com/research/electric-bicycles

[4] W. H. Organization. (2013, Mar.) Obesity and overweight. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs311/en/

[5] ——, *Depression*, October 2012. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs369/en/index.html

[6] E. of Depression, *Health benefits of physical activity: the evidence*, 1993. [Online]. Available: http://www.cmaj.ca/content/174/6/801.full.pdf+html

[7] W. H. Organization. (2009) Society at a glance. [Online]. Available: http://www.oecd.org/berlin/42675407.pdf

[8] U. N. L. of Medicine. (2010) Sedentary behaviors increase risk of cardiovascular disease mortality in men. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2857522/

[9] A. global breif on hypertension. (2010) Sedentary behaviors increase risk of cardiovascular disease mortality in men. [Online]. Available: http://apps.who.int/iris/bitstream/10665/79059/1/WHO_DCO_WHD_2013.2_eng.pdf

[10] W. H. Organization. (2010) Sedentary behaviors increase risk of cardiovascular disease mortality in men. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs317/en/index.html

[11] R. D. to Inactivity in Physically Capable Older Adults. (2006) Health benefits of physical activity: the evidence. [Online]. Available: http://www.cmaj.ca/content/174/6/801.full.pdf+html

[12] C. L. H. Ian H. Gotlib, *Epidemiology of Depression*, 1993. [Online]. Available: http://www.cmaj.ca/content/174/6/801.full.pdf+html

[13] U. T. R. Container, "Analysis of bicycling trends and policies in large north american cities," Tech. Rep., 2011. [Online]. Available: http://www.utrc2.org/sites/default/files/pubs/analysis-bike-final_0.pdf

[14] U. D. of Transportation. (2013) Traffic safety facts. [Online]. Available: http://www-nrd.nhtsa.dot.gov/Pubs/811743.pdfl

[15] C. KOMANOFF and M. J. SMITH. (2000) The only good cyclist. [Online]. Available: http://www.rightofway.org/research/cyclists.pdfl

[16] O. Bolton, "Electrical bicycle," Dec. 31 1895, uS Patent 552,271. [Online]. Available: http://www.google.com/patents/US552271

[17] EcoBusiness. (2013) Electric bike sales soar worldwide. [Online]. Available: http://ecowatch.com/2013/11/06/electric-bike-sales-soar-worldwide/

[18] Superpedestrian. (2014) The copenhagen wheel. [Online]. Available: https://www.superpedestrian.com/

[19] C. D. of Motor Vehicles. (2014) V c section 25950 color of lamps and reflectors. [Online]. Available: https://www.dmv.ca.gov/pubs/vctop/d12/vc25950.htm/

[20] Garmin. (2013) Soft premium heart rate strap. [Online]. Available: https://buy.garmin.com/en-US/US/shop-by-accessories/fitness-sensors/soft-strap-premium-heart-rate-monitor/prod15490.html

[21] A. Devices. (2013, Mar.) Moving average filters. [Online]. Available: http://www.analog.com/static/imported-files/tech_docs/dsp_book_Ch15.pdf

XIII. Glossary

1) FPGA: A Field Programmable Gate Array is a re-programmable array of logic elements that can be used to represent digital logic. These devices are configured using a hardware descriptive language like Verilog or VHDL.
2) HDL: Hardware descriptive language that is used to program the FPGA.
3) ANT+: Communication standard for a wireless protocol in sensor network technology.
4) UART: Universal Asynchronous Receiver/Transmitter. Translates between serial and parallel.
5) I2C: Inter-Integrated Circuit. A 2 wire multi master serial communication bus.
6) Deterministic: In the realm of computer engineering a deterministic process will take the same time complete given a set of initial conditions.
7) BLDC Motor: A DC electric motor without internal brushes for contact points.
8) ESC: Electronic Speed Controller. Device that controls the input to the BLDC motor.
9) PWM: Pulse width modulation. Allows for the duty cycle of a square wave to be adjusted.
10) SLA: Sealed Lead Acid. A common type of battery that is cheap and rugged.
11) Parallelism: The ability to perform multiple operations at once.
12) Phase Locked Loop: Analog circuitry that generates an output signal whose phase is related to the phase of an input signal
13) DAC: Digital to Analog Converter.
14) R2R: DAC topology made from resistor ladder network.
15) DMV: Department of Motor Vehicles
16) DoT: Department of Transportation
17) PCB: Printed Circuit Board
18) Perfboard: Material for prototyping electric circuits. Contains isolated copper pads for soldering.
19) LED: Light Emitting Diode
20) USB: Universal Serial Bus
21) BJT: Bipolar Junction Transistor
22) MOSFET: MetalOxideSemiconductor Field-Effect Transistor
23) RTL: Register Transfer Level

XIV. Appendices

A. Included Datasheets

This project used many devices which all come with their own documentation. We have compiled the most pertinent of this literature and included it on the ECS Hive site. A listing of the individual files is below in Table XX

TABLE XX
OVERVIEW OF INCLUDED DATASHEETS

Document Name	Device	subsystem device is found in
9DoF-Stick-v13	Sensor stick	Control system
ADC128S022	ADC for control system	Control system
ADXL345	Accelerometer	Control system
Arduino Bluetooth Matev13	Bluetooth Module Schematic	User Interface
Bluetooth-RN-41-DS	Bluetooth Module Datasheet	User Interface
Cyclone IV Device Datasheet	FPGA	Control system
ANT Message Protocol and Usage	ANT protocol	Control system
Bluetooth-RN-41-DS	Bluetooth Module Datasheet	User Interface
DE0 Nano User Manual	FPGA Development Board	Control system
de0-nano-c4-rev-c	FPGA Development Board Shematic	Control system
HMC5883L-FDS	Gyroscope	Control system
IS42S16160B	SDRAM Datasheet	Control system

B. HDL to FPGA Pin Mapping

TABLE XXI
FPGA Pin Assignment Overview

Pin Name/Usage	FPGA Location	Direction
rx	A4	input
waveFormsPin	A5	output
leftBlinker	A7	input
ADC_SDAT	A9	input
ADC_CS_N	A10	output
LED[3]	A11	output
DACout[6]	A12	output
LED[1]	A13	output
LED[0]	A15	output
LED[6]	B1	output
tx	B5	output
cadence	B7	input
ADC_SADDR	B10	output
DACout[3]	B11	output
DACout[7]	B12	output
LED[2]	B13	output
ADC_SCLK	B14	output
epcs_sdo	C1	output
DRAM_WE_N	C2	output
GND	C5	gnd
brakes	C6	input
blips	C8	input
DACout[0]	C9	output
DACout[4]	C11	output
LED[4]	D1	output
epcs_sce	D2	output
rightBlinker	D8	input
brakeLightOut	D9	output
DACout[5]	D11	output
EnableAmplifier	D12	output
VCCD_PLL2	D13	power
PWMout	E6	output
headLight	E7	input
leftBlinkerOut	E8	output
rightBlinkerOut	E9	output
DACout[1]	E10	output
DACout[2]	E11	output
LED[5]	F3	output
horn	F8	input
headLightOut	F9	output
IMU_SCL	F13	bidir
DRAM_DQ[1]	G1	bidir
DRAM_DQ[0]	G2	bidir
epcs_dclk	H1	output
epcs_data0	H2	input
DRAM_DQ[6]	J1	bidir
DRAM_DQ[5]	J2	bidir
ANT_Reserved2	J13	output
ANT_RequestToSend	J14	output
ANT_BaudRate[2]	J16	output
DRAM_DQ[15]	K1	bidir
DRAM_DQ[4]	K2	bidir
DRAM_DQ[3]	K5	bidir
ANT_Reserved1	K15	output

TABLE XXII
Continued FPGA Pin Assignment Overview

Pin Name/Usage	FPGA Location	Direction
DRAM_CAS_N	L1	output
DRAM_RAS_N	L2	output
LED[7]	L3	output
DRAM_ADDR[12]	L4	output
DRAM_CKE	L7	output
DRAM_DQ[2]	L8	bidir
ANT_BaudRate[0]	L13	output
ANT_BaudRate[1]	L14	output
ANT_nTest	L15	output
DRAM_BA[1]	M6	output
DRAM_BA[0]	M7	output
DRAM_ADDR[3]	M8	output
ANT_rx	M10	input
DRAM_ADDR[11]	N1	output
DRAM_ADDR[10]	N2	output
DRAM_DQ[14]	N3	bidir
DRAM_ADDR[1]	N5	output
DRAM_ADDR[2]	N6	output
DRAM_ADDR[6]	N8	output
ANT_tx	N14	output
ANT_Sleep	N16	output
DRAM_ADDR[9]	P1	output
DRAM_ADDR[0]	P2	output
DRAM_DQ[13]	P3	bidir
DRAM_CS_N	P6	output
DRAM_ADDR[4]	P8	output
ANT_PortSelect	P14	output
ANT_nReset	P15	output
DRAM_ADDR[8]	R1	output
DRAM_DQ[11]	R3	bidir
DRAM_CLK	R4	output
DRAM_DQ[12]	R5	bidir
DRAM_DQM[0]	R6	output
DRAM_DQ[7]	R7	bidir
CLOCK_50	R8	input
ANT_nSuspend	R14	output
DRAM_DQ[9]	T2	bidir
DRAM_DQ[10]	T3	bidir
DRAM_DQ[8]	T4	bidir
DRAM_DQM[1]	T5	output
DRAM_ADDR[7]	T6	output
DRAM_ADDR[5]	T7	output
IMU_SDA	T15	bidir

C. FPGA HDL Overview

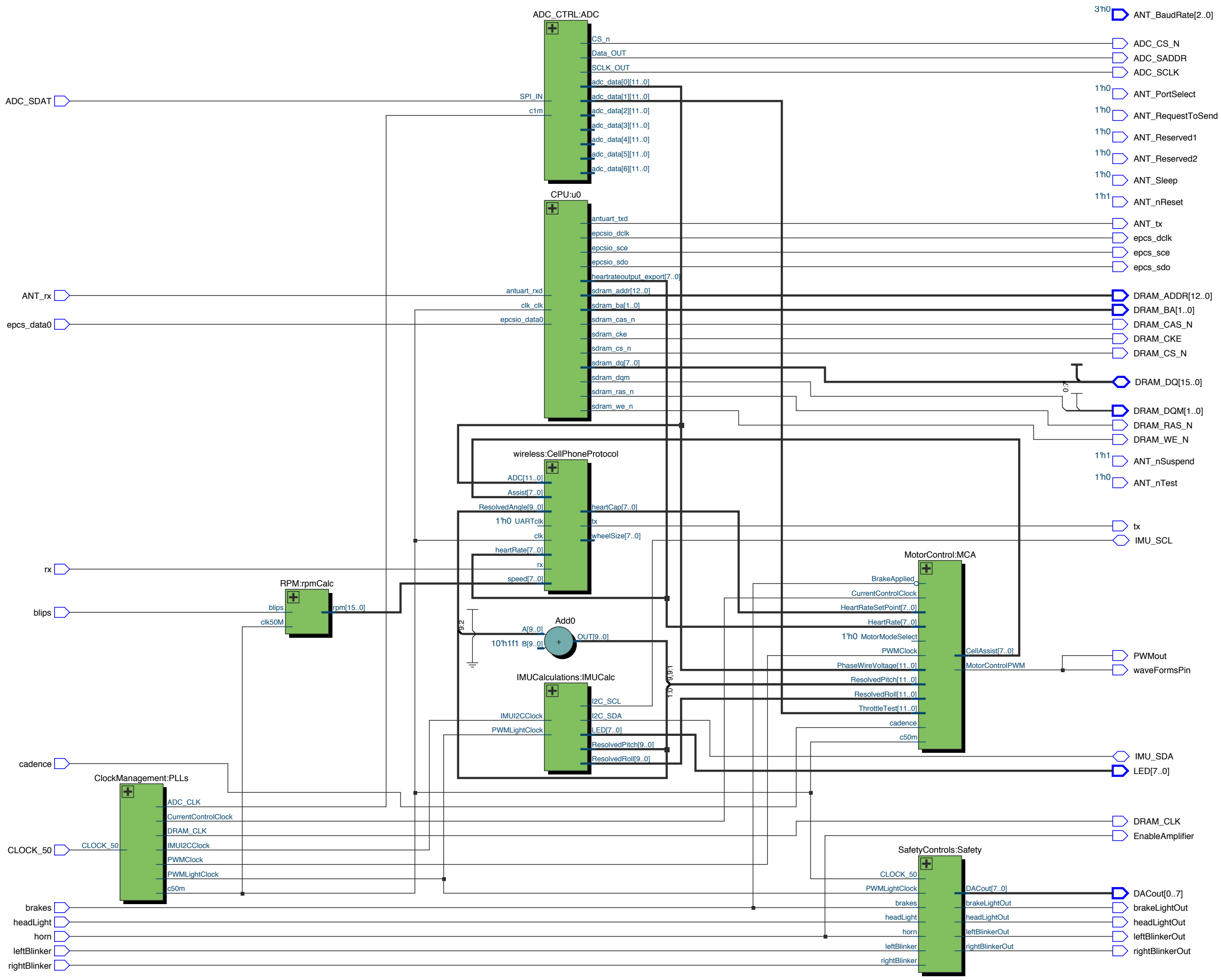
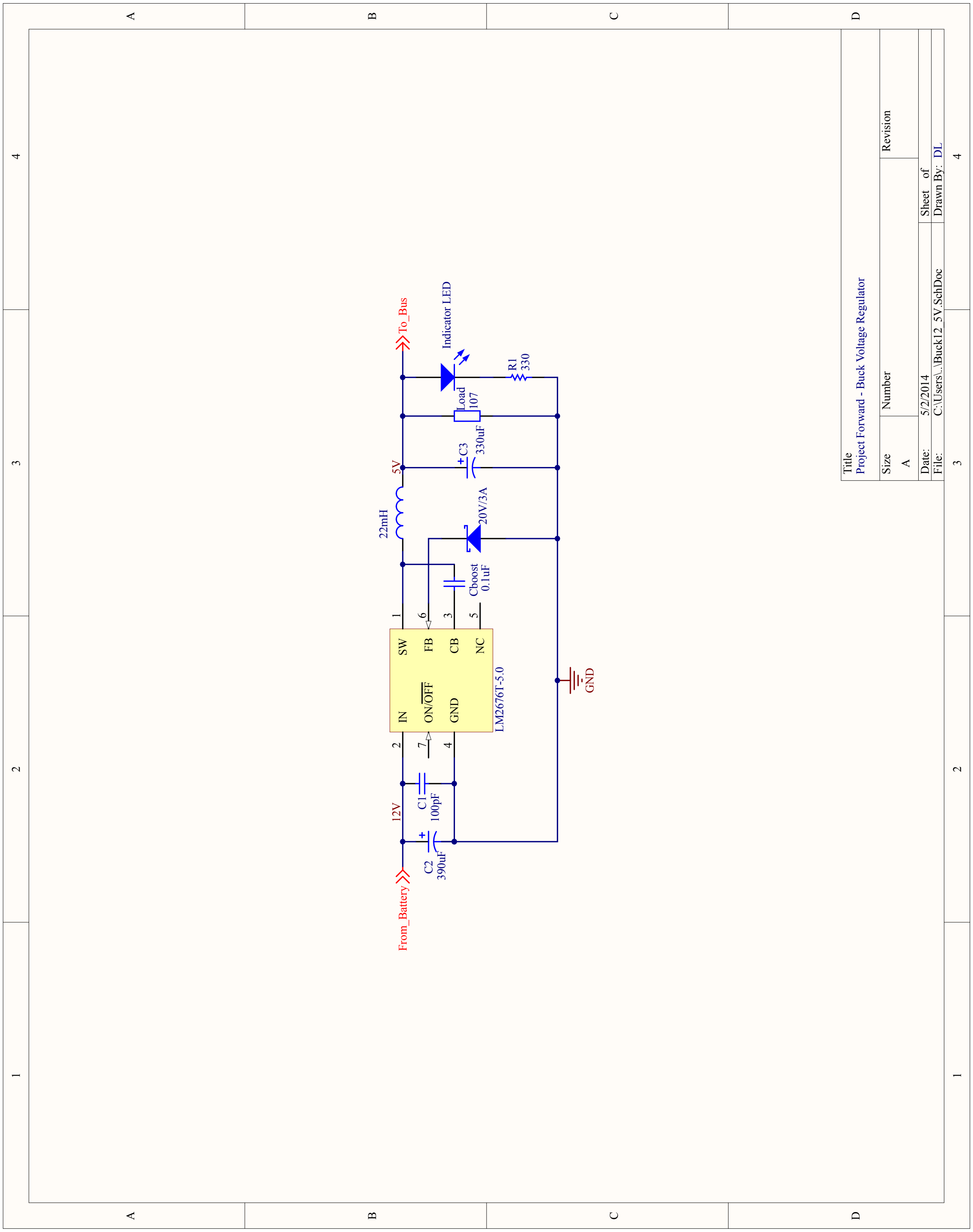
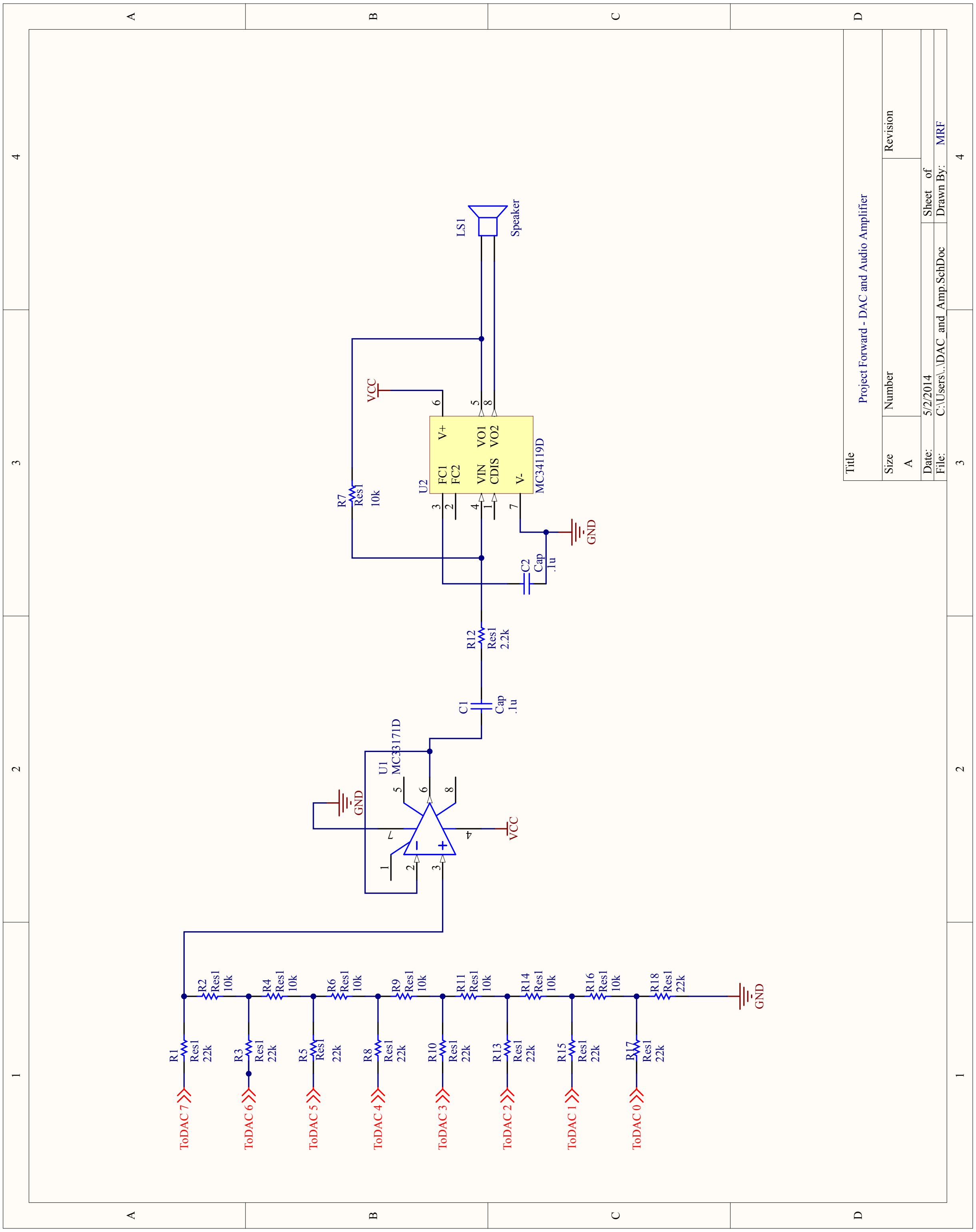


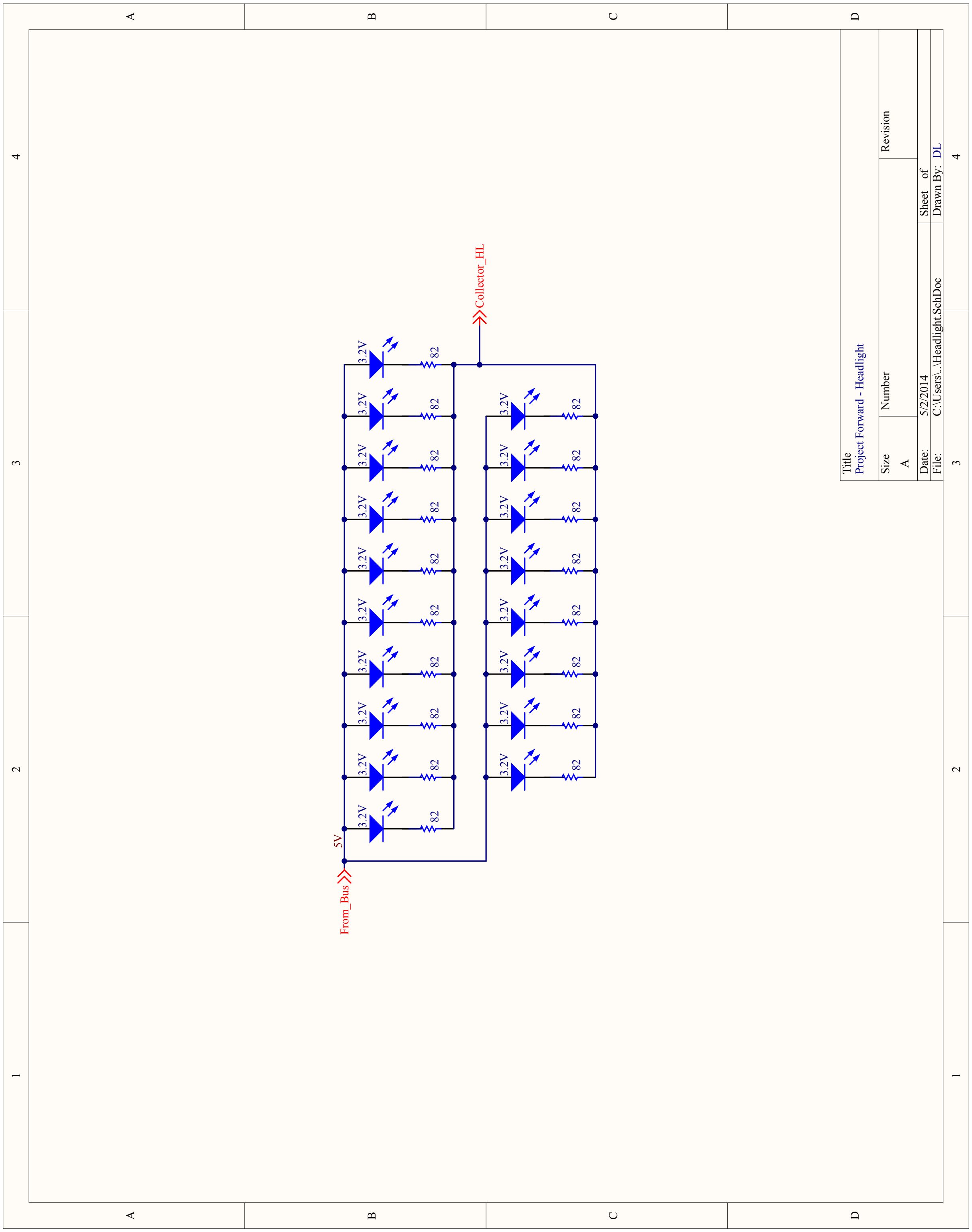
Fig. 53: Full View of FPGA Logic as RTL

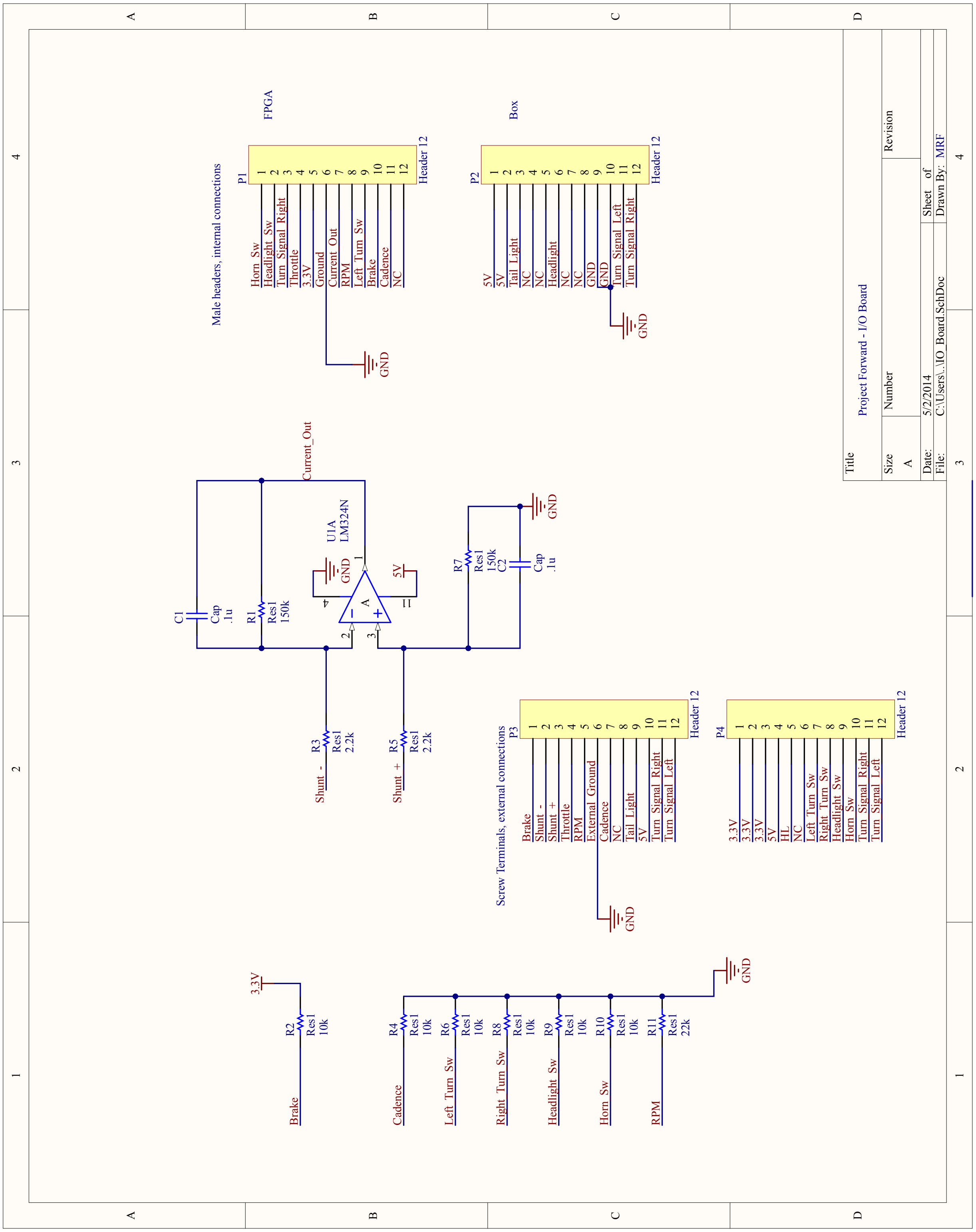
D. Schematics

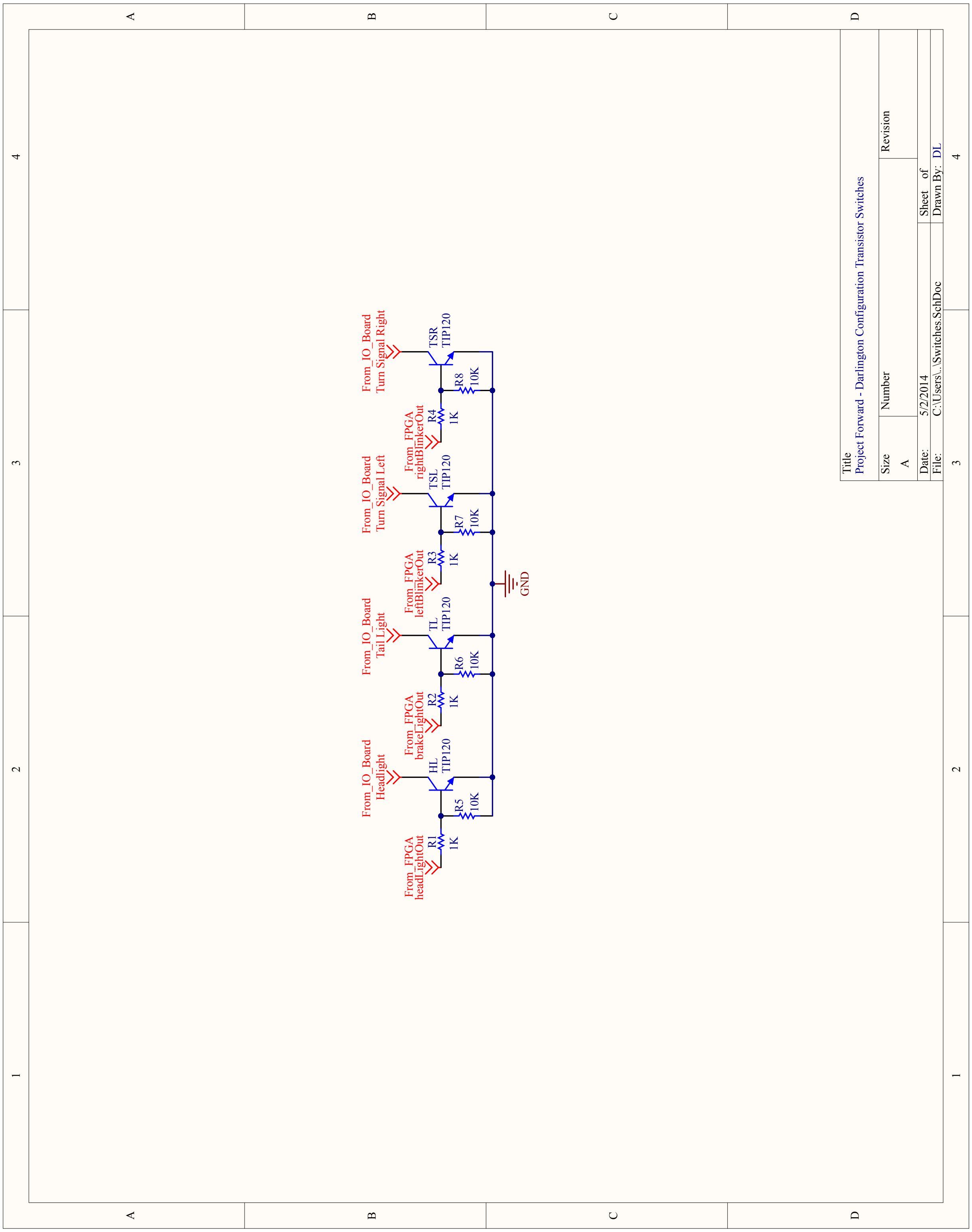


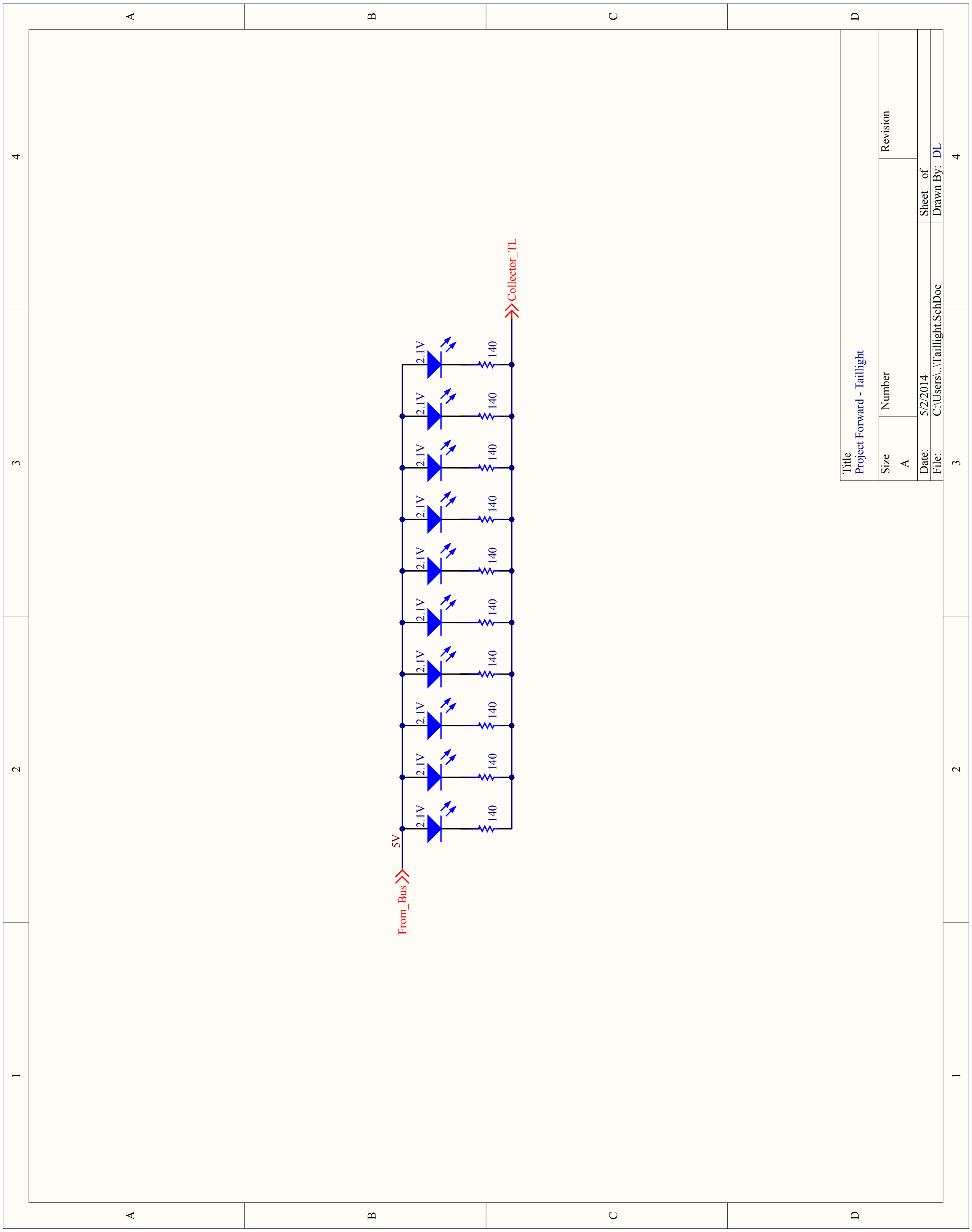


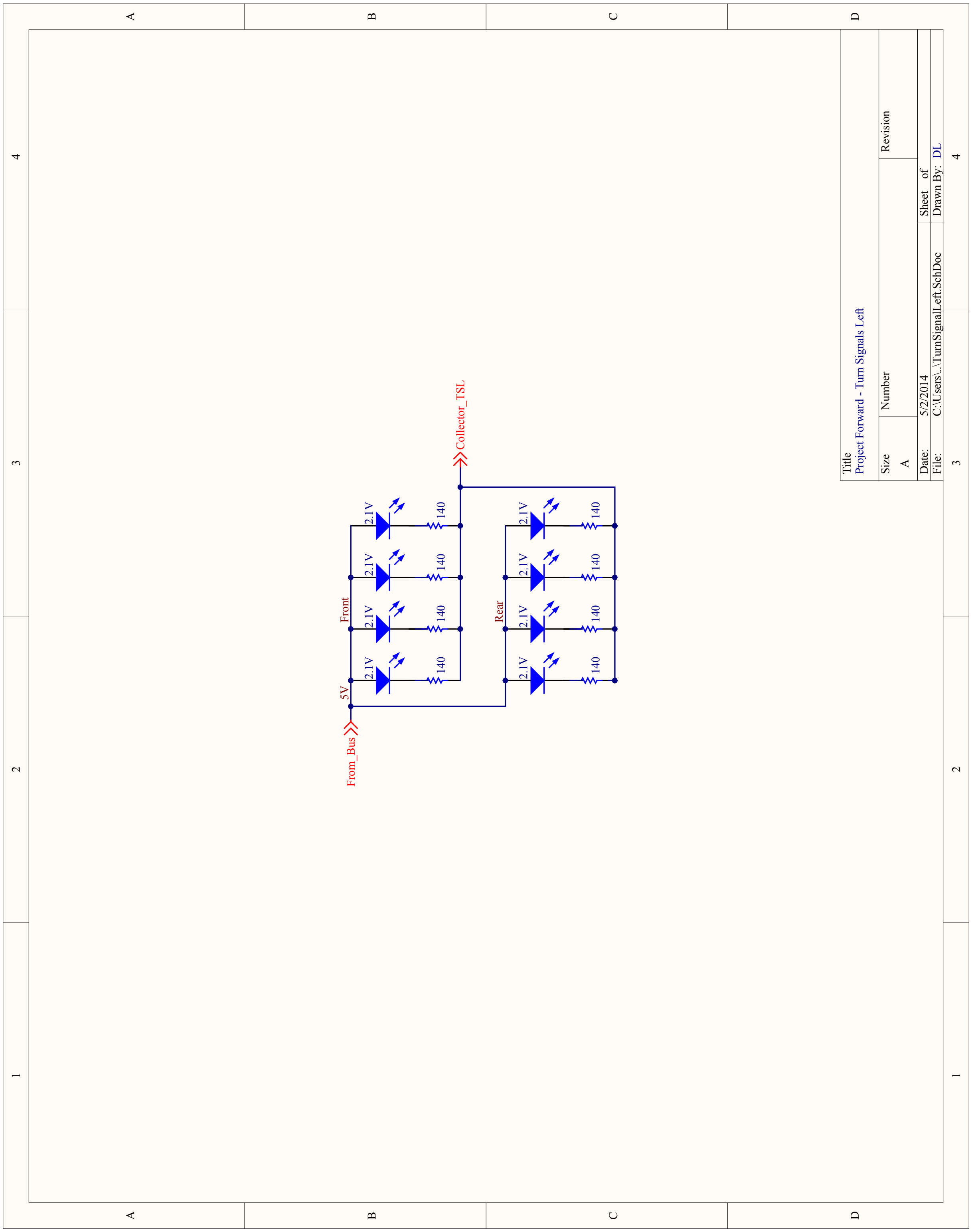
ToDAC 7
ToDAC 6
ToDAC 5
ToDAC 4
ToDAC 3
ToDAC 2
ToDAC 1
ToDAC 0
R1 Res1 22k
R2 Res1 10k
R3 Res1 22k
R4 Res1 10k
R5 Res1 22k
R6 Res1 10k
R8 Res1 22k
R9 Res1 10k
R10 Res1 22k
R11 Res1 10k
R13 Res1 22k
R14 Res1 10k
R15 Res1 22k
R16 Res1 10k
R17 Res1 22k
R18 Res1 22k
GND
U1 MC33171D
VCC
GND
C1 Cap .1u
R12 Res1 2.2k
C2 Cap .1u
R7 Res1 10k
U2
FC1
FC2
VIN
CDIS
V-
V+
VO1
VO2
MC34119D
VCC
GND
LS1
Speaker
Title
Project Forward - DAC and Audio Amplifier
Size
A
Number
Revision
Date: 5/2/2014
File: C:\Users\..\DAC and Amp.SchDoc
Sheet of
Drawn By: MRF

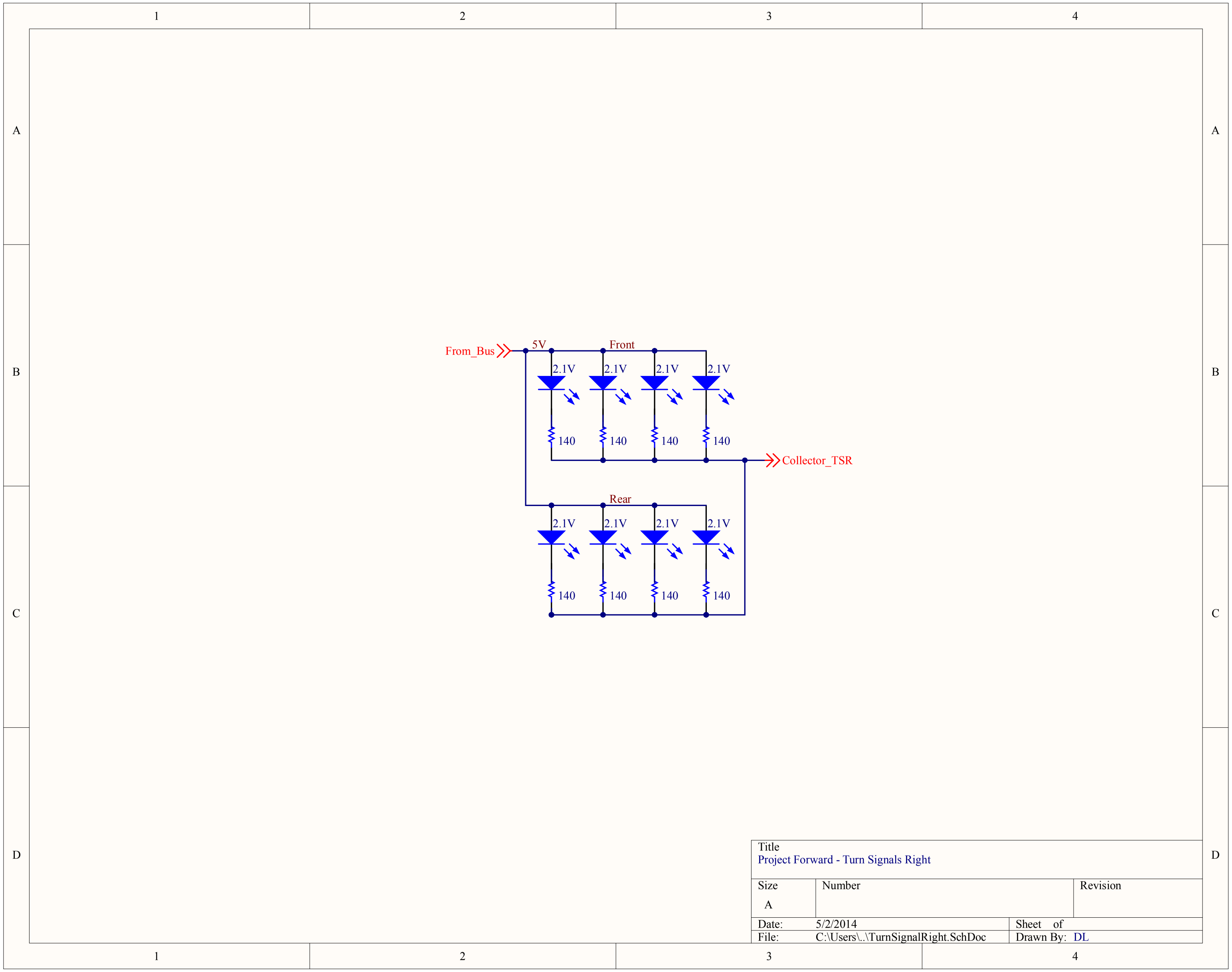

From_Bus
5V
Collector_HL
3.2V
82
Title
Project Forward - Headlight
Size
A
Number
Revision
Date: 5/2/2014
Sheet of
File: C:\Users\..\Headlight.SchDoc
Drawn By: DL


Male headers, internal connections
FPGA
P1
Header 12
Horn Sw
Headlight Sw
Turn Signal Right
Throttle
3.3V
Ground
Current Out
RPM
Left Turn Sw
Brake
Cadence
NC
GND
Box
P2
5V
5V
Tail Light
NC
NC
Headlight
NC
NC
GND
GND
Turn Signal Left
Turn Signal Right
Current_Out
U1A
LM324N
5V
C1
Cap
.1u
R1
Res1
150k
R7
Res1
150k
C2
Cap
.1u
R3
Res1
2.2k
Shunt -
R5
Res1
2.2k
Shunt +
Screw Terminals, external connections
P3
Brake
Shunt -
Shunt +
Throttle
RPM
External Ground
Cadence
NC
Tail Light
5V
Turn Signal Right
Turn Signal Left
P4
3.3V
3.3V
3.3V
5V
HL
NC
Left Turn Sw
Right Turn Sw
Headlight Sw
Horn Sw
Turn Signal Right
Turn Signal Left
3.3V
R2
Res1
10k
Brake
R4
Res1
10k
Cadence
R6
Res1
10k
Left Turn Sw
R8
Res1
10k
Right Turn Sw
R9
Res1
10k
Headlight Sw
R10
Res1
10k
Horn Sw
R11
Res1
22k
RPM
Title
Project Forward - I/O Board
Size
A
Number
Revision
Date: 5/2/2014
Sheet of
File: C:\Users\..\IO Board.SchDoc
Drawn By: MRF

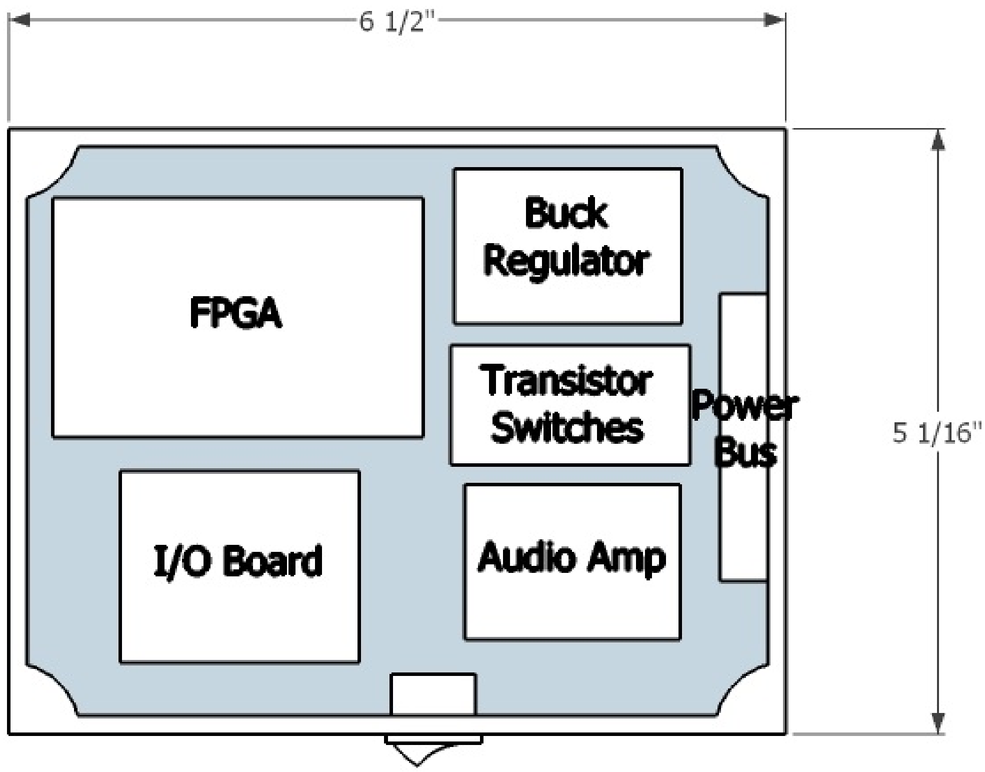

From_FPGA
headLightOut
R1
1K
R5
10K
HL
TIP120
From_IO_Board
Headlight
From_FPGA
brakeLightOut
R2
1K
R6
10K
TL
TIP120
From_IO_Board
Tail Light
From_FPGA
leftBlinkerOut
R3
1K
GND
R7
10K
TSL
TIP120
From_IO_Board
Turn Signal Left
From_FPGA
rightBlinkerOut
R4
1K
R8
10K
TSR
TIP120
From_IO_Board
Turn Signal Right
Title
Project Forward - Darlington Configuration Transistor Switches
Size
A
Number
Revision
Date: 5/2/2014
Sheet of
File: C:\Users\..\Switches.SchDoc
Drawn By: DL

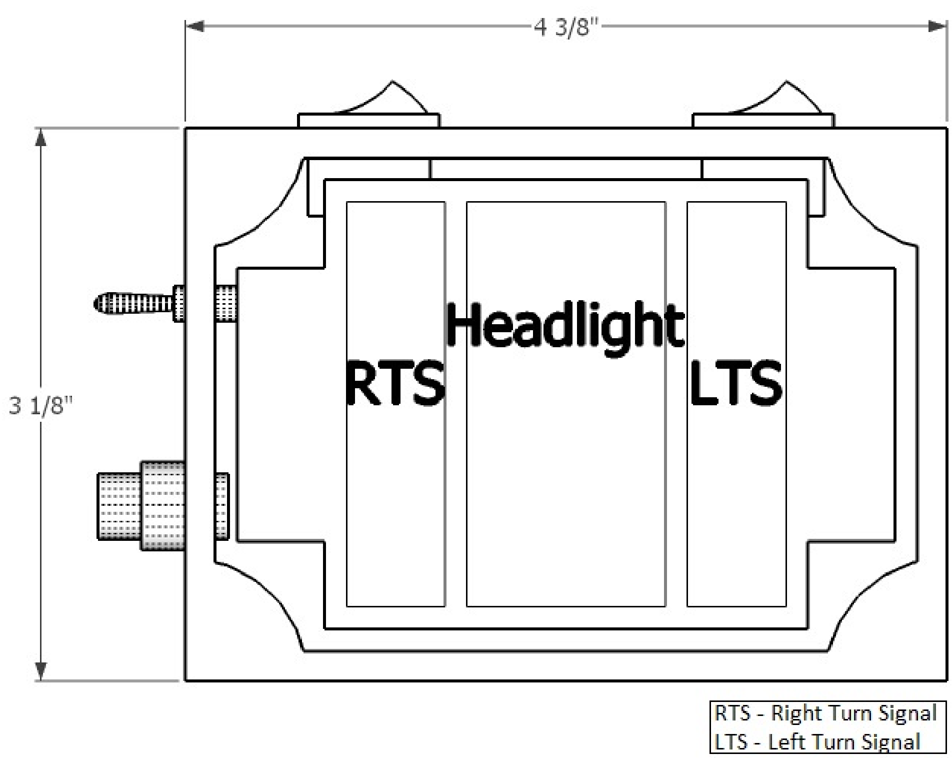

From_Bus
5V
2.1V
140
Collector_TL
Title
Project Forward - Taillight
Size
A
Number
Revision
Date: 5/2/2014
File: C:\Users\..\Taillight.SchDoc
Sheet of
Drawn By: DL

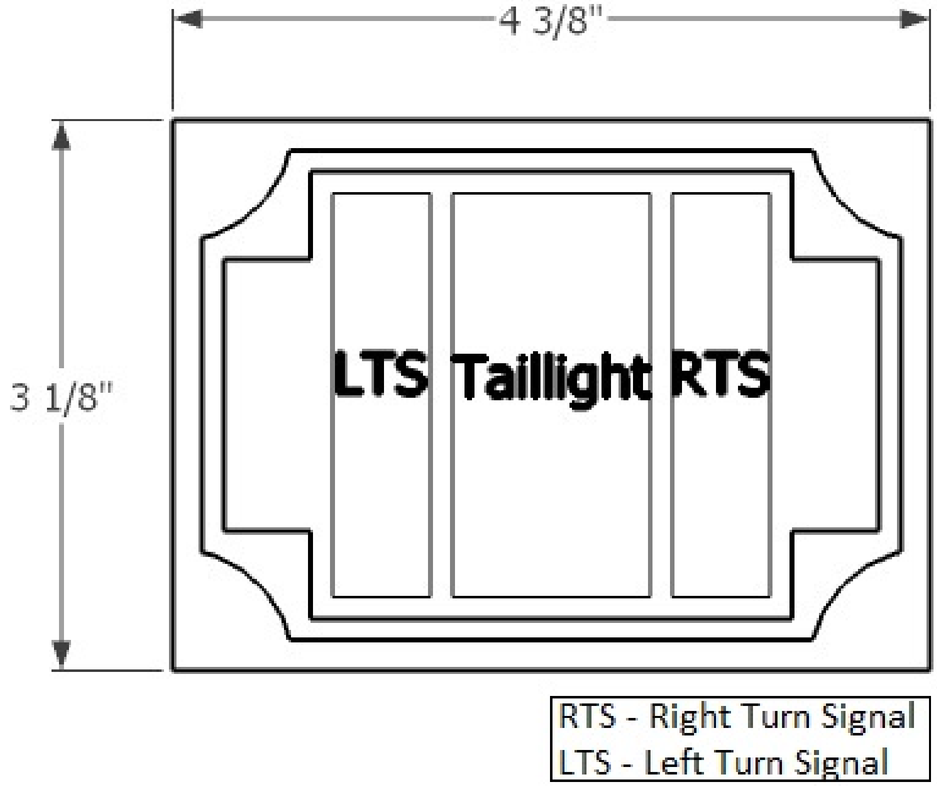

From_Bus
5V
Front
Rear
Collector_TSL
2.1V
140
Title
Project Forward - Turn Signals Left
Size
A
Number
Revision
Date: 5/2/2014
File: C:\Users\..\TurnSignalLeft.SchDoc
Sheet of
Drawn By: DL


From_Bus
5V
Front
Rear
2.1V
140
Collector_TSR
Title
Project Forward - Turn Signals Right
Size
A
Number
Revision
Date: 5/2/2014
Sheet of
File: C:\Users\..\TurnSignalRight.SchDoc
Drawn By: DL

E. Hardware Drawings






4 3/8"
3 1/8"
LTS
Taillight
RTS
RTS - Right Turn Signal
LTS - Left Turn Signal

XV. Group Member Resumes

Ben Smith

Objective: To obtain a Intern position in the field of Embedded Systems for the summer of 2014 leading to full time employment after graduation in December 2014

Work Experience

Miranda Technologies — Summer 2013
Hardware Engineering Intern

- Microcontroller and FPGA
- Bus arbitration
- Deterministic Systems
- 3GB/s SDI Video
- System level debugging
- Prototype development

Worked with a team of interns to develop an end-of-life replacement product through derivative design. The specification required compatability with a legacy control system that caused a failure of unknown origin. Enabling the control involved debugging multiple, deterministic, control channels based on a number of communication schemes.

Mechanic, sales at Bicycle Emporium — 2008 – Current
Boutique bicycle retail

Sound Technician at Underground Cafe — 2005 – 2008
Live sound and stage management

Manager at River Rat Raft and Bike — 2004 – 2008
Mechanic, Daily operations and payroll

Projects

Senior Project:
Electrically assisted bicycle with biometric feedback

- FPGA control system
- FIR Data filtering
- I2C bus implementation
- IMU processing

Developed closed loop biometric control system for an electric bicycle. System varies the amount of electrical assistance based on the user's heart rate.

Cereal Hack 3:
An inexpensive open source implementation of a motion capture system was developed during a Forty eight hour rapid prototyping competition. An accelerometer, gyroscope, and rotary encoders were used to digitize movement.

Self balancing FPGA robot:
Accelerometer based PID control system which balanced a robot on two wheels. Solution very similar to inverted pendulum problem

CPE64 Verilog labs:
Modernized introductory logic design laboratory to use new Altera development board based on the Cyclone IV FPGA.

PCB design for Hornet Racing:
Interdisciplinary work with Formula SAE team mechanical engineers to develop a electrical design and PCB for a brake light to FSAE standards.

Benjamin.Smith.cs@google.com

Linked In goo.gl/UJQGMY
Github goo.gl/JRDCHP

Engineering Skills

Programmable Logic	Altera, Xilinx FPGA/CPLD Logic analyzer debugging Mentor Graphics Modelsim
HDL Languages	Verilog, System Verilog
Microcontrollers	Microchip, Atmel NIOS II softprocessor
Bus architectures	SPI, I^2C, UART
Programming Languages	Intermediate: C, Java Basic: Python
EDA Technologies	Altium,Mentor DX Designer Cadence Orcad Autodesk Autocad
Sensor Development	Accel., Gyro., Mag. Fusion algorithms for IMUs
Digital Signal Processing	FIR/IIR Transfer function realization
Misc. Languages	Advanced: LaTeX Intermediate: BASH

Education

2008 – Present	Senior Year B.S. (GPA: 3.5) Computer Engineering *CSU, Sacramento*
Miranda Technologies	DX Designer EDA training
Altera	FPGA designer curriculum

Awards

2014	**Tau Beta Pi Officer** *California Upsilon Chapter*
2013	**Jeff Evans Memorial Scholarship** *Miranda Technologies*
2013	**Second Place: Cereal Hack 3** *The Hackerlab, Sacramento*

David J Larribas

engineer@email.com (555)555-5555 www.linkedin.com/in/engineer/

Objective: Obtain an internship in the field of Electronics Engineering with Barco which will utilize my technical and soft skills.

Qualifications

Experience with: PSpice • MATLAB • Microcontrollers • Oscilloscope • Function Generator • C Code • Quartus II • Analog Discovery • SketchUp • AutoCAD

- Strong leadership and team skills developed and refined through engineering projects, music performance, clubs and organizations.
- Advanced writing skills through weekly project reports and other professional documentation utilizing MS Project, Google Docs, Lucidchart, and LaTeX..
- Practiced presentation skills developed through presentation competitions and periodic project presentations utilizing PowerPoint and Prezi.

Project Experience

Smart Electrically Assisted Bicycle

Novel control system for electrically assisted bicycle. Heart rate sensor is being used in conjunction with an accelerometer and gyroscope to control a hub motor. Objective of the project is to enable more people to choose an alternative commuting method and integrate exercise into their daily routine. Managing the safety system which consists of voltage regulation, Darlington switching, lighting and an audible alert.

2-Axis Gimbal*

Implementing a gimbal system in order to maintain a constant attitude of a camera, mounted on a moving platform, through the control of two servo motors. We are utilizing the accelerometer and gyroscope from the MPU-6050 and are writing the I2C driver and sensor fusion algorithm to control camera orientation using PID control and a digital filter.

3-D Tracking*

Utilizing two synchronized cameras to produce a stereoscopic image to be used for object tracking including object distance and velocity.

LED Stage Lighting

Designed and constructed cheap, cool, and efficient LED light bulbs to be used as stage lighting. Performed AC to DC conversion and voltage step-down.

**Spring 2014*

Education

In progress: BS Electrical Engineering • CSU, Sacramento • GPA 3.5 • December 2014

Related Courses:

Feedback Systems	Machine Vision*	Digital Control Systems*
Product Design I & II*	Probability and Random Signals	Modern Comm. Systems*
Microprocessors	C Code	Digital Signal Processing*
Electronics I & II	Network Analysis	

**Spring 2014*

Work Experience

Promoter/Performing Musician — ***Self Employed*** — *6/07- 9/12*

Organized music events by contacting venue owners and musicians through phone or email. Kept a tight schedule during the night of performance. Manned and organized the merchandise stand and sold products to customers.

Express Delivery Driver — ***PG&E*** — *6/09- 9/09*

Promptly supplied PG&E stations from Stockton to Madera with their daily packages.

Activities & Accomplishments

- Officer, Tau Beta Pi Engineering Honor Society – Industry liaison, Cataloger
- National Action Council for Minorities in Engineering Scholarship
- Engineering Futures Modules Certificate: Advanced people skills, analytical problem solving, effective presentation skills
- Dean's Honor List
- Institute of Electrical and Electronics Engineers Member
- Mathematics Engineering Science Achievement - Mentor
- Society of Hispanic Professional Engineers Star Member – Academics Committee
- Life Member, Boy Scouts of America

Michael Frith

Objective: Obtain an internship or start my career in hardware or product design as an Electrical Engineer.

Education

BS Electrical and Electronic Engineering • CSU Sacramento • CSUS GPA 3.76 • Fall 2014
CSUS Dean's Honor Roll: Spring 2012 through Fall 2013
AS Mathematics • Sierra College • 2011
AS Natural Science • Sierra College • 2011

Knowledge & Skills

Electronics: Circuit Analysis • Schematic Creation and Use • Circuit Design • Oscilloscope and Multimeter • Function Generator • Soldering • Logic Analyzer • Computer Hardware

Software: PSpice • Mentor Graphics • Matlab • Mathematica • Multisim • C Programming • Verilog (Quartus II) • MS Office • Python • LaTeX

Communications:

- Experience creating technical reports and documents.
- Able to create effective PowerPoint presentations.
- Experience speaking to large and small groups to explain complex concepts in simple language
- Extensive experience providing excellent customer service in high stress, high volume venues.

Work Experience

Student Intern — ***Miranda Technologies*** — 6/13 to 8/13
Managed replacement of an end of life product through the removal of $800 worth of components from an existing design. Saved an additional 30% in manufacturing costs while improving yield rates versus the end of life design. Assisted a computer engineer to ensure product's software compatibility with legacy systems in the field. Wrote specification on acceptance levels for Tri-Level sync for two products.

Math Tutor — ***Sierra College Math Center*** — 8/11 to 12/11
Individualized tutoring from basic to advanced mathematics and physics.

Waiter and Expediter — ***Hacienda Del Rio*** — 7/07 to Present
Provide excellent customer service while providing input to other servers and managers to continuously improve restaurant. Develop new product ideas and identify computer system improvements. As expediter- interface between wait staff, kitchen staff, and customers – facilitating continuously smooth food delivery. Identify and rectify any and all safety issues.

Personal Shopping Assistant — ***Best Buy*** — 3/04 to 6/07
Analyzed customer needs, and facilitated positive shopping experience. Led training of new and existing employees. Also responsible for customer support, effective display of product, and troubleshooting technical problems.

Project Experience

Smart Electrically-Assisted Bicycle (in progress) – Lead small multidisciplinary team to develop a smart electrically assisted bicycle. Bicycle utilizes terrain and biometric feedback to deterministically limit user exertion. Incorporates a safety system featuring lighting and sound.

Linear DC Power Supply - Designed PCB from schematic; built and assembled chassis from sheet aluminum.

*More projects available on LinkedIn

Relevant Courses

Product Design*	Introduction to Microprocessors	Introduction to Logic Design
Electronics I and II	Signals and Systems	Engineering Statics and Dynamics
Electronic Instrumentation	Applied Electromagnetics*	C Programming
Advanced Analog Integrated Circuits*	Network Analysis	Fabrication Techniques I
Digital Signal Processing*	Intro to Circuit Analysis	CIE Fundamentals-Mechatronics
Introduction to Feedback Systems	Electromechanical Conversion	Probability and Random Signals

***** *Spring 2014*

Professional Activities & Accomplishments

Member: Tau Beta Pi California Upsilon Chapter - Obtained by being in top 12.5% of Junior Class.
Engineering Futures Modules Completed: Advanced People Skills, Effective Presentation Skills, Analytical Problem Solving
Member: IEEE Sacramento Valley Chapter
Working 12 to 20 hours a week while carrying a full course load and maintaining a 3.76 CSUS and a 3.29 Overall GPA. US Citizen.

Devin Moore

EDUCATION

In progress: Bachelor of Science, Computer Engineering, CSU-Sacramento,
GPA: CSUS-3.88; Cumulative-3.22 (Expected Graduation: Fall 2014)

Transfer for B.S. Comp. Engineering, Santa Rosa Junior College

Related Courses

Data Structures and Algorithms(C++, Java)	Computer Interfacing
Comp Architecture(x86 assembly, HDL)	Physical Electronics
Circuit Analysis	Linear Algebra
Signals and Systems	Operating Systems
System Programming (Unix/C)	Embedded Systems
Advanced Logic Design	Differential Equations

KNOWLEDGE AND SKILLS

Programming Languages: C · C++ · x86 Assembly · Verilog · VHDL · HTML ·Java

Software: DevC++, Eclipse, Multisim, Matlab, Modelsim, Altera Quartus, Xilinx ISE, MS Office, OpenOffice

Equipment: Multiple brands of DMMs, oscilloscopes, function generators and power supplies. Experience with the Amani GTX and working with the Arduino Uno, Parallax Propeller, Digilent Analog Discovery, Microchip PICKit3, De0-Nano, Cyclone IV, Spartan 3E, and Raspberry Pi.

Communication and Organization:

- Great public speaking and presentation skills
- Exceptional problem solving and analytical skills
- Excellent with multitasking
- Thrives with heavy workload or stressful environments
- Strong verbal and written communication skills

PROJECTS

Inverted Pendulum Bot Using H-bridge with DC motors to balance a small robot using an accelerometer on the Cyclone IV FPGA on the De0-Nano development board.

Senior Project: SMART Electrically Assisted Bicycle Developed on an FPGA using calculated inclination and heart rate to provide unique assistance to rider. Year long project with team of four people. Multiple bus structures, wireless protocols, DSP, Verilog, C, Java.

Source: https://github.com/2014SeniorProject

WORK EXPERIENCE

Food Service **Sonoma Valley Bagel and Deli** 04/05 -05/08

Working with a small team to prepare and sell products. Eventually became assistant manager, with tasks including scheduling, opening/closing store, deposits, and delegating tasks.

Mechanic/Sales **Windsor Bicycle Center** 05/08 -11/09

Working alone or with one or two others to build, repair, tune, and sell bicycles. Repairing and building helped exercise problem solving and analytical skills.

Barista **Starbucks** 08/09 -Present

Working closely with a team of diverse people in a very fast paced environment. The constant expectation of efficiency and teamwork has been very rewarding. Having the opportunity to learn to work well with so many different types of people has been invaluable.

Achievements: working 25-30 hrs/week, Dean's Honor Role Fall 2012-Present, Tau Beta Pi Member