Battle of the Bikes - an Interactive Cardio Game

Adam Brower, Allison Kovarik, Nicolas Leocadio, Bo Williams

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Originally conceived as a concept to encourage young adults to frequent the gym more often, Battle of the Bikes is a project which aims to pit two users against each other in an isolated, self-powered video game that is driven by real world components. This project generates its own electricity from two stationary bicycles and stores it in order to run a competitive racing simulation, where the two users' bicycles communicate wirelessly through our self-made transceivers.**

***Index Terms* — Power Generation, Wireless Communication, App Development, SPI, Agile Software Development**

I. INTRODUCTION

Bicycle generators are becoming increasingly well-known in today's world as we move toward a society that's ever increasingly reliant on renewable energy sources, and as we find simpler and cheaper means to generate electricity in remote and isolated areas of the world. For our senior design project, we wanted to use the concept of cheap renewable energy in order to power a video game that could be entirely isolated from any grid in order to allow two users to compete against each other while still being able to perform some exercise, and to minimize any running costs of our system.

One of the inspirations behind creating this project was that we wanted to find a way to encourage competitive people of all ages to frequent the gym more often. Our project serves as a way of both improving cardiovascular health, and also hopefully encourages our users to stay in the gym afterwards, with our game being the first of many exercises that our users would perform.

A. System Setup

The overall setup of our system is simple: it consists of two stationary bicycles placed next to each other, 4 feet apart. Each bicycle then houses several identical systems, including one for automatic tension changing, one for sending generated power from each bicycle to the battery, and a mobile application that enables the users to see their current progress, their statistics, and allows for interaction with the other user's bicycle. The two bicycles are connected via a pair of wireless transceivers, and they also share a connection to send and receive their generated power from a single battery.

II. SYSTEM DESIGN

The most practical way to describe our project is to first explain each system, why we need it, and how it works. The main systems involved in this project are as follows: a power generation and battery charging system to create and store all of our generated electricity, a bicycle resistance altering system to allow for the users to interact with each other throughout the game, a voltage regulation system to deliver the correct voltages to all of our components, a wireless communication system which allows our two bikes to communicate with each other, and a mobile application which ties together all of our systems so that the users can experience a competitive and compelling interactive experience.

A. Power Generation

The most important part of this project is the power generation system - without it, none of our other components could work as they would not have any power. The following figure displays how our motor is currently attached to the stationary bicycle in order to generate power.

Fig. 1. Setup of motor generator

This is a simple system, and functions similarly to most other bicycle generators that we have seen: we have a motor on each bicycle attached to the spinning wheel with a high-friction belt. This belt could be attached to either the smaller inner-wheel of the stationary bicycle, where the larger wheel is driven from, or to the actual larger wheel. Below

is a picture of the actual system. Using the larger wheel would produce the most power, since we would get more revolutions on our motor from the larger wheel ratio. However, this would also produce about 100 watts of power per bicycle - much more than we are able to dump into our system to store or use. Therefore, we are utilizing a wheel attached to our motor that is about the same size as the inner wheel of the stationary bike. During testing, we were able to produce about 60 watts of power from each 12V generator motor, with our peak power coming from 6 volts at 10 amps - more than enough to power our entire system and charge the battery simultaneously.

B. Battery Charging

In order to charge our 12V, 6-cell battery efficiently and in a manner in which it would not be damaged under any circumstances, we purchased a DC-DC battery charger. Finding a DC-DC battery charger was a challenge in and of itself, but we were happy with our final choice as it gave us many different modes to test how our battery would be charged, and it also allowed us to hack into it to enable the charging process to start via a microcontroller pin. This was done by opening the charger up and connecting wires to the start button of the charge controller - the way it worked was simply by shorting two terminals inside of the board. In order to simulate this action, we attached these two terminals to a relay that will then be triggered by our microcontroller at the beginning of the race. We then send a signal for 10 seconds to tell the charger to begin charging the battery once the charger is actually powered on from our users' pedaling. The downside of this battery charge controller is that it requires an input voltage of 10 volts, which our generator setup does not presently reach. Our solution to this problem was to use some boosting voltage regulators in order to reach this desired 10 volts. These make it so that we have 10 volts going into our battery charger, and also limit the current that is allowed to be drawn by both the charger and the battery. With our current setup, we have can charge the battery at about 1.6 amps at peak performance of the users, which is more power than our system needs at any point.

C. Resistance System

Our project also employs a mechanical system that changes the tension applied to each bike. These consist of a set of 3D printed pieces and a servo which are all epoxied onto the bicycles. The image below in figure 1 shows the pieces that we printed attached to the servo motor.

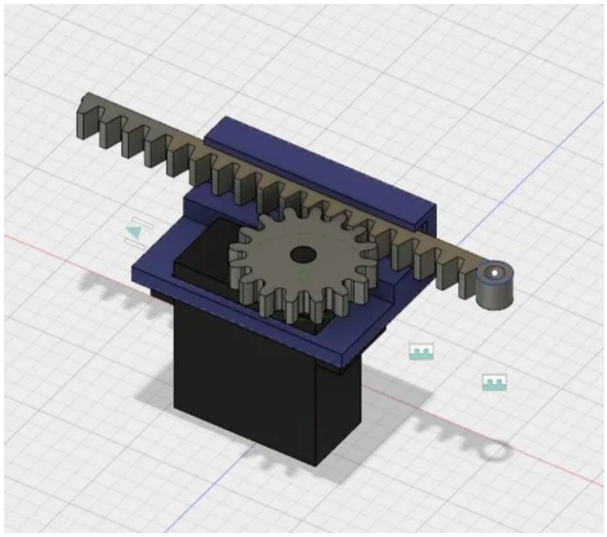


Fig. 2. 3D printed system for servo motor control of tension on the stationary bicycle.

In the picture, it is clear to see the goal of the printed pieces - we converted the rotary motion of the servo into linear motion. We then used this linear motion to move the rod that controls the amount of resistance on the bicycle up and down. The following figure shows this actual setup attached to the bicycle.

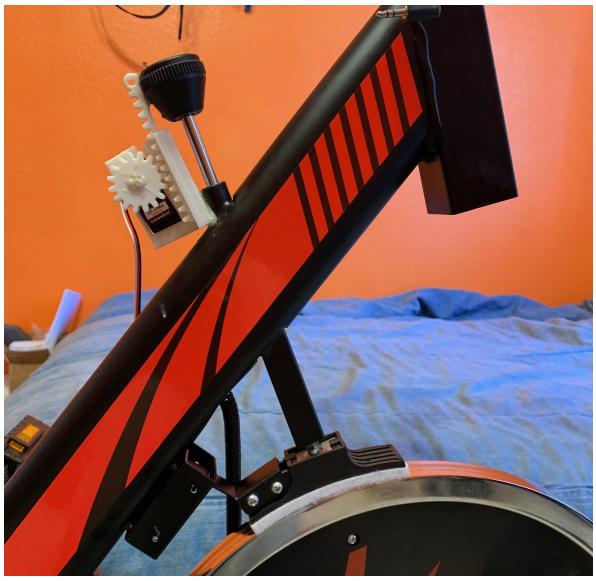


Fig 3. Bicycle resistance system attached to stationary bicycle.

This linear motion was achieved by attaching the toothed rod from the figure onto the knob of the bicycle's rod, and then attaching the casing to a the actual body of the bicycle, which allowed us to apply a large amount of force to the rod from the servo without worry. During our initial testing,

we were having some issues with the servo slipping. However, we later found out that our issue was that we were simply not allowing enough current to go into the servo as we were powering it from our development Arduino board as tbe board could only supply a peak current of about 130mA. When the servo was connected directly to our power supply, it was able to draw up to 900mA and move the rod within our desired ranges to make this project functional.

The reason for this system is that it is our way of adding competition to our game. Each user's resistance is directly controlled by the opposing user. After a user has gone a certain distance, they will earn a power-up in the game which will allow them to make pedaling harder for the other user. These power-ups may be used immediately or strategically in order to best create an advantage for each user.

D. Speed Sensing

In order to measure how fast each user is going and how far they have gone, we also employed a speed-sensing system on each bicycle. These are fairly straight-forward - they use a hall-effect sensor and a magnet placed on the wheel of each bicycle. As the wheel of the bicycles complete each revolution, the hall-effect sensor picks up the magnetic field of the magnets and increases a counter. We then determine each users' speed and distance traveled based on how frequently this counter is updated. This is also the only data that is sent over our transceivers along with when the power up button is pressed, since with it, we can do all of our other calculations to find the current metrics of the bicycle.

E. Voltage Regulators

There are two voltage regulators used in the design in order to provide the proper voltages required for each component of all systems to operate. The first regulator is connected directly to the battery charging system (BCS). The BCS provides a constant 12V from the battery to the first regulator. The regulator selected for this is a Texas Instruments LT8608 Adjustable Synchronous Step-Down Regulator. The purpose of this chip is to reduce the 12V input voltage into a 5V 1.5A signal. This signal is necessary to provide the V_{cc} voltage for several of the systems components; this includes the Atmega328p microcontroller, Atmega16U2 flash chip, servo motors, and multiple components in the transceiver system. The design for the 5V synchronous buck regulator system is shown in Fig. 4.

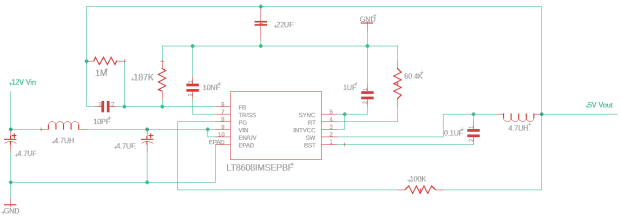


Fig. 4. LT8608 5V regulator.

The reasoning for choosing a switching regulator as opposed to a linear type is that of the relatively large step-down from 12V to 5V. A switching regulator performs this task while producing significantly less heat than a linear regulator at an equivalent voltage change.

The second voltage regulator takes the 5V signal produced by the LT8608 circuit and steps the voltage down again; this time to a 3.3V signal. The 3.3V signal is required to powered additional components of the transceiver system as well as the Bluetooth module. Because the step-down here is relatively small a linear regulator is used; specifically, the Texas Instruments LM3940. Unlike the LT8608 the LM3940 is a fixed-output regulator and thus the circuit only requires filtering capacitors to be placed at both the input and output signals of the chip. This diagram is as follows.

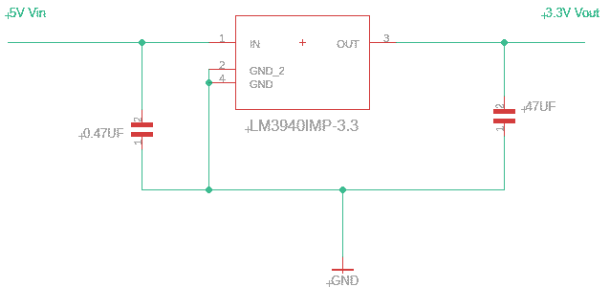


Fig. 5. LM3940 3.3V regulator.

F. Wireless Communications

For our project, we needed two different kinds of wireless communication. The first was Bluetooth to communicate between the stationary bicycles and the mobile application. For our system, we wanted to use as little power as possible to not draw an excessive amount of power because we want our system to be self-reliant. So, for that reason we chose Bluetooth Low Energy. Bluetooth Low Energy could be more than 50% more power efficient than normal Bluetooth and can still deliver the results that we need from it. Bluetooth Low Energy can send around 100 kbps which is far more than we need to transmit for our system. The Bluetooth module will be used to send strings

of data to be parsed by either side of the system, the mobile application or the microcontroller, and will act as a bridge between the user and the system.

The second wireless communication we will use in our system is a transceiver. The transceiver will be used to communicate between each bike and will pass the number of rotations of the user's bike and whether there is a power up signal to change the tension on the bike to the other user's microcontroller. We will go into more detail on how the transceiver works later in section III part B of the paper.

G. Electronics Housing

The electronics housing's sole purpose is to keep the electronics safe from being tampered with or harmed in any way. After debating over the many options that we could have chosen for material, we decided to choose wood. Wood was the favorable option for a few reasons: durability, ease of manipulation, and electronic signal interference.

Wood was an easy choice because we needed something that was easy to work with and would not interfere with our electronic signals while the system is in use, while still being able to provide a stable environment to house all our hardware. With our need for hall effect sensors to read the number of wheel rotations, we had to find a place to mount them where they could pick up the magnet on the outside of the bicycle wheel. So we mounted the sensor on the inside of the box right next to where the wheel will be. We then had to cut out a space from the wood to allow the wheel to fit far enough inside the box to allow the magnet to pass over the sensor to count the number of wheel rotations. We also needed to cleverly mount the DC motors to the bottom of the box to create enough tensions between the motor and the rubber belt that is attached to the wheel of the stationary bicycle. To enclose the box we decided to allow the top of the enclosure to be opened when needed to for setup and/or repairs to the system, so we put hinges on the top panel for easy access to the electronics while still providing safety for the hardware.

H. Calorie Calculations

When people workout, they generally like to know their caloric burn for the workout. So in our design we wanted to give the users that information at the end of the ride. Calories burned in a workout are determined off a number of things which can be represented in (1).

$$Caloric\ Burn = 0.0175 * MET * Weight * Minutes \quad (1)$$

The user's weight is needed in kilograms, the minutes refers to how long the user was riding for and the user's MET value is described in the following table. [1]

TABLE I
MET Values and their Corresponding Average Bike Speeds

MET	Description
4	Less than 10 mph on average
6	10 - 11.9 mph on average
8	12 - 13.9 mph on average
10	14 - 15.9 mph on average
12	16 - 19.9 mph on average
16	Greater than 20 mph on average

III. BOARD DESIGN

In this project there are four total individual circuit boards; two for each bicycle. Each bicycle has both a motherboard and a transceiver that our group designed. The two systems are identical to one another so there are two specific printed circuit board designs that are implemented.

A. Motherboard

The primary function of the motherboard for this project is to house the microcontroller and supply the appropriate connections between said microcontroller and all of the other systems. Our group needed to make sure that included on this board is the ability to reprogram the microcontroller. For this reason a 32KB flash chip was added along with a USB-A interface in which we can transfer updated programming to the microcontroller via a common USB flash drive or directly from a computer. The flash chip used is the Atmega16U2, chosen for its compatibility with the ATmega328p microcontroller. Also on this PCB are the previously mentioned voltage regulators LT8608 and LM3940. The last component housed on the motherboard is the HM-10 Bluetooth module. The reasoning for choosing to place both the HM-10 and the regulators on the motherboard is to distance the sensitive transceiver board from as much signal noise emanating from the wireless channel and the embedded signal lines as possible.

B. QAM Transceiver Board

The data is sent from bike to bike by means of transceivers. The board contains four main modules as well as supporting circuitry and noise rejection elements. The PCB is separated into transmit and receive sides with an RF switch dictating which mode the transceiver is in. Data is

transferred from the motherboard through the serial peripheral interface to an intermediary ATmega328p. Once the intermediary, or slave, ATmega receives the data it then converts it into a parallel format to be transferred to the transceiver board. The digital-to-analog converter (DAC) interfaces directly with the slave Atmega328p. The DAC selected was the AD9714. The DAC is a low power option with 8-bit resolution. This resolution was selected based on the relatively small amount of data being sent per transmission. The AD9714 contains an on-chip 1.8V LDO to drive the digital core logic. 3.3V is supplied by the motherboard to drive the analog and digital inputs and outputs. The AD9714 offers a dynamic current output for analog logic, allowing the user to fine tune the output DC bias with the aid of a voltage divider circuit. The DC bias was set to 500 mV with a 1Vp-p differential swing based on the input parameters of the next module downstream, the modulator. The DACs main function is to take the parallel inputs and split the input into its respective In-phase (I) and Quadrature (Q) data. The lower bits are selected for the I data and Q data consists of the upper bits. The data sent from the DAC has a frequency set to 1MHz. It is called the baseband signal and it is still in phase until it reaches the modulator.

The purpose of the modulator is to take the I/Q data from the DAC and offset the Quadrature data by 90°. It then up-converts the baseband signal to the desired frequency. The up-conversion is accomplished using a Local Oscillator (LO). The ADRF6703 was selected to perform these functions. It is powered by a 5V voltage supply and a 3.3V digital logic pin. The ADRF6703 contains an internal LO which can be programed to be an output as well. The LO generated by the modulator will also be used by the demodulator on the transmit path. The LO is set to 2.49 GHz. The benefit of using the I/Q data and QAM modulation is general is the same amount of data is sent over a much smaller bandwidth. Once the ADRF mixes up the signal it is output in a signal RF path. The signal travels down a coplanar waveguide with a width of 44 mils and spacing to ground of 9 mils. The RF signal goes through a bandpass filter centered around the up-converted frequency and through an amplifier which adds 15.3dBm of gain. Once filtered and amplified it passes through an RF switch. The RF switch is controlled by firmware and switches between transmit and receive with the rest of the PCB. The signal then travels to a ¼ wavelength antenna through an SMA connector.

The RF signal is then received by the other bike. The signal passes through the switch which is currently in receive mode and travels through a bandpass filter before reaching a balun. The balun is used to convert the 50Ω transmission line into a 50Ω differential pair. This differential pair is then input into the third main module on the board, the demodulator. The demodulator essentially has the opposite functionality of the modulator. The demodulator selected was the ADL5382, which runs off a 5V power supply. There is no on-board LO generator due to the fact both sent and received LO should come from the same source to reduce error. The demodulator uses the LO sent over differential lines from the modulator to down-convert the RF signal back to baseband levels (1 MHz) with a magnitude of 2.4V and split the data into the respective differential quadrature data.

The baseband signal then travels to an analog-to-digital converter (ADC) which processes the data and converts it to digital to be sent to the rest of the system. The ADC selected is the AD7729 which operates using a 13MHz clock and 3.3V supply. The ADC has a 15-bit resolution and was chosen due to the available extra quadrature preprocessing it does. The data is output using a serial interface back to the slave ATmega. The I and Q data are sent separately. The I-data is sent over two bytes followed by the Q-data to then be combined and post processed in the slave ATmega before it is sent back to the mot

IV. SOFTWARE

The software portion of this project is a critical piece that binds each subsystem of the entire project together. The software allows the system to provide information to be presented to the users and also allows the users to send signals to the hardware.

A. Embedded Software

The embedded software was written in the Arduino IDE since the microcontroller being used for this project was the ATmega328-p, which is one of the microcontrollers used for Arduino. The embedded software is in charge of maintaining control, and keeping statistics, of the stationary bicycle it is connected to. Connected to the microcontroller will be a hall effect sensor, servo motor, a self-made transceiver, and a Bluetooth Low Energy module. The software deals with each piece of hardware in different ways.

For the hall effect sensor, the microcontroller keeps track of the number of times a magnet passes over the sensor by using an interrupt service. The sensor will send a signal to the microcontroller any time a magnet passes over it and the signal goes from low to high. By using a few variables, the microcontroller is able to keep track of the total number of wheel rotations and also the number of wheel rotations that has occurred in the last second. We need to keep track of both total rotations and rotations within the last second for a few reasons. The first reason is to not run into an overflow problem with the transceiver, which can only send and receive 8 bits of data. With that limited amount of bits, the largest total distance we could keep track of would be less than a mile, which would not be useful for the project.

So, to solve that issue, we programmed the microcontroller to keep track of rotations per second which would never be more than 2 to 3 rotations. Also, by sending number of rotations per second to the mobile application, the user will have a more accurate value of current speed displayed to them.

For the servo motor, the microcontroller keeps track of the current position of the motor. The Arduino platform has a servo library which allowed us to implement the servo motor with the microcontroller by defining a pin to control with a single pin. To change the position of the motor, we utilize pulse width modulation on the microcontroller which is needed to change the position of the servo motor. Pulse width modulation is a means to decrease the average power delivered by an electrical signal by breaking the entire signal into many smaller discrete signals. To implement pulse width modulation, we need to change the position on the motor one degree at a time. To achieve this, a for loop is used and is run from the degree the motor is currently at and changes up, or down, to the desired degree by one degree per iteration. There will be predetermined distances that, when reached, will automatically adjust the tension to give the feel of riding on different terrain. The tension could also be adjusted if the other user sends a power up signal to change the tension for a short amount of time. The tension will automatically change back after the power up signal has faded.

The microcontroller will use the Bluetooth Low Energy module as a means to interact with the users on the mobile application. The Arduino platform has a software serial library that allows us to define a set of pins on the microcontroller as receive and transmit pins to send, and receive, information to, and from, the app. The Bluetooth Low Energy module sends and receives information as strings to be interpreted for use. To begin the game, the users will send their information through the Bluetooth to be captured by the microcontroller to set up, and begin, the game. After the game begins, the values of how many rotations each user has will be sent every one second to constantly update the mobile application.

For the transceiver, the microcontroller will utilize the Serial Peripheral Interface (SPI) to transmit data. The microcontroller will utilized SPI so that it can send information back-and-forth between the two bikes synchronously. The data that will be sent across the transceivers will be the number of rotations and whether the other user used a power up to change the tension on the other bike. This is needed so that each microcontroller will have both bikes information to be sent to the user's mobile application and so that each user can have control over the other user's tension if a power up is available.

B. Mobile Application

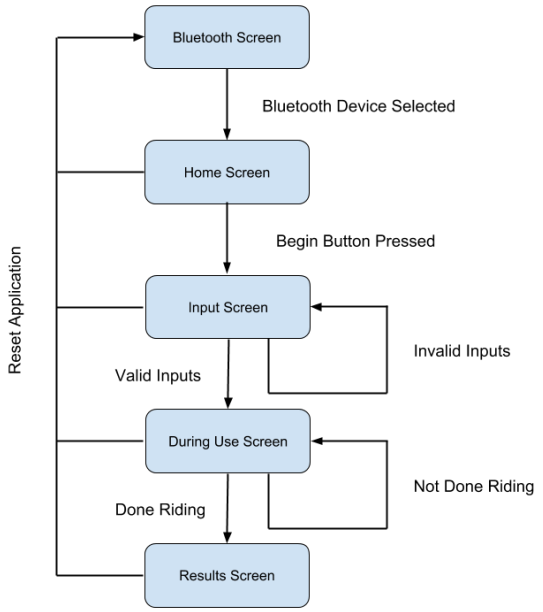


Fig. 6. Mobile application flowchart.

The mobile application for this project was needed for creating a game that was able to be modified to fit the desires for the current users. This application was developed for Android mobile devices in Android Studio using Java as the programming language.

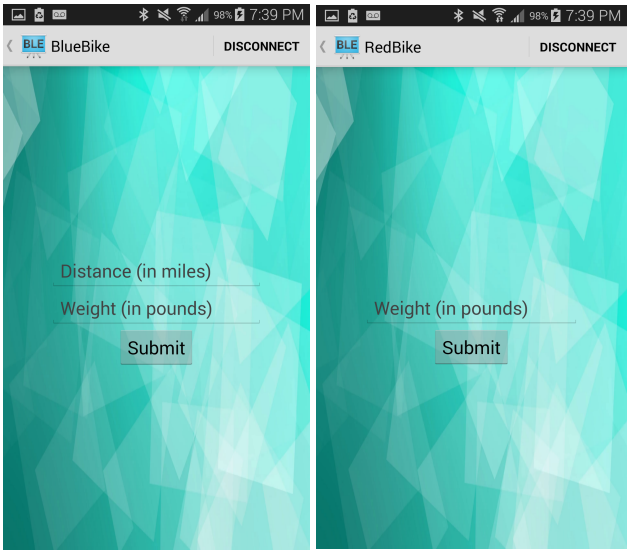


Fig. 7. Mobile application input screens for "master bike" (left) and "slave bike" (right).

Before the application can begin, the user must give the application permission to access location and Bluetooth so

that it can show the available Bluetooth Low Energy devices. After the user selects a device to connect to, the application will establish a connection between the user's phone and the stationary bike that they are riding. Once a connection is established, the user will be taken to their corresponding input page. There are two separate input pages based off of which bike the user is on. The "master bike" will have total control of the distance to ride for the game and will also ask the user for their weight for later calculations. The "slave bike" will only have to input their weight for later calculations. Each of these inputs will have bounds on accepted values. If the user enters an invalid input, the application will automatically catch the exception and will display text informing the user that the input was invalid and that they must send in a value that fits the necessary parameters. Once the user enters valid inputs, and clicks the submit button, the application will send the string over the air to the Bluetooth module it is connected to.

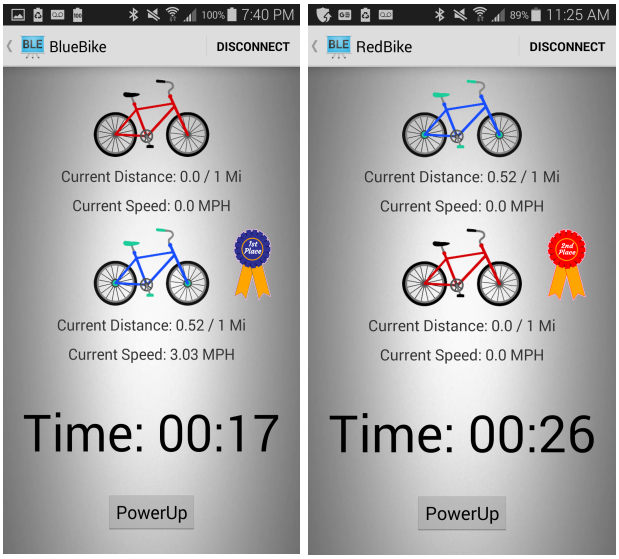


Fig. 8. Mobile application screens while the game is currently active.

After valid inputs are sent in, the screen will be updated to how the game looks during use. Depending on which bike the user is riding, it will display that bike at the center of the screen with the other bike at the top of the screen. Accompanying the two bikes will be a timer and a button to send a power up signal. If both users are not ready, it will show a static screen with both nothing updating on the screen. Once both players are on the same screen, the game will begin. Each users current distance ridden and current speed will be displayed under the appropriate bike to show how each user is currently doing. There will also be a ribbon that pops up next to the user's bike that shows whether they are currently in, a blue ribbon indicates first place while the red ribbon indicates second place. Each ribbon will also have its corresponding place value shown in the middle of it. If the user currently has a power up and presses the power up button, a signal will be sent to the microcontroller to change the other bike's tension for a short amount of time.

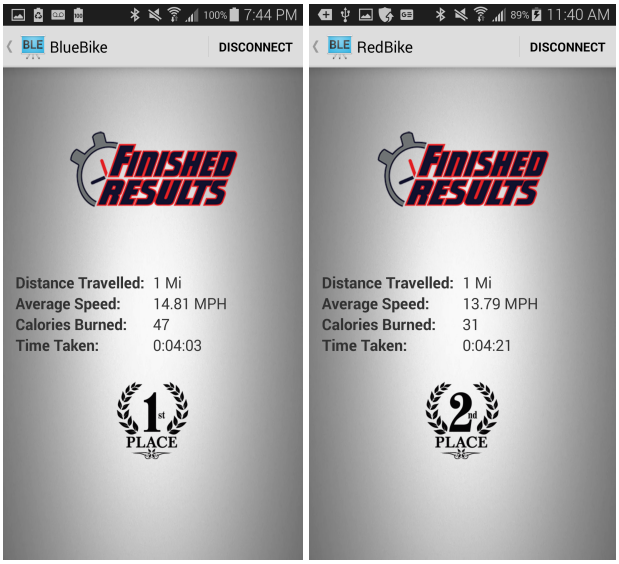


Fig. 9. Mobile application results screens for the two users.

Once the set distance has been reached by the user, another screen will be displayed to show the user their statistics for the game. These statistics include if they were in first or second, the distance they rode, the time it took them to ride that distance, the average speed at which they rode, and the calories they burned during the ride. This information is displayed in a two column fashion in order to make reading the values easier for the user.

C. Game Design

For creating the game, we wanted to keep the users engaged in such a way that they remained interested during use. In order to do so, we have the system update itself into a couple of different states. The current tension of the bike is what defines the different states. We created the game to cycle through three tension states to give the users the feeling of changing the incline on which they are riding. The three states are flat, uphill, and downhill. There is also a fourth state that can be reached in the case that one player is in the "Uphill" state and the other user activates a power-up that increases the tension even more so to send that user to the "Steep Uphill" state.

There are multiple ways to change states in the game. The first way for the tension state to change is by reaching the next checkpoint in the game. The total distance the users will be riding will be split into 5 equidistant stages.

Throughout the ride, it will cycle through the three states – flat, uphill, and downhill. The five stages will end up being, from start to finish – flat, uphill, flat, downhill, flat. As the user reaches the next distance checkpoint, the tension will automatically change from the state it is currently in, to the next state in the game progression. The other way that the tension state can change is if the other user uses a power-up. When this happens, the tension state will increase for a short amount of time and will automatically change back when the power-up time is reached. A finite state machine of the different tensions is shown in Fig. 10.

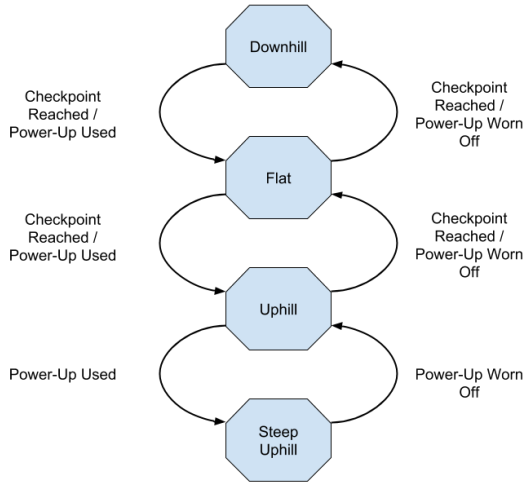


Fig. 10. Finite state transition diagram for the tension during use.

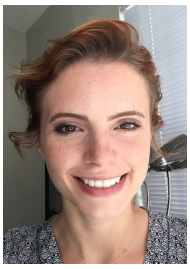
V. CONCLUSION

To conclude, we are proud and excited with the way that our project has turned out. When embarking on this two semester journey we were originally anxious about the magnitude and scope of this project. However, after having completed it we are glad to have had this learning experience.

First and foremost, one of the biggest takeaways from this entire experience is the group aspect of this project - we all came in as equals looking for a solution to a problem and, as the project progressed, we were able to establish our own roles within the team to maximize our productivity. Next, we were also able to experience our first real project designing experience, from the conception and design phase all the way through the building and finalization stage. This is invaluable experience that will allow us to move forward in our careers as we now have the knowledge required to more efficiently and effectively finish a project of this magnitude in the future.

THE ENGINEERS

Adam Brower is an Electrical Engineering student graduating from the University of Central Florida with his BS. He is currently pursuing his career options in the field and considering applying for graduate school for his MSEE.



Allison Kovarik is a 27-year-old pursuing an undergraduate degree in Electrical Engineering. Allison has accepted a job with Lockheed Martin MFC in Orlando, FL as a RF Engineer.

Nicolas Leocadio is a 22-year old Electrical Engineering student graduating from the University of Central Florida. Nick is currently seeking positions in the field of robotics and plans on attending graduate school after having some experience working.

Bo Williams is a 22-year old Computer Engineering student graduating from the University of Central Florida with his BS. Bo is hoping to start his career working at Amazon as a software development engineer located in Seattle, WA.

ACKNOWLEDGEMENT

This project could not have been completed without the invaluable help and expertise of our professors at the University of Central Florida and mentors, all of whom guided us in ways that allowed for this project to be finished in a timely and concise manner. We would especially like to give a warm thank you to Dr. Samuel Richie and Dr. Lei Wei. We would also like to thank our committee members who took the time out of their days to listen to, and judge us on, our project.

REFERENCES

[1] Average MET Values for Activities with Fitness Components. community.plu.edu/~chasega/met.html#1.