MarioKart Integrated Exercise Bike

Reed Hester - Chase Griffin - Leah Faulkner

Tennessee Technological University

Conception and Design

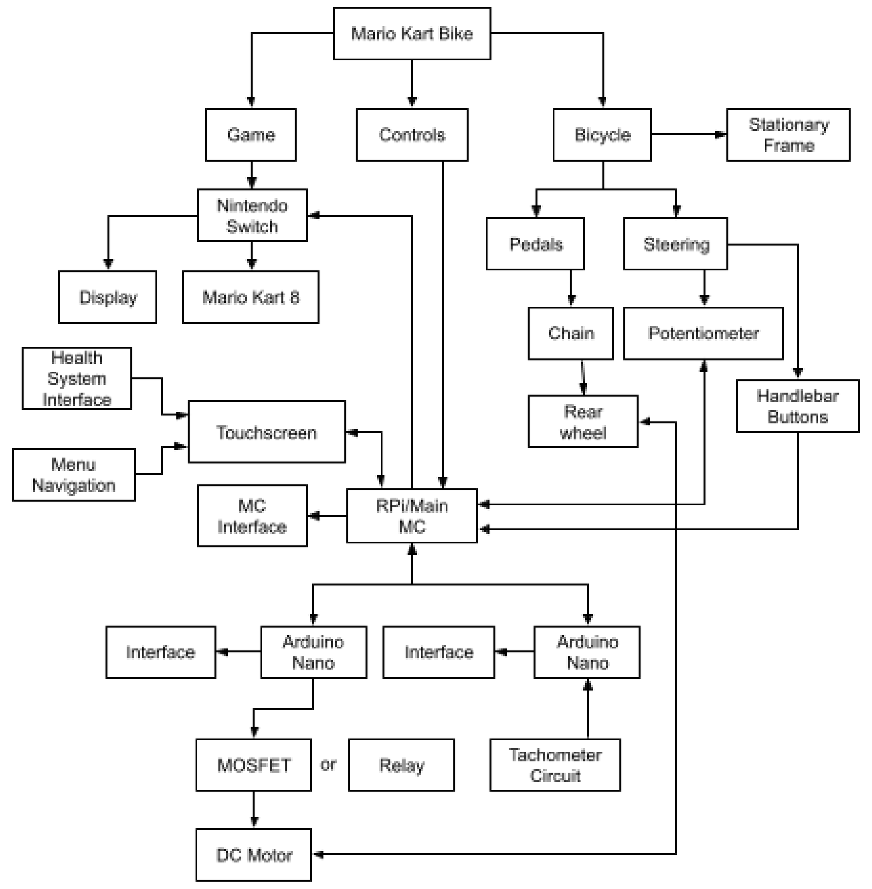

Figure 1. System Block Diagram

This project is a proposed solution to a problem existing across most video games: a lack of exercise. The goal was to produce a system that combines the competitive nature of gaming with the benefits of physical exercise. Consequently, this group has found that exercise is easier and even fun when the distraction of a competitive game takes over.

The approach the group used is described in the block diagram to the left. It has been split into three subsystems: the game, the controls, and the physical bike. The game subsystem is intended to run the MarioKart game and display it to the user. The controls subsystem is where the translation from sensor data to game data takes place, as well as the control of the dynamic resistance. The bike subsystem houses the sensors, the stationary bike itself, and the drives that influence the dynamic resistance system.

Figure 2. CAD Model

Pictured to the left is a CAD model of the final design. The intention is that any standard bicycle can be attached to the system. The bike is suspended by a steel pipe frame to allow for steering and for added stability. The rear wheel is suspended by a trainer stand, where it presses against a roller and translates the motion to the resistance system. The computing components are housed within a box at the front of the stand, along with the monitor that displays the game.

Controlling the Game

The control system has been designed and constructed in a practical and user-friendly way that integrates all of the standard racing controls from MarioKart onto the frame of the bike. The steering is operated by turning the handlebars which, through a system of gears, turns a potentiometer. Acceleration is done by pedaling normally, which causes the back tire with several marked sections to rotate. An infrared object sensor counts the sections as they pass and a calculation is performed to determine RPM and thus cause the in-game vehicle to accelerate correspondingly. Brake-lever style buttons on the handlebars control the item-throwing and drifting functions directly. The control signals are collected, translated, and sent to the Nintendo Switch console as controller data via a Raspberry Pi.

Dynamic Resistance

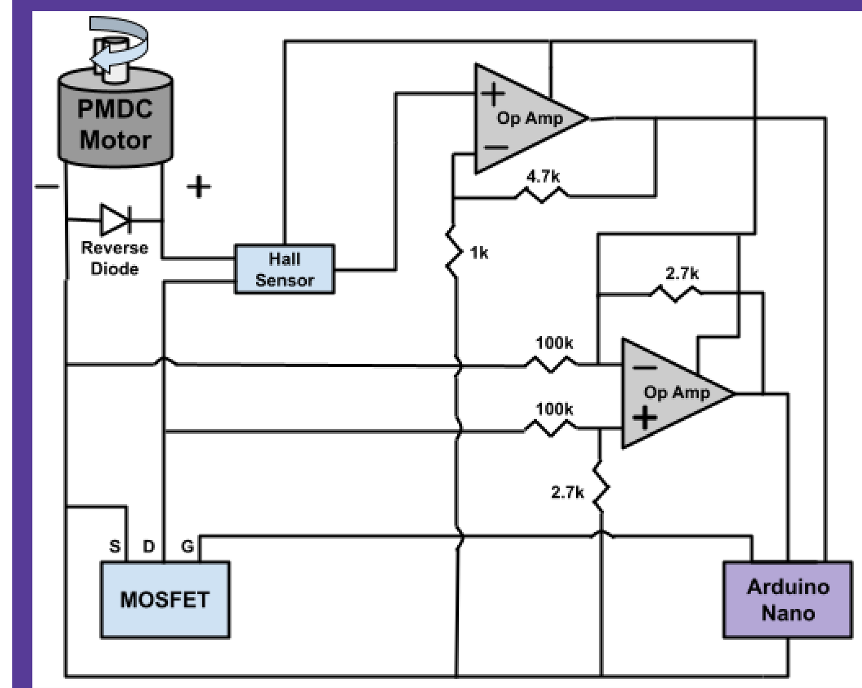

Figure 3. MOSFET Control

This concept began with the idea of using a DC motor as a brake. When the positive and negative leads are shorted, the motor experiences full back electromotive force (EMF) and resists motion. To control when this would occur, a **MOSFET** would be used. The **MOSFET** would act as a variable resistor by tuning the gate voltage so that it operates in the linear or Ohmic region. Additionally, the voltage and current generated by the motor would be measured at all times to ensure that the **MOSFET** is behaving correctly. However, the group faced many difficulties with the device. In the linear region, the **MOSFET** produces heat due to lower power dissipation compared to the fully **ON** state. This led to the physical failure of many **MOSFETs**.

Instead, the group has chosen to replace the **MOSFET** with a high-power relay. Like a transistor, a relay has both an **OFF** and an **ON** state. However, there is no way to operate between those states. That means that the rider can only ever experience zero resistance, or the full resistance. To keep the element of exercise alive, the group has programmed the relay to switch **ON** for a brief moment at random intervals of about 15-30 seconds throughout the race. It was planned that the resistance system would respond to in-game events such as turtle shells and bananas hitting the rider by using controller vibration signals. However, this could not be achieved because the Nintendo Switch has a highly advanced rumble system with hundreds of vibration states to decipher.

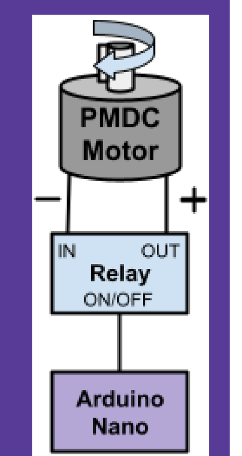

Figure 4. Relay Control

MET Health System

The the health system approximates the number of calories burned during a game. The equation that the calculation was based off of was:

calories=(duration*60*MET*3.5*weight)/200

The health system graphical user interface (GUI) is built off of the tkinter library, which is a Python wrapper for the Tk GUI toolkit. It implements menu navigation controls and a number pad to enter user weight for the MET calculation.

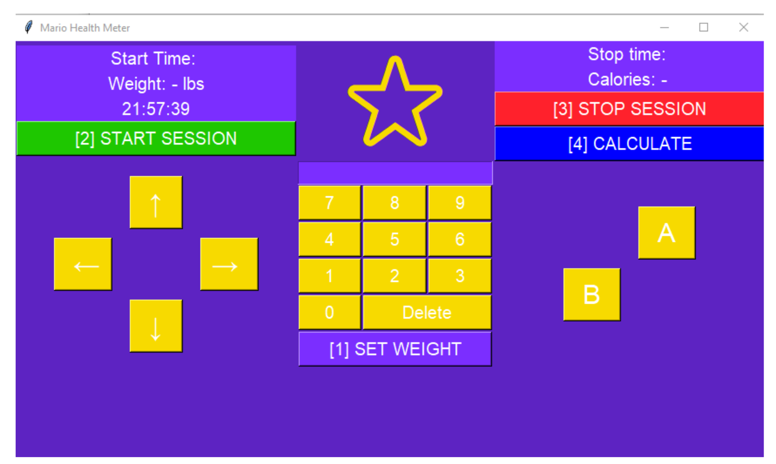

Figure 5. Python program for GUI

Experimentation and Future Work

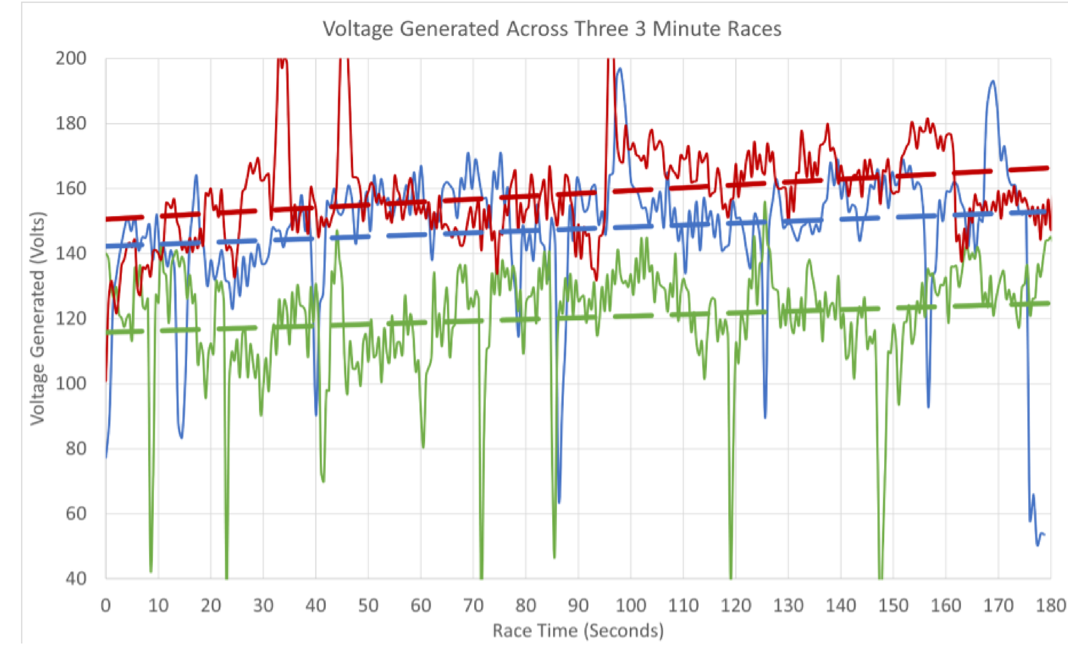

Figure 6. The dips show when the relay was **ON**, causing the rider to slow down. Dashed lines show the average.

As shown above, the DC motor generated over 100 Volts on average throughout the course of the races. Because a relay was used, the voltage generated is the open-circuit voltage. That means there is virtually zero current generated. However, there is still large potential for energy collection using this system.

For the control systems, steering was tuned via a logarithmic mapping onto in-game steering data which allowed the user to make sharper turns with only modest movement of the handlebars, and acceleration was gauged on the consistency of the RPM data (and therefore the consistency of the in-game vehicle's motion). Lastly, the other external control systems were tested for accuracy upon press and ease of operation during a race. That said, control experimentation was mostly subjective (i.e. whether or not the game was realistically playable given the fidelity and practicality of the control systems). To measure rider satisfaction, a survey was prepared that asks a series of questions about each component of the bike, as well as their overall gaming experience. It has been found that the majority of riders are satisfied with the equipment and had an enjoyable experience.

In the future, some improvements could be implemented such as: heat management, improved dynamic resistance system, more control fidelity, greater control immersion, and a broader consideration of standards in the event this were to become a marketable product.

About the Group and Acknowledgement

- Reed Hester: Electrical Engineering - Lead electronics/structural designer
- Leah Faulkner: Computer Engineering - Designer of MET health system and surveyor
- Chase Griffin: Computer Engineering - Lead programmer and team leader

The group would like to thank Mr. Jesse Roberts for his immense support and for the class structure he provided. They would also like to thank Dr. Van Neste for his help in the design of the dynamic resistance system.