Self Balancing Remote Control Toy Bike

A Technical Report submitted to the Department of Electrical Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering T R N P

Josiah Suwargo

Spring, 2024

Technical Project Team Members

Nguyen Ky

Pierre-Louis Dylan

Ramkumar Pranav

Tecson Lance

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Adam Barnes, Department of Electrical Engineering

Statement of Work

Ky Nguyen:

Ky was responsible for developing a mobile controls application, embedded software, and chassis design for a bike control system. Ky developed an iOS application using Swift, designed to enable user interaction with the bike's control system via Bluetooth, allowing functions such as turning and acceleration. The application featured a custom-designed UI/UX and was responsible for sending and receiving Bluetooth signals without delay. In terms of the embedded software, Ky created control logic to interpret Bluetooth commands for motor control, ensuring the bike responded accurately to turns and acceleration. Additionally, Ky focused on optimizing the chassis design for balanced weight distribution, enhancing the bike's stability and maneuverability.

Pranav Ramkumar:

Pranav was responsible for the physics analysis, electrical design, embedded software, chassis design, and CAD design. He gauged the feasibility of the reaction wheel bike concept, visualizing relationships between the bike and wheel through moments of inertia equations. In regards to the electrical design, Pranav researched the required electrical components as well as electrical specifications and parameters to set out a guideline for the bike circuitry. His contribution to the embedded software revolved around setting the abstract layout for the program, refactoring the code, and programming for the balancing control system. Within the balancing control system, he particularly focused on the conversion of IMU data into usable information, and PID tuning for stable balancing. Finally, he impacted the physical design through design decisions on the structure and material of the chassis, part placement for weight distribution, and the creation of the steering column CAD designs.

Dylan Pierre Louis:

Dylan was responsible for the design of the electrical system for the self-balancing bike, developing a complex system created to interconnect all components via a printed circuit board, leading to efficient integration and overall hardware functionality. After thorough research, the system was engineered with specific parts such as resistors, regulators, and converters which control electrical parameters, including power consumption, dissipation, thus providing correct voltages and signals essential for system operation. Once the circuitry was functional, the next step was to fabricate the printed circuit board with the internally implemented circuitry, soldering all necessary parts for continued operation. As the project progresses through the semester, the plan is to address any technical deficiencies brought to our attention, troubleshooting and resolving them as needed.

Josiah Suwargo:

Josiah was responsible for the mechanical aspect of the project, this includes developing a proper chassis, creating CAD files for components that are used in the bike, 3D printing the bike and its main parts, and assembling the bike as a whole. After doing the research necessary, Josiah used various resources in order to create the build for the bike, this includes utilizing the Ultimaker Cura software in order to use the 3D printers from both the RMC 3D printing studio and the MOL makers lab, the 3D prints that were made were used in most of the bike’s mechanical components, with features such as the wheel, the holders, the battery carriage, servo holder, and many more that was either created individually or done in collaboration with the other members in order to meet the requirements of what is needed for the bike to succeed. The assembly process was also done in congruent to the 3D printing process as most of the components were realized during creating and assembling the 3D printed components.

Lance Tecson:

Lance was responsible for the embedded portion of the project. This included research: on the techniques used to create the algorithm which dictated the behavior of the reaction wheel and the communication protocols between the components. Additionally, Lance made the necessary modifications to the base chassis which included measuring and drilling holes for screws, 3D printed components in order to attach the parts, and placement of the motors such that bands were taught and the motor was secure. Planning for the amount of pins used (for the knowledge of the PCB maker, Dylan) was done by Lance and was dictated by the communication protocols decided by Lance given the constraints. During optimization of the balancing algorithm, Lance worked with Ky and Pranav to fine tune it such that the algorithm was better at balancing the bike, this included rigorous testing of PID values and reducing noise of the input. Upon experimentally finding the PID values, Lance worked with Ky and Pranav to enable simultaneous turning and balancing.

Table of Contents

Table of Figures

Abstract

The Self-Balancing Remote Control Bike is a toy designed to introduce children to the principles of physics through interactive play. This RC-controlled bike incorporates a reaction wheel, mounted to the lower front of the bike. It actively adjusts the bike’s balance using Newton's Third Law to avoid the bike tipping over. The reaction wheel is engineered to balance any force exerted through the bike, ensuring stability and continuous upright posture. The bike’s primary goal is to provide a hands-on learning experience to middle and high school children through demonstrating fundamental physics concepts such as angular momentum, dynamics, and equilibrium. Within the toy kit, the bike comes with modular ramps that children can configure to create custom courses, turning their home into a fun and interactive physics experiment. At the heart of this bike are two dedicated motors: one powers a reaction wheel that dynamically adjusts to maintain the bike's balance, while the other propels the rear wheel for movement. In addition to the motors, there is a servo in the front of the bike that is tuned for accurate turning controls. Complementing this setup is an onboard accelerometer that continuously monitors the bike's tilt angle, providing real-time data to the control system to ensure it remains upright. All of these components are connected through a STM32 microcontroller. The bike comes with an intuitive software interface, with custom designed UI/UX, phone application that allows children to easily control the bike. This project aims to stimulate curiosity, enhance learning, and encourage exploration in an interactive manner for middle to high school STEM enthusiasts.

Background

Reaction wheel: Flywheel designed to apply opposing force on the attached body through Newton's Third Law. Newton's Third Law states "the actions of two bodies upon each other are always equal and always opposite in direction". Thus, when a reaction wheel is forced to rotate in one direction, an equal opposing force rotates the body in the opposite direction. NASA's Spinoff program [1] defines this in the context of spacecraft, suggesting that "by adding or removing energy from the flywheel, torque is applied to a single axis of the spacecraft, causing it to react by rotating". While this technology can be broadly applied to any physical body, there is particular focus placed on space applications, including satellites, where orientation is of major emphasis. In the Earth's exosphere, where satellites reside, no external bodies exist to enable efficient force generation in the rotational plane. Given the energy efficiency and lack of required outside assistance, reaction wheels are an effective form of orientation manipulation in satellites. This application falls off in relevance when we reach the Earth's surface, as there exist more energy efficient methods of changing orientation that take advantage of gravitational and normal forces. However, with the advent of new fields and increased autonomy of technologies, there will be a demand increase for autonomous control features, like orientation control. The following table showcases projects reliant on reaction wheels to control orientation.

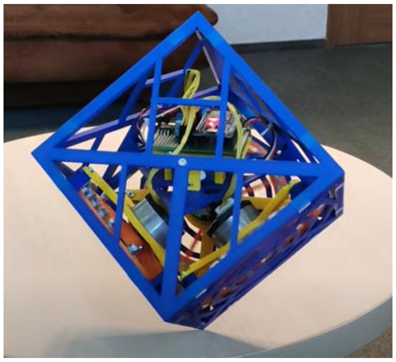
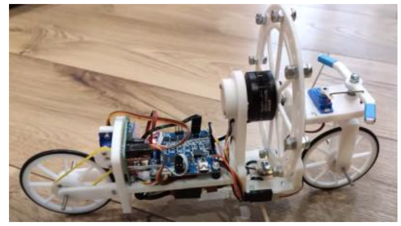
Self-Balancing Cube	Reaction Wheel Unicycle [2]	Reaction Wheel Bike [3]
		
• Uses reaction wheels to balance on a single vertex of the cube • Controls orientation on 3 axes	• Uses reaction wheel to balance on single driving wheel • Controls orientation on the pitch axis	• Uses reaction wheel to balance the bike • Controls orientation on the pitch axis

Figure 1. Comparison between past projects

Similar to these projects, we intend to utilize the reaction wheel in an effort to balance the bike. However, most of these projects do not consider the practical application of reaction wheels in translational motion. Specifically, the cube is not capable of moving at all and the unicycle [2] moves forward solely to maintain roll axis equilibrium. Even the bike by [3], a device capable of precise forward and backward motion, has a semi-unstable configuration given the high placement of the motor and reaction wheel, making long-term balancing less feasible. Our addition to the bike implementation is the inclusion of a dynamic pitch equilibrium as well as the thoughtful placement of the components such that the heaviest parts are on the bottom. Additionally, our placement of the reaction wheel behind the steering column is unique given other designs have the steering column form right angles which is less structurally secure than our 45 degree design.

Given this extension and the general nature of work we intend to complete, a strong background in the field of electronics is critical to our success. We will be taking the knowledge and coursework from many Computer and Electrical Engineering courses offered at the University of Virginia, including but not limited to the ECE Fundamentals Series, Introduction to Embedded Computing Systems, Electromagnetic Energy and Power Conversion, Software Development Essentials, and Physics 1. The core electrical engineering principles are useful to creating the PCB and ensuring electrically valid connections. On the programming front, embedded programming will assist in correct interfacing between the microcontroller and other components, while the skills in software development will ensure the controller application is user-friendly and up to standard. Finally, physics will be important to analytically proving the physical viability of our goals.

Social Impact Constraints

The bike has been designed with safety as a foremost consideration and presents no major safety concerns. While there is exposed wiring, it carries a low voltage of no more than 11V, posing minimal risk. Due to time constraints, we were unable to develop a fully enclosed chassis that would further enhance safety by covering all exposed wires and components. However, all materials used in the bike's construction are non-toxic, safe., sustainable, and recyclable

Additionally, the design carefully avoids the inclusion of any small, detachable parts that could be swallowed. Plans for a fully enclosed chassis were initially in place, which would have concealed all wiring and mitigated any risks associated with exposed, sharp output pins. This feature remains a consideration for future enhancements to improve safety further.

Environmental Impact

In addressing the environmental considerations of the bike, particular attention has been given to the selection and disposal of the Lithium-Ion battery power source. Aware of the environmental hazards posed by these Lithium Ion batteries, our project takes a responsible stance towards minimizing our ecological footprint. Lithium Ion batteries, while containing substances that can be harmful to the environment if discarded improperly, offer a significantly longer lifespan compared to traditional battery types. This extended usability reduces the frequency of battery replacements and, consequently, the volume of waste generated over the product's life cycle. Furthermore, the rechargeable nature of Lithium-Ion batteries aligns with our commitment to sustainability, as it decreases the reliance on single-use batteries, reducing necessary disposal. Battery disposal involves techniques including pyrometallurgy, responsible for breaking down used batteries into valuable nonferrous components, which “is a highly energy-demanding process, resulting in GHG emissions and the generation of toxic gasses or hazardous slag” outlined in [7]. Thus, reduced disposal leads to reduced pollution. However, recognizing the importance of responsible battery disposal, our project includes detailed guidance on recycling Lithium-Ion batteries in the user manual, encouraging users to dispose of batteries appropriately at designated recycling centers.

Sustainability and Scalability

The bike incorporates sustainability into both its design and functionality, with a strong focus on durability and environmental consciousness. It is constructed using robust 3D printing PLA plastic that is recyclable, along with exterior cover for extended durability; [8] outlines these traits for PLA. The bike is designed to be very battery efficient and could last for several hours on constant load. This minimizes the battery recharge cycles, which increases the battery lifespan.

Health and Safety

Toys hold unique and strict requirements regarding health and safety regulations. Considering the general lack of maturity held by the target end user of 11-15 year olds, tighter regulations must be followed to lessen injury probability from unexpected applications. In the case of our bike, many electrical and physical components can pose a risk to the health of children if not accounted for. This includes “sharp edges of toys or other electrical, mechanical, or flammable characteristics (that) may cause accidents” [9]. Consequently, we will address these concerns through calculated specifications in the design.

Firstly, to mitigate the risks posed by the bike's electrical components, the entire structure has been encased in durable, highly resistant plastic. This casing is designed to withstand moderate stress, collisions, and damage, safeguarding the internal components from exposure and the children from potential electrical hazards. Additionally, a majority of the external area of the bike is covered with strips of foam. This foam acts as a buffer, significantly reducing the force of impact in the event of a crash, thus minimizing the risk of injury to the child.

Another critical safety consideration is the bike's and reaction wheel's speed. To prevent the possibility of harm due to excessive acceleration, we have calculated and set the maximum force output of the motor in relation to the mass of the bike. This ensures that the bike operates within a safe velocity range, preventing it from reaching speeds that could pose a danger to children. The motor's power output is carefully maintained within a controlled environment, guaranteeing that the bike's speed remains within these safe limits. The reaction wheel is encased in soft material covers to ensure safety, allowing it to be safely touched even when spinning at maximum speed.

The use of a Lithium-Ion battery introduces concerns regarding durability and the potential risk of safety issues over long time use. Recognizing this, we have implemented measures to facilitate easy replacement and maintenance of the battery. Detailed instructions for battery care, replacement procedures, and safety precautions are provided in the user manual. This ensures that users are well informed about managing the battery's life cycle, significantly reducing the risk of safety incidents related to battery deterioration.

Ethical Considerations

In the development of the bike, ethical considerations have been integrated into every aspect to ensure both environmental friendliness and safety for children. The choice of recyclable 3D printing plastics and the inclusion of an automatic shutdown feature are measures to minimize the environmental footprint. In addition, the bike is designed so that no data collection is necessary. Users can fully engage with the bike's features, free from concerns regarding their personal privacy.

Physical Constraints

After accounting for parts that were ordered but not used in the final design or were repurchased due to electronic or physical failures, the bike is relatively cost-effective and highly reproducible on a large scale. From a software perspective, the STM32 G071RBT6 microcontroller proved to be robust, handling all necessary processing efficiently. It also provided ample RAM to manage the data inputs from the IMU.

Manufacturability

In terms of manufacturing, the primary component requiring professional production is the PCB, for which we had JLCPCB constructed the PCB board. The bulk of our chassis was custom made using the UVA Lacy hall materials and tools. Other components such as the steering columns, gears, battery carriages, reaction wheel, wheels, and chassis supports were produced using the 3D printing facilities at the UVA 3D printing lab. The mechanical aspects of the system are a combination of components sourced from our own homes and purchased from online vendors including Digikey, Amazon, and McMasterCarr.

Parts availability

The essential components required for the project include the STM32 microcontroller, printed circuit board (PCB), accelerometer, motor driver, motors, and a power source. These components are readily available, with the majority sourced from Digikey and Amazon. Comprehensive research was conducted on each part's online datasheet to verify compatibility with the system's other components. Additionally, the team has obtained numerous spare parts in their homes and engineering laboratories, serving as replacements due to many shipping delays. The component that requires the most constraints is the motor for the reaction wheel, which requires precise specifications to deliver the correct torque for maintaining the bike's balance. As a result, this component required careful consideration in design and integration choices to make sure the bike consistently stays balanced.

Economic and Cost Constraints

We were assigned a flexible budget ceiling of $500 by the university. The cost of individual electrical components proved to be relatively inexpensive, which meant financial constraints were minimal due to our strategic use of 3D printers and available spare materials. Throughout the project, a significant portion of our expenses stemmed from the need to order multiple replacements for components like motors, servos, and gears, which did not initially meet our system requirements. To note, initial price calculations did not account for the cost of constructing the frame, as we had free access to 3D printing facilities. However, for mass production, the expense of building the frame will need to be included. It is estimated that the frame, comprising two pieces of plastic held together by six plastic standoffs, would cost between $10 to $20. Consequently, the total cost of the project, including the frame, is expected to cost around $150.

Tools Utilized

Our team utilized a variety of software tools to develop the project, including STM32Cube Programmer, STM32Cube IDE, NI Multisim, NI Ultiboard, Arduino IDE, Ultimaker Cura, Fusion360, and Google Colab. The STM32Cube programs and Arduino IDE were essential for programming the STM32 microcontroller in the C programming language. Given that none of the team members had prior experience with the STM32, having started with the MSP432, we all learned the software from scratch. This involved study of datasheets given by STM and engagement with online tutorials to improve with the STM32Cube IDE. The

Arduino IDE was specifically used to program the Arduino-compatible pins on the STM32, which controlled several of our hardware components. Fortunately, team members Ky and Pranav brought familiarity with Arduino boards and coding to the team, making a smoother learning curve for the group. For the design and fabrication of the PCB, Dylan predominantly utilized NI Multisim and Ultiboard. His extensive knowledge with these tools allowed him to efficiently design a PCB that effectively managed the power and signal connections of all hardware components, ensuring compatibility and functionality within the bike's design.
The connections were routed in Ultiboard and the electrical designs were made in Multisim. In terms of 3D printing, Fusion360 was our go-to software for CAD modeling of all custom components, including steering columns, gears, battery carriages, the reaction wheel, wheels, chassis supports, and standoffs. After modeling, these designs were transferred to Ultimaker Cura to facilitate communication with the 3D printers to initiate the prints. The entire team possessed prior experience with 3D printing. Additional training was undertaken to utilize the University of Virginia's 3D printers effectively, as they were available to us at no cost. Lastly, Google Collab was used to develop an algorithm that picked the best components for the PCB.

External Standards

Standards

US law mandates that all toys adhere to comprehensive safety standards and testing guidelines as specified by ASTM F963. This regulation is designed to prevent injuries related to choking, sharp edges, and other potential hazards through five principal requirements:

1. Physical Properties: Toys must be constructed to minimize potential injuries. This includes the design factors such as eliminating sharp edges, using parts large enough to prevent choking, and avoiding elements that could cause entanglement. Furthermore, toys should be sufficiently durable to avoid breaking into dangerous fragments.

2. Flammability: Toys and their components should be manufactured to resist ignition and melting when exposed to heat or fire, ensuring they do not present a fire hazard.

3. Chemical Composition: Toys must be free from harmful chemicals and must not exceed limits set for substances like lead, phthalates, heavy metals, and other potentially hazardous materials, safeguarding children from chemical exposure.

4. Electrical and Thermal Properties: Toys that use electricity or generate heat must comply with standards designed to prevent injuries such as shocks or burns, ensuring they are safe for handling by children.

5. Labeling and Marking: Proper labeling on toys is crucial. They must clearly indicate the appropriate age group, highlight any potential hazards, and provide instructions for safe use and maintenance to guide consumers in ensuring a safe play environment.

In addition to ASTM F963, our project also adheres to ISO 8124, a standard that is commonly applied internationally. While ISO 8124 largely mirrors the safety guidelines and criteria of ASTM F963, it differs primarily in its documentation and labeling requirements, regional focus, testing procedures, and threshold levels. This dual compliance ensures that our toy meets safety standards both in the U.S. and internationally, accommodating a broader market and reinforcing our commitment to child safety.

Intellectual Property Issues

Overall, the holistic design of our bike is unpatentable. However, there are unique aspects of our bike. A unique aspect is the placement of the PCB, motors, servo, and reaction wheel; the placement was planned such that the heaviest components would be at the lowest possible placements. Additionally, all 3D printed parts were custom designed and therefore belong to us. Lastly regarding software, the algorithmic techniques used are conceptual and do not closely follow any algorithms patented; however, our choices of communication protocols (inspired by the physical limitations of the amount of pins on the STM) is possibly unique to us. The physical limitations of the STM lead us to incorporate SPI (accelerometer) and UART (Bluetooth). Our design overall does not encroach on the patents of similar products. Three similar patented products are: [4], [5], [6]. [4] describes a patent for a chassis of a RC bike where the battery is movable. Our battery and its holder are static and not movable. [5] describes a patent on a method of unique identification between a receiver and transmitter. The patented method is the communication of a GUID, globally unique identification, which once communicated between a receiver and transmitter, allows only data with a matching GUID to be received. Our communication between our transmitter and receiver is through Bluetooth and follows the standard communication through Bluetooth, not utilizing a GUID. [6] describes a patent on determining the angular velocity in regards to servo control. Our servo does not take in any control feedback and instead is solely controlled by PWM.

Project Description

Performance objectives and specifications

The self-balancing bike is designed to serve as an innovative toy which introduces children to fundamental physics concepts in an interactive and enjoyable way. Specifically designed for middle to highschoolers, this project will encourage hands-on experiments focusing on concepts like angular momentum, dynamics, and equilibrium which would help foster both curiosity and learning. Within the project, the complex system will be designed by three major parts: a efficient electrical system which interconnects all components through a printed circuit board (PCB), a STM-32 Microcontroller which will be the brains of the system, leading to efficient integrations of all hardware components and software functionality, as well as various other components contributing to the bike's overall design.

Physical Viability

The feasibility of the reaction wheel in regards to balancing was verified through rotational kinematics. The most important factors to the bike balancing are the moment of inertia of the bike, reaction wheel, and torque of the motor. The moment of inertia (MOI) of the bike and wheel are represented through the following equations.

Bike MOI: $I_b = m_b(w^2/12 + h^2/3)$

w=width of bike, h=height of bike, m_b=mass of bike

Wheel MOI: $\frac{1}{2} m_w r^2 < I_w < m_w r^2$

m_w =mass of wheel, r=radius of wheel

The bike MOI is directly correlated to the length, width, and mass of the bike. Thus, any increase or decrease in those properties will result in the respective change for the MOI. Similarly, the wheel MOI is directly correlated to the radius and mass of the wheel, so it will respectively change with these properties. An additional note to address for both MOI equations is the impact of the displacement of mass. Mass displaced closer to the pivot point of the object results in a lower MOI, so the mass was displaced lower on the bike to reduce the actual MOI compared to this expected value. Comparable changes were made to the wheel, by moving the mass displacement to the outside with screws, increasing the MOI without increasing the mass.

This information is important to address, as similar to mass in the linear plane of motion, MOI impacts the resistance to angular acceleration (AA). The following equation expresses this.

Wheel vs Bike MOI: TM=Ibb=Iww

TM=motor torque,b=bike AA, w=wheel AA

In the context of the bike, reducing the bike MOI directly impacts the ability of the motor to balance the bike. Specifically, a lower bike MOI suggests greater potential angular acceleration and more effective balancing. The MOI of the wheel is less consequential, although there is a minimum bound so as to avoid motor saturation. Saturation describes the motor reaching its maximum RPM, which must be avoided since this limits the motor's ability to accelerate. The motor torque must also be great enough to support the acceleration requirements for balancing.

All together, these equations and relationships were used to determine the bike dimensions and weight, wheel size and weight, and motor specifications. Additional optimizations were necessary given motors with more powerful specifications tend to be heavier, increasing the MOI of the bike, however this guideline provided the current setup for the bike. The required torque of the motor was determined to be 0.159Nm, while the maximum mass of the bike was determined to be 3kg with the height and width not to exceed 20 cm each. With these specifications, the bike was expected to balance +/- 10 degrees of upright. The current bike configuration holds a motor capable of 0.21Nm of torque, is around 2kg, and has a height and width around 15cm, so the bike is expected to balance as well if not better than the analytical predictions.

How it works

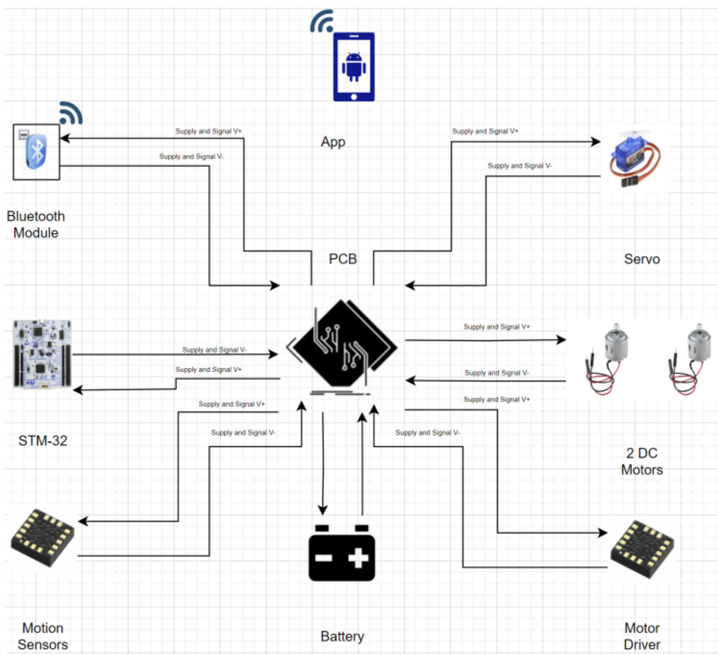


Figure 2. General System Overview

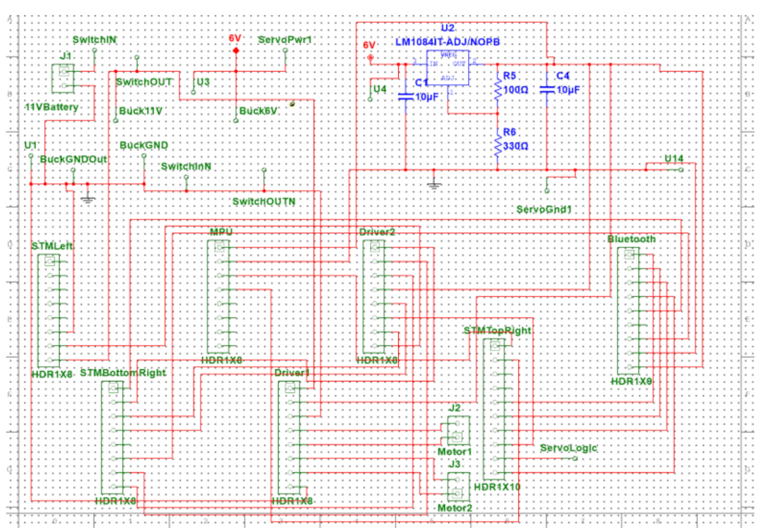


Figure 3. Final System Schematics

Hardware and components include three motors: a DC motor connected to the aft wheel serving as the propeller of the bike, another DC motor for the reaction wheel which maintains overall balance via Newton's Third Law, as well as a Servo motor connected to the forward wheel for accurate turning controls. The motor drivers will control the speed and positioning of the wheels, so this will help with the conservation of energy and enhance overall performance. An accelerometer will also be part of the system which will continuously monitor the bike's tilt angle, providing real-time data to the controller ensuring that the bike will stay upright, even while encountering external obstacles/disturbances. Since this will be a remote controlled system, the team has decided to also include a Bluetooth module which will enable remote communication between the user and the bike's system. Figure 2 shown above provides a general overview of the system, while Figure 3 delves into the detailed schematics.

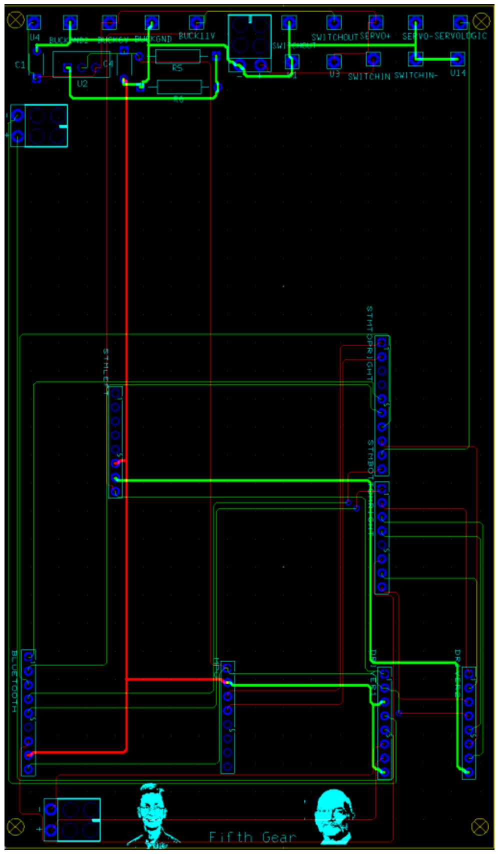


Figure 4. PCB Layout

The user will control the bike via an intuitive software interface phone application which will send commands to the STM-32. Powered by the PCB shown above in Figure 4, the STM-32 will interface with the entire system to execute user commands based on the real-time data being received from the motion sensors and the Bluetooth module. The motor drivers will control both the rotation of the reaction wheel, dynamically adjusting and maintaining the bike's balance as well as control the Servo motor, which will direct the bike in any direction in regards to the user's commands . The efficient coordination between all these components will ensure the bike's performance since the STM-32 will be managing the bike's operations effectively.

Technical Details of Design

Hardware

Figure 5. Dantona 11.1V Lithium-Ion Battery Pack

The complex system consists of components enabling the functionality of the balancing bike seen above in Figures 2, 3 and 4. While understanding the necessity for a stable power supply to accommodate sudden changes in tilt axis while users interact with the bike like propelling it forward, backward, and steering, we opted for a rechargeable Lithium-ion 11.1V 2600mAh battery pack as the power source which can be seen in Figure 5. This battery pack will be able to provide overall about 28.86 watts of power. With the determined power source, our focus shifted to finding components which would be suitable for use all while considering the limitations of the battery life.

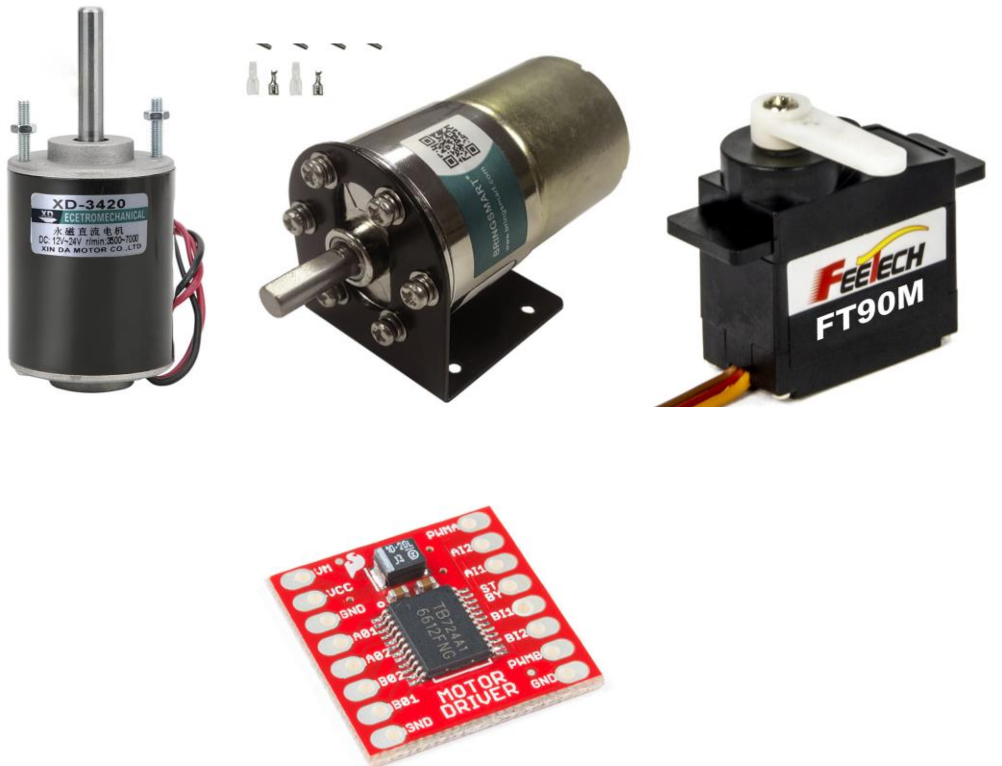


Figure 6. Reaction wheel motor (left), rear wheel motor (middle), FEETECH servo (right) and Motor Driver (bottom)

The technical design of the balancing bike had to incorporate 3 distinct motors and a motor driver. The forward motor was selected with the intention of providing a considerable amount of torque which would serve as the reaction wheel. It is a high-torque 12V DC permanent magnet DIY motor [12], rated at 3500 RPM with a max current rating of 400mA, consuming about 4.8 watts of power. The rear motor was selected with the intention to drive the back wheel in forward and reverse motion. It is a 12V DC Gearbox DIY motor, rated at 100 RPM with a max current rating of 600mA, utilizing about 7.2 watts of power. To efficiently control both motors, we integrated a SparkFun Dual TB6612FNG motor driver [11] into the design. This driver amplifies electrical signals to power and control the motors, offering precise speed control, over-current and over-temperature protections. Capable of providing up to 13.5V to both motors, the driver operates at 5.5V with a current rating of 0.4A, consuming about 2.2 watts of power. To control the direction of the forward wheel, a FEETEC positional servo motor [15] was implemented. Operating at 6V with a current of 130 mA, it utilizes about 0.78 watts of power.   In total, the motors and the driver account for 14.98 watts of power being utilized. All three motors and the motor driver are shown above in Figure 6.

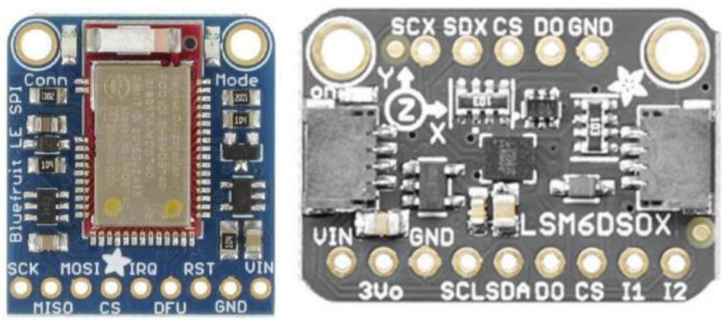


Figure 7. Adafruit Bluetooth Module (left) and Gyroscope Sensor (right)

In order to continuously calculate the bike's tilt axis for stability, we integrated the Adafruit Gyroscope sensor [13]. The sensor is meant to measure changes in the bike's speed, direction and angular motion. Operating at 5V DC with a miniscule current, it consumes roughly about 0.021 watts of power. For user interaction, we implemented the Bluefruit LE SPI Friend Bluetooth Transceiver [14] which allowed SPI communication between the microcontroller and the user application. Also operating at 5V DC with a miniscule current, it consumes roughly about 0.02196 watts of power. Both the bluetooth module and gyroscope sensor can be seen above in Figure 7.

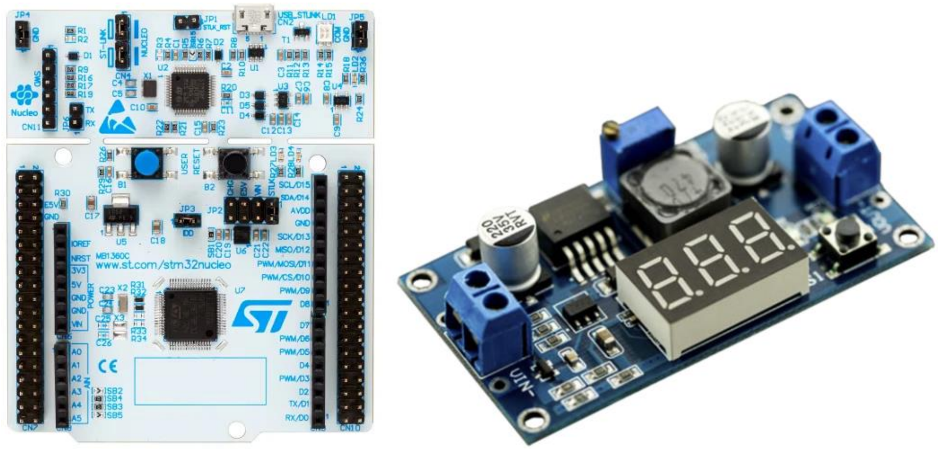


Figure 8. STM-32 Microcontroller (left) and Buck Converter (right)

The STM-32 microcontroller was selected as the central processing unit, which met all requirements necessary for proper functioning of the balancing bike. Powered by 5V DC with miniscule current, the microcontroller ensures smooth functionality. The STM-32 Microcontroller can be seen above in Figure 8. Overall, the system requires roughly about 15 watts of power to operate fully, providing a battery life of up to 1.93 hours

Shown earlier in Figure 3 is the electrical system printed circuit board, illustrating the circuitry designed for the bike’s operation. The circuit includes a buck converter, one voltage regulator, two resistors and two capacitors. While both the reaction wheel and the aft wheel motor require 11.1V, the servo requires 6V, and all other components require about 5V. To efficiently reduce the voltage, the buck converter [16] shown above in Figure 8 was implemented to drop the 11.V coming from the battery to 6V. Not only does the converter reduce the voltage, it simultaneously increases the current provided by the source. Lastly, the voltage regulator chosen converts the 6V DC to 5V DC, providing up to 5A of current to the components connected.

CAD Model Design:

In order to create components necessary to empower and achieve the goal of the bike, we had to design a model in a CAD software beforehand, we decided to use Autodesk's Fusion 360 as it not only stores created files in its cloud server, but also has smooth file conversion from f3d (Fusion's default design file) into stl (standardized 3D printing file type). Using Fusion 360, we created multiple CAD designs through our bike that underwent multiple iterations. Among these many designs, we wanted to highlight the components that managed to be a key part of our bike.

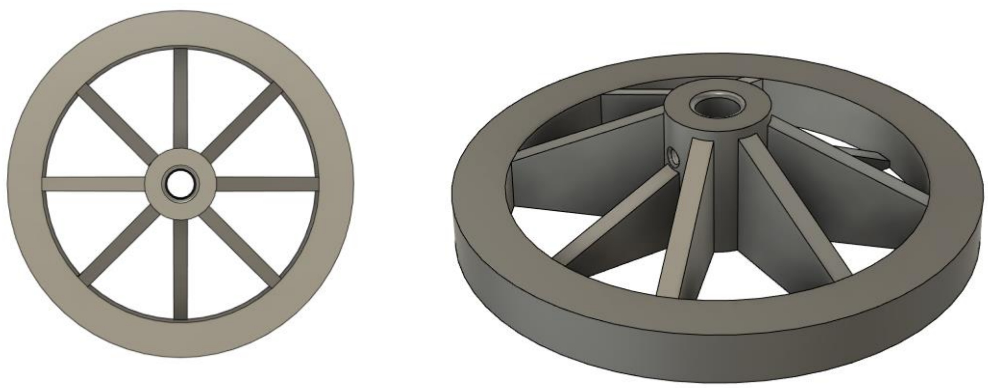


Figure 9. Reaction Wheel CAD Design

- Reaction Wheel: For the design of the reaction wheel, we wanted to ensure that it fulfills two requirements, be big enough to be able to generate the momentum it needs to balance the bike and have its size fit the space between the ground and the Servo Mount to avoid friction. With these specifications in mind, we made a reaction wheel with an outer diameter of 100 mm and an inner diameter of 80 mm, these specifications not only fit our requirements, but it also ensures that the thickness of the wheel will make it sturdy enough to withstand any unexpected occurrences that may happen to the bike. The arches around the inner circle are made to promote sturdiness of the reaction wheel when it is around the motor, the arches were also made with the thought of preventing the reaction wheel from hitting the chassis since we had little room to work with. The hole in the inner circle is made to screw in the reaction wheel, ensuring it doesn't slip or move when the motor is moving it. In order to compensate for the hole being that deep in the inner circle insertion, we had to adjust two of the arches as seen in Figure 9.

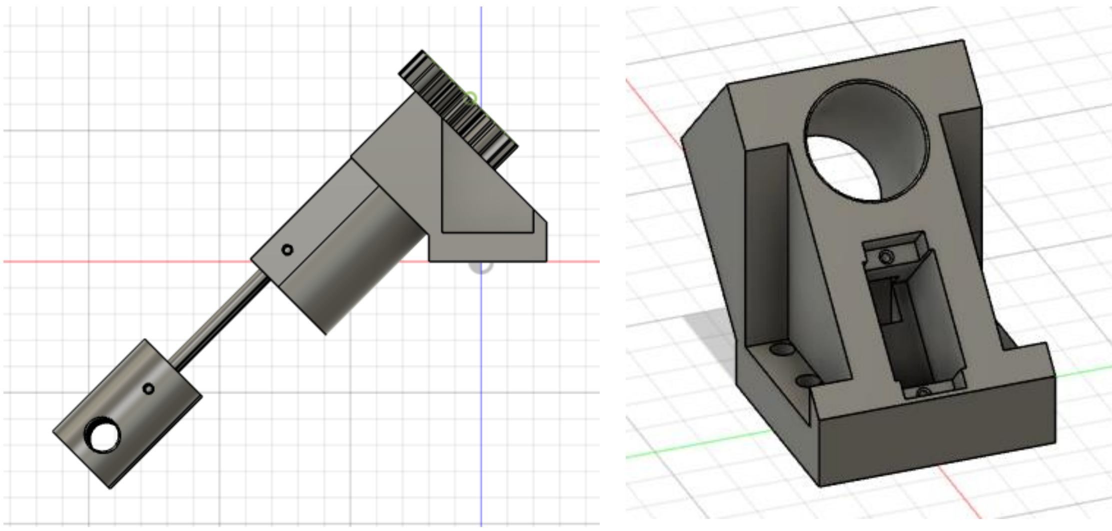


Figure 10: Steering Column and Servo Mount CAD Designs

- Servo Mount and Steering Column: The servo mount and steering column act as the structural components of the bike to assist in steering. The servo mount is critical, as it secures the servo to the chassis and has a fitting to connect the servo to the steering column. In addition, it orients the steering column at 45 degrees to provide ample space for the reaction wheel that fits underneath. The steering column itself is a fairly basic design, including a column that fits to the servo mount and legs to position the front wheel on the axle.
- Rear End Supports: Our bike needed a connector between the top part of our chassis, our bottom part of our chassis, and to our back wheel. In order to achieve this we created two connectors that not only act as a channel to the three components we want to connect, but also as a support to increase the sturdiness of the bike overall.

- Wheel and Tires: After discovering that the wheels we ordered were unfit to help balance our bike, we decided to 3D print our own wheels which would also come with tires, we fine-tuned both the wheels and tires to fit our desired conditions which include, being wide enough so that it gives the bike more room to balance but also not be too wide to the point where the bike balances itself with no help. In the design process of the tires, we wanted to allow our bike to be able to move past certain minor obstacles, in order to help with this, we designed 1 mm tall grooves on the tires which gives it more grip when it is trying to go over a specific obstacle.
- Battery Carriage: Upon initial testing, we realized that the battery kept falling and that something to hold it in place would be essential. We also noticed that there were some holes already drilled in the chassis that we developed, with this in mind, we designed the carriage to fit both the holes and the specifications of the battery itself.

3D Printing:

To realize our designs, we used UVA's resources that were made available to us, this includes the 3D printers from both the Maker Open Lab (MOL) located in the Chemistry Building and the RMC 3D Printing Studio located in Clemson Hall. We mainly used the Ultimaker S3 and S5 with their PLA materials as upon testing, we realized that these printers provided us with the best prints compared to the others, there was one instance however where we used the Lulzbot TAZ 6 to print the tires as it was the only printer that had the TPU filament available.

Software

Embedded software design:

The embedded software is split into two files: a main file and a header file. The header file contains all of the constant values that should not change during runtime for greater code readability and comprehension. Changing global variables were declared at the top of the main file to distinguish from the constant values. The main file contains two functions: a setup function and a while loop. The setup function initializes the bluetooth module, IMU, and pins. The while loop contains functions that are to be executed each run of the loop such as: retrieving data from the IMU and bluetooth module, processing the obtained data, and sending electrical signals to motors / the servo.

The STM retrieves data from the bluetooth module in the form of a 4 element char buffer. Index 0 and 1 are interpreted as a single input that dictates the turning of the servo. Index 2 dictates the forward or backward motion of the driving motor. Finally, index 3 toggles the reaction wheel on and off. Index 0 and 1, concatenated, holds a number 1-99; an instruction of 50 or 0 instructs to not rotate while 1-49 refers to steering left. Index 2 is a number 0, 1, or 2. {0: nothing, 1: forward, 2: backward}. Index 3 is either 0 or 1 {0: reaction wheel off, 1: reaction wheel on}. This data is actively received every iteration of the loop, as this was experimentally determined to create the quickest response times and SPI communication reduces the likelihood of lost data.

The data from the IMU is also retrieved every iteration of the loop. The pertinent data is the x, y, and z angular acceleration and velocity vectors, passed through a complementary filter to receive mostly noiseless tilt data. The complementary filter reduces noise by calculating a weighted sum of the current measured tilt and expected tilt based on the current angular velocity and angular displacement.

The reaction wheel algorithm, based around a PID (Proportional, Integral, and Derivative Gain) controller, integrates the tilt data to create a control feedback system to operate the reaction wheel. Functionally, “P” represents the current error or the current angle off of the target angle (where 0 refers to standing upright), “I” represents the accumulated error, and “D” represents the change in error from past measurements. These terms are weighted by scalar PID coefficients that impact the PID output and ultimately, the angular velocity of the reaction wheel. Through PID tuning, we experimentally found the ideal coefficients providing the most accurate angular error.

The velocity at which the reaction wheel rotates is then calculated by mapping the PID output to the motor PWM signal through a linear mapping function that accounts for the bounds of both variables. The reaction motor is then inputted a voltage according to the outputted PWM signal, controlling the angular velocity. The direction of this angular velocity is determined by the sign of the current tilt, executed by an if conditional capable of reversing the polarity of the motor power wires.

Controller IOS mobile application design:

The objective of this iOS application is to establish a successful connection between any iPhone and the bike, enabling users to control the bike wirelessly. We chose the Adafruit Bluefruit LE SPI Friend as our Bluetooth module, which communicates via SPI. Consequently, we ensured that the send and receive functions within the mobile application were compatible with this communication protocol. Below is a diagram illustrating the general structure of our software code:

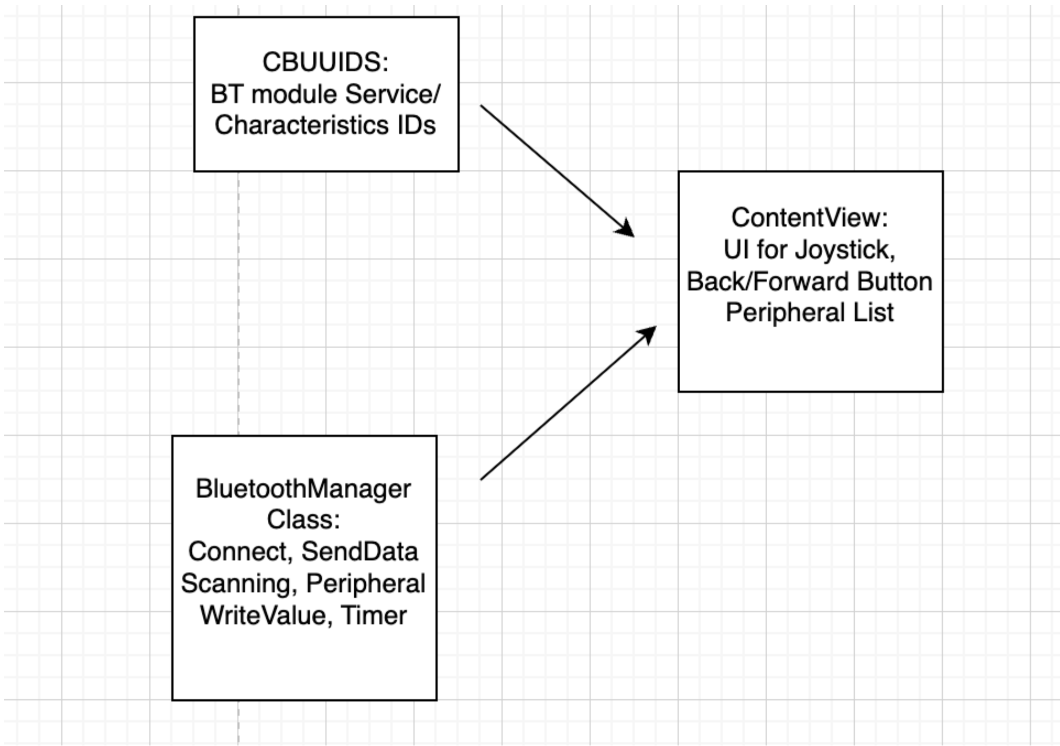


Figure 11. IOS Application Design Diagram

Our team opted to develop an iOS application rather than an Android version for several reasons. Firstly, all team members possess iPhones, simplifying our testing environment. Secondly, Ky's prior experience with Swift and iOS development significantly streamlined our progress.

To initiate the development process, Ky secured an iOS developer certificate. Developing for iOS proved to be more stringent than for Android, requiring us to complete numerous administrative steps before beginning actual development. The application is structured around three main components: ContentView, CBUUIDS, and BluetoothManager class.

- ContentView Class: This is the primary class responsible for the layout of all user interface elements, such as the joystick and buttons. It serves as the frontend of the application and interfaces with the backend, namely the BluetoothManager and CBUUIDS classes, ensuring seamless operation between the two.

- CBUUIDS Class: Essentially serving as a header file, this class contains critical information pertaining to the SPI Bluetooth identification number for the Adafruit Bluefruit Bluetooth module. This header is important for enabling the application to connect to the correct Bluetooth module through the SPI protocol. This class contains Bluetooth Service , Send , and Receive IDs.
- BluetoothManager Class: This class stores all essential Bluetooth functions needed for transmitting and receiving packets to and from the Bluetooth module. It handles the control information—such as commands for moving forward, backward, tilting, and maintaining connection. Messages sent to the Bluetooth module consist of a three-character message plus a terminating character, sent every few milliseconds. The first two characters represent the tilt angle, scaled from 0-180 down to 00-99 for data compactness. The third character indicates acceleration: '0' for neutral, '1' for forward, and '2' for backward. The class includes a built-in timer to regulate data transmission, preventing data overflow to the Bluetooth module. It also scans for and connects to the Adafruit module specifically, incorporating safeguards against multiple simultaneous connections to the same module.

Using these three classes, our iOS-controlled Bluetooth bike controller functioned flawlessly, with minimal to no delay in responsiveness. Pictures of the mobile application prototype are shown in the Final Results section.

Lessons and Experiments

Designed to teach middle and high school students fundamental physics concepts, the framework for bike experiments and teachings is based on the SOL, the Science Standards of Learning for Virginia public schools. The SOL serves as a guide for educators to meet science education goals and foster students' exploration of the natural world. Lessons and experiments suitable for both middle and high schoolers include:

1. Experimenting and documenting the relationship between the bike's center of mass and stability by placing it in different positions and/or inclined planes.
2. Setting up obstacle courses with various obstacles like cones to analyze the bike's response to different obstacles and terrain variations.
3. Exploring the bike's system by measuring kinetic and potential energy changes during balancing and movement.
4. Redesigning specific components to enhance balance, efficiency, or safety applying engineering principles.
5. Observing and creating mathematical models to simulate the bike's behavior under different conditions.
6. Adding weights as a payload on the bike to observe the relationship between load capacity and performance.
7. Adjusting parameters of the control system software to optimize balance and response characteristics

Problems Encountered

Hardware

- The instability of the IMU was an afterthought; it was installed into headers (which we soldered onto the PCB) rather than directly onto the PCB. This allowed it to move slightly between runs, offsetting what the bike considered "up right". By the time this was realized, the headers were already soldered and it was too difficult to remove them in order to solder the IMU directly onto the PCB.

Embedded Software

- More research should have been done since much time was spent on how to implement PID controllers for the reaction wheel; this includes implementing a complementary filter earlier in the process to reduce noise
- The amount of data sent in between the STM, IMU, and user led to some stalling initially which caused the bike to be unable to steer, drive, and balance simultaneously; this was resolved by changing the frequency of instructions the user could send to the controller. It was reduced enough to not overload the instruction queue of the STM while negligible lag was introduced between the instruction send and execution.

Mobile Application Software

- The development of the mobile application presented several challenges, with the most significant being the configuration of the backend software structure. It was crucial to align this structure meticulously with the Adafruit Bluefruit specifications to effectively utilize the Bluetooth API. This required experimenting with various software architectures until we identified the most suitable one.

- Another notable challenge was designing a user interface that was both simple and intuitive. Our goal was to ensure that each element within the app was user-friendly, concise, and reliable. Extensive time was invested in conceptualizing the UI, starting with sketches on scratch paper to solidify our ideas before beginning the coding process. This meticulous planning was essential to create an interface that met our standards of simplicity and functionality.

3D Printing

- The 3D printers are slow and somewhat unreliable in their availability, there were several times that software had to wait on new 3d printed parts. For example, the values for the PID controller were for adjusted for the new parts

Mechanical design

- The center of mass of the bike was not always middled leading to the bike having a tendency to fall to one side. This led to it being very stable when resisting external forces from one side but weak from the other.
- Imprecise measurements of screw holes led to multiple holes being drilled into the chassis that did not end up being used since they were slightly off the screw holes of the component we were trying to secure.
- For a majority of the project lifetime, we were using wheels which were far too slim to make the bike stand, even with the assistance of a reaction wheel. This led to much time tinkering with the algorithm to better stabilize the bike even though

it would've required extremely precise PID values to make the bike stand with the slim wheels.

Improvements if we could go back

If given the opportunity to revise our design choices, we would opt for a lighter brushless motor. Throughout the project, we recognized that maintaining the bike's stability at its central balance point was a significant challenge, exacerbated by the excessive weight of the bike. By choosing a motor that is approximately 90% lighter than our current model but with similar performance specifications, we could substantially enhance the bike's balance. Additionally, a lighter motor would allow us to reduce the overall size of the bike by about 50%. This reduction in size would not only improve the bike's stability but also decrease the workload on the reaction wheel, making it more efficient. Consequently, with less strain on the reaction wheel, the process of tuning the PID control algorithm would become more straightforward and effective, leading to better handling and stability of the bike.

Test Plan

Electrical System Test Plan

I. Connect the 11.1V battery pack to the printed circuit board, and verify that the system boots up, powering all components.

   A. If the system does not boot up, disconnect and recharge the battery pack using the manufacturer's smart charger.

   B. If the previous step does not fix the issue, acquire a multimeter and verify the required voltages at these pin points:

1. 10-11.1V at SWITCHIN and BUCK11V. If the voltage is not present, then troubleshoot ground connections.
2. 6V at U3 and U4. If voltage is not present, then troubleshoot the buck converter.
3. 5V at U2 pin 2. If voltage is not present, then troubleshoot the voltage regulator

II. Utilizing the interfacing app, ensure that components are performing actions desired by the user. Perform the following concurrently with verifying software/coding:

A. If the STM-32 does not respond, verify that it is receiving its required voltage (5V) at the 5V pin on the mid-left side of the board. If 5V is being received, ensure that jumper JP2 on the board is placed over E5V. If the issue still persists, repeat step 1B. Further issues may lead to the replacement of the STM-32.

B. If the servo does not respond, verify that it is receiving its required voltage (6V) at pin SERVO+. If 6V is being received, replace the Servo. If not, repeat step 1B.

C. If the motors (front and/or aft) do not respond, verify that it is receiving its required voltage (11V) at junctions J1 and J2. If 11V is being received, replace motors. If not, verify that 11V is being received by the motor driver at pin VM. If not, repeat step 1B. If 11V is being received, verify that 5V is being received by the motor driver at pin VCC. If 5V is received, replace the motor driver. If not, repeat step 1B.

D. If the bluetooth module does not respond, verify that it is receiving its required voltage (5V) at pin VIN. If 5V is being received, replace the module. If 5V is not being received, repeat step 1B.

E. If the MPU does not respond, verify that it is receiving its required voltage (5V) at pin VCC. If 5V is being received, replace the MPU. If not, repeat step 1B.

Software Test Plan

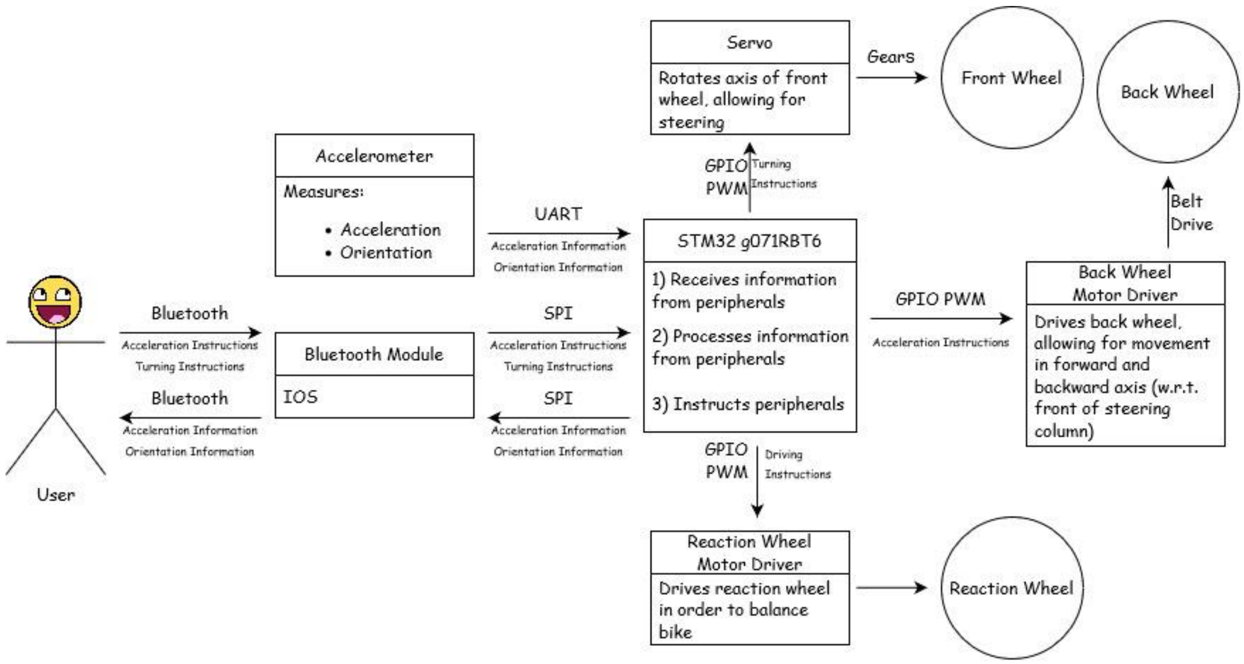


Figure 12. Communication Methods between Components

I. Ensure components are connected and software connections are stable.

A. Verify the STM32 is fully functional through print statements verifying the execution of the setup and loop functions.

B. Verify the bluetooth module is connected and ready to send data through an error message that displays when the connection fails to initialize.

C. Verify the IMU is connected and ready to relay tilt through an error message that displays when the connection fails to initialize.

D. Verify STM32 connection with the motor driver through manual writes of PWM signals to each motor/servo.

II. Ensure the balancing system is operational.

A. Verify the accelerometer and gyroscope data from the IMU.

1. Print the accelerometer and gyroscope data to ensure the STM32 is polling from the IMU.
2. Ensure the tilt matches the physical tilt of the bike by printing the tilt while standing the bike upright. If the tilt does not read 0 degrees, recalibrate the IMU or equilibrium variable.

B. Verify complementary filtering accurately represents the bike tilt while removing extraneous noise.

1. Overlay the raw tilt data and filtered data on a plot.
2. Ensure the filtered data does not drift excessively from the raw data.
3. Ensure considerable noise is removed in the filtered data (Noise is defined by inconsistent spikes in tilts that do not match the physical tilt changes in the bike)

C. Verify the accuracy of the target angle by plotting the target angle against the respective current tilt of the bike.

D. Verify the PID controller produces an output reflective of the error inputs via printing in many tilt cases.

1. Ensure the PID bounds match the intended max tilt bounds by printing the outputs at maximum tilt cases
2. Ensure the PID bounds match the intended delta error bounds by printing the PID outputs at maximum velocity cases

E. Verify the reaction wheel motor has the right speeds and direction written to it

1. Ensure the PID output maps to the motor PWM values by printing the PWM values for different tilt cases
2. Ensure the direction of the reaction wheel matches the sign of the current error by observing the angular velocity of the reaction wheel during a tilt sign change.

III. Ensure the Bluetooth controller/driving is operational.

A. Verify the Bluetooth module connection to the phone via the turning on of the blue LED on the Bluetooth module.

B. Verify the Bluetooth module is relaying data from the phone to the STM32 via a print statement within a condition that tests the availability of data.

C. Verify accurate commands are being received by the STM32 char buffer.

1. Ensure the character buffer can be written to by manually writing data into it and sending Bluetooth data to the buffer.
2. Parse the buffer into the individual instructions and use if conditionals and print statements to ensure the expected values are being sent (Reference the implementation for specific expectations)

D. Verify the impact of the commands on the motors and servo.

1. Ensure the steering commands map to the correct servo PWM values by matching the servo behavior to the steering input.
2. Ensure the drive motor direction matches the command sent through physical observation of the motor angular velocity.
3. Ensure the reaction wheel is toggled on and off based on physical observation of its behavior

IV. System

A. Using an iPhone, connect to the bluetooth module

1. Interacting with the interface, one’s inputs to accelerate or turn should reflect unto the RC bike consistently
2. Tilting the bike should spin the reaction wheel in the same direction as the tilt; a higher magnitude of tilt should lead to a higher speed of rotation
3. Rotating the servo should cause the balance point (the tilt at which the reaction wheel no longer spins) to change such that the bike leans but has no angular velocity (the bike is stable at a non zero tilt)

Timeline

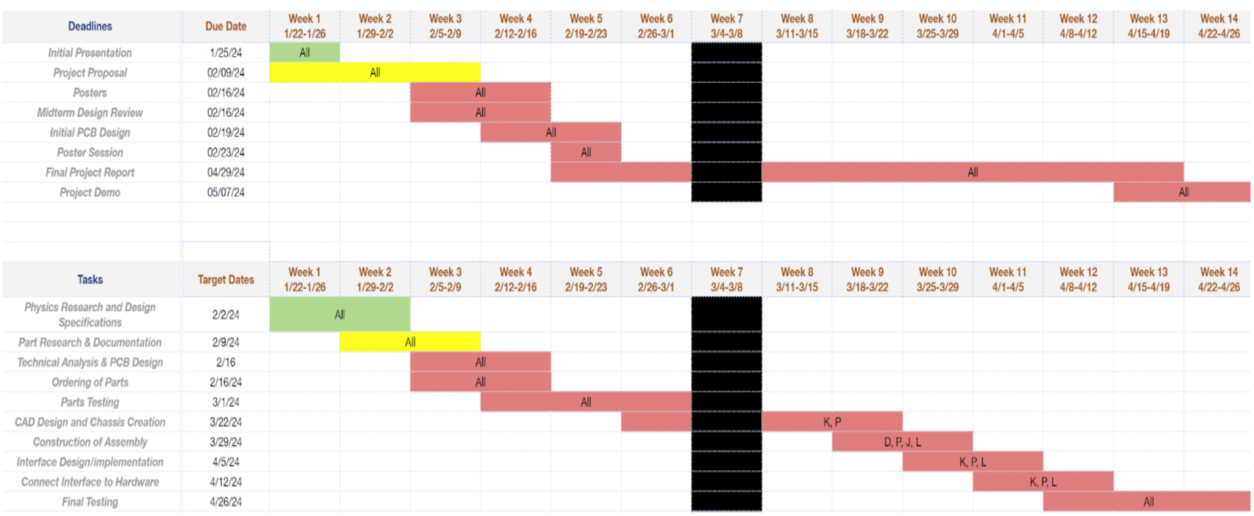


Figure 13. Initial Gantt Chart

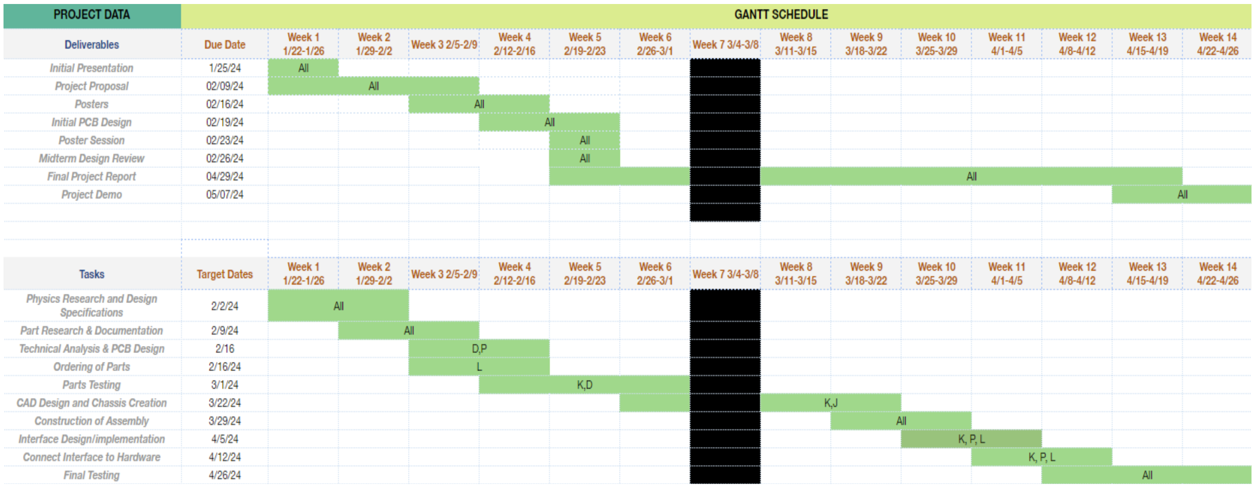


Figure 14. Finalized Gantt Chart

Visually, there does not seem to be any differences between the initial and final Gantt Chart besides the color coding as red indicates not started, yellow indicating in- progress, and green indicating finished, we also modified the members involved in each task as each member is denoted by their first letter of their name. However along the way, our group had to make some improvisations as our initial plan involved creating the chassis first and then attempting to create the code for the separate components to work. However, due to complications in both creating the chassis and obtaining the materials necessary for it, our group proceeded to start the coding and embedded process in order to compensate for the chassis production being slightly later than our initial expectations. In terms of our deliverables, nothing really changed and our group managed to meet every deliverable needed on time. It is also important to note that the project demo has not been done at the time of this Final Report, however our group is well prepared enough for it thus explaining why it is marked green in the Gantt Chart.

Costs

The detailed cost breakdown for each component used in the final version of the self-balancing bike is available in the appendix. Here is a summary of the finalized costs for key parts:

- STM32G071RB: $15.00
- Adafruit Bluefruit LE UART Friend: $17.50
- Feetech Positional Servo: $5.50
- MPU6050 IMU: $6.50
- Dantoma 11V rechargeable Batteries: $27.50
- Reaction Wheel DC Motor 3500 RPM: $25.00
- Back Wheel DC Motor 600 RPM: $15.00
- SparkFun Motor Driver Dual: $21.00
- PCB Manufacturing: $10.00
- Miscellaneous electrical and mechanical parts (sprockets, bearings, gears, etc.): $50.00
- 3D-printed components (steering column, wheels, etc.): $15.00

The total cost of materials for the bike is $208.00. Our total expenditure, including experimental parts, reached $428.70, reflecting approximately $220.70 spent on components that did not meet our expectations but served as backups. A significant cost reduction was achieved through the use of UVA's free 3D printing services and spare parts we found in the engineering laboratory. Considering the cost of PLA, it would cost around $15.00 for a 1 kg roll.

In terms of scalability and mass production for 1,000 units, the base cost would be $208,000. Through bulk purchasing from online vendors like Digikey, substantial discounts are available for components like the MPU6050, batteries, and servos, potentially reducing costs by up to 50%. This could result in savings of $20,000 just for these components. While the STM32 microcontroller offers smaller bulk discounts, up to 10%, savings could still amount to $1,500 for 1,000 units. However, the motor driver and Bluetooth module did not have bulk discounts available. To optimize costs, we are considering alternative parts with similar specifications but at significantly lower prices. This is possible because for these parts only the chips themselves are needed, so we could cut up to one 80%. By redesigning the PCB to accommodate these cheaper parts, we estimate a potential saving of $30 per unit, totaling $30,000 for 1,000 units. For 3D printing, the slow production speed of conventional methods forces us to consider professional mass production options, which offer both discounts and increased efficiency. In addition, transitioning to a company that provides complete PCB assembly will increase initial costs but streamline the manufacturing process.

To further scale production, investments in molds and mechanical assembly lines are necessary. Research estimates these capital expenditures could range from $50,000 to $100,000, specifically for toy focused facilities, depending on the facility's scale. In summary, while the unit cost is $208, scaling up production to 1,000 units could halve per-unit costs. However, this requires significant upfront investment in production facilities and equipment.

Final Results

Electrical System Testing Results

Following intensive electrical systems tests provided above in “Project Descriptions,” the self-balancing bike system has been confirmed to be fully functional with minimal operational errors, thus rendering the bike safe for use in educational environments. The implemented components demonstrate exceptional performance, fulfilling their intended functionality. All calculated voltages, resistances, and readings aligned with expectations as well. Initially, certain components were designed to be powered by two distinct voltage regulators; However, certain adjustments were made to the motors to accommodate the bike’s weight, which led to modifications to the hardware and electrical specifications on the printed circuit board. An issue also arose with the microcontroller for some time in regards to its integrated voltage regulator. Despite being specified to handle 11.1V, applying this voltage resulted in instability and pretty frequent burnouts/resets during multiple iterations of testing. Upon identifying this issue, the power source was promptly adjusted from 11.1V to its 5V input, supplying only the required voltage for the microcontroller. This modification effectively prevented potential future issues.

Software Testing Results

To verify the effectiveness of the balancing algorithm, we initially tested the tilt angle of the bike to ensure accurate readings. Subsequently, this raw tilt data was processed through a complementary filter to validate the consistency and accuracy of the data. The figures below illustrate a comparison between the raw and filtered tilt data.

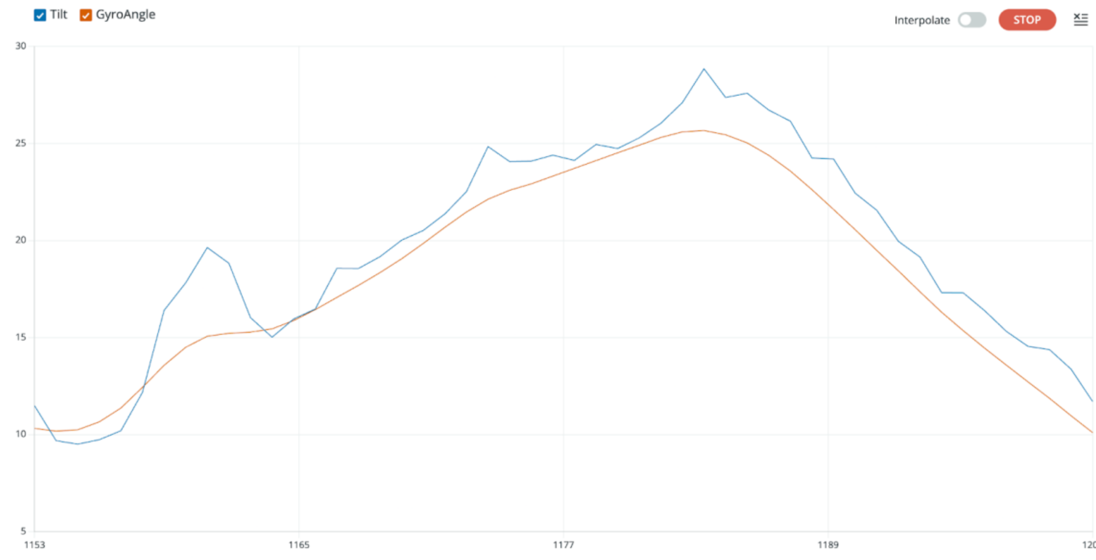


Figure 15. Filtered Angle (orange) vs. Raw Angle (blue)

As depicted, the filtered angle effectively smooths out fluctuations in the raw tilt data, eliminating abrupt changes in the angle. This demonstrates the efficacy of our complementary filter in maintaining a stable readout. Following this, we utilized the filtered angle to determine the "target angle," which is the desired angle the bike needs to achieve to remain balanced. This target angle is calculated based on our PID tuning equation and represents the corrective stance the bike should adopt to counteract any imbalances. The figures below demonstrate the operation of the reaction wheel algorithm:

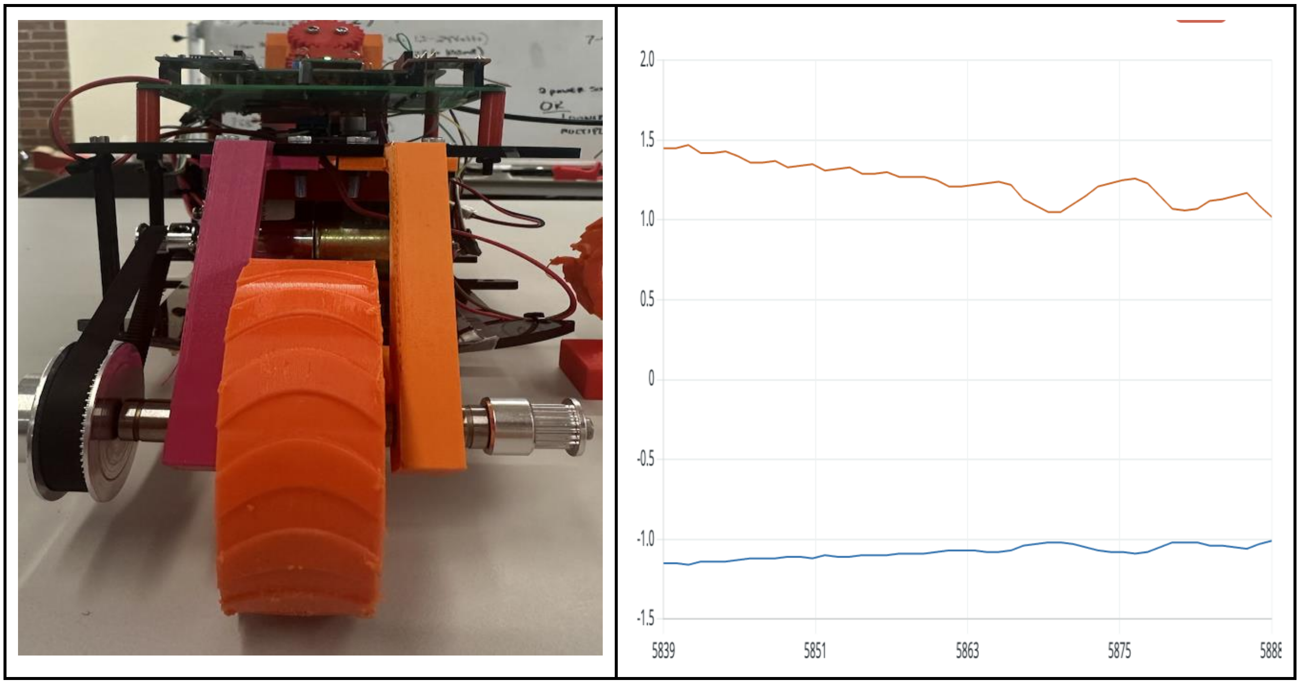


Figure 16. Filtered (orange) Vs. Target (blue)

The comparison reveals that the filtered angle (orange) and the target angle (blue) are nearly equivalent in magnitude but opposite in polarity (+1.5 VS. -1.2). This alignment indicates that the bike recognizes its central balance point and adjusts to maintain this position. The angle values are also close to 0 degrees, which is the central stable point.

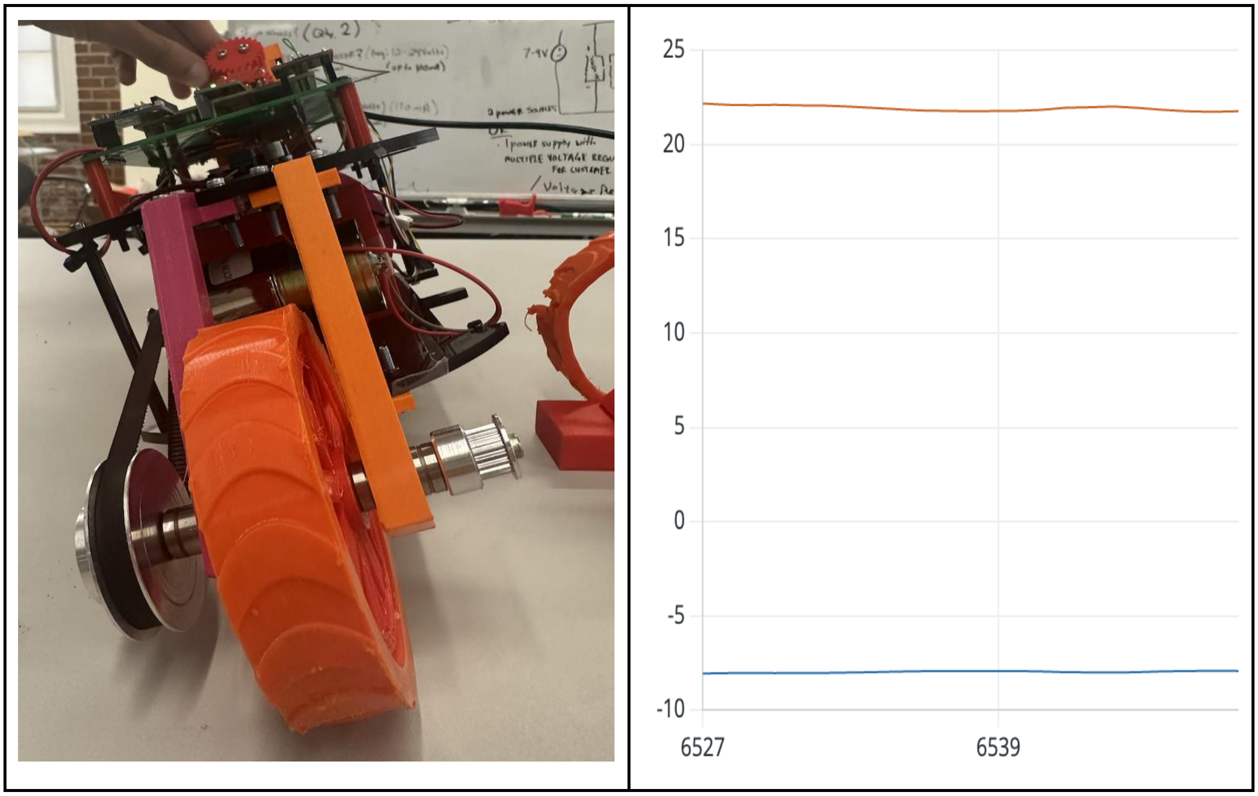


Figure 17. Bike Target Angle When Tilted to the Left

When the bike is tilted to one side, the complementary angle indicates a tilt of approximately 23 degrees, validating the sensor's accuracy. The system responds by adjusting the target angle negatively to counter the tilt and guide the bike back towards a neutral position. As the bike approaches 0 degrees, the target angle is progressively reduced to fine-tune the balancing act and prevent overshooting the equilibrium.

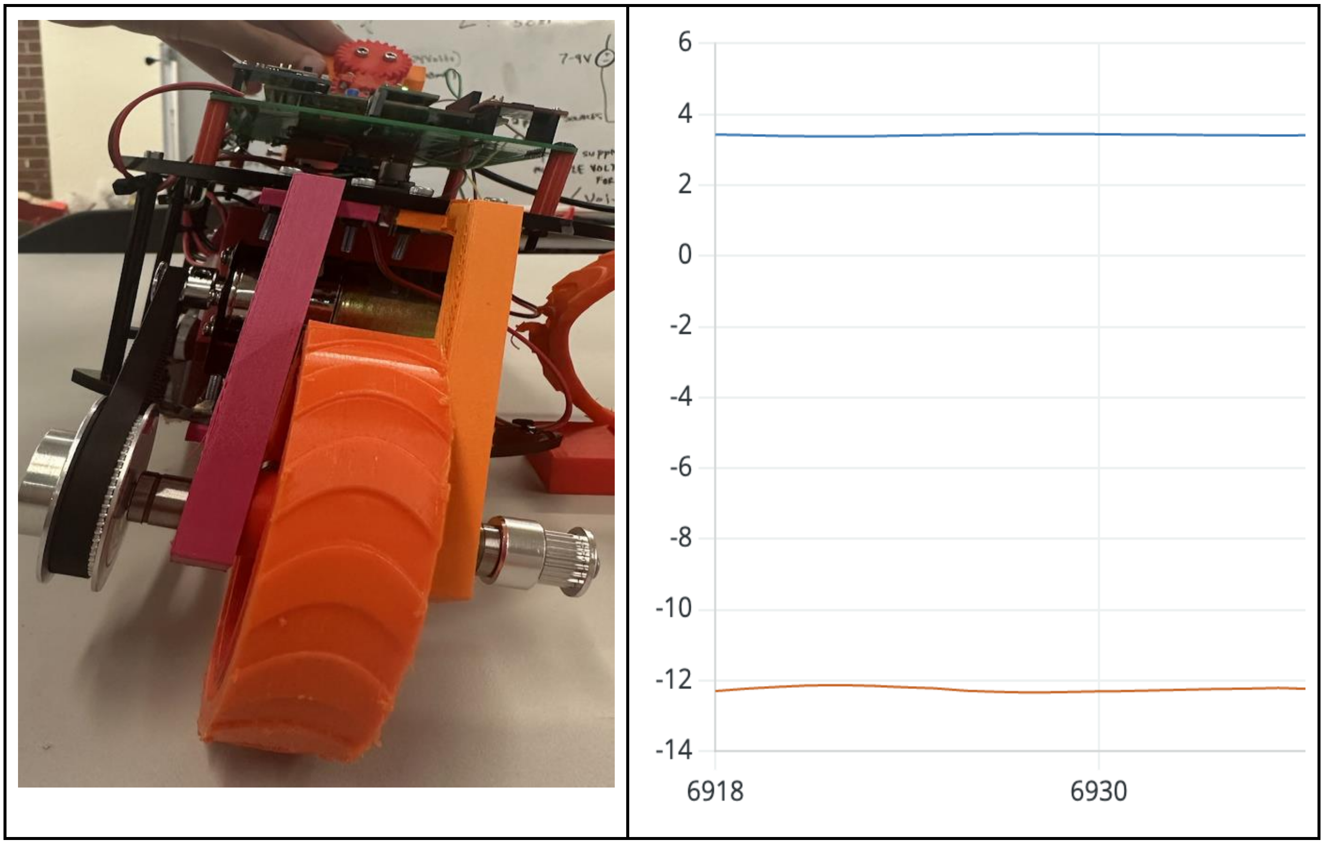


Figure 18. Bike Target Angle When Tilted to the Other Side

The response is consistent when the bike leans to the opposite side. The target angle shifts positively, compensating for the bike's lean in the negative direction.

These results confirm the reliability and precision of our balancing algorithm in real-time adjustments, ensuring the bike efficiently returns to and maintains balance under varying conditions.

In terms of Bluetooth and mobile app communication, the primary requirement is ensuring that the mobile app can accurately locate the correct Bluetooth module, establish a connection, and initiate precise data transmission. Below, we present the final user interface of the iOS mobile application:

Figure 19. Final IOS bike control application UI/UX

The design objective for the application's user interface (UI) is to maintain simplicity to avoid confusing users, particularly children. Upon launching the app, the initial screen (left) displays a 'not connected' status. To connect to the bike, users simply tap "Scan for Devices". If within range, the app will display the appropriate Bluetooth module for the bike. By selecting the "Adafruit Bluefruit LE" banner, the user connects to the bike and gains control capabilities. Controlling the bike is straightforward: pressing the up or down buttons adjusts the bike’s speed, and toggling the "RXN" button enables or disables the reaction wheel. The semi-circle on the right visualizes the servo's angle for turning the bike. Based on multiple users who used the application, they all agreed the app is "pretty intuitive to use", which meets our goal.

To validate the interface, we confirmed the transmission of correct information from the app to the STM32 microcontroller, as detailed in the "Implementation" section. The STM32 is programmed to receive a specific four-digit code separated by a "!" as a terminator, which represents the character "h" in the prints. Verification of correct receipt is illustrated below:

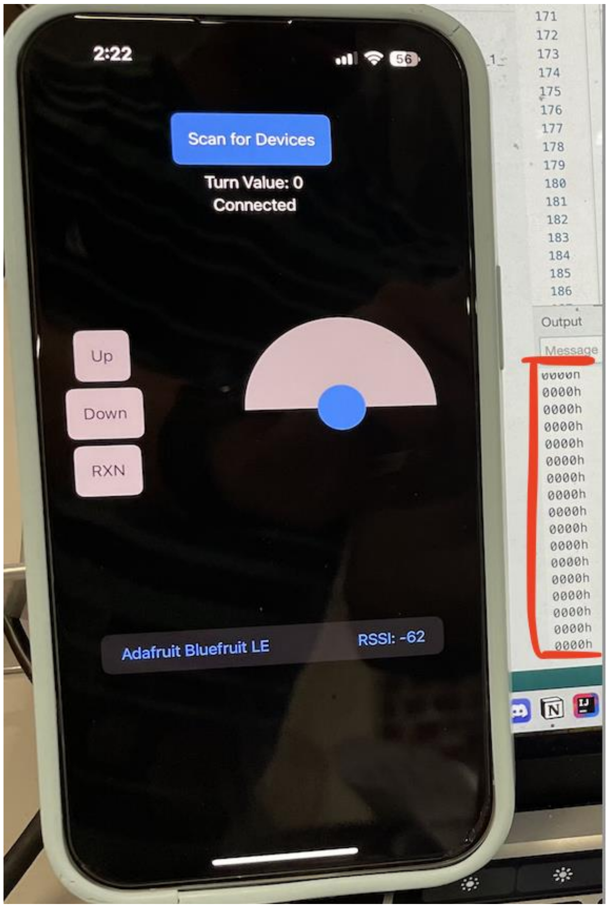


Figure 20. Bluetooth confirmation initial connection

The displayed digits initially read "0000", indicating no changes have been made within the app—a result that aligns with our expectations.

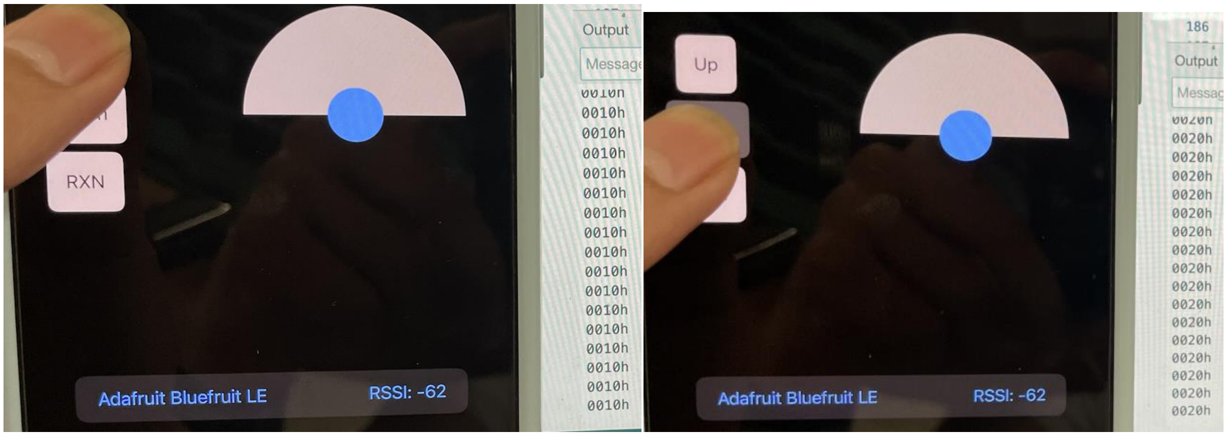


Figure 21. Up (left) and Down (right) button pressed.

When the up button is held down, the third digit changes to "1", confirming that the STM32 correctly receives the command to drive forward. The repeated printing of this command is due to the STM32’s continuous loop that operates every few microseconds; it will persist as long as the button is pressed. Similarly, pressing the down button changes the third digit to "2", indicating a reverse command.

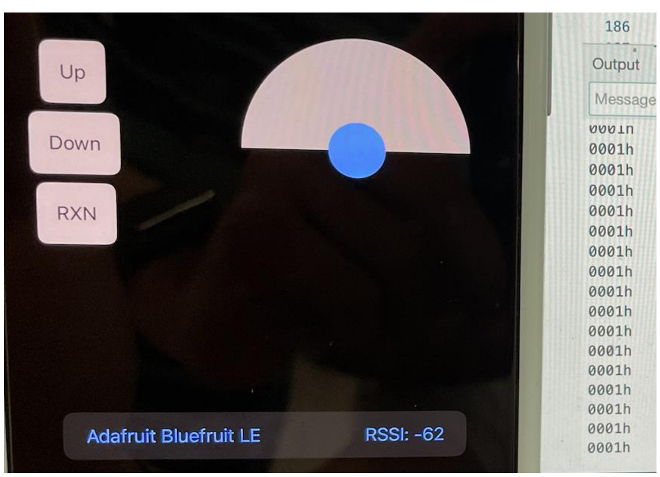


Figure 22. “RXN” button toggled to turn on/off the reaction wheel

Toggling the reaction wheel button "RXN" changes the fourth digit to "1", and it will stay as such until toggled again.

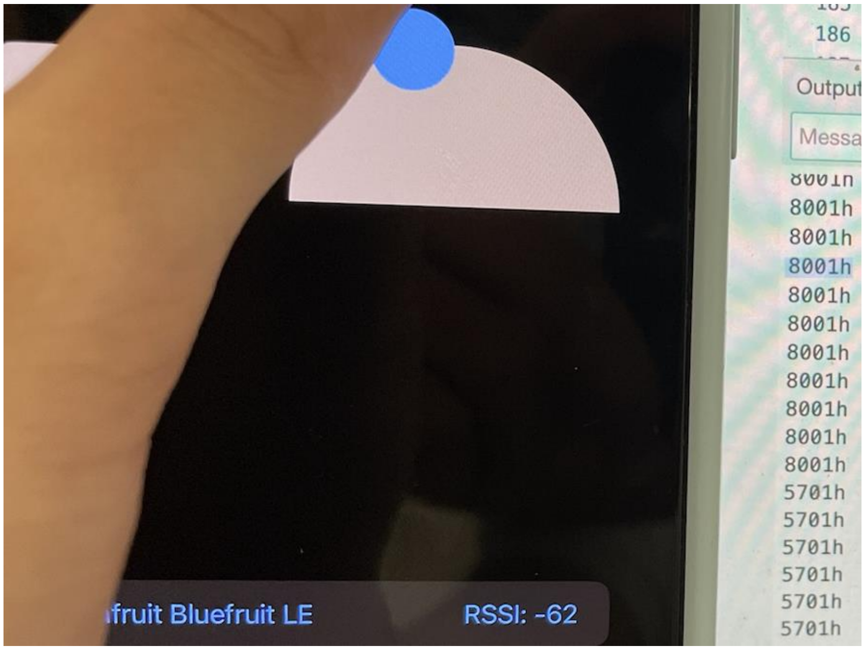


Figure 23. User turning servo.

The servo control knob adjusts as expected, with the first two digits representing integers from “01” to " 99". The farther to the right the knob is turned, the smaller the number, and vice versa.

These test results demonstrate the seamless integration and correct functionality of our iOS application in controlling the bike via Bluetooth.

Overall Testing Results

Almost all initial goals were met. One is able to control the forward and backward movement as well as the steering of the bike through an ios app. The instructions sent to the STM are executed fast enough due to minimal processing such that there is no data loss and all instructions are executed. Even with rapid change of instructions and simultaneous steering and driving, the STM is able to execute the user's instructions as well as create instructions for itself to balance by rotating the reaction wheel. The instructions are consistent, with the same input resulting in the same output. Throughout the entire testing process, no physical problems have been identified including electrical or relating to the chassis. Regarding balancing, the bike is consistent in its ability to balance when still and driving and allows for resistance against reasonable and time limited external forces. Though the bike is able to steer, it's ability to balance while steering and balance while steering and driving is very poor and almost non-existent. The chassis of the bike is strong enough to meet its needs and throughout thorough testing, there seems to be no notable stress on any of the parts. The bike is able to drive very slowly due to a weaker driving motor to the extent that it is unable to drive over any electrical cords. The battery is somewhat loose in its cage and is able to offset the center of mass of the bike when not secured with tape. Overall, the bike is able to complete all of the outlined goals well except for balancing while steering.

Future Work

The project our group created went very smooth overall, however there are some steps that could've been improved to ensure a smoother iteration in the future. This includes resolving the problems outlined in the "Problems Encountered" section. This project can be expanded by a larger and real life iteration of a bike, this iteration would realistically be able to help people with needs and disabilities mainly ones involving their legs or a lack of. We had a lot of expectations coming into the project that came into difficulties, mainly involving the parts that we ordered and their functionality, some were faulty after several uses, some showed that it did not fit our criteria for the bike to balance, and some caused us to reinvent components of the bike at times. As a group we would advise to be more careful with the testing and ensure edge cases in case of overheating or overpowering the circuit boards, to be more careful with the evaluation of the components that are used, and to be prepared to change some or if not most of the project during certain situations. Specifically, we did not take into account the effect of center of mass on the balancing of the bike; several times the placement of the heavier parts needed to be changed such that the center of mass of the bike along the front axis (while facing the bike) was even. Another problem was the wheels of the bike and the non-electrical / code parts of the bike in general; we discounted the importance of the bike's ability to balance itself depending on the width of its wheels. If there are to be future iterations of our project, we do think that it is important to focus on getting a high quality light brushes (brushless motors are lighter and have a higher starting torque which allows for quire precise movements of the reaction wheel) motor to ensure that the reaction wheel can spin fast enough, and to either buy or create the reaction wheel big and heavy enough to make the balancing easier. Some differences would need to be made if the bike is scaled to be usable by humans. Some include the removal of bluetooth control, since the human would be able to steer the bike, a chassis more protective of the hardware in case of falls, and a more robust balancing algorithm such that it accounts for the user's mass. The encountered problem that gave the most trouble was the placement and mass / size of the chassis parts. We should have accounted for this since it would've allowed for the algorithm to be more forgiving if the bike was already able to balance to an extent on its own.

References

[1] "Attitude control," NASA, https://spinoff.nasa.gov/spinoff1997/t2.html

(accessed Apr. 23, 2024).

[2] T.-L. Liao, S.-J. Chen, C.-C. Chiu, and J.-J. Yan, "Nonlinear Dynamics and control of a cube robot," MDPI, https://www.mdpi.com/2227-7390/8/10/1840 (accessed Apr. 23, 2024).

[3] M. A. Rosyidi, E. H. Binugroho, E. R. Charel, R. S. Dewanto, and D. Pramadihanto, "Speed and Balancing Control for Unicycle Robot," IEEExplore, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7860969 (accessed Apr. 23, 2024).

[4] "Chassis for a remotely controlled vehicle," Justia, https://patents.justia.com/patent/9592724 (accessed Apr. 23, 2024).

[5] "Methods of completing a remotely controlled model vehicle system with a separate controller," Justia, https://patents.justia.com/patent/8672726 (accessed Apr. 23, 2024).

[6] "Servomotor control systems and methods," Justia, https://patents.justia.com/patent/9360846 (accessed Apr. 23, 2024).

[7] W. Mrozik, M. A. Rajaeifar, O. Heidrich, and P. Christensen, “Environmental impacts, pollution sources and pathways of spent lithium-ion batteries,” Energy & Environmental Science, https://pubs.rsc.org/en/content/articlelanding/2021/ee/d1ee00691f (accessed Apr. 23, 2024).

[8] A. Isa et al., “A review on recycling of carbon fibres: Methods to reinforce and expected fibre composite degradations,” MDPI, https://doi.org/10.3390/ma15144991 (accessed Apr. 23, 2024).

[9] A. Kumar and P. Pastore, “Lead and cadmium in soft plastic toys,” ResearchGate, https://www.researchgate.net/publication/288358511_Lead_and_cadmium_in_soft_plastic_toys (accessed Apr. 23, 2024).

[10] “ASTM F963 requirements,” U.S. Consumer Product Safety Commission, https://www.cpsc.gov/Business--Manufacturing/Business-Education/Toy-Safety/ASTM-F-963-Chart (accessed Apr. 23, 2024).

[11] Toshiba, “Driver IC for Dual DC Motor Manual,” Toshiba, https://www.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf (accessed Apr. 26, 2024).

[12] Handson Technology, “XD3420 Motor Specification.,” Handson Technology, https://handsontec.com/dataspecs/motor_fan/XD3420-Motor.pdf (accessed Apr. 26, 2024).

[13] B. Siepert, “LSM6DSOX, ISM330DHC, & LSM6DSO32 6 dof imus,” Adafruit Learning System, https://learn.adafruit.com/lsm6dsox-and-ism330dhc-6-dof-imu/downloads (accessed Apr. 25, 2024).

[14] K. Townsend, Bluefruit Le Spi Friend - Adafruit Learning System, https://learn.adafruit.com/downloads/pdf/introducing-the-adafruit-bluefruit-spi-breakout.pdf (accessed Apr. 26, 2024).

[15] “FEETECH FS90R Micro continuous rotation servo,” Pololu Robotics and Electronics, https://www.pololu.com/product-info-merged/2820 (accessed Apr. 25, 2024).

[16] DFROBOT, Adjustable DC-DC Buck Converter Manual, https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2308/DFR0379_Web.pdf (accessed Apr. 26, 2024).

Appendix

Project Budget Outline

Item	Amount	Cost	Total	Weight(lb)	Bought	Using
DC Motor (251 rpm)	1	29	29	0.452	Yes	
DC Motor (159 rpm)	1	19.9	19.9	0.5	Yes	
Servo	1	6.15	6.15	0.1	Yes	yes
Accelerometer (I3G4250DTR)	1	11.09	11.09	0.00033	Yes	
Bluetooth module	1	11.58	11.58	0.001	Yes	
Motor Driver (L6230Q)	1	4.65	4.65	0	Yes	
STM32	1	0	0	0	Yes	yes
Batteries (rechargable pack)	2	27.54	55.08	0.3	Yes	yes
Front Wheel	1	4.86	4.86	0.018	Yes	
Back Wheel	1	12.42	12.42	0.32	Yes	yes
Screws for reaction wheel	#TODO	0	0	0	Yes	yes
Chassis (3d printed)	1	0	0	3	Yes	yes
Reaction wheel (3d printed)	1	0	0	0.5	Yes	yes
Buck Converter	1	4.9	4.9	0.04	Yes	Yes
PCB Fabrication and Shipping	1	0	0	0.1	Yes	
Other Nessesary Tools	#TODO	0	0	n/a	Yes	
Extraneous Cost (Shipping, extra parts, etc)	0	0	0	n/a	Yes	
Motors V2	2	25.09	50.18	2.2	Yes	yes
Bluetooth Module V2	1	10.39	10.39	0.011	Yes	yes
Accelerometer V2	1	6.49	6.49	0.004	Yes	yes
Pin Headers (1-pin)	30	0.24	7.2		Yes	yes
Pin Headers (2-pin)	5	0.28	1.4		Yes	yes
Voltage Regulator	1	5.35	5.35		Yes	yes
Switch	1	2.1	2.1		Yes	yes
Voltage Regulator New	1	2.78	2.78		Yes	yes
Female-Male Connector	1	7.97	7.97		Yes	yes
positional servo	1	5.39	5.39		Yes	yes
Bluetooth Module	1	17.5	17.5		Yes	yes
Accelerometer V3	1	11.95			Yes	yes
Battery Charger	1	12.99	12.99		Yes	yes
Voltage Regulator	2	2.78	5.56		Yes	yes
Battery Charger	1	12.99	12.99		Yes	yes
Motor pulley sprocket - 8mm	1	13.99	13.99	0.2	Yes	Yes
6mm d-shaft	1	10.77	10.77	0.7	Yes	Yes
Motor pulley sprocket - 6.35mm	1	11.99	11.99	0.2	Yes	Yes
6-32 Hexagon Socket Screws	1	6.99	6.99	0.49	Yes	Yes
Female Header 8 Position	5	0.66	3.3		Yes	Yes
Female Header 9 Position	5	0.68	3.4		Yes	Yes
Female Header 10 Position	5	0.99	4.95		Yes	Yes
Steering struts	1	7.99	7.99		Yes	Yes
Standoffs	1	9.09	9.09		Yes	Yes
Bearing	1	11.99	11.99	0.13	Yes	Yes
M3 20mm Screws	1	11.99	11.99		Yes	Yes
Female Header 10 Position	5	0.66	3.3		Yes	Yes
Motor Driver	1	21.03	21.03		Yes	Yes
BUDGET			500			
TOTAL SPENT			428.7			
REMAINING FUNDS			71.3			