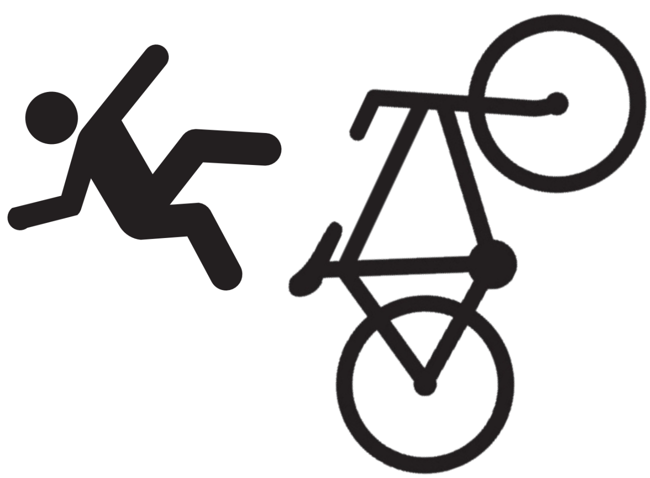
WheelieKing Trainer
Project Report

June 6, 2017

Prepared for: *Greg Sesser*

Team: Mustang Wheelie
Nathaniel Fox
Harold Hall
Thomas Niemisto

Advisor: *Jim Widmann*



Disclaimer

Team Mustang Wheelie (“the Team”), and its members or sponsors are not responsible for any death, injury, or damage resulting from or relating directly to or indirectly to any part of this document and the information contained within it. The content delivered on the following pages (“the Project”) is provided “as-is” and is intended for academic and informational purposes. Any execution of steps outlined or assembly of components detailed in this Project may be done by qualified persons.

Any unauthorized redistribution of the information contained in this document may be punishable under applicable local regulations.

Table of Contents

This page intentionally left blank.

List of Tables

List of Figures

List of Nomenclature

A (units)	Amperes
ADC	Analog to Digital Converter
F	Force
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
IPxx	Intrusion Prevention (dust)(water)
lb (units)	pound(s)
mAh (units)	milliamp-hours
R	radius
RPM (units)	Rotations Per Minute
V (units)	Potential Difference (Volts)

Executive Summary

In this report we will detail the design and implementation process of the WheelieKing Trainer project, a device that helps people learn how to do wheelies on a bicycle by preventing backward falls. Formal project requirements are specified, followed by the ideation and iteration process to meet those requirements. The components and methods used to create the device are described in detail. The results of the development process and usage test results of the device are included. Appendices at the end of this report include references, supporting analyses, and project management and timeline details.

I. Introduction

Team Mustang Wheelie's mission is to design and develop an exciting and reliable product that allows aspiring wheelie artists around the world to safely perfect their bicycle wheelie skills. The primary stakeholder in this project is Greg Sesser, a longtime wheelie enthusiast and entrepreneur.

The problem we aim to solve is that of preventing backward falls while performing bicycle wheelies due to a rider's inexperience or mistakes. The device must not impede normal operation of the bike or change dynamics of the wheelie.

We aim to build a system to help prevent injuries in addition to helping riders learn to do a wheelie on a bicycle. We define a "wheelie" as the state where the front wheel on the bicycle is not in contact with the ground for an extended period of time. A backward fall can occur when the angle of the bicycle wheels with respect to gravity exceeds a threshold of about 70 degrees (which varies by user and type of bicycle).

Learning to do a wheelie on a bicycle is a dangerous but rewarding endeavor. The key method of preventing injury while doing a wheelie is to apply a light braking force on the rear wheel (by using the bicycle brakes) when the bicycle tips too far backwards. However, many beginners do not act quickly or reliably enough to correct their angle. A wheelie bar (an existing product we discuss later) can prevent the user from tipping over backwards, but it does not provide the other features mentioned and they can prevent the user from learning to find the proper balance point on their own.

Because the product we aim to design does not interfere with the wheelie process except in preventing a backwards fall, it will help the user learn where their point of balance is, so that they will eventually feel comfortable doing wheelies without the training device. Above all else, our product will not impede normal riding.

Customer Requirements

- Device will reliably prevent a backward fall due to wheelie overshoot.
- Device will not interfere with the normal operation of the bike.
- Device will not interfere with the experience of performing except as a safety device.
- Device should be lightweight, weather-resistant, easy to install and compatible with the majority of common bicycle types, particularly inexpensive ones.
- Device shall have a production cost of less than $100 in its final form factor.

Engineering Requirements

Table 1: Project Wheelie King Trainer Formal Engineering Requirements

Spec. #	Parameter Description	Requirement or Target (units)	Tolerance	Risk	Compliance
1	Adjustable Actuation point	45-90 (degrees)	Max	M	T
2	Brake Cable Tension	10 (lb)	Max	M	T
3	Force application time	0.5 (second)	Max	L	T
4	Battery Life	200 actuator cycles	Min	L	T
5	Impact shock resistance	Free Fall onto concrete from 2 ft	Min	L	T
6	Universal installability	90 % (of bikes)	+- 10%	L	A,T
7	Weight	3lbs	Max	M	T
8	Force application time variability	.25 (second)	Max	M	A,T
9	Actuation angle variability	10 (degrees +-)	Max	L	T
10	Actuation response delay	.25 (second)	Max	L	A,T
11	Dust*/Waterproofing	IP55	Min	L	T
12	Operating Temperature	0°C to 45°C	+-10%	L	T

Note: numbers preliminary and subject to change. Compliance Codes: T - Testing, A - Analysis
** Actuator not included in dustproofing*

Project Management

Duties will be divided up among the project engineers according to individual areas of strength. Nate Fox will be primarily involved in the development of the electrical hardware and will supplement development of the control software. Additionally, Nate Fox will be the primary handler of contact between the team and the sponsor. Harold Hall will lead the development of the control software and interfacing with the mechanical hardware. Thomas Niemisto will lead the mechanical development of the product.

Table 2: Team Mustang Wheelie Division of Responsibilities

Name	**Major**	**Strengths / Responsibilities**
Nate Fox	Computer Engineering	Programming (C, Java) Microcontroller interfacing Electrical components Sponsor contact
Harold Hall	General Engineering	3D Printing Microcontroller interfacing Programming (C, Java, Python, C#, C++, Assembly)
Thomas Niemisto	Mechanical Engineering	Electromechanical interfacing Wheelie Dynamics Analysis 3D CAD Modeling MatLab Manufacturing Prototype Alpha Tester

According to the Mustang Wheelie Team Contract signed by the team members on 10/6/2016, the team members agree to:

A. Attend all ENGR-459/60/61 class meetings prepared and willing to participate
B. Keep teammates apprised of individual progress made toward the project goals
C. Budget additional time outside of class to work toward project goals as needed
D. Produce quality work and provide quality documentation of work
E. Make decisions in accordance with consultation of teammates, sponsor, adviser, etc.
F. Have a positive attitude
G. Communicate early if any of these items cannot be followed on a particular occasion.

All team decisions will be made by consensus, which is achieved by discussing the facts of each option and weighing the benefits/drawbacks of each choice. In the event of a conflict, the team shall consult the team advisor Jim Widmann.

A Gantt chart was developed to plan the timeline of the project's development. Time targets are estimates. The Gantt chart is included in Appendix G.

II. Background

The most common way to alleviate the risk of learning to wheelie is to use a wheelie bar. There are several existing patents for these, shown in Table A1 in Appendix A. However, wheelie bars present several issues:

- They prevent the rider from acquiring a feel for the balance point as well as the point at which the bicycle is tilted too far backward to make a safe recovery.
- They add a large amount of weight to the rear axle, which modifies the balance point. When the rider is finished with training and removes the wheelie bar, the balance point will change, so the rider will have to relearn the new balance point without the safety of the wheelie bar.
- If the rider uses the bike for other tasks, such as commuting, the wheelie bar may interfere with those other uses, unless it is uninstalled for that task and subsequently reinstalled for the rider's next wheelie training session, which is a cumbersome process.
- Wheelie bars are uninspiring to many people wanting to learn a wheelie. Wheelie bars look like crutches for bicycles.

Another existing approach is to use a wheelie simulator machine that sits stationary in a garage, and these have nearly identical issues to the wheelie bars.

- They don't let the rider acquire a feel for the balance point in a real-world wheeling application. The rider will have to learn the balance point for the simulator, and then re-learn it for a bike.
- The simulator is a relatively expensive, dedicated machine that serves no other tasks— such as commuting— and takes up lots of space in the garage. When the rider is finished with training, he/she will have to remove it, either by selling it or disposing it.
- Simulators are uninspiring to many people wanting to learn a wheelie.

Our design will mitigate these issues to provide a superior wheelie training experience.

III. Design Process

The Wheelie King Trainer system consists of three primary components: the brake actuator, the angle sensor, and the microcontroller. Several different designs for the actuator and different types of microcontrollers were proposed.

To decide on specific designs/components, we created decision matrices and analyzed the important qualities of each design/component. Because the majority of accelerometer/gyroscope sensor boards are similar in form and function, we focused on comparisons of brake actuator designs and microcontrollers.. These decision matrices are included in Appendix B.

The chosen design of the brake actuator will interface with the bike's existing rear braking system, which will minimize cost, installation time, and maintenance of the device while being quick and effective in correcting an impending backfall.

Bike Braking Systems

The most common brake actuation system used on bikes today is the Bowden brake cable system, which consists of a cable that runs inside a housing and is engaged by pulling on one end of the cable (under tension) with respect to the housing. The housing will have an equal and opposite reaction force (under compression). Because the motion and force of the cable is relative to the housing, the brake can also be actuated by pushing the housing relative to the cable (as shown in Figure 1 below) and our brake actuator will utilize this principle by pushing any terminus of the the housing away from the the cable stop socket in which it is seated.

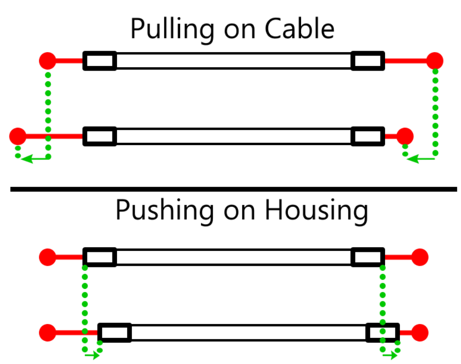


Figure 1: Example of Analogous Bowden Brake Cable Motion

One of the most common calipers used on commuter bikes and casual mountain bikes are cantilever-type rim calipers with two individual arms pulled together by a cable. The packaging of our actuator will be optimized for the Avid caliper (shown in Figure 2 on the next page) which is a common quick-release caliper used on bikes, yet it will be compatible with any caliper that uses a Bowden brake cable, including those used for disc brakes, albeit with less optimal packaging.

Per instructions from our sponsor, we will not incorporate compatibility with other less common bike braking systems, such as hydraulic brakes.

Figure 2: Bowden Cable and Avid Caliper

Method of Actuation

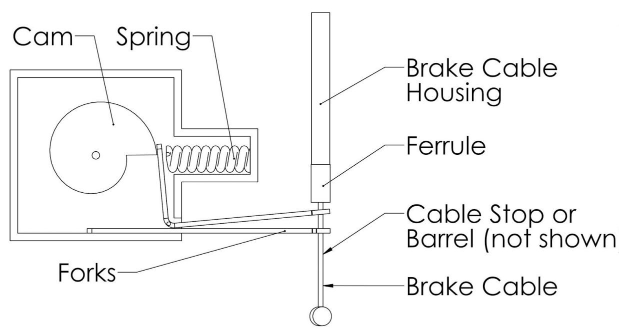


Figure 3: Brake Actuator Design Concept

The actuator motion will be produced by a constantly-increasing-radius cam (shown in Figure 3 above) that loads a spring with potential energy. When the microcontroller commands an actuation cycle, the cam will release the spring energy to the brake cable, which will apply braking force. The cam will rotate a full 360 degrees back to its original position, which will release brake cable tension and reload the spring for the next actuation cycle. Each rotation of the cam will be completed in 0.5 seconds, and it will not consume power when it is armed. The advantage to this design over servos, solenoids, linear actuators, and pneumatic actuators are

- the brake actuation latency is minimal (though the entire cycle will be about 0.5 seconds, the force will begin to be applied nearly instantaneously)
- it draws the least amount of power, which is important for minimizing the size of the battery pack and power circuitry, and
- the microcontroller and actuator can share the same power supply.

Cam Analysis

To select an appropriate motor to drive the cam, the amount of torque required to drive the cam needed to be calculated. As shown in Figure 4 below, the spiral shape creates a contact surface that is not normal to a vector that points straight towards the axis of rotation. This results in a normal-to-surface force on the cam by the follower that does not intersect the axis of rotation, thereby creating a torque load on the cam. The motor must be able to drive this load to move the follower, plus any torque load created by tangential friction between the cam and the follower.

A constantly-increasing-radius cam was sketched in SolidWorks to calculate the arm to graphically calculate the arm at the various radiuses, as shown in Figure 4 below. Note that the arm remains nearly constant to within 3 significant figures of 0.158r.

Radius	Arm
r	0.1572r
1.25r	0.1578r
1.5r	0.1582r
1.75r	0.1585r
2r	0.1586r

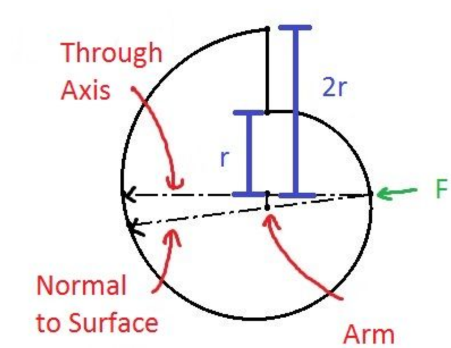


Figure 4: Constantly Increasing Radius Cam

The cam can be scaled to any size that best fits its application. The tradeoffs in selecting a size are:

- A larger cam will provide a larger range of motion
- A smaller cam will require less torque to drive
- A smaller cam will require less space and can be packaged in a smaller space

The size we selected is r = 0.5 inches, and with a desired nominal cable tension of 10 lbf, this cam will need an estimated 40 oz-in of driving torque to load the spring. Our desired actuation cycle is a mere 0.5 seconds, so the cam will need to rotate at a speed of 120 RPM.

Motor Selection

As derived from the cam analysis, the operating point of the motor will be 40 oz-in. At 120 RPM. The motor we selected for this task is the Pololu 37D x 70L mm with a 50:1 gear reduction (Figure 5). As shown in the Speed vs. Torque graph in Figure 6 below, the desired operating point is within the operating limits of this motor. The motor will need about 10 volts for this operating point, but can accept up to 12 volts without exceeding the rated voltage. If the actuation speed needs to be lowered, the motor will be pulse-width modulated to an appropriate speed.

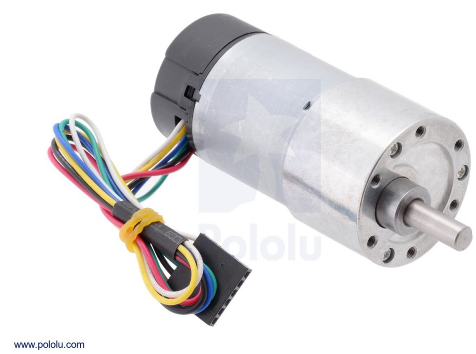


Figure 5: Pololu 37D x 70L mm 50:1 Metal Gearmotor

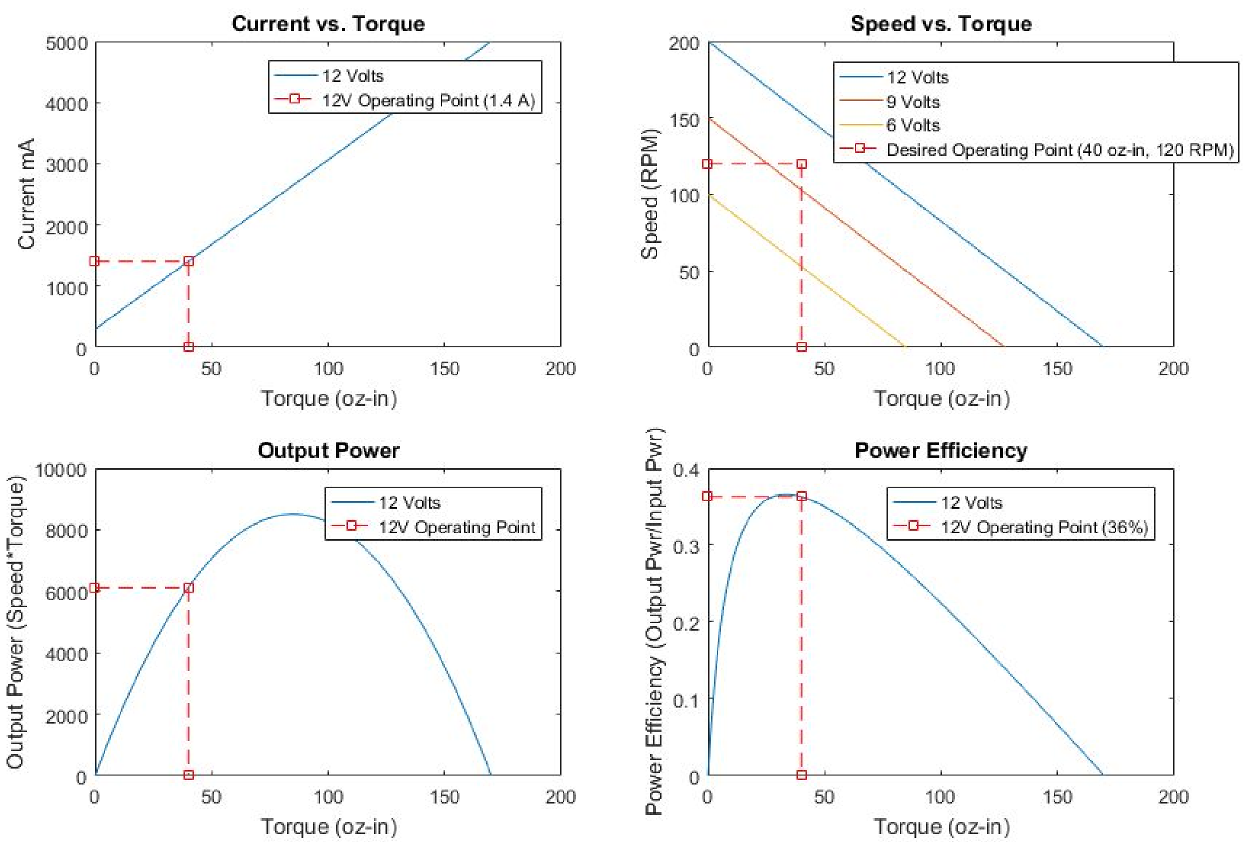


Figure 6: Motor Operating Point Performance Metrics

Actuator Mechanism Design Process

An early conceptual design of the brake cable actuator is shown in Figure 9 below. The cam would load a spring with energy. When the microcontroller signals a brake actuation cycle, the cam would release the spring energy, and the spring would apply its force to the brake cable via forks that seat between a cable housing terminus and its socket. The cam would rotate back to its starting point to release the brake cable and reload the spring for the next actuation.

Figure 7: Cam Actuator Design Concept Solid Model

The next iteration of the actuator design accounted for its mountability to the bike. It would have been impractical to have the actuator and the bike brake cable rigidly attached via a linkage, so the linkages were relocated remotely and mechanically linked to the actuator via a Bowden cable. While other kinds of springs were considered for this iteration (tensional and torsional, to name a few), we decided on a compression spring as it would provide a direct, tensional force on the Bowden cable. The spring pushes the piston towards the cam. When an actuation cycle is desired, the cam will release the stored spring energy which will pull on the cable. The cam then continues its cycle to store energy back into the spring for the next actuation cycle. The housing was designed ex post facto to conform and contain all the internal components. The linkage design was left for a later time when the actuator would begin going through manufacture.

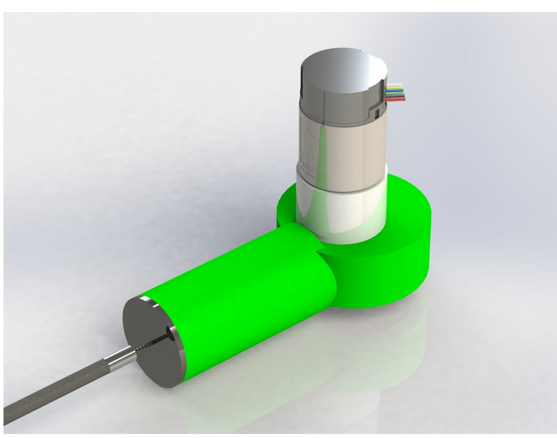
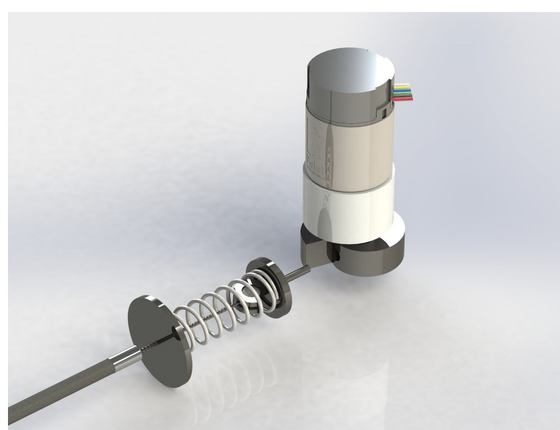



Figure 8: Refactored Actuator Design Pulling on Brake Cable

The final iteration modified the design to be low cost and easy to manufacture. The two primary components that needed attention to accomplish this were the piston and the housing. The piston would be made from stock cylindrical wood with holes drilled out to hold the cable barrel nipple and a rod for the cam follower. A foot for the cam follower would be made of the same plastic material as the cam to remedy any differences in material hardness at the point of contact as the point of contact would have high-pressure sliding friction.

The cam is housed in a stock aluminum box, and the spring and piston is housed in PVC pipe. During testing, the spring preload can be adjusted by swapping the PVC pipe for another one with a different length. The end of the PVC pipe is capped off with a block of wood. A limit switch was added to the housing so that the microcontroller could calibrate the quadrature encoder position data on power-up. The usage of the limit switch is described in the following state diagram:

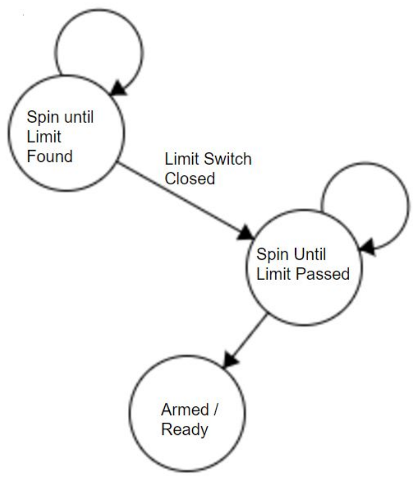


Figure 9: Limit Switch Motivation - Cam State Diagram

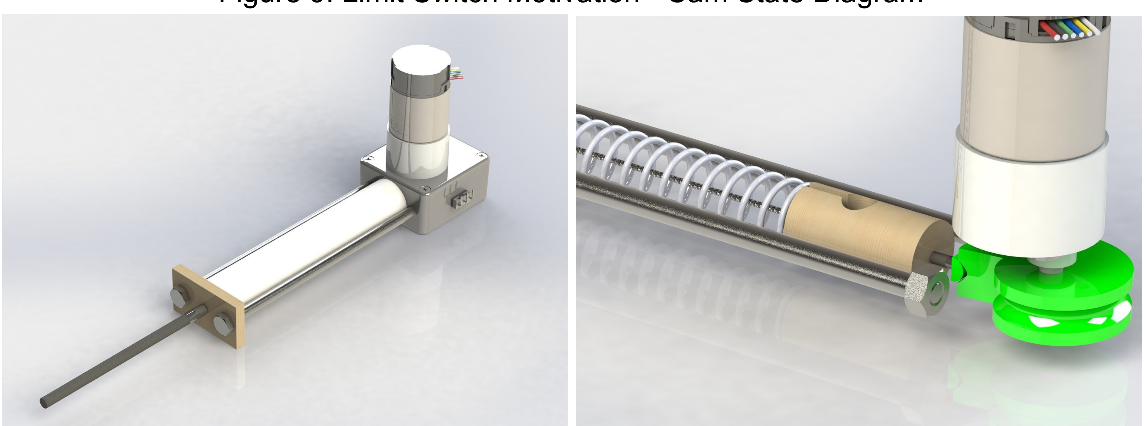


Figure 10: Actuator design modified to lower prototype cost.

The next step was to interface the the actuator cable with the bike brake cable. In principle, this required two linkages: One to link each cable together, and another to link each cable housing together. We decided on making the linkages out of 3D-printed plastic because of their complexity and the need to make several design iterations quickly. If the linkages were to be mass-produced, they would likely be made of hardened steel using the investment casting or die casting manufacturing process.

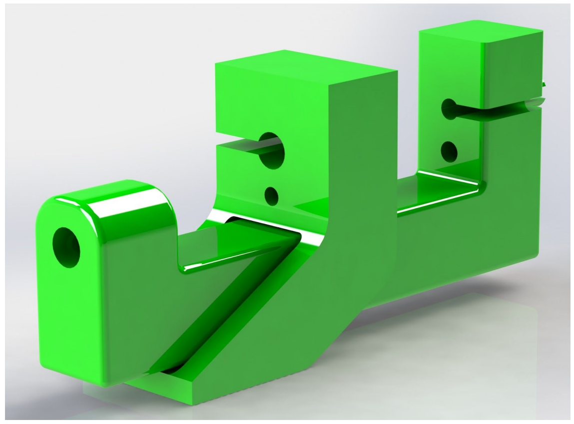


Figure 11: Brake Cable Linkages Solid Model.

Actuator System Final Design Overview

The actuator consists of a spring that is compressed by a cam to store energy. When an actuation cycle is needed, the cam releases the spring energy which pulls on the cable. The cam then reloads the spring to prepare for the next actuation cycle. The cam is driven by a brushed DC motor with a gear reduction and a quadrature encoder. A limit switch is positioned about 90 degrees ahead of the piston to give the microcontroller the ability to find the cam position when the system is first turned on.

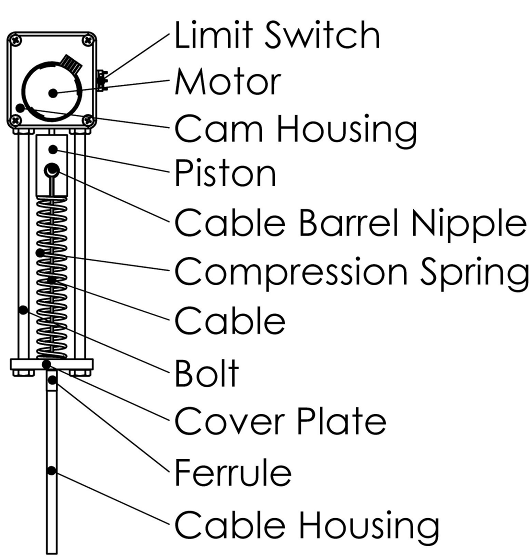


Figure 12: Actuator components. PVC pipe not shown.

Table 3: Actuator Components

Component	Model	URL
Motor	Pololu 50:1 Metal Gearmotor 37Dx70L mm with 64 CPR Encoder	https://www.pololu.com/product/2824
Cam Housing	Hammond Manufacturing 1550Q	https://www.digikey.com/product-detail/en/hammond-manufacturing/1550Q/HM1204-ND/2211554
Limit Switch	E-Switch SS075Q102F035V2A	https://www.digikey.com/products/en?keywords=EG4929-ND
Spring	McMaster-Carr 302 Stainless Steel Precision Compression Spring	https://www.mcmaster.com/#9435K147

The cable, cable housing, and ferrule are generic Shimano stock components used for brake cables; The bolts are generic hardware store components; and the piston is made of stock cylindrical wood with a steel rod cam follower, both from the hardware store.

Battery and Electrical System

With the motor chosen, a power delivery system needed to be designed around it. Implementing a system of kinetic recharging would introduce too much additional complexity to the project, so a simple rechargeable battery system was chosen. The system will consist of lithium-ion batteries and will provide the motor with the required 6-12V and will support the stall current of 5A for short periods of time.

The Wheelie King system will be supported by a 12-volt 2850 mAh battery pack comprised of three lithium-ion 18650 batteries in series. Each 18650 cell provides a peak charge voltage of 4.2V. The typical (nominal) operating voltage of each cell is 3.6V, and the cutoff voltage is 2.5V. This means that with three cells in series, the peak voltage will be 12.8V, the nominal voltage will be 10.8V, and the system will shut off when the voltage drops to 7.5V. Because the DC motor operates between 6V and 12V, and the Elegoo R3 has an onboard 5V regulator that can accept between 6V and 12V, the entire system will run on the 7.5V-12.8V supplied by the battery pack without any additional regulating hardware. A basic circuit diagram is shown in Figure 10 below.

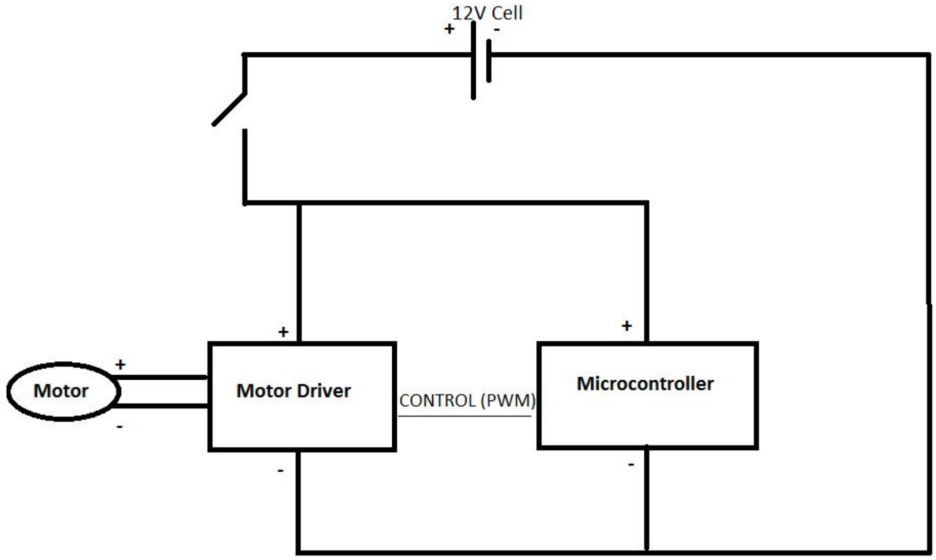


Figure 13: Power Circuit Initial High-Level Block Diagram Design

During testing, it was observed that flyback current from the motor had a catastrophic effect on the electronics assembly. To protect the digital devices, a flyback diode was added in parallel with the motor:

Figure 14: Flyback Diode Bridge

The battery pack will be housed in a protective cradle such as the one shown in Figure 11 below.

Figure 15: Battery Protection Cradle

A cradle such as the one depicted above will include overcharge detection (triggering at 4.325±0.025V), over discharge protection (triggering at 2.4±0.08V), and over current protection (6±1A). It is imperative that this hardware be included so that there is little to no risk of a catastrophic cell failure. The cradle will be wrapped in shrink-wrap to make the battery pack weather-resistant.

The battery pack will be charged with a 120V AC > 12.6V DC 1A charging adapter which will also include overcharge, short circuit, and over-temperature protection, such as the one pictured in Figure 12 below.

Figure 16: Battery Pack Charging Adapter

The connection between the main unit and the actuator will consist of a waterproof durable cable that will allow for the main unit to be removed from the bicycle to be charged. The charging port will feature a rubber plug that will protect it from dust and water. The cable connection will feature detection of whether the cable is properly secure and will disable the so that the port will not short if there is a disconnection.

Sensing, Control and Trigger System

Inertial Measurement Unit

In order to detect the instantaneous pitch angle of the bicycle we have selected an Inertial Measurement Unit (IMU) that integrates a three-axis accelerometer and a three-axis gyroscope into a single unit that is accessible through a serial interface. The MPU6050 that we selected also has a motion coprocessor onboard that will do all the signal processing necessary to know the current angle at which the unit is rotated. This allows us to get a much better angle reading on a processor with limited floating point capabilities like the ATmega328p which we will be using. While the specific calculations done by the motion co-processor on the MPU6050 are proprietary and the specifics are not known to us, we have done testing and determined that the resultant values are not subject to drift over time, do not lag behind the actual position of the sensor by any noticeable amount, and are unaffected by sudden linear accelerations applied to the IMU. A discussion of how the IMU will be used to detect the bike angle is included in Appendix F.

Our wheelie training solution will be using the Digital Motion Processing (DMP) systems on the MPU6050 to do motion processing rather than develop our own algorithms. This will allow for a much faster development time without an unreasonable increase in system cost. It will also allow the use of a slower and less expensive main processor (such as the ATmega328P used by the UNO) since the processor will not have to do complicated signal processing and filtering. The DMP on the MPU6050 also does sensor synthesis for us to cover the normal limitations of accelerometers and gyroscopes such as drift and temperature based error.

There will also be a limit switch within the actuator housing to detect the position of the cam. This will allow for us to inexpensively detect that the cam has reached the ready position. Our prototype will include an encoder to allow us to track the exact position of the motor to allow for more precise testing to be done.

Our motor driver system provides an analog feedback signal of the current used by the motor. Using this we can detect stall conditions where the motor is drawing much higher current. When we detect the motor stalled for too long, we will be able to stop driving the motor to prevent overheating, save power, and allow for whatever is stalling the motor to be removed. The user will be notified through the user interface when this occurs.

Control System

The control system for our design will be based around the Elegoo UNO controller. The Elegoo UNO is fully compatible with and analogous to the Arduino UNO at a much lower price point. This will allow us to use standard parts so as to minimize development costs, limit unexpected behaviour, and accelerate prototyping. In the final product, we would be able to design a custom control board that would integrate the ATmega328P and the other digital components onto a single printed circuit board (PCB).

The UNO provides digital and analog inputs and digital outputs, as well as integrated hardware for I2C (two-wire interface), an onboard voltage regulator, a USB programming interface, and runs at 16MHz with a logic voltage of 5V. The I2C interface will be used to communicate with the motion tracking chip (as connected in Appendix D Fig. D2), a digital input will read the power button with a digital output controlling the indicator, and an analog input will be used to read the amount of current being used by the motor.

Motor Driver

With the motor and power system chosen, we were able to select a suitable motor controller for the system. Our motor has a 5 amp stall current, and it will run at 1.4 amps during nominal conditions at our desired speed and expected load. For the highest reliability, it needed to be capable of handling a continuous motor stall condition without negative side effects and should be capable of handling more voltage than the fully charged power system outputs. The Pololu Dual MC33926 Motor Driver Shield for Arduino (electrical schematic in Fig. D1) that we chose meets meet these requirements. While the specifications for the motor driver shield states a max transient current of 5A and a continuous throughput of 3A, the microchips used are actually capable of a full 5A continuous current if properly cooled. Since we are not properly cooling these chips they may overheat, but that is not a problem for these motor drivers. These motor drivers detect overheating in themselves and reduce their active duty cycle as temperature rises so that they will not overheat or go into full thermal shutdown as many other temperature aware motor drivers would. As such there is no chance of a permanent damage to these motor controllers, and only a risk of temporary performance reduction if we were to stall the motors for too long. However, another feature of this motor driver will allows us to detect stall conditions and stop driving the motor if it stalls. The driver features an analog feedback signal that is proportional to the current going through the motor. We can read this value using an analog input on the microcontroller to determine if the motor stalls.

Overall, the Pololu Dual MC33926 Motor Driver Shield for Arduino exceeds all of our specifications and is an excellent motor driver solution.

User Interface

The user interface is an important part of any consumer product, and our product is not exempt from this. The user interface for our product will primarily consist of systems located within the main unit. These systems include the trigger angle configuration dial and the power button and indicator.

Trigger Angle Configuration Dial

The Trigger Angle Configuration Dial (TACD) is a rotatable device that is used for the fine-tuning of what angle the user wants the device to trigger at as well as the much more coarse adjustment to account for different main unit mounting positions. This can be used without Wheelie Angle Calibration Assistant (WACA). Because the angle is completely user-definable, the WACA was deemed unessential. The user may simply experiment with the trigger angle settings when present at a particular practice area until the system triggers at the desired angle.

Power Button and Indicator

The power button and indicator thereof will be used for the operator to power on/off the device. The indicator light will show the current power status of the device. This will allow the user to know when the device needs to be charged, and to know whether the device is on or off.

Wheelie Angle Calibration Assistant (deprecated)

To account for differences in angles at which the product would be mounted to a user's bicycle, The Wheelie Angle Calibration Assistant was proposed. The "WACA" (Figure 13 on the next page) is a colored ring that will be included with the device. The image printed onto this will allow the user to better understand the angle to set the trigger angle configuration dial at. Quite simply, the ring will be placed around the dial, and will rotate with gravity like a gyroscope. The brown semicircle will align with the horizontal ground beneath the bicycle. This is a simple and elegant low-tech solution to compensate for variance in the pitch angle (with respect to gravity) at which the sensor box is mounted to the bicycle. This is a confusing and strange concept, but we promise it will make sense in implementation. Essentially, we're eliminating the need for a complex dynamic gravity reference vector adjustment calculation (which is potentially unreliable and difficult to test) with some hand-wavy pseudoscience based on a single step of user-input. Trust us.

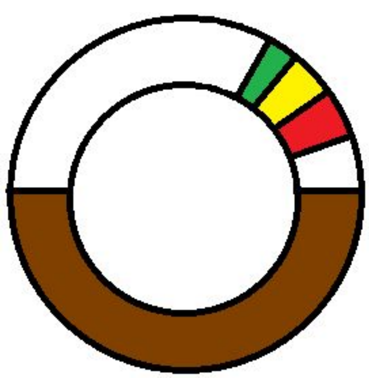


Figure 17: Wheelie Angle Calibration Assistant (WACA) preliminary design (deprecated)

Code Development

The code was adapted from the MPU6050 DMP demo code that included the required code to properly load the motion processing code onto the MPU6050. This demo code was part of the i2cdevlib repository maintained on GitHub by jrowberg.

We used a private GitLab repository to track changes in our code and allow for access from any computer.

For access to the WheelieKing code repository, send a request to natefox.av@gmail.com.

We also used the DualMC33926MotorShield and quadrature encoder libraries available through Arduino IDE to interface with our motor driver and encoder, respectively. From the MPU6050 demo code we utilize the Pitch of the Yaw-Pitch-Roll output to determine how far the sensor is tipped. This utilizes the gravity vector that is pulled from the IMU output. In order to allow for adjustment of the trigger point, we rotate the gravity vector based on the TACD potentiometer input.

To allow for better development and testing we added a serial command interface to our code that lets commands be sent over the serial console while the program is running. These commands could query several values and adjust or set the encoder setpoint.

It was found that the Elegoo UNO R3 ADC introduces serious nonlinearity into the analog readings of the potentiometer. This was with the ends of the potentiometer connected between 5V and ground, the analog input connected to the wiper(middle) pin, and ARef being tied to 5V. To correct this, a set of analog values for different angles was created and we used linear interpolation between these points.

In order to eliminate certain inconsistencies in the device's behavior we also decided to make it so that the device would not be able to trigger for a short time after triggering. This eliminated the edge-case scenarios where the device would trigger multiple times when passing over the trigger angle.

Installation

The motion sensor will be mounted near the rear axle of the bicycle between the chainstay and seatstay tubes to minimize tangential accelerations caused by the bike nosing up and down during a wheelie stunt. The motion sensor's orientation will be mounted on the bike's longitudinal plane and parallel to the chainstay tubes. The angle of the chainstay tubes with respect to the bike's horizontal frame of reference varies between different bike designs, and the user calibration process will account for this uncertainty.

The microcontroller and actuator will each be contained in an IP55-rated housing, which will not allow in dust that would disturb operation and will be secure against light water spray from any direction. The housing for the microcontroller will sit in a quick-release cradle that is mounted to the bike so that the microcontroller can be removed as an anti-theft measure when the bike is left unattended. The microcontroller cradle and actuator housing will mount to the bike tubes with either hose clamps, ratcheting tie-downs similar to cable ties, or velcro, all of which hold rigid to the bike yet are easy to install and remove.

A hypothetical design for the placement of the sensor / electronics enclosure is shown in Figure 8 below. The placement of the enclosure is not likely to deviate much from this depiction, though it may take on a different shape depending on constraints related to the proportions of the microcontroller, sensor board and battery pack.

Figure 18: Sensor/Controller Box Mounting Concept

Final Design Recap (Figures)

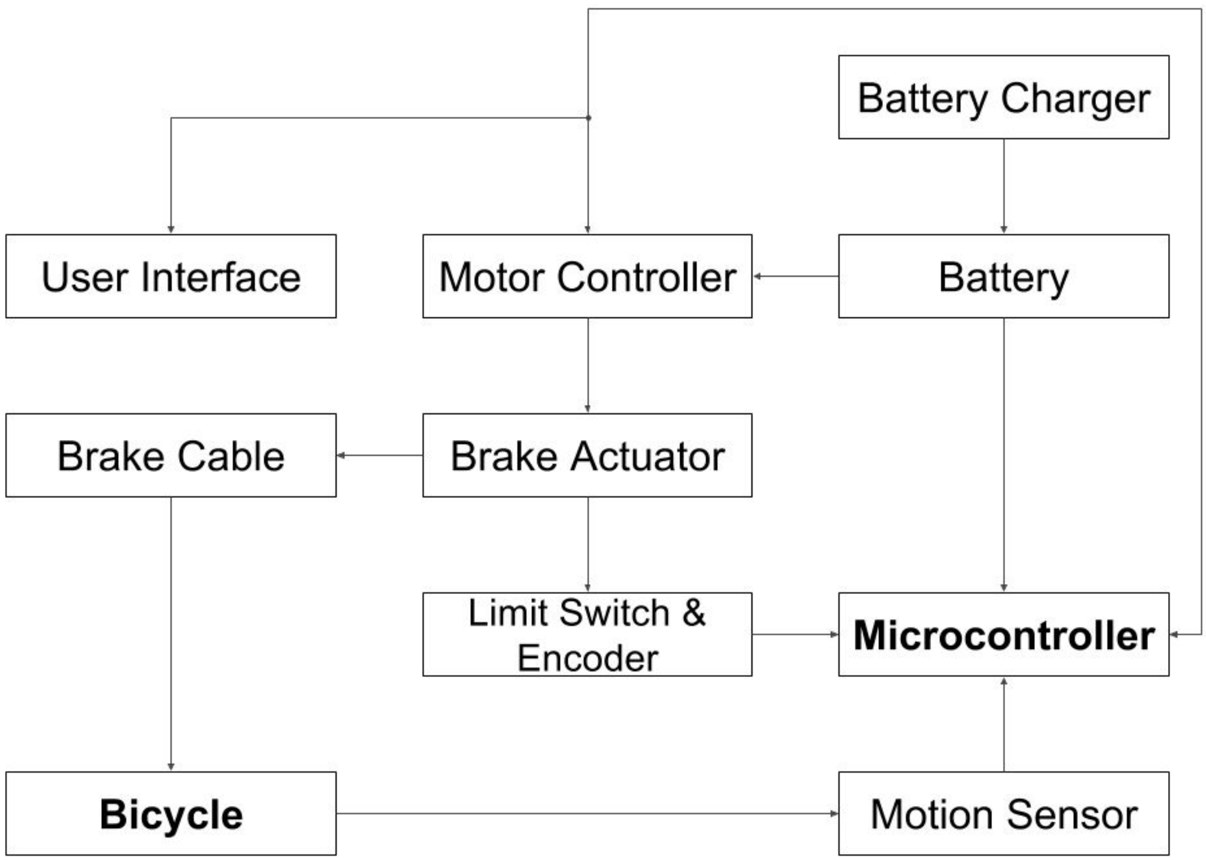


Figure 19: High Level System Functionality Diagram

Figure 20: Design Implementation and Integration with Bicycle

VI. Testing

In order to ensure reliability of the device, it will need to meet testing benchmarks. We are primarily interested in verifying that the firing of the brake actuator is consistent when the target pitch threshold is met. Additionally, we will ensure the device meets our engineering requirements.

Much of the testing can be accomplished in a lab setting. For example, to test the actuator firing, we will manually tilt the sensor box (not attached to a bicycle) to its target pitch and verify via software logs and visual confirmation that the actuator has fired. This particular parameter will be tested rigorously with at least 100 trials, since it represents the reliability of the product as a safety device. Once lab data has been collected and the product performs nominally, testing will continue with the product mounted to the bicycle and a skilled rider performing wheelies.

Additional parameters to be tested first in a lab environment include braking torque, force application timing and initial response latency, and angle detection variability. Weather resistance, operating temperature, and impact tolerance will also be tested in a lab environment, though the product's durability in real-world usage will prove more informative for these parameters.

Sample test result tables are included below.

Table T1: Actuator Firing At Target Pitch Suggested Test

Test Number	**Result**
1	PASS/FAIL
...	
100	PASS/FAIL
Pass Rate	x%

Table T2: Actuator Initial Response Latency Suggested Test

Test Number	**Result**
1	Time in Seconds
...	
10	Time in Seconds
Average	Time in Seconds

Test Results

Tests were done over time and in various settings. Rather than repeated trials, each parameter function was observed over the course of several days of bench testing and actual real-world device use.

Test Item	Target	Pass/Fail
Braking force applied to rear wheel upon each actuation cycle	>99% rate of successful halting of wheel in bench testing	PASS (no failures observed)
Adjustable trigger point	Consistent triggering at set point with at least 30 degree range	PASS (no inconsistencies observed)
Actuation Time	< 5 seconds	PASS (0.2 seconds)
Battery Life	> 200 Cycles	PASS

Table 4: Test Results

Conclusions and Recommendations

The device was test-driven and performed as intended. Brake actuation is nearly instantaneous, and it quickly releases brake force while arming for the next actuation cycle. User input was successfully implemented to provide an adjustable angle trigger point. With real-world testing by the project's sponsor, it was verified that the device functions as intended, and automatically lowers the nose of the bicycle upon wheelie angle overshoot.

The project's sponsor noted the team's success via email. An excerpt is included below:

> *"Make no mistake, what you have achieved with the Wheelie King so far is amazing. You definitely proved the concept works! It's a testimony to a great team effort."*
>
> -Greg Sesser via email, 6/3/2017

Recommendations for continuing improvements to this project:

- Condense the packaging
- Customize PCB fabrication
- Simplify installation
- Manufacture the cam, follower, and linkages out of hardened steel
- Use more spring force
- Modify the design principles to sum user and actuator braking forces together
- Add feedback to the user interface
- Use an adaptive braking control system to help maintain balance point, instead of bringing the rider all the way back down

Appendix A: References

Existing Wheelie Training Technology

Table A1: US Patents for Wheelie Training Devices

US Patent No.	Published Date	Title	Inventor	Legal Status
4,367,883	1/11/1983	Wheelie support	Ray C Anderson	Expired
5,330,221	7/19/1994	Bicycle wheelie balancing device	Steven W Sutton	Expired
6,149,179	11/21/2000	Wheelie stabilizer and safety device	Thomas B Long	Expired
6,530,598	3/11/2003	Anti-tip devices for wheeled conveyances including wheelchairs and method related thereto	Ronald Lee Kirby	Active
7,270,545	9/18/2007	Motorcycle wheelie simulator	Dori Milner	Expired
8,075,011	12/13/2011	Adjustable motorcycle wheelie device	Jeffrey Allen Duzzny	Active

Braking Systems:

- http://a.co/cNXz7nU
- http://www.cyclingabout.com/wp-content/uploads/2011/10/180055.jpg
- http://www.sheldonbrown.com/cables.html
- http://www.sheldonbrown.com/rim-brakes.html

Motor:

- https://www.pololu.com/product/2824

Appendix B: Design Decision Tools

		Engineering Requirements												Benchmarks
		Weighting (1 to 5)	Does not Explode	Minimize moving parts	Any failures occur in a safe manner	Prevents falling over backwards	Simple UI	Consistent angle reading	Consistent Brake actuation	Weighs less than __	Works on 95% of bicycles	Impact shock resistant	End user adjustable	Wheelie Bar
Customer Requirements	Safety	5	9	3	9	9	1	1	1			1	1	
	Lets you learn how to wheelie	5			1	9	3	9	9				9	
	Easy to install	3					1	1	1	0.5	9		3	
	works on my bike	4						1	1		9		9	
	maintainable	3	1	9	3	1	1	1	1		1	3	1	
	weather resistant	4	9	1	1				1			1	3	
	robust	3	9	9	3		3	3	3	1		9		
	lifespan	2	1	3		1	3					9		
	easy to use	4			3	3	9	3	3		3		9	
	light weight	1		3						9		3		
	reliability	4		9	9	9	1	9	9			1		
	Units													
	Targets													
	Benchmark #1													
	Benchmark #2													
	Importance Scoring		113	118	120	143	81	117	121	13.5	78	70	146	
	Importance Rating (%)		77	81	82	98	55	80	83	9	53	48	100	
● = 9	Srong Correlation													
○ = 3	Medium Correlation													
Δ = 1	Small Correlation													
Blank	No Correlation													

Figure B1: Quality Function Deployment Table

The QFD table was used to correlate items from the customer's requirements with specific engineering requirements. It also assesses the importance of the engineering requirements and guides our design process.

Decision Matrices

Microcontroller Matrix Criteria Key

1	2	3	4	5	6	7	8	9	10	11	12
Multi-Threading	IO protocols (SPI, I^2C)	IO capacity	RTOS capable	Price	Wireless Communication	Power Consumption	Programming Ease (Harold)	Programming Ease (Nate)	Size	Weight	Analog Input

Table B1: Microcontroller Decision Matrix

								Datum	
Micro-controller	Raspberry Pi	Snicker-doodle	Arduino	Beagle-bone Black	Parallax Propellor	MicroChip Pic	MSP 430	ARM M3	Snicker-doodle Black
Criteria	1	2	3	4	5	6	7	8	9
1	+	+	-	+	+	-	-	S	+
2	S	+	S	-	-	+	S	S	+
3	S	+	S	S	-	+	S	S	+
4	S	+	-	+	-	-	-	S	+
5	S	-	+	-	S	S	+	S	-
6	+	+	-	S	S	S	-	S	+
7	-	-	+	-	S	+	+	S	-
8	S	+	S	+	+	S	S	S	+
9	S	-	S	S	-	S	+	S	-
10	-	-	-	-	S	+	S	S	-
11	-	-	-	-	S	+	S	S	-
12	-	+	S	-	-	S	S	S	+
Sum of +	2	7	2	3	2	5	3		7
Sum of -	4	6	5	6	5	2	3		5
Sum of S	6	0	5	3	5	5	6		0
		Size, weight, power consumption, and price are not as relevant for a prototype.							
Sum of +	2	7	0	3	2	2	1		7
Sum of -	1	1	3	2	5	2	3		1
Sum of S	5	0	5	3	1	4	4		0
		Unfortunately, Snickerdoodles are not yet available for purchase nor is the ARM M3. The ARM M3 should be available mid December.							

		Therefore we will use the Raspberry Pi for development (due to the high availability of documentation), but it isn't as good for the final implementation..
		Price is important for the final implementation. Due to price and ease of use, the Arduino platform will be used for the final design.

Table B2: Brake Actuator Concept Decision Matrix

	Solenoid Linear Actuator	Pneumatic Actuator	Cam Actuator	Servo	Stepper Motor
Fast Response	D	+	+	-	0
Low Power Consumption	D	-	+	+	0
Lightweight	D	-	+	0	-
Easy to Install	D	-	0	0	0
Fail-Safe	D	+	0	-	-
Reliable	D	+	-	-	0
+	0	3	3	1	0
0	0	0	2	2	4
-	0	3	1	3	2
TOTAL	0	0	2	-2	-2
DECISION			X		

Appendix C: Bill of Materials

Table C1: Bill of Materials

Item	**link/source**	**cost**	**quantity**	**used [0/1]**	Prototype	**used [0/1]**	Final
Motor							
Cam Motor	https://www.pololu.com/product/2824	$39.95	1	1	$39.95	1	$39.95
Microcontroller							
Elegoo UNO R3	https://www.amazon.com/Elegoo-ATmega328P-ATMEGA16U2-Compatible-Arduino/dp/B01EWOE0UU/	$10.86	1	0	$0.00	1	$10.86
Raspberry Pi w/ power supply	https://www.amazon.com/CanaKit-Raspberry-Micro-Supply-Listed/dp/B01C6FFNY4/	$43.99	1	1	$43.99	0	$0.00
MicroSD Card	https://www.amazon.com/gp/product/B004ZIENBA/	$7.99	1	1	$7.99	0	$0.00
Mechanical Hardware							
Metal?		$10.00	1	1	$10.00	1	$10.00
Rubber?		$5.00	1	1	$5.00	1	$5.00
Plastic?					$0.00		$0.00
Electrical Hardware					$0.00	0	$0.00
Stackable headers for Motor Shield	https://www.pololu.com/product/2748	$2.25	1	1	$2.25	0	$0.00

Wires for motor		$5.00	2	1	$10.00		1	$10.00
Power regulator for Pi	https://www.pololu.com/product/2119	$4.95	1	1	$4.95		1	$4.95
IMU	https://www.sparkfun.com/products/13762	$14.95	1	0	$0.00		0	$0.00
1x Makerfire GY-521	https://www.amazon.com/gp/product/B00NH8Z6BU/	$7.09	1	1	$7.09		0	$0.00
10x MPU-6050 GY-521	https://www.amazon.com/dp/B01NA7GJP4?psc=1	$32.99	0.1	0	$0.00		1	$3.30
Motor Driver/controller					$0.00			$0.00
Motor Driver Shield for Raspberry Pi	https://www.pololu.com/product/2753	$7.49	1	1	$7.49		1	$7.49
Motor Driver Shield for Arduino	https://www.pololu.com/product/2503		1	1				
				Totals				
					$178.70			$91.55
				Safety Factor	3			1.5
					$536.10			$137.32

Appendix D: Vendor-Supplied Schematics and Specifications

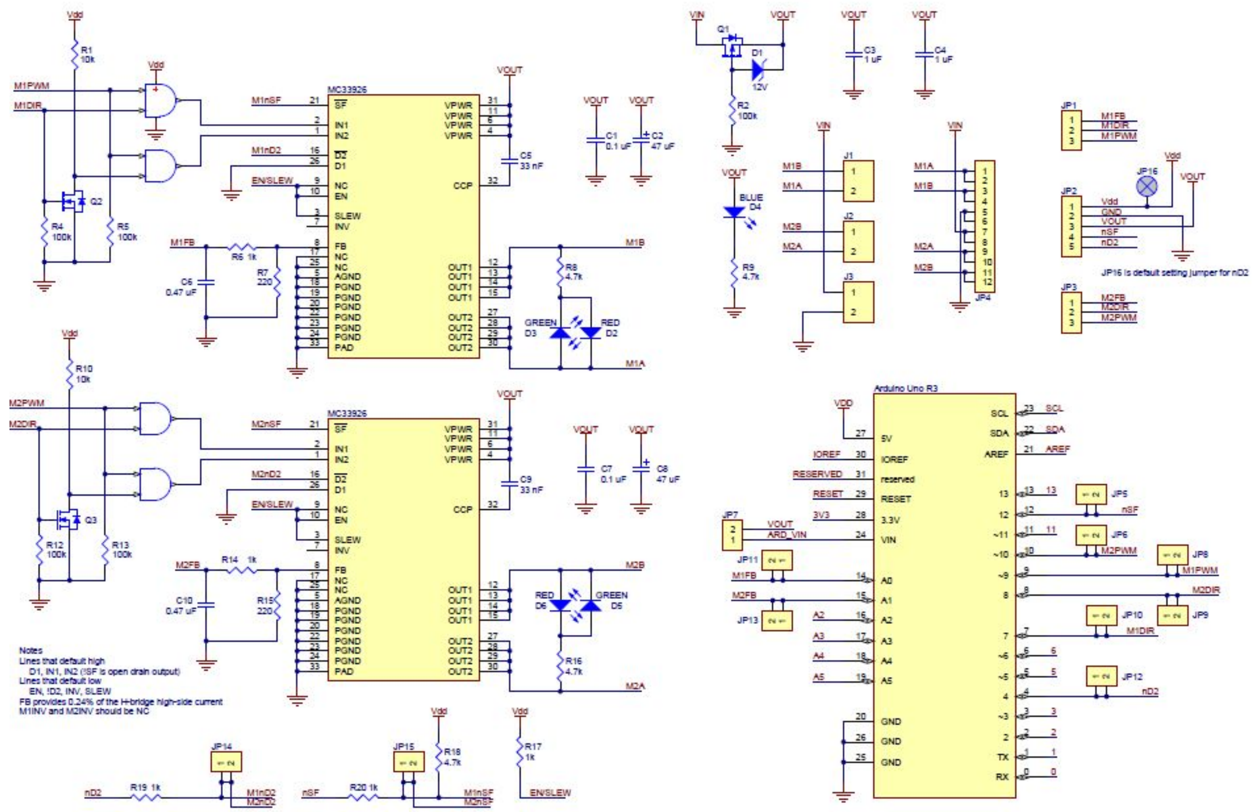


Figure D1: Schematic diagram for the Dual MC33926 Motor Driver Shield for Arduino (Pololu)

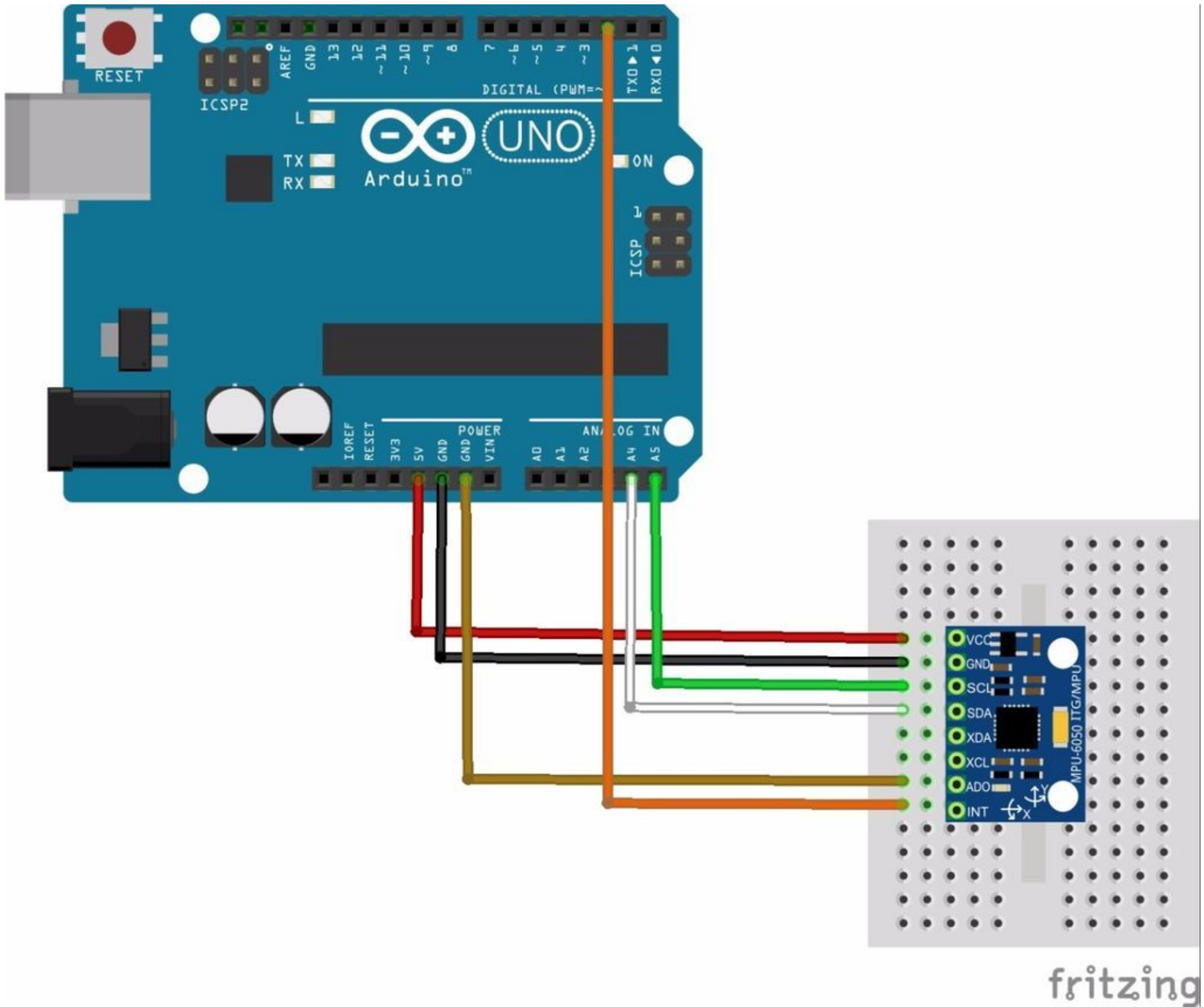


Figure D2: IMU Wiring Diagram for Arduino Uno. Source: http://www.instructables.com/id/MPU6050-Arduino-6-Axis-Accelerometer-Gyro-GY-521-B/

Appendix F: Supporting Analysis

We will sense bike pitch (angle) through the accelerometer sensor. The accelerometer will provide data representing two (or three) vector components (depending on whether we use a 2-axis or 3-axis accelerometer, but the third axis is unused). We will interpret the pitch based on the vector toward Earth's gravity. This angle θ of a vectors can be calculated from the components as follows:

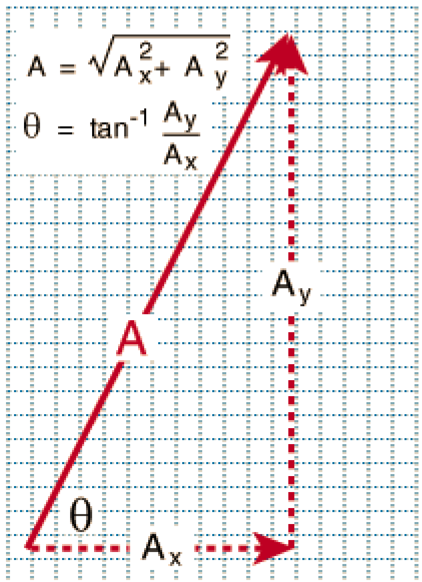


Figure F1: Component Vector Summation & Angle Calculation

We will combine this calculation with data from a gyroscope so that we can be highly confident that the bike's pitch has changed (rotational motion has indeed occurred). This is because the accelerometer will be sensitive to disturbances such as bumps in the riding surface, overshoot from the sudden upward movement of the front wheel, etc.

Appendix G: Gantt Chart

Figure F1 shows the Gantt Chart Hierarchal table, which contains the task names, resources (the individuals the task is assigned to), the target start date and the target completion date. This data is represented graphically on the chart itself in figure G2. Following the timeline shown in G2, the project shall be completed by June 2016.

		Name	Duration	Start	Finish	Predecessors	Resources
1		⊟**Wheelie King Trainer**	**170d?**	**09/22/2016**	**06/21/2017**		
2		⊟**Conceptual Prototype**	**15d**	**09/22/2016**	**10/12/2016**		
3		Drawings of mechanical components	15d	09/22/2016	10/12/2016		Thomas Niemis
4		Drawings of electrical circuits	15d	09/22/2016	10/12/2016		Nate Fox[50%]
5		Drawings of code task & state diagrams	7.5d	09/22/2016	10/03/2016		Nate Fox[50%],H
6		Foam board mock-up	15d	09/22/2016	10/12/2016		Harold Hall[50%
7		⊟**Proof-of-Concept (POC) Prototype**	**45d**	**11/15/2016**	**02/14/2017**	**2**	
8		Dynamics Calculations	45d	11/15/2016	02/14/2017		Harold Hall[25%
9		Mechanical CAD Drawings	30d	11/30/2016	01/31/2017		Thomas Niemis
10		⊟**Functional Prototype**	**86d?**	**02/14/2017**	**06/21/2017**	**7**	
11		⊟**Functional Prototype Design and Analysis**	**15d**	**02/14/2017**	**03/07/2017**		
12		Component Research: Availability and Cross-Compati	5d	02/14/2017	02/21/2017		Nate Fox,Harold
13		User Interface Design	5d	02/21/2017	02/28/2017	12	Harold Hall
14		Mechanical CAD Drawings	5d	02/21/2017	02/28/2017	12	Thomas Niemis
15		Electric Circuit Diagrams	5d	02/21/2017	02/28/2017	12	Nate Fox
16		Packaging design & fabrication	5d	02/14/2017	02/21/2017		
17		Design Analysis	5d	02/28/2017	03/07/2017	14,15	Thomas Niemis
18		⊟**Functional Prototype Manufacture**	**20d**	**03/07/2017**	**04/11/2017**	**11**	
19		Purchase and rapid-prototype all components	10d	03/07/2017	03/21/2017		Harold Hall[25%
20		Component Assembly	5d	03/21/2017	04/04/2017	19	Thomas Niemis
21		Microcontroller Programming and Debugging	10d	03/21/2017	04/11/2017	19	Nate Fox,Harold
22		⊟**Functional Prototype Alpha Testing**	**5d?**	**04/11/2017**	**04/18/2017**	**18**	
23		Shop-test all components	1d?	04/11/2017	04/12/2017		
24		Test-ride bike and verify prototype operates as desired	5d?	04/11/2017	04/18/2017		Thomas Niemis
25		⊟**Functional Prototype Beta Testing**	**71d?**	**03/07/2017**	**06/21/2017**		
26		Find beta testers and schedule product testing with th	15d	03/07/2017	04/04/2017	11	Nate Fox
27		Test-ride by beta testers and collect performance data	10d	05/16/2017	05/30/2017	18	Harold Hall,Thor
28		Performance analysis	4d?	06/15/2017	06/21/2017	27	Nate Fox,Harold

Figure G1: Gantt Chart Hierarchal Table

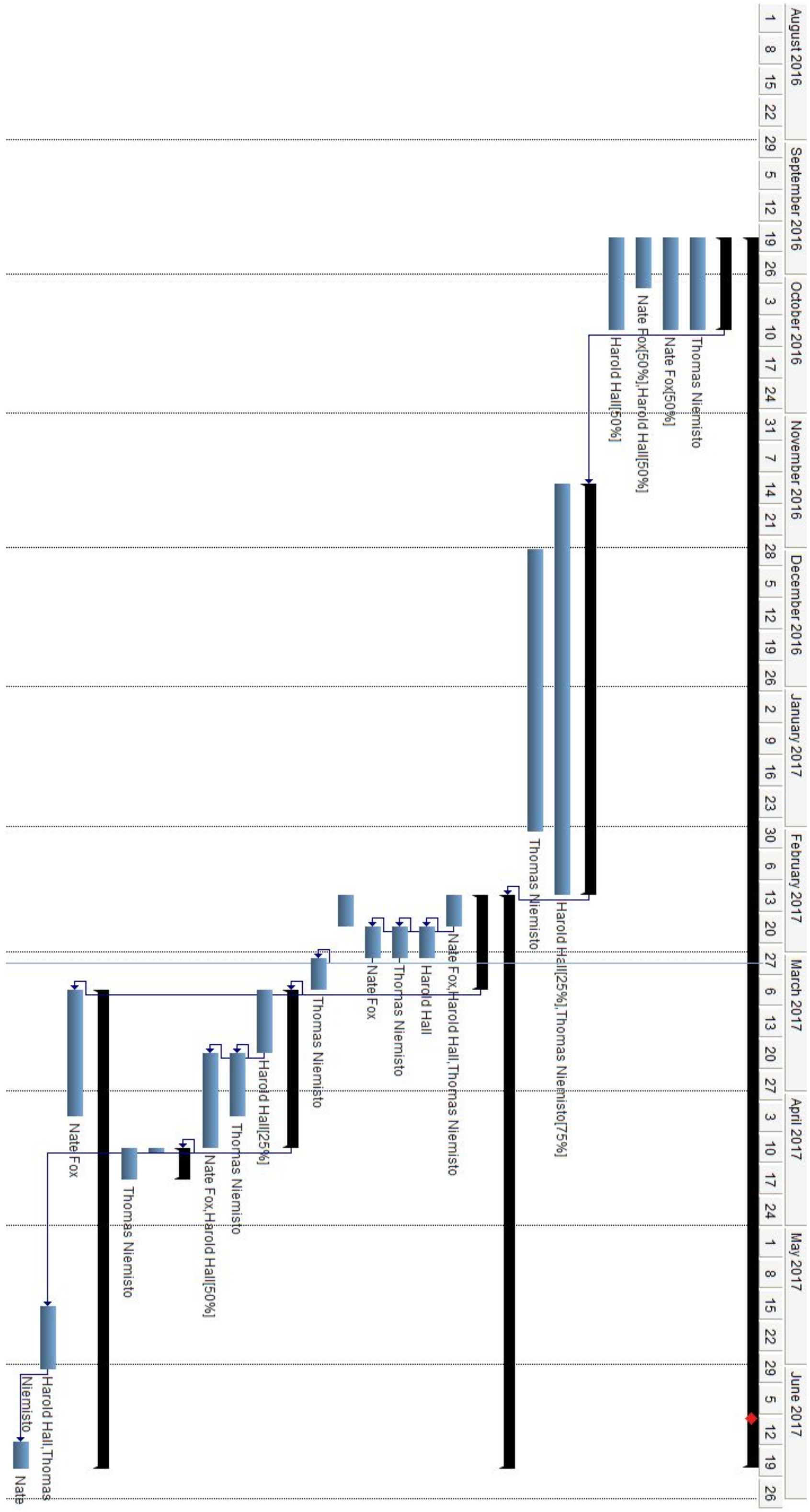


Figure G2: Gantt Chart

Appendix H: Hazard Identification Checklist

Y	N	
☒	☐	Will any part of the design create hazardous revolving, reciprocating, running, shearing, punching, pressing, squeezing, drawing, cutting, rolling, mixing or similar action, including pinch points and shear points?
☐	☒	Can any part of the design undergo high accelerations/decelerations?
☐	☒	Will the system have any large moving masses or large forces?
☐	☒	Will the system produce a projectile?
☐	☒	Would it be possible for the system to fall under gravity creating injury?
☐	☒	Will a user be exposed to overhanging weights as part of the design?
☐	☒	Will the system have any sharp edges?
☐	☒	Will all the electrical systems properly grounded?
☐	☒	Will there be any large batteries or electrical voltage in the system above 40 V either AC or DC?
☒	☐	Will there be any stored energy in the system such as batteries, flywheels, hanging weights or pressurized fluids?
☐	☒	Will there be any explosive or flammable liquids, gases, dust fuel part of the system?
☐	☒	Will the user of the design be required to exert any abnormal effort or physical posture during the use of the design?
☒	☐	Will there be any materials known to be hazardous to humans involved in either the design or the manufacturing of the design?
☐	☒	Can the system generate high levels of noise?
☒	☐	Will the device/system be exposed to extreme environmental conditions such as fog, humidity, cold, high temperatures ,etc…?
☒	☐	Will the system easier to use safely than unsafely?
☐	☒	Will there be any other potential hazards not listed above?