

MTB DAQ

Senior Project Report

Prepared for: Dr. Joseph Mello, Cal Poly Mechanical Engineering
Department

by

Theo Philliber
Max Ringrose
Ronan Shaffer
Dylan Ruiz

Statement of Disclaimer

Since this project is a result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is done at the risk of the user. These risks may include catastrophic failure of the device or infringement of patent or copyright laws. California Polytechnic State University at San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.

Abstract

Mountain bike suspensions systems are set up and tuned based only on qualitative, 'feeling'-based metrics. The MTB DAQ, sponsored by Dr. Joseph Mello with the Cal Poly Mechanical Engineering Department, is a data acquisition system we developed to improve suspension tuning with a data-driven approach. The system mounts to a bicycle frame and has sensors that measure vibration and travel speed to characterize the bike's motion. While similar systems exist, they cost in the thousands of dollars range and are very specific to individual bike geometry. Our system was designed to be more affordable to the average rider as well as generalized to many bikes and easy to set up. From our preliminary field testing, we found promising results from data collected with our system at different suspension setups. Future work can be done to collect more data and improve the processing algorithms, allowing a rider to simply plug in data from a ride and receive a recommended change to their tuning setup.

Introduction

Mountain bike suspension systems have several adjustable parameters: rebound, compression, and pressure. These all can be ‘tuned’ to alter the performance of the suspension, such as how quickly it will compress in response to force or return after compression. These parameters are set by the riders based on feeling, but many riders might not know what each setting does or what level to set them at. Our device is meant to improve the setup process by providing riders with data from their individual rides, which can be used to in a quantitative way to optimize their suspension systems.

The device is based on work from a former master’s thesis project by Cal Poly mechanical engineering alum Steven Waal. He created the original data acquisition unit to measure and model suspension systems on bikes. We started out with the intention to use his product to develop code and algorithms which generate tuning recommendations. However, as our project progressed, we ran into many unexpected issues preventing us from testing as planned. This resulted in our project’s scope changing from a focus on algorithm development to redesigning and improving the data acquisition system itself, adding additional sensors and functionality while identifying and fixing bugs.

This report documents the research, conceptualization, manufacturing, and testing of this project. It also details further work which needs to be done to make this a viable product. The report is composed of four main sections, based on our interim reports created for our sponsor. The Scope of Work describes the background research of our project, our problem definition, and initial planning. The Preliminary Design Review details the conceptualization and prototyping phase of the design process. Critical Design Review section shows our system design before manufacturing, as well as our plans to manufacture and test the project. Finally, the Final Design Review section shows our most updated version of the design, manufacturing results, test results, and further discussion of our project.

Table of Contents

PART 1 – SCOPE OF WORK

PART 2 – PRELIMINARY DESIGN REVIEW

PART 3 – CRITICAL DESIGN REVIEW

PART 4 – FINAL DESIGN REVIEW

MTB Suspension Tuning DAQ

Scope of Work (SOW)

October 15, 2021

Team

MTB – F11

By:

Dylan Ruiz

dyruiz@calpoly.edu

Ronan Shaffer

rmshaffe@calpoly.edu

John Ringrose

jringros@calpoly.edu

Theo Philliber

phillibe@calpoly.edu

Sponsor:

Professor Joseph Mello

Mechanical Engineering Department

California Polytechnic State University

San Luis Obispo

Abstract

This document outlines the preliminary work necessary to continue the development of a mountain bike suspension tuning tool. The development of this existing device will be performed by a team of mechanical engineers who are attending California Polytechnic State University, San Luis Obispo. This team will be working under the guidance of Dr. Joseph Mello. The work on this device will consist of refining the design, building a new unit, and testing on California's Cuesta Pass. The goal is to refine the device to a more user-friendly product that generates performance metrics leading to suspension recommendations for the user. In this document, one can find background research, objectives for the product, and a plan to make further progress on the design. The key results of this document are the state-of-the-art investigations section, engineering specifications table, and next steps section.

Table of Contents

1. Introduction.....	1
2. Background.....	1
2.1. Steven Waals’ Thesis Project.....	1
2.2 Interview with Sponsor	2
2.3. Stakeholder/Needs Research.....	3
2.4 Technical Information.....	3
2.4.1 Characterizing Vibration	4
2.5 Existing Designs and Patent Research	4
2.5.1 BYB Telemetry.....	6
3. Project Scope	7
3.1 Boundary Diagram	7
3.2 Stakeholder’s Needs & Wants.....	7
3.3 Functional Decomposition	9
4. Objectives	9
4.1 Problem Statement	9
4.2 Quality Function Deployment (QFD)	10
4.3 Engineering Specifications.....	10
5. Project Management	12
5.1 Design Process	12
5.2 Timeline	12
5.3 Next Steps	13
6. Conclusion	13
References.....	14

1. Introduction

The suspension on a mountain bike (MTB) is crucial to its performance. It allows the rider to safely land jumps and go over small obstacles. Most modern mountain bikes have two shock-absorbing components: the ‘fork’ above the front tire, and the rear ‘shock’ below the seat. Though they look different, both have a spring and a damper component. The spring provides a resistive force in response to an applied load, and the damper helps to smooth out the forces experienced by the rider. The stiffness of the spring and damper is variable on most bikes, allowing the user to increase or decrease them to their liking. The purpose of this project is to enable the user to use a data-driven approach to better tune mountain bike suspensions.

The foundation of this project is based on a master’s thesis project, which will be described more thoroughly in Section 2.1. It is sponsored by Dr. Joseph Mello, a mechanical engineering professor at Cal Poly SLO. The team consists of four mechanical engineering students completing their bachelor’s degrees: Dylan Ruiz, Theo Philliber, and Ronan Shaffer, concentrating in mechatronics, and John Ringrose, concentrating in general Mechanical engineering.

This report is a Scope of Work with which we seek Dr. Mello’s approval in sponsoring this project. This report consists of our background research including: stakeholders’ preferences, existing products, and technical documents; the project scope including: desired functions and our planned deliverables; our objectives including: problem definition, design specifications, and quality function deployment; and our project management plan including: our planned design process and key milestones.

2. Background

This section of the paper discusses background and user Interviews that was performed by the team.

2.1. Steven Waals’ Thesis Project

The foundation of this project is a master’s thesis project titled ‘A Quantitative Approach for Tuning a Mountain Bike Suspension’, completed by Steven Waal in 2020 [1]. His goal was to be able to adjust the suspension settings on a mountain bike in an analytical, data-driven way.

The first major task that Waal accomplished was to create a data acquisition system or a DAQ. The end product is a pair of accelerometers, located on the front and rear wheels. These sensors were connected to a central unit, which stored the data during a ride. Afterward, the data could be analyzed with a computer algorithm to find suggestions for tuning parameter adjustments.

Next, Waal created a mathematical model of a mountain bike. Using MATLAB, he developed a model with 2 degrees of freedom to represent the front and rear suspension. The model could take in an obstacle or trail feature and simulate the positions and forces acting on the bike tires.

With the model developed, Waal could then move on to optimizing the tuning parameters. He made a 'genetic algorithm', which takes in accelerometer data and the suspension settings and iteratively optimizes them in an evolutionary way. His optimizing function aimed to minimize the peak readings from the accelerometer data and return a score for each set of inputs.

With all these components, Waal had successfully found a way to analyze the suspension settings on a mountain bike. However, there are still lots of areas in which his project can be improved. Since he only considered the accelerations in his optimizing function, the suspension was tuned to be as soft as possible without bottoming out. It does not consider other factors, such as speed or handling. This could be resolved by adding a sensor on the bike to track velocity and another sensor to track rotation of the bike. Another issue is that the sensors were connected to the main computing unit with long wires, which risk coming loose and getting caught on the rider or the tires as they spin. The overall goal of our project is to continue Waal's work and to improve it into a more functional and polished product.

2.2 Interview with Sponsor

The sponsor of this project, Dr. Joseph Mello, has provided many insights and suggestions for what we should do differently following Waal's thesis work. One thing was to move to a more experimental and empirical approach, as in moving away from MATLAB modeling and towards more analysis of real data. While the model created an optimized algorithm, it was also limited in what types of data could be analyzed.

Another change Mello wants from us is to reevaluate the metrics that we are measuring and optimizing with. Instead of an accelerometer on each shock absorber, we could have one above and below the suspension, allowing us to isolate the effect of the suspension. There are various locations on the bike we will record data and that is a large portion of this project: record data and see if we can find correlations to accurately provide suspension tune recommendations. We could also measure different forces, such as using a GPS sensor to accurately find the bike's speed and position. Mello let us know using potentiometers to track position is a much more accurate way of tracking position. After the first interview with Dr. Mello, it is apparent that we need to make the current device a system ready for the end user. This means highly designed sensor housings, universal mounting, and a foolproof user interface. Where this project will differ from Steven Waals' thesis is that we will be focusing much more of the data collection and creating new metrics to quantify suspension quality.

2.3. Stakeholder/Needs Research

Multiple versions of a questionnaire were created and distributed to gather input from as wide a variety of mountain bikers as possible. Paper versions were handed to employees and customers at bike retailers locally, as well as to students in the Cycling Club on campus. A revised digital copy was then created and distributed online through social media platforms and relevant forums. These surveys asked people about their level of experience and what aspects of riding and suspension were important to them. While responses are still being collected, here is what we learned so far:

- 90% of respondents identified being intermediate or advanced mountain bikers
- 75% of respondents “knew just enough about suspension settings to ride comfortably”
- 60% of respondents only adjust their suspension settings a few times a year
- 70% of respondents are moderate to very focused on improving their speed downhill
- 90% of respondents considered comfort while riding a moderate to high priority
- 2/3 of respondents are interested in this device, 1/3 said “maybe”
- 80% of respondents would spend between \$100 and \$400 on a working device
- 70% of respondents want data for a trail segment of a duration between 2.5-10 minutes
- The most common response for weight requirements desired a device less than 1 lb.
- All respondents identified waterproofing the device as being important
- All respondents identified visual aesthetics as moderately to very important

The number of responses thus far have totaled up to 150 users and we will continue to collect responses from these questionnaires. The data collected from this will reinforce the list of needs and wants of customers as well as engineering specifications.

2.4 Technical Information

Research on technical literature included journal papers documenting the development of simulated dynamic models of a relevant physical system, published research on accelerations and impact forces found during mountain bike testing, data collection configurations, data processing methods, and figures of merit used to evaluate performance.

For our purposes, a 2 Degree-of-freedom (DOF) model was found to be sufficiently precise in evaluating and optimizing suspension parameters [2]. The accelerations we can expect to measure in the system are on the order of 50 m/s^2 [3]. One paper testing a suspension system on a Hummer with accelerometers mounted on each wheel, cross members, at the center of gravity of the vehicle, and driver’s location, recorded similar accelerations [4].

As identified in Waals’ thesis, accelerometer drift becomes an issue with this device after a duration of 15 seconds. Data processing methods such as Kalman filters have been researched and identified as possible solutions to eliminating noise and acceleration drift [5]. After reading a formula SAE article, accelerometer attachment methods could be a source of this “drift” [6]. Other forums point to using high-pass and low-pass frequency filters to remove noise as well as using multiple accelerometers to calculate an average value [7].

The accelerometers used in Waal's thesis project are capacitive micro-electrical-mechanical systems (MEMS). As the chips undergo acceleration, the components inside move in response. This results in a measurable change in capacitance, which is the output of the sensor [8]. Due to this use of capacitance, MEMS accelerometers are susceptible to anything which could affect capacitance. Temperature is one variable affects the capacitance of materials [9]. There are many articles which describe different approaches to reduce this error [10]. It is likely that we can ignore this temperature-based error, as well as other sources of minor error, because our project will not require a very high degree of accuracy.

After meeting with Dr. Mello and conducting research on technical literature, we decided our next steps should be to continue researching suspension tuning metrics and sensor configurations to assess suspension performance and bike behavior, as well as developing methods for simpler data processing.

2.4.1 Characterizing Vibration

One of the major challenges of this project is with quantifying the data we collect. More specifically, we need a numerical way to determine what suspension settings are 'good' or 'bad' in terms of comfort, which is a subjective and qualitative feeling. With this metric, we can then go about finding the relationships between parameter settings and comfort.

There are many research reports on this topic, especially from automotive companies. The *Handbook of Human Vibration* [13] details one way to calculate a metric that accounts for both the magnitude and frequency of the vibration. This represents the 'psychophysical magnitude' of the vibration, its perceived harshness. We could use a metric such as this in our algorithm's minimizing function, to determine which datasets (and related suspension settings) are optimal. Similar methods for quantifying vibrational discomfort are described in [14] and [15]. The exact metrics that we use to quantify vibrational discomfort will be determined later in this project if approved.

2.5 Existing Designs and Patent Research

The background research included existing products that function similarly to what the problem statement describes. These products were rated on a scale of 1(poor) to 5(great) in terms of their performance for each of the customer's needs and wants. This type of product has only one type of interpretation on the market. The rest of the products that were compared functioned similarly but were not specific to MTBs. The most competitive product that related to the problem statement was the Quarq Shockwiz [11]. The Shockwiz is a device that can plug into the fork or rear shock and analyze pressure data to suggest tuning changes. It connects to a smartphone app via Bluetooth to show the results of the data analysis. The Shockwiz only works with shocks that have a positive air chamber. This limits the use of the Shockwiz for potential users. The total list of existing designs can be found below in Table 1.

Table 1: State of the Art Product Table

<i>Product Name</i>	<i>Image of Product</i>	<i>Product Description</i>
Shockwiz	 <p>[11]</p>	Plugs into the fork or rear shock and analyses pressure data to suggest set-up options. There are four tuning styles, Efficient, Balanced, Playful, and Aggressive. Only works with forks and shocks that have a positive air chamber.
Flight Attendant	 <p>[16]</p>	System integrated on selected bikes. Collects data on the forks, shocks, and crankset. Automatically switches suspension between three options, Open, Pedal, and Lock.
EI System	 <p>[17]</p>	Integrated with only three bikes. Takes data via accelerometers and cadence sensors to determine a setting to be applied to rear shock. Automatically switches between Locked Out, Middle, Open and Fully Open.
Motion Instruments Downhill Pro System	 <p>[18]</p>	DAQ System used for collecting and analyzing data for mountain bikes. Used for downhill segments only.
Motool Slacker	 <p>[19]</p>	Suspension Tuner used mainly for motorcycles. Used to measure sag using a retractable nylon cord.

A patent search was done to see what kinds of methods and devices had been patented. Since the solution to the problem statement is very specific, no patents are accomplished in providing a full solution. However, there were multiple methods of setting up a DAQ system on wheeled vehicles, which can be used when designing the installation methods. Patents that solved the problem of mounting the sensors and DAQ were also investigated. After reading many patents that relate to accelerometer mounting, it is established how important the orientation of mounting sensors is. An example “Ensures that the axial orientation of the vibration data collected by the accelerometer properly corresponds to the intended axis of measurement of the machinery component and thus allows for consistent, repeatable installations of the accelerometer onto the machinery component without introducing alignment errors which can generate erroneous vibration data with respect to one or more axes of measurement” [12]. Many different types of mounts could be used to solve this problem, however the most relevant were ones consisting of a clamp to attach to the bike frame or strong adhesives. We will continue our investigation as the project continues to evolve.

2.5.1 BYB Telemetry

There exists one product on the market which is very similar to our goal: the *BYB Telemetry* [13]. Like ours, this is a data acquisition system that can be mounted to nearly any bike and will collect and analyze data for the rider. It gives suggestions on tuning parameters based on data collected from accelerometers, gyroscopes, wheel speed sensors, fork and shock sensors, and a GPS sensor. It does not make any changes to the suspension itself, but passively suggests improvements based on quantitative methods.

Our product will have the same goal as the BYB Telemetry system, to improve suspension tuning through data-driven analysis. However, we aim to make a simpler and more budget-friendly product that would be accessible to an average or mid-level rider. The biggest flaw of the BYB Telemetry is its shocking \$1600 price tag, making it useful only for professional MTB teams or very wealthy hobbyists. We are aiming for the low hundreds range and have a specific target of \$150 for the system. This will mean a more streamlined and stripped-down system with fewer sensors and less thorough data than the Telemetry. We will need to use different metrics for quantifying suspension performance and may need a different way to mount the system. Considering that MTB suspensions have a limited number of settings, having more ride data will not necessarily improve the ride experience.

3. Project Scope

This subsection of the paper defines all of the necessary specifications of the product to make a well-designed MTB Daq.

3.1 Boundary Diagram

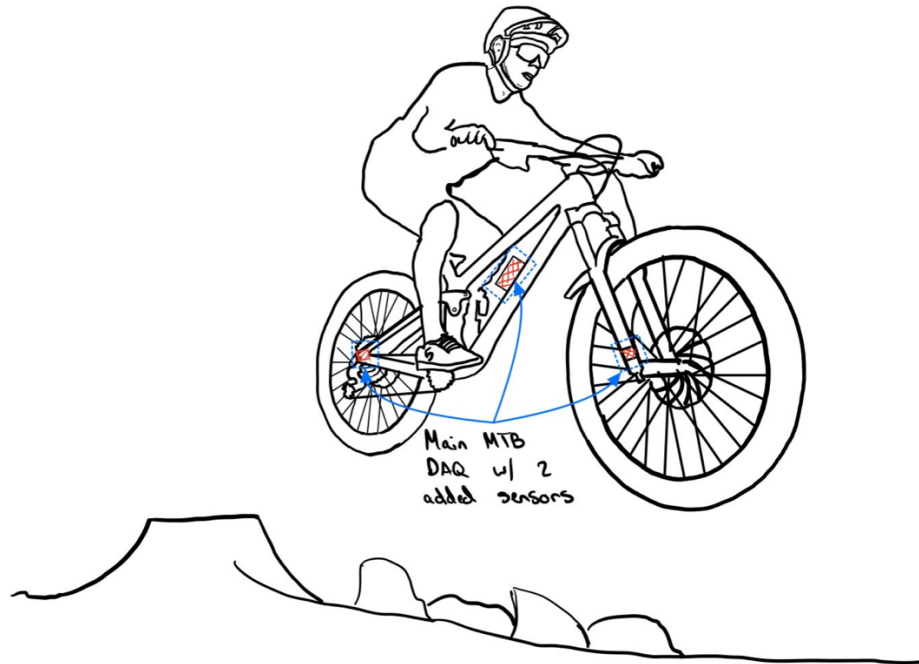


Figure 1: *Boundary Diagram for MTB Suspension DAQ*

The boundary sketch seen in Figure 1 was created to outline the functions and placements of DAQ while riding the MTB. The system will only be comprised of the main MTB DAQ along with 2 other sensors secured to the MTB in specific positions. The external references that will influence the product’s function are the terrain, the MTB, and the rider. The external terrain will directly affect the way we set up the DAQ and its sensors to acquire viable data. The rider will affect the inputs that will be used to calibrate the DAQ to provide accurate data. Finally, the MTB will affect the way the mounts are designed for both the sensors and DAQ as well as their placement.

3.2 Stakeholder’s Needs & Wants

To help define a problem, our team created a table of wants and needs which were influenced by our stakeholder research on potential customers’ preferences. This can be seen in Table 2 below.

Table 2. Summary of customer needs and wants.

Needs	Wants
Resistant to water and dust	Easy to setup
Sensors have a small profile (doesn't obstruct wheel movement)	Less than 1lb
Battery lasts long enough for the entire ride(s)	Simple controls/interface
Cables secured (don't get snagged on bike or rider)	Enough data storage for the entire ride(s)
Can withstand vibration without breaking or giving incorrect data	Battery life of 10 "rides"
Can mount to bike frame (universality)	Recommended Compression/rebound direction
Simple Installation	Low Cost
Low profile	Sag Recommendation

This table of needs and wants was generated originally by speaking with Dr. Mello and taking note of the specifications he stated were important for the device. To develop an unbiased and larger list of needs and wants we distributed multiple rounds of questionnaires to all levels of mountain bikers. This list of needs/wants will continue to evolve and refine, ultimately steering the design of this product in the correct direction.

3.3 Functional Decomposition

To describe all functions required to create a well-rounded Data Acquisition system, we need to break down the overall function of the device into smaller functions. This breakdown of functions can be seen in Figure 2 below.

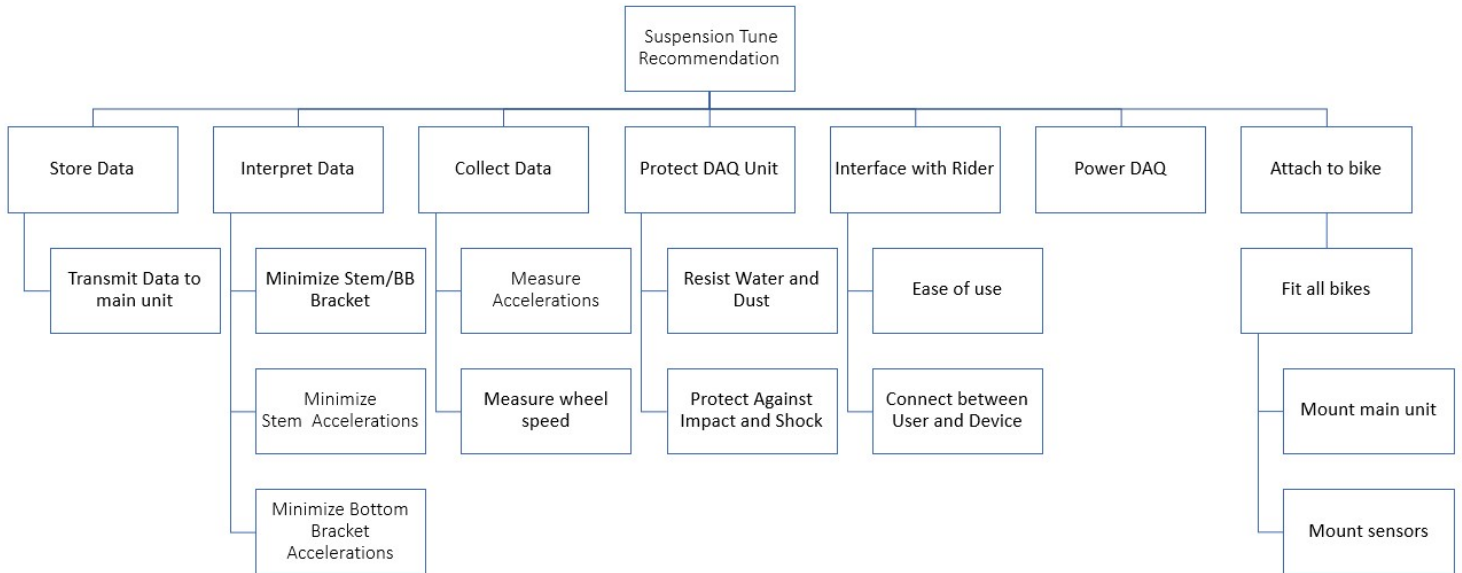


Figure 2. Functional Decomposition Diagram

At a high level, this device must provide suspension tune recommendations (as seen, the general function). Many sub functions are leading up to this high-level function and all must be executed to solve the problem statement. The subfunctions in this diagram were influenced by our background research, the state of the current product, Dr. Mello, and potential user questionnaires. All functions in this diagram use high level language to allow for solutions to be viable. As this project continues, new functions will certainly be added while original tasks may no longer be important.

4. Objectives

This subsection of the paper defines the goals of the DAQ and discuss the strategies leading up to desired specifications of the product.

4.1 Problem Statement

The problem statement for this project is as follows: Mountain bikers of all skill levels need a straightforward way to quantitatively tune their suspensions. Current tuning metrics are done purely based on rider feel, but this can be an issue for inexperienced riders or professional riders who desire an optimized suspension setup using a data-driven approach. This solution will provide

feedback on improving suspension settings based on data taken through riding a mountain bike downhill. The mechanism will be easy to install and be user-friendly, lightweight, and accurate in providing suggestions. The needs and wants of the customers include the mechanism to be durable, lightweight, and low cost among others. All the needs and wants can be found in Table 1.

4.2 Quality Function Deployment (QFD)

The Quality Function Deployment process determined the product specifications required for the DAQ system. Using customer surveys and personal interviews, the needs and wants of the specific customers were taken into consideration and evaluated. Each need and want was rated on a 1-10 scale of importance for each type of customer, 10 being the most important. Similarly, competitor products were rated on a 1-5 scale on how much they satisfy customer needs and wants, with 5 being the most satisfactory. Finally, competitor products were also rated on a 1-5 scale on how well they meet the product specifications. A House of Quality was developed with these processes in Appendix A.

From this process, it was determined that the product Shockwiz satisfied the customers' wants and needs the best. The most important needs/wants for all the customers are that the product is lightweight, durable, accurate, and resistant to water and dust. These needs and wants then lead to a more value based list of engineering specifications. Once the prototype is built these specifications will have dedicated tests to see if their pass criteria are met.

4.3 Engineering Specifications

Table 3 is a brief description of each specification as well as their proposed test to determine whether the set target has been met.

Table 3. Specification Table listing all the parameters to test to meet customer requirements.

Spec #	Parameter	Target	Tolerance	Risk*	Compliance**
1	Size	5"x3"x1" 1.5"x1.5"x1.5"	Max	M	A, I
2	Weight	500g	Max	M	A, I
3	Cost	\$150	Max	H	A
4	Battery Life	1 Hr	Min	L	A, T
5	Ingress Protection	IP54	Min	M	I, T
6	Foolproof	100% pass by user testing	Min	M	S, T
7	Maximum Recording Time	8 gb	Min	L	A, T
8	Mounting Universality	All bikes	Min	L	I, T
9	Aesthetically Pleasing	80% approval by user testing	Min	L	S
10	Suspension Tune recommendation	5% faster	Min	H	A, T

*Risk of meeting specification: (H) High, (M) Medium, (L) Low

** Test Methods: (A) Analysis, (I) Inspection, (S) Customer Survey, (T) Test

The highest risk of the specifications above is the ultimate task of the DAQ; provide suspension tune recommendations. Given that the current device only collects data, the most difficult task will be relating the data phenomenon to a better specific suspension tune. The current device falls short of meeting most requirements stated above, but with a refined device (Proto 2) specifications will be met.

1. Size

After creating the prototype, a simple measurement of the external geometry will dictate pass/fail criteria for the prototype. The larger X*Y*Z dimension refers to the main DAQ hub and the smaller dimension refers to the sensors.

2. Weight

Using a digital scale, weighing the complete system after a prototype has been built.

3. Cost

This cost is relevant to the total cost required by the team to build the device. This excludes purchasing mistakes and analyzes what the product would cost to make at mass production.

4. Battery Life

The power consumption of all the electrical components can be added up to estimate the total system power usage. This can be multiplied by the battery's listed capacity to analytically find the battery life. Once the prototype is complete, we can also time how long the battery lasts to validate the calculated result.

5. Ingress Protection

The product shall be protected from limited dust ingress and protected from water spray in all directions. The testing to determine these traits will follow the industry's standards.

6. Foolproof

The prototype should be easy to use, with minimal room for user error. We will need to conduct a user survey, allowing potential customers to set up and operate the prototype. If they do not struggle with this, then the test is passed.

7. Data Storage Capacity

After sampling data for the stated maximum allowable time (10hr) there is enough storage on the device to hold all recorded data. This allows the user to track past rides.

8. Mounting Universality

The product should be able to accommodate a variety of bike shapes and sizes. We can test this by simply attaching the device to a variety of bikes and see if it all fits.

9. *Aesthetically Pleasing*

The product should be aesthetically pleasing to customers'. To test this, we can simply survey the customers opinions on the look of the device.

10. *Suspension Tune Recommendation*

The end goal of this product is to recommend changes the user should make to their bike suspension settings. This is a pass/fail specification, as the device will either accomplish this or it won't. We will test if it does with experimental data to see if it passes under use-case conditions.

5. Project Management

This subsection of the paper discusses the planned next steps to designing the updated DAQ.

5.1 Design Process

The first part of this project was all about gaining understanding. We read through the master's thesis, got familiar with the DAQ and code given to us, and did background research. During this time, we have also conducted surveys with potential consumers to get an idea of the needs and wants of end-users. These activities have led us to the construction of this document. The next goal will be to determine what metrics we will use in our optimization algorithm, as well as what sensors we will want to take data. After this, we need to create the necessary hardware for the DAQ, including the universal mounts and potentially any circuitry needed for new sensors. After that, we would start taking data and running it through the model, adjusted to find the new metrics. We would iterate through different settings and finally, observe the results.

5.2 Timeline

We will use a Gantt chart to plan and track timelines of key tasks and goals. This will help us stay on track and organize which team member will do what task. The full chart is listed in Appendix B, and the major deliverables are tabulated below in Table 2.

Table 4. Major deliverables with their estimated delivery dates

Major Deliverables	Completion Date
Function Brainstorming	10/19/21
Function Ideation Models	10/26/21
Concept Models	10/28/21
Controlled Convergence	11/1/21
Proto 1 Decision	11/3/21
CAD of Proto 1 Mount	11/8/21

Proto 1 Manufacturing	11/10/21
Assembly and Mounting Proto 1	11/11/21
Proto 1 testing	11/13/21
PDR Results	11/18/21

5.3 Next Steps

After gaining approval of this document, we propose to continue our research leading up to the Preliminary Design Review. We will also begin our brainstorming and concept ideation process where we utilize our background research to come up with concepts to improve the current design. After iterations of ideation are complete, we will share our results with our sponsor to gain feedback. We are aiming to solve the key issues of extended data collection, more dependable data, and ultimately, an empirical value derived from data to tune a bicycle's suspension. To solve these key issues this means our team must quickly create working mounts and housings to begin collecting data. The collection of data will be a large part of figuring out how to create a metric for tuning the suspension and will likely take many runs to figure out proper sensor positions and sensor outputs.

6. Conclusion

This document proposes a continuation of finding a quantitative method of tuning a mountain bike suspension. It compiles the objectives, background research, process, and estimated timeline of the project. The next step would be to start evaluating different metrics and sensors that could be used to better characterize a mountain bike suspension, as this would govern the direction of the project. This means getting the unit on the bike and begin data collection as soon as possible. We first need to design reliable housings and mounts to achieve dependable data. If approved, the next key deliverable will be a preliminary design review (PDR), delivered on November 18, 2021.

References

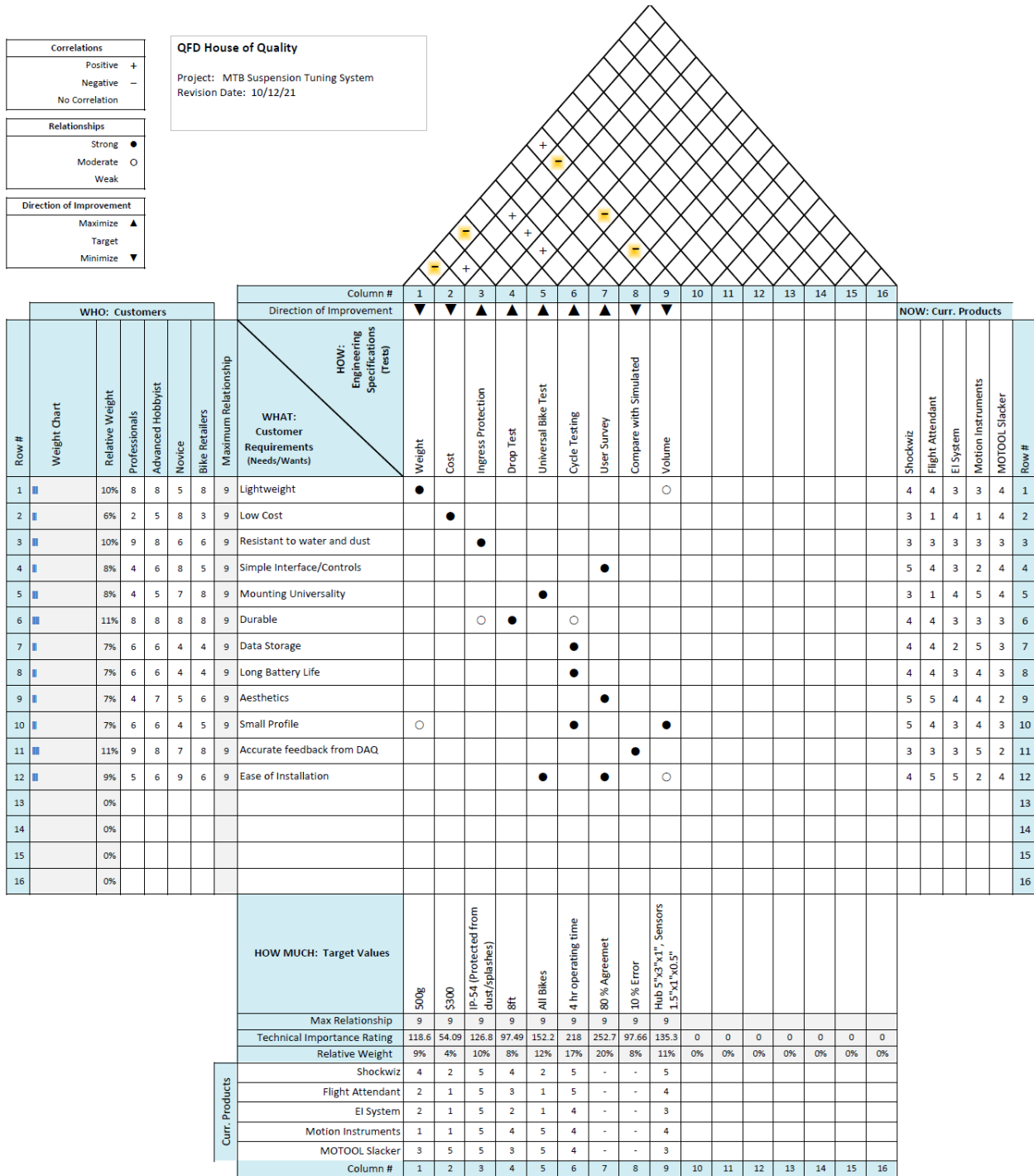
S. R. Waal, "A Quantitative Approach for Tuning a Mountain Bike Suspension", M.S. thesis, California Polytechnic Univ., San Luis Obispo, CA, 2020.

- [1] Yan Bu, Zhongxia Xiang and Tian Huang, "A multi-body model for the simulation of rider and mountain bike coupled system," 2008 7th World Congress on Intelligent Control and Automation, 2008, pp. 1669-1672, doi: 10.1109/WCICA.2008.4593169.
- [2] P. W. Macdermid, P. W. Fink, and S. R. Stannard, "Transference of 3D accelerations during Cross Country Mountain Biking," *Journal of Biomechanics*, vol. 47, no. 8, pp. 1829–1837, Jun. 2014.
- [3] A. C. Wray, F. B. Hoogterp, S. Garabedian, E. Anderfaas, and B. Hopkins, "Magneto-rheological fluid semi-active suspension performance testing," *The Nation's Laboratory for Advanced Automotive Testing*, 2003.
- [4] E. N. Chatzi and C. Fuggini, "Online correction of drift in structural identification using artificial white noise observations and an unscented Kalman filter," *Smart Structures and Systems*, vol. 16, no. 2, pp. 295–328, 2015.
- [5] Nye, Alan "RIT Formula SAE Data Acquisition System Preliminary Design Report" *Rochester Institute of Technology*
- [6] Lardy, Julien "How can I avoid data drifting when integrating acceleration signal?" *Research Gate*.
- [7] "Introduction to MEMS Accelerometers." *PCB Piezoelectronics*, <https://www.pcb.com/resources/technical-information/mems-accelerometers>.
- [8] Gao, JM., Zhang, KB., Chen, FB. et al. "Temperature characteristics and error compensation for quartz flexible accelerometer." *Int. J. Autom. Comput.*, vol. 12, pp. 540–550 (2015). <https://doi.org/10.1007/s11633-015-0899-5>.
- [9] Zhu, M., et al. "Temperature Drift Compensation for High-G MEMS Accelerometer Based on RBF NN Improved Method." *Applied Sciences*, 2019, vol. 9, no. 4, doi: 10.3390/app9040695.
- [10] "ShockWiz - Quarq." *SRAM*, <https://www.sram.com/en/quarq/series/shockwiz>.
- [11] Baker Hughes Oilfield Operations LLC, 2020. Triaxial Accelerometer mounting Adapter. 16867670.
- [12] "BYB Telemetry." *BYB Telemetry*, <https://www.bybtelemetry.com/telemetry>.
- [13] Griffin, MJ. "Handbook of Human Vibration.," *Institute of Sound and Vibration Research*, Southampton, U.K., 1996.
- [14] *Mechanical Vibration and Shock – Evaluation of Human Exposure to Whole-Body Vibration*. ISO Standard 2631-1:1997(E).
- [15] Torbic, D., El-Gindy, M., Elefteriadou, L. "Methodology for quantifying whole-body vibration experienced by bicyclists." *International Journal of Vehicle Design*, 2003, vol. 31, no. 4, pp. 452-480, doi: 10.1504/IJVD.2003.003357.
- [16] "Flight Attendant." *RockShox*, <https://www.sram.com/en/rockshox/collections/flight-attendant>.

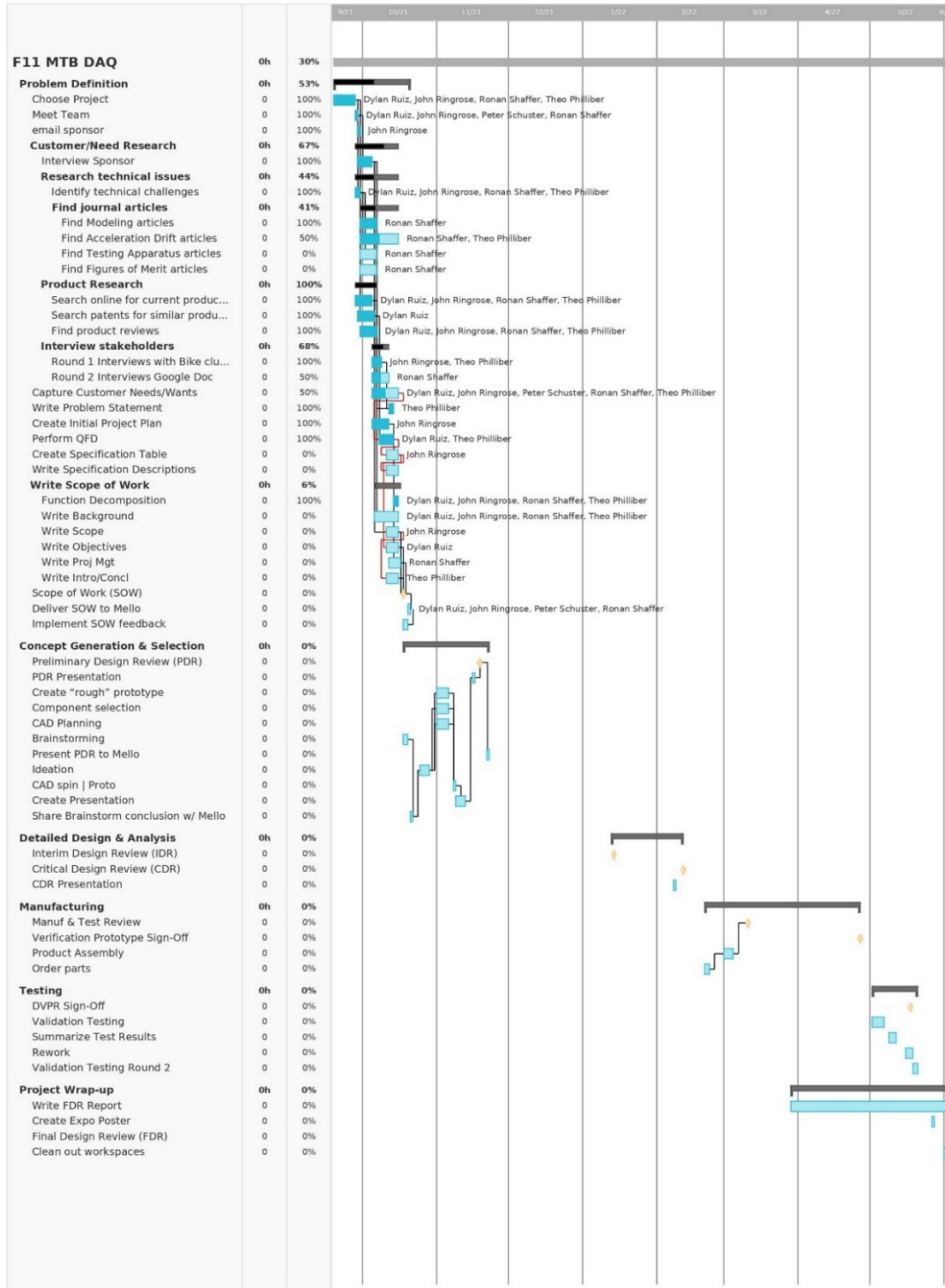
- [17] Cunningham, Richard. "First Look: Rockshox/Lapierre Show Electronic Shock at Morzine." *Pinkbike*, 27 June 2012, <https://www.pinkbike.com/news/ei-and-RockShox-Show-Electronic-Shock-2012.html>.
- [18] "Downhill Pro System." *Motion Instruments*, <https://motioninstruments.com/collections/downhill/products/downhill-pro>.
- [19] Motool Inc. "Slacker Digital Suspension Tuner - V4." *Motool Inc*, <https://motool.co/products/slacker-digital-suspension-tuner-v4>.

Appendices

A. QFD House of Quality



B. Gantt Chart



MTB Suspension Tuning DAQ Preliminary Design Review (PDR)

November 20, 2021

Team
MTB – F11

By:

Dylan Ruiz
dyruiz@calpoly.edu

Ronan Shaffer
rmshaffe@calpoly.edu

John Ringrose
jringros@calpoly.edu

Theo Philliber
phillibe@calpoly.edu

Sponsor:

Professor Joseph Mello
Mechanical Engineering Department
California Polytechnic State University
San Luis Obispo

Abstract

This Preliminary Design Review (PDR) details the ‘MTB Suspension Tuning DAQ’ senior project, performed by four mechanical engineering students at California Polytechnic State University, San Luis Obispo. The project, under the guidance of Dr. Joseph Mello with the mechanical engineering department, aims to quantify the suspension settings of mountain bikes (MTB) to improve the riding performance and reduce vibrational discomfort. A data acquisition system (DAQ) will collect data during a ride, which will be analyzed after the fact to suggest changes to the tuning parameters of the suspension. In this document, the initial concept development and design choices are described. Our team created an initial design for the sensor housings to be mounted at the front and rear axles. We also developed preliminary metrics to characterize the motion of the bike and rider and therefore analyze the effect of a particular suspension tuning configuration. This report also lays out future steps if the PDR is approved, including redesigning the printed circuit board (PCB) to incorporate more sensor inputs and a redesigned central DAQ unit housing to protect the PCB and resist water and debris ingress. This work will further the progress towards a mountain bike suspension tuning system that will use data to output meaningful information on adjusting suspension settings that will allow the user to ride faster and more comfortably.

Table of Contents

1. Introduction	1
2. Concept Development.....	2
Figure 1. Functional Tree of the MTB DAQ System. Bolded Functions were Further Considered in Pugh Matrices (See Section 2.1-2 and Appendix B).....	2
2.1 Concept Ideation	2
2.2 Pugh Matrices	3
2.3 Top Five System Level Concepts	3
2.4 Weighted Decision Matrix.....	5
3. Concept Design	6
3.1 CAD Design of the Concept Prototype.....	6
3.2 Manufacturing and Materials of Concept Prototype.....	9
4. Concept Justification	11
5. Future Work	13
5.1 Suspension Balance Metric.....	13
5.2 Rider Bike Metric	14
5.3 Handlebar/Fork Metric.....	15
6. Project Management	17
Concept Ideation	18
Design Prototype.....	18
Rework Optimization.....	18
7. Conclusions	19
References	19
Appendices.....	17
APPENDIX A: IDEATION CONCEPTS (BEST IDEAS)	
APPENDIX B: PUGH MATRICES	
APPENDIX C: WEIGHTED DECISION MATRIX	
APPENDIX D: GANTT CHART	
APPENDIX E: DESIGN HAZARD CHECKLIST	

1.Introduction

The goal of this report is to document the selected design direction of the next iteration of Dr. Mello's Mountain bike (MTB) data acquisition system (DAQ). The MTB DAQ will collect accelerations, angular velocities and speed that will be transferable to a computer through an SD card. Using this data, the user will be able to process the data based on metrics that our team defines and justifies. The next iteration of design will be supported by the findings in the first round of prototyping as well as the planned next steps to implement the new features of the system. Since the Scope of Work document was released, new features of the system were needed to succeed in developing the main goal of the system; that is to provide a user with tuning recommendations for their suspension settings based on metrics designed by the MTB DAQ team. Our team will develop our own metrics to characterize suspension behavior and evaluate a rider's performance resulting from their selected settings. Our first stage of research into these suspension metrics is outlined, as well as the team's upcoming goals for designing, building, and testing new system features. We outline the process used to formulate and refine our design concepts in the Concept Development section. We explain the design features and material aspects of our current prototype for the sensor housings in the Concept Design section. The Concept Justification section presents our reasoning for why the design will work as intended. The Future Work section describes a few figures of merit our team will test for their usefulness and relevance in characterizing motion of the bike and whether they correlate with improvements in speed and comfort. Finally, our project management section outlines the timeline of our upcoming project work including research, ideation, design reviews, testing, purchasing and manufacturing.

2. Concept Development

The concept development process consists of organized and sequential steps that allow the selection of five system-level concepts. Concept ideation methods were used to produce ideas, both reasonable and unfeasible. These ideas were then implemented into Pugh Matrices for a specific function. Finally, the results of the Pugh Matrices inspired the creation of five system-level ideas for the MTB mounting system. These system-level ideas were evaluated through a weighted decision matrix to select the design direction of the MTB mounting system. To provide context for the main functions of the system, a functional tree is composed below in Figure 1.

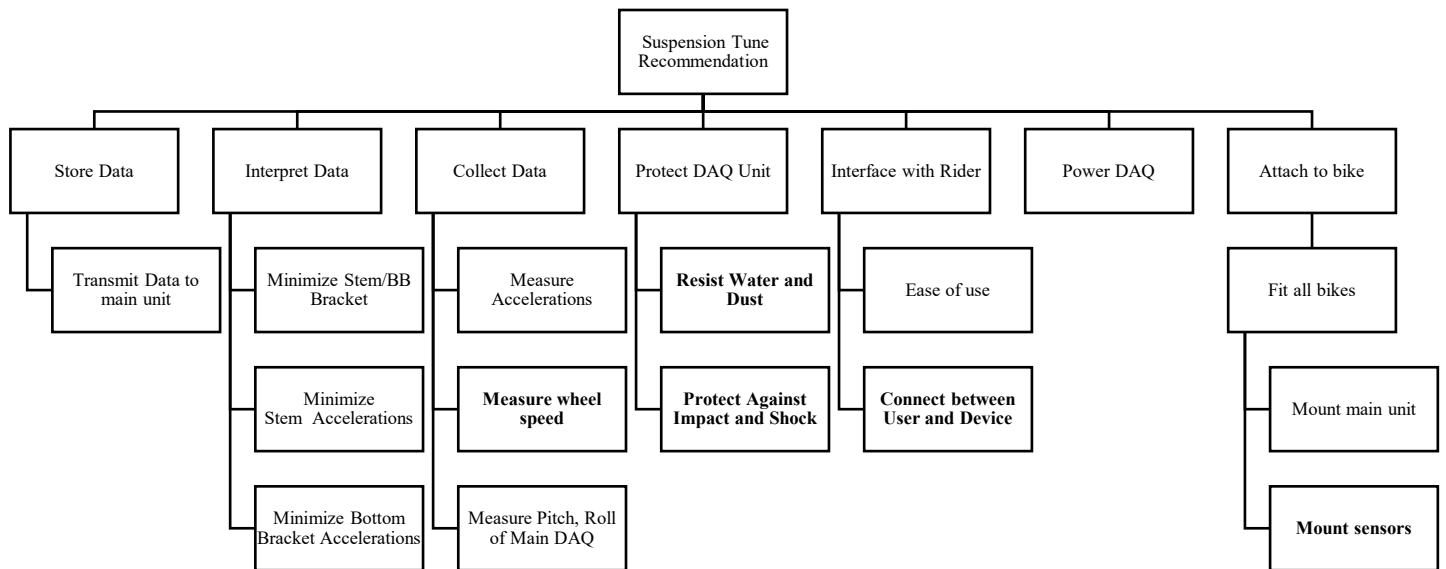


Figure 1. Functional Tree of the MTB DAQ System. Bolded Functions were Further Considered in Pugh Matrices (See Section 2.1-2 and Appendix B)

2.1 Concept Ideation

To develop multiple ideas for each function, a functional tree divided the system into subfunctions which the ideas were based on. Only functions relevant to the initial set-up of the MTB were referenced due to the priorities of preparing for preliminary testing. These functions include mounting to sensors to the bike, speed measurement, user interface, and sensor protection. The goal was to create 50 ideas per function through brainstorming, then select the best five ideas from each function to further evaluate in the Pugh Matrices. We generated ideas

using the “How Might We” templates. Some of the functions were more technical than others, such as ways to measure speed. The top ideas are listed in Appendix A.

Using these top five ideas for each function, ideation models were developed mainly from cardboard, glue, and popsicle sticks to help visualize a few of the ideas, two examples of which are shown in Appendix A. These concept models were focused on the mounting portion of the functions, which consisted of the mounting of the sensors to the MTB and the housing for the sensors.

2.2 Pugh Matrices

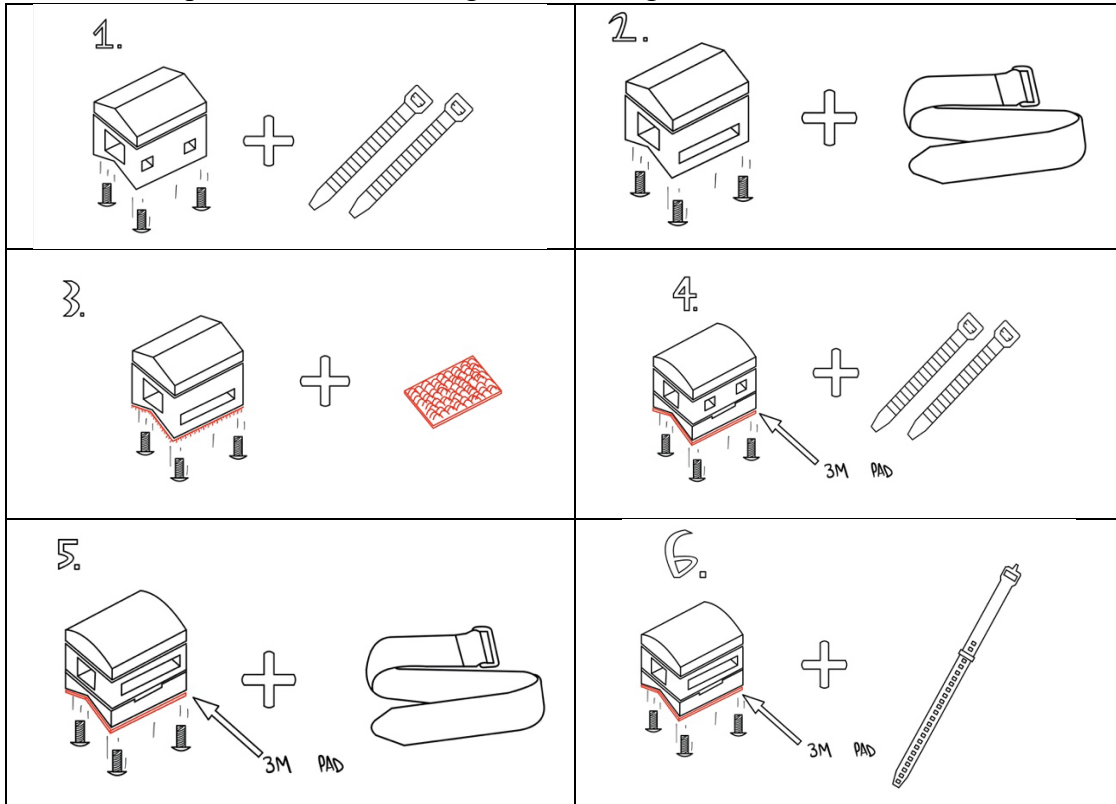
A Pugh matrix compares the different ideas for a certain function using a set of relevant criteria. One of the ideas is used as a “datum” to compare whether the other ideas are better or worse for a certain criterion. The values +, -, S are used to represent where the ideas compare to the datum for each criterion, with “S” being equal to the datum, “+” being better than the datum and “-“ being worse than the datum. The scores are summed up for each solution, giving a rough idea of which solution will be the best, although it does not consider the importance of each criterion. The final concept ideas were determined using Pugh Matrices.

Each of the four Pugh matrices in Appendix B were developed for one of the MTB DAQ system’s functions: interface with rider, attach to bike, collect data and protect DAQ unit. Because the mounting of sensors to the bike was the priority, only the ideas for mounting and housing the sensors were used to create the top five system level concepts. The other two functions are independent of the mounting and can incorporate virtually any idea. From the Pugh matrices related to mounting, it was found that permanent attachments to the bike, such as welding or adhesives, were unfavorable due to the effects it would have on the bike’s frame. Instead, Velcro and tie straps proved to be more efficient. Also, a tapered base was more favorable than a rounded base because it ensures at least two points of contact to a variety of bike frames.

2.3 Top Five System Level Concepts

Using the results from the two relevant Pugh matrices, six concept level ideas were created. The concept level ideas incorporated the combinations of the top two ideas from the sensor housing Pugh matrix and the top three ideas from the mounting to bike Pugh matrix. These six concept level ideas can be seen in Table 1.

Table 1. Six Concepts for Sensor Housings & Mounting Methods



Each of these concepts provided a unique way of housing and mounting the sensors to the bike. All the concepts use a tapered base that allows the housing to mount to multiple diameters within a range. Concept 1 utilizes zip ties, a cheap and accessible product, to mount the housing to the bike. The housing has a lid that can be opened to insert the accelerometer and utilizes screws to secure the accelerometer board to the housing. Concept 2 is the same design as Concept 1 except it uses a Velcro strap instead of zip ties, which is better for reusability. Concept 3 uses a Velcro pad to mount the housing to the bike instead of zip ties or a Velcro strap. Concept 4 consists of a lid with a hinge, zip ties and a rubber pad to both increase stability through friction and protect the bike from scratches. Concept 5 is like Concept 4, substituting the zip ties for a Velcro strap. Finally, Concept 6 is similar to both Concept 5 and Concept 4 but uses a OneUp strap to mount the housing to the bike. The OneUp strap is an off-the-shelf part made of a UV stabilized polyurethane strap.

2.4 Weighted Decision Matrix

The weighted decision matrix in Appendix C compared the six mounting concepts against each other using a set of criteria and weights developed through team consensus. Each design was rated 1-10 for each criterion based on how well it satisfied the criterion, with a rating of 10 indicated the design perfectly met the criterion to the highest standard. The rating of each criterion was also multiplied by each of the criterion's weight, creating a total score for each criterion. The total scores for each criterion were then summed up for each design to determine the best design based on our specifications.

Each team member filled out their own decision matrix based on their best judgement. This first step was done individually to minimize the impact of peer pressure among team members. The final decision matrix was created based on the average of all the team members matrices. The best concept, determined by the highest score in the decision matrix, was Concept 6 due to its high rankings in the most important criteria. These criteria include durability, vibrational dampening, universality, and security. The second and third place concepts were Concepts 5 and 4 respectively. These designs both included the rubber base which seemed to be a big factor for the most important criteria. Although Concept 6 was determined to be the design direction for the mounting system, this is mainly based on theory and is subject to change based on the preliminary testing results.

3. Concept Design

Our selected concept design consists primarily of two newly designed parts: the sensor housing and the mounting platform. The purpose of the housing is to secure the accelerometer and ethernet port to protect the components from the outdoor conditions being operated in. The purpose of the mounting platform is to attach the housing to the bike. Our design also includes the data acquisition (DAQ) unit and the accelerometer sensors, designed and built by Steven Waal, the master's student whose work formed the basis of our project.

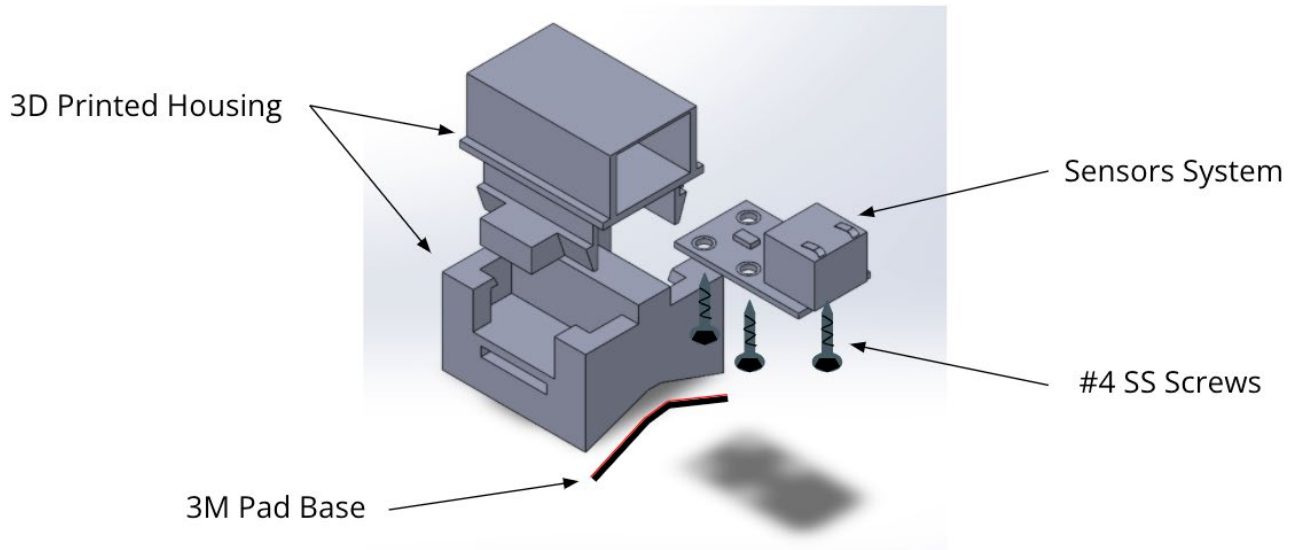


Figure 2. Assembly model of the sensor, housing, and platform fitting together

3.1 CAD Design of the Concept Prototype

The accelerometer sensor shown below in Figure 3 will measure acceleration data in XYZ directions; this data will be transmitted via ethernet cables to the main DAQ unit, where it will be stored on an SD card.

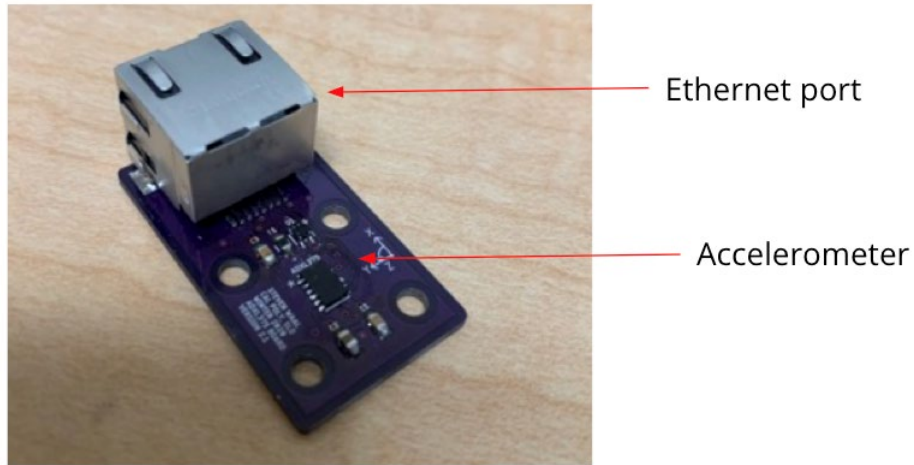


Figure 3. Picture of current accelerometer sensor

The sensor housing pictured below in Figure 4 features an internal chamber to fit the board, components, and port. The board will slide along the floor of the chamber into the slot, allowing the holes in the board and housing to align. M1 screws will go through the bottom of the housing up into the internal protrusions to prevent lateral movement of the sensor. The open face of the housing allows the user to insert/remove the sensor and to plug the ethernet cable into the port. The prongs on the bottom of the housing will be flexible and act as a buckle to clip to the platform. This will serve as an easy connection point to put the housing on and take it off the platform. To detach the housing from the platform, the buckle prongs are squeezed inward towards the center of the part until the lip no longer contacts the interfacing surface of the platform.

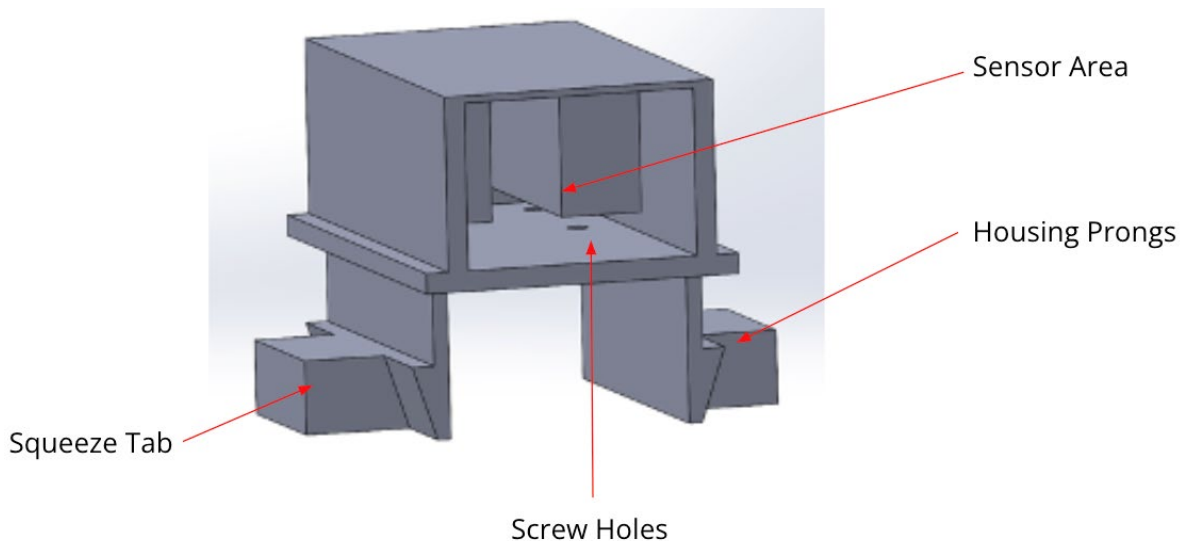


Figure 4. CAD model of sensor housing

The mounting platform depicted in Figure 5 receives the sensor housing prongs to prevent it from detaching upwards, with the centered uppermost surface sitting flush against the

underside of the housing. The endcaps on either side of the buckle cavity prevent the housing from sliding laterally and detaching from the platform. The slot running underneath the floor of the buckle cavity will accommodate a polyurethane strap that will wrap around the bike fork or chain stay, securing the assembly to the bike. The angled surfaces on the underside of the platform will interface with the bike, providing two contact points. A thin rubber layer will be adhered to the angled surfaces of the platform, acting as an interface between the platform and the bike frame, to prevent vibrational chatter and dampen unwanted noise during data collection, as well as widen the contact surface and increase friction to secure the assembly to the bike.

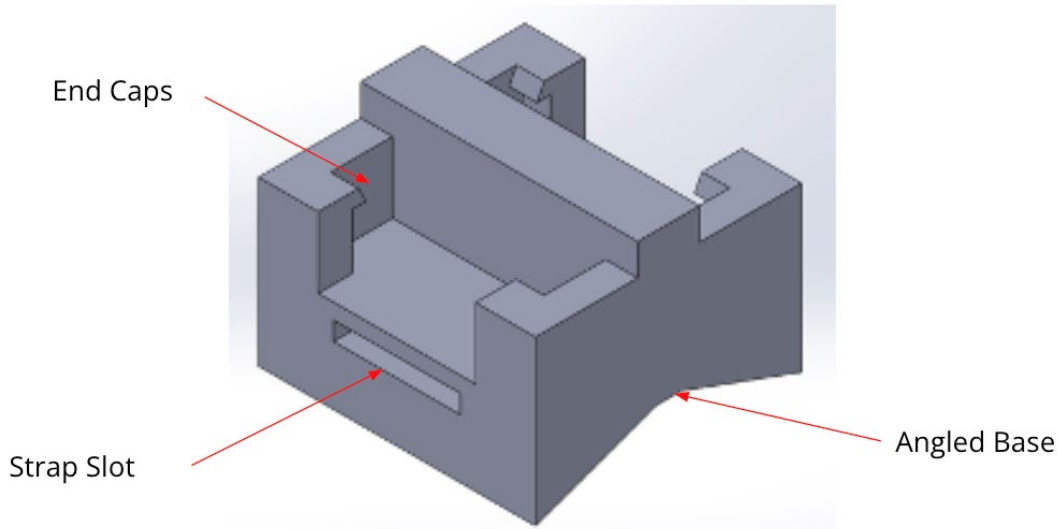


Figure 5. CAD model of platform mount

As previously developed, the main DAQ unit modeled in Figure 5 consists of two ethernet cable ports for receiving accelerometer data, three LEDs to indicate the status of the system's power, charging, and recording functions, a record button to start and stop data collection, a 4-character display, a Micro SD card slot, and a USB Mini B port. Two M5X0.8 screws run through the unit to attach it to the water bottle bosses on the main frame of the bike. Two circuit boards are housed in the electronics enclosure: a UI board which connects to the LEDs, record button, and display, and the main board, which contains the rest of the electronic components including the microcontroller, crystal oscillators, power filtering circuit, battery charging circuit, Micro SD slot, ethernet ports, and USB port.



Figure 6. CAD model of the main DAQ housing

The housing and platform are both roughly 1 inch wide by 2 inches long, with the assembly sitting 2 inches high off the bike surface. One sensor will be mounted to the fork housing, close to the axle of the front wheel and the other will be mounted to the chain stay, close to the axle of the rear wheel. The DAQ housing is 5 in x 3 in x 1 in and will be mounted to the bike's main frame, close to the center of gravity of the bike-rider system.

3.2 Manufacturing and Materials of Concept Prototype

For our concept prototype, the accelerometer housing and platform components were 3D printed with PLA using an Ultimaker 3 printer. These 3D printed parts were used for preliminary testing to determine the viability of this manufacturing method. If the method and material yield good test results and are deemed reliable, durable, and safe enough to use for our project, we will continue to use the 3D printed parts throughout this project because of its manufacturing ease and cost savings. However, if these components do not yield good test results, other materials and manufacturing methods will be considered, most likely 6061 aluminum alloy and material removal machining methods such as CNC or manual milling, waterjet cutting, and drilling. The DAQ housing is made of aluminum with plastic endcaps that have cutouts for the cable ports and SD card slot.

If we find from testing that plastic works well for housings and meets all our requirements, we will likely use it for the final product as well. If we do, we won't 3D print it due to the slow time requirements and high amount of labor required to clean up each part. Instead, we would most likely use injection molding, as this is much faster and more efficient for

mass production. If the plastic doesn't work, we would most likely use aluminum, which could either be milled as described above, or cast in molds.

Currently, the main DAQ unit is not fully defined and will be redesigned to include an accelerometer and gyroscope on its main board as well as additional ethernet ports to accommodate more sensors if we decide they are necessary to characterize suspension behavior. The DAQ housing will be redesigned to protect the electronics from water, dust, and debris entering through holes and gaps. If possible, the user interface will be redesigned to sit on the handlebars of the bike for easier access by the user, including an expanded, more modern display showing elapsed time during data collection and suspension adjustment recommendations afterwards to the user. However, this is a high-risk aspect of the design, as it will demand lots of effort and isn't necessary for the completion of our project. It is more of a 'stretch goal' than a main focus of ours.

The final placement and sensor configuration are also not yet fully defined. Our team will decide if and where to add additional accelerometers to the bike if it's necessary. A more detailed explanation of options we are currently considering can be found in Section 5.

4. Concept Justification

We have demonstrated proof of concept with the first round of preliminary testing of the sensor housing/mount assembly. The first draft of the CAD model was 3D printed and assembled to examine the print quality of the chosen settings, specifically the orientation of the part on the build plate and the tolerances of the fit between the housing buckle and the platform lip.



Figure 7. Concept Prototype Housing for Accelerometer

The housing was printed with the open face of the chamber upwards to minimize overhanging features which are problematic when 3D printing. This minimized the number of overhanging supports and resulted in a cleaner surface finish and tighter tolerances. The holes that accommodate the screws going through the sensor PCB were undersized and printed parallel to the build plate which is not ideal but allowed for the larger overhanging features to come out more cleanly. The holes were then effectively tapped directly by inserting the screws which worked well to secure the sensor in the housing when mounted to the bike.

The platform was printed with the thru slot perpendicular to the build plate to eliminate the need for supports that would be difficult to remove and potentially interfere with the strap passing through it. However, the thin overhanging surface that catches the buckle protrusions on one side had more imperfections. Because this small surface is critical to the functionality of the snap-fit design, a second iteration of this print would orient the platform such that this lip is not overhanging. Regardless, the snap-fit worked reasonably well by simply pressing the housing into the platform until it clips in, with all interfacing surfaces between the parts sitting flush with each other. One important detail we learned of is that the center ridge of the platform was too wide and interfered with the heads of the screws in the housing.

This snap-fit design was intended to make it easier for attaching and detaching the sensors to the bike. However, while functional, we learned it was unnecessary to add this feature because of the ease with which the assembly can be strapped to the bike. The snap-fit prongs introduce a potential failure mode: being the thinnest features of the part, these prongs could yield from fatigue stress after enough cyclic loading from the squeezing action to buckle and

unbuckle. With this realization, we will combine the housing and platform into one piece to reduce the number of parts needed and simplify the design. This way the form factor of the mount will be reduced and the clearance between the mount and the rider's shoe will be maximized, avoiding interference with the rider's motion. The system will also be lighter, easier to set up, and more robust and durable in the face of natural elements in an outdoor environment.

The angled surface of the bottom of the platform fits well onto the fork housing and seat stay, maintaining two points of contact with each component of the bike tested, and will fit any size and shape of fork housing and seat stay on the market.

The rubber pad between the platform and the bike significantly increased friction and, along with the polyurethane strap, this design fulfills the universal compatibility requirement from our design criteria.

The 3D printed PLA is waterproof and will protect from dust and debris, but we are still researching solutions to waterproof the electrical connections where openings in the housings of the sensors and main DAQ are necessary.

5. Future Work

In the coming weeks we will begin our first round of metric testing using the current state of the system along with our prototype 1 sensor housings. The three following subsections of this document give a brief introduction into each of the metrics we plan to test initially and how we will collect data per sensor locations. All possible sensor locations are depicted in Figure 8 (credit given to SireAnko for the original drawing).

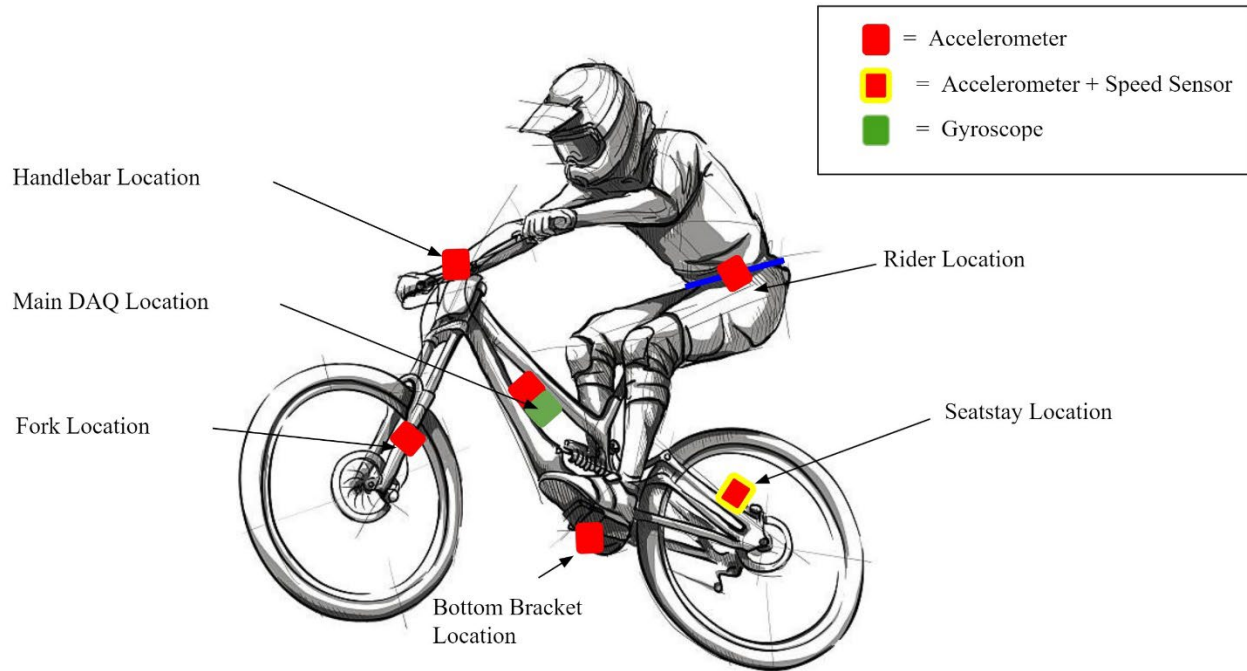


Figure 8. Sensor Types and Potential Locations

5.1 Suspension Balance Metric

One metric we will potentially use is a comparison of the front and rear axle behavior. Specifically, Figure 9 would incorporate the root-mean-square values of measured accelerations at each location and compare the magnitude of each value for the instant of time that the measurements were taken. From this metric, we hope to determine whether the front fork or rear shock's spring rate or damping is too high or low relative to the other one. In this way, the suspension settings could be adjusted to minimize the difference in acceleration, and therefore reduce extraneous pitching motion, which may lead to a more balanced, smoother, faster ride.

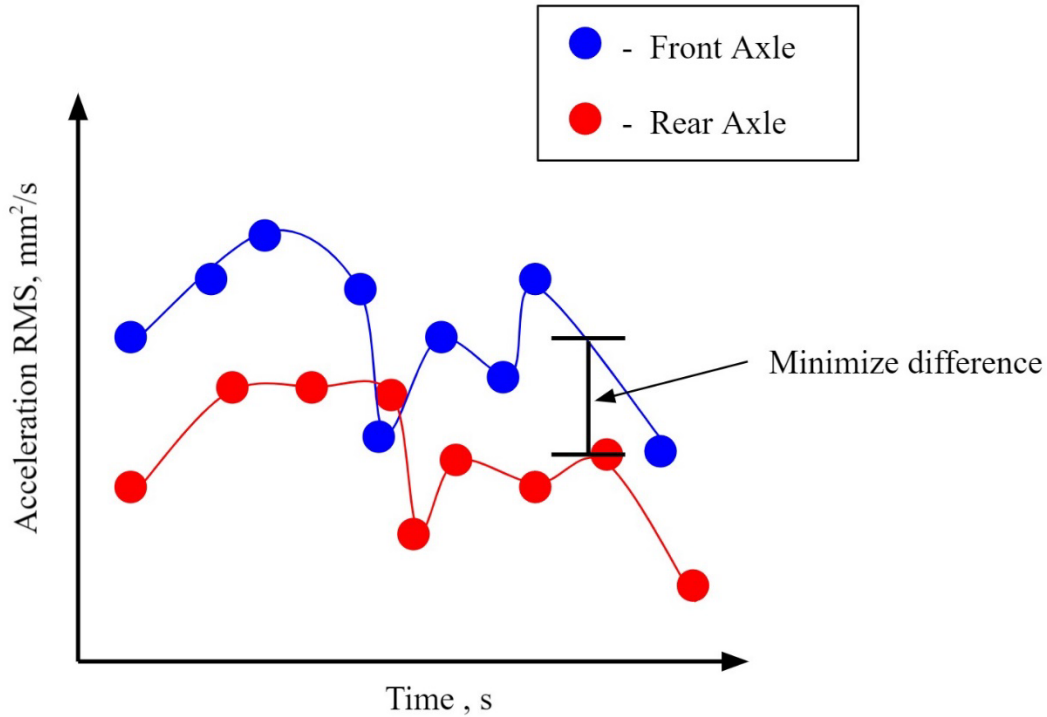


Figure 9. Plot demonstrating a comparison of RMS acceleration values for suspension balance

5.2 Rider Bike Metric

Another metric we will consider includes motion of the rider's body and compares it to the motion of the bike. While the rider's head position is being tracked in Figure 10, it could also be advantageous to track the motion of the rider's center of gravity with an accelerometer attached to a belt worn by the user. Body positioning is critical when biking competitively as it has a large effect on the rider's speed and control when navigating trail features. By collecting acceleration data of the rider's CG and comparing it to the bike frame's acceleration, we hope to analyze this motion in a suspension transmissibility context, similar to the next metric discussed below. Challenges for incorporating this metric include the added complexity of the rider's motion separate from the bike frame and the additional degrees of freedom that follow, as well as data transmission to the main DAQ which would be difficult with our current method using ethernet cables.

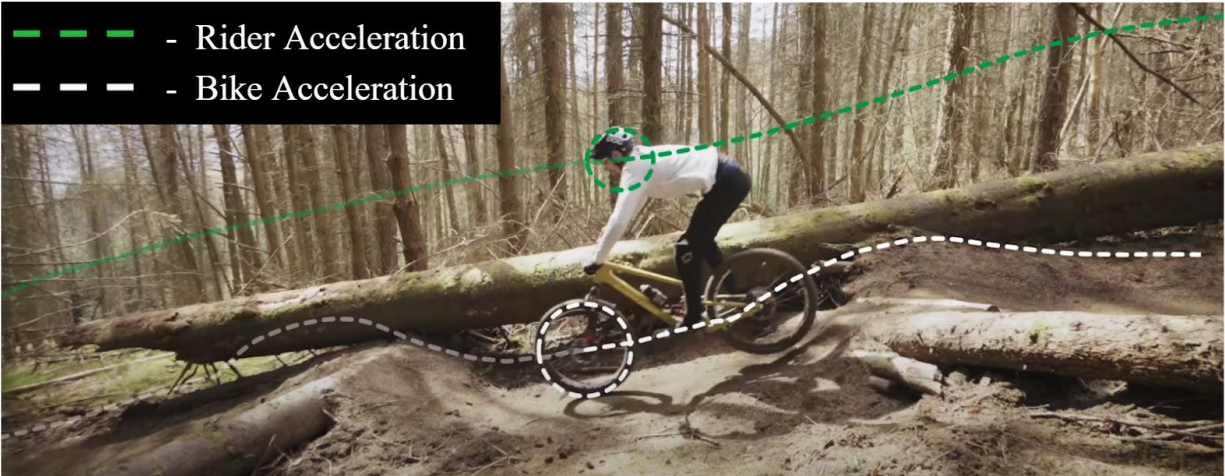


Figure 10. Positional/acceleration comparison of rider and bike
(credit for picture given to Ben Cathro)

5.3 Handlebar/Fork Metric

The transmissibility of acceleration between the front axle and the handlebar is a metric that could lead to a more comfortable and faster ride [2]. Ideally, the handlebar accelerations would be minimized as much as possible, allowing the user to experience less vibration and have more control during the ride. This may also help to reduce extraneous movement which could lead to a longer path traveled by the rider and therefore be slower. It could also provide more information pertinent to keeping the wheels contacting the ground, which is faster than when the bike and rider are airborne.

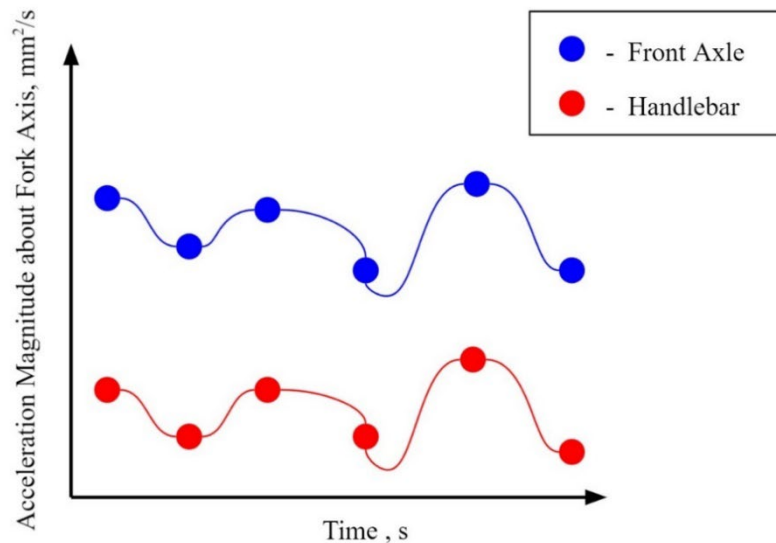


Figure 11. Example Plot of Handlebar and Front Axle Accelerations

Figure 11 represents an example plot of low transmissibility between the handlebar and front axle. The suspension will reduce or increase the transmissibility based on the selected settings. By minimizing the transmissibility, we can ensure the fork is not too stiff or bottoming out, as both cases would increase the transmissibility and cause discomfort, as well as contribute to a slower ride for the user.

We plan to assess the validity of each metric with an abundance of testing and data collection to see what correlates with faster ride times and more comfort and control experienced by the rider. Combining this with abundant background research on current techniques used in the racing industry and published methods in technical literature, we are confident in our ability to identify, develop, and implement meaningful metrics to quantify and evaluate suspension performance.

6. Project Management

The following section discusses what has been completed thus far and the planned tasks for the next major phase of this project: design and testing. This next quarter there will only be one main deliverable, the critical design review; therefore, the steps documented below will explain how we will reach the next design phase, engineering validation.

At a high level our goals for the DAQ leading up to the CDR include round 1 of metric testing, adding sensors, an updated PCB, and updated main/sensor housings. To achieve these new features and metrics we will be working in parallel. This means each one of us will be owning a feature and leading the process. We will maintain the proper team dynamic by holding design reviews with the team and Dr. Mello. The major dates of the design phase are documented below in Table 2.

Table 2. Key Tasks Leading to CDR

Major Deliverables	Completion Date
New Component Research	1/6/22
New Component Selection	1/8/22
Round 1 Metric Testing	1/11/22
Interim Design Review	1/13/22
PCB Rework/ Redesign	1/14/22
PCB Review	1/21/22
DAQ Housing Redesign	2/2/22
Critical Design Review	2/11/22
Spring Break (2/19-2/27)	
Test Protocol Drafts	3/4/22
Component Testing	3/11/22
Round 2 Metric Testing	3/18/22

These new features will follow the same design path: research on sensors, selection of the new component, CAD implementation, PCB additions, programming, and component testing. We have come to understand that if we want a variety of options when determining metrics for tuning a MTB suspension, we do not want to be limited on the sensors we have available to us. A complete timeline for the entire project lifespan can be found on the Gantt chart in Appendix D. The following categories breakdown the processes we have followed and will follow through the different design phases of our device.

Concept | Ideation

After the completion of the SOW, we spent one week brainstorming and ideating on the main functions of the device. We generated over 100 ideas total for the functions we were brainstorming on. Then we turned our focus to the mount and housing for the accelerometers so that we could begin preliminary testing with the DAQ and gain experience. The next two weeks we performed concepting activities and generated a few main concepts that boiled down to our main prototype. This process will continue throughout the lifespan of our project as this progression sets up for a smooth design phase.

Design | Prototype

After determining the main concept that would be prototyped, it was necessary to place an order on the off-the-shelf (OTS) parts that we were integrating into our system. By ordering these ahead of time, this allowed for us to take measurements on the OTS parts and begin the CAD work of the housing. After succeeding in the 3D printing of our housing we were able to mount it to the bike and perform our proof-of-concept tests. Following this process, we were able to find flaws in our design, but that is why we do create rapid prototypes. We plan to perform these steps again when adding new components and reworking our system. The specific dates for ordering of parts can be found in the Gantt chart.

Rework | Optimization

Following the completion of component testing, we will begin our investigation of tune metrics. This means that our system will need to be shook down to verify that the data we collect is reliable. During this process it is likely that we will have to rework our CAD/programs/PCB to correct our data collection. By giving ourselves time to make these corrections we should be able to achieve the level of data collection we desire. Once again, throughout the lifespan of our project we will rework our design until we deem the quality we desire has been achieved.

7. Conclusions

This Preliminary Design Review describes the development of the initial MTB Suspension Tuning DAQ concepts and prototype development. It details the specifications and design requirements, and how the selected concept meets them. For the sensor housings, we came up with a design involving a plastic holder that locks into a plastic base secured to the bike's frame with an elastic strap. Three possible metrics are described, taking into account acceleration data at the suspension, fork, and rider in different ways. Through testing we will determine which of the three metrics, as well as any additional variables, are needed to accurately represent the quality of the suspension setup. The document then goes on to describe future plans of the project if this current direction is approved. The next major steps after this PDR will be to begin testing the concept prototype and begin redesigning the DAQ components to create our full prototype system.

References

- [1] S. R. Waal, "A Quantitative Approach to Tuning a Mountain Bike Suspension," thesis, 2020.
- [2] S. Hanly, "Top 12 vibration metrics to monitor & how to calculate them," *enDAQ Blog for Data Sensing and Analyzing*. [Online]. Available: <https://blog.endaq.com/top-vibration-metrics-to-monitor-how-to-calculate-them>. [Accessed: 18-Nov-2021].
- [3] S. J. Park and M. Subramaniam, "Evaluating methods of vibration exposure and ride comfort in car," *Journal of the Ergonomics Society of Korea*, vol. 32, no. 4, pp. 381–387, 2013.

Appendices

Appendix A: Ideation Concepts (best ideas) and Models

Sub-Functions	Idea
Attach Housing to Bike	Velcro
	Welding
	Adhesives
	C-Clamp
	Belt Loop
	Magnets
	Zip ties
	Ratchet Strap
	Watch Band
	Glue
Measure Speed	GPS (phone)
	App (Strava)
	Timer Clock
	Integrate Acceleration
	Mechanical Speed Sensor
	Checkpoints with Time Intervals
	GPS onboard DAQ
	OTS sensor online
	Button on Handlebars to initiate/terminate
	iPhone App
User Interface	Sensor detection to initiate/terminate
	Touch Display on main device
	Foot button
	Link with Strava
	Link with Garmin
	Able to pause instead of terminating
	Two buttons to interface (start/stop; Keymark important times)
	3d-Print
	Bottom Tapered
	Bottom Radial
Housing	Hinged lid
	Locked lid
	Slide sensor through side
	Screws to hold board
	String to hold board

Appendix A (cont.)

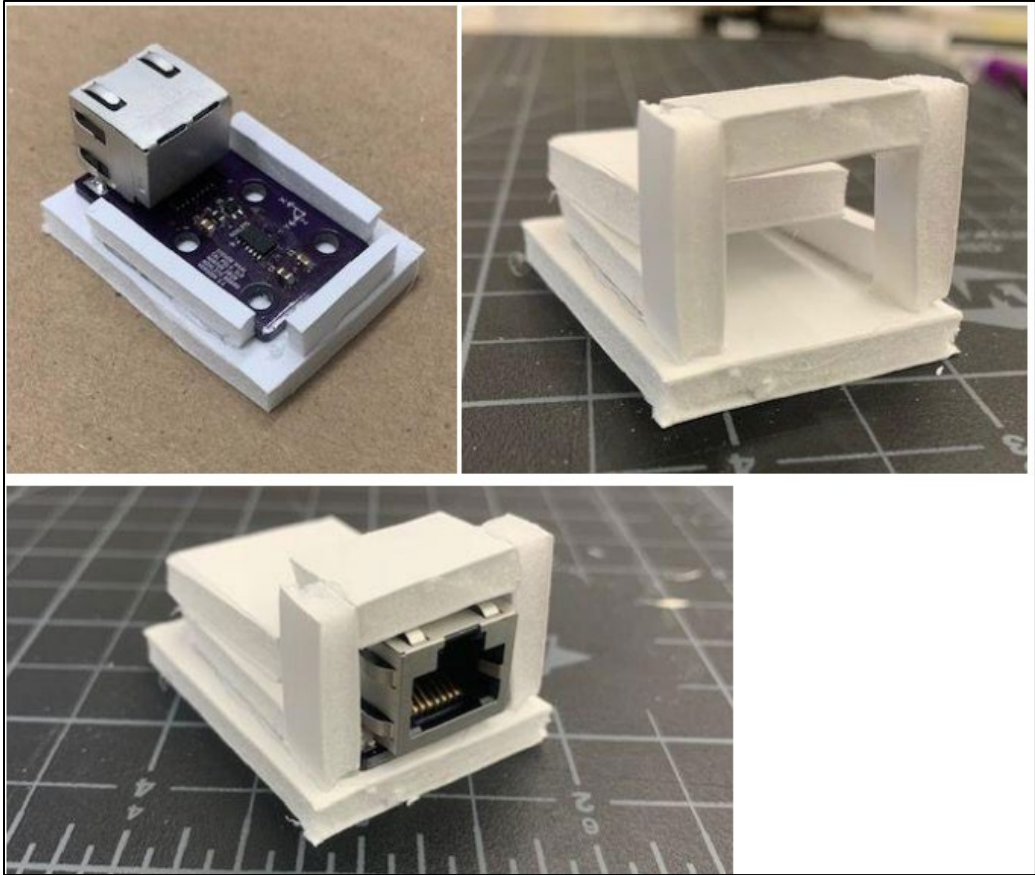


Fig A.1: Ideation Model Showing Possible Sensor Housing

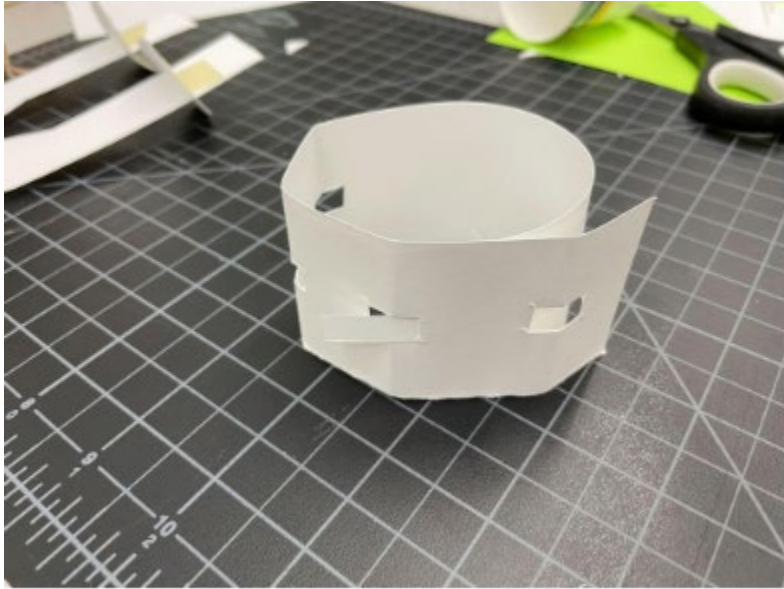
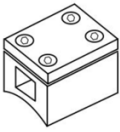
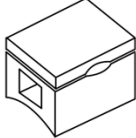
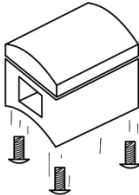
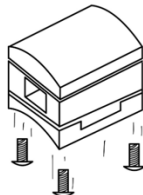
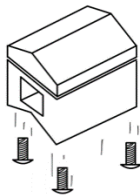
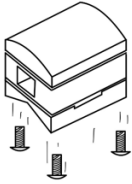


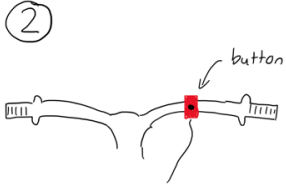
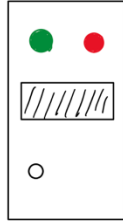
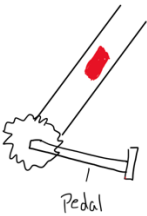
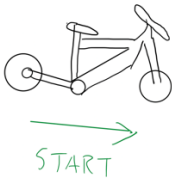


Fig A.2: Ideation Model Showing Possible Securement Strap for Mounts

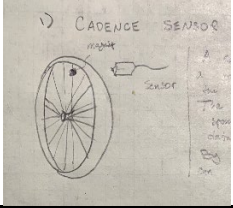
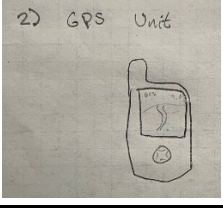
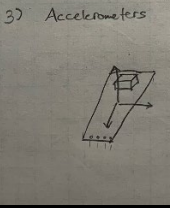
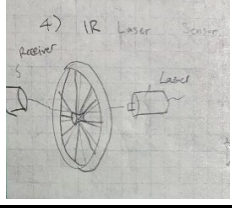
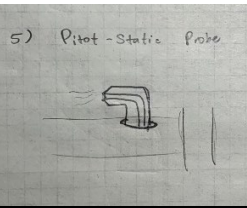
Appendix B: Pugh Matrices

Sensor Housing							
Low Cost	+	-	+	-	+	+	S
Aesthetics	S	+	+	+	+	+	S
Manufacturability	+	+	-	-	-	-	S
Weight	+	+	+	+	+	+	S
# Of Parts	S	+	-	+	+	-	S
Waterproofing	+	+	+	+	-	+	S
Foolproof	S	-	-	+	+	+	S
Universal	-	-	-	-	+	+	S
Protection	-	-	+	+	+	+	S
Total	2	2	1	3	5	6	0

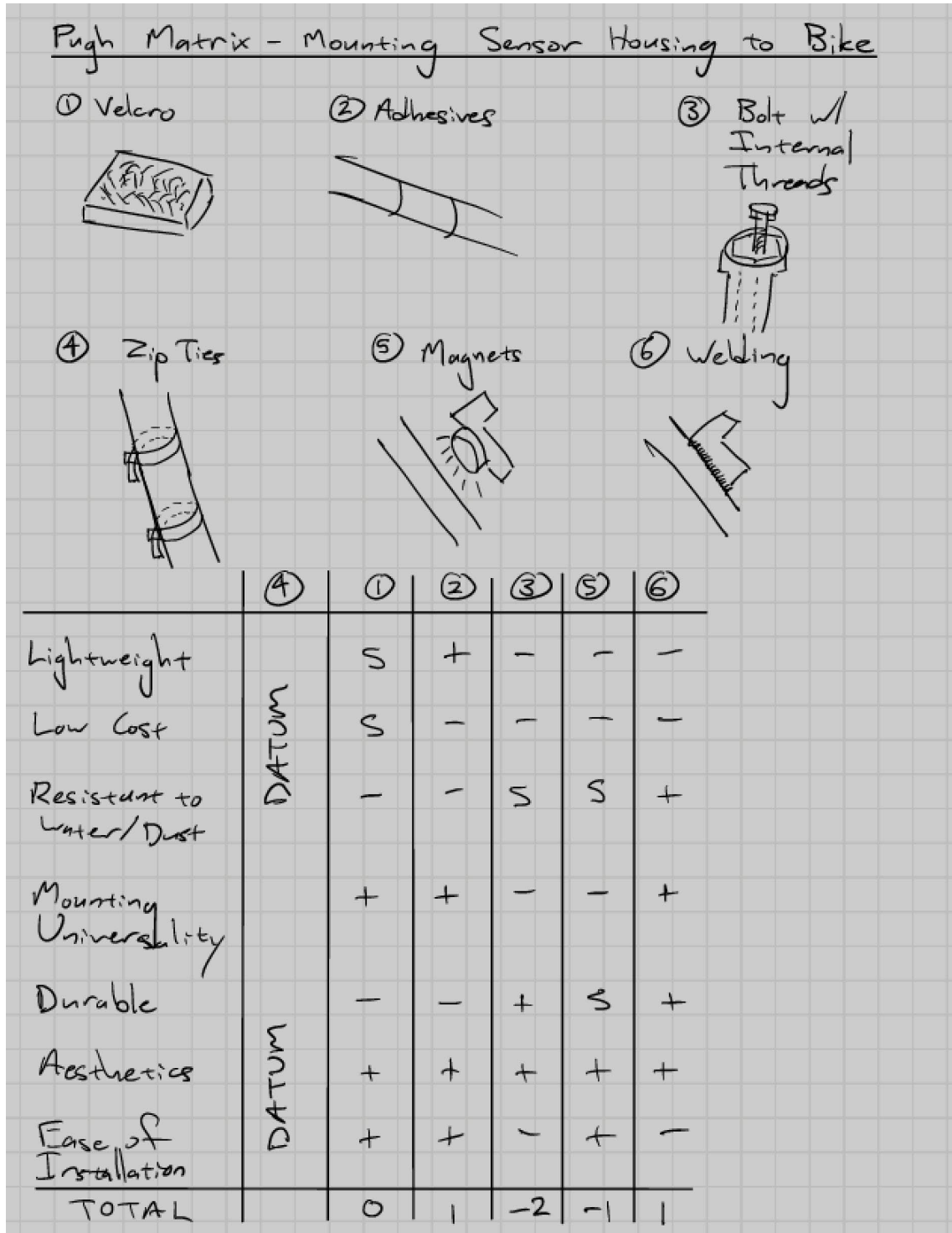
User Interface	① 	② 	③ 	④ 	⑤ 
Weight	S	+	+	+	+
Low Cost	S	S	S	S	-
Water resistant	S	S	-	S	S
Simplicity	S	+	+	+	+
Universality	S	S	S	S	S
Durable	S	-	-	-	S
Aesthetics	S	-	-	-	-
Small Profile	S	+	+	+	+
Easy to Install	S	-	+	-	+

Total	0	0	1	0	1
-------	---	---	---	---	---

Appendix B: Pugh Matrices (cont.)

	1. Cadence Sensor 	2. GPS Unit 	3. Accelerometer 	4. IR Laser Sensor 	5. Pitot-Static Probe 
Speed Sensor					
Weight	S	-	S	S	+
Low Cost	S	-	-	S	+
Water resistant	S	+	+	-	-
Simplicity	S	+	-	S	-
Universality	S	+	-	S	-
Durable	S	+	+	S	-
Aesthetics	S	+	S	S	-
Small Profile	S	S	+	S	-
Easy to Install	S	+	S	S	-
Total	0	4	1	0	-5

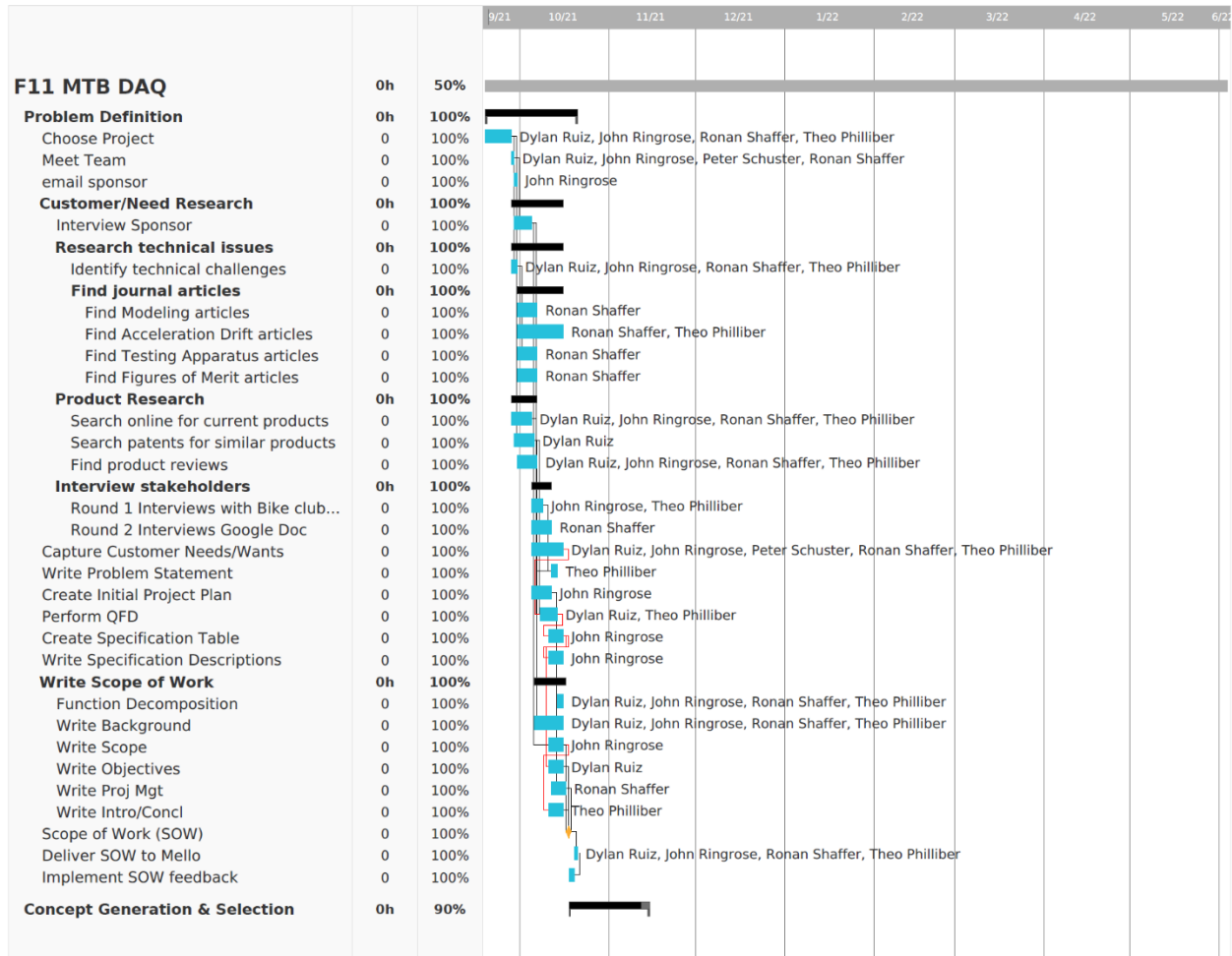
Appendix B: Pugh Matrices (cont.)



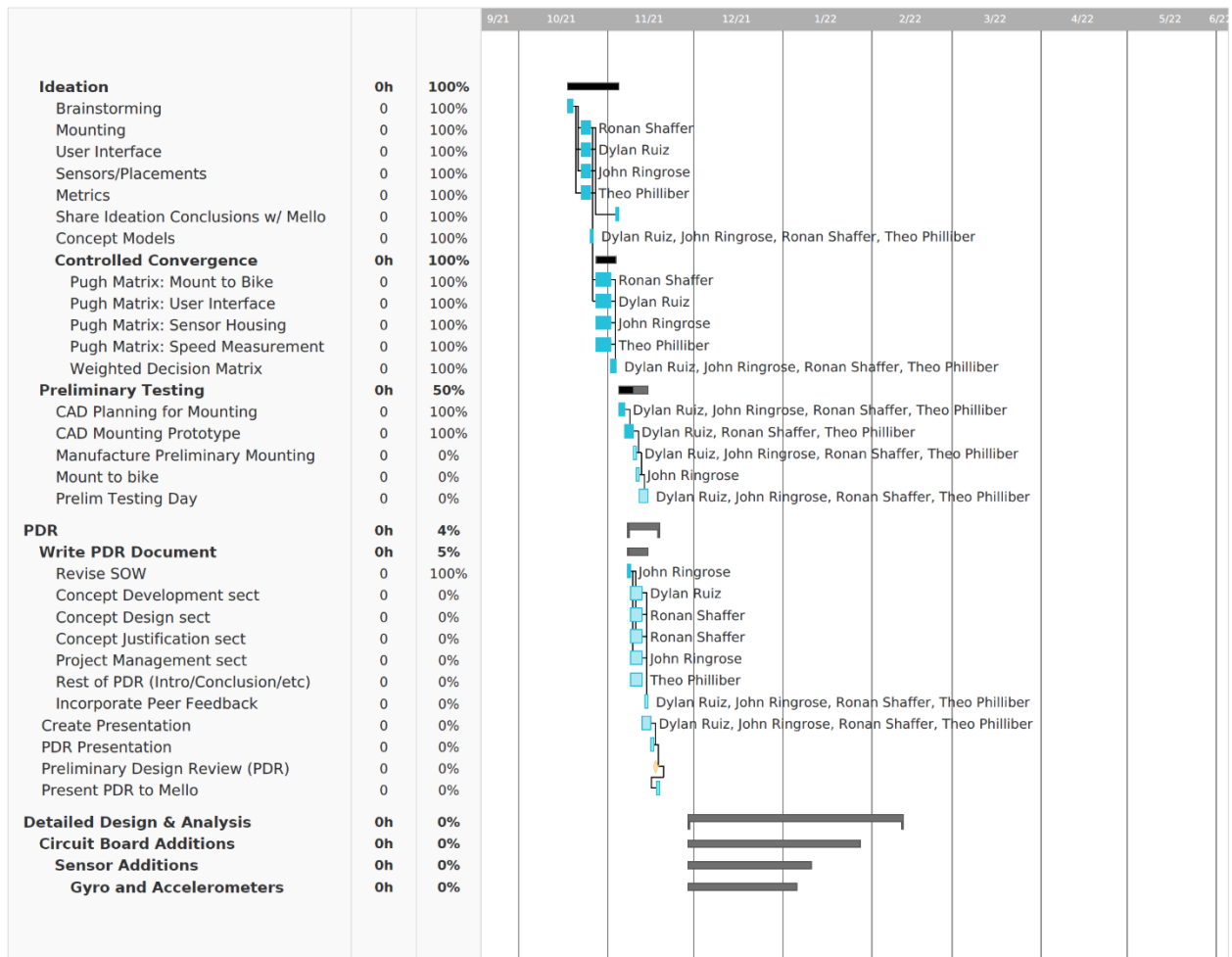
Appendix C: Weighted Decision Matrix

Final		Options											
		Design 1		Design 2		Design 3		Design 4		Design 5		Design 6	
Criteria	Weight	Score	Total	Score	Total	Score	Total	Score	Total	Score	Total	Score	Total
Easy Installation	0.09	7.25	0.65	8.25	0.74	8.75	0.79	7.25	0.65	8.50	0.77	8.00	0.72
Waterproof	0.08	8.00	0.64	6.88	0.55	6.63	0.53	8.00	0.64	7.13	0.57	8.63	0.69
Durability	0.12	7.50	0.90	6.50	0.78	6.00	0.72	7.88	0.95	6.50	0.78	9.13	1.10
Vibrational Dampening	0.17	5.00	0.85	5.50	0.94	7.25	1.23	8.75	1.49	9.00	1.53	9.25	1.57
Universality	0.12	8.00	0.96	8.63	1.04	8.88	1.07	8.50	1.02	9.13	1.10	8.75	1.05
Security	0.12	7.00	0.84	6.38	0.77	4.75	0.57	8.50	1.02	8.13	0.98	8.75	1.05
Manufacturability	0.05	9.50	0.48	8.50	0.43	9.00	0.45	8.00	0.40	7.75	0.39	8.00	0.40
Number of Parts	0.06	7.50	0.45	8.00	0.48	8.00	0.48	6.25	0.38	6.13	0.37	6.25	0.38
Cost	0.05	8.50	0.43	8.13	0.41	8.75	0.44	6.75	0.34	6.25	0.31	5.50	0.28
Aesthetics	0.08	4.75	0.38	6.75	0.54	7.63	0.61	4.75	0.38	7.38	0.59	9.00	0.72
Weight	0.06	8.00	0.48	7.63	0.46	8.63	0.52	7.38	0.44	6.75	0.41	6.75	0.41
Total	1												
Sum		7.05		7.12		7.40		7.70		7.78		8.35	

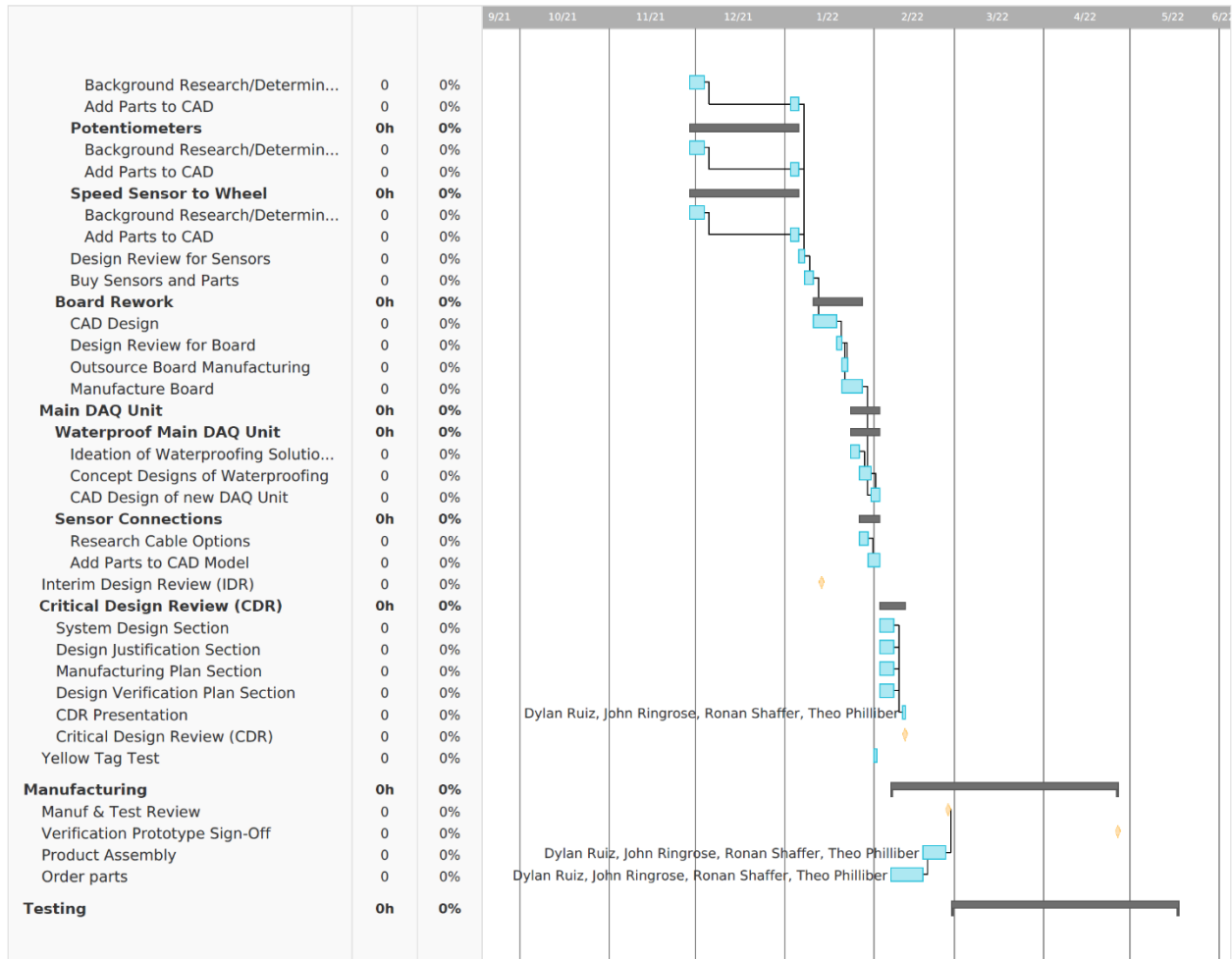
Appendix D: Gantt Chart



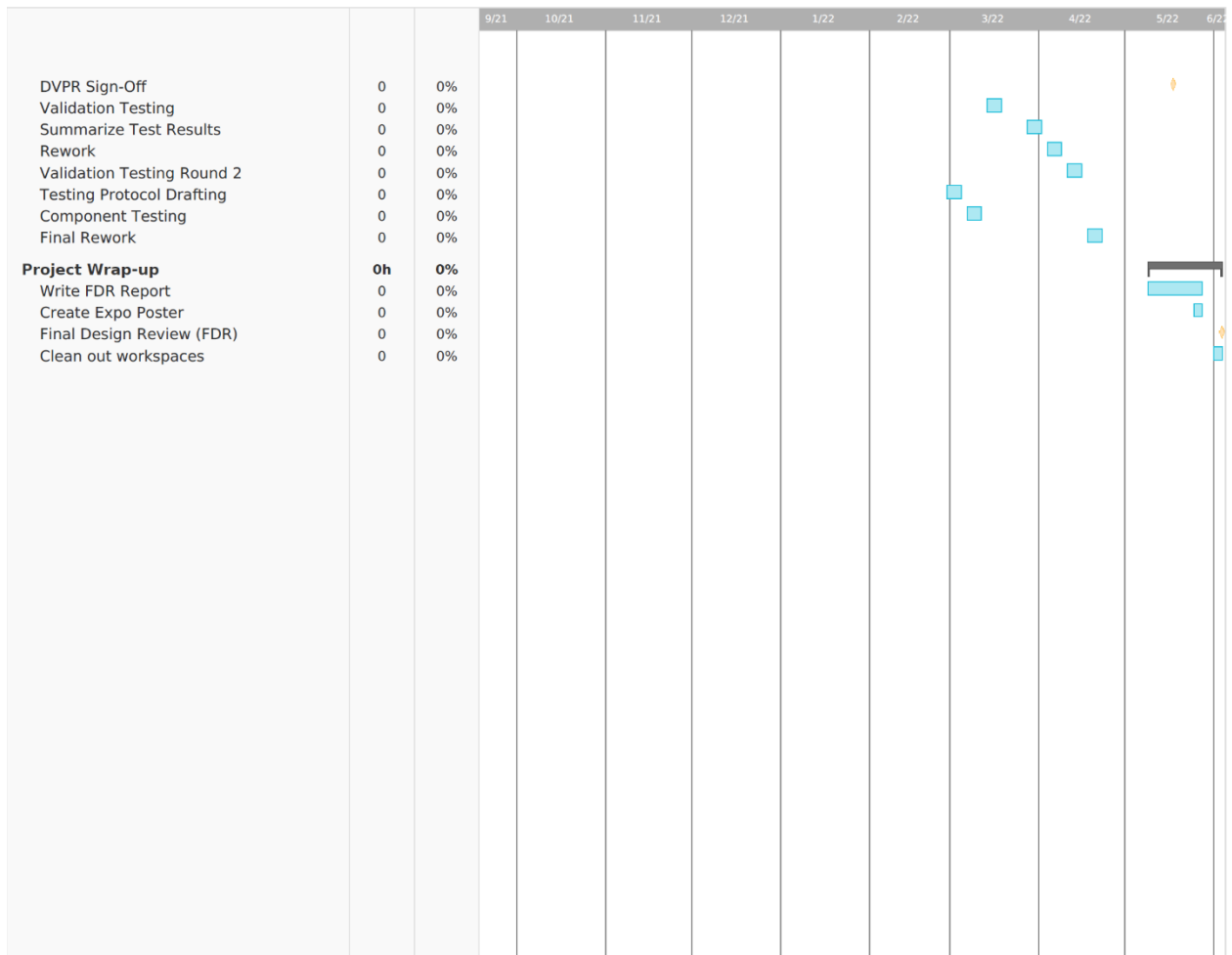
Appendix D: Gantt Chart (cont.)



Appendix D: Gantt Chart (cont.)



Appendix D: Gantt Chart (cont.)



Appendix E: Design Hazard Checklist

PDR Design Hazard Checklist

F11 MTB DAQ

Y	N	
	●	1. 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?
●		2. Can any part of the design undergo high accelerations/decelerations?
●		3. Will the system have any large moving masses or large forces?
	●	4. Will the system produce a projectile?
	●	5. Would it be possible for the system to fall under gravity creating injury?
	●	6. Will a user be exposed to overhanging weights as part of the design?
	●	7. Will the system have any sharp edges?
	●	8. Will any part of the electrical systems not be grounded?
	●	9. Will there be any large batteries or electrical voltage in the system above 40 V?
●		10. Will there be any stored energy in the system such as batteries, flywheels, hanging weights or pressurized fluids?
	●	11. Will there be any explosive or flammable liquids, gases, or dust fuel as part of the system?
●		12. Will the user of the design be required to exert any abnormal effort or physical posture during the use of the design?
●		13. Will there be any materials known to be hazardous to humans involved in either the design or the manufacturing of the design?
	●	14. Can the system generate high levels of noise?
●		15. Will the device/system be exposed to extreme environmental conditions such as fog, humidity, cold, high temperatures, etc.?
	●	16. Is it possible for the system to be used in an unsafe manner?
	●	17. Will there be any other potential hazards not listed above? If yes, please explain on the reverse.

Appendix E: Design Hazard Checklist

Description of Hazard	Planned Corrective Action	Planned Date	Actual Date
The design will undergo high accelerations based on the way the user of the design rides the bike the design is attached to.	When testing, we will have an experienced rider wear safety protection while being mindful of riding the bike safely.	11/20/2021	
The system itself will not be large in mass, but it is attached to a bike that will be moving fast. A fast-moving bike can be a hazard to spectators.	When testing, we will spectate the rider from a safe place. We will have a specified segment the rider will take when testing, allowing us to know the path the rider will take.	11/20/2021	
There is currently a battery within the main DAQ system.	Currently, this hazard is low-risk due to the housing of the main DAQ providing protection from the electrical components.	11/20/2021	
The user will have to be riding a mounting bike to use this design.	There will be a cautionary notice before the use of the device listing this hazard. Since this hazard is not affected by our design, this is the most we can do	1/11/2022	
The manufacturing process will include PCB rework. There are hazards with the tools used such as a solder.	The people manufacturing will be trained in safety precautions before operating the tools.	1/19/2022	

The device will be used in various environments.	The sensors and the main DAQ unit will be waterproofed for safer use.	2/2/2022	
--	---	----------	--

MTB Suspension Tuning DAQ Critical Design Review (CDR)

By: Dylan Ruiz

dyruiz@calpoly.edu

Ronan Shaffer

rmshaffe@calpoly.edu

John Ringrose

jringros@calpoly.edu

Theo Philliber

phillibe@calpoly.edu

February 28, 2022

Sponsor:

Professor Joseph Mello

Mechanical Engineering Department

California Polytechnic State University

San Luis Obispo

Abstract

This Critical Design Review (CDR) describes the current state of the 'MTB Suspension Tuning DAQ' senior design project. This project aims to quantify the suspension settings of mountain bikes (MTB) to improve the riding performance and reduce vibrational discomfort. A data acquisition system (DAQ) will collect data during a ride, which will be analyzed after the fact to suggest changes to the tuning parameters of the suspension. The CDR details the overall design and operation of the system and justifies the design choices made. Further, the plans for manufacturing and testing the verification prototype are laid out and explained.

Since the Preliminary Design Review, most of the progress has been on the electrical systems in the DAQ. The new sensors were selected, and circuits were designed to integrate them into a new iteration of the DAQ. Finally, a new PCB schematic was created, which will be sent off for manufacturing after approval of this CDR. To go along with the electrical system, new code was written to communicate with the new sensors. The next steps involve manufacturing the PCBs, soldering on the electrical components, validating the prototype, and designing the tuning algorithm.

Table of Contents

- 1. Introduction..... 1
- 2. System Design 2
- 3. Design Justification..... 5
 - 3.1 Housing Design Justification 5
 - 3.2 Sensor Data Collection Justification..... 6
 - 3.3 Safety, Maintenance and Repair Consideration..... 8
 - 3.4 Unresolved Issues and Concerns 9
- 4. Manufacturing Plan..... 10
 - 4.1 Procurement 10
 - 4.2 Manufacturing..... 10
 - 4.3 Assembly..... 11
- 5. Design Verification Plan..... 12
 - 5.1 Uncertainty Analysis..... 13
- 6. Conclusions..... 15
- 7. References..... 15
- Appendix A. Drawing and Specs Package i
- Appendix B.1 Project Budget xi
- Appendix B.2: Bill of Materials..... xii
- Appendix C. Uncertainty Propagation..... xiii
- Appendix D. FMEA..... xv
- Appendix E. Design Verification Plan..... xvi
- Appendix G. Gantt Chart xvii
- Appendix H. Design Hazard Checklist..... xxi
- Appendix I. Accelerometer Calibration Test Description xxiii

1. Introduction

In this CDR report, the full design of the Mountain Bike Data Acquisition System is described. The purpose of the DAQ is to collect data during a mountain bike ride. This data will then be processed to recommend tuning changes to improve the performance of the suspension in reducing vibration and increasing bike speed.

Since PDR, many changes have been made to the electrical system of the DAQ, as well as the firmware and the housings. The scope of our project now focuses on building a data acquisition system to start collecting data and justifying our metrics. Section 2 (System Design) describes the design in its entirety, and the way in which it will function. Section 3 (Design Justification) explains why decisions were made about the new design. Section 4 (Manufacturing Plan) goes on to explain how the verification prototype will be produced, from the procurement of materials to the final assembly of created components. Finally, Section 5 (Design Verification Plan) continues to describe how the prototype will be tested to verify that it meets our previously laid out specifications.

After CDR, the remainder of the project will be spent creating the prototype, developing the recommendation algorithm, and field testing. Due to issues with the provided DAQ this project was based around, the scope of this project changed somewhat. Most of this quarter was spent troubleshooting and redesigning the firmware and hardware of the DAQ system, rather than focusing on the algorithm and figures of merit as planned. The Spring Quarter will be mostly spent testing and tuning. As we test, we will be able to iterate and improve our algorithm until it demonstrably improves the bike's speed and comfort.

2. System Design

The MTB DAQ System is a modular, portable system that is compatible with any bike a person would ride off roads or on mountains. The system consists of one central unit and two auxiliary sensor units that can be mounted in several different locations on a bike. By strapping on the sensors and main unit and connecting them with cables, data can be collected with the press of a button without impeding the user's ability to ride their mountain bike down their favorite trail. Our system can functionally be broken down into 6 subsystems:

1. Mount & Protect Unit
2. Power Unit
3. Interface with Rider
4. Collect Data
5. Store Data
6. Interpret Data

To mount our system components, sensors will be inserted into custom designed 3D printed housings made of PLA, shown in Figure 1. The printed circuit board containing the sensor and associated circuitry slides into a slot until the unit is fully enclosed and snug within the housing, at which point the holes in the PCB align with holes in the housing and screws are inserted to fix it in place. The angled outer surface opposite the screws provides two points of contact with the fork housing and seat stay for each respective sensor.

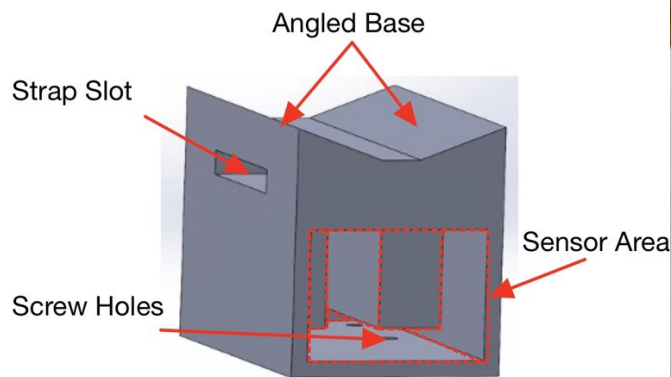


Figure 1: Accelerometer CAD and Housing with Accelerometer PCB Inserted

The rear sensor on the seat stay contains an accelerometer only, while the front sensor on the fork housing contains an accelerometer and a hall effect sensor. This component senses the presence of the spoke magnet clipped to the front wheel and sends a signal when the magnetic fields interact. This will record angular speed of the wheel and, with the diameter of the wheel, can be used to calculate the velocity of the bike assuming a no slip condition between the wheel and ground. These accelerometer positions can be seen in Figure 2.



Figure 2: Accelerometers Positioned at the Fork and Seat-stay Respectively

As shown in Figure 3, The main unit is mounted to the center of the bike frame with two screws fastening the housing to the water bottle bosses. This unit contains the main board with accelerometer and gyroscope, microcontroller, ethernet ports, Micro SD card slot, and USB Mini B port for charging. It also contains a connected User Interface board with LED indicators, record button, and display screen. Additionally, 5 batteries are connected in parallel to supply power to the unit. Specifications for all electronic components can be found in Appendix A.



Figure 3: Mounted Main DAQ System on Bike Frame

During operation, the rider flips the power switch to turn on the main unit and then holds the bike still on flat ground while the accelerometers calibrate so their biases can be calculated. Once the rider is ready, they press the record button to begin data collection, which increments the

log count on the display screen, creates a new file on the SD card, and lights up an LED to indicate recording is in progress. All 3 accelerometers, the gyroscope, and the hall effect sensor then start sending data through ethernet cables and PCB traces back to the microcontroller in the main unit, which sends this data to the SD card to store in memory. The system will continue to collect data until the record button is pressed again, at which point the file will be closed and the LED will turn off.

Once the data is collected and stored on the SD card, it is later processed on a PC with MATLAB using metrics that will be developed by this team next quarter.

Firmware

All code responsible for running the program used for data collection during operation was written in MicroPython. The communication protocol used is SPI for all accelerometers and the hall effect sensor because of their high data output rate, while the gyroscope will necessarily use its manufacturer configured I2C protocol to send data back to the microcontroller. Using a FIFO buffer for all sensor data, the MCU will write the data to the SD card using SDIO where it will be stored in a binary format to save memory.

Cost and Budget

In Table 1 below, the costs associated with each category of component is listed, along with estimated shipping and tax costs, for **two verification prototypes** to be built, as requested by our sponsor. A more specific breakdown of the project budget can be seen in Appendix B.1.

Table 1. Project Cost and Budget

System	Component	Cost
Housings	Housings	\$12.00
	Housing Straps	\$16.50
Electronics	Sensors	\$44.00
	Electrical Components	\$4.00
	Update PCBS	\$88.00
Mechanical	Mounting Hardware	\$5.00
	Spoke Magnet	\$10.00
Shipping + Tax	-	\$90.00
Total Cost		\$269.50
Project Budget		\$500.00
Funding Left		\$230.50

3. Design Justification

At the request of our sponsor, our designs for the sensor housings and system design are to be created with the intent of collecting riding data, without the concerns for a product to be used by other customers. Because of this, our design justification will be based on the specifications needed to effectively collect data for metric testing, without the concerns of uncommon failure modes that can occur when it is being used by a mass audience.

3.1 Housing Design Justification

With this in mind, we designed the sensor and main DAQ housings to be 3D printed with PLA filament. Under normal operation, there will be no significant mechanical stresses applied to the housings or system itself, so FEA and Stress analysis were not included in our analyses. Our main failure modes would be that which affects the quality of data collection, rather than what would affect the function of the system based on the use of a mass audience. The sensors we selected are rated to have a large shock tolerance that can be shown from their respective datasheets (Appendix A).

To justify our housing design in terms of the quality of data collection, we had to ensure the housing would not slip during operation and the rubber pad wouldn't dampen the accelerations to a great extent. To ensure this, we subjected the accelerometers and housings to a shake table test. Comparing both orientations, horizontal and vertical, as well as testing with and without the rubber pads, we confirmed that the rubber pads add enough friction between the housings and the frame to eliminate potential slip and does not dampen the accelerations to a significant effect. These orientations can be seen in Figure 4.

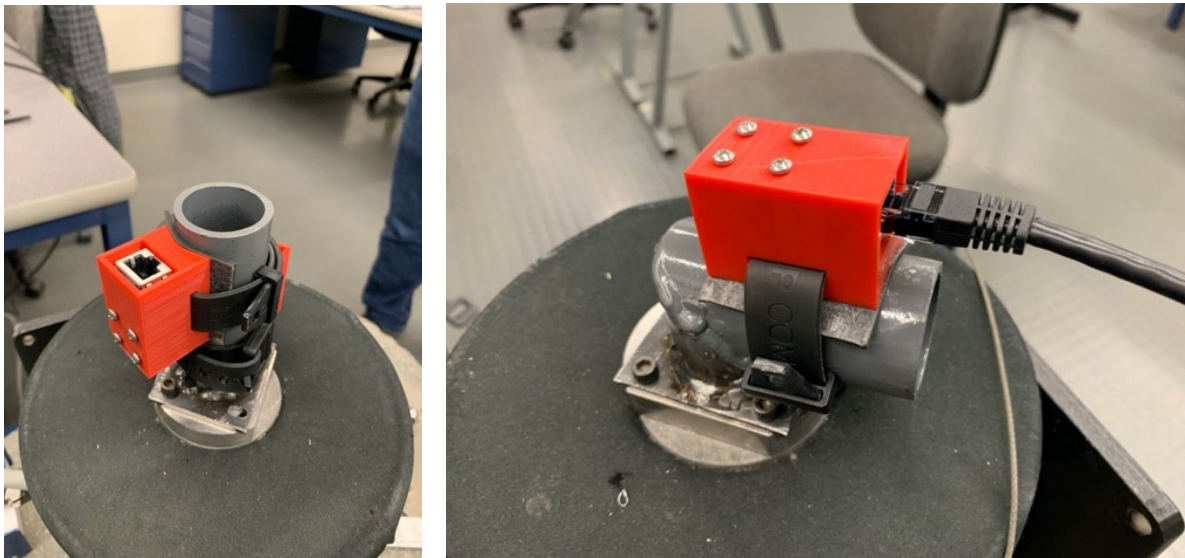


Figure 4: Accelerometer Vertical and Horizontal Positions Respectively for Lab Testing

Figure 5 shows the similarities between the accelerometers in a horizontal position, both being subjected to 20hz on the shake table. The accelerometers show offsets that our team will calibrate in future tests to make the data relevant.

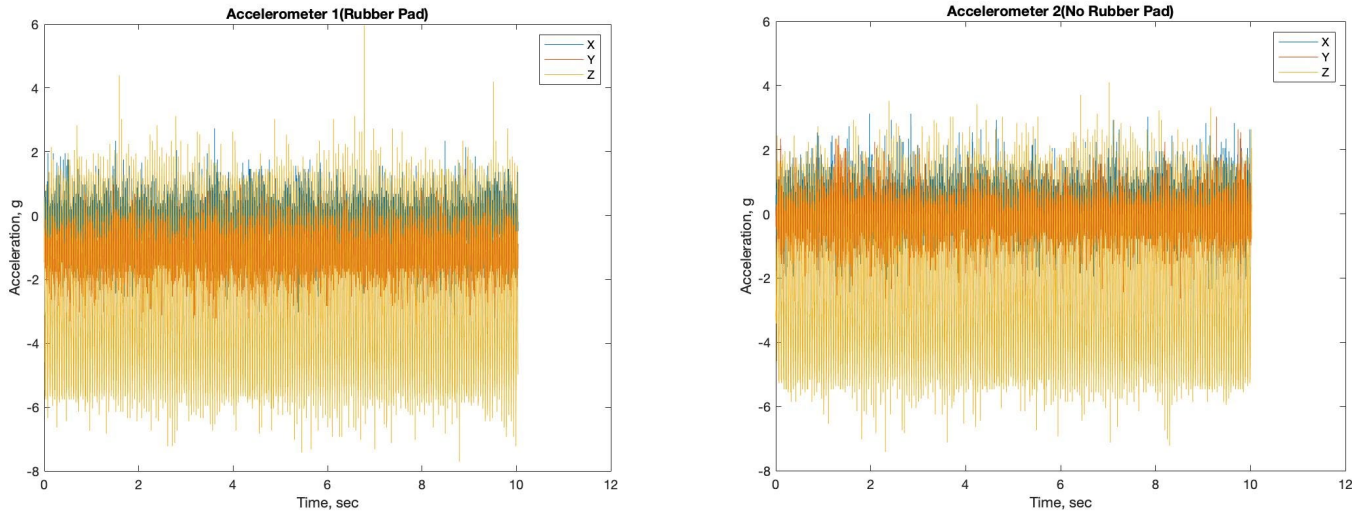


Figure 5: Accelerometers in Vertical Position Subjected to 20Hz on the Shake Table

Currently, DAQ systems made for mountain bikes are similarly designed in the way they mount their housings to the bike’s frame. Instead of the fastening band that our design implements, other DAQ systems often used wire ties. The intended audience for the other DAQ systems is for professional use, so they did not design them with the mass audience in consideration. Wire ties are similar to our band design, except we incorporated a less permanent attachment due to the constant testing we will have to do with our system.

3.2 Sensor Data Collection Justification

Similar to the housings, the way the sensors collect data is also a design that needs to be justified. This design includes the MicroPython code used to drive the sensors and the sensor selection itself. We did not have to design more code to drive the accelerometers because Steven Waal’s version of the accelerometer driver works as is intended, collecting data at a rate of 1600Hz. This was justified through multiple field tests and lab tests using the accelerometer to collect data.

The gyroscope was the new sensor added and was selected based on its output frequency range and the fact that it is 3-axis. The specifications of the gyroscope are shown in Appendix A. The gyroscope is collecting data at a slower rate than the accelerometers because the angular velocity is not rapidly fluctuating. We planned to verify that our driver works with the selected gyroscope by breadboarding it onto a Nucleo and collecting accurate data from it. However, due to shipping problems we did not have the necessary components to complete this bench test and will proceed with the test after this review. Although our system does not use a Nucleo as a microprocessor, the code can be modified to fit our intended system and microcontroller.

Continuing with the accelerometers, we tested in a range from 0hz to 1600hz to verify that the given accelerometers were able to accurately collect and transmit data on files that we can process through MATLAB. The following data is taken from the accelerometers in the horizontal position at rest.

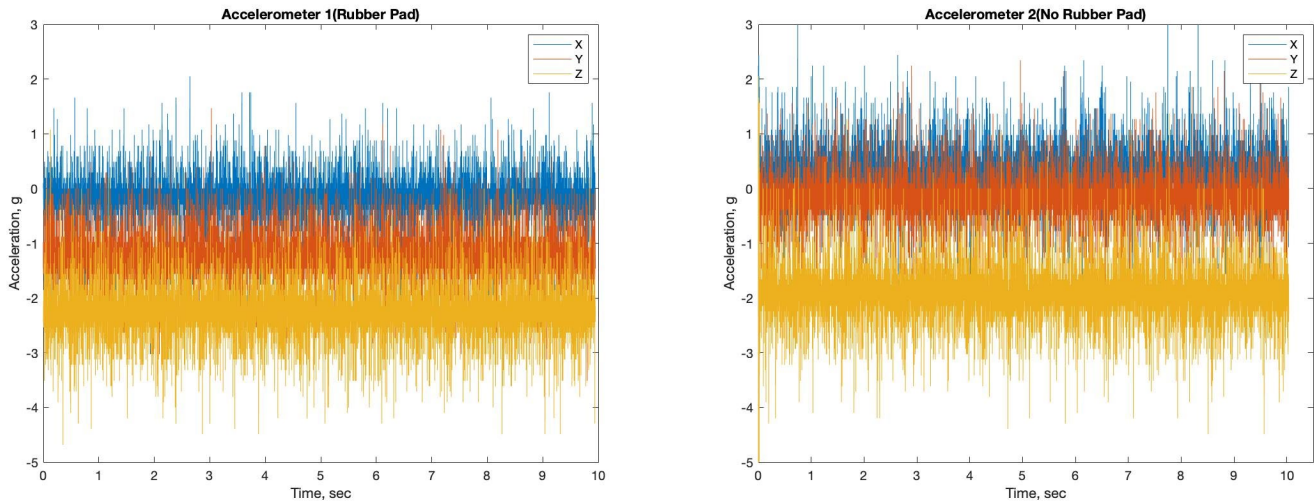


Figure 6: Accelerometer Data Taken at a Rest Position

As shown in Figure 6, the data displays the biases of both the accelerometers. Since both tests were taken when the accelerometers were at rest, the magnitude of their accelerations should be equal to 1g, however both accelerometers show biases in each axis. We will have to calibrate the accelerometers in a perfectly horizontal position to ensure accurate data is being collected and to make use of previous data taken.

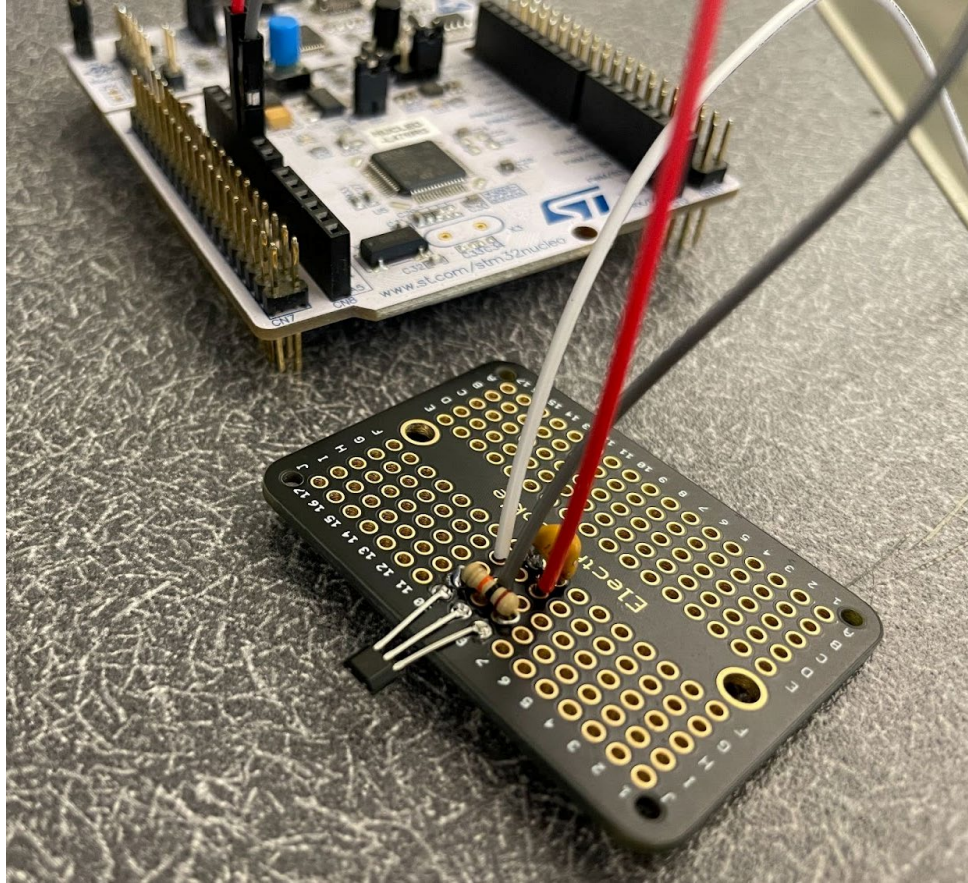


Figure 6. Hall Effect Sensor on Breadboard, Powered by STM32 Nucleo-L476RG Prototyping Board [5]

A prototype circuit was also made to verify the function of the Hall effect sensor. For this, we attached the sensor and all other components in the circuit to a solderable breadboard. The output and ground pins were attached to a voltmeter, and a magnet was passed in front of the sensor. As the magnet approached the sensor, the voltmeter read a value of 5V, and as the magnet departed, the voltmeter read 0V. The sensor and circuit functioned the way we designed them to, but not perfectly.

3.3 Safety, Maintenance and Repair Consideration

The safety of the user and the securement of the device is an important consideration that our team took into consideration. The team reviewed the safety of the design by creating a Failure Modes and Effects Analysis, which is attached in Appendix D. Using this process, our team can determine how the design will fail, how these failures can affect the customers and the most critical potential issues. Because our objective with our design is to create this DAQ solely as a testing device, most of our failures will be software related and not affect the user's safety.

Some safety precautions include the design of small form factors for the housings to ensure no interference between the user's path and the housings. There will be no exposed conductors and wire ties are used to keep the ethernet cables attached to the bikes frame.

In order to protect the device from potential damage, we added housings with a thickness that is tolerable to potential crashes from riding the bike. These housings will protect the electronic components from external damage, as the sensors are already tolerable to shock. Other protective measures include a video demonstrating how to properly install the DAQ onto a mountain bike as well as a written manual on how to operate the device, which will be available with the verification prototype.

With our current design, there should be no need for maintenance within the system. However, in the case that bugs and software problems occur, there will be well documented code and documentation on the design of the DAQ system that the user can reference.

3.4 Unresolved Issues and Concerns

There are concerns related to the current DAQ design that we cannot ignore for future iterations of the design. The power supply design to the DAQ is not the most efficient. Most of our problems came from the insufficient power supply to the DAQ. The main board was designed to have batteries connected in parallel to supply the power. This would cause the rechargeable batteries life to drain faster than if it was powered by a singular battery. The power supply will need to be altered in our final iteration of the board design.

We found several issues during the Hall Effect Sensor verification test. The first was the proximity of the magnet to the sensor; the magnet had to be between 0.5 to 1 inch for the sensor to toggle on. This obviously leads to issues with the mounting of the system, as it allows little clearance for the spokes and sensor to pass by each other while still picking up velocity data. A second issue was with our selection of a latch-type sensor. The sensor turns on with a positive magnetic field and continues to stay there, only turning off again in the presence of an equal negative magnetic field. This means the magnet must be oriented so both poles can be read, which reduces the magnetic field's strength at the sensor and lowers the range of the circuit. We could also have a second magnet to turn off the sensor, but this is clearly not ideal either. We will continue to test with the other type of Hall effect sensors, which only output the measured intensity of a magnetic field, instead of latching on and staying there. While these issues exist, the underlying concepts were proven to work by this testing. Further experimentation with the Hall effect circuit will occur after CDR.

4. Manufacturing Plan

The following section presents the manufacturing processes that will be required in order to build the verification prototype. This includes the procurement of materials/components, manufacturing of custom parts, outsourcing of part manufacturing, and lastly, the assembly process.

4.1 Procurement

The manufacturing of materials and components consists of a variety of electrical components, sensors, 3D printed parts, and mounting hardware. Ronan is the elected purchaser of the components, however, the team and sponsor will hold component reviews prior to purchasing to ensure the components are accurately selected.

The electrical components (resistors, capacitors, crystals, etc.) will be purchased through a supplier called Digikey. With their wide variety of products, every electrical component can be purchased through them. Prior to purchasing components, each manufacturer will be researched to ensure the component is high quality.

The mechanical hardware will be purchased at a variety of suppliers. The OneUp straps used to secure the sensor housings to the bike will be purchased directly through them on their website. The spoke magnet that attaches to the rim will be purchased on the REI website. The remaining mounting hardware (screws and bolts) will be purchased through McMaster.

The final iteration of 3D printing will take place in the ME Department using the Formlabs 3+ printer. In order access this high-quality printer we will need to pay for a maintenance fee (\$10) and technician fee (\$45). We have selected a specific Formlabs material called “Tough 200 Resin” which will be purchased directly through Formlabs (\$175). The product spec sheet supplied by Formlabs can be found in the Appendix A.

4.2 Manufacturing

The fabrication of the updated main PCB and updated Accelerometer + Hall Effect PCB will be manufactured by JLC PCB. This manufacturer was selected based on their capability to produce high quality boards with a very short turn around and low price. One of the team members as well as Steven Wahl have used this manufacturer in the past and had good experiences with JLC PCB as well.

As stated in section 4.1, our housings will be manufactured with a Formlabs 3+ printer. The Formlabs 3+ printer uses Low Force Stereolithography (LFS) which is an advanced form of SLA printing that uses a flexible tank and linear illumination to turn liquid resin into the desired part. This print style was selected due to its excellent surface finish, part accuracy, and material strength/stiffness, all of which will make a finished look product but at a lower cost. The printer is located in the ME Department and we will be working with a shop technician to have the parts properly printed.

Lastly, the software development must be procured. The process of developing this code includes Steven Wahl's original code (modifying it for the new board pinouts) as well as utilizing code from our ME 305 and 405 scripts for the gyroscope. Pseudo code has been developed for the new sensors and the code can be validated using the nucleo/breakout boards during the manufacturing process for the PCBs. This allows the team to stay on track during the 1–2-week lead time for the PCB manufacturing.

4.3 Assembly

The surface mounting (SMT) of the components onto the board will be hand soldered by the team. The team has prior experience with soldering of electrical components on PCBs and is confident that components will be properly placed on the board. To ensure the boards are reliable, a rigorous quality control (QC) plan will be put into place. This QC plan begins with in-circuit testing using the designed access points in the board and comparing these values to Eagle Simulations as well as hand-calculations.

If the hand assembly is not successful, JLC PCB has the capability to do so. Their outgoing quality control (OQC) includes visual inspection, solder paste inspection (SPI,) x-ray inspection, and automated optical inspection. Their capabilities are much greater than that of hand soldering the components, however, their price to do so is another cost that could be avoided.

Once the PCBs have had all components soldered to the board, they will be mounted to the housings. The OneUp straps will be put through the sensor housings, ready to be placed on the bike. When the user is ready to collect the data, all that is necessary is to buckle down the sensors and tighten the through bolts on main DAQ housing.

5. Design Verification Plan

After completion of the verification prototype, we will need to identify if it meets all of our design specifications. For a full table of specifications and testing, see the Design Verification Plan in Appendix E.

Size — Main Hub and Peripheral Housings

The physical dimensions of the system are very important to its function, as they should not interfere with the rider's ability to operate the bicycle. We selected a maximum size for the hub of 12.5 cm long, 7.5 cm wide, and 2.5 cm thick. For the sensor housings, we want a maximum of 4 cm for length, width, and thickness. Measuring these will simply involve using a ruler or calipers to find each dimension.

Weight

Since the DAQ is targeted towards competitive riders, minimizing the weight of the entire system is crucial. More weight translates to a slower ride and more effort to ride the bike. Our specification for the entire system is a maximum mass of 500g, which weighs roughly 1.1 lbf. Testing the weight of the system will only require a weight scale, and it does not need to be particularly precise.

Cost

To differentiate from similar products, keeping our product affordable was a key focus. We want our entire system to cost less than \$150. While the current prototype design will be made of the more expensive Formlabs resin, the final version would be made of a cheaper, more mass-producible material and process. We will be able to calculate this directly from our budget. See Appendix B for the entire budget.

Battery Life

The system should be able to operate on a single charge long enough for the rider to get in a day's worth of rides. The DAQ will mainly be used on downhill portions of trails, which makes up a fraction of the entire ride duration. We specified a minimum battery life of one hour. We can estimate the battery life by measuring the current consumed by the device using a multimeter. The battery operates at a (near) fixed voltage and multiplying by current gives the wattage of the device. Battery capacity is given in Watt-hours, which we can divide by the wattage to find roughly how long the batteries will last.

Ingress Protection

While riding, the DAQ will experience somewhat harsh environmental conditions. We defined an ingress protection level of IP54. This means that the system is protected against dust interfering with the DAQ's functionality. It also means that the system will be able to withstand

splashes of water. We chose these specifications because they represent actual conditions that the DAQ might reasonably go through during operation.

To test the ingress protection specification, we will first remove the internal electronics from the DAQ and replace them with paper. Then we will splash the system with water and dust and observe if the paper is wet or any dust entered. If the inside is dry and dust-free, the test is successful.

Foolproof Operation

The system should not be complicated to operate. We will give the verification prototype to various “customers” and provide them with basic operation instructions. If they run into any issues with how to use the DAQ, this will count as an unsuccessful test.

Maximum Recording Storage

The system needs to have enough storage to contain data from multiple rides. We decided on a minimum of 8 gigabytes, which corresponded to well over 20 hours of recording time. Testing this specification will only require examining the SD card used for data storage.

Mounting Universality

The DAQ is not specific to any model of bike and should be able to fit across a range of frame geometries. We will test the system’s universality by trying to mount it on a multitude of bicycles. We can find a variety of bikes either through a biking-related club, a bike shop, or individual personal bikes.

Aesthetics

The appearance of the DAQ system should be attractive to potential customers. This is a subjective criterion which we can test by surveying potential customers. If over 80% of those surveyed agree that the system is visually appealing, this specification is considered met.

Suspension Tuning Recommendation

The overall purpose of this product is to produce suspension tuning recommendations which increase the bike’s performance. To test this, we will ride the bike on a trail with the suspension tuned randomly. After riding, we will adjust the suspension based on the DAQ’s recommendation and test again. We can run this test at different ‘untuned’ configurations to ensure the system works for a range of test cases. If the average speed increase of the bike is 5% or more, this specification will be marked a success.

5.1 Uncertainty Analysis

To ensure the validity of our data, we will need to estimate the uncertainty of the sensors. We will not need the data to be extremely precise in our application, as the vibrations and rotations are going to have high nominal values. The uncertainty of the sensors will likely be negligible

compared to our data. However, we will still analyze the uncertainty of the sensors to be sure any variation in data not attributed to actual vibrations is not significant.

MEMS devices, including the accelerometers and gyroscope in our system, have many sources of error. These are listed on the manufacturer's datasheets, along with typical values of uncertainty for each source of error. Error arises from the construction of the devices, as well as from noise, misalignment, and offset due to temperature.

From the datasheets, the estimated maximum uncertainty of the accelerometers and gyroscope were calculated. The conditions were assumed to be stationary with one axis oriented perpendicular to gravity, and at 25°C, the defined ambient temperature for both manufacturers. Uncertainty was also calculated for every angle to be measured in this calibration test. If the uncertainty at zero is sufficient to characterize the entire range of data, we may use the zero data point to calibrate the system as needed. These calculations can be seen in Appendix C. To compare against these uncertainties, we will take measurements with the devices at different known angles over a period of time. The devices will not be moving, allowing us to examine the fluctuations in readings. This will be compared against the estimated maximum uncertainty at zero. This will give us a basic idea of how the devices are performing, and if they meet the expected uncertainty specifications.

A full description of the test procedures is laid out in Appendix I.

6. Conclusions

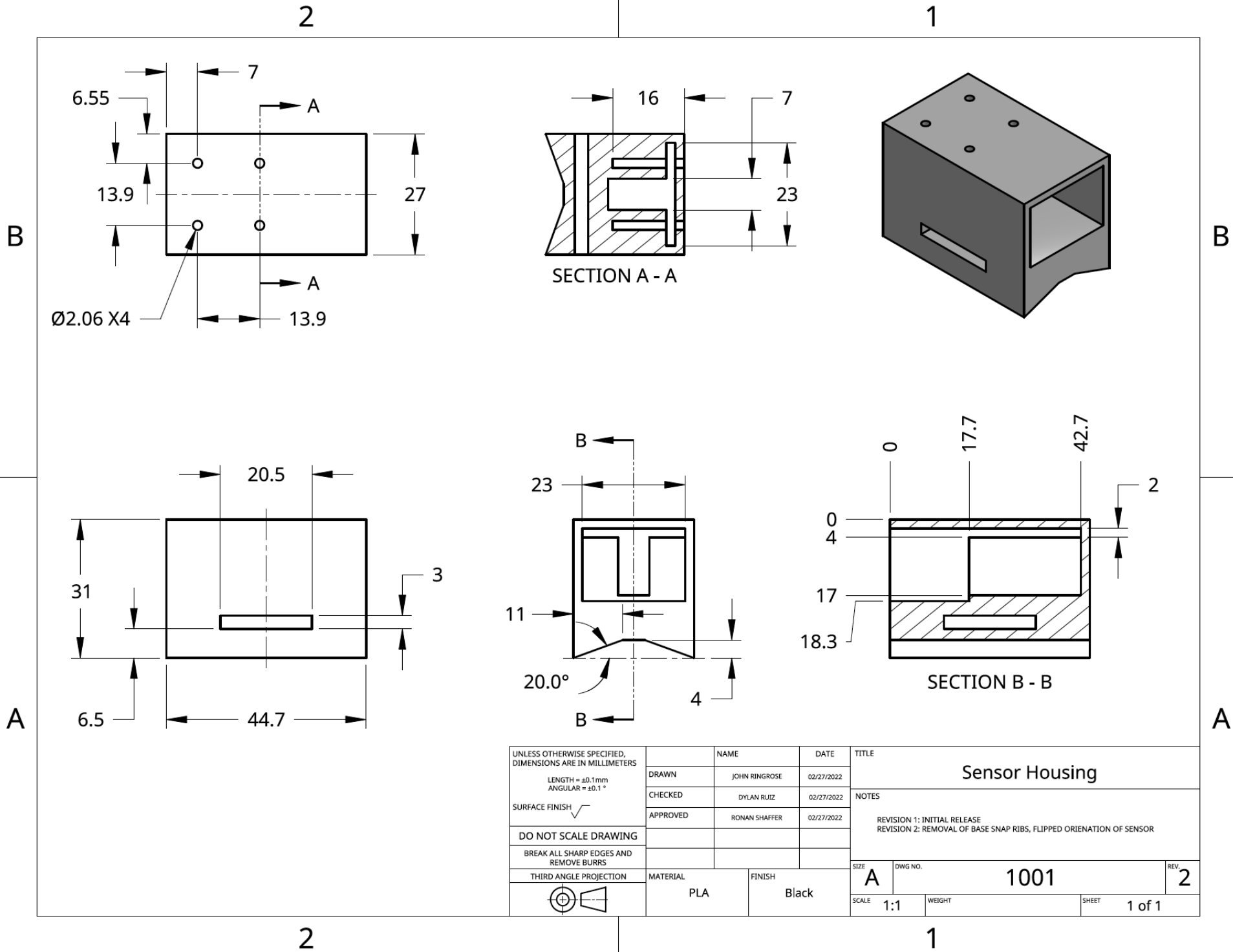
This report documents and presents our progress towards completing the MTB DAQ system. Our team has encountered difficulties with getting the previous design of the data acquisition system to work properly over the past few weeks, making the verification of working sensors and circuit design our team's focus for our structural prototype. Our team verified the collection of accelerometer data through the shake table test and found a potential bias in both accelerometers. The hall effect sensor was tested for usable range and verified circuit design. The gyroscope was unable to be tested due to shipping problems, so the bench test to verify data collection could not be performed.


The next steps include analyzing the lab test data to calibrate the sensors and eliminate the biases, installing the gyroscope to our designed circuit and verify its data collection, and to complete the design of the Main DAQ PCB. With the permission of our sponsor, we will commit to our purchasing, building and test plans.

7. References

- [1] Accelerometer datasheet: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL375.PDF>
- [2] Gyroscope Datasheet: <https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-3050A-00-v2-7.pdf>
- [3] Hall Effect Sensor Datasheet: <https://www.digikey.com/en/htmldatasheets/production/2007903/0/0/1/aps12205luaabu.html?site=US&lang=en&cur=USD>
- [4] Formlabs 3+ Tough 2000 Resin Specifications Table: <https://formlabs-media.formlabs.com/datasheets/2001340-TDS-ENUS-0P.pdf>

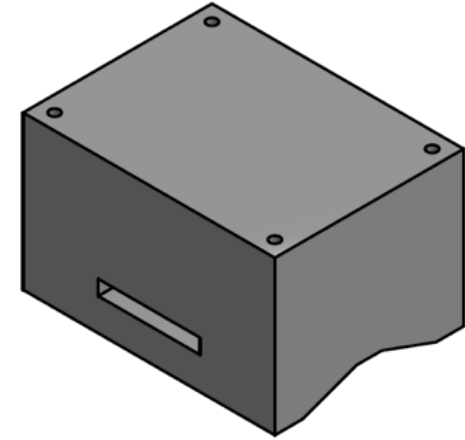
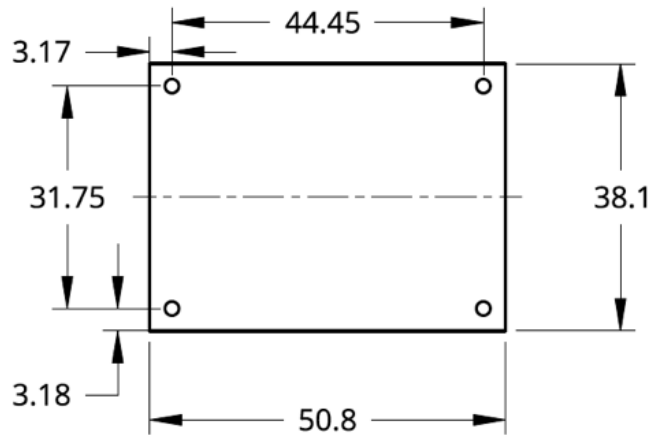
Appendix A. Drawing and Specs Package



UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS LENGTH = $\pm 0.1 \text{ mm}$ ANGULAR = $\pm 0.1^\circ$ SURFACE FINISH $\sqrt{\quad}$ DO NOT SCALE DRAWING BREAK ALL SHARP EDGES AND REMOVE BURRS THIRD ANGLE PROJECTION 	NAME	DATE	TITLE						
	DRAWN	JOHN RINGROSE	02/27/2022	Sensor Housing NOTES REVISION 1: INITIAL RELEASE REVISION 2: REMOVAL OF BASE SNAP RIBS, FLIPPED ORIENTATION OF SENSOR					
	CHECKED	DYLAN RUIZ	02/27/2022						
	APPROVED	RONAN SHAFFER	02/27/2022						
MATERIAL	PLA	FINISH	Black	SIZE	A	DWG. NO.	1001	REV.	2
				SCALE	1:1	WEIGHT	SHEET		1 of 1

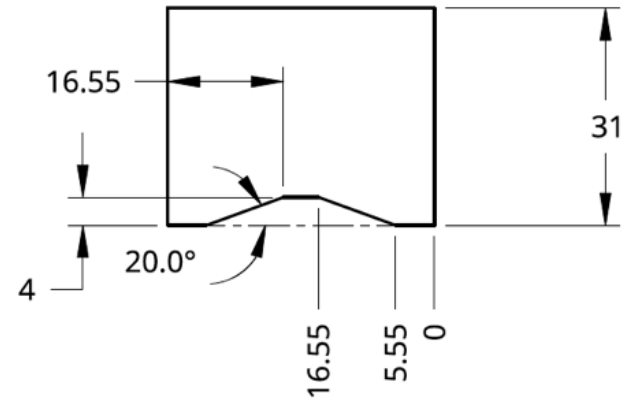
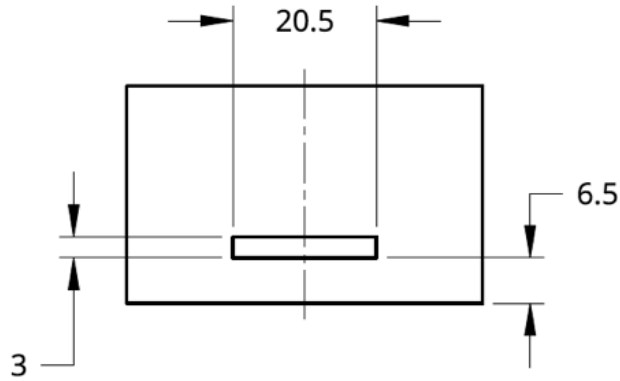
2

1



B

B



A

A

UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS		NAME	DATE	CADENCE PROTO MOUNT				
LENGTH = ±0.1mm ANGULAR = ±0.1°		DRAWN	JOHN RINGROSE	03/01/2022	NOTES REVISION 1: INITIAL RELEASE			
SURFACE FINISH $\sqrt{\quad}$		CHECKED	DYLAN RUIZ	02/27/2022				
DO NOT SCALE DRAWING		APPROVED	RONAN SHAFFER	02/27/2022				
BREAK ALL SHARP EDGES AND REMOVE BURRS					SIZE A	DWG NO. 1004	REV. 1	
THIRD ANGLE PROJECTION		MATERIAL	PLA	FINISH	BLACK	SCALE 1:1	WEIGHT	SHEET 1 of 1

2

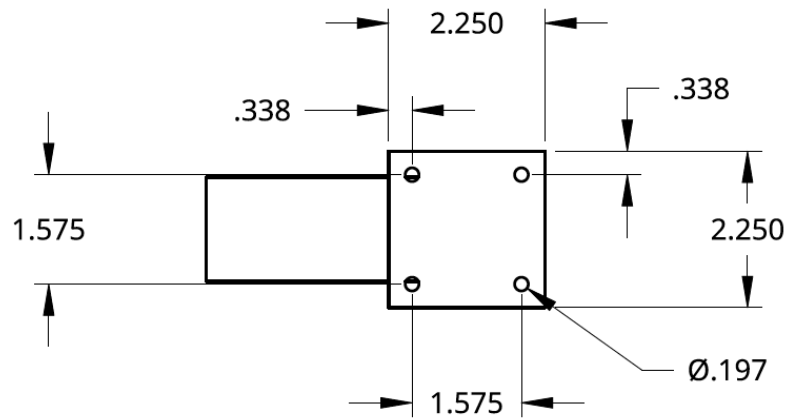
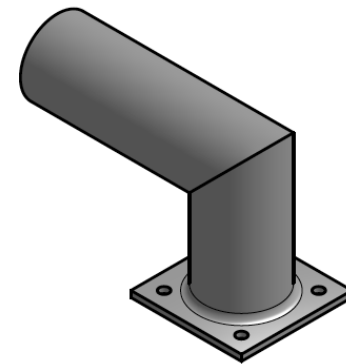
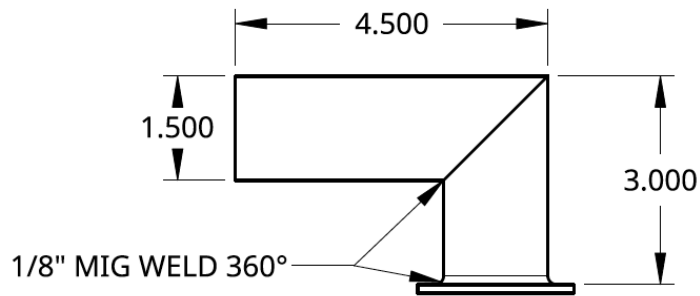
1

2

1

B

B



A

A

UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES ANGULAR = ±1° FRACTIONAL = ±1/32 SURFACE FINISH	NAME	DATE	TITLE	
	DRAWN	JOHN RINGROSE	02/02/2022	HORIZONTAL SHAKE TABLE MOUNT NOTES REVISION 1: INITIAL RELEASE REVISION 2: CHANGE HOLE SPACING TO CORRECT DIM (4mm x 4mm)
	CHECKED	RONAN SHAFFER	02/27/2022	
	APPROVED	DYLAN RUIZ	02/27/2022	
DO NOT SCALE DRAWING	MATERIAL		SIZE	DWG NO.
BREAK ALL SHARP EDGES AND REMOVE BURRS	STEEL		A	1002
THIRD ANGLE PROJECTION	FINISH		SCALE	WEIGHT
	GREY PAINT		1:2.5	
	SHEET			REV.
	1 of 1			2

2

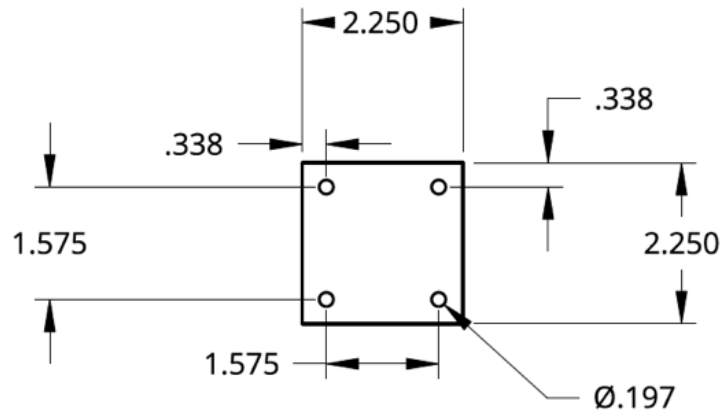
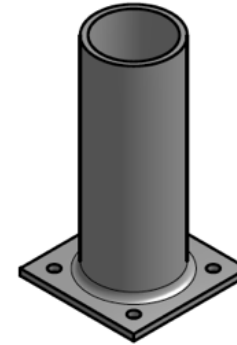
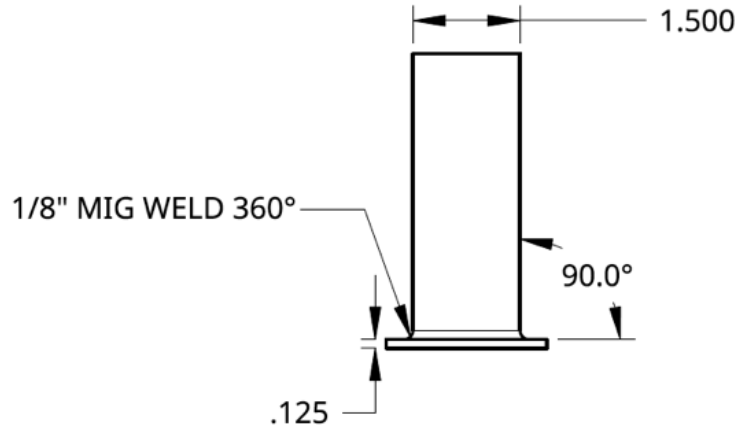
1

2

1

B

B




A

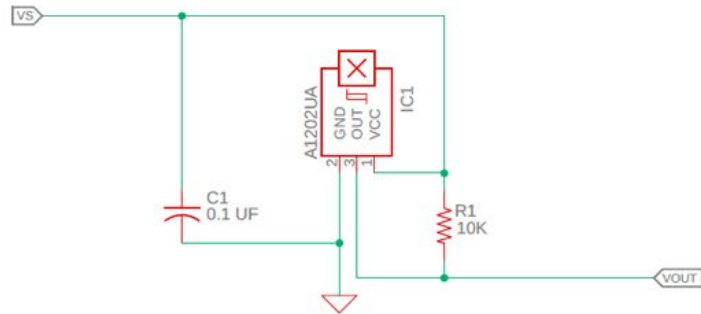
A

2

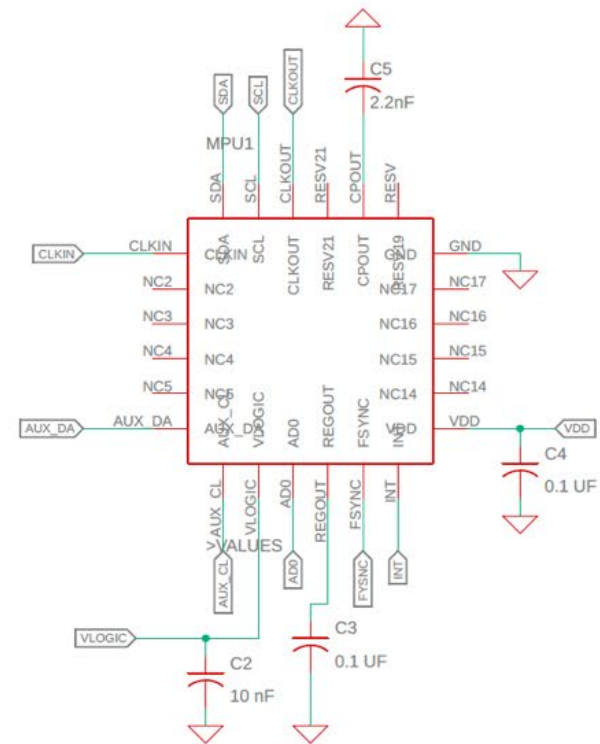
1

UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES ANGULAR = ±1° FRACTIONAL = ±1/32 SURFACE FINISH $\sqrt{\quad}$ DO NOT SCALE DRAWING BREAK ALL SHARP EDGES AND REMOVE BURRS THIRD ANGLE PROJECTION 	NAME		DATE		TITLE	
	DRAWN	JOHN RINGROSE	02/02/2022		VERTICAL SHAKE TABLE MOUNT	
	CHECKED	RONAN SHAFFER	02/27/2022		NOTES	
	APPROVED	DYLAN RUIZ	02/27/2022		REVISION 1: INITIAL RELEASE REVISION 2: CHANGE HOLE SPACING TO CORRECT DIM (4mm x 4mm)	
MATERIAL		FINISH		SIZE	DWG NO.	REV.
STEEL		GREY PAINT		A	1003	2
SCALE				WEIGHT		SHEET
1:2.5						1 of 1

Hall Effect



Gyro



DRAWN BY:	
JOHN RINGROSE	
TITLE: Structural Prototype	
Document Number: 2001	REV: 1
Date: 02/07/2022	Sheet: 1/1

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 2.5\text{ V}$, $V_{DDIO} = 2.5\text{ V}$, acceleration = 0 g, $C_S = 10\text{ }\mu\text{F}$ tantalum, $C_{IO} = 0.1\text{ }\mu\text{F}$, output data rate (ODR) = 800 Hz, unless otherwise noted.

Table 1.

Parameter	Test Conditions/Comments	Min	Typ ¹	Max	Unit
SENSOR INPUT					
Measurement Range ²	Each axis	±180	±200		g
Nonlinearity	Percentage of full scale		±0.25		%
Cross-Axis Sensitivity ³			±2.5		%
SENSITIVITY					
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT} ^{2,4}	Each axis ODR ≤ 800 Hz	18.4	20.5	22.6	LSB/g
Scale Factor at X_{OUT} , Y_{OUT} , Z_{OUT} ^{2,4}	ODR ≤ 800 Hz	44	49	54	mg/LSB
Sensitivity Change Due to Temperature			±0.02		%/°C
0 g OFFSET					
0 g Output for X_{OUT} , Y_{OUT} , Z_{OUT}	Each axis	-6000	±400	+6000	mg
0 g Offset vs. Temperature			±10		mg/°C
NOISE					
	X-, y-, and z-axes		5		mg/√Hz
OUTPUT DATA RATE AND BANDWIDTH⁵					
Output Data Rate (ODR) ^{4,6}	User selectable	0.1		3200	Hz
SELF-TEST⁷					
Output Change in Z-Axis			6.4		g
POWER SUPPLY					
Operating Voltage Range (V_S)		2.0	2.5	3.6	V
Interface Voltage Range (V_{DDIO})		1.7	1.8	V_S	V
Supply Current					
Measurement Mode	ODR ≥ 100 Hz		145		μA
	ODR ≤ 3 Hz		35		μA
Standby Mode			0.1		μA
Turn-On and Wake-Up Time ⁸	ODR = 3200 Hz		1.4		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	°C
WEIGHT					
Device Weight			30		mg

¹ Typical specifications are for at least 68% of the population of parts and are based on the worst case of mean ± 1 σ distribution, except for sensitivity, which represents the target value.

² Minimum and maximum specifications represent the worst case of mean ± 3 σ distribution and are not guaranteed in production.

³ Cross-axis sensitivity is defined as coupling between any two axes.

⁴ The output format for the 1600 Hz and 3200 Hz output data rates is different from the output format for the other output data rates. For more information, see the Data Formatting at Output Data Rates of 3200 Hz and 1600 Hz section.

⁵ Bandwidth is the -3 dB frequency and is half the output data rate: bandwidth = ODR/2.

⁶ Output data rates < 6.25 Hz exhibit additional offset shift with increased temperature.

⁷ Self-test change is defined as the output (g) when the SELF_TEST bit = 1 (DATA_FORMAT register, Address 0x31) minus the output (g) when the SELF_TEST bit = 0. Due to device filtering, the output reaches its final value after $4 \times \tau$ when enabling or disabling self-test, where $\tau = 1/(\text{data rate})$. For the self-test to operate correctly, the part must be in normal power operation (LOW_POWER bit = 0 in the BW_RATE register, Address 0x2C).

⁸ Turn-on and wake-up times are determined by the user-defined bandwidth. At a 100 Hz data rate, the turn-on and wake-up times are each approximately 11.1 ms. For other data rates, the turn-on and wake-up times are each approximately $\tau + 1.1\text{ ms}$, where $\tau = 1/(\text{data rate})$.

3 Electrical Characteristics

3.1 Sensor Specifications

Typical Operating Circuit of Section 4.2, VDD = 2.5 V, VLOGIC = 2.5 V, T_A=25°C.

Parameter	Conditions	Min	Typical	Max	Unit	Notes
GYRO SENSITIVITY						
Full-Scale Range	FS_SEL = 0		±250		%s	4, 7
	FS_SEL = 1		±500			
	FS_SEL = 2		±1000			
	FS_SEL = 3		±2000			
Gyro ADC Word Length			16		Bits	3
Sensitivity Scale Factor	FS_SEL = 0		131		LSB/(°/s)	1
	FS_SEL = 1		65.5			
	FS_SEL = 2		32.8			
	FS_SEL = 3		16.4			
Sensitivity Scale Factor Tolerance	25°C	-6	±2	+6	%	1
Sensitivity Scale Factor Variation Over Temperature	-40°C to +85°C		±2		%	8
Nonlinearity	Best fit straight line; 25°C		0.2		%	6
Cross-Axis Sensitivity			2		%	6
GYRO ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±20		%s	1
ZRO Variation Over Temperature	-40°C to +85°C		±0.15		%/°C	8
Power-Supply Sensitivity (1-10 Hz)	Sine wave, 100mVpp; VDD = 2.2 V		0.2		%s	5
Power-Supply Sensitivity (10 – 250 Hz)	Sine wave, 100mVpp; VDD = 2.2 V		0.2		%s	5
Power-Supply Sensitivity (250 Hz – 100 kHz)	Sine wave, 100mVpp; VDD = 2.2 V		4		%s	5
Linear Acceleration Sensitivity	Static		0.1		%s/g	6
GYRO NOISE PERFORMANCE						
Total RMS Noise	FS_SEL=0 DLPFCFG = 2 (100 Hz)		0.1		%s-rms	1
Low-frequency RMS noise	Bandwidth 1 Hz to 10 Hz		0.033		%s-rms	1
Rate Noise Spectral Density	At 10 Hz		0.01		%s/√Hz	3
GYRO MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	1
Y-Axis		27	30	33	kHz	1
Z-Axis		24	27	30	kHz	1
GYRO START-UP TIME						
ZRO Settling	DLPFCFG = 0 to ±1%/s of Final		50		ms	5
TEMPERATURE RANGE						
Specified Temperature Range		-40		85	°C	2

Notes:

1. Tested in production
2. Based on characterization of 30 parts over temperature on evaluation board or in socket
3. Based on design, through modeling and simulation across PVT
4. Typical. Randomly selected part measured at room temperature on evaluation board or in socket
5. Based on characterization of 5 parts over temperature
6. Tested on 20 parts at room temperature
7. Part is characterized to Full-Scale Range. Maximum ADC output is $2^{16} / (\text{Sensitivity} \times 2)$
Example: For Sensitivity of 131 LSB/(°/s), $[2^{16} / (131 \times 2)] = \pm 250$ °/s.
8. Based on characterization of 48 parts on evaluation board or in socket

**APS12205,
APS12215,
and APS12235**

**High-Temperature Hall-Effect Latches
for Low Voltage Applications**

ELECTRICAL CHARACTERISTICS: Valid over full operating voltage and ambient temperature range, unless otherwise noted

Characteristics	Symbol	Test Conditions	Min.	Typ. ^[1]	Max.	Unit ^[2]
ELECTRICAL CHARACTERISTICS						
Forward Supply Voltage	V_{CC}	Operating, $T_J < 175^\circ\text{C}$	2.8	–	5.5	V
Supply Current	I_{CC}	$V_{CC} = 5.5\text{ V}$	–	2	4	mA
Output Leakage Current	I_{OUTOFF}	$V_{OUT} = 5.5\text{ V}$, $B < B_{RP}$	–	–	10	μA
Output Saturation Voltage	$V_{OUT(SAT)}$	$I_{OUT} = 5\text{ mA}$, $B > B_{OP}$	–	50	500	mV
Output Current	I_{OUT}	Recommended value used during characterization	–	5	–	mA
Output Short-Circuit Current Limit	I_{OM}	$B > B_{OP}$	30	–	60	mA
Power-On Time ^[3]	t_{ON}	$V_{CC} \geq 2.8\text{ V}$, $B < B_{RP(\text{min})} - 10\text{ G}$, $B > B_{OP(\text{max})} + 10\text{ G}$	–	–	25	μs
Power-On State, Output ^[3]	POS	$V_{CC} \geq V_{CC(\text{min})}$, $t < t_{ON}$	Low			–
Chopping Frequency	f_C		–	800	–	kHz
Output Rise Time ^{[3][4]}	t_r	$R_{PULL-UP} = 1\text{ k}\Omega$, $C_L = 20\text{ pF}$	–	0.2	2	μs
Output Fall Time ^{[3][4]}	t_f	$R_{PULL-UP} = 1\text{ k}\Omega$, $C_L = 20\text{ pF}$	–	0.1	2	μs
MAGNETIC CHARACTERISTICS						
Operate Point	B_{OP}	APS12205	5	22	40	G
		APS12215	15	50	90	G
		APS12235	100	150	180	G
Release Point	B_{RP}	APS12205	–40	–22	–5	G
		APS12215	–90	–50	–15	G
		APS12235	–180	–150	–100	G
Hysteresis	B_{HYS}	APS12205	10	45	80	G
		APS12215	30	100	180	G
		APS12235	200	300	360	G

^[1] Typical data are at $T_A = 25^\circ\text{C}$ and $V_{CC} = 5\text{ V}$, and are for initial design estimations only.

^[2] 1 G (gauss) = 0.1 mT (millitesla).

^[3] Guaranteed by device design and characterization.

^[4] C_L = oscilloscope probe capacitance.

Tough 2000 Resin Material Properties Data

	METRIC ¹		IMPERIAL ¹		METHOD
	Green ²	Post-Cured ³	Green ²	Post-Cured ³	
Mechanical Properties					
Ultimate Tensile Strength	29 MPa	46 MPa	4206 psi	6671 psi	ASTM D 638-14
Tensile Modulus	1.2 GPa	2.2 GPa	174 ksi	329 ksi	ASTM D 638-14
Elongation at Break	74 %	48 %	74 %	48 %	ASTM D 638-14
Flexural Properties					
Flexural Strength	17 MPa	65 MPa	2465 psi	9427 psi	ASTM D 790-15
Flexural Modulus	0.45 GPa	1.9 GPa	65 ksi	275 ksi	ASTM D 790-15
Impact Properties					
Notched IZOD	79 J/m	40 J/m	1.5 ft-lbf/in	0.75 ft-lbf/in	ASTM D256-10
Unnotched IZOD	208 J/m	715 J/m	3.9 ft-lbf/in	13 ft-lbf/in	ASTM D256-10
Thermal Properties					
Heat Deflection Temp. @ 1.8 MPa	42 °C	53 °C	108 °F	127 °F	ASTM D 648-16
Heat Deflection Temp. @ 0.45 MPa	48 °C	63 °C	118 °F	145 °F	ASTM D 648-16
Coefficient of Thermal Expansion	107 µm/m/°C	91 µm/m/°C	59 µin/in/°F	50 µin/in/°F	ASTM E 831-13

¹Material properties can vary with part geometry, print orientation, print settings, and temperature.

²Data was obtained from green parts, printed using Form 2, 100 µm, Tough settings, washed and air dried without post cure.

³Data was obtained from parts printed using Form 2, 100 µm, Tough 2000 settings, and post-cured with a Form Cure for 120 minutes at 80 °C.

Appendix B.1 Project Budget

Vendor (name, website, phone, or fax)	Product Name (paste the exact product title, include all text)	Part Number	Qty	Price/Ea	Total	Design Location	Payment	Date Purchased	Currently Located
Digikey	MPU-3050	1428-1001-1-ND - Cut Tape (CT)	2	\$ 8.26	\$ 16.52	Main DAQ	Reimburse	2/3/2022	In Hand
Digikey	APS12205LUAA	620-1964-ND	2	\$ 0.98	\$ 1.96	Fork Sensor	Reimburse	2/3/2022	In Hand
Digikey	LP402535JU+PCM+2 WIRES 50MM	1908- LP402535JU+PCM+2WIRES50MM- ND	8	\$ 9.49	\$ 75.92	Main DAQ	Reimburse	2/3/2022	In Hand
Digikey	BATTERY LITHIUM 3.7V 1.2AH	1528-1838-ND	2	\$ 9.95	\$ 19.90	Main DAQ	Reimburse	2/3/2022	In Hand
Arrow	Accelerometer Triple ±200g 2.5V/3.3V 14-Pin LGA T/R	ADXL375BCCZ-RL7	2	\$ 11.24	\$ 22.48	Main DAQ/Front/Rear	Reimburse	2/3/2022	In Hand
Jenson USA	ONEUP COMPONENTS EDC GEAR STRAP	TL186J05	2	\$ 16.50	\$ 33.00	Fork/Rear Sensor	Reimburse	2/3/2022	In Hand
My Bike Shop	MSW Universal Speed Sensor Spoke Magnet	EC3311	2	\$ 5.00	\$ 10.00	Front Wheel Spoke	Reimburse	2/3/2022	In Hand
JLC PCB	Sensor PCB	Custom	2	-	\$ 34.00	Fork Sensor	Reimburse		
JLC PCB	Main DAQ PCB	Custom	2	-	\$ 54.00	Main DAQ	Reimburse		
FormLabs	Housings	Custom	4	-	\$ 100.00	Main DAQ/Sensors	Reimburse		
Shipping/Handling/Tax					\$ 97.00				
Total					\$ 464.78				
Project Budget					\$ 500.00				
Budget Remaining					\$ 35.22				

Appendix B.2: Bill of Materials

MTB DAQ											
Indented Bill of Material (iBOM)											
Assy Level	Part Number	Descriptive Part Name				Qty	Part Cost	Source	URL	More Info	
		<i>Lvl0</i>	<i>Lvl1</i>	<i>Lvl2</i>	<i>Lvl3</i>	<i>Lvl4</i>					
0	100000	Final Assy						-----			
1	110000	Main DAQ						-----			
2	111000		Housing			1	\$ 1.50	custom		Maintenance	
2	111100			Bolts		1	\$ 0.59	McMaster		item 45792A	
2	111400		PCB						-----		
3	111410			Gyroscope		1	\$ 1.50	custom		vac-formed PET	
3	111420			Accelerometer		1	\$ 2.15	McMaster		item 98725	
3	111430			Nucleo		1	\$ 0.30	McMaster		item 48005	
				RJ45 Ports							
				Batteries		4					
2	111600		Cables						-----		
3	111610			Ethernet Cables		2	\$ 0.35	McMaster		item 48250	
3	111620		Rear Sensor						-----		
4	111621			Housing		1	\$ 1.50	custom		mold in ABS	
4	111622			Strap		1	\$ 0.50	Bearing Inc.		item 27-100	
2	111700			Screws				-----			
3	111710			Accelerometer		1	\$ 7.00	custom		machined aluminum	
3	111720			PCB		1	\$ 2.50	Bendix		item US259874	
1	120000		Front Sensor				1	\$ 1.50	custom		mold in ABS
				Housing							
	TL186J05			Strap		1	\$ 16.50	Oneup	Link	OTS	
				Screws							
				Accelerometer		1					
1				Hall effect sensor			\$ 0.98	DigiKey	Link	OTS	
1	207107			Spoke magnet		1	\$ 10.40	REI	Link	OTS	
				PCB							
1	130000					4	\$ 0.72	Home Depot		#3-1/2-in	
Total Parts						19	\$ 47.99				

Appendix C. Uncertainty Propagation

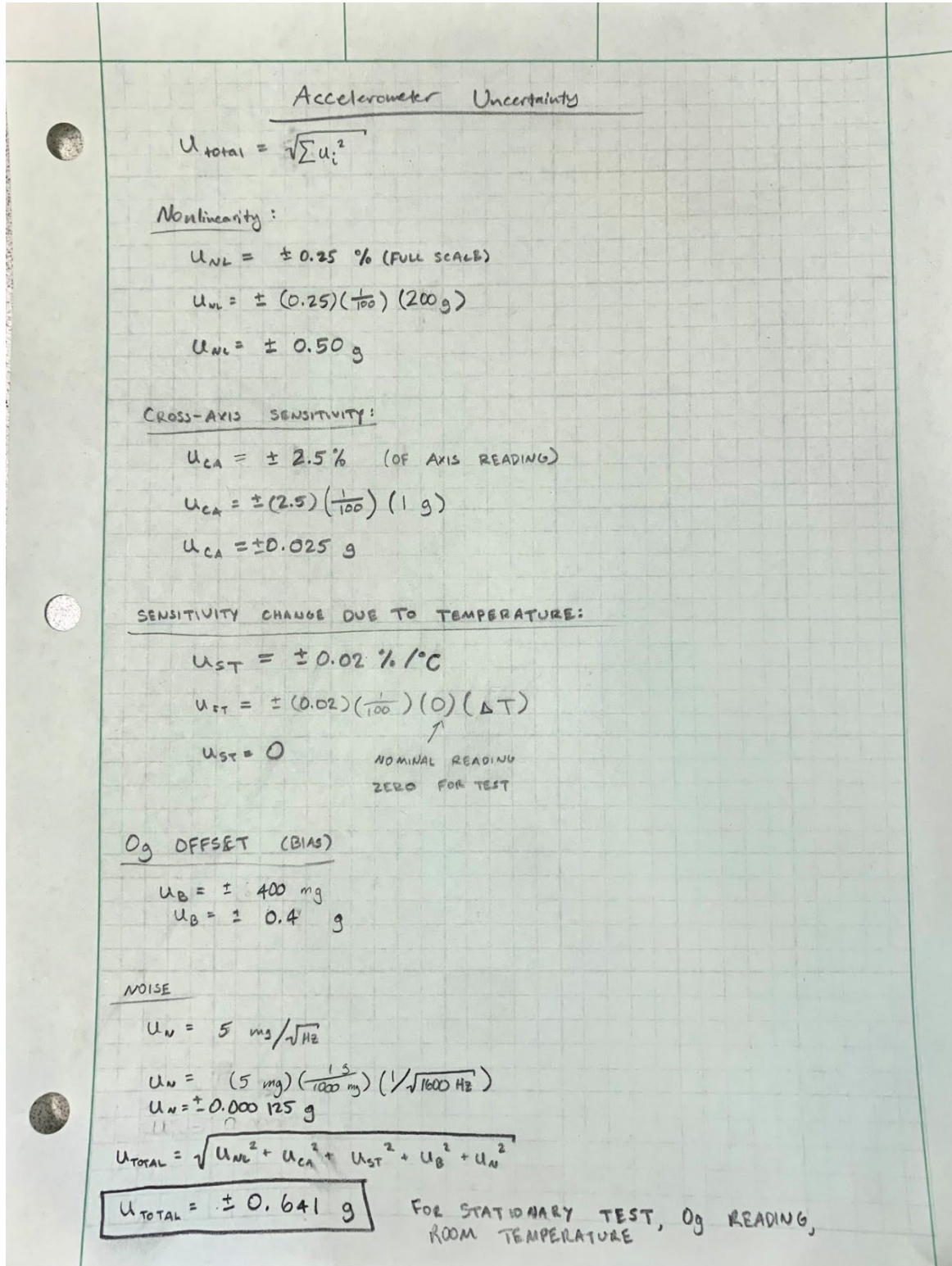


Fig. Uncertainty estimation for stationary accelerometer, with axis oriented along 0g axis, at ambient temperature (25 °C). Uncertainty values from manufacturer's datasheet [Appendix A].

GYROSCOPE UNCERTAINTY

INITIAL ZERO TOLERANCE

$$u_z = \pm 20 \%$$

POWER SUPPLY SENSITIVITY (250 Hz - 100 kHz)

$$u_{ps} = \pm 4 \%$$

LINEAR ACCELERATION SENSITIVITY

$$u_{LA} = \pm (0.1 \text{ } \%/g) (1g)$$

$$u_{LA} = \pm 0.1 \%$$

RESOLUTION

$$u_{res} = (1 \text{ LSB}) \left(\frac{(1 \text{ } \%/s)}{(16.4 \text{ LSB})} \right)$$

$$u_{res} = 0.061 \%$$

TOTAL UNCERTAINTY (OF STATIC TEST:)

$$u_{TOTAL} = \sqrt{u_z^2 + u_{ps}^2 + u_{LA}^2 + u_{res}^2}$$

$$u_{TOTAL} = \sqrt{(20 \%)^2 + (4 \%)^2 + (0.1 \%)^2 + (0.061 \%)^2}$$

$$u_{TOTAL} = \pm 20.40 \%$$

Fig. Uncertainty estimation for stationary accelerometer at ambient temperature (25 °C).
Uncertainty values from manufacturer's datasheet [Appendix A].

Appendix D. FMEA

Team F11											Action Results				
System / Function	Potential Failure Mode	Potential Effects of the Failure Mode	Severity	Potential Causes of the Failure Mode	Current Preventative Activities	Occurrence	Current Detection Activities	Detection	Priority	Recommended Action(s)	Responsibility & Target Completion Date	Actions Taken	Severity	Occurrence	Criticality
Algorithm / Interpret Data	Incorrect Interpretation of Data	1. Uncomfortable Rider 2. Slower Trail Time	7	1. Improper filter of Sensor data 2. Irrelevant Metric 3. Inaccurate data	1. Industry & Technical Research 2. Iterative implementation, testing and redesign	7	Comparative testing between run times	8	1	Iterate through testing and editing of algorithm parameters until they result in real-world improvements in performance.	Entire team. Final algorithm: April 18				
Main DAQ / Store Data	Unusable data	No tuning recommendation	8	1. SD card full 2. SD card not formatted properly 3. Data corrupted 4. SD card damaged	1. Design & order PCB from manufacturer 2. Test EE components	3	1. Test System for maximum run time 2. Start-up alert if SD error	3	4	Main DAQ redesign (along with UI)	Entire team. Target completion March 7.				
Sensors / Measure Data	Inaccurate measurement	1. Uncomfortable Rider 2. Slower Trail Time	6	1. Sensor miscalibration 2. Too fast sampling rate 3. Too slow sampling rate 4. Sensor damage	1. Robust autocalibration software design 2. Design & order PCB from manufacturer 3. Test EE components 4. Iterative implementation, testing and redesign	4	1. Calibrate on testing setup with known data 2. Trail testing 3. Post-test: data analysis	7	2	Create test setups to compare data to known measurements (i.e. accelerometer angles). Develop procedure to automatically calibrate sensors on bike. Validate data with trail testing and analysis.	Theo. Target completion January 25.	Performed Shake Table Lab Test February 22.			
Housings / Protect DAQ & Sensors	Sensor/PCB damage	1. Uncomfortable Rider 2. Slower Trail Time	7	1. Water ingress 2. Dust/debris ingress 3. Destructive impact	1. Industry & Technical Research 2. Iterative implementation, testing and redesign 3. Purchase or design waterproof connection point	6	System testing	6	5	Iterate through stress testing and adjusting geometry of housing to improve.	Dylan. Initial waterproofing concepts January 27.				
Main DAQ / User Interface	Unresponsive UI	Difficulty operating	6	1. Button sticks 2. Button connection broken 3. LED breaks 4. Character display breaks	1. Industry & Technical Research 2. Iterative implementation, testing and redesign	3	System testing by design team and users	3	8	Main DAQ redesign including button UI for the time being.	Whole team is responsible, March 7th Target Completion				
Main DAQ / Power DAQ	Improper power supply	Unresponsive system	7	1. Battery damaged 2. Battery supplies too high voltage 3. Battery supplies too low voltage 4. Circuit disruption	1. Industry & Technical Research 2. Iterative implementation, testing and redesign	8	System testing	4	3	Research/explore issues with power supply in current DAQ, and design new board to fix them.	Max. Target completion: March 7th.	Researched potential battery replacement for future design of Main DAQ.			
Housings / Mount Sensors & DAQ	Sensor moves from desired position	1. Bad tuning recommendation 2. Interference with rider's motion	6	1. Mount slips on bike 2. Mount detaches from bike 3. Mount interferes with rider's motion	Trail testing	6	Trail testing	7	6	Attach housings to bike, test on trail, and observe any movement. Adjust design if needed.	Ronan. Target completion: January 7.	Attached housings to bike. Rode trails with them attached.			
Housings / Mount Sensors & DAQ	Sensor / Housing movement independent of bike	1. Uncomfortable Rider 2. Slower Trail Time	6	1. Rubber Pad Dampens Vibrations 2. Strap to bike too loose. 3. Screws fixing sensors to housing insufficient	1. Trail testing 2. Lab testing	4	Trail testing	7	7	Mount accelerometers to a shake table in lab, observe results. Adjust design if needed.	Ronan. Target completion: Feb 22.	Shake Table test conducted.			
Cables / Data Transmission	Connection Loss / Cable Damage	Unresponsive system	6	1. Connection is disrupted (wires disconnect, are damaged, etc.) 2. Cable is damaged during crash	1. Secure design of Ethernet cable snap 2. Strapping cables to frame out of way 3. System start-up connection error alert	3	System start up connection error test	7	9	Implement cable lock, not within the scope of our project	Stretch goal, only work on it if all other aspects within scope are completed.				

Appendix E. Design Verification Plan

DVP&R - Design Verification Plan (& Report)											
Project:	F11 - MTB DAQ	Sponsor:	Dr. Joseph Mello					Edit Date: 2/18/2022			
TEST PLAN								TEST RESULTS			
Test #	Specification	Test Description	Measurements	Acceptance Criteria	Required Facilities/Equipment	Parts Needed	Responsibility	TIMING		Numerical Results	Notes on Testing
								Start date	Finish date		
1	Main Hub Size	Measure physical dimensions of main hub.	Lengths	5"x3"x1" or less	Calipers or ruler	SP/FP	Theo	4/18/2022			
2	Sensor Housings Size	Measure physical dimensions of peripheral sensor housings.	Lengths	1.5"x1.5"x1.5" or less	Calipers or ruler	SP/FP	Theo	4/18/2022			
3	Weight	Weigh entire system (hub, sensors, cables, straps) on scale.	Mass	500g or less	Weight Scale	SP/FP	Ronan	4/18/2022			
4	Cost	Add up entire cost of final system	Dollars	Under \$150	None	None	Ronan	4/18/2022			
5	Battery Life	Turn on and run system for target battery life, see if it runs out of power.	Hours	1 Hour or more	None	FP	Dylan	4/18/2022			
6	Ingress Protection	Remove internal electronics from housings and replace with paper. Spray with moderate amount of water, and toss dust at system. See if either has penetrated housings.	Pass/Fail	No water or dust in system	Water, dust	FP	Dylan	4/18/2022			
7	Foolproof	Give system to users with provided manual/instructions, see if they run into any issues.	Pass/Fail	100% pass by user testing (no issues)	Customer Survey	FP	Max	4/18/2022			
8	Maximum Recording Storage	Check maximum storage capacity of SD card	Gigabytes	8 gb or more	None	SP	Max	4/18/2022			
9	Mounting Universality	Attempt to attach system to variety of bikes.	Pass/Fail	System fits on 100% of bikes	Variety of bikes	SP/FP	Theo	4/18/2022			
10	Aesthetics	Survey potential customers, asking if they find the system visually appealing	Pass/Fail	Over 80% Approval	Customer Survey	FP	Ronan	4/18/2022			
11	Suspension Tuning Recommendation	Test the system on a mountain biking trail, adjust according to tuning recommendations, and ride again.	Trail Time	Over 5% faster	Bike trail, bike	FP	Max	4/18/2022			

Appendix G. Gantt Chart

F11 MTB DAQ

Problem Definition

- Choose Project
- Meet Team
- email sponsor

Customer/Need Research

- Interview Sponsor

Research technical issues

- Identify technical challenges

Find journal articles

- Find Modeling articles
- Find Acceleration Drift articles
- Find Testing Apparatus articles
- Find Figures of Merit articles

Product Research

- Search online for current products
- Search patents for similar produc...
- Find product reviews

Interview stakeholders

- Round 1 Interviews with Bike club...
- Round 2 Interviews Google Doc

- Capture Customer Needs/Wants

- Write Problem Statement

- Create Initial Project Plan

- Perform QFD

- Create Specification Table

- Write Specification Descriptions

Write Scope of Work

- Scope of Work (SOW)
- Deliver SOW to Mello
- Implement SOW feedback

Concept Generation & Selection

Ideation

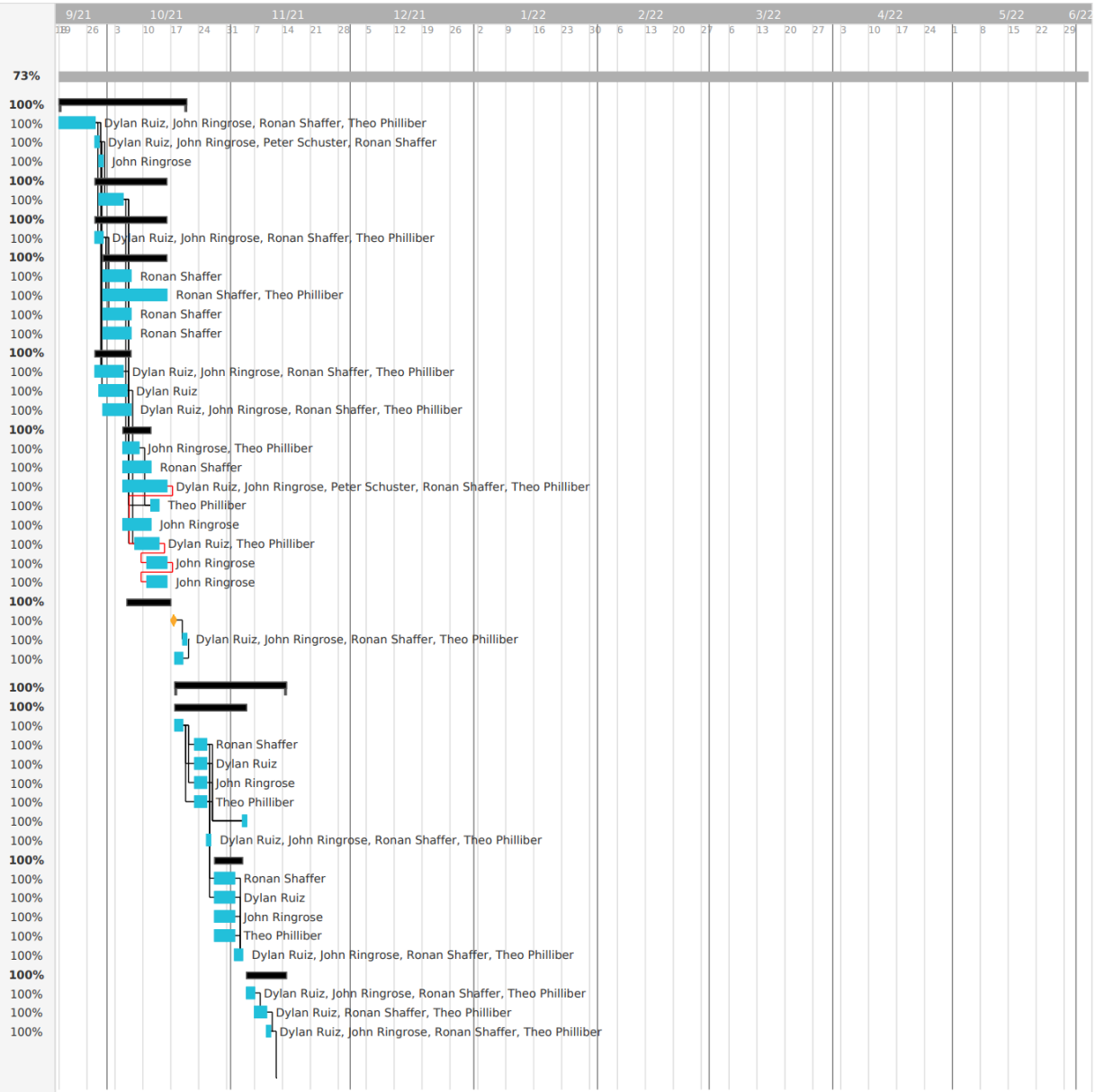
- Brainstorming
- Mounting
- User Interface
- Sensors/Placements
- Metrics
- Share Ideation Conclusions w/ Mello
- Concept Models

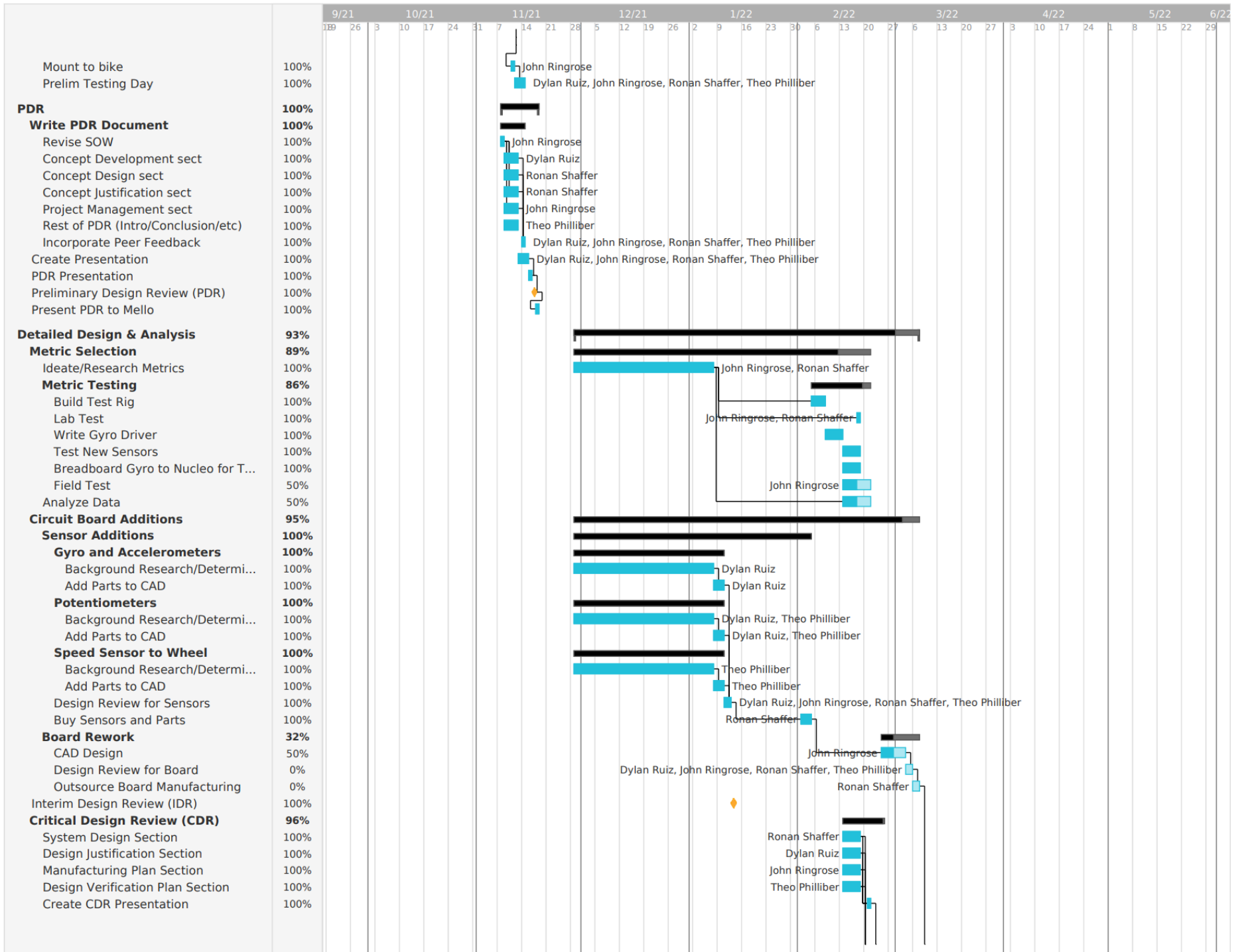
Controlled Convergence

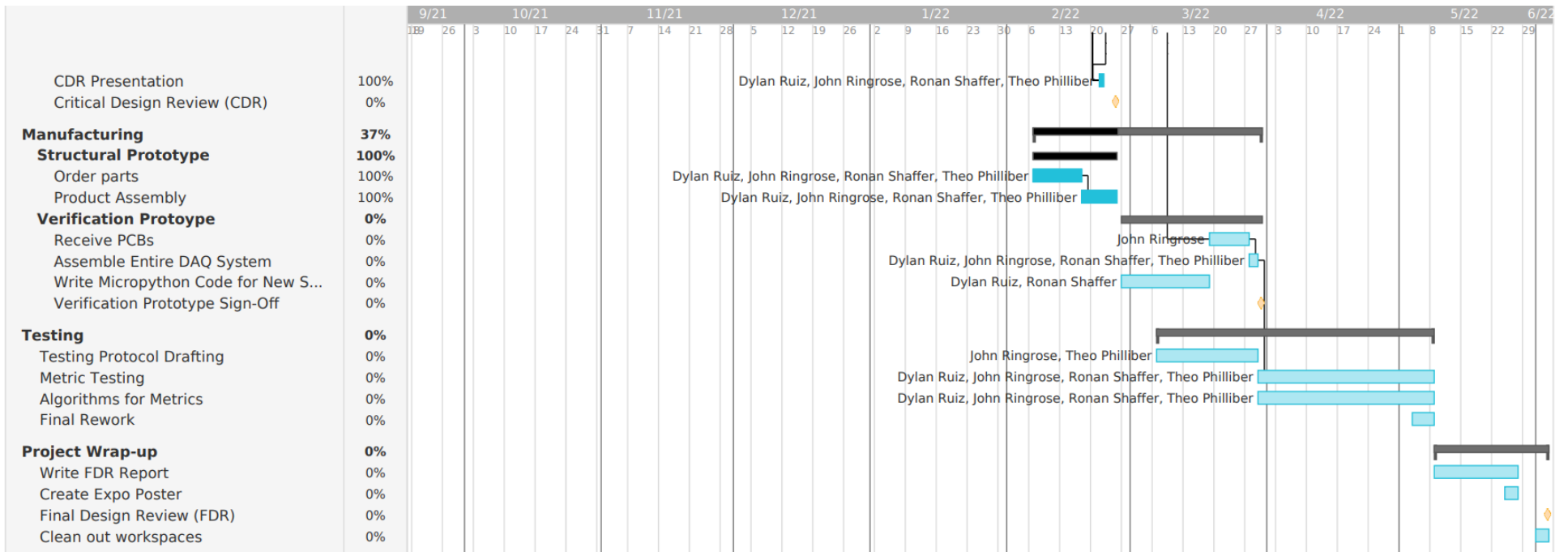
- Pugh Matrix: Mount to Bike
- Pugh Matrix: User Interface
- Pugh Matrix: Sensor Housing
- Pugh Matrix: Speed Measurement
- Weighted Decision Matrix

Preliminary Testing

- CAD Planning for Mounting
- CAD Mounting Prototype
- Manufacture Preliminary Mounting







Appendix H. Design Hazard Checklist

CDR Design Hazard Checklist

F11 MTB DAQ

Y	N	
☐	☐	1. 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?
☐	☐	2. Can any part of the design undergo high accelerations/decelerations?
☐	☐	3. Will the system have any large moving masses or large forces?
☐	☐	4. Will the system produce a projectile?
☐	☐	5. Would it be possible for the system to fall under gravity creating injury?
☐	☐	6. Will a user be exposed to overhanging weights as part of the design?
☐	☐	7. Will the system have any sharp edges?
☐	☐	8. Will any part of the electrical systems not be grounded?
☐	☐	9. Will there be any large batteries or electrical voltage in the system above 40 V?
☐	☐	10. Will there be any stored energy in the system such as batteries, flywheels, hanging weights or pressurized fluids?
☐	☐	11. Will there be any explosive or flammable liquids, gases, or dust fuel as part of the system?
☐	☐	12. Will the user of the design be required to exert any abnormal effort or physical posture during the use of the design?
☐	☐	13. Will there be any materials known to be hazardous to humans involved in either the design or the manufacturing of the design?
☐	☐	14. Can the system generate high levels of noise?
☐	☐	15. Will the device/system be exposed to extreme environmental conditions such as fog, humidity, cold, high temperatures, etc.?
☐	☐	16. Is it possible for the system to be used in an unsafe manner?
☐	☐	17. Will there be any other potential hazards not listed above? If yes, please explain on the reverse.

Description of Hazard	Planned Corrective Action	Planned Date	Actual Date
The design will undergo high accelerations based on the way the user of the design rides the bike the design is attached to.	When testing, we will have an experienced rider wear safety protection while being mindful of riding the bike safely.	11/20/2021	11/20/2021
The system itself will not be large in mass, but it is attached to a bike that will be moving fast. A fast-moving bike can be a hazard to spectators.	When testing, we will spectate the rider from a safe place. We will have a specified segment the rider will take when testing, allowing us to know the path the rider will take.	4/08/2022	
There is currently a battery within the main DAQ system.	Currently, this hazard is low-risk due to the housing of the main DAQ providing protection from the electrical components.	11/20/2021	11/20/2021
The user will have to be riding a mounting bike to use this design.	There will be a cautionary notice before the use of the device listing this hazard. Since this hazard is not affected by our design, this is the most we can do	1/11/2022	
The manufacturing process will include PCB rework. There are hazards with the tools used such as a solder.	The people manufacturing will be trained in safety precautions before operating the tools.	2/19/2022	2/19/2022

Appendix I. Accelerometer Calibration Test Description

Theo Philliber, Dylan Ruiz

Max Ringrose, Ronan Shaffer

2/28/2022

Experimental Planning Worksheet

Project Title: MTB DAQ (F11)

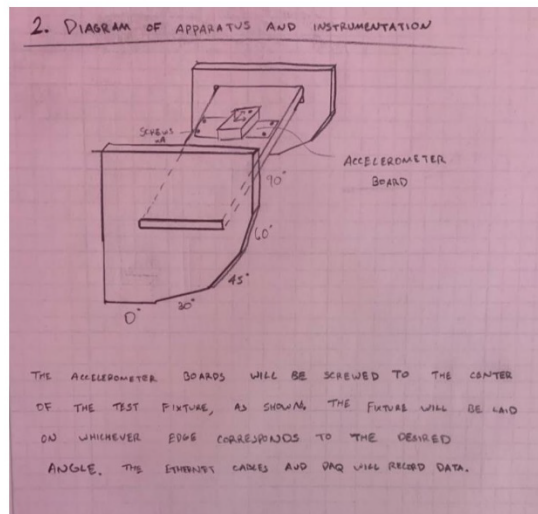
Description:

This preliminary experiment will be to test the performance of the accelerometers. We will take data from each accelerometer at known angular orientations and compare the expected output to the actual outputs. This will give us an idea of the biases and errors present in each sensor and allows us to determine if we need to calibrate or adjust each accelerometer in the firmware.

1. *Desired Results with Required Uncertainty*

The results of this test will be accelerometer data. The raw output of each sensor is in bits, which are then multiplied by a sensitivity (g/LSB) provided by the manufacturer. The manufacturer also provides some uncertainty values in the specifications. From these, we can determine the expected value and uncertainty at each angular setpoint, seeing if our data falls within that range. Our data will have its own uncertainty, caused by the resolution error of the known angles and statistical uncertainty.

2. *Diagram of apparatus and instrumentation*



3. Priority list of measurements to be undertaken

The only measurement taken will be the acceleration data of each sensor at each angular orientation.

4. Schedule Including calibration, zero/tare, baseline, repeats

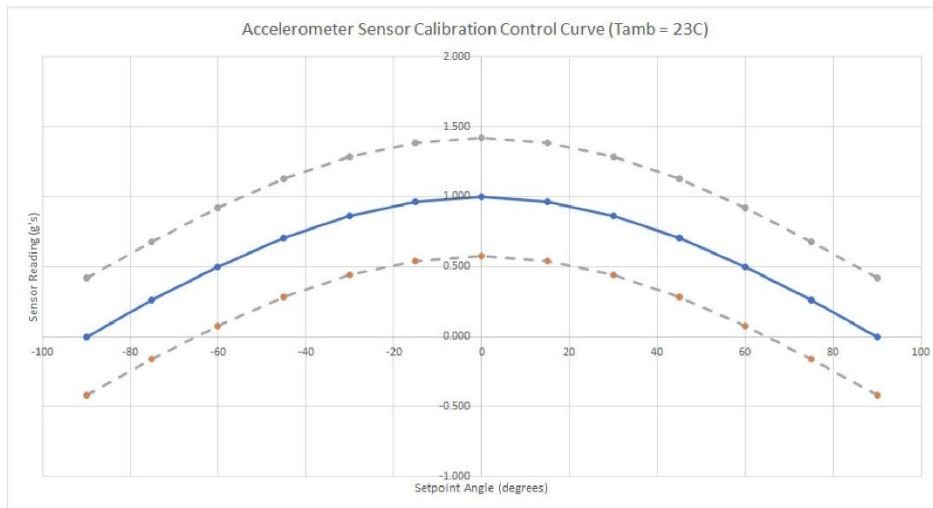
First the fixture will be oriented at 0 degrees. The sensor will be allowed a few seconds (around 5-10) to settle, after which 10 readings will be taken across a period of 10 seconds. Next, the fixture will be moved to 15 degrees, and the readings will be taken. This continues for all angles, positive and negative, and the entire process is repeated once more. The whole process is repeated for all sensors.

5. Data analysis equations/spreadsheet with uncertainty

Angular Setpoint [°]	Uncertainty (+/-) [g]						Experimental Results						
	Expected Reading [°]	Nonlinearity	Temperature (Bias and Sensitivity)	Qg Offset	Noise	Total +/- g	Min [g]	Max [g]	Average Reading [g]	Standard Deviation [g]	Statistical Uncertainty (95%)	Angle Resolution	Pass/Fail
-90	0.000	0.0000	0.02000	0.4	0.000125	0.42013	-0.42013	0.42013					
-75	0.259	0.0006	0.02010	0.4	0.000125	0.42088	-0.16206	0.67969					
-60	0.500	0.0013	0.02020	0.4	0.000125	0.42158	0.07843	0.92158					
-45	0.707	0.0018	0.02028	0.4	0.000125	0.42218	0.28493	1.12928					
-30	0.866	0.0022	0.02035	0.4	0.000125	0.42264	0.44339	1.28866					
-15	0.966	0.0024	0.02039	0.4	0.000125	0.42293	0.54300	1.38885					
0	1.000	0.0025	0.02040	0.4	0.000125	0.42303	0.57698	1.42303					
15	0.966	0.0024	0.02039	0.4	0.000125	0.42293	0.54300	1.38885					
30	0.866	0.0022	0.02035	0.4	0.000125	0.42264	0.44339	1.28866					
45	0.707	0.0018	0.02028	0.4	0.000125	0.42218	0.28493	1.12928					
60	0.500	0.0013	0.02020	0.4	0.000125	0.42158	0.07843	0.92158					
75	0.259	0.0006	0.02010	0.4	0.000125	0.42088	-0.16206	0.67969					
90	0.000	0.0000	0.02000	0.4	0.000125	0.42013	-0.42013	0.42013					

Temperature [°C]: 23
 (example to show realistic uncertainty values)

6. Expected results (control curves)



MTB Suspension Tuning DAQ

Final Design Review (FDR)

By: Dylan Ruiz

dyruiz@calpoly.edu

Ronan Shaffer

rmshaffe@calpoly.edu

John Ringrose

jringros@calpoly.edu

Theo Philliber

phillibe@calpoly.edu

June 6th, 2022

Sponsor:

Professor Joseph Mello

Mechanical Engineering Department
California Polytechnic State University
San Luis Obispo

Table of Contents

- 1. Design Updates 1
- 2. Manufacturing..... 3
 - 2.1 Procurement 3
 - 2.2 Outsourcing & Manufacturing 3
 - 2.3 Assembly..... 8
- 3. Design Verification..... 10
 - 3.1 Specifications 10
 - 3.2 Tests and Results..... 12
 - 3.3 Challenges and Recommended Testing 17
- 4. Discussion and Recommendations 19
- 5. Conclusion 20
- References..... 21

1. Design Updates

Since CDR, there have been three main design changes: utilize the original DAQ housing and microprocessor, redesign the main hub printed circuit board (PCB), and an updated auxiliary sensor housing for the addition of the hall effect sensor.

Utilizing the Original Microprocessor

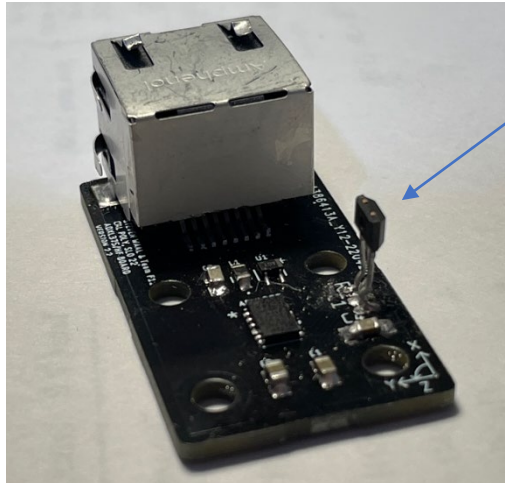
Since CDR, the intent was to integrate a Renesas Electronics 176 Pin microprocessor in order to have more pin connections than the original STMicroelectronics 64 Pin microprocessor. This large increase in number of pins was not a team decision but was due there being a chip shortage. After incorporating the Renesas Microprocessor into the electrical circuit design on Eagle, we held a design review with Cal Poly professor and mechatronics expert, Charlie Refvem. During the review, we learned that incorporating this microprocessor would lead us to more work than we intended. To our advantage, Charlie had two spare microprocessors that were the exact same as Steven Wahl had used with his design (STM32F205RGT6) and were able to just have enough pins on the microprocessor to connect the additional sensors.

Redesigned Main PCB

With the original microprocessors in the hands of the team, we had to redesign the main PCB for a second time. The Renesas microprocessor had a 25.5mmx25.5mm footprint so when we changed back to the smaller STM microprocessor (10mm x 10mm) the original board form factor was a possibility. By reorganizing components of original board design and adding the new components for the additional sensors, we were able to maintain the same board size as the original device. This allowed us to use the same main DAQ aluminum housing which saved the team from having to perform design work on an updated housing. By using the original microprocessor, we were also able to use the original firmware designed for the STM pinouts.

Updated Auxiliary Sensor Housing for Hall Effect

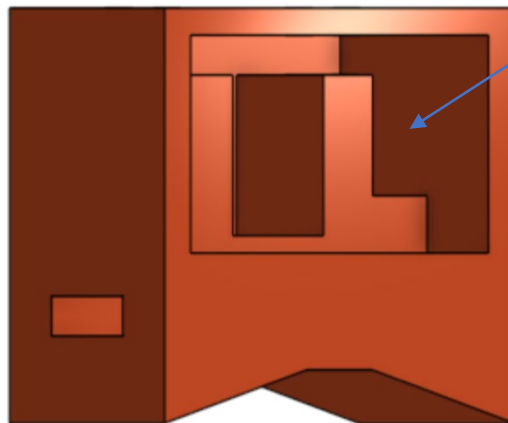
After the completion of the critical design review, the team still did not have a design for incorporating the hall effect to the front wheel auxiliary sensor. Once the new boards were manufactured, and we were able to start hand soldering the electrical components onto the boards, we could measure the final dimensions of the hall effect in reference to the PCB. Figure 1 below shows a picture of the board for reference.



Hall Effect Sensor

Figure 1. Auxiliary Sensor Board with Hall Effect

The Hall effect sensor is indicated by the red arrow. Since the sensor stands proud of the board by 10mm, the team added a slot in the sensor housing to incorporate the sensor. Figure 2 shows a screenshot of the final CAD model of the auxiliary sensor housing with an added slot for the hall effect sensor.



Hall Effect Sensor Slot

Figure 2. Auxiliary Sensor Housing with Hall Effect Sensor Slot

2. Manufacturing

The following section discusses the manufacturing processes that was executed to complete the verification prototype. The processes led to the development of the new data acquisition system.

2.1 Procurement

Our part procurement process was largely accomplished through purchasing from online vendors. All surface mounted (SMT) electrical components that will be soldered to the PCBs, including resistors, capacitors, batteries, switches, LEDs, and even the aluminum enclosure that houses the main unit were purchased from Digikey. We chose this vendor because of their exceptional variety and quantity of parts available, which is critical during supply shortages. We were able to save money by buying in bulk and consolidating most of our components into one shipment. We selected our components based on our system requirements and how compatible they are with the previous version of the DAQ to minimize redesign efforts and potential errors.

We ordered our PCBs and the stencils that we used to apply solder paste to the boards from JLC PCB because of their cheap costs, quick turnaround time, and reputation when it comes to quality manufacturing.

2.2 Outsourcing & Manufacturing

The outsourcing of our components begins with the manufacturing of the Main DAQ PCB, Accelerometer + Hall Effect PCB, and Accelerometer only PCB. These 2-layer PCBs were manufactured from JLC PCB. In conjunction with this order, we also requested to have stainless steel PCB stencils manufactured. These stencils allowed us to skim solder paste onto the PCBs. A zoomed in picture of the main stencil is shown in Figure 3.

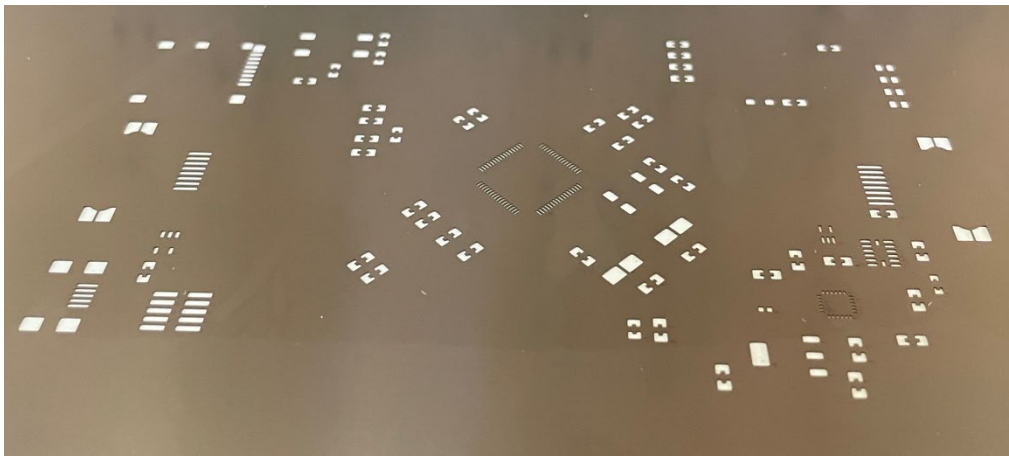


Figure 3. Main DAQ Stencil close up

In order to have these PCBs and Stencils manufactured, we submitted the necessary Gerber files to JLC. Gerber files describe the layout and properties of the PCB as defined by our CAD model. Below are pictures of the four different boards manufactured by JLC.

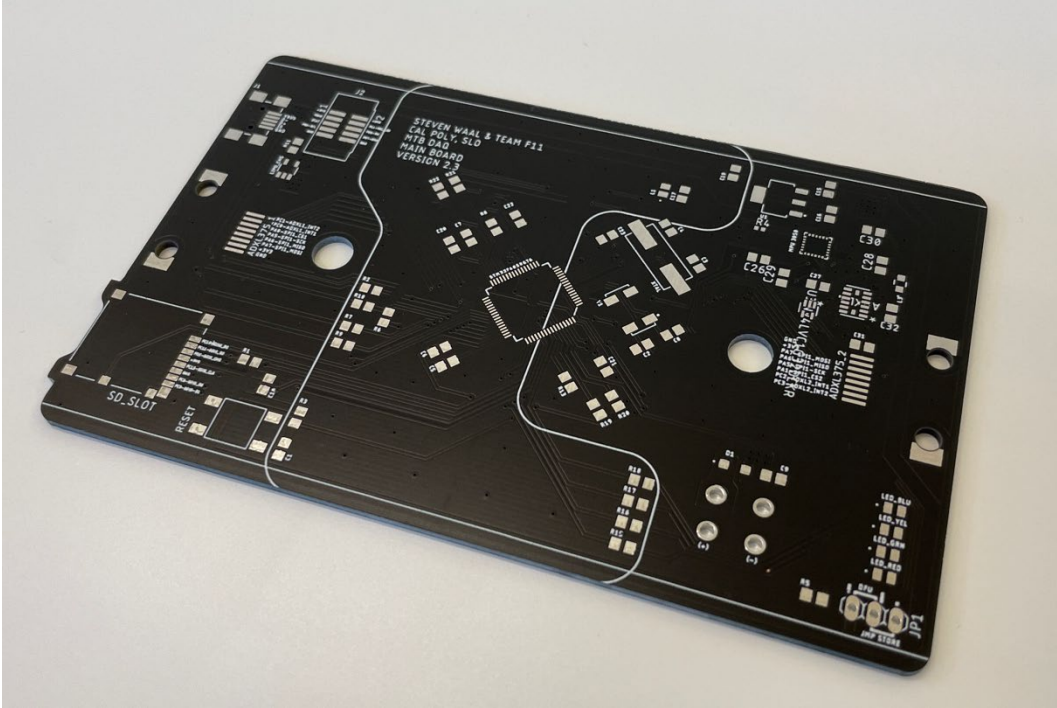


Figure 4. Front of Main DAQ PCB

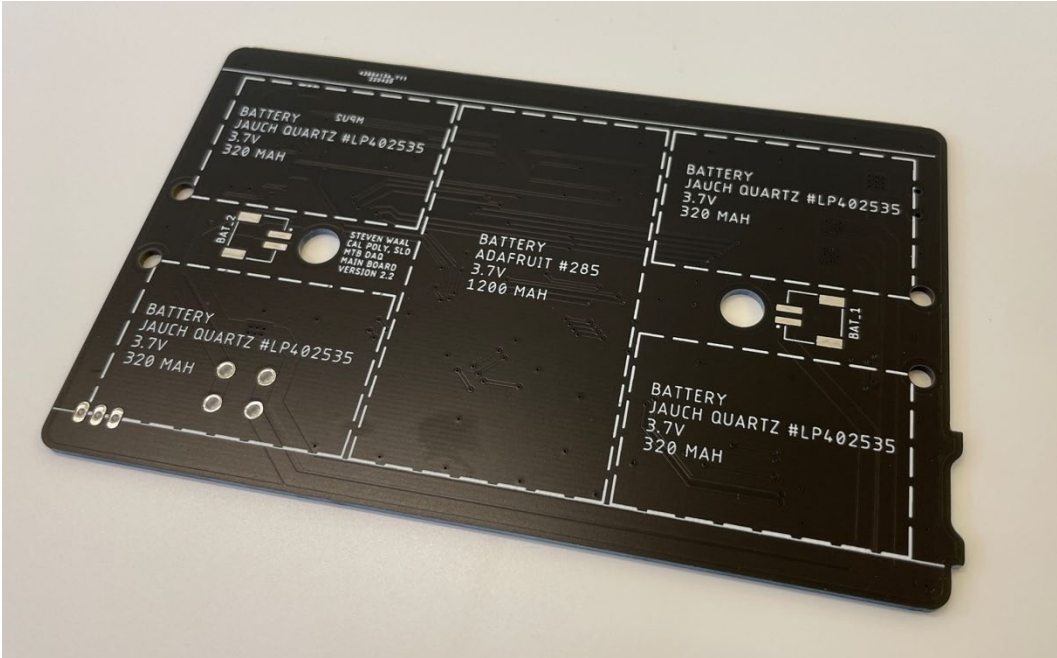


Figure 5. Back of Main DAQ PCB

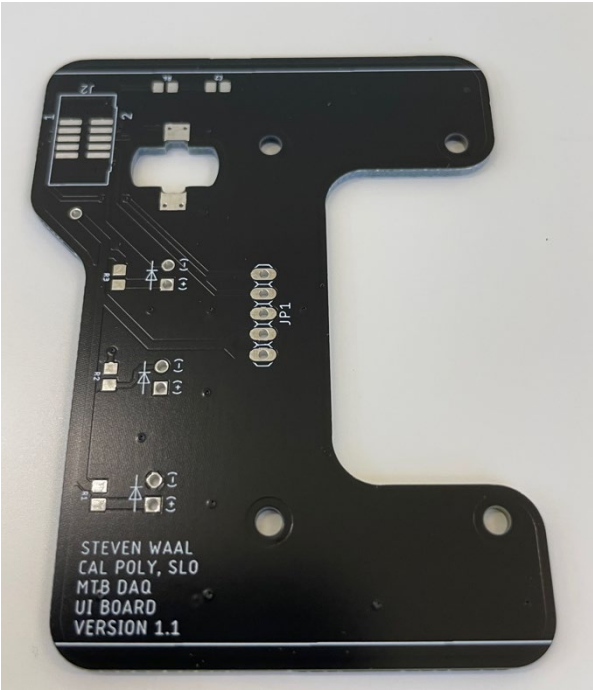
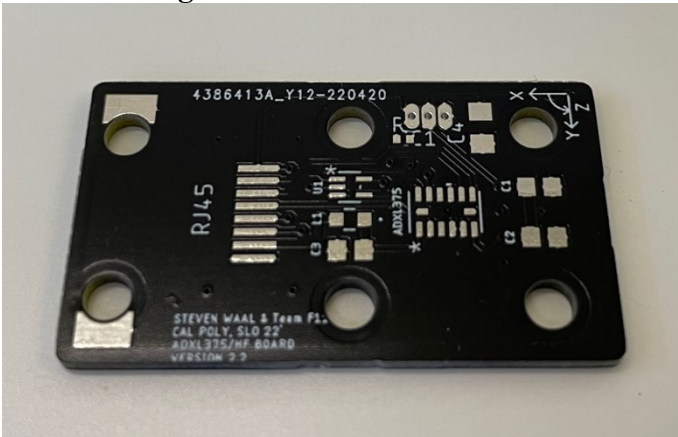


Figure 6. User Interface PCB



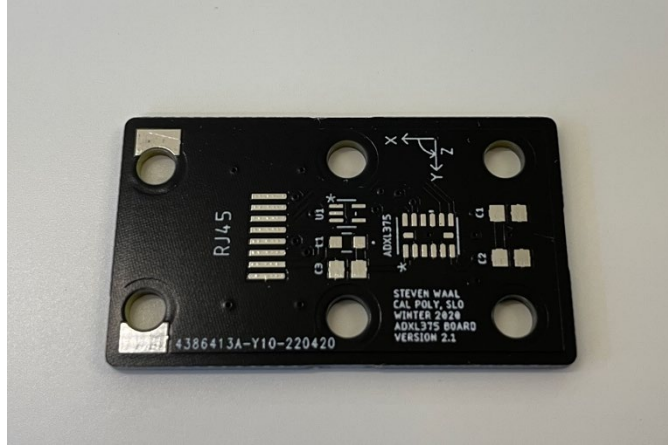


Figure 7. Auxiliary Sensor Boards PCB

In addition to manufacturing the PCBs, we needed to produce more housings for the components. For the main DAQ unit, we purchased an aluminum enclosure (see Figure 8). This enclosure had to be modified in two ways. The faceplate needed to have holes cut out for the LEDs, display, and bolt holes. Second, the side panels will be 3D printed to allow the cables and SD card to be inserted.



Figure 8. Blank Aluminum Enclosure for Main DAQ Unit

In order to machine the faceplate, we 3D printed a stencil showing all the features we needed to cut into it. This was placed over the faceplate and drilled through for the circular holes. The square display hole was traced through the stencil, and then the corners were drilled out and the sides cut with an angle grinder. Finally, the corners and sharp edges were cleaned up with files. The stencil is shown in Figure 9, followed by the machined faceplate in Figure 10.



Figure 9. DAQ Housing Faceplate with 3D-Printed Manufacturing Stencil

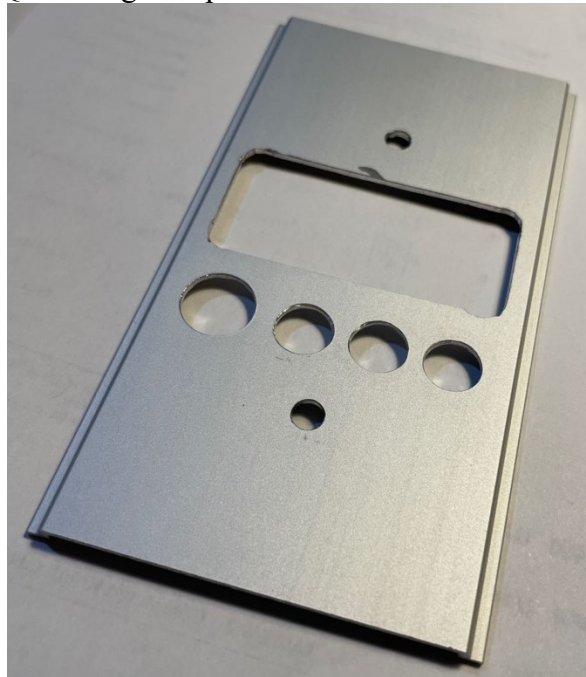


Figure 10. Machined Faceplate

2.3 Assembly

The assembly process for our system includes adding all the electronic components onto the PCB's making them effectively complete. To add all the SMT components to the PCB's we simulated a reflow soldering process using a hot plate in place of a reflow oven.

First, we oriented the stencils on top of the PCBs, making sure the slots were perfectly aligned with the pads on the PCBs. Next, we applied solder paste to the top of the stencil slots, scraping off the excess paste with a putty knife to ensure the paste was applied evenly.



Figure 11. Applying Solder Paste Using the Stencil

After the paste was applied, we removed the stencil carefully, without disturbing the paste applied to the PCB. Then we placed all of the SMT components carefully atop of the solder paste.

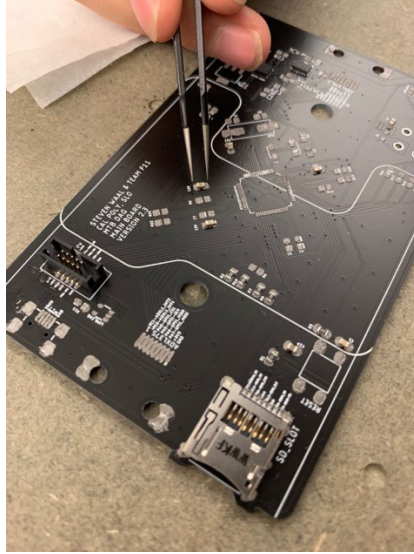


Figure 12. Placing all Components on the Applied Solder Paste

Using the hot plate, we simulated the reflow process by heating up the board and melting the solder within the solder paste to create viable connections between our components and the board.



Figure 13. Using Hotplate to Simulate Reflow Oven

During this assembly process, we considered hand soldering all the components onto the boards, however this posed a high risk. Hand soldering SMT components onto a board can be challenging without the right equipment, and one mistake can make the entire board non-functional. Due to these reasons, we decided to pursue other options first and leave this as a last resort. Next, we planned to simulate the reflow process using a heat gun. This method is primarily used to rework certain components on the board and would have been inefficient for our use as we are assembling the entire board which consists of many components. Finally, without the use of a reflow oven, we decided to use a hot plate. This would allow us to control the temperature of the entire board rather than certain segments. Using a hot plate proved to be the most efficient method to assemble our PCBs. Charlie Revfem assisted our team in applying the hot plate method for our PCB assembly.

With the hot plate, we attempted to follow the solder paste manufacturer's provided thermal profile [1]. We used this as an approximate goal since we didn't have the sophisticated equipment required to precisely control the board's temperature.

3. Design Verification

The following section discusses the results of the testing conducted on the MTB DAQ verification prototype. The original specifications designed during the SOW are revisited to determine if the new device passed or failed. Then the limited metric testing is discussed to answer whether the device can provide suspension tune feedback.

3.1 Specifications

Our team prepared multiple specifications that were important for our product's design based on our initial plan. However, as our team decided to prioritize the functionality and metric testing of our system, many of these specifications were not able to be met. The following specifications are referenced within the DVP&R Table in Appendix F.

Main Hub Size

We verified the Main Hub size using calipers. Since our new design of the main hub was not altered from the original design, despite the additions of the new accelerometer and gyroscope sensors, our Main Hub dimensions are within the scope of our specification, being around 5"x3"x1" in size.

Sensor Housings Size

It is important that our sensor housings do not obstruct any of the bikes' mechanical parts or the user's natural pathway when operating the mountain bike. With this in mind, our team decided to keep the sensor's housings limited to 1.5"x1.5"x1.5" if possible. Our final design was a sensor housing that was 1.7"x1.2"x1.06". Although these dimensions do not match our initial projection, we determined that this size was sufficient for avoiding obstruction while also providing enough protection for the sensors. These measurements were also conducted with calipers.

Weight

The weight of our entire system, including the main hub, two external sensors, housings and cables were measured on a scale. The weight of the entire system is 685g. The weight specification requires the whole system to be less than 500g. It is desirable to add as little weight as possible to the mountain bike, as the system is meant to enhance the performance of the mountain bike. A weight similar to that of a full water bottle was desired to justify its use for customers.

Cost

Our design team initially decided to limit the cost to produce this device to be less than \$150. This specification was decided on when our goal was to create a customer ready product. However, since our scope has changed, the cost to produce the device is no longer relevant as most of the cost comes from multiple shipping orders.

Battery Life

The battery life was tested during our metric testing. We set a time limit of at least one hour of operation in our specifications for our design. This was set with the intention of it being used for testing sessions that could last at least one hour with no available power sources nearby. Our team was able to use it for a little over two hours while testing, having no problems with data corruption from the device with low battery.

Ingress Protection

Due to the change in scope, ingress protection for our device was not designed or tested. Our main device's main purpose was to use for metric testing and would fail this test if it were performed with our current design.

Foolproof

This test focused solely on the use of the Main DAQ system itself, not understand or post-processing the data. Although our team believes it is important to be able to understand and post-process the data to achieve results within testing, it is required to have previous knowledge with the computing software MATLAB.

Maximum Recording Storage

This test solely relies on the SD card used. Since our design implements an external storage unit (SD Card), the amount of data it can store is varied by the resources of the user. Our design does

not have a fixed storage amount and easily exceeds or design specification of storing 8gb or more. Our team used a 16gb SD card for our testing and had no problems with the memory being full.

Mounting Universality

This specification ensures the possibility of using the DAQ device on all types of full suspension MTBs. Upon testing this specification, our team encountered an unexpected problem that resulted in our design to fail this specification. While all MTB's have a water bottle bosses that are standard in spacing, the position of the water bottle bosses can vary. This resulted in our test to fail, as we were unable to insert the ethernet cables into the Main Hub after attaching it to the MTB due to the obstruction of the frame.

Aesthetics

This specification was also not able to be testing and lost its importance once we switched from the primary goal of creating a customer ready product. Since the design of the new device was primarily to test for metrics when optimizing suspension tuning settings, the aesthetics of the design was no longer important and not within the scope of our project.

Suspension Tuning Recommendation

This specification is out of our project's scope as we are primarily using the device to validate the metrics used to create a suspension tuning recommendation. Our device will no longer give a tuning recommendation itself and is not designed to perform that desired action. It will, however, be used to collect data that can be used to interpret what settings to change based on our metric testing.

3.2 Tests and Results

Metric Testing

Metric testing is the main test that evaluates the performance of the MTB DAQ. Since the overall goal of the project is to quantitatively provide suspension tune recommendations, we had to gather as much data on different suspension settings. There are three suspension tune recommendations the DAQ can provide feedback on; stiffness, compression damping, and rebound damping. As a team, we concluded a proven suspension tune recommendation must be backed by large amounts of data. This data was gathered while a rider rode the same trail on many different suspension tune setups. This includes modifying each one of the settings individually as well as at the same time. The validation of a tune would be based on speed and qualitative feedback. Speed meaning how fast did the rider get from point A to point B. Qualitative feedback from the rider would be their opinion on how it felt and what it was doing on the trail. With these two validation points, the metric tested during these trails would hopefully backup the faster speed and better rider feedback with some sort of sensor rates.

The first tested metric was a fork handlebar transmissibility test. This test had one accelerometer unit positioned on the handlebar and one accelerometer unit positioned on the fork lower. During the first day of testing, we looked at different fork stiffnesses only. The fork used during this testing was a Fox 38 with air pressures ranging from 80-100 psi with 5 psi intervals. This range of pressures was decided on since the recommended pressure by fox was 90 psi for the rider. We assumed that with this starting point recommendation, the fork stiffness would not be changed more than ± 10 psi. The testing was performed on the first upper segment of Shooters on the Cuesta Grade. This trail is roughly 1 minute long and features high speed flat corners, chattering sections, and one small jump. The team collected 2 trials per fork stiffness, totaling up to 10 runs. For post processing the data we first took the rms value of the fork and handlebar at a time step of 1ms. This plot ended up showing no clear distinction run-to-run. We then increased this time step until we could identify spikes or reductions in amplitudes. The final time step we arrived at was ~ 2 sec. The 80 psi and 100 psi data are shown below in Figure 14.

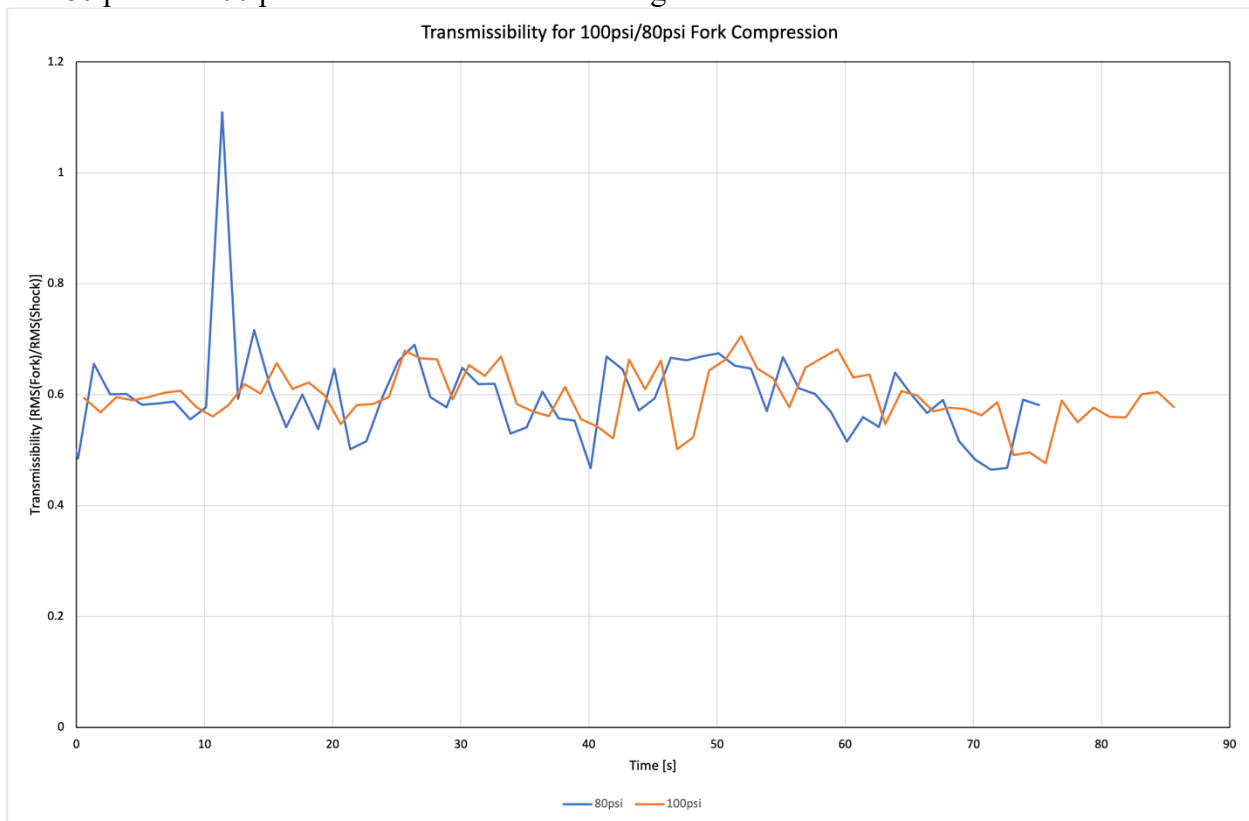


Figure 14. Fork-Handlebar Transmissibility Plot of 80psi and 100psi Fork Stiffness

There were a few things to take away from this testing. First, we needed to identify when the trail begins on the plots. This could be as simple as introducing a small feature at the beginning of trail that should show a spike on the plot. Secondly, it seems that the fork-handlebar transmissibility metric may not be the valid way to provide a suspension tune as we did not see dramatic changes in the data collected.

The next session took place on May 31st, 2022 and looked at fork handlebar transmissibility again. Max rode the bike during this set of runs. The suspension component being altered was rebound.

We did 6 total runs, this time identifying the start of the trail by riding over a 2”x4” block to signify a peak in the data. From there we could chop the data collection at the actual start and end of the trail. Here is a plot of the transmissibility for the fastest rebound setting, lowest rebound setting, and Fox Factory rider recommendation rebound (for Max’s rider weight).

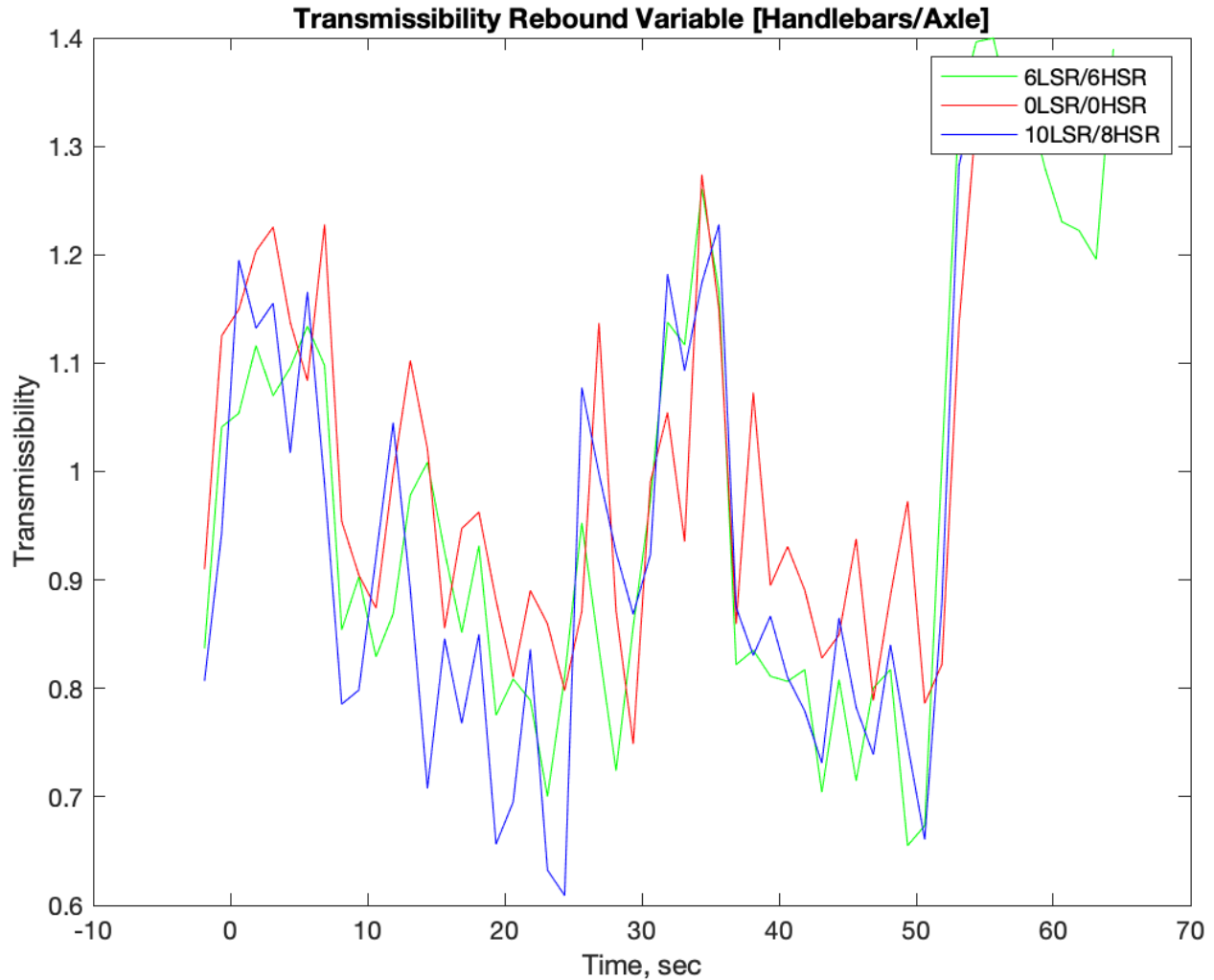


Figure 15. Fork-Handlebar Transmissibility Plot of Slow, Middle, and Fast Rebounds

Figure 15 looks at the RMS values of the fork and handlebar accelerations, but the data was post processed with a time step of ~1.2 seconds. By looking at the graph, we can see that there is a large difference in transmissibility between different setups. Less intuitive, but the setting with the least number of rebound clicks (slowest rebound) showed the largest transmissibility. We took the post processing one step further and took the rms of these data sets with a time step of the entire length of the trail. Figure 16 shows the results.

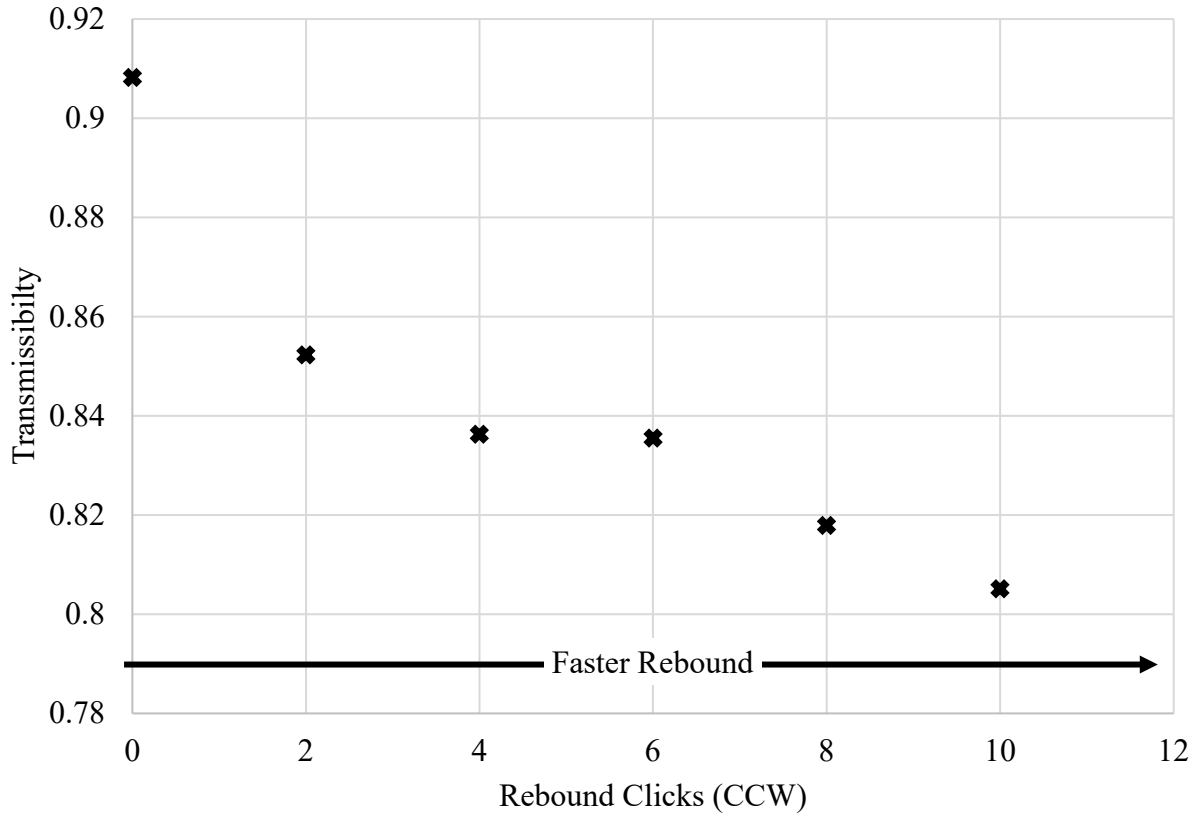


Figure 16. Transmissibility of Entire Run with Varying Rebound

From Figure 16, we can see that as the rebound setting in the front fork increases this transmissibility metric decreases. The results of this testing are contradictory to the qualitative feedback we recorded from Max. The optimal rebound for Max's fork was 6 clicks and two clicks in either direction felt very similar, but as the setting went in either direction the suspension started to feel unstable. In the faster rebounds, the fork felt stiffer and had less traction, whereas the slower rebounds felt too soft and still did not have traction. Our conclusion on the trend of decreasing transmissibility with faster rebound settings is that with faster rebounds the accelerometer on the fork lower was able to accelerate more. The equation below shows the transmissibility function and the variables changing due to faster rebound.

$$\downarrow \text{Transmissibility} = \frac{\text{RMS}(\text{Handlebar Acceleration})}{\text{RMS}(\text{Fork Lower Acceleration})} \uparrow$$

Accelerometer Calibration Verification

Due to time constraints and hardware issues, this test did not end up being performed. The planned procedures are laid out below for future groups who wish to perform such a test.

MEMS accelerometers have a certain amount of inaccuracy inherent to their design. That is, the accelerations they read may not accurately reflect the true accelerations they undergo. To get an

idea of how accurate our ADXL375 accelerometers are, we tested them with known accelerations to compare the data to.

Providing a constant known acceleration to the sensors is a somewhat difficult task. Instead, we used the known acceleration of gravity to get an idea of how the sensors perform. By positioning the sensors at different known angles, we can compare the output of the sensors to the theoretical value. To position the sensors, we created a fixture (Figure 17 below) with slots cut at different angles. For each 15-degree increment, we averaged 30 seconds of data from each axis, and compared them to the expected values. Finally, we took the average and standard deviation of the error at each angle and calculated them to 99.7% uncertainty (3-sigma confidence interval).

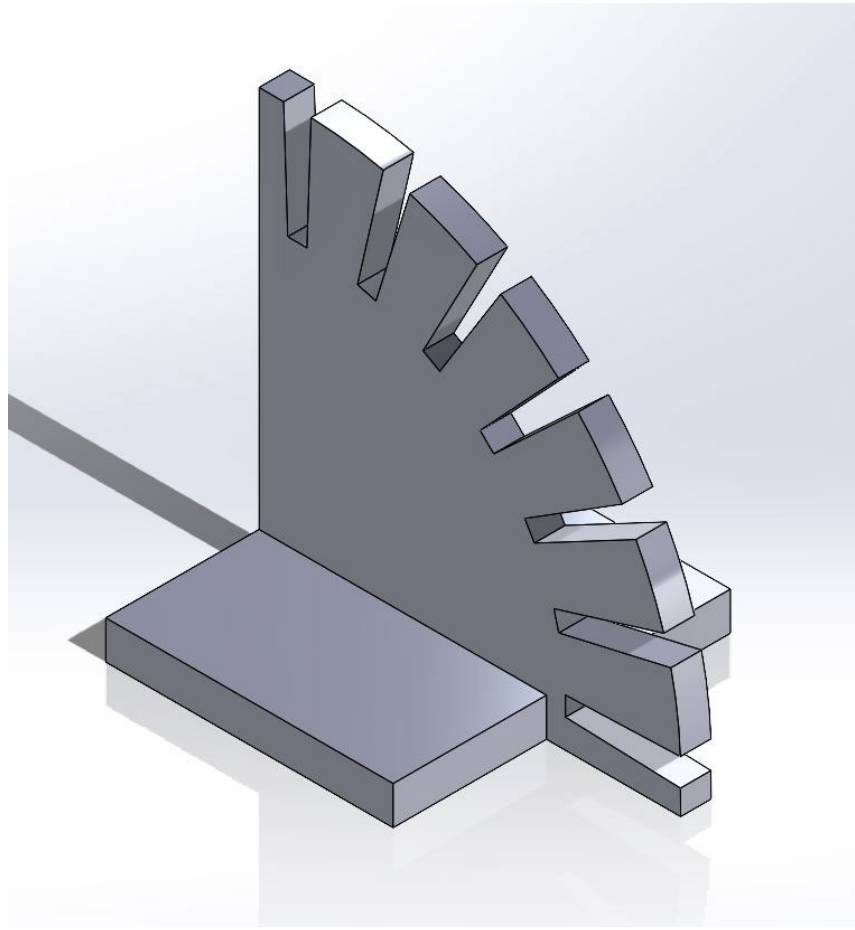


Figure 17. Calibration verification fixture. Accelerometer board would be inserted into the angular slots, cut at 15° intervals from 0° to 90°.

From this calculation, we would get an observed measurement error. This would be compared to the expected measurement error, based on the accelerometer manufacturer's posted uncertainties and the uncertainty of the test rig.

3.3 Challenges and Recommended Testing

Debugging and Troubleshooting

Identifying and solving problems with the original device proved to be a greater challenge for us than we anticipated. Some issues, such as broken soldered connections for buttons, caused bugs that were intermittent and sporadic. This unpredictable behavior made it harder to pinpoint the problem. Another issue that took our team some time to identify and fix was an inadequate power supply, which was probably a result of depleted batteries connected in parallel so that when they were unequally charged after some use they would drain into each other, so they would discharge faster than expected. Buying new batteries seemed to fix this issue, as well as disconnecting the power circuit when not in use.

Procuring Materials and Components

Cal Poly has limited resources for PCB manufacturing and assembly, and electronic components are in short supply globally. This made it difficult to buy replacement components and locate tools and hardware on campus that we needed to use during assembly. There was also a learning curve for our team, as no one had prior experience working with PCBs. Because of this, it took our team longer than anticipated to put together our verification prototype. Charlie Refvem was a big resource for our team and provided us with electronic components and tools we needed to assemble the PCBs that we otherwise would not have had access to.

Metric Testing Challenges

Limited time and logistics, and isolating suspension effects were some of the challenges we faced when testing the original device once it was up and running. We wanted to get realistic data with our system, which meant riding an actual mountain bike trail segment repeatedly, as opposed to some artificially constructed course closer to campus or home. To conduct this testing, we needed to drive up to Cuesta Ridge, set up the bike and DAQ, and ride down a section of trail and pedal back up for each run of data collection. This ended up taking 3+ hours for 10 runs just varying a single suspension parameter, so it's very time intensive. Once we have collected some data and identified some trends, it is not always clear if the cause of the trend is due to the change in suspension settings or something else, like the rider getting more comfortable on the trail and taking a better line or becoming more tired and riding slower. There are many factors that affect the end result. Additionally, there are many different methods to process the data, and it is not always obvious how to analyze it to expose differences between runs and make sense of the behavior.

Recommendations for Testing

First and foremost, we recommend an abundance of testing to collect data on the effects of each individual suspension parameter on the bike's behavior. Collecting a wealth of data with a variety of settings, riders, bikes, and trails will help clarify trends in the data and isolate the effects from specific inputs. It is helpful to have identifiable features that leave distinctive data points as it makes it easier to compare results between runs. This can be difficult for certain trails, so choosing a simple trail that can be ridden in around one minute would be optimal. This trail should have different known significant features that can easily be identifiable when looking at the data. Also, when starting the trial runs, implement something to roll over to indicate when you start the actual trial run. This allows you to identify the start of each run through data processing rather than quickly starting the trial right after you press the button.

4. Discussion and Recommendations

From this design project, we learned that root cause analysis and troubleshooting electronics hardware is difficult and time consuming. When there are dozens of components in a few square inches of circuit board and hundreds of lines of code, there are a multitude of potential underlying causes that might be contributing to the buggy behavior. We also learned that designing and assembling the PCBs takes painstaking time and effort as well as technical skills that we had not practiced until we assembled our final prototype. This is a crucial step in the manufacturing process because one faulty soldered connection can disrupt the functionality of the entire system.

To continue this design, we would fix the board design to add any missing traces and fix any soldering mistakes with the existing components so that every sensor is fully functional. We would also conduct more testing and collect as much data as possible to have more room for developing metrics and identifying trends that result from suspension changes.

If we were to continue refining this design to meet the needs of the customer, we would make the system more user friendly to set up and operate. Specifically, we would redesign the UI to include a battery charge level indicator, move the record button to the handlebars for easier access, and recess the power switch below the surface of the housing to prevent inadvertent switch flips. We would also use batteries that are more standard and safer, such as 18650 cylinders, to avoid connecting batteries of varying capacities in parallel. Furthermore, we would improve the housings to eliminate openings to the interior that could allow dust and water to enter the same space as the electronics and incorporate another mounting option for the central unit, so the rider does not have to choose between bringing a water bottle or the DAQ on a ride. Finally, we would include Bluetooth modules to transmit and receive data wirelessly, so the rider does not have to fidget with messy ethernet cables and zip ties.

If we were to build this prototype again, we would outsource the assembly of the PCBs to streamline the manufacturing process and minimize any soldering mistakes that could result from tediously soldering by hand every component onto the board. We would also like to test the final circuit with a bed of nails to detect voltages and currents at many different grid points to ensure power and signals are being sent where they should be sent.

To produce a high volume of devices at a reasonable price and in a time efficient manner, we would outsource the soldering/assembly process to a facility who specializes in PCB production and has the means to mass produce high quality, reliable boards. We would also buy components in bulk from suppliers to lower the price per component.

Our team recommends using this design to continue testing potential metrics that will lead to the development of an algorithm that can be used to suggest optimal suspension tuning settings. Using the User Manual in Appendix E, anybody should be able to use this design efficiently as our team covers known bugs and issues with the device.

5. Conclusion

In retrospect, this project was not the best fit for a Cal Poly Senior Design Project, given the structure of the class and timeline of deliverables. The emphasis on problem definition and ideation in the first quarter is better suited for teams starting a design from scratch, and we feel that we could have achieved more of our goals if we had been able to test, troubleshoot, and refine the existing design earlier in the year [2].

In the end, we were able to fully debug the original device and collect good data with a mountain bike on the trail. We designed and built a partially functional data acquisition device, capable of collecting data with our two auxiliary accelerometers. We also redesigned and manufactured sensor housings that are more universally compatible with different bikes. Finally, we analyzed data that we collected to start identifying trends and experiment with post-processing methods.

We did not achieve a fully functional device capable of collecting data from all three accelerometers, the gyroscope, and the hall effect speed sensor. We also did not fully develop and refine enough metrics to robustly process data and draw meaningful conclusions. We attribute these shortcomings to our mismanagement of time at the beginning of the year, when we should have been prioritizing testing and hardware troubleshooting. However, we feel that another reason we failed to achieve these end goals is the ambitious scope of this multi-faceted project. Considering the experience of our team coming into this project and the resources and guidance available to us throughout the year, the mismatch between this project's needs and the course's structure and requirements, and the miscommunication about critical flaws in the original device, we feel we were not adequately prepared to successfully complete our goals that we set in the beginning of the year.

If we were to do this project over again, we would refine our goals to focus on getting the hardware and firmware right to build a fully functional data acquisition system from scratch. We would only move on to metric testing, development, and data processing afterwards, or separate this section into its own project entirely.

References

[1] Solder Paste Data Sheet - <https://www.chipquik.com/datasheets/SMD291AX.pdf>

[2] S. R. Waal, "A Quantitative Approach for Tuning a Mountain Bike Suspension", M.S. thesis, California Polytechnic Univ., San Luis Obispo, CA, 2020.

Appendix

A - Main DAQ Python Code

```
# MAIN
#=====
# @file main.py
#=====
# ABOUT:
# This code runs a simple data acquisition system on the MTB-DAQ v2.2
# board running Micropython PYBv1.1 version 1.12 firmware. The code
# records data from three ADXL375BCCZ accelerometers at 1600Hz over SPI and
# one MPU-3050 gyroscope and writes the data to binary file on a Micro SD card.
#=====
# WRITTEN BY: Steven Waal / Updated by Team F11
# DATE: 04.12.2022
#=====
# NOTES:
# 05.28.2019 - File created (SRW)
# 06.07.2019 - Updated to use SPI bus no. 2 instead of no. 1 (not sure why
#             but for some reason SPI bus no. 1 is not working properly)
# 01.29.2020 - Implemented RTOS system
# 02.12.2020 - Implemented display and SD card tasks.
# 02.12.2020 - Having trouble with timing. I can't get faster than 5msec
#             between measurements. I tried moving the record state for
#             each task to the very top (to minimize if statements it
#             takes to get there) but that didn't do too much.
# 02.19.2020 - Implemented code to support ISR
# 03.10.2020 - Changed over to PyBoard. Update pinouts and code to check
#             presence of micro SD card.
# 05.23.2020 - Adapted code to work with MTB DAQ v2.2 main board.
# 06.11.2020 - Final comments added, code cleaned up
# 04.12.2022 - Added third accelerometer and gyroscope
#=====
# COPYRIGHT:
# @copyright This program is copyrighted by Steven Waal and released under
# the GNU Public License, version 3.0.
#=====

#=====
# MISC. NOTES
#=====
# - There is a completely blank file called "SKIPSD" that is loaded onto
#   the board. This prompts the microcontroller to boot from the
#   the internal flash instead of the SD card when an SD card is inserted
#   before powering on.
```



```

#=====
# IMPORT MODULES
#=====
# Import modules for use in this file. Note that there are certain modules
# that automatically come with downloading micropython onto the board. To
# see a list of these modules, type help("modules") in the REPL. This will
# return a list of the available modules. Custom modules can be added by
# saving them as separate *.py files, uploading them to the board, and
# referring to them here.

# HARDWARE MODULES
from ADXL375_driver import ADXL375
from ht16k33_seg import Seg14x4
from MPU3050_Driver import MPU3050

# # GENERAL MICROPYTHON MODULES
import micropython, pyb, utime, gc, machine, os

# MISC.
from helperFunctions import clear_accel_buf
from helperFunctions import decode_data
from helperFunctions import twos_comp
from helperFunctions import get_ODR

#=====
# ALLOCATE MEMORY FOR ERROR REPORTS
#=====
# According to Micropython docs, "If an error occurs in an ISR,
# MicroPython is unable to produce an error report unless a special buffer
# is created for the purpose. Debugging is simplified if the following
# code is included in any program using interrupts."

micropython.alloc_emergency_exception_buf(100)

```

```

#=====
# DEFINE ACCELEROMETER PARAMETERS
#=====
# Determines how many data points are stored in the FIFO buffer before an
# an interrupt is generated. Maximum is 32.
FIFO_BUFF_COUNT    = micropython.const(20)

#=====
# DEFINE BUFFERS
#=====
CMD_RD             = bytearray((0b11110010, 0, 0, 0, 0, 0, 0)) # ADXL375_1.
# Command to read multiple bytes starting with X data
buf1_7            = bytearray(7) # Buffer of ADXL375_1. Make buffer large enough to
# read data from X, Y, and Z
buf2_7            = bytearray(7) # Buffer of ADXL375_2. Make buffer large enough to
# read data from X, Y, and Z
buf3_7            = bytearray(7) # Buffer of ADXL375_3. Make buffer large enough to
# read data from X, Y, and Z
buf4_7            = bytearray(7) # Buffer of MPU-3050. Make buffer large enough to
# read data from X, Y, and Z
buf_HE            = bytearray((0b00000000)) # Buffer of hall effect sensor. Make
# buffer 1 byte to store how many passes of the magnet have occurred.

#=====
# CREATE PIN OBJECTS
#=====
# Create pin objects. Pins are labeled according to the MTB DAQ v2.2 main
# board schematic.

```

```

# SD chip detect pin
CD = pyb.Pin(pyb.Pin.cpu.A8, pyb.Pin.IN, pyb.Pin.PULL_UP)
# Enables the chip detect pin with internal pull up resistor

# ADXL375 1 chip select pin
SPI1_CS1 = pyb.Pin(pyb.Pin.cpu.A0, pyb.Pin.OUT)
# ADXL375 2 chip select pin
SPI1_CS2 = pyb.Pin(pyb.Pin.cpu.A1, pyb.Pin.OUT)
# ADXL375 3 chip select pin
SPI1_CS3 = pyb.Pin(pyb.Pin.cpu.B9, pyb.Pin.OUT)

# Record Button
REC_BTN = pyb.Pin(pyb.Pin.cpu.B3, pyb.Pin.IN)
# Record LED
REC_LED = pyb.Pin(pyb.Pin.cpu.C4, pyb.Pin.OUT)
# Interrupt Pin
ADXL1_INT1 = pyb.Pin(pyb.Pin.cpu.C0, pyb.Pin.IN, pyb.Pin.PULL_DOWN)
# Set internal pull-up resistor so we always know state of pin when not in use

# Callback function to store hall effect reading. Upon the magnet's presence,
# the interrupt will be triggered and run this function.

def HALL_EFF_CB(IRQ_src):
    global buf_HE
    buf_HE[0] = 1

# Hall Effect Interrupt Pin
HALL_EFF_INT = pyb.ExtInt(pyb.Pin.cpu.C1, pyb.ExtInt.IRQ_RISING, pyb.P

# VLogic pin for
VLogic = pyb.Pin(pyb.Pin.cpu.C13, pyb.Pin.OUT)
VLogic.high()
#Create and initialize Gyro I2C object
Gyro_I2C = pyb.I2C(2, pyb.I2C.MASTER)
Gyro_I2C.init(baudrate = 40000)
utime.sleep_ms(100)

#=====
# CREATE DISPLAY OBJECT
#=====
#Create display object. Default to off.
Display = Seg14x4(machine.I2C(1))
Display.text(' ') # Clears display
Display.show()

```

```

=====
# INITIAL CHECK IF SD CARD IS PRESENT
=====
# If SD card is not present, program waits until user inserts one and flashes
# "SD" on the display.
# Once the SD card is present, the proper files and directories are created
# if they have been erased.

# If SD is not detected, flash "SD" on the display.
if CD.value():
    Display.text(' SD')
    Display.blink_rate(2)
    Display.show()

# Wait for user to insert SD card if not already done.
while CD.value():
    # Make display LEDs blink. This will hold up the program and ensure that
    # the SD card is mounted before continuing.
    utime.sleep_ms(100)

# Clear display once SD card has been inserted.
utime.sleep(3)
Display.text(' ') # Clears display
Display.blink_rate(0)
Display.show()

try:
    # Once user has inserted SD card, mount it.
    os.mount(pyb.SDCard(), '/sd')
except:
    Display.text('OFF ')
    Display.blink_rate(2)
    Display.show()

# Remake 'log' directory if it has been erased. If not, ignore error
try:
    os.mkdir('/sd/log')
except:
    pass

# Remake 'count' directory if it has been erased. If not, ignore error
try:

```

```

    os.mkdir('/sd/count')
except:
    pass

# Check if 'count.txt' was erased. If it was, remake it
try:
    file = open('/sd/count/count.txt', 'x') # The 'x' argument indicates that
    # if the file already exists, through an error

    file.write('0\n')
    file.close()
except:
    pass

# Change directory to SD card to prepare for writing files to it
os.chdir('/sd/log')

```

```

#=====
# CREATE SPI OBJECT
#=====
# Create SPI object in order to use the spi protocol

# Set baudrate to maximum of 5 MHz
# Set polarity and phase as specified by sensor datasheets.
spi_1 = pyb.SPI(1, pyb.SPI.MASTER, baudrate=5000000, polarity=1, phase=1,
                bits=8, firstbit=pyb.SPI.MSB)

```

```

#=====
# CREATE ADXL375 ACCELEROMETER OBJECTS AND CONFIGURE SETTINGS
#=====
# *****
# *****ADXL375 1 OBJECT*****
# *****

```

```

ADXL375_1 = ADXL375(spi_1, SPI1_CS1)
ADXL375_1.standby() # puts accelerometer in standby mode. this is necessary to
# configure it
# DATA RATE AND POWER MODE CONTROL
ADXL375_1.odr(ADXL375_1.ODR_1600HZ) # sets data rate to 1600 HZ
ADXL375_1.normal_power_mode()
# DATA FORMAT
ADXL375_1.spi_4_wire()
ADXL375_1.right_justify()
# SETUP INTERRUPTS
ADXL375_1.int_disable(ADXL375_1.Watermark_enable) # Make sure interrupts are
# disabled before configuring as per datasheet
ADXL375_1.FIFO_Mode_FIFO() # Configures the FIFO buffer to operate in FIFO mode
ADXL375_1.trigger_int1() # Configures the interrupt to pin INT1
ADXL375_1.interrupt_active_high() # Configures the interrupt to be active high
ADXL375_1.set_samples(FIFO_BUFF_COUNT) # Sets the number of samples before the
# watermark bit is set

# *****
# *****ADXL375 2 OBJECT*****
# *****
ADXL375_2 = ADXL375(spi_1, SPI1_CS2)
ADXL375_2.standby() # Puts accelerometer in standby mode. this is necessary to
# configure it
# DATA RATE AND POWER MODE CONTROL
ADXL375_2.odr(ADXL375_2.ODR_1600HZ) # Sets data rate to 1600 HZ
ADXL375_2.normal_power_mode()
# DATA FORMAT
ADXL375_2.spi_4_wire()
ADXL375_2.right_justify()

# *****
# *****ADXL375 3 OBJECT*****
# *****
ADXL375_3 = ADXL375(spi_1, SPI1_CS3)
ADXL375_3.standby() # Puts accelerometer in standby mode. this is necessary to
# configure it
# DATA RATE AND POWER MODE CONTROL
ADXL375_3.odr(ADXL375_3.ODR_1600HZ) # Sets data rate to 1600 HZ
ADXL375_3.normal_power_mode()
# DATA FORMAT
ADXL375_3.spi_4_wire()
ADXL375_3.right_justify()

# *****
# *****Gyroscope OBJECT*****
# *****
Gyro = MPU3050(Gyro_I2C)

```

```

#=====
# GET FILE COUNT
#=====
print('GET FILE COUNT')
print()

countFile = open('/sd/count/count.txt', 'r')
last_line = int(countFile.readlines()[-1])
file_count = last_line
countFile.close()

```

```

#=====
# INITIALIZE RECORD LED AND CLEAR DISPLAY
#=====
print('INITIALIZE RECORD LED AND CLEAR DISPLAY')
print()

# Turn off record LED
REC_LED.value(1)
# Clear display
Display.text(' ') # Clears current text/numbers on display
Display.number(file_count) # Prints the desired number
Display.blink_rate(0)
Display.show() # Updates the display

```

```

while True:

```

```

#=====
# WAIT FOR INPUT FROM RECORD SWITCH
#=====

```

```

print('WAITING FOR INPUT...')
print()

while REC_BTN.value() == True: # Wait for user to press button
    pass
while REC_BTN.value() == False: # Wait for user to let go of button
    pass

#=====
# BEGIN RECORDING
#=====
print('RECORDING')
print()

# Turn on record LED
REC_LED.value(0)
# Increment file count and save to count file
file_count += 1 # Increment file count
countFile = open('/sd/count/count.txt', 'a')
countFile.write(str(file_count) + "\n")
countFile.close()

# Update the display
Display.text(' ') # Clears current text/numbers on display
Display.number(file_count) # Prints the desired number
Display.blink_rate(0)
Display.show() # Updates the display

# Clear the accelerometer buffer. Make sure it is in standby mode first
ADXL375_1.standby()
clear_accel_buf(ADXL375_1)
ADXL375_2.standby()
ADXL375_3.standby()

# Create the data file.
file = open('data' + str(file_count) + '.bin', 'wb')

#=====

```



```

# RECORDING!
#=====
# Enable interrupts and start measuring!
ADXL375_1.int_enable(ADXL375_1.Watermark_enable)
ADXL375_1.measure()
ADXL375_2.measure()
ADXL375_3.measure()
HALL_EFF_INT = pyb.ExtInt(pyb.Pin.cpu.C1, pyb.ExtInt.IRQ_RISING,
                          pyb.Pin.PULL_NONE, callback=HALL_EFF_CB)

while REC_BTN.value() == True: # Wait for user to press button
    while ADXL1_INT1.value() == False: # If the INT1 pin is low, wait for
        # accelerometer to collect more data
        pass
    for i in range(FIFO_BUFF_COUNT): # Store values onto SD card in
        # 'log.bin' file

        #Read Accelerometers and gyroscope
        SPI1_CS1.low(); spi_1.send_recv(CMD_RD, buf1_7); SPI1_CS1.high()
        # Read ADXL375_1
        SPI1_CS2.low(); spi_1.send_recv(CMD_RD, buf2_7); SPI1_CS2.high()
        # Read ADXL375_2
        SPI1_CS3.low(); spi_1.send_recv(CMD_RD, buf3_7); SPI1_CS3.high()
        # Read ADXL375_3
        Gyro.i2c.mem_read(buf4_7, 0b1101000, 0x29)
        # Read Gyro

        #Write data bytes to SD card
        file.write(buf1_7) # Write data to log.bin (bytes 0-6)
        file.write(buf2_7) # Write data to log.bin (bytes 7-13)
        file.write(buf3_7) # Write data to log.bin (bytes 14-20)
        file.write(buf4_7) # Write Gyro data to log.bin (bytes 21-28)
        file.write(buf_HE); # Write hall effect data to log.bin (byte 29)

while REC_BTN.value() == False: # Wait for user to let go of button
    pass

#=====
# DONE RECORDING!
#=====
print('DONE RECORDING')

```

```

print()

# Close data file
file.close()

# Turn off accelerometer
ADXL375_1.standby()
ADXL375_2.standby()
ADXL375_3.standby()
# Turn off record LED
REC_LED.value(1)
# Turn off hall effect interrupt
HALL_EFF_INT = pyb.ExtInt(pyb.Pin.cpu.C1, pyb.ExtInt.IRQ_RISING,
                          pyb.Pin.PULL_NONE, callback=None)

print('FINISHED')
print()

Display.text(' ') # Clears current text/numbers on display
Display.number(file_count) # Prints the desired number
Display.blink_rate(0)
Display.show() # Updates the display

```

B - Gyroscope Driver Python Code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
    @file          Gyro.py
    @brief         Driver class that sets up and receives data from an Gyroscope
    """

from pyb import I2C
import struct
import time
import os

class Gyro:
    ''' @brief      An gyro driver class
        @details    Objects of this class can be used to receive gyro data.
        ...
    '''
    ## Constructor
    def __init__(self):
        ''' @brief      Initializes and returns an gyro object
            @details    Initializes the gyro in master mode, allowing the
                        retrieval of data.
            ...
        '''
        self.i2c = I2C(1, I2C.MASTER)

        #FIFO Enable
        self.i2c.mem_write(0b1110001, 0b110100, 0x12)

        #SAMPLE RATE DIVIDER (8KHZ)
        self.i2c.mem_write(0b0000111, 0b110100, 0x15)

        #Parameter Configuration
        self.i2c.mem_write(0b00011000, 0b110100, 0x15)

    def Who_am_I(self, Identity):
        self.i2c.mem_write(Identity, 0b110100, 0x0)

    def USER_CONTROL(self, Control):
        self.i2c.mem_write(Control, 0b110100, 0x3D)

    def sleep(self, Power_Management):
        self.i2c.mem_write(Power_Management, 0b110100, 0x3E)

    def gyro_offsets(self, offsets):
        self.i2c.mem_write(offsets, 0b110100, 0xC)

    def Read_FIFO_COUNT(self):
        return self.i2c.mem_read(2, 0b110100, 0x3A)

    def READ_FIFO_DATA(self):
```

```
# confused on what a burst read refers to
return

## Method to read angular velocity from the Gyro
def omega(self):
    ''' @brief Method to read angular velocity from the Gyro to use as
        state measurements
    '''
    self.omega_signed_ints = struct.unpack('<hhh',
                                           self.i2c.mem_read(6, 0b110100,
                                                             0x29))
    self.omega_vals = tuple(self.omega_int/16 for self.omega_int in
                           self.omega_signed_ints)
    return self.omega_vals
```

C - Data Processing MATLAB Code

6/6/22 10:52 AM Accelerometer_Data_Testing.m

1 of 8

```
%% Accelerometer_Data_Testing.m
%=====
% ABOUT:
% This code interprets the binary data saved from the ADXL375BCCZ
% accelerometer using the "Convert_ADXL375_Data" function and then plots
% the results.
%=====
% WRITTEN BY: Steven Waal
% DATE: 02.29.2021
% UPDATED BY: DYLAN RUIZ & THEO PHILLIBER
% DATE: 06.02.2022
%=====
% NOTES:
% 02.29.2021 - File created (SRW).
% 06.02.2022 - Post Processed Data for Transmissibility Metric
%=====

% Clear statements
clear all;
close all;
clc;

%=====
% First Test Run
%=====
% Set the filepath here!
% filePath = "/Volumes/DATA/log/data1.bin"; % Filepath to where the binary
% data file is stored
filePath = ('data155.bin');
frequency = 1600; % [Hz] Frequency of the accelerometers

% Run "Convert_ADXL375_Data" function
out = Convert_ADXL375_Data(filePath, frequency);

% Rename outputs for clarity
time_1 = out(1:92500,1);
x1 = out(1:92500,2);
y1 = out(1:92500,3);
z1 = out(1:92500,4);
x2 = out(1:92500,5);
y2 = out(1:92500,6);
z2 = out(1:92500,7);
sum1 = 0;
sum2 = 0;
factor = 2000; % Factor used to create RMS timesteps for post processing.
% The larger the factor, the bigger timestep between RMS calcs.
% length(x1) > factor > 1

% Calculates the total magnitude of accelerations per accelerometer
for i = 1:length(x1)
    total_mag_1(i) = (x1(i)^2+y1(i)^2+z1(i)^2)^(1/2);
    total_mag_2(i) = (x2(i)^2+y2(i)^2+z2(i)^2)^(1/2);
end
```

```
% calculates the total RMS of each accelerometer for the entire run
for n = 1:length(total_mag_1)
    sum1 = sum1+(total_mag_1(n))^2;
    sum2 = sum2 + (total_mag_2(n))^2;
end
Accelrms_1 = ((1/length(total_mag_1))*sum1)^(1/2);
Accelrms_2 = ((1/length(total_mag_2))*sum2)^(1/2);

% Transmissibility of the entire run
Transmiss = Accelrms_2/Accelrms_1;

% Calculates the rms of each timestep based on the factor.
for n = 1:(length(total_mag_1)/factor)-1

    rms_array_1 = total_mag_1((n-1)*factor+1 : (n)*factor);
    rms_array_2 = total_mag_2((n-1)*factor+1 : (n)*factor);
    time_array = time_1((n-1)*factor+1 : (n)*factor);

    for i = 1:length(rms_array_1)
        sum1 = sum1+(rms_array_1(i))^2;
        sum2 = sum2+(rms_array_2(i))^2;
    end

    Accel_rms_1(n) = ((1/length(rms_array_1))*sum1)^(1/2);
    Accel_rms_2(n) = ((1/length(rms_array_2))*sum2)^(1/2);
    time_list(n,1) = mean(time_array);
    sum1 = 0;
    sum2 = 0;
end
trans = (Accel_rms_2./Accel_rms_1)';

% Plot raw data accelerometer 1
figure
plot(time_1, x1, time_1, y1, time_1, z1);
legend('X', 'Y', 'Z');
xlabel('Time, sec');
ylabel('Acceleration, g');
title('Accelerometer 1');

% Plot raw data accelerometer 2
figure
plot(time_1, x2, time_1, y2, time_1, z2);
legend('X', 'Y', 'Z');
xlabel('Time, sec');
ylabel('Acceleration, g');
title('Accelerometer 2');
%axis([0,10.5,-5,3])

% Plot magnitude accelerometer 1
figure
plot(time_1, total_mag_1);
xlabel('Time, sec');
```

```
ylabel('Acceleration Magnitude, g');
title('Accelerometer 1 Magnitude');
%axis([0,10.5,0,7])

% Plot magnitude accelerometer 1
figure
plot(time_1, total_mag_2);
xlabel('Time, sec');
ylabel('Acceleration Magnitude, g');
title('Accelerometer 2 Magnitude');
%axis([0,10.5,0,7])

% Plot RMS accelerometer 1
figure
plot(time_list, Accel_rms_1);
xlabel('Time, sec');
ylabel('Acceleration RMS, g');
title('Accelerometer 1 RMS');
%axis([0,10.5,0,5])

% Plot RMS accelerometer 2
figure
plot(time_list, Accel_rms_2);
xlabel('Time, sec');
ylabel('Acceleration RMS, g');
title('Accelerometer 2 RMS');
%axis([0,10.5,0,5])

% Plot Transmissibility
figure
plot(time_list, trans);
xlabel('Time, sec');
ylabel('Transmissibility');
title('Transmissibility [Handlebars/Axle]');

%%=====
% Second Test Run
%%=====

% Set the filepath here!
%filePath = "/Volumes/DATA/log/data1.bin"; % Filepath to where the binary
% data file is stored
filePath = ['data156.bin'];
frequency = 1600; % [Hz] Frequency of the accelerometers

% Run "Convert_ADXL375_Data" function
out = Convert_ADXL375_Data(filePath, frequency);

% Rename outputs for clarity
time_2 = out(:,1);
x1 = out(:,2);
y1 = out(:,3);
z1 = out(:,4);
x2 = out(:,5);
```

```

y2          = out(:,6);
z2          = out(:,7);
sum1 = 0;
sum2 = 0;
total_mag_1 = 0;
total_mag_2 = 0;
rms_array_1 = 0;
rms_array_2 = 0;

% Calculates the total magnitude of accelerations per accelerometer
for i = 1:length(x1)
    total_mag_1(i) = (x1(i)^2+y1(i)^2+z1(i)^2)^(1/2);
    total_mag_2(i) = (x2(i)^2+y2(i)^2+z2(i)^2)^(1/2);
end

% calculates the total RMS of each accelerometer for the entire run
for n = 1:length(total_mag_1)
    sum1 = sum1+(total_mag_1(n))^2;
    sum2 = sum2 + (total_mag_2(n))^2;
end
Accelrms_11 = ((1/length(total_mag_1))*sum1)^(1/2);
Accelrms_12 = ((1/length(total_mag_2))*sum2)^(1/2);

% Transmissibility of the entire run
Transmiss_2 = Accelrms_12/Accelrms_11;

% Calculates the rms of each timestep based on the factor.
for n = 1:(length(total_mag_1)/factor)-1

    rms_array_1 = total_mag_1((n-1)*factor+1 : (n)*factor);
    rms_array_2 = total_mag_2((n-1)*factor+1 : (n)*factor);
    time_array = time_2((n-1)*factor+1 : (n)*factor);
    for i = 1:length(rms_array_1)
        sum1 = sum1+(rms_array_1(i))^2;
        sum2 = sum2+(rms_array_2(i))^2;
    end

    Accel_rms_11(n) = ((1/length(rms_array_1))*sum1)^(1/2);
    Accel_rms_22(n) = ((1/length(rms_array_2))*sum2)^(1/2);
    time_list_11(n,1) = mean(time_array);
    sum1 = 0;
    sum2 = 0;
end

trans_1 = (Accel_rms_22./Accel_rms_11)';

% Plot raw data accelerometer 1
figure
plot(time_2, x1, time_2, y1, time_2, z1);
legend('X', 'Y', 'Z');
xlabel('Time, sec');
ylabel('Acceleration, g');

```



```
title('Accelerometer 1');

% Plot raw data accelerometer 2
figure
plot(time_2, x2, time_2, y2, time_2, z2);
legend('X', 'Y', 'Z');
xlabel('Time, sec');
ylabel('Acceleration, g');
title('Accelerometer 2');

% Plot Magnitude accelerometer 1
figure
plot(time_2, total_mag_1);
xlabel('Time, sec');
ylabel('Acceleration Magnitude, g');
title('Accelerometer 1 Magnitude');

% Plot Magnitude accelerometer 2
figure
plot(time_2, total_mag_2);
xlabel('Time, sec');
ylabel('Acceleration Magnitude, g');
title('Accelerometer 2 Magnitude');

% Plot RMS accelerometer 1
figure
plot(time_list_11, Accel_rms_11);
xlabel('Time, sec');
ylabel('Acceleration RMS, g');
title('Accelerometer 1 RMS');

% Plot RMS accelerometer 2
figure
plot(time_list_11, Accel_rms_22);
xlabel('Time, sec');
ylabel('Acceleration RMS, g');
title('Accelerometer 2 RMS');

%%=====
% Third Test Run
%%=====

filePath = ['data157.bin'];
frequency = 1600; % [Hz] Frequency of the accelerometers

% Run "Convert_ADXL375_Data" function
out = Convert_ADXL375_Data(filePath, frequency);

% Rename outputs for clarity
time_3 = out(:,1);
x1 = out(:,2);
```

```

y1      = out(:,3);
z1      = out(:,4);
x2      = out(:,5);
y2      = out(:,6);
z2      = out(:,7);
sum1 = 0;
sum2 = 0;

% Calculates the total magnitude of accelerations per accelerometer
for i = 1:length(x1)
    total_mag_1(i) = (x1(i)^2+y1(i)^2+z1(i)^2)^(1/2);
    total_mag_2(i) = (x2(i)^2+y2(i)^2+z2(i)^2)^(1/2);
end

% Calculates the total RMS of each accelerometer for the entire run
for n = 1:length(total_mag_1)
    sum1 = sum1+(total_mag_1(n))^2;
    sum2 = sum2 + (total_mag_2(n))^2;
end
Accelrms_1 = ((1/length(total_mag_1))*sum1)^(1/2);
Accelrms_2 = ((1/length(total_mag_2))*sum2)^(1/2);

% Transmissibility of the entire run
Transmiss_3 = Accelrms_2/Accelrms_1;

% Calculates the RMS of each timestep based on the factor.
for n = 1:(length(total_mag_1)/factor)-1

    rms_array_1 = total_mag_1((n-1)*factor+1 : (n)*factor);
    rms_array_2 = total_mag_2((n-1)*factor+1 : (n)*factor);
    time_array = time_3((n-1)*factor+1 : (n)*factor);

    for i = 1:length(rms_array_1)
        sum1 = sum1+(rms_array_1(i))^2;
        sum2 = sum2+(rms_array_2(i))^2;
    end

    Accel_rms_31(n) = ((1/length(rms_array_1))*sum1)^(1/2);
    Accel_rms_32(n) = ((1/length(rms_array_2))*sum2)^(1/2);
    time_list_3(n,1) = mean(time_array);
    sum1 = 0;
    sum2 = 0;
end
trans_3 = (Accel_rms_32./Accel_rms_31)';
%rms_array_1(1:length(total_mag_1),1) = Accel_rms_1;
%rms_array_2(1:length(total_mag_2),1) = Accel_rms_2;

% Plot raw data accelerometer 1
figure
plot(time_3, x1, time_3, y1, time_3, z1);
legend('X', 'Y', 'Z');
xlabel('Time, sec');

```

```
ylabel('Acceleration, g');
title('Accelerometer 1');

% Plot raw data accelerometer 2
figure
plot(time_3, x2, time_3, y2, time_3, z2);
legend('X', 'Y', 'Z');
xlabel('Time, sec');
ylabel('Acceleration, g');
title('Accelerometer 2');

% Plot Magnitude accelerometer 1
figure
plot(time_3, total_mag_1);
xlabel('Time, sec');
ylabel('Acceleration Magnitude, g');
title('Accelerometer 1 Magnitude');

% Plot Magnitude accelerometer 2
figure
plot(time_3, total_mag_2);
xlabel('Time, sec');
ylabel('Acceleration Magnitude, g');
title('Accelerometer 2 Magnitude');

% Plot RMS accelerometer 1
figure
plot(time_list_3, Accel_rms_31);
xlabel('Time, sec');
ylabel('Acceleration RMS, g');
title('Accelerometer 1 RMS');

% Plot RMS accelerometer 2
figure
plot(time_list_3, Accel_rms_32);
xlabel('Time, sec');
ylabel('Acceleration RMS, g');
title('Accelerometer 2 RMS');

% Plot Transmissibility Run 3
figure
plot(time_list_3, trans_3);
xlabel('Time, sec');
ylabel('Transmissibility');
title('Transmissibility [Handlebars/Axle]');

% Plot the Transmissibilities of the three runs
figure
plot(time_list, trans, 'g', time_list_11, trans_1, 'r', time_list_3, trans_3, 'b');
xlabel('Time, sec');
ylabel('Transmissibility');
title('Transmissibility Rebound Variable [Handlebars/Axle]');
legend('8LSR/8HSR', '4LSR/4HSR', '2LSR/2HSR')
```

D - Converting Data MATLAB Code

6/6/22 10:58 AM Convert_ADXL375_Gyro_Data.m

1 of 3

```
%% Convert_ADXL375_Data.m
%=====
% ABOUT:
% This function interprets the binary data saved from the ADXL375BCCZ
% accelerometer.
%=====
% ARGUMENTS:
%   filePath      = The filepath to the binary file.
%
%   frequency     = The frequency at which the data was collected.
%=====
% OUTPUTS:
%   out           = [time, x_acceleration_1, y_acceleration_1,
%                   z_acceleration_1, x_acceleration_2, y_acceleration_2,
%                   z_acceleration_2];
%                   The interpreted data from both accelerometers (1 and
%                   2) for the x, y, and z axes.
%=====
% WRITTEN BY: Steven Waal
% DATE: 05.06.2019
% UPDATED BY: DYLAN RUIZ
% DATE: 06.02.2022
%=====
% NOTES:
%   05.06.2019 - File created (SRW). Referenced EOM_2DOF (older version).
%               (SRW)
%   09.12.2020 - Cleaned up code and added more comments. (SRW)
%   06.02.2022 - Added functionality for third accelerometer, gyro and hall
%               effect sensor.
%=====

function out = Convert_ADXL375_Data(filePath, frequency)

% CONSTANTS
SCALE_FACTOR = 0.0488;           % [g/LSB] (see ADXL375 data sheet)
SCALE_FACTOR_GYRO = 1/16.4;     % [(deg/s)/LSB] (see MPU3050 data sheet)
                                % (FS_SEL = 3)

% Open binary data file from MTB DAQ
file = fopen(filePath);

% Read binary data from file and store in variable 'data'
data = fread(file);

% Close binary data file
fclose(file);

data = reshape(data, 29, []);

% Decode data

% Add LSBs and MSBs together to reconstruct X, Y, and Z data

% Accel 1
x1 = data(2,:) + 256*data(3,:);
```

```
y1 = data(4,:) + 256*data(5,:);
z1 = data(6,:) + 256*data(7,:);

% Accl 2
x2 = data(9,:) + 256*data(10,:);
y2 = data(11,:) + 256*data(12,:);
z2 = data(13,:) + 256*data(14,:);

% Accl 3
x3 = data(16,:) + 256*data(17,:);
y3 = data(18,:) + 256*data(19,:);
z3 = data(20,:) + 256*data(21,:);

% Gyro
x4 = data(23,:) + 256*data(24,:);
y4 = data(25,:) + 256*data(26,:);
z4 = data(27,:) + 256*data(28,:);

% Hall Effect
HE = data(29,:);

% Take two's complement to get sign of data
for i=1:length(x1)
    if x1(i)>32767
        x1(i)=x1(i)-65536;
    end
    if y1(i)>32767
        y1(i)=y1(i)-65536;
    end
    if z1(i)>32767
        z1(i)=z1(i)-65536;
    end
    if x2(i)>32767
        x2(i)=x2(i)-65536;
    end
    if y2(i)>32767
        y2(i)=y2(i)-65536;
    end
    if z2(i)>32767
        z2(i)=z2(i)-65536;
    end
    if x3(i)>32767
        x3(i)=x3(i)-65536;
    end
    if y3(i)>32767
        y3(i)=y3(i)-65536;
    end
    if z3(i)>32767
        z3(i)=z3(i)-65536;
    end
    if x4(i)>32767
        x4(i)=x4(i)-65536;
    end
end
```

```
    if y4(i)>32767
        y4(i)=y4(i)-65536;
    end
    if z4(i)>32767
        z4(i)=z4(i)-65536;
    end
end

x1 = x1.*SCALE_FACTOR;
y1 = y1.*SCALE_FACTOR;
z1 = z1.*SCALE_FACTOR;
x2 = x2.*SCALE_FACTOR;
y2 = y2.*SCALE_FACTOR;
z2 = z2.*SCALE_FACTOR;
x3 = x3.*SCALE_FACTOR;
y3 = y3.*SCALE_FACTOR;
z3 = z3.*SCALE_FACTOR;
x4 = x4.*SCALE_FACTOR_GYRO;
y4 = y4.*SCALE_FACTOR_GYRO;
z4 = z4.*SCALE_FACTOR_GYRO;

time = [0: 1/frequency: (length(x1)-1)/frequency];

% Output the acceleration values, in g's, rotational velocity in deg/s
out = [time', x1', y1', z1', x2', y2', z2', x3', y3', z3', x4', y4', z4', HE'];

end
```

E - User Manual

The following user manual provides instructions to operate the MTB DAQ as well as important safety information. Read this section this prior to operation and see the troubleshooting section if problems arise.

Operation of DAQ System

Flashing the Main DAQ (directly from Steven Waal's Thesis)

The main board was designed based on the PYBv1.1 schematic. As a result, the firmware and flashing instructions are the same as those for the PyBoard. The main board utilizes the device firmware update (DFU) protocol that comes embedded with each STM32 microcontroller. DFU mode allows for a simple way to update the firmware of an STM32 without requiring specialized hardware. It was mainly designed for updating the firmware remotely on devices that have already been released. To flash firmware to the board, use the following steps:

1. Make sure that a DFU utility program is installed on the computer that will be used to flash the firmware. “dfu-util” is a free DFU utility program than runs in terminal. Install this program via the package manager.

2. With the power to the board turned off, move the jumper on port JP1 from “JMP STORE” position to the “DFU” position. This will tie the DFU pin of the microcontroller to 3.3V. Figure 4.11 depicts these positions. When the board is powered on, the microcontroller will enter DFU mode upon boot.

3. Connect to the board to a computer via USB.

4. Use the dfu-util commands to flash the firmware to the board. For more details on using this software to flash the board, refer to.

5. Once the firmware has been loaded on, power off the board and return the jumper on JP1 to the “JMP STORE” position. The main board runs off the standard released PYBv1.1 DFU firmware files available on the Micropython website. At the time of this writing, the most current version that worked with the main board was pybv11-20191220-v1.12.dfu. [Reference: Steven Waal's Thesis Defense]

Once the proper firmware has been loaded onto the board, the main board will appear as a standard USB device when connected to the computer. At this point, the Micropython files outlined in Section 4.5 can be loaded on to the board using a standard method for transferring files. Note that when first loading on the files, make sure that there is no Micro SD card loaded in, as this will appear instead of the USB device representing the microcontroller internal flash memory. It is important to load the files onto the internal flash memory and not the Micro SD card. Once this has been done, the MTB DAQ is ready for operation.

Formatting the SD Card

The Micro SD card needs to be formatted according to the SD card association. In order to remove old data, it is important to completely erase both the “log” and “count” folders and all of the contents in them. These folders will be remade if they don’t already exist, and the proper files will be generated upon the next power up of the main unit. If only the “log” file is deleted, the system will continue counting based off of the “count.txt” file. The data will still be saved, and the proper number will be displayed on the display of the system, but the numbering will not start over as desired. Reformatting the Micro SD card is a good way to ensure that it is completely erased and ready for a new testing session. [Reference: Steven Waal’s Thesis Defense]

Mounting the System

Attaching the MTB DAQ System is fast and easy to do with any bike you might be using.

The OneUp straps slide through the slots in both auxiliary sensor housings, and then they are placed with the angled surface against the bike so that the strap wraps around the chain stay, fork housing, or handlebars. It is important to place the cadence sensor on the fork housing, adjacent to the magnet location on the spokes of the front wheel, so that the wheel speed may be measured if desired. This sensor should be oriented with the cable port facing the handlebars so that the magnet is in the range of the hall effect sensor.

The central unit mounts to the middle of the main frame, with bolts going through the designated holes in the housing through the water bottle boss to secure the unit in place. Make sure the faceplate with the display screen and indicator lights is face up and visible.

Finally, the ethernet cables plug in to each auxiliary sensor unit, connecting them to the central unit. Any excess cable should be secured to the bike frame so that it does not interfere with the rider’s motion. This will prevent injury to the rider and damage to the DAQ system.

Collecting the Data (directly from Steven Waal’s Thesis)

Before powering on the MTB DAQ, make sure both ADXL375 accelerometers are plugged in to the main unit. The main unit configures the accelerometers upon startup and is not able to re-configure after the power has been turned on. If the accelerometers are not plugged in before the power is turned on, turn off the main unit, plug them in, and turn it back on.

The MTB DAQ is powered on and off via the power switch. Power status is indicated by the power LED (green). Upon power up, the main unit will check for the presence of a Micro SD card. If no Micro SD card is inserted, “SD” will flash continuously on the display. Once a Micro SD card is inserted, “SD” will flash three more times until the main unit detects the card. Once the card has successfully been detected, the display will show the number of the current data file stored on the Micro SD card and the record LED (red) will turn off. If the Micro SD card has just been formatted,

the display will show “0” indicating that no data has been logged. The MTB DAQ is now in standby mode and ready to record data.

Data recording is started by pressing down on the record button. After the first press, the record LED (red) will turn on and the display will increment the displayed number indicating that the main unit is recording data. The number that is displayed during and after recording indicates the number of the data file associated with that recording session. Pressing the record button a second time will make the record LED (red) turn off indicating that the main unit is done recording. The display will continue to show the number from that most recent recording session. The main unit is immediately available for another recording session. When testing is finished, it is best to turn the main unit off before removing the Micro SD card. [Reference: Steven Waal’s Thesis Defense]

Interpreting the Data

After collecting the data on the Micro SD card, the files on the card should be in the format ‘.bin’. Download the MATLAB scripts named “Convert_ADXL375_Gyro_Data.m” and “Accelerometer_Data_Testing.m” (Testing script). The first file is used to convert the accelerometer, gyroscope, and hall effect data from binary to its intended values. The first file is a function file, so it will be referenced when the testing script is run. To run the testing script, you must first insert the file name that you wish to process in the line below,

```
28      filePath = ('data155.bin');
```

For the transmissibility metric, our team added a post-processing segment to the testing script. The post processing of the data includes turning the raw acceleration data into a total magnitude of acceleration, not dependent on direction. Using the magnitudes, we find the total RMS of the entire run and calculate the RMS at certain timesteps to make the Transmissibility plot easier to understand. The timestep is changed based on a factor, which the user can manipulate to change the timestep. The timestep is calculated to be (factor/1600) seconds. You can change the factor in the line below,

```
44      factor = 2000; % Factor used to create RMS timesteps for post processing.
```

The script will ultimately produce three transmissibility values for the entire run, one for each run within the script and plots similar to Figure 13.

Troubleshooting and Known Issues

Our team’s project was a continuation of Steven Waal’s Master’s Thesis, in which he designed a DAQ system consisting of only two external accelerometers. Our team worked off his design and encountered many problems with his data acquisition system. The following are some of the problems/bugs we encountered.

Battery Issues

Throughout our time working with Waal’s version of the DAQ, we encountered problems with the battery design. The DAQ at times would not display the correct message upon bootup or display no message at all. This could be due to the batteries being dead, even if it happens within

a day or two of no use. Our team believes that the DAQ design is leaking power through one of the connections, even when it is switched off. If this problem occurs, unplug the batteries from the ports until you must use the DAQ again. This will keep the batteries charged and ready.

Accelerometer Issues

When operating the DAQ, at least one external accelerometer must be plugged in. The DAQ will not function correctly if the accelerometer is not plugged in via an ethernet cable. The DAQ will show it is recording, but the button will not be able to end the recording session.

Display Bugs

Sometimes, when starting a new recording session, the number on the display won't change. This is a common bug that happens, but it does not affect the file creation of the recording session. If another recording session is created, the display will skip the number of the previous recording and display the correct file number.

Reset Button

On the PCB of the main DAQ, there is a small black button labeled with the word 'reset'. Pressing this button does not restart the system, but instead clears all of the program files off of the MCU. If this button is pressed, the device must be plugged into a computer with USB, and the files copied back onto the device labeled 'PybFlash'.

Soldered Connections

If issues not listed above are occurring, they may be the result of broken solder joints. We have dealt with the display not working properly and identified the issue to be a broken solder joint in the record button. This button is held in place by solder, and the force of pressing the button can break the connection. Through visual inspection and voltmeter readings, the PCBs can be analyzed to find these iss

F - Design Verification Plan & Report

DVP&R - Design Verification Plan (& Report)												
Project:	F11 - MTB DAQ		Sponsor:	Dr. Joseph Mello				Revision Date: 6/2/22				
TEST PLAN									TEST RESULTS			
Test #	Specification	Test Description	Measurements	Acceptance Criteria	Required Facilities/Equipment	Parts Needed	Responsibility	TIMING		Numerical Results	Notes on Testing	Pass/Fail
								Start date	Finish date			
1	Main Hub Size	Measure physical dimensions of main hub.	Lengths	5"x3"x1" or less	Calipers or ruler	SP/FP	Theo	4/18/22	6/2/22	4.96"x2.94"x0.95"	All three dimensions pass.	Pass
2	Sensor Housings Size	Measure physical dimensions of peripheral sensor housings.	Lengths	1.5"x1.5"x1.5" or less	Calipers or ruler	SP/FP	Theo	4/18/22	6/2/22	1.7"x1.2"x1.06"	The length exceeds our specification, but the width and depth are acceptable. These are the important dimensions, as they would obstruct the rider whereas length wouldn't. *Partial Success	Pass*
3	Weight	Weigh entire system (hub, sensors, cables, straps) on scale.	Mass	500g or less	Weight Scale	SP/FP	Ronan	4/18/22	6/2/22	685 grams	Ethernet Cables are too long for device, add much more weight than expected.	Fail
4	Cost	Add up entire cost of final system	Dollars	Under \$150	None	None	Ronan	4/18/22	6/2/22	\$700	This test was created with the idea of creating in bulk. Since we moved to a testing device design, most of the money spent went on different iterations.	Fail
5	Battery Life	Turn on and run system for target battery life, see if it runs out of power.	Hours	1 Hour or more	None	FP	Dylan	4/18/22	5/26/22	> 2 Hours	System was powered on for two hours during testing, and still operated normally.	Pass
6	Ingress Protection	Remove internal electronics from housings and replace with paper. Spray with moderate amount of water, and toss dust at system. See if either has penetrated housings.	Pass/Fail	No water or dust in system	Water, dust	FP	Dylan	4/18/22	5/26/22	Fail	During ride on trail, dust entered through the ethernet ports and settled on the PCBs (main and peripheral).	Fail
7	Foolproof	Give system to users with provided manual/instructions, see if they run into any issues.	Pass/Fail	100% pass by user testing (no issues)	Customer Survey	FP	Max	4/18/22	6/2/22	100%	20 students of varying bike experience levels surveyed, all approved.	Pass
8	Maximum Recording Storage	Check maximum storage capacity of SD card	Gigabytes	8 gb or more	None	SP	Max	4/18/22	4/18/22	32 gb	Depends on micro SD card used.	Pass
9	Mounting Universality	Attempt to attach system to variety of bikes.	Pass/Fail	System fits on 100% of bikes	Variety of bikes	SP/FP	Theo	4/18/22	5/26/22	Fail	Passed on all standard road and mountain bikes tested. Failed on electric bikes which have thicker frames.	Fail
10	Aesthetics	Survey potential customers, asking if they find the system visually appealing	Pass/Fail	Over 80% Approval	Customer Survey	FP	Ronan	4/18/22	6/2/22	95%	20 students of varying bike experience levels surveyed, all but one approved.	Pass
11	Suspension Tuning Recommendation	Test the system on a mountain biking trail, adjust according to tuning recommendations, and ride again.	Trail Time	Over 5% faster	Bike trail, bike	FP	Max	4/18/22	6/2/22	Fail	Scope of project changed, focused became on manufacturing and adding sensors rather than developing complex metrics and testing extensively.	Fail

Design Verification Plan & Report (DVP&R)

G - Risk Assessment

Senior Design Project F11 Risk Assessment

2/16/2022

designsafe Report

Application: Senior Design Project F11 Risk Assessment Analyst Name(s): Dylan Ruiz, Max Ringrose, Ronan Schaffer, Theo Philliber

Description: This assesment shows the risks and hazards when using our product. Does not include the regular hazards when riding a mountainbike because our product is not related to the riding portion. Company: Cal Poly San Luis Obispo

Product Identifier: Facility Location: San Luis Obispo, CA

Assessment Type: Detailed

Limits:

Sources: personnel experiences, ANSI B11 standards, assembly drawings W-Z

Risk Scoring System: ANSI B11.0 (TR3) Two Factor

Guide sentence: When doing [task], the [user] could be injured by the [hazard] due to the [failure mode].

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment		Risk Reduction Methods /Control System	Final Assessment		Status / Responsible /Comments /Reference
			Severity Probability	Risk Level		Severity Probability	Risk Level	
1-1-1	Mountain Bike Rider normal operation	slips / trips / falls : User Interference with Wires / Housings User does not install product correctly, wires and housing are in the path of the user when riding a mountain bike.	Moderate Likely	Medium	Include instruction manual giving step by step instructions on how to install device, other design change, fixed enclosures / barriers	Moderate Unlikely	Low	TBD Theo
1-1-2	Mountain Bike Rider normal operation	fire and explosions : Battery Explosion Batteries explode when impacted with high force	Serious Unlikely	Medium	Create the housing with a sturdy material (we will just be using PLA) and incease thickness of housing, warning label(s)	Serious Remote	Low	Complete [2/16/2022] Ronan



Mounting Universality Test



By Team F11 – MTB DAQ

Theo Philliber

Dylan Ruiz

Ronan Shaffer

Max Ringrose

Created: March 7th, 2022

Revision: 2



Purpose

To ensure our design for the mounting apparatus will effectively apply to any bike frame, without the effect of distorting data.

Test Equipment Required

- Helmets for rider and attendees
- Rider for each bike
- Stopwatch
- Tape
- Calipers

Hazards

Most hazards will be bike-riding related. All attendees will be positioned uphill from the rider, except the person taking the time. The person taking the time will stand at a comfortable distance from the bike path.

Safety Issues	Responses
Rider crashes bike while riding.	While this is a possibility, the rider will be experienced and will be wearing the proper safety gear.
Rider crashes into others.	The people conducting the tests will not stand in the path of the rider, and the one taking the time at the bottom of the downhill segment will be standing at a minimum of 20 feet from the path.
Wires interfere with the rider.	The wires will be tied to the frame of the bike, allowing enough clearance between the rider's natural pedal path and the wires.

Procedure

- 1) Gather ten different bike models and riders (Friends and people from bike club).
- 2) Go to a short, dedicated segment on a bike trail.
- 3) Set all bikes' suspension settings to similar settings (max rebound and compression dampening)
- 4) Mount the DAQ to the water bottle boss and the accelerometers to both the rear axle and front axle on one of the bikes.
- 5) Mark the position of the Accelerometers using tape.
- 6) Coordinate start of segment riding with stopwatch person. Press the Record button on the DAQ.
- 7) Driver rides segment to finish at ~ 5 sec.
- 8) Press the Record button on the DAQ to stop recording. Stopwatch person inspects the accelerometer's position from the tape using calipers. Records the movement of the accelerometer from initial position. Repeat steps 4-7 to sample each bike.
- 9) Analyze the data, accelerations should be within the same magnitude of each other.



Results

Pass:

- Accelerometer displacement must be less than .2in
- Data between bike models should be reasonably similar

Fail:

- Accelerometer displacement is greater than .2in
- Data between bike models is distorted (RMS values of acceleration should be within +/- 1g of control data)

Test Dates:

Test Results:

Table 1. Accelerometers Offset

Bike Model (Insert Bike models here)	Front Axle Accelerometer Displacement (in)	Rear Axle Accelerometer Displacement (in)

Customer Survey Procedures



By Team F11 – MTB DAQ

Theo Philliber

Dylan Ruiz

Ronan Shaffer

Max Ringrose

Created: April 5th, 2022
Revision: 1

Purpose

The purpose of this test is to validate that the DAQ meets the subjective criteria of ‘aesthetics’ and ‘foolproof-ness’. These will both be performed by surveying people to represent customers.

Scope

The scope of this protocol is to validate the aesthetic appeal and ease-of-use of the DAQ unit. Since both specifications will be performed by directly surveying customers, they will be performed at the same time and are laid out together in this document. The ease-of-use specification will be tested by providing users with a basic manual and the DAQ and letting them operate it. The aesthetics will be a numerical scoring by the users after they have tested it out.

Equipment

The following list requires all equipment necessary to complete this test protocol.

- DAQ Unit (Full Prototype)
- People to survey

Hazards

There are no potential hazards to this procedure.

PPE Requirements

- None

Facility

This test will take place anywhere people agree to meet on an individual basis.

Procedure

1. EASE OF USE:
 - a. Meet with a person who agreed to be interviewed (hereafter referred to as the user)
 - b. Provide user with operating manual and DAQ.
 - c. Allow user to attempt to use DAQ system, providing minimal feedback or guidance.
 - d. Afterwards, ask user of any issues they had operating the device, and ask to rank the ease-of-use on a 1-10 scale.
2. AESTHETICS
 - a. After user has gotten familiar with operating the system, ask them to rank the aesthetic appeal of the device on a 1-10 scale.

Results

- Ease-of-Use Pass Criteria: No major user issues, average rating of 8 or higher.
- Aesthetics Pass Criteria: Average rating of 8 or higher.
- Number of Samples: 3

Test Date:

Test Results:

User #	Ease of Use	Aesthetics	Comments
1	10	10	-
2	10	9	Dislikes cables
3	10	9	Cables too long
4	10	10	-
5	10	9	Profile too big
6	10	10	Likes main DAQ enclosure
7	10	10	Likes main DAQ enclosure
8	10	10	-
9	10	10	Likes main DAQ enclosure
10	10	9	Cables too long
11	10	8	Cables too long
12	10	10	-
13	10	10	-
14	10	10	-
15	10	9	Dislikes sensor housing prints, too 'blocky'
16	10	10	-
17	10	9	Cables too long
18	10	9	Cables too long
19	10	10	Likes main DAQ enclosure
20	10	9	Cables too long
AVG:	10	9.5	Nobody had issues with operating or connecting the device. Everybody liked the aesthetics, with the only common complaint being the length/size of the cables.

Ingress Protection Validation Test



By Team F11 – MTB DAQ

Theo Philliber

Dylan Ruiz

Ronan Shaffer

Max Ringrose

Created: March 7th, 2022

Revised: April 21st, 2022

Revision: 2

Purpose

To evaluate the extent to which the housings protect the sensitive electronic components contained inside.

Scope

This test is for the waterproof/dustproof feature of the housings

Equipment

The following list requires all equipment necessary to complete this test protocol.

- 3D printed housings
- Mock ethernet cable
- Paper or tissue
- Tape
- Water
- Dirt/dust

Hazards

N/A

PPE Requirements

None

Facility

Teammate's backyard

Procedure

- 1) Remove electronics from housings
- 2) Insert tissue paper in housing
- 3) Plug mock ethernet cable into port
- 4) Splash water by hand or spray water with bottle or hose at housing, especially at port opening
- 5) Remove tissue paper and inspect for water marks
- 6) Replace with fresh tissue paper, surrounded with tape, sticky side facing outwards
- 7) Plug mock ethernet cable into port
- 8) Throw dirt at housing, especially at port opening
- 9) Remove tape and inspect for dirt/debris accumulation



10) Write down notes from visual inspection with a 0-10 rating for ingress protection

0 – Water: Soaking wet tissue paper, lots of water inside

Dirt/Debris: Tape is covered with dirt/dust

10 – Water: Bone dry tissue paper, no trace of moisture

Dirt/Debris: Perfectly clean tape, no trace of dirt or debris

Results

Pass Criteria: 9/10 for both water and dirt

Fail Criteria: 8/10 or lower

Conduct each test 3 times for both main and auxiliary housings to ensure accuracy/repeatability of result

Test Date(s):

Test Results:

Performed By:

Battery Life Validation Test



By Team F11 – MTB DAQ

Theo Philliber

Dylan Ruiz

Ronan Shaffer

Max Ringrose

Created: April 5th, 2022
Revision: 2

Purpose

The purpose of this test protocol is to validate the battery life of the DAQ system during operation. Specifically, to verify that the duration of operation on a single full charge meets specification.

Scope

The scope of this protocol is to validate the battery life of the DAQ system during operation. The life of the battery will be estimated by measuring the current and voltage used by the system and calculating the power from this. This will be compared with the battery capacity to calculate how long the DAQ will run on a single charge.

Equipment

The following list requires all equipment necessary to complete this test protocol.

- DAQ Unit
- Multimeter
- Jumper wires

Hazards

The hazards of this test are entirely based on the batteries. Electric shock can occur, but will not likely be harmful at the low current and voltages supplied. The other risk could be from puncturing the pouch-style batteries, which can then become explosive. Care should be taken with the batteries to not puncture them.

Hazard	Danger Level	Likelihood	Preventative Action
Battery Puncture	Dangerous	Low	Following the procedures should have little to no possibility of puncturing the batteries. If the batteries aren't easily going back in when closing the case, take them back out and reseat the wires so they fit better.
Electric Shock	Low	Medium	Power system off before making changes to wiring or measuring the current and voltage.

PPE Requirements

- Safety Glasses

Facility

This test will take place in the ME 305/405 Laboratory, or anywhere with a multimeter.

Procedure

1. VOLTAGE MEASUREMENT:
 - a. Set the multimeter to voltage mode.
 - b. With the DAQ off, connect the multimeter probes to the positive and negative terminals of the battery.
 - c. Power the DAQ and mark the voltage.
 - d. Try other operations on the DAQ (start/stop recording, etc.) and observe (if any) changes in power consumption from the DAQ. Power on and off DAQ twice and observe changes.
2. CURRENT MEASUREMENT:
 - a. Power the DAQ off and set the multimeter to current mode.
 - b. Disconnect the battery from the PCB board via the quick-connect jack.
 - c. Plug a jumper wire from the ground terminal of the quick-connect to ground terminal on PCB.
 - d. Connect multimeter wire to positive terminal of battery and to PCB.
 - e. Power the DAQ and read the current.
 - f. Power on and off DAQ twice more and observe any changes in current.



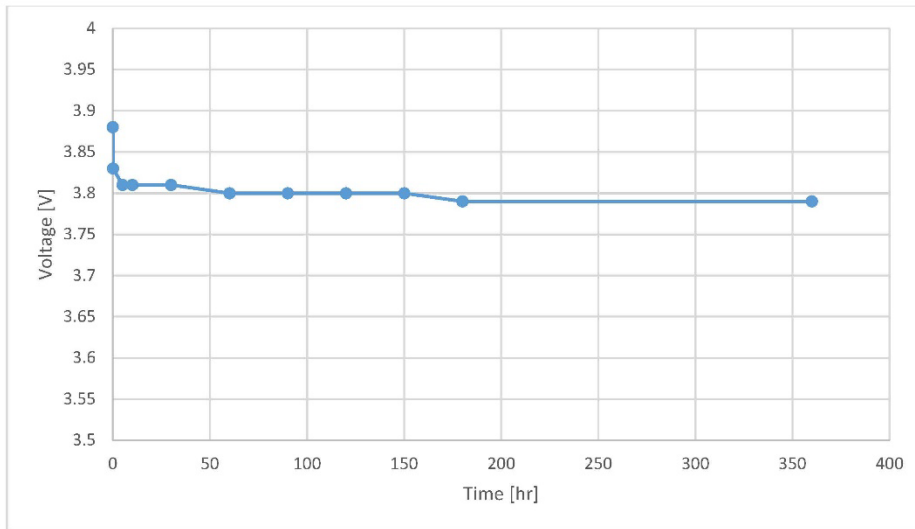
Results

- Pass Criteria: Voltage stays above 3.5V for at least one hour of continuous operation.

Test Date: 5/9/22

Test Results: PASS

Time (mins)	Voltage (V)
0	3.88
0.1	3.83
5	3.81
10	3.81
30	3.81
60	3.80
90	3.80
120	3.80
150	3.80
180	3.79
360	3.79



Based on the data collected in this test, the battery will last for far longer than one hour. It loses some voltage immediately, but then stays mostly consistent for our 6 hour test range, far exceeding our specification of over 3.5V for 1 hour.

Hall Effect Validation Test



By Team F11 – MTB DAQ

Theo Philliber

Dylan Ruiz

Ronan Shaffer

Max Ringrose

Created: March 7th, 2022
Revision: 2



Purpose

The purpose of this test protocol is to validate the speed collected by the Allegro MicroSystems APS12205LUAA hall effect sensor is accurate and precise for all conditions of riding.

Scope

The scope of this protocol is to validate the hall effect on a benchtop truing stand using a calibrated rotational speed sensor. The results of the speed collected by the hall effect will be compared with the calibrated rotational speed sensor. The goal of this test is to develop code that will provide speed accuracy of $\pm 5\%$ using the hall effect sensor as the independent variable and the rotational speed sensor as the control. The dependent the code constant of spoke magnet radius r will be modified until trials achieve the accuracy criteria.

Equipment

The following list requires all equipment necessary to complete this test protocol.

- 29" MTB Wheel
- 15x110 MTB through-axle
- Wheel Truing Stand
- 2 C-clamps
- Wahoo rpm cycling speed sensor
- MTB DAQ Front sensor and main DAQ
- Spoke Magnet
- Drill Adapter
- Cordless drill

Hazards

The hazards of this test include the wheel rotating at high speeds so stay clear of the spokes of the wheel. Ensure that the axle is secured in the truing stand and the truing stand is secured to the table. Make sure the drill is held firmly and to use lower speeds. If something is to be tangled in the wheel, immediately stop the drill.

PPE Requirements

- Safety Glasses

Facility

This test will take place in the ME 305/405 Laboratory.

Procedure

- 1) Using the 2 C-clamps, attach the truing stand to a desktop.
- 2) Attach the 29" MTB wheel to the Truing stand using the through-axle and end nuts.
- 3) Attach the drill adapter to the axle and secure the drill to the adapter.
- 4) Begin rotational trials. Record speeds every 10 seconds for 2 minutes.

Note: Stay clear of the spinning wheel



Results

- Pass Criteria: $\pm 5\%$ percent error of rotational speed sensor
- Number of Samples: 5 tests, 2 minutes long

Test Date:

Test Results:

Hall Effect
Rotational Speed Sensor

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
Speed #1					
Speed #2					
Speed #3					
Speed #4					
Speed #5					
Speed #6					
Speed #7					
Speed #8					
Speed #9					
Speed #10					
Speed #11					
Speed #12					

I – iBOM (Indented Bill of Materials)

Indented Bill of Material (iBOM)											
Assy Level	Part Number	Descriptive Part Name				Qty	Part Cost	Total Cost	Source	URL	More Info
	Lvl0	Lvl1	Lvl2	Lvl3	Lvl4						
0	1.0.0.0	Final Assy									
1	1.1.0.0	----- Main DAQ									
2	1.1.1.0	-- Housing Assy									
3	1.1.1.1			-- Housing	1			custom		3D Printed in PLA	
3	1.1.1.2			-- Bolts	2	\$ 0.31	\$ 0.62	McMaster	https://www.mcm.com/M5X0.8-35MM-LONG-HEX-CAP-SCREW		
3	1.1.1.3			-- Rubber Pad	1	\$ 8.99	\$ 8.99	Amazon	https://www.amazon.com/3M-Self-Stick-Rubber-Anti-Skid-Pad		
3	1.1.1.4			-- OneUp Strap	1	\$ 16.50	\$ 16.50	Amazon	https://www.amazon.com/OneUp-Components-EDC-Gear-Straps-Grey-Pair		
2	1.1.2.0			-- Power Switch	1	\$ 1.96	\$ 1.96	Digikey	https://www.digikey.com/Power-Switch-SPST-10A-AC-125V-Panel-Mount-Snap-In		
2	1.1.3.0			-- Quick Connect Cable Cramp	4	\$ 1.61	\$ 6.44	Digikey	https://www.digikey.com/CONN-QC-RCP-14-18AWG-0.187		
2	1.1.4.0			-- Receptacle Sleeve	4	\$ 0.18	\$ 0.72	Digikey	https://www.digikey.com/CONN-RCP-SLEEVE-0.187-1POS-CLR		
2	1.1.5.0			-- Ribbon Cable	1			Digikey	https://www.digikey.com/CABLE-ASSEM-05-10POS-F-F-2		
3	1.1.6.0			-- Ethernet Cables	2	\$ 5.05	\$ 10.10	McMaster	https://www.mcm.com/3ft-Ethernet-Cord-RJ45		
2	1.1.7.0			-- Thread Locker	0.01	\$ 15.35	\$ 0.15	McMaster	https://www.mcm.com/ADJUSTABLE-THREAD-LOCKER-0.34-OZ-CAN		
2	1.1.8.0			-- PCB Assy							
3	1.1.8.1			-- PCB	1	\$ 54.00	\$ 54.00	JLC PCB		Custom printed circuit board for main DAQ	
3	1.1.8.2			-- Gyroscope (MPU-3050)	1	\$ 8.26	\$ 8.26	Digikey	https://www.digikey.com/GYROSCOPE-PC-24-QFN-4x4		
3	1.1.8.3			-- Batteries Assy							
4	1.1.8.3.1			-- 370 mAh Battery	4	\$ 9.49	\$ 37.96	Digikey	https://www.digikey.com/3.7V-Lithium-Polymer-Battery-Rechargeable-Secondary-370mAh		
4	1.1.8.3.2			-- 1.2 Ah Battery	1	\$ 9.95	\$ 9.95	Digikey	https://www.digikey.com/3.7V-Lithium-Ion-Battery-Rechargeable-Secondary-1.2Ah		
3	1.1.8.4			-- Accelerometer (ADXL375)	1	\$ 11.24	\$ 11.24	Arrow	https://www.arrow.com/Accelerometer-Triple-250g-2.5V/3.3V-14-Pin-LGA-T/R		
3	1.1.8.5			-- Accel Breakout Board	1	\$ 7.79	\$ 7.79	DigiKey	https://www.digikey.com/LGA-14-TO-DIP-14-SMT-ADAPTER		
3	1.1.8.6			-- Ferrite Bead	1	\$ 0.26	\$ 0.26	DigiKey	https://www.digikey.com/30-Ohms-100-MHz-1-Power-Line-Ferrite-Bead-0603-6A-8mOhm		
3	1.1.8.7			-- 0.1 uF Capacitor	6	\$ 0.15	\$ 0.90	DigiKey	https://www.digikey.com/0.1-uF-10%50V-Ceramic-Capacitor-X7R-1206		
3	1.1.8.8			-- 1 uF Capacitor	2	\$ 0.13	\$ 0.26	Digikey	https://www.digikey.com/product-detail/en/kemet/C0805C105K4RACU/399-1284-1-ND/416060		
3	1.1.8.9			-- 2.2 uF Capacitor	2	\$ 0.25	\$ 0.50	Digikey	https://www.digikey.com/product-detail/en/kemet/C0805C225K8RACAU/399-6951-1-ND/3314459		
3	1.1.8.10			-- 4.7 uF Capacitor	3	\$ 0.13	\$ 0.39	Digikey	https://www.digikey.com/product-detail/en/kemet/C0805C475K9PACU/399-3134-1-ND/551639		
3	1.1.8.11			-- 10 uF Capacitor	1	\$ 0.18	\$ 0.18	DigiKey	https://www.digikey.com/10-uF-10%25V-Ceramic-Capacitor-X5R-0805		
3	1.1.8.12			-- 2 pF Capacitor	2	\$ 1.29	\$ 2.58	Digikey	https://www.digikey.com/product-detail/en/kemet/CBR08C209BAGAC/399-8701-1-ND/3480301		
3	1.1.8.13			-- 24 pF Capacitor	2	\$ 1.49	\$ 2.98	Digikey	https://www.digikey.com/product-detail/en/kemet/CBR08C240JAGAC/399-8758-1-ND/3481277		
3	1.1.8.14			-- 2200 pF Capacitor	1	\$ 0.02	\$ 0.02	DigiKey	https://www.digikey.com/2200-pF-10%50V-Ceramic-Capacitor-X7R-0402		
3	1.1.8.15			-- Schottky Diode	1	\$ 0.35	\$ 0.35	Digikey	https://www.digikey.com/DIODE-SCHOTTKY-20V-1A-SOD123W		
3	1.1.8.16			-- Mini-B USB Jack	1	\$ 1.08	\$ 1.08	Digikey	https://www.digikey.com/SMT-USB-Jack-for-USB-Mini-B-cable		
3	1.1.8.17			-- Quick Connect 0.187"	1	\$ 0.23	\$ 0.23	Digikey	https://www.digikey.com/product-detail/en/molex/0197084011/WM6456-ND/459342		
3	1.1.8.18			-- RJ45_SURFACE_MOUNT	2	\$ 1.60	\$ 3.20	Digikey	https://www.digikey.com/SMT-RJ45-Jack-for-ethernet-cable		
3	1.1.8.19			-- Blue LED	1	\$ 0.18	\$ 0.18	Digikey	https://www.digikey.com/LED-BLUE-CLEAR-0805-SMD		
3	1.1.8.20			-- Green LED	1	\$ 0.18	\$ 0.18	Digikey	https://www.digikey.com/LED-GREEN-CLEAR-0805-SMD		
3	1.1.8.21			-- Red LED	1	\$ 0.18	\$ 0.18	Digikey	https://www.digikey.com/LED-RED-CLEAR-0805-SMD		
3	1.1.8.22			-- Yellow LED	1	\$ 0.18	\$ 0.18	Digikey	https://www.digikey.com/LED-YELLOW-CLEAR-0805-SMD		
3	1.1.8.23			-- LPO Charger	1	\$ 0.56	\$ 0.56	Digikey	https://www.digikey.com/IC-CONTROLLER-LI-ION-4.2V-SOT23-5		
3	1.1.8.24			-- 100 KΩ Resistor	5	\$ 0.10	\$ 0.50	Digikey	https://www.digikey.com/RES-SMD-100K-OHM-1%1/3W-0805		
3	1.1.8.25			-- 10 KΩ Resistor	1	\$ 0.50	\$ 0.50	Digikey	https://www.digikey.com/RES-SMD-10K-OHM-0.1%1/4W-0805		
3	1.1.8.26			-- 4.7 KΩ Resistor	5	\$ 0.10	\$ 0.50	Digikey	https://www.digikey.com/CR-GCQ-0805-4K7-1%1		
3	1.1.8.27			-- 560 Ω Resistor	6	\$ 0.10	\$ 0.60	Digikey	https://www.digikey.com/RES-SMD-560-OHM-1%1/8W-0805		
3	1.1.8.28			-- 22 Ω Resistor	2	\$ 0.10	\$ 0.20	Digikey	https://www.digikey.com/RES-SMD-22-OHM-1%1/8W-0805		
3	1.1.8.29			-- Reset Switch	1	\$ 0.41	\$ 0.41	Digikey	https://www.digikey.com/SWITCH-TACTILE-SPST-NO-0.05A-12V		
3	1.1.8.30			-- SD Slot	1	\$ 2.45	\$ 2.45	Digikey	https://www.digikey.com/CONN-MICRO-SD-CARD-PUSH-PUSH-R/A		
3	1.1.8.31			-- Microcontroller	1	\$ 11.61	\$ 11.61	Digikey	https://www.digikey.com/STM32F405RGT6-IC-MCU-32BIT-1MB-FLASH-64LQFP		
3	1.1.8.32			-- Voltage Regulator	1	\$ 0.42	\$ 0.42	Digikey	https://www.digikey.com/IC-REG-LINEAR-3.3V-1A-SOT223		
3	1.1.8.33			-- 10-Pin Connector Header	1	\$ 2.78	\$ 2.78	Digikey	https://www.digikey.com/CONN-HEADER-SMD-10POS-1.27MM		
3	1.1.8.34			-- 12 MHz Clock	1	\$ 0.50	\$ 0.50	Digikey	https://www.digikey.com/CRYSTAL-12.0000MHZ-18PF-SMD		
3	1.1.8.35			-- 32.768 KHz Clock	1	\$ 0.77	\$ 0.77	Digikey	https://www.digikey.com/CRYSTAL-32.7680KHZ-6PF-SMD		
3	1.1.8.36			-- Jumper	2	\$ 0.31	\$ 0.62	Digikey	https://www.digikey.com/JUMPER-W/TEST-PNT-1X2PINS-2.54MM		
3	1.1.8.37			-- Battery Connector	2	\$ 1.86	\$ 3.72	Digikey	https://www.digikey.com/CONN-HEADER-SMD-R/A-2POS-2MM		
3	1.1.8.38			-- 3 Pin Header	4	\$ 0.13	\$ 0.52	Digikey	https://www.digikey.com/CONN-HEADER-VERT-3POS-2.54MM		
3	1.1.8.39			-- 2K Ω Resistor	10	\$ 0.47	\$ 4.74	Digikey	https://www.digikey.com/RES-SMD-2K-OHM-0.1%1/4W-0805		
3	1.1.8.40			-- 44.2 Ω Resistor	6	\$ 0.64	\$ 3.84	Digikey	https://www.digikey.com/RES-SMD-44.2-OHM-0.1%1/4W-0805		

J – Total Expenditures

Team F11 MTB DAQ Senior Project Budget

Item Description	Quantity	Vendor	Vendor Part No.	F11 Part No.	Cost	S&H+ Tax	Purchase Method	Account	Date Purchased	Location
Al. Enclosure	2	Digi-Key	HM890-ND	1.0.0.0.0	\$ 41.32	\$ 8.61	Reimbursement	RMS	4/21/2022	Bonderson
Jumper	2	Digi-Key	732-2678-ND	1.1.8.36.0	\$ 0.62					
Switch Rocker	2	Digi-Key	EG1526-ND	1.1.2.0.0	\$ 3.92					
Wire Sleeve	8	Digi-Key	2-170823-8-SI-ND	1.1.4.0.0	\$ 1.68					
Conn. Receptor	8	Digi-Key	A110952CT-ND	1.1.3.0.0	\$ 1.68					
Rbn Cable	2	Digi-Key	SAM8899-ND	1.1.5.0.0	\$ 32.30					
Conn. Tab	4	Digi-Key	WM14275CT-ND	1.1.8.17.0	\$ 0.52					
Conn. Mod Jack	10	Digi-Key	RJCSE538001CT-ND	1.1.8.18.0	\$ 16.31					
Conn. Head 2pos	4	Digi-Key	3-292173-2-SI-ND	1.1.8.37.0	\$ 3.72					
1 uF Capacitor	12	Digi-Key	399-C0805C105K4RAC7800CT-ND	1.1.8.8.0	\$ 1.25					
4.7 uF Capacitor	10	Digi-Key	311-3422-1-ND	1.1.8.10.0	\$ 1.18					
2 pF Capacitor	10	Digi-Key	399-8701-1-ND	1.1.8.12.0	\$ 10.21					
24 pF Capacitor	10	Digi-Key	399-8758-1-ND	1.1.8.13.0	\$ 11.78					
2.2 uF Capacitor	10	Digi-Key	587-3421-1-ND	1.1.8.9.0	\$ 1.49					
10 uF Capacitor	16	Digi-Key	399-11939-1-ND	1.1.8.11.0	\$ 2.91					
Schottky Diode	4	Digi-Key	1727-5191-1-ND	1.1.8.15.0	\$ 1.76					
USB Mini B RCPT	3	Digi-Key	WM5461CT-ND	1.1.8.16.0	\$ 4.26					
Conn. Head 3pos	4	Digi-Key	732-5316-ND	1.1.8.38.0	\$ 0.52					
Ferrite Bead 120	8	Digi-Key	732-1613-1-ND	1.1.8.6.0	\$ 1.76					
Blue LED	4	Digi-Key	732-4982-1-ND	1.1.8.19.0	\$ 0.72					
Green LED	4	Digi-Key	732-4986-1-ND	1.1.8.20.0	\$ 0.72					
Red LED	4	Digi-Key	732-4985-1-ND	1.1.8.21.0	\$ 0.72					
Yellow LED	4	Digi-Key	1516-1089-1-ND	1.1.8.22.0	\$ 1.28					
320 mAh Battery	10	Digi-Key	1908-LP402535JU+PCM+2WIRES50MM-ND	1.1.8.3.1	\$ 86.63					
100k Resistor	14	Digi-Key	A126415CT-ND	1.1.8.24.0	\$ 1.72					
2k Resistor	10	Digi-Key	A110572CT-ND	1.1.8.39.0	\$ 4.74					
4.7k Resistor	14	Digi-Key	A129757CT-ND	1.1.8.26.0	\$ 0.81					
560 Resistor	16	Digi-Key	A126376CT-ND	1.1.8.27.0	\$ 1.18					
22 Resistor	10	Digi-Key	A126352CT-ND	1.1.8.28.0	\$ 0.56					
Tactile Switch	2	Digi-Key	EG4375CT-ND	1.1.8.29.0	\$ 0.95					
MicroSD Conn	2	Digi-Key	WM24066CT-ND	1.1.8.30.0	\$ 7.32					
12 MHz Crystal	2	Digi-Key	XC1289CT-ND	1.1.8.34.0	\$ 0.90					
Yellow Indic.	2	Digi-Key	492-1576-ND	1.1.9.4.0	\$ 5.88					
Green Indic.	2	Digi-Key	492-1571-ND	1.1.9.5.0	\$ 5.88					
44.2 Resistor	6	Digi-Key	A110413CT-ND	1.1.8.40.0	\$ 3.84					
80.6 Resistor	4	Digi-Key	A110438CT-ND	1.1.8.41.0	\$ 2.56					
Switch Button	2	Digi-Key	EG5385-ND	1.1.9.11.0	\$ 19.78					
0.1 uF Capacitor	18	Digi-Key	399-C0805C104K3RAC7800CT-ND	1.2.2.8.0	\$ 1.64					
10 uF Capacitor	18	Digi-Key	1276-6456-1-ND	1.2.2.9.0	\$ 1.58					
Accelerometer	5	Digi-Key	505-ADXL375BCCZ-ND	1.3.2.1.0	\$ 59.45					
Gyroscope	1	Digi-Key	1428-1001-1-ND	1.1.8.2.0	\$ 9.49					
IC Gate	5	Digi-Key	296-SN74AHCLG32DCK3CT-ND	1.2.2.5.0	\$ 2.20					
IC Volt Reg	2	Digi-Key	488-NCP5661MNB3T2GCT-ND	1.1.8.32.0	\$ 3.18					
10pos Conn.	4	Digi-Key	1175-2649-1-ND	1.1.8.33.0	\$ 3.48	\$ 35.47	Reimbursement	RMS	4/19/2022	Bonderson
32.768 kHz Crystal	2	Digi-Key	2388-SWS8346D16-32.768K-ND	1.1.8.35.0	\$ 3.96	\$ 9.34	Reimbursement	RMS	4/19/2022	Bonderson
Gyroscope	1	Digi-Key	1428-1001-1-ND	1.1.8.2.0	\$ 8.26					
Hall Effect Sensor	1	Digi-Key	620-1964-ND	1.2.2.2.0	\$ 0.98					
320 mAh Battery	4	Digi-Key	1908-LP402535JU+PCM+2WIRES50MM-ND	1.1.8.3.1	\$ 41.76	\$ 12.60	Reimbursement	RMS	2/7/2022	Bonderson
Rubber Pad	1	Amazon	-	1.1.1.3.0	\$ 8.89					
OneUP Strap	1	Amazon	-	1.1.1.4.0	\$ 19.99	\$ 2.53	Reimbursement	JMR	11/4/2021	Bonderson
Solder Stencil	4	JLC	73145000	2.0.0.0.0	\$ 28.00					
PCB	25	JLC	Y9-12	1.1.8.2.0	\$ 21.60	\$ 86.39	Reimbursement	JMR	4/20/2022	Bonderson
Breakout Board	1	Amazon	-	1.1.8.5.0	\$ 12.99	\$ 1.14	Reimbursement	JMR	2/13/2022	Bonderson
MCU	2	Charlie Refvem	-	1.1.8.31.0	\$ 30.00	\$ -	Reimbursement	TJP	4/5/2022	Bonderson
Total					\$ 542.83	\$ 156.08				
Grand Total					\$ 698.91					