
REMOTE TORQUE AND SPEED DATA LOGGER

by

Russell Case and Jeremiah Moreno

A report presented to the
British Columbia Institute of Technology
in partial fulfilment of the requirement for the degree of
Bachelor of Engineering (Mechanical)

Faculty Advisor: Adam Marciniak

Program Head: Mehrzad Tabatabaian

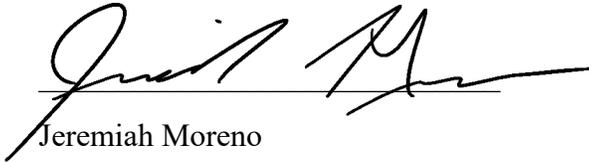
Burnaby, British Columbia, Canada, 2019

© Russell Case, Jeremiah Moreno, 2019

We hereby declare that we are the sole author of this report.

Handwritten signature of Russell Case in black ink, written over a horizontal line.

Russell Case

Handwritten signature of Jeremiah Moreno in black ink, written over a horizontal line.

Jeremiah Moreno

We further authorize the British Columbia Institute of Technology to distribute digitally, or on printed paper, copies of this report, in total or in part, at the request of other institutions or individuals for the purpose of scholarly activity.

Handwritten signature of Russell Case in black ink, written over a horizontal line.

Russell Case

Handwritten signature of Jeremiah Moreno in black ink, written over a horizontal line.

Jeremiah Moreno

Abstract

Baja SAE is an intercollegiate design competition hosted by SAE International, previously known as the Society of Automotive Engineers. At each event, more than 100 teams compete to design, build, and race light off-road vehicles. The British Columbia Institute of Technology is represented by its SAE-affiliated team, BCIT Racing, which has competed in four SAE events since 2015, and is in the early stages of designing and building a third-generation race vehicle.

This paper describes the design, function and use of a wireless torque and rotation data logger device, which attaches to a vehicle's drive axle for the purpose of observing drivetrain loads. The project was proposed by BCIT faculty member Adam Marciniak, in his role as BCIT Racing's faculty supervisor. Upon completion, the data logger prototype will be used by BCIT Racing to determine peak force loads in the gearbox and drivetrain of the team's Baja SAE race vehicle. This data will be used during the design of the team's third-generation vehicle.

The data logger project can be divided into three functional components: the housing, electronics, and software. The housing provides protection for sensitive electronics and sensors, and the software is used to gather sensor data and transmit it to a mobile phone application.

At the time of this report, all sensors have been validated, and data is being successfully captured and stored. Recommendations for future work will be included in the report conclusions.

Acknowledgments

We would like to acknowledge the following people for their contributions to this project:

Adam Marciniak, for his enthusiastic support as a project sponsor, and his willingness to go above and beyond in assisting the development of every aspect of the project. Adam put countless hours of work into his role, assisting with everything from mechanical concepts to charging circuits and mobile applications.

Christopher Townsend, for his seemingly endless energy, spent in his support of BCIT students' work. Chris not only provided training to countless project groups, but also sacrificed his weekends and holidays to provide countless extra hours of machine shop time.

Sanesh Iyer, for his contributions as a BCIT Racing team captain, and for his continued support of the team after graduation. Having designed the team's second-generation gearbox, Sanesh helped provide valuable information regarding its design process, and what assumptions were made at that time.

Jason Brett, for sharing his parts, equipment, and expertise in all manner of electronic wizardry.

Johan Fourie, for his no-nonsense management of the capstone process.

Stephen McMillan, for his help in optimising the design for both additive manufacturing and CNC machining.

Taco Niet, David Romalo, and Dave Stropky, for the time and effort they put into creating and teaching world-class courses on the theory and application of sensors, microcontrollers, and programming. This success of this project directly depended on the concepts and knowledge gained in MECH 8135, ELEX 7345, and MECH 7171.

Dedication

To our friends and families,

Long time, no see!

Contents

Abstract	iii
Acknowledgments	iv
Dedication	v
Contents.....	vi
1. Introduction	1
1.1. Background	1
1.2. Requirements	2
2. Problem Statement	3
2.1. BCIT Racing’s Gearboxes: Designing Around Assumptions	3
2.2. Simulation and Testing	7
3. Theoretical Background and Prior Art	8
3.1. Torque Sensors in Racing Applications	8
3.2. Comparison of Rotational Torque Measurement Systems	9
3.3. Viability of Torsional Strain Gauge for a Baja SAE Vehicle	10
3.4. Measurement of Angular Velocity	12
Inductive Speed Sensors	12
MEMS Accelerometers	13
MEMS Gyroscopes.....	13
4. Design, Testing, and Manufacturing	14
4.1. Concept Generation and Iteration	14
Concept 1: Discrete Speed Sensor	14
Concept 2: Refined Clamshell with Integrated MEMS Gyroscope	15
Concept 3: Reduction of Complexity and Weight	16
Final Design: Rebuilt from the Ground Up	16
4.2. Rapid Prototyping and Iteration of Housing Designs	17
4.3. Sensor and Electronics Validation	18
Gyroscope/Accelerometer Validation.....	18
Torsion Test Validation	19

Mechanical Calibration and Test Procedure	22
Force/Voltage Relation for Torsion	22
Positive and Negative Torsion Measurement	25
4.4. Electrical System Design	26
Wiring Diagrams and Schematics.....	26
Battery System.....	26
PCB Design and Manufacture	27
PCB Manufacturing Procedure.....	28
5. Discussion of Results	29
5.1. Torsion Test Results.....	29
5.2. Eliminating Zero Offset	30
5.3. Battery System Results	31
5.4. PCB Results.....	31
5.5. Android Mobile Application.....	32
6. Conclusion.....	33
6.1. Summary and Assessment of Success.....	33
6.2. Lessons Learned.....	33
6.3. Future Work.....	33
6.4. Design Hurdles.....	34
Faulty Breadboard	34
PCB Manufacturing.....	35
Microcontroller Selection	35
Appendix A: Source Code.....	36
Appendix A.1. Gyroscope Test Script.....	36
Appendix A.2. Torsion Test Script.....	38
Appendix A.3. Multiple Data Point String Bluetooth Test.....	40
Appendix B. Selections from ADS 1115 Datasheet.....	43
Appendix B.1. Overview	43
Appendix B.2. Absolute Maximum Ratings	44

Appendix B.3. Electrical Characteristics	45
Appendix B.4. Programming	47
Appendix B.5. Connecting Multiple I ² C Devices	49
Appendix B.6. Description of Differential Comparator Function	50
Appendix C. Dynamometer Plots	51
Appendix D. ICM-20601 Data Sheets.....	54
Appendix D.1. Guide to Interfacing with ICM-20601 Breakout Board.....	54
Appendix D.2. Gyroscope Configuration	55
Appendix D.3. Low-Power Mode Configuration.....	56
Appendix D.4. Register Map	57
Appendix E. BCIT Racing Request for Proposal	59
Appendix F. Strain Gauge Application Guide.....	61
Appendix G. Wiring Diagrams.....	65
Appendix G.1. Torsion Test Rig	65
Appendix G.3. Battery Charging Circuit	67
Appendix H. Torque Transducer Notes.....	68
Appendix I. 16 Bit ADC Resolution MATLAB Code	69
References	71

Table of Figures

Figure 1: Proposed Location of Torque/Speed Data Logger Module.....	1
Figure 2: The Rear Suspension of BCIT Racing's V2 Vehicle, 2019.....	2
Figure 3: RIT Engine Verification [1].....	3
Figure 4: Hill Climb Free Body Diagram [1].....	4
Figure 5: Moments Before Gearbox Failure Due to Impact in Oregon, 2015.....	4
Figure 6: Former Team Captain Adam Marciniak Repairs the V1 Gearbox in Oregon, 2015.....	5
Figure 7: Field Repair of Failed Weld on V1 Axle in Rochester, 2016.....	5
Figure 8: Full Failure of V1 Axle Shaft in Rochester, 2016.....	6
Figure 9: Kevin Radford Posing with His Victim, 2018.....	6
Figure 10: V1 and V2 Gearbox FEA Results.....	7
Figure 11: Destructive Testing of BCIT Racing Baja Vehicle V2, 2019.....	7
Figure 12: Magnetoelastic Torque Sensor Diagram.....	8
Figure 13: A Cylinder in Torsion, with Detail of Stress Element [6].....	10
Figure 14: Fatigue Life of Strain Gauges.....	11
Figure 15: Rocket Motor Test, MECH 8135 Lab (BCIT, 2018) [8].....	11
Figure 16: Typical Automotive Inductive Speed Sensor [9].....	12
Figure 17: MEMS Gyroscope [10].....	13
Figure 18: Initial Concept (Rev. 1) with Discrete Speed Sensor.....	14
Figure 19: Housing Rev.2: Details Added for Mounting Electronics and Attaching to Axle.....	15
Figure 20: Details of Axle Clamp (Left) and Swappable Access Panel (Right).....	15
Figure 21: Housing Rev.3: Reduced Size, Simplified Seals and Access Port.....	16
Figure 22: Housing Rev. 4: Final Design with Further Optimization of Size and Complexity.....	16
Figure 23: Housing Design Iterations with Test PCB and Clamps.....	17
Figure 24: Rapid Prototyping: 3D Printing to Check Revised Housing.....	17
Figure 25: Gyroscope and Bluetooth Validation Test.....	18
Figure 26: Torsion Test Rig.....	19
Figure 27: Diagram of a Typical Torsion Rosette with Half-Bridge Gauges.....	19
Figure 28: Arduino IDE Serial Plot of Torsion Data.....	20
Figure 29 Torsion Tester Wiring Schematic.....	21
Figure 30: Gain Error.....	22
Figure 31: Voltage Divider Circuit.....	25
Figure 32: Full System Wiring Schematic.....	26
Figure 33 Battery System Wiring Diagram.....	27
Figure 34 Top and Bottom PCB Layout.....	27
Figure 35: PCB in Etching Tank.....	28
Figure 36: Toner Removal in Acetone Bath.....	28
Figure 37: Torsion Calibration Test in Progress.....	29
Figure 38: Strain Gauge Data Label.....	29
Figure 39: Non-Zeroed Torsion Reading.....	30
Figure 40: Torsion Reading 25lb Clockwise.....	30
Figure 41 Battery Charging PCB Layout.....	31
Figure 42: Custom PCBs.....	31
Figure 43: Gyroscope Data Displayed on Grapher App.....	32
Figure 44: Faulty Breadboard.....	34
Figure 45: ESP 32 Microcontroller.....	35
Figure 46: Purdue Dynamometer Reading, 2012 [11].....	51
Figure 47: University of Arkansas Dynamometer Readings, 2012 [12].....	52
Figure 48: RIT Dynamometer Results [1].....	53

1. Introduction

1.1. Background

BCIT Racing is in the process of gathering force load test data to be used in the design of its third-generation Baja SAE race vehicle. While the second-generation (V2) vehicle was significantly lighter than its predecessor, significant work remains in identifying areas of potential weight savings.

Determining peak torque and power loads at the drive axles permits the team to optimize and lighten several sub-systems, including:

- Chassis structure
- Drivetrain mounts
- Brakes
- Axles
- Gearbox

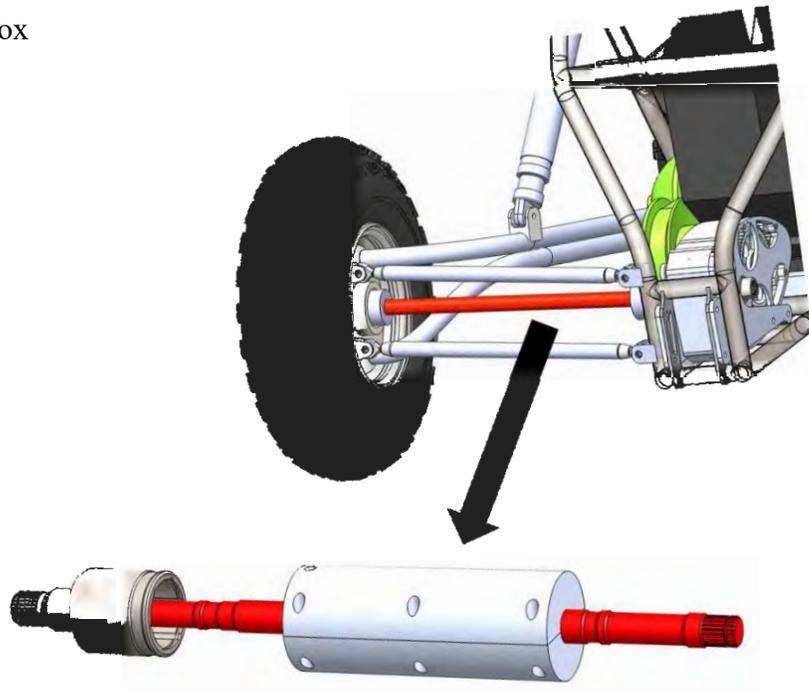


Figure 1: Proposed Location of Torque/Speed Data Logger Module

To save weight while maintaining performance and reliability standards, the vehicle's designers should have an understanding of the peak loads experienced during normal use. These peak loads generally occur when the vehicle is being driven in race-like conditions, and so a need has been identified for the development of a robust and compact torque and speed data logger, that will not interfere with vehicle operation. For convenience, this device should be capable of wirelessly transmitting data for storage and analysis.

1.2. Requirements

This report will document the design and manufacture of a wireless torque and speed data logger for a Baja SAE vehicle. The requirements for this project are outlined in the Request for Proposal in Appendix E.

The RFP lays out the design team's role and responsibilities:

The Data Logger design team will be responsible for conducting relevant research and planning, generating/presenting concepts, identifying required components, and creating a functional prototype.

Additionally, the following design parameters are set out:

- Must measure axle torque and rotation speed
- Must be removable with minimal effort, i.e., under 10 minutes
- Components must be contained in a watertight housing
- Must be calibrated to make accurate measurements upon delivery
- Must permit normal vehicle operation when installed
- Must transmit and store data for later analysis



Figure 2: The Rear Suspension of BCIT Racing's V2 Vehicle, 2019

2. Problem Statement

As of 2019, BCIT Racing has designed and built two gearboxes for its vehicles. Because the team's designers lacked empirical test data, peak loads were estimated, based on engine dynamometer graphs like RIT's results shown in Figure 3, along with rough calculations based on static loads. To allow the team to improve its designs, real-world data must be gathered. This project will specifically focus on designing a tool which will enable BCIT Racing members to gather torque data directly at the drive axles.

2.1. BCIT Racing's Gearboxes: Designing Around Assumptions

When loads are not precisely known, safety factors must be introduced, at a severe weight penalty. The V1 gearbox, for example, weighed 60 pounds without its shifter linkage. In comparison, a commonly used off-the-shelf gearbox, the Dana/Schafer H12 FNR weighs only 34 pounds, and top teams have raced with gearboxes weighing approximately 10 pounds. [1]

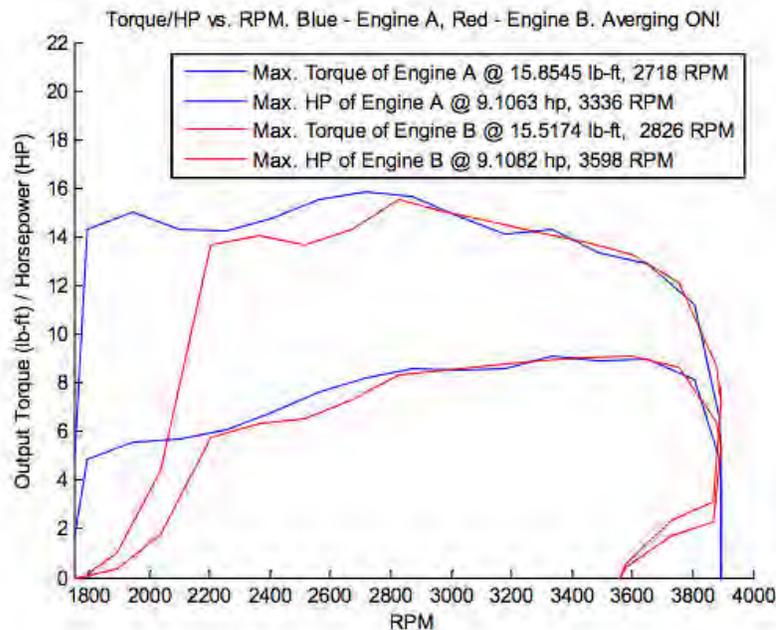


Figure 3: RIT Engine Verification [1]

An example of the gearbox design process being performed without direct empirical data can be seen in Sanesh Iyer's report, *Power Transmission Design: BCIT Baja SAE 2015/17* [1]. Iyer, a former BCIT Racing team captain, based the worst-case V2 gearbox torque capacity on a Baja SAE hill-climb event. This resulted in a calculated maximum torque of approximately 490 ft-lbs. According to the design report *SAE Baja Drivetrain* [2], by Marciniak et al., The V1 gearbox design team estimated the maximum torque as a multiple of the measured engine torque, after multiplication through the gearbox. This resulted in a calculated design torque of approximately 600 ft-lbs.

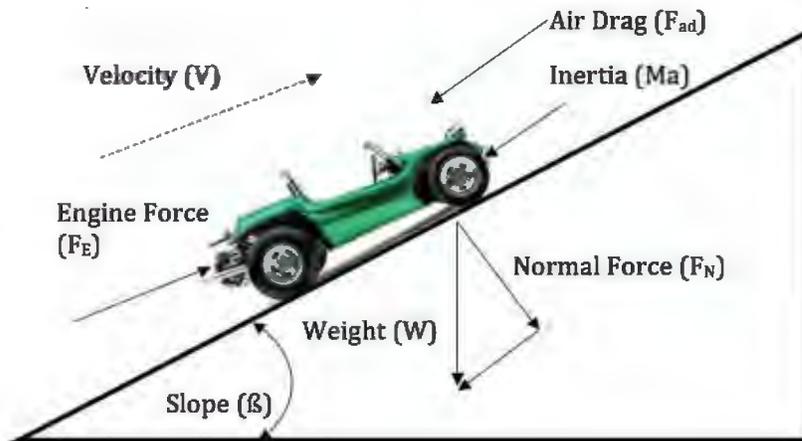


Figure 4: Hill Climb Free Body Diagram [1]

Importantly, that torque value is based on static calculations and does not account for dynamic impact loads. For example, when a vehicle encounters an obstacle, or when there is a relative wheel velocity during a jump landing, instantaneous torques may be much higher than static design loads. These high-load, short-duration scenarios are precisely when a team may encounter sudden part failure, as shown below in Figures 5 through 9.



Figure 5: Moments Before Gearbox Failure Due to Impact in Oregon, 2015



Figure 6: Former Team Captain Adam Marciniak Repairs the V1 Gearbox in Oregon, 2015



Figure 7: Field Repair of Failed Weld on V1 Axle in Rochester, 2016



Figure 8: Full Failure of VI Axle Shaft in Rochester, 2016



Figure 9: Kevin Radford Posing with His Victim, 2018

2.2. Simulation and Testing

Both the V1 and V2 gearbox designers performed finite element analysis to evaluate bearing loads, seen below in Figure 10. However, the simulated loading conditions were based on static calculations, and did not consider dynamic loading. As a result, safety factors must be used to account for the possibility of underestimated loads. For a race vehicle that prioritizes weight reduction, safety factors would ideally approach 1.0, representing a perfect matching of loads and designs. What's more, even with a safety factor in place, without testing there is no guarantee of it being sufficient.

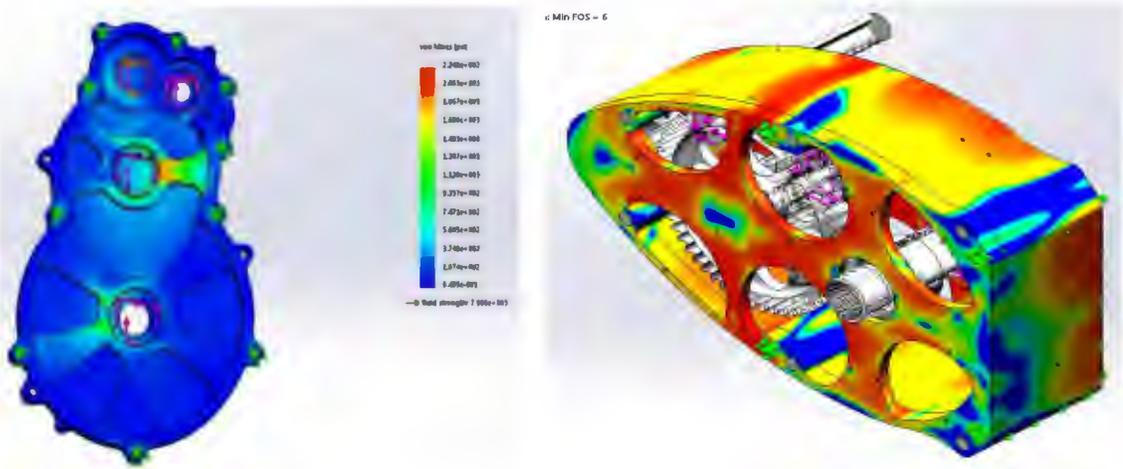


Figure 10: V1 and V2 Gearbox FEA Results

With an absence of load data, BCIT Racing must use methods such as destructive testing to determine whether designs are sufficiently strong. An example of this testing can be seen below, in Figure 11. This test resulted in the destruction of three heim joints and a trailing arm mount. It should be noted that non-destructive testing with load sensors avoids the time expense of repairs following a successful destructive test.



Figure 11: Destructive Testing of BCIT Racing Baja Vehicle V2, 2019

3. Theoretical Background and Prior Art

3.1. Torque Sensors in Racing Applications

In the SAE technical paper *Development of a Magnetoelastic Torque Sensor for Formula 1 and CHAMP Car Racing Applications*, Bitar et al. describe the necessity of a racing torque sensor which is installed on the car, rather than on a test bench. They note the difficulty of fully representing a racing vehicle with a simulation model, and state that such methods “rarely yield fully representative results.” The key performance metric to be measured is “true power” delivered to the wheels. For F1 teams, this can be used in several ways, including determining drag and downforce when comparing expected to actual accelerations, producing data which can inform torque vectoring differential controls, and producing true-to-life dynamometer readings that cannot be simulated even in a wind tunnel. [3]

For an SAE Baja team, the race vehicle’s speed and power is orders of magnitude lower, and this allows the use of a different approach to measurement. In F1 racing, it is advantageous to remove the sensor from the axle itself in order to lower rotational inertia and reduce the impact of the measurement device. This necessitates the use of novel solutions like magnetoelastic torque sensing, which permits torque measurement without directly contacting the vehicle’s drive axle. A diagram of this system can be seen below, in Figure 12.

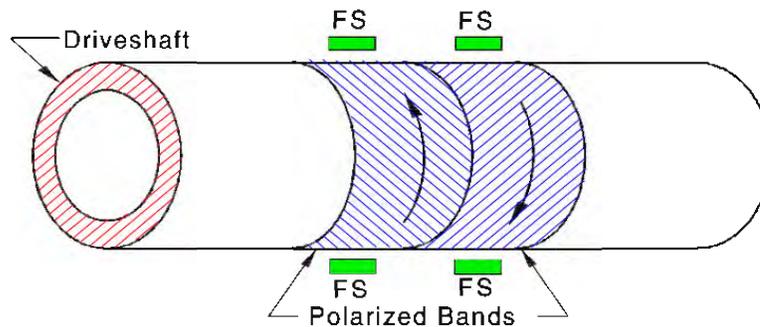


Figure 12: Magnetoelastic Torque Sensor Diagram

F1 vehicles can travel at speeds greater than 200 miles per hour. With wheels sized to the maximum allowable diameter of 470 mm [4], an F1 axle can approach angular speeds of more than 3500 RPM. Baja SAE vehicles, on the other hand, travel at speeds well below 50 miles per hour, and axle shaft angular speeds will rarely exceed 600 RPM. As a result, attaching a sensor package directly to the drive axle becomes more feasible. The primary advantage of keeping the sensor package in direct contact with the axle is that it permits the use of strain gauge or load cell technology – a well-documented, cheap, and popular strain measurement system. The relative advantages of several strain measurement systems will be elaborated on in the next section.

3.2. Comparison of Rotational Torque Measurement Systems

Table 1, below, lists several potential torque measurement methods. Many of these methods are more appropriate for stationary industrial equipment, where sensing modules may remain installed for long periods of time. The final measurement method highlighted in grey has been selected for this project. This method uses a self-contained module to house both data and power electronics, and is not necessarily suitable for a factory as a piece of permanent equipment. However, it may suit a race car that may require axle swapping, and offers few locations for reliably mounting stationary hardware.

Table 1: Comparison of Torque Measurement Methods

Sensor Type	Advantages	Disadvantages
Magnetoelastic [4]	<ul style="list-style-type: none"> • Air gapped • Hard-wired data transmission • Minimal axle modification • Does not add rotational inertia 	<ul style="list-style-type: none"> • Difficult to apply polarized magnetic bands • Techniques not well-documented • More complicated
Strain Gauge on Axle (Slip-Ring Power and Data) [5]	<ul style="list-style-type: none"> • Minimal axle modification • Negligible added rotational inertia • Hard-wired data transmission 	<ul style="list-style-type: none"> • May require sealing with oil • Susceptible to friction wear • Data may be subject to noise due to brush movement
Gimbaled Gearbox with Load Cell [5]	<ul style="list-style-type: none"> • No axle modification • No effect on rotating mass • Hard-wired data transmission 	<ul style="list-style-type: none"> • May not fully isolate torque measurement • May compromise gearbox mounting strength • Mechanically complex
Strain Gauge on Axle (Wireless data, inductive power) [5]	<ul style="list-style-type: none"> • Batteries not located on rotating assembly • No potential for errors caused by brushes 	<ul style="list-style-type: none"> • Requires wireless transmitter to be attached to axle • Complex electronic design
Clamp-on Module with Beam Load Cell [5]	<ul style="list-style-type: none"> • Can be easily attached and detached • No modification to axle • Strain gauge applied to module, not axle 	<ul style="list-style-type: none"> • Adds mass and rotational inertia to system • Clamped rings must be as rigid as possible • Complex and precise machining
Strain Gauge on Axle (Wireless data, battery power)	<ul style="list-style-type: none"> • Self-contained as a module • Reliable power and data transmission • Housing does not require precise machining or high strength 	<ul style="list-style-type: none"> • Highest mass and rotational inertia of all choices • Highest software and firmware complexity of all choices

3.3. Viability of Torsional Strain Gauge for a Baja SAE Vehicle

Moving forward in the design process having selected strain gauges as a potential torque measurement solution, the viability of the strain gauge must be calculated for use on a Baja SAE race vehicle. First, the estimated strain at the point of measurement should be calculated. This will determine whether the strain gauge setup will be sensitive enough, and if so, what amplification gain, and which Wheatstone bridge setup should be used. Second, the fatigue characteristics of the strain gauge itself should be accounted for.

The following calculations estimate a strain (γ) of approximately $79 \mu\text{strain}$ when the gauge is applied to a solid steel shaft of a constant 1-inch diameter, 6.5 inches from the CV joint. The torque value of 180 lb-in is taken as an estimate of engine output from the dynamometer results in Appendix C.

$$\phi = \frac{TL}{JG} = \frac{(180 \text{ lb} \cdot \text{in})(6.5 \text{ in})}{\left(\frac{\pi}{32} \text{ in}^4\right)(11.6E6 \text{ PSI})} = 0.00103 \text{ rad}$$

$$\text{Strain, } \gamma = \frac{\rho\phi}{L} = \frac{(0.5 \text{ in})(0.00103 \text{ rad})}{6.5 \text{ in}} = 79.2 \mu\text{strain}$$

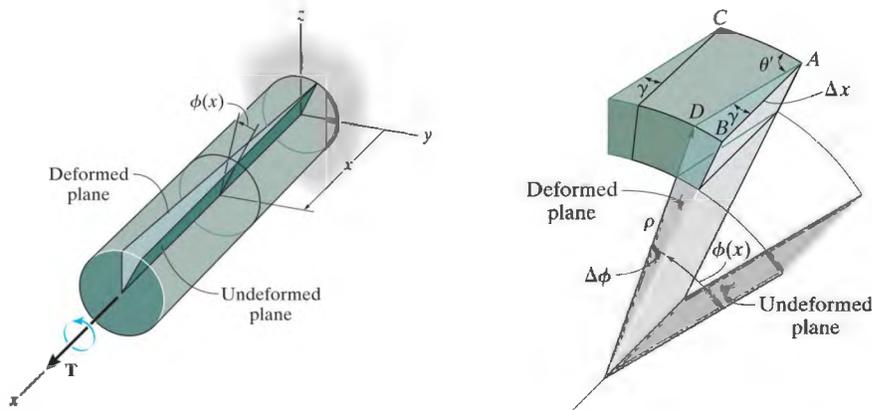


Figure 13: A Cylinder in Torsion, with Detail of Stress Element [6]

The fatigue characteristics of an average WK-type strain gauge is shown on the following page, in Figure 14. [7] Note that the endurance limit shown is approximately $\pm 2000 \mu\text{strain}$, and so the calculated strain produced by a typical Briggs & Stratton Baja SAE engine on a solid 1-inch axle shaft is far below levels that would cause fatigue failure.

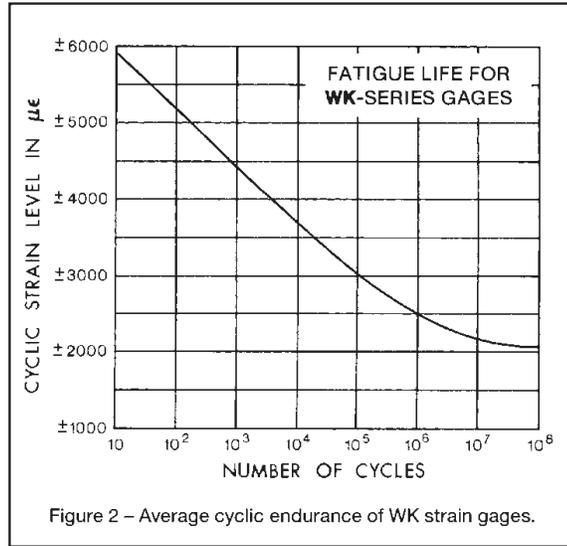


Figure 14: Fatigue Life of Strain Gauges

This magnitude of estimated strain also appears to be appropriate for measurement without extremely sensitive instruments, and a half-bridge circuit coupled with a 16-bit Analog-Digital Converter should provide a wide measurement range and resolution.

For reference, Figure 15 shows the results of a rocket motor test performed by this project team in a MECH 8135 lab, supervised by BCIT instructor Taco Niet and assistant instructor Adam Marciniak. [8] The test was performed with a quarter-bridge gauge configuration, measured a load of 10N producing an estimated 32.3 μ strain, and was recorded by a 12-bit ADC. In comparison, with a higher-amplification half-bridge configuration and a higher-resolution ADC, the low range of torques on the Baja vehicle should be easily measured.

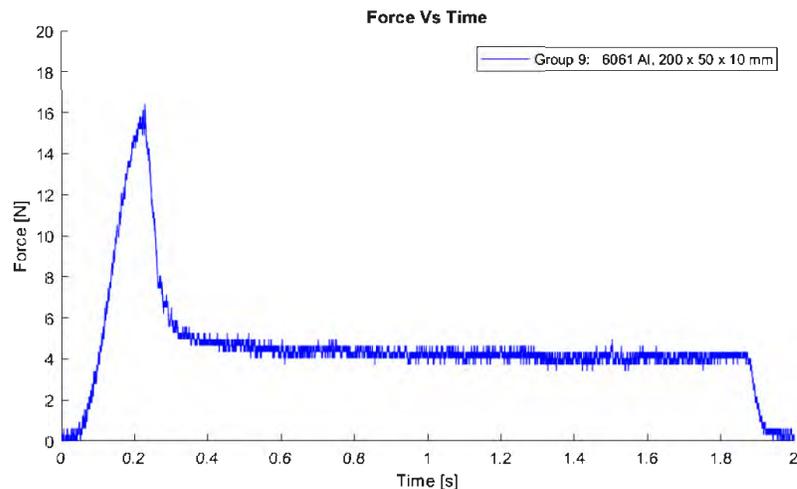


Figure 15: Rocket Motor Test, MECH 8135 Lab (BCIT, 2018) [8]

3.4. Measurement of Angular Velocity

Although angular velocity measurement is not necessary for determining design loads, coupling torque measurements with measured wheel speed can help to illustrate what, exactly, was occurring at a given moment in time. For example, it may be helpful to know if the vehicle was slowing down, speeding up, or landing a jump when a torque spike was detected. Additionally, taking a coupled speed measurement allows the calculation of instantaneous power at every moment, which can be helpful for tuning and analysis.

Inductive Speed Sensors

Most commonly, speed sensing for automotive applications is done with passive or active inductive sensors. Both produce a measurable wave pattern when a tooth on a tone ring passes the sensor, so the sensitivity of the system depends on how fine the tooth pattern on the ring can be made. This is a robust solution that reliably records the wheel speed of the vehicle; however, it does have some disadvantages in the context of this Torque/Speed data logger project.

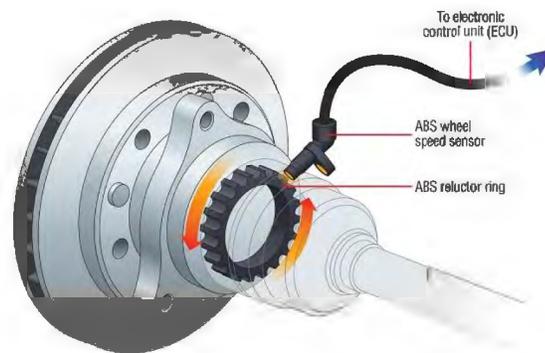


Figure 16: Typical Automotive Inductive Speed Sensor [9]

The first problem with using an inductive sensor is that it should be permanently affixed to the vehicle, either inboard, mounted to the gearbox/differential, or on the outside of the axle, affixed to the wheel hub. This introduces new mechanical complexity to the system, and would require machined mounts and precise alignment between the inboard and outboard components. It would also need to account for the sliding action of the CV joints, and so would most likely have to use a floating mount.

The second major problem is that using a discrete speed sensor introduces a severe complexity penalty to the system's electronics and software, which would otherwise be self-contained, with no external components. The speed sensor module would require power and a data link, meaning it would either need a potentially fragile wired connection to the torsion sensing module, or it would require an independent power supply along with its own wireless connection.

MEMS Accelerometers

One alternative to using an inductive sensor is to use an accelerometer based on Micro-Electro-Mechanical Systems (MEMS) technology. MEMS components are often extremely small, with minimal power consumption and low cost. They have become popular in robotic products like quadcopters, and in consumer electronics like mobile phones. By placing an accelerometer at a known distance from the center axis of the vehicle axle, angular speed can be calculated.

The problem with using an accelerometer as a speed sensor is that any translational accelerations experienced by the vehicle's suspension will have a direct impact on the calculated speed readings taken from the sensor. This effect will be most noticeable when the vehicle encounters large obstacles or high suspension articulation at low speeds. Unfortunately, this is exactly the type of driving scenario expected for a Baja SAE vehicle. The translational effects will be the least noticeable at high speeds, when radial accelerations are of a significantly higher magnitude than translational accelerations.

MEMS Gyroscopes

Fortunately, MEMS gyroscopes are also available, and offer direct angular speed measurement in a tiny sensor package. These MEMS gyroscopes are small chips in which two known masses are oscillated in opposing phase, as seen in Figure 17. This coupled mass system permits the detection of Coriolis forces caused by rotation while ignoring translational accelerations.

One problem with using MEMS gyroscopes is that the most popular sensors are used in drones and small robots for inertial measurement. The maximum rotational speeds experienced in these applications are fairly low, and so the lowest-cost and best-documented sensors like the LSM9DSX series have a maximum measurement speed of 2000 degrees per second (dps). While this would be enough for a Baja SAE vehicle at low-to-medium speeds, a potential downhill top speed of 50 MPH necessitates a sensor that can measure up to 4000 dps.

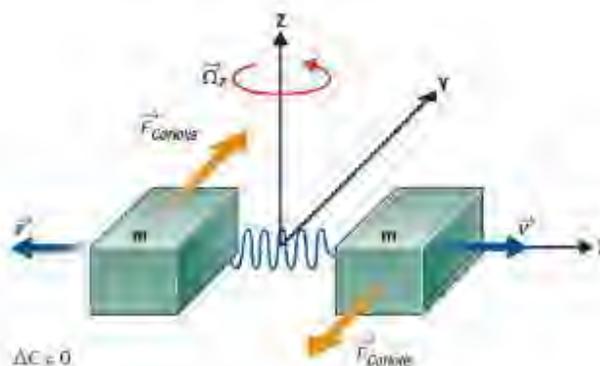


Figure 17: MEMS Gyroscope [10]

4. Design, Testing, and Manufacturing

4.1. Concept Generation and Iteration

The data logger housing must protect the device's sensors and electronics, while remaining light and compact enough to avoid affecting the test vehicle's performance. It should be water-resistant, and robust enough to absorb moderate impacts. Additionally, user serviceability, flexibility, and ease of use must be considered.

Because the device's own rotational inertia is expected to be one of the primary stress sources, the housing's material must be lightweight as well as tough. Acetyl (Delrin) was chosen as a suitable material, however this choice necessitates the use of CNC machining. If we consider the housing to be a wear item that may require replacement, 3D printing with ABS and PLA becomes a viable manufacturing solution.

The data logger is intended to be used as a testing tool for many years, and it is expected that electronics and batteries may need to be replaced over that time. While minimizing the data logger's physical size reduces inertial stresses and improves durability, a priority was given to laying out electronic components in a manner that is easy to understand, replace, and modify.

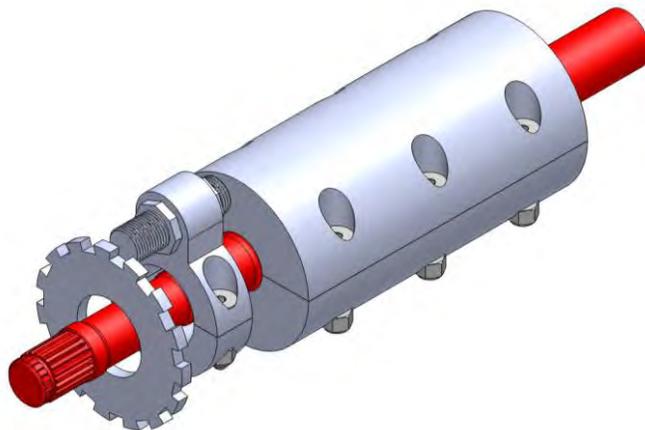


Figure 18: Initial Concept (Rev. 1) with Discrete Speed Sensor

Concept 1: Discrete Speed Sensor

The initial concept for the Torque/Speed data logger is shown in Figure 18, above. The clamshell design was chosen to allow for ease of installation, and ease of access to internal components. While waterproofing was a concern during the design process, it was decided that silicone sealant can be applied to the housing faces during installation.

Also shown is a concept for an external speed reference sensor, similar to the ABS sensors seen on automobiles. If this concept had been pursued further, the sensor would have likely been located between the CV joint housing and the car's gearbox to avoid the angular changes and sliding geometry between the CV housing and axle shaft.

Concept 2: Refined Clamshell with Integrated MEMS Gyroscope

For the first major revision of the data logger concept, the clamshell design was refined and detailed to include features like a swappable access panel, channels for sealant, an area for PCB mounting, and a pair of waterjet aluminum clamps which are used to hold the housing in place on the shaft and locate the two batteries. While all of these elements have undergone significant simplification and revision, the basic functional concepts have remained unchanged through to the final design.

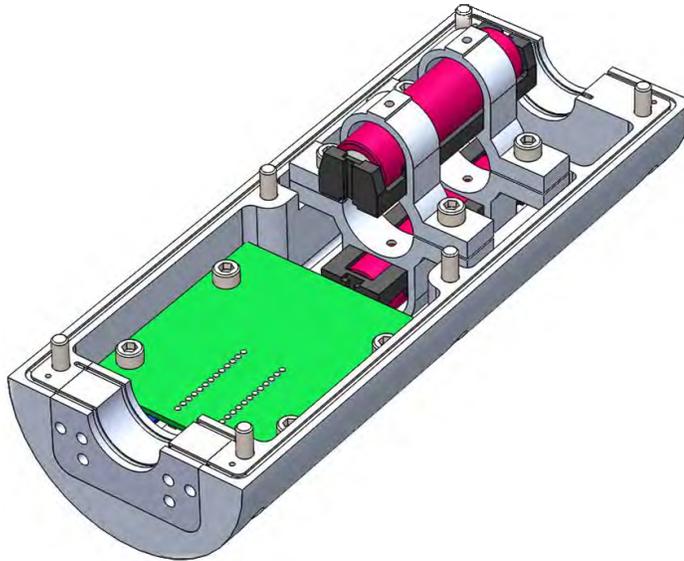


Figure 19: Housing Rev.2: Details Added for Mounting Electronics and Attaching to Axle

Detailed views of the axle clamps and swappable access port can be seen in Figure 20, below. The waterjet-cut aluminum axle clamps are designed to act as the sole interface between the housing and axle. In order to allow the housing to be made of low-strength printed plastic, it is necessary to avoid compressive forces, and so the housing is not tasked with clamping to the axle. The swappable access panel is designed to allow modification and improvements to the data logger interface in the future.

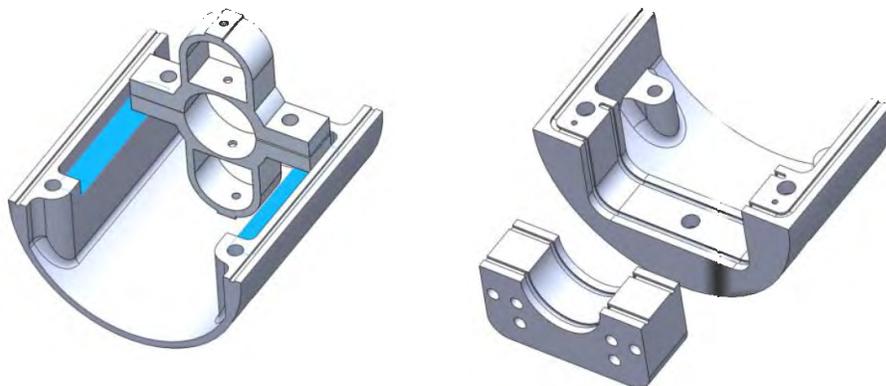


Figure 20: Details of Axle Clamp (Left) and Swappable Access Panel (Right)

Concept 3: Reduction of Complexity and Weight

Further revisions were made to reduce complexity and weight while retaining the same basic functionality seen above. Revision 3, pictured in Figure 21 below, saw the simplification of the access port design, a simplification of the waterproofing seal design, and a reduction of the housing's length and diameter.

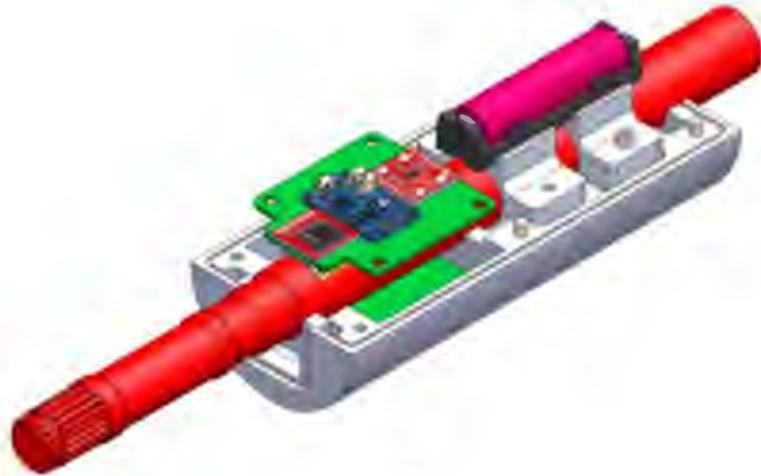


Figure 21: Housing Rev.3: Reduced Size, Simplified Seals and Access Port

Final Design: Rebuilt from the Ground Up

For the final housing design, a new part file was built from scratch, loosely based on the Revision 3 concept geometry. Thought was put into creating geometry that is easily machined on a 3-axis mill, and the total housing length was further reduced from 7.25 inches to 6.6 inches. The sealant groove was removed, as the team agreed that full waterproofing was not necessary on a typical Baja SAE endurance course.

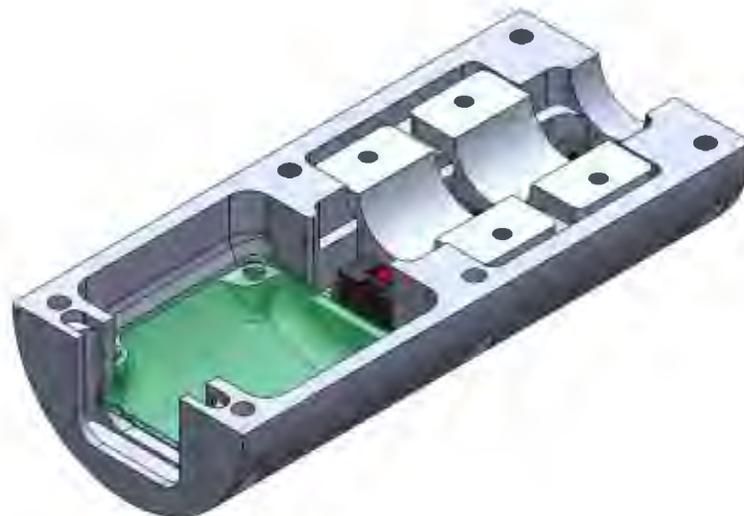


Figure 22: Housing Rev. 4: Final Design with Further Optimization of Size and Complexity

4.2. Rapid Prototyping and Iteration of Housing Designs

During the design revision process, each version of the housing was 3D-printed to evaluate the size and shape of the data logger when installed on the Baja vehicle. This allowed the team to check the fitment of electronic components, to ensure the device's clearance within the vehicle frame, and to get a real sense of the comfort and convenience of its user interface.



Figure 23: Housing Design Iterations with Test PCB and Clamps

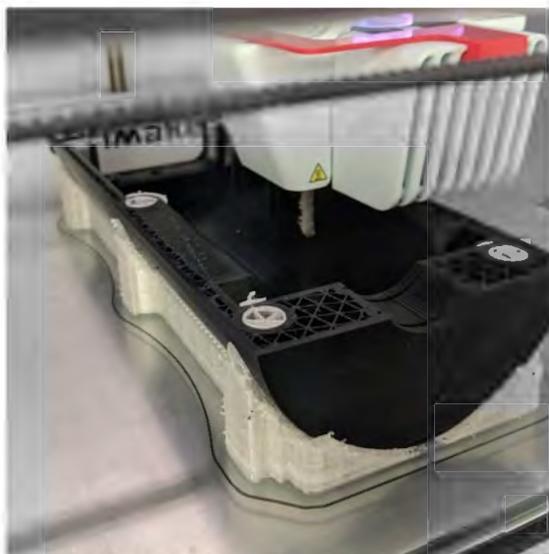


Figure 24: Rapid Prototyping: 3D Printing to Check Revised Housing

4.3. Sensor and Electronics Validation

Before design finalization and manufacturing began, it was necessary to validate that the proposed sensors, microcontrollers, and wireless modules functioned as intended. This was accomplished by creating calibration tests that demonstrated the system's output in response to known physical inputs, such as a given mass, or a known speed.

Gyroscope/Accelerometer Validation

The Gyroscope/Accelerometer validation test module was designed to be placed in a 3-jaw lathe chuck. It was used to validate three functions:

1. MEMS Gyroscope rotation sensing
2. Accelerometer radial acceleration sensing
3. Bluetooth connectivity during rotation

This test was critical in confirming the basic design of the datalogger. If neither the gyroscope nor accelerometer proved suitable, a discrete optical or magnetic encoder would have had to be used. This would have complicated the datalogger's electrical and mechanical designs, and would have harmed the device's durability and reliability.

Fortunately, the test validated the team's assumptions. The gyroscope output the expected value to within 10%, and the Bluetooth connectivity functioned well despite the housing being rotated at approximately 300 RPM.



Figure 25: Gyroscope and Bluetooth Validation Test

Torsion Test Validation

To determine if the torsion measurement method of strain gauges would be acceptable for the final device, a testing rig was built. This test rig allowed for proof of concept, calibration, and debugging of the torsion measurement system while it was isolated from other electronic systems of the device.



Figure 26: Torsion Test Rig

The torsion measurement test rig pictured in Figure 26 was built around a spare Baja vehicle axle, with a strain gauge rosette applied. A typical torsion strain rosette, seen in Figure 27, has two gauges mounted at 45°-degree angles to the axis of the shaft, plus one gauge along the axial direction which we will not be using. Because the torsion rosette gauges are aligned out of the box, it is simple to set up a half-bridge Wheatstone bridge configuration, producing a sensor that is twice as sensitive to strain as a quarter-bridge.

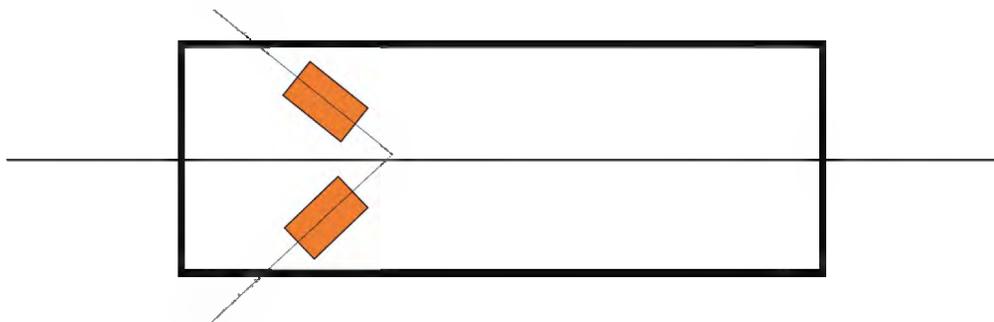


Figure 27: Diagram of a Typical Torsion Rosette with Half-Bridge Gauges

A 13-inch length of ¼ inch flat bar was used to create the loading bar. A hole was drilled at one end using a drill press to allow loads to be hung. A female splined CV hub was welded to the loading bar, allowing it to be placed on the end of the axle shaft when needed.

To remove any irregularities on the shaft surface that could cause incorrect strain readings, the axle shaft was smoothed. Depending on the axle, after paint is removed, grooves may need to be taken down with grinders, sandpaper, or a lathe. A standardized procedure for strain gauge installation has been provided by the manufacturer, and can be found in Appendix F.

Validation was achieved by applying a known force to the loading bar connected to the end of the drive shaft, causing torsion in the shaft. An example of a typical serial output plot can be seen below, in Figure 28. Using the relationship between torsional strain and the resistance of the strain gauges, a numeric value for the applied torque can be measured and displayed via the microcontroller’s serial output in the Arduino IDE.

An example script using a 16-bit ADS1115 sensor can be found in Appendix A.2, or on BCIT Racing’s GitHub page at:

https://github.com/BCITRacing/STorq/tree/master/ESP32_Sensor_Test_Scripts

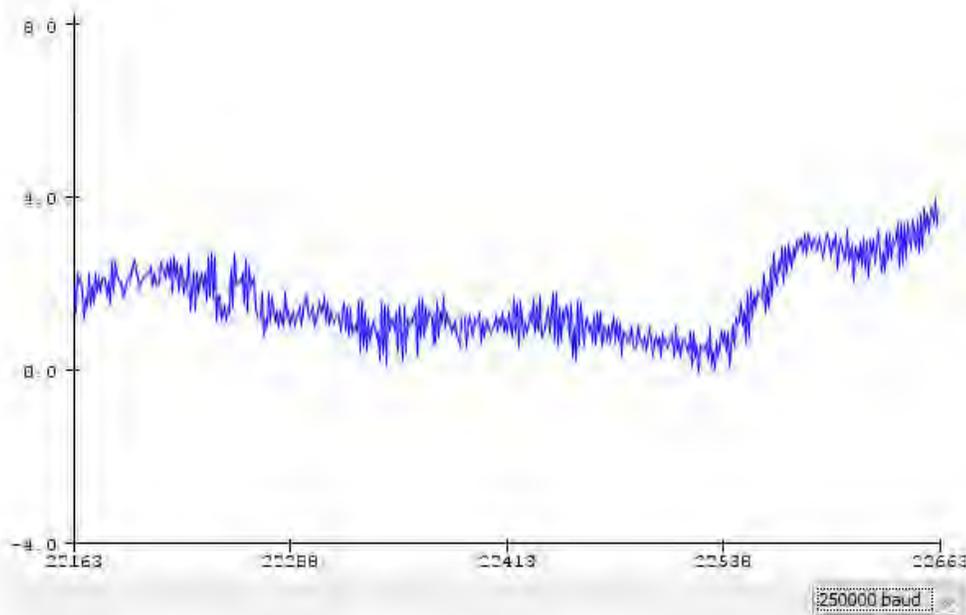


Figure 28: Arduino IDE Serial Plot of Torsion Data

The torsion test rig circuit is pictured below, in Figure 29. To process the signals from the Wheatstone bridge, the circuit uses an op-amp, an instrumentation amplifier, gain resistors, and a 16-bit ADS1115 ADC which feeds either an Arduino or an ESP32. All electrical components were mounted to the breadboard and connected with wire. It should be noted that voltage divider resistors may be necessary if using an Arduino as the microcontroller.

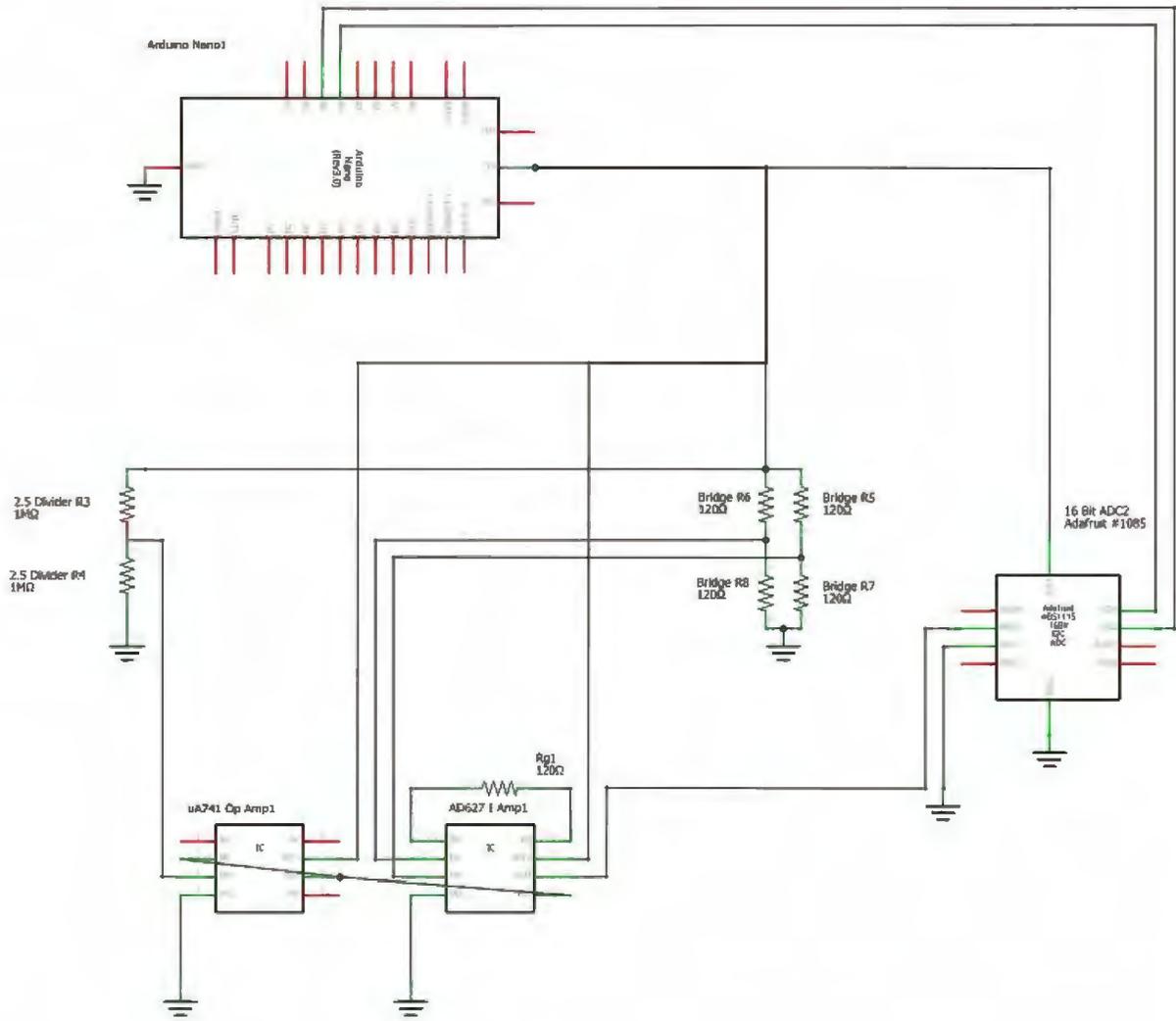


Figure 29 Torsion Tester Wiring Schematic

Mechanical Calibration and Test Procedure

Calibration of the torsion measuring system can be performed by loading the axle with known set of weights. When multiple weights are measured and recorded, the gain (slope) of the line can be found for the system. If there is a gain error, it can be removed by programming a correction factor into the code for the microcontroller. It may also be necessary to correct the zero offset error of the system.

The figure below shows an assumed linear output of the strain gauge along with the measured output of the device with an incorrect gain.

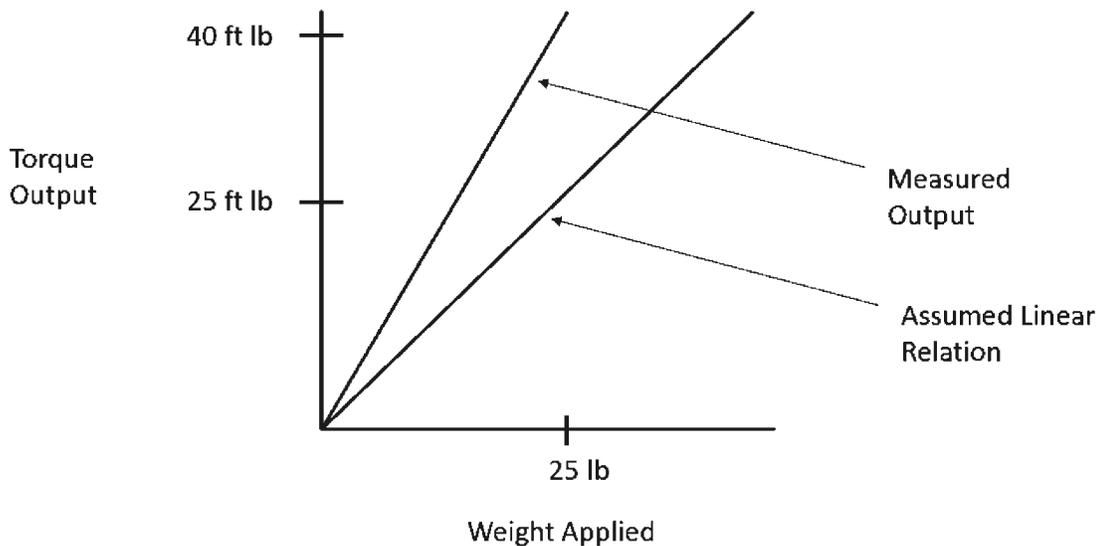


Figure 30: Gain Error

Force/Voltage Relation for Torsion

The force voltage relationship derived below was used to convert voltage signals from the Wheatstone bridge into a value corresponding to a measured torque.

τ = Shear stress

σ = Stress

T = Torque

c = Distance to outer edge of shaft from centre

$J_{solid\ shaft}$ = Polar moment of inertia

D = Diameter of the shaft

E = Young's Modulus

ν = Poisson's ratio

SG = strain gauge constant

ϵ = strain

R = resistance

V_i = bridge input voltage

V_o = bridge output voltage

The final derived equation was coded into the microcontroller to convert voltage into a usable torque value.

Strain-Force Relation

$$\tau = \frac{Tc}{J}$$

For a solid shaft,

$$J_{solid\ shaft} = \frac{\pi D^4}{32}$$

$$c = \frac{D}{2}$$

Therefore,

$$T = \frac{\tau J}{c} = \frac{\tau \pi D^3}{16} \quad (1)$$

$$\sigma = \tau = \frac{\epsilon E}{1 + \nu} \quad (2)$$

Combining 1 and 2,

$$T = \frac{\epsilon E \pi D^3}{(1 + \nu) 16}$$

Strain relation:

$$\frac{\Delta R}{R} = SG * \epsilon$$

Therefore,

$$\frac{\Delta R}{R} = SG * \frac{T(1 + \nu)16}{E\pi D^3}$$

Half Bridge Voltage Input Output

$$V_o = \frac{1}{2} \frac{\Delta R}{R} V_i$$

$$V_o = \frac{1}{2} SG * \frac{T(1 + \nu)16}{E\pi D^3} V_i$$

Torque-Voltage Relation

$$T = \frac{V_o E \pi D^3}{8 SG (1 + \nu) V_i} = \frac{1 V_o * (29000000) \pi (1)^3}{8 (1.5) * (1 + 0.27) * (5)}$$

The following constant values are used:

$$D = 1 \text{ inch}$$

$$V_i = 5V$$

$$J_{\text{solid shaft}} = \frac{\pi D^4}{32} = \frac{\pi}{32}$$

$$E_{\text{steel}} \approx 29 * 10^6 \text{ psi}$$

$$\nu_{\text{steel}} = 0.27 - 0.31$$

$$SG = 1.3 \pm 0.2$$

Positive and Negative Torsion Measurement

The main challenge for the torsion device was to be able to measure torsion in both directions about the axis of the drive shaft. It is important for the device to read both directions, so driveline loads can be measured for different vehicle obstacles such as jumps and landings. This was a challenge because the final device on the vehicle would not be connected to a bench power supply, which could easily supply the device with both positive and negative voltages. Instead the microcontroller would only have +3.3V and ground.

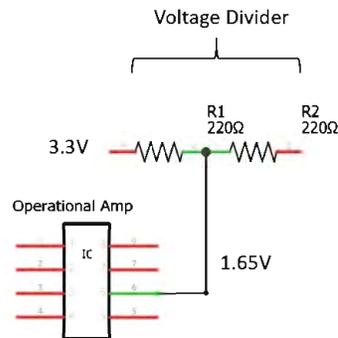


Figure 31: Voltage Divider Circuit

To allow the torsion rig to measure both direction of torsion, a reference voltage of +1.65V was used. This reference voltage was fed into the reference pin of an instrumentation amp to allow a signal of 0V to 1.65V to measure on direction and >1.65V to 3.3V to measure the other direction. The circuit diagram below shows the configuration of the electronics used to make the +1.65V.

An additional op amp was needed in order to have a +1.65V available for the instrumentation amp. The op amp was used in a unity gain configuration so that its output was consistent. The source for this op amp was a voltage dividing circuit consisting of two equal resistors. This provided a steady 1.65V reference without the power losses associated with a linear regulator.

4.4. Electrical System Design

Wiring Diagrams and Schematics

Electrical schematics were created with the software Fritzing. This program offers a quick interface for creating circuit diagrams, and contains libraries of many of the components that were used for the project. Using Fritzing simplified the process of making changes to the electrical system, and allowed for quicker implementation of changes. Fritzing also allows for the electrical system to be translated into a layout to produce a printed circuit board, with automatically-generated PCB traces.

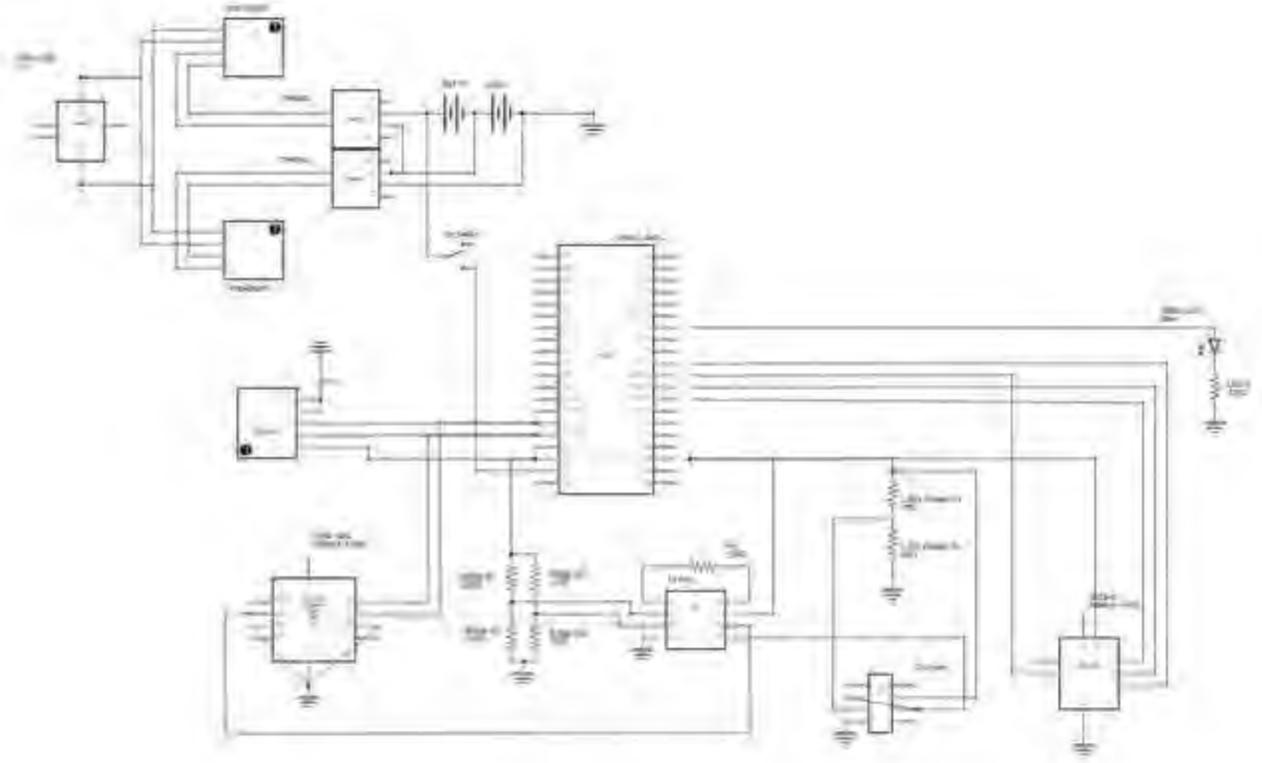


Figure 32: Full System Wiring Schematic

Battery System

The battery charging system was custom-built, due to the data logger using a non-standard 7.4V system voltage. Specialized charging boards do exist for this voltage, however, due to cost and shipping times, the project team went ahead with designing a semi-custom solution.

The charging system uses a 5V USB input fed into two DC-DC converters which isolate the batteries. In this configuration, the batteries are charged independently in parallel. The batteries are configured in series to get a system voltage of +7.4V which is in the supply voltage range of the microcontroller.

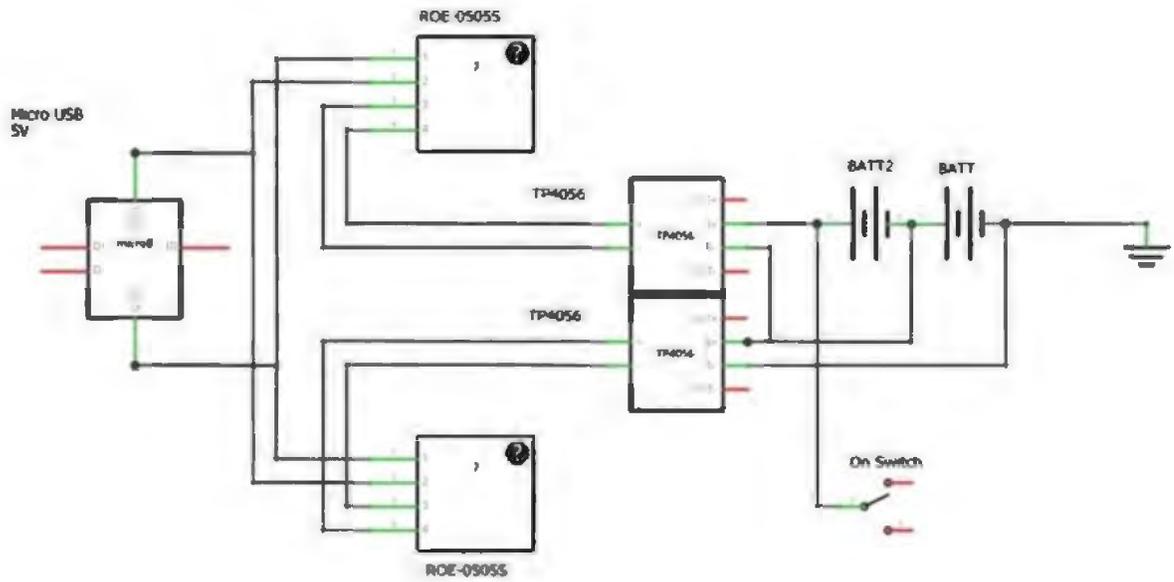


Figure 33 Battery System Wiring Diagram

PCB Design and Manufacture

Two printed circuit boards were designed and manufactured to simplify wiring and locate electronic components. Also, circuit boards allowed for secure connections which were sometimes lacking in the breadboards used in the projects. The two boards were designed in the Fritzing software and manufactured in house in the BCIT electronics laboratory. The boards were designed to be as compact as possible to fit inside the enclosed space of the device housing.

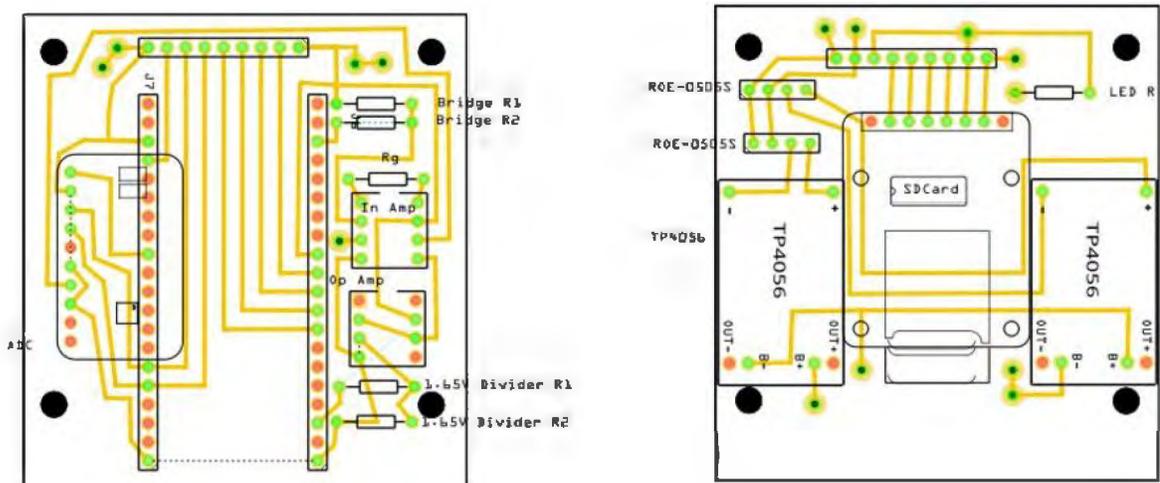


Figure 34 Top and Bottom PCB Layout

PCB Manufacturing Procedure

The printed circuit board were manufactured in the BCIT electronics lab. The process required printing the circuit board designs on special transfer paper. The toner from the print was transferred to a copper covered board with the help of a heated T-shirt press. Once the toner was transferred to the copper a foot activated shear was used to trim the dimensions of the board.



Figure 35: PCB in Etching Tank

The next step in manufacturing a PCB was to put the board in an etching tank to remove unwanted copper from the board. When the desired copper was fully removed, the board was wiped with acetone to remove the layer of toner. The board was completed by drilling the appropriate holes for the electronic components with a drill press.



Figure 36: Toner Removal in Acetone Bath

5. Discussion of Results

5.1. Torsion Test Results

To validate the output of the torsion strain gauge, a spare axle was converted into a test rig. A strain rosette was applied to the shaft of the axle, and a block was welded to its base to allow clamping in a vise. Hanging a known mass at a set distance from the axis of rotation permits calibration of the amplifier, and an estimation of the full-scale voltage range that the analog-to-digital converter should be ranged within.

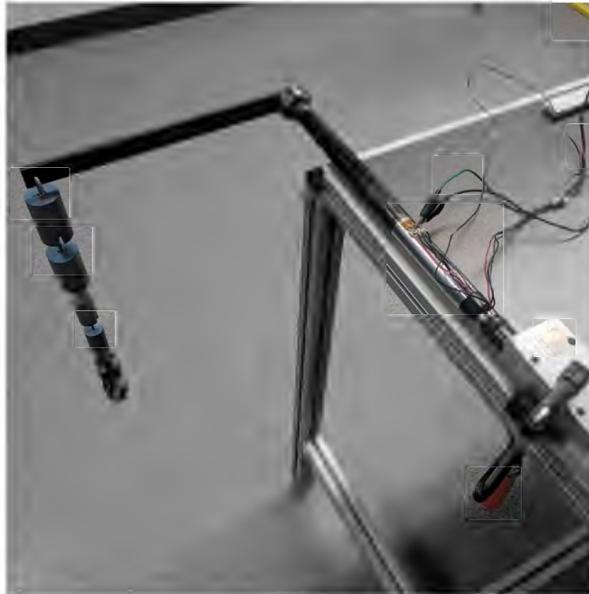


Figure 37: Torsion Calibration Test in Progress

The initial results from the torsion test using a 25 lb weight resulted in values of that were higher than anticipated. The output from the graphing code showed that the torque output was exactly 1.6 times greater than should have been. Review of the microprocessor code and equation derivations showed that the strain gauge factor SG was missing from the voltage strain relation equation. This value was found on the packaging of the acquired strain gauges. Spec given on the package was 1.3 ± 0.3 . After adding this number to the code, the output of the test rig was much closer to anticipated values.



Figure 38: Strain Gauge Data Label

5.2. Eliminating Zero Offset

It was observed during testing that the serial output of the torsion test rig has a natural zero offset error. That is, when no load is applied, a non-zero torque reading is shown. To eliminate this offset, a y-intercept bias adjustment can be added to the code. It was also observed that sometimes when wires on the breadboard were disturbed, torsion reading would be left with a permanent bias. This should be corrected with PCB-mounting.

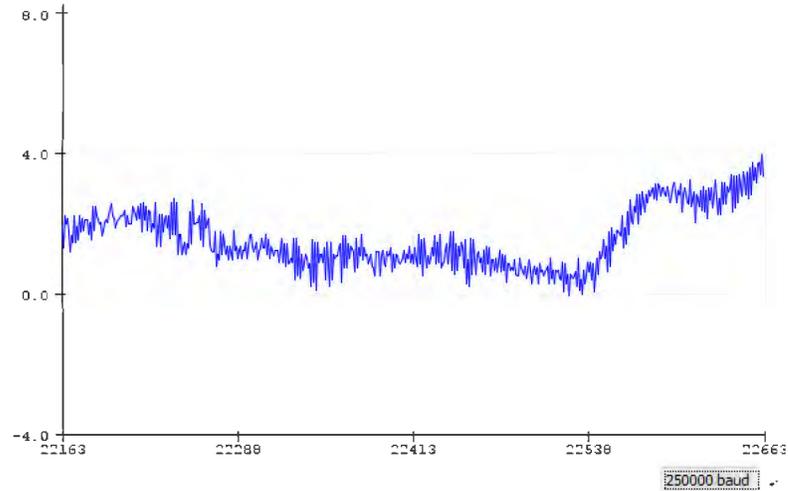


Figure 39: Non-Zeroed Torsion Reading

Figure 38 shows the output of the system with a 25 lb. weight applied to the loading bar. The unloaded output showed a zero offset approximately 6 lbs, and the reading with the load is approximately 32 lbs. If the system was properly zeroed, the measured load would have been 26lb which is 1 lb of the expected value. When the axle is assembled, it should be calibrated to make an attempt at eliminating zero-offset, however the magnitude of this offset will likely be minor compared to the torques experienced during vehicle testing.

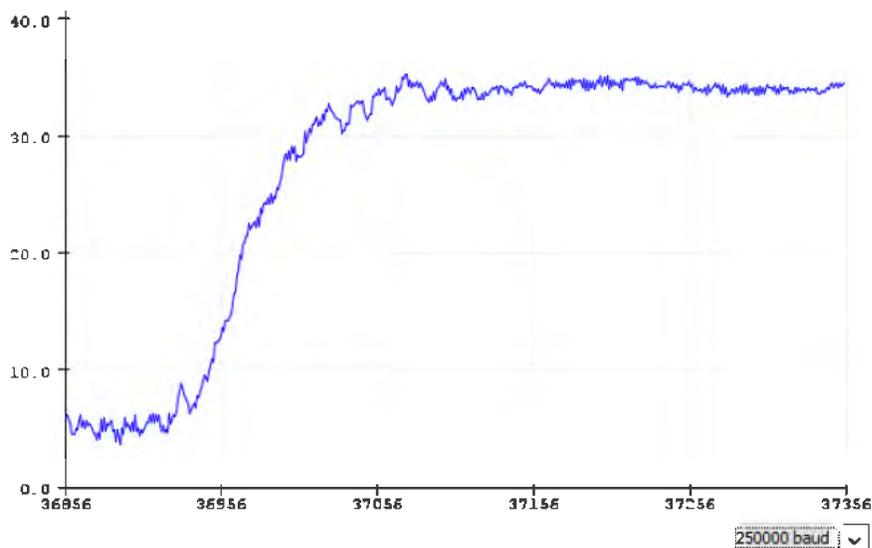


Figure 40: Torsion Reading 25lb Clockwise

5.3. Battery System Results

The operation test of the battery charging system showed that the batteries charged independently as predicted. Also, the system voltage from the batteries in series measured in the required range for the micro controller to use. No overheating of any battery system components was found. Safety was a large concern for designing the system. However, no problem could be reported.

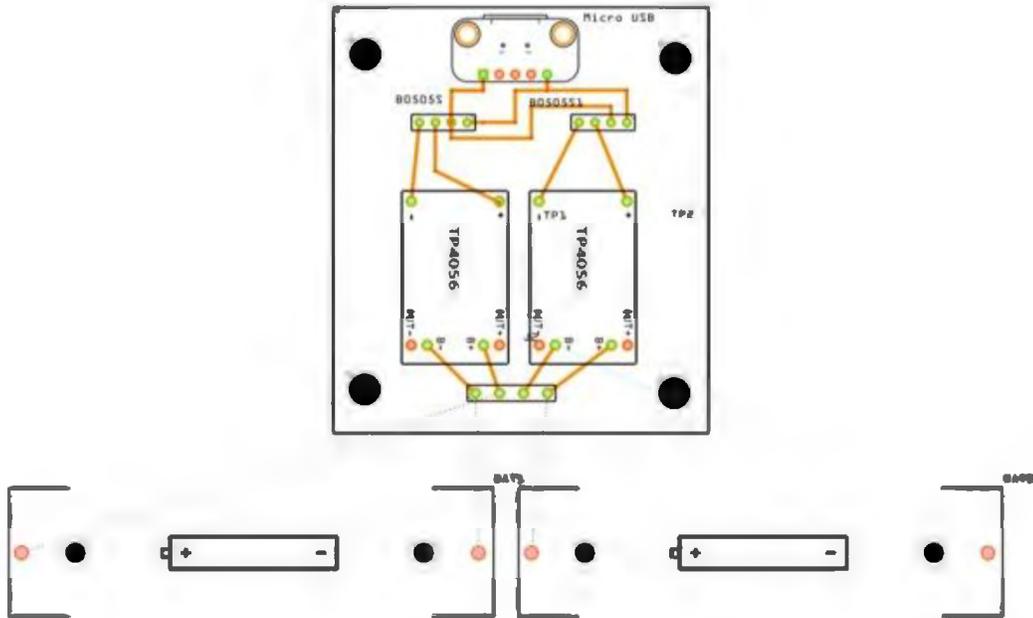


Figure 41 Battery Charging PCB Layout

5.4. PCB Results

Ultimately, the team was successful in unifying all the data logger components into a system of two PCBs. Testing shows that torque and angular velocity sensing is working as intended.

It should be noted that some problems were encountered during manufacturing. These included problems with sizing the PCBs for the housing, and electrical path errors that had to be fixed by scratching out leads on the final PCBs.

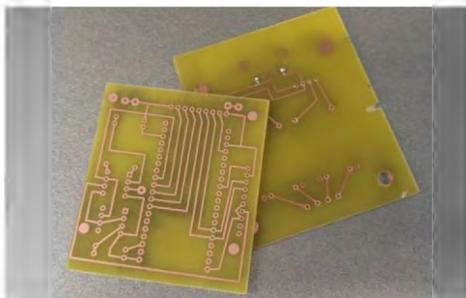


Figure 42: Custom PCBs

5.5. Android Mobile Application

At the time of this report, the data logger's corresponding mobile application is successfully graphing and storing a single stream of data transmitted by the ESP32 microcontroller over Bluetooth.

Test code has been written for the ESP32 which outputs a three-variable string, including a microcontroller-side timestamp, however the Android app must be updated to parse the new data structure. Previously, the app received a single value as a string over Bluetooth, and parsed it into a double. Now, it must receive three comma delimited values transmitted in a single string, and parse them into a convenient data structure for graphing and storing the information.

The multi-variable transmission test code for an ESP32 can be found in Appendix A.3., however this code will continue to be updated and the latest version can be found on BCIT Racing's GitHub at the following URL:

https://github.com/BCITRacing/STorq/tree/master/ESP32_Sensor_Test_Scripts/Multiple_Data_String_Bluetooth_Test

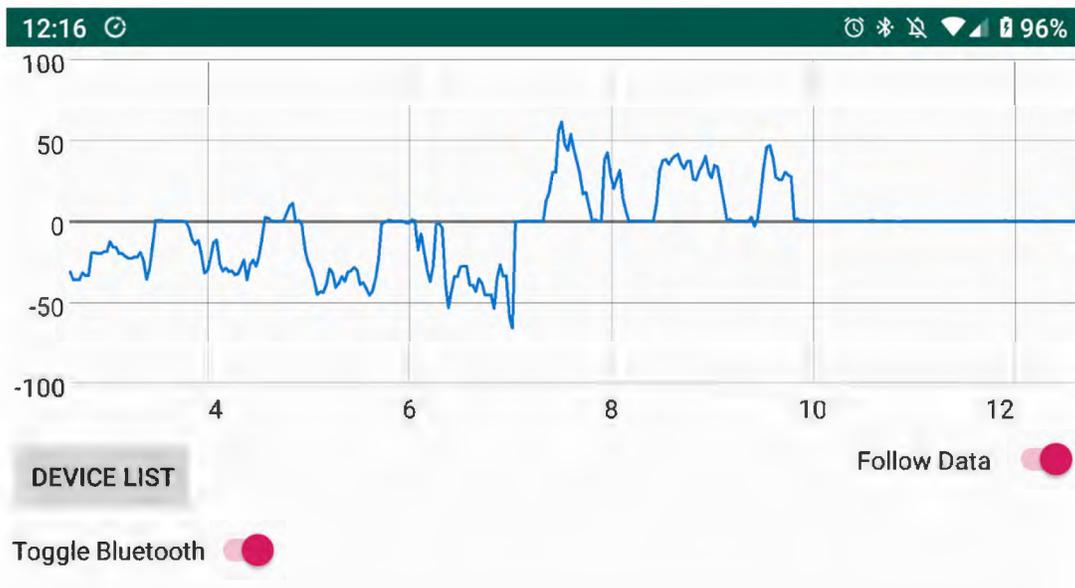


Figure 43: Gyroscope Data Displayed on Grapher App

6. Conclusion

6.1. Summary and Assessment of Success

Overall, this project was a success, despite some remaining functionality needing to be added to the Android mobile app. For this reason, the application source code has been uploaded to BCIT Racing's GitHub page at the following URL:

<https://github.com/BCITRacing/STorq>

This GitHub repository will contain the most up-to-date version of the graphing App, and will also contain scripts for testing the ESP32 with sensor packages, as well as a final ESP32 script, ESP32_FINAL.ino.

Additionally, this will be where user guide documentation will be supplied to the team.

6.2. Lessons Learned

Through the process of completing the Torque/Speed data logger project, some important lessons have been learned. These include:

1. Redundancy in manufacturing methods is incredibly important. The housing for the data logger was initially designed to be CNC machined from Aluminum or Acetyl, but the design was strengthened and simplified along the way to allow it to be printed with additive methods as well. This turned out to be a good choice, as time and machine availability became scarce in the spring.
2. Similarly, redundancy of functional components became an important consideration. The ICM-20601 MEMS gyroscope chosen for its 4000 dps capabilities was challenging to integrate, and was very nearly abandoned. A backup plan was in place to use an LSM9DS1 sensor if necessary.
3. Time management is always key. For a short project, it is important to recognize when a concept has been sufficiently proven or validated, and to recognize when it is time to push onward. For example, while sensor validation is important, fine calibration is critical upon final assembly, not mid-project.

6.3. Future Work

The success of the Torque/Speed data logger may enable BCIT Racing members to press forward with interesting design projects in future race seasons. Some examples may include:

1. Design of a slip clutch torque limiter, either integrated as a two-piece gear, or as an external part. This project might be similar in function to the *Propeller Torque Limiter Design and Manufacture* capstone completed this year by Yioldassis, Thomson, Sorace and Musil. A torque limiter with spring-tensioned ball bearings

would allow the team to closely calibrate maximum allowable back-driven torques, allowing the use of negligible safety factors in the design of drivetrain components like gears and axles.

2. Improvements to the Android application interface to expand the control of testing parameters like averaging, sample rate, and calibration. Additionally, it would be beneficial to create an Arduino sketch that allows users to download stored data from the on-board SD card, and to wipe it clean.
3. As new axles are designed, strain gauges may be attached to them. If the axles remain the same diameter, the housing should not require modification. If a larger diameter is used, the housing will need significant revision, while a slimmer axle can be easily adapted to.
4. The Torque/Speed data logger concept may benefit from an iterative continuation of the design process after the current model has been tested and evaluated. Improvements could be made to making the data logger lighter, smaller, and simpler to manufacture and use.
5. Investigate the feasibility of using the cheap and well-documented LSM9DS1 gyroscope. While this sensor package's gyroscope only records up to 2000 dps, it may be feasible to program a switchover function to calculate rotation speed with the unit's accelerometers after a certain speed is reached. This would make programming the data logger simpler due to the existence of Adafruit Arduino libraries, and it would be far cheaper to produce.

6.4. Design Hurdles

Faulty Breadboard

One of the major difficulties with the project was proofing the torsion measurement concept using strain gauges. The difficult with this part of the project was with getting usable reading out of the Wheatstone bridge when mounting all the electronics onto a breadboard. It was found that a faulty breadboard was interrupting the flow of electricity in the circuit. Once the breadboard was changed, usable signal readings were read by the micro controller.

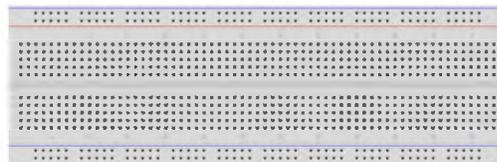


Figure 44: Faulty Breadboard

PCB Manufacturing

Manufacturing the PCBs caused some problems and delays in the project for two reasons. The first was that both members of the team were unfamiliar with the process, and the second was that any mistakes in the design of the board required the board to be remade. The available times that the electronics lab made it difficult to remake boards when mistakes were found. After the team became familiar with the process and the all the bugs were found on the board, the team was able to complete this stage of the project.

Microcontroller Selection

The initial selection for the microcontroller to be used with the project was an Arduino Nano. It was later decided to switch to an ESP32 microcontroller after the electrical system was becoming increasingly complex as due to the voltage dividers required by multiple sensors and accessories designed for 3.3V circuits. The ESP32 naturally runs its digital I/O at 3.3V. The ESP32 also integrates Bluetooth functionality, which otherwise had to be added with a separate HC-05 module. It should be noted that the ESP32 can be programmed with the Arduino IDE, when the proper libraries are installed, making it as simple to use as a typical Arduino.

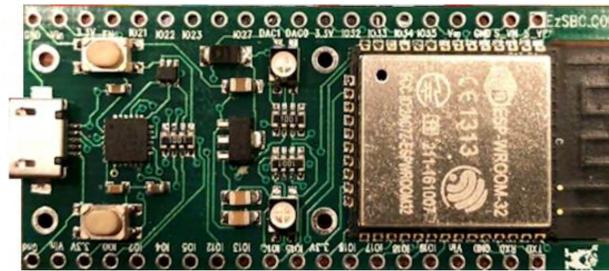


Figure 45: ESP 32 Microcontroller

Appendix A: Source Code

NOTE: This code remains live, and may be updated by BCIT Racing team members. Please check BCIT Racing's GitHub page for the most up-to-date version:

https://github.com/BCITRacing/STorq/tree/master/ESP32_Sensor_Test_Scripts

Appendix A.1. Gyroscope Test Script

```
//This code is used to validate the transmission of a single data point in
string form via Bluetooth serial.
//Use an app like Serial Bluetooth Terminal to observe data transmission:
https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_te
rминаl
//By Russ Case, 2019
```

```
#include "BluetoothSerial.h"
#include <Wire.h>

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

BluetoothSerial SerialBT;

void setup() {
  Serial.begin(19200);
  SerialBT.begin("BCIT Racing Gyro Test"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetooth!");

  Wire.begin(); // join i2c bus as master
  Wire.setClock(400000);
  Serial.println("Open i2c bus");

  Wire.beginTransmission(0x68); // transmit to device #0x68
  Wire.write(byte(0x1B)); // sets register pointer to the gyro command
register (27)
  Wire.write(byte(0x18)); // sets gyro to 4000 dps
  int error1 = Wire.endTransmission(); // stop transmitting

  Wire.beginTransmission(0x68); // transmit to device #0x68
  Wire.write(byte(0x6B)); // sets register pointer to the power
management 1 register (107)
  Wire.write(byte(0x9)); // turn off sleep mode, disable temp
sensor, set clk type (see pg 43 of datasheet)
  int error2 = Wire.endTransmission(); // stop transmitting

  Wire.beginTransmission(0x68); // transmit to device #0x68
  Wire.write(byte(0x6C)); // sets register pointer to the power
management 2 register (108)
```

```

    Wire.write(byte(0x38));          // disable accelerometers, enable gyros
    (see pg. 44 of datasheet)
    int error3 = Wire.endTransmission(); // stop transmitting

    Serial.print(error1);
    Serial.print(error2);
    Serial.print(error3);
    Serial.println("\nSetup Complete");
    delay(1000);
}

short reading = 0;
double rpm = 0.0;

void loop() {

    Wire.beginTransaction(0x68);      // get slave's attention
    Wire.write(byte(0x45));          // sets register pointer to echo
    Xout_h
    Wire.requestFrom(byte(0x68), 2, false); // request 1 byte from
    slave device #0x68
    int error4 = Wire.endTransmission(); // stop transmitting

    if (2 <= Wire.available()) { // if two bytes were received
        reading = Wire.read(); // receive high byte (overwrites previous reading)
        reading = reading << 8; // shift high byte to be high 8 bits
        reading |= Wire.read(); // receive low byte as lower 8 bits
        //rpm = (double(reading))*0.02034505;
        rpm = (double(reading))*4000/6/32768;
        SerialBT.println(rpm); // print the reading
    }
}
}

```

Appendix A.2. Torsion Test Script

```
// This code is used to validate the serial output of a strain gauge reading
sent through an ADS1115 ADC.
// By Jeremiah Moreno - 2019

#include <Wire.h>
#include <Adafruit_ADS1015.h>

Adafruit_ADS1115 ads; /* Use this for the 16-bit version */
//Adafruit_ADS1015 ads; /* Use thi for the 12-bit version */

float result_sum = 0.0;

void setup(void)
{
  Serial.begin(250000);
  //ads1115.begin(); // Initialize ads1115
  ads.begin(); // Initialize ads1115

  //
  ADS1115                                     ADS1015
  //
  -----
  //ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V 1 bit = 3mV
0.1875mV (default)
  ads.setGain(GAIN_ONE); // 1x gain +/- 4.096V 1 bit = 2mV
0.125mV
  // ads.setGain(GAIN_TWO); // 2x gain +/- 2.048V 1 bit = 1mV
0.0625mV
  // ads.setGain(GAIN_FOUR); // 4x gain +/- 1.024V 1 bit = 0.5mV
0.03125mV
  // ads.setGain(GAIN_EIGHT); // 8x gain +/- 0.512V 1 bit = 0.25mV
0.015625mV
  // ads.setGain(GAIN_SIXTEEN); // 16x gain +/- 0.256V 1 bit = 0.125mV
0.0078125mV
}

void loop(void)
{
  //Define those variables yo
  int16_t results, adc0;
  double volt;
  double torque;
  double Vi = 3.3;
  double D = 1.0;
  double pie = 3.141592654;
  double E = 27.5E6; //Somewhere between 27E6-31E6
  double nu = 0.31;
  double zerovalue = 2.0544373989;
  double ampGain = 1000;
  double SG = 1.5; //0.13+-0.2
  double ADCconversion = 0.000125;// This value is found in the above table
depending on the chosen gain
```

```

//Get the Voltage difference from the IN-AMP
results = ads.readADC_Differential_0_1();
volt = (double)results * ADCconversion;
volt = volt - zerovalue; //The zero no load voltage of shaft
volt = volt / ampGain;
torque = volt * pie * pow(D, 3) / SG / 8.0 * (E / (1.0 + nu)) / Vi / 12.0;
//the divide 12 is a foot
Serial.println(torque, 2);

//Code for zeroing and debugging
/*
//Get the Voltage difference from the IN-AMP
results = ads.readADC_Differential_0_1();
volt = (double)results * 0.0001875;
Serial.print("voltage:");
Serial.print(volt, 10);
Serial.print("\n");
//zero the reading
volt = volt - zerovalue; //The zero no load voltage of shaft
//remove the IN AMP Gain
volt = volt/ampGain;

Serial.print("corrected voltage:");
Serial.print(volt, 10);
Serial.print("\n");

//Calculate Torsion

torque = volt * pie * pow(D, 3) / 8.0 * (E / (1.0 + nu)) / Vi/12.0;//the
divide 12 is a foot conversion
Serial.print("torque:");
Serial.print(torque, 2);
Serial.print("\n");
delay(1000);
*/
}

```

Appendix A.3. Multiple Data Point String Bluetooth Test

```
//This code sends two axes readings from the ICM-20601 gyro along with uptime
in ms, through bluetooth serial as a string.
//Use an app like Serial Bluetooth Terminal to observe data transmission:
https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_te
rминаl
//By Russ Case - 2019

#include "BluetoothSerial.h"
#include <Wire.h>

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

BluetoothSerial SerialBT;

void setup() {
  Serial.begin(19200);
  SerialBT.begin("Adobo Analytics ESP32"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetooth!");

  Wire.begin(); // join i2c bus as master
  Wire.setClock(400000);
  Serial.println("Open i2c bus");

  Wire.beginTransmission(0x68); // transmit to device #0x68
  Wire.write(byte(0x1B)); // sets register pointer to the gyro command
register (27)
  Wire.write(byte(0x18)); // sets gyro to 4000 dps
  int error1 = Wire.endTransmission(); // stop transmitting

  Wire.beginTransmission(0x68); // transmit to device #0x68
  Wire.write(byte(0x6B)); // sets register pointer to the power
management 1 register (107)
  Wire.write(byte(0x9)); // turn off sleep mode, disable temp
sensor, set clk type (see pg 43 of datasheet)
  int error2 = Wire.endTransmission(); // stop transmitting

  Wire.beginTransmission(0x68); // transmit to device #0x68
  Wire.write(byte(0x6C)); // sets register pointer to the power
management 2 register (108)
  Wire.write(byte(0x38)); // disable accelerometers, enable gyros
(see pg. 44 of datasheet)
  int error3 = Wire.endTransmission(); // stop transmitting

  Serial.print(error1);
  Serial.print(error2);
  Serial.print(error3);
  Serial.println("\nSetup Complete");
}
```

```

short reading1 = 0;
short reading2 = 0;
double rpm = 0.0;

int time_0;
int time_now;
int i = 0;

void loop() {

  String msg = "";
  String r1 = "";
  String r2 = "";
  String t = "";

  // Record time in ms
  if (i <= 0) {
    time_0 = millis();
    i++;
  }

  //SerialBT.println('A'); // Send signal indicating that
  ESP32 is ready to transmit data

  time_now = (millis() - time_0); // record time of readings
  t = String(time_now, DEC);

  delay(250); // Delay for human readability. Take
  this out later

  //if (SerialBT.read() == 'B') { // Detect phone signal indicating
  it is ready to receive data

    // Call X-Gyro
    // (NOTE: replace the X-gyro call with an ADS1115 request, we want to send
    time, torsion and RPM. This is just to check functionality)
    Wire.beginTransaction(0x68); // get slave's attention
    Wire.write(byte(0x43)); // sets register pointer to echo
  X axis high byte
    Wire.requestFrom(byte(0x68), 2, false); // request 1 byte from slave
  device #0x68
    Wire.endTransmission(); // stop transmitting

    if (2 <= Wire.available()) { // if two bytes were received
      reading1 = Wire.read(); // receive high byte (overwrites
  previous reading)
      reading1 = reading1 << 8; // shift high byte to be high 8
  bits
      reading1 |= Wire.read(); // receive low byte as lower 8
  bits
      r1 = String(reading1);
    }
  }
}

```

```

    // Call Y-Gyro
    Wire.beginTransaction(0x68);           // get slave's attention
    Wire.write(byte(0x45));               // sets register pointer to echo
Y axis high byte
    Wire.requestFrom(byte(0x68), 2, false); // request 1 byte from slave
device #0x68
    Wire.endTransmission();               // stop transmitting

    if (2 <= Wire.available()) {           // if two bytes were received
        reading2 = Wire.read();            // receive high byte (overwrites
previous reading)
        reading2 = reading2 << 8;         // shift high byte to be high 8
bits
        reading2 |= Wire.read();           // receive low byte as lower 8
bits
        r2 = String(reading2);
    //}

    msg = t + "," + r1 + "," + r2;         // Assemble time, reading1, and
reading 2 into a string to transmit

    SerialBT.println(msg); // print the reading
}
// use this conversion factor if you want to convert to RPM before
transmission.
// rpm = (double(reading))*4000/6/32768;
}

```

Appendix B. Selections from ADS 1115 Datasheet

Appendix B.1. Overview

Folder Now Documents Software Community



ADS1113, ADS1114, ADS1115
SBAS444D –MAY 2009–REVISED JANUARY 2018

ADS111x Ultra-Small, Low-Power, I²C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator

1 Features

- Ultra-Small X2QFN Package:
2 mm × 1.5 mm × 0.4 mm
- Wide Supply Range: 2.0 V to 5.5 V
- Low Current Consumption: 150 μ A
(Continuous-Conversion Mode)
- Programmable Data Rate:
8 SPS to 860 SPS
- Single-Cycle Settling
- Internal Low-Drift Voltage Reference
- Internal Oscillator
- I²C Interface: Four Pin-Selectable Addresses
- Four Single-Ended or Two Differential Inputs
(ADS1115)
- Programmable Comparator (ADS1114 and
ADS1115)
- Operating Temperature Range:
–40°C to +125°C

2 Applications

- Portable Instrumentation
- Battery Voltage and Current Monitoring
- Temperature Measurement Systems
- Consumer Electronics
- Factory Automation and Process Control

3 Description

The ADS1113, ADS1114, and ADS1115 devices (ADS111x) are precision, low-power, 16-bit, I²C-compatible, analog-to-digital converters (ADCs) offered in an ultra-small, leadless, X2QFN-10 package, and a VSSOP-10 package. The ADS111x devices incorporate a low-drift voltage reference and an oscillator. The ADS1114 and ADS1115 also incorporate a programmable gain amplifier (PGA) and a digital comparator. These features, along with a wide operating supply range, make the ADS111x well suited for power- and space-constrained, sensor measurement applications.

The ADS111x perform conversions at data rates up to 860 samples per second (SPS). The PGA offers input ranges from ± 256 mV to ± 6.144 V, allowing precise large- and small-signal measurements. The ADS1115 features an input multiplexer (MUX) that allows two differential or four single-ended input measurements. Use the digital comparator in the ADS1114 and ADS1115 for under- and overvoltage detection.

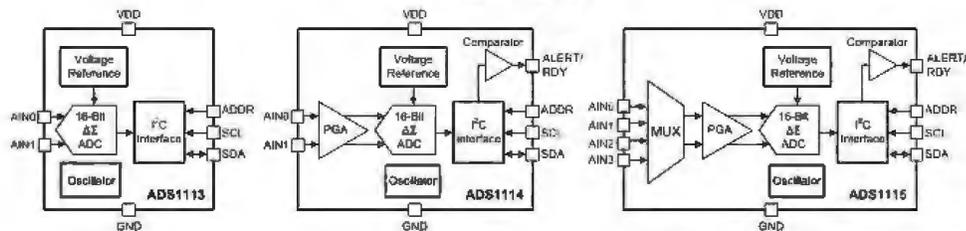
The ADS111x operate in either continuous-conversion mode or single-shot mode. The devices are automatically powered down after one conversion in single-shot mode; therefore, power consumption is significantly reduced during idle periods.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (mm)
ADS111x	X2QFN (10)	1.50 mm × 2.00 mm
	VSSOP (10)	3.00 mm × 3.00 mm

(1) For all available packages, see the package option addendum at the end of the data sheet.

Simplified Block Diagrams



Copyright © 2016, Texas Instruments Incorporated

IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Appendix B.2. Absolute Maximum Ratings



ADS1113, ADS1114, ADS1115

SBAS444D—MAY 2008—REVISED JANUARY 2018

www.ti.com

7 Specifications

7.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)⁽¹⁾

		MIN	MAX	UNIT
Power-supply voltage	VDD to GND	-0.3	7	V
Analog input voltage	AIN0, AIN1, AIN2, AIN3	GND - 0.3	VDD + 0.3	V
Digital input voltage	SDA, SCL, ADDR, ALERT/RDY	GND - 0.3	5.5	V
Input current, continuous	Any pin except power supply pins	-10	10	mA
Temperature	Operating ambient, T _A	-40	125	°C
	Junction, T _J	-40	150	
	Storage, T _{stg}	-60	150	

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

7.2 ESD Ratings

		VALUE	UNIT
V _{ESD}	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾	±2000
		Charged device model (CDM), per JEDEC specification JESD22-C101 ⁽²⁾	±500

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

7.3 Recommended Operating Conditions

		MIN	NOM	MAX	UNIT
POWER SUPPLY					
	Power supply (VDD to GND)	2		5.5	V
ANALOG INPUTS⁽¹⁾					
FSR	Full-scale input voltage range ⁽²⁾ (V _{IN} = V _(AINP) - V _(AINN))	±0.256		±6.144	V
V _(AINN)	Absolute input voltage	GND		VDD	V
DIGITAL INPUTS					
V _{DIG}	Digital input voltage	GND		5.5	V
TEMPERATURE					
T _A	Operating ambient temperature	-40		125	°C

(1) AINP and AINN denote the selected positive and negative inputs, AINx denotes one of the four available analog inputs.

(2) This parameter expresses the full-scale range of the ADC scaling. No more than VDD + 0.3 V must be applied to the analog inputs of the device. See [Table 3](#) for more information.

7.4 Thermal Information

THERMAL METRIC ⁽¹⁾	ADS111x		UNIT	
	DGS (VSSOP)	RUG (X2QFN)		
	10 PINS	10 PINS		
R _{thJA}	Junction-to-ambient thermal resistance	182.7	245.2	°C/W
R _{thJC(top)}	Junction-to-case (top) thermal resistance	67.2	68.3	°C/W
R _{thJB}	Junction-to-board thermal resistance	103.8	172.0	°C/W
ψ _{JT}	Junction-to-top characterization parameter	10.2	8.2	°C/W
ψ _{JB}	Junction-to-board characterization parameter	102.1	179.8	°C/W
R _{thJC(bottom)}	Junction-to-case (bottom) thermal resistance	N/A	N/A	°C/W

(1) For more information about traditional and new thermal metrics, see the [Semiconductor and IC Package Thermal Metrics](#) application report.

Appendix B.3. Electrical Characteristics



ADS1113, ADS1114, ADS1115

www.ti.com

SBAS444D – MAY 2009 – REVISED JANUARY 2018

7.5 Electrical Characteristics

At VDD = 3.3 V, data rate = 8 SPS, and full-scale input voltage range (FSR) = ±2.048 V (unless otherwise noted). Maximum and minimum specifications apply from TA = -40°C to +125°C. Typical specifications are at TA = 25°C.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT	
ANALOG INPUT						
Common-mode input impedance	FSR = ±6.144 V ⁽¹⁾		10		MΩ	
	FSR = ±6.096 V ⁽¹⁾ , FSR = ±2.048 V		6			
	FSR = ±1.024 V		3			
	FSR = ±0.512 V, FSR = ±0.256 V		100			
Differential input impedance	FSR = ±6.144 V ⁽¹⁾		22		MΩ	
	FSR = ±6.096 V ⁽¹⁾		15			
	FSR = ±2.048 V		4.8			
	FSR = ±1.024 V		2.4			
	FSR = ±0.512 V, ±0.256 V		710			kΩ
SYSTEM PERFORMANCE						
Resolution (no missing codes)		16			Bits	
DR	Data rate		8, 16, 32, 64, 128, 250, 475, 860		SPS	
	Data rate variation	All data rates		±10%		
	Output noise		See <i>Noise Performance</i> section			
INL	Integral nonlinearity	DR = 8 SPS, FSR = ±2.048 V ⁽²⁾		1	LSB	
Offset error	FSR = ±2.048 V, differential inputs		-3	±1	3	LSB
		FSR = ±2.048 V, single-ended inputs		±3		
	Offset drift over temperature	FSR = ±2.048 V		0.005	LSB/°C	
	Long-term Offset drift	FSR = ±2.048 V, TA = 125°C, 1000 hrs		±1	LSB	
	Offset power-supply rejection	FSR = ±2.048 V, DC supply variation		1	LSB/V	
	Offset channel match	Match between any two inputs		3	LSB	
	Gain error ⁽²⁾	FSR = ±2.048 V, TA = 25°C		0.01%	0.15%	
Gain drift over temperature ⁽³⁾	FSR = ±0.256 V			7	ppm/°C	
		FSR = ±2.048 V		5		
		FSR = ±6.144 V ⁽¹⁾		5		
	Long-term gain drift ⁽³⁾	FSR = ±2.048 V, TA = 125°C, 1000 hrs		±0.05	%	
	Gain power-supply rejection			80	ppm/V	
	Gain match ⁽²⁾	Match between any two gains		0.02%	0.1%	
	Gain channel match	Match between any two inputs		0.05%	0.1%	
OMRR	Common-mode rejection ratio	At DC, FSR = ±0.256 V		105	dB	
		At DC, FSR = ±2.048 V		100		
		At DC, FSR = ±6.144 V ⁽¹⁾		90		
		f _{CMR} = 60 Hz, DR = 8 SPS		105		
		f _{CMR} = 60 Hz, DR = 8 SPS		105		
DIGITAL INPUT/OUTPUT						
V _{IH}	High-level input voltage		0.7 VDD	5.5	V	
V _{IL}	Low-level input voltage		GND	0.3 VDD	V	
V _{OL}	Low-level output voltage	I _{OL} = 3 mA	GND	0.15	V	
	Input leakage current	GND < V _{DI} < VDD		-10	10 μA	

(1) This parameter expresses the full-scale range of the ADC scaling. No more than VDD + 0.3 V must be applied to the analog inputs of the device. See Table 3 for more information.

(2) Best-fit INL, covers 99% of full-scale.

(3) Includes all errors from onboard PGA and voltage reference.

Electrical Characteristics (continued)

At VDD = 3.3 V, data rate = 8 SPS, and full-scale input voltage range (FSR) = ±2.048 V (unless otherwise noted). Maximum and minimum specifications apply from TA = -40°C to +125°C. Typical specifications are at TA = 25°C.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
POWER-SUPPLY					
I _{VDD} Supply current	Power-down	TA = 25°C	0.5	2	μA
				5	
	Operating	TA = 25°C	150	200	
				300	
P _D Power dissipation	VDD = 5.0 V		0.9		mW
	VDD = 3.3 V		0.5		
	VDD = 2.0 V		0.3		

7.6 Timing Requirements: I²C

over operating ambient temperature range and VDD = 2.0 V to 5.5 V (unless otherwise noted)

		FAST MODE		HIGH-SPEED MODE		UNIT
		MIN	MAX	MIN	MAX	
f _{SCL}	SCL clock frequency	0.01	0.4	0.01	3.4	MHz
t _{BUF}	Bus free time between START and STOP condition	600		160		ns
t _{HDATA}	Hold time after repeated START condition. After this period, the first clock is generated.	600		160		ns
t _{SUSTA}	Setup time for a repeated START condition	600		160		ns
t _{SUSTO}	Setup time for STOP condition	600		160		ns
t _{HDDAT}	Data hold time	0		0		ns
t _{SUDAT}	Data setup time	100		10		ns
t _{LOW}	Low period of the SCL clock pin	1300		160		ns
t _{HIGH}	High period for the SCL clock pin	600		80		ns
t _r	Rise time for both SDA and SCL signals ⁽¹⁾		300		160	ns
t _f	Fall time for both SDA and SCL signals ⁽¹⁾		300		160	ns

(1) For high-speed mode maximum values, the capacitive load on the bus line must not exceed 400 pF.

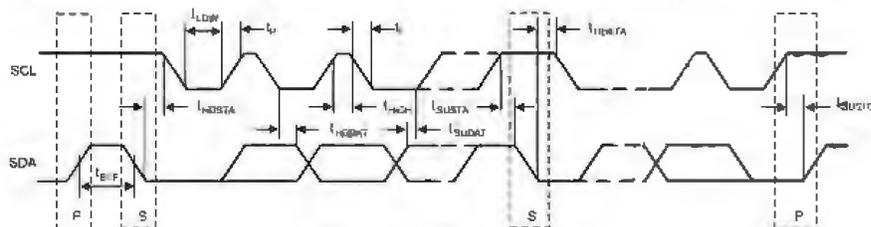


Figure 1. I²C Interface Timing

Appendix B.4. Programming



ADS1113, ADS1114, ADS1115

SBAS444D—MAY 2009—REVISED JANUARY 2018

www.ti.com

9.5 Programming

9.5.1 I²C Interface

The ADS111x communicate through an I²C interface. I²C is a two-wire open-drain interface that supports multiple devices and masters on a single bus. Devices on the I²C bus only drive the bus lines low by connecting them to ground; the devices never drive the bus lines high. Instead, the bus wires are pulled high by pullup resistors, so the bus wires are always high when no device is driving them low. As a result of this configuration, two devices cannot conflict. If two devices drive the bus simultaneously, there is no driver contention.

Communication on the I²C bus always takes place between two devices, one acting as the master and the other as the slave. Both the master and slave can read and write, but the slave can only do so under the direction of the master. Some I²C devices can act as a master or slave, but the ADS111x can only act as a slave device.

An I²C bus consists of two lines: SDA and SCL. SDA carries data; SCL provides the clock. All data are transmitted across the I²C bus in groups of eight bits. To send a bit on the I²C bus, drive the SDA line to the appropriate level while SCL is low (a low on SDA indicates the bit is zero; a high indicates the bit is one). After the SDA line settles, the SCL line is brought high, then low. This pulse on SCL clocks the SDA bit into the receiver shift register. If the I²C bus is held idle for more than 25 ms, the bus times out.

The I²C bus is bidirectional: that is, the SDA line is used for both transmitting and receiving data. When the master reads from a slave, the slave drives the data line; when the master sends to a slave, the master drives the data line. The master always drives the clock line. The ADS111x cannot act as a master, and therefore can never drive SCL.

Most of the time the bus is idle; no communication occurs, and both lines are high. When communication takes place, the bus is active. Only a master device can start a communication and initiate a START condition on the bus. Normally, the data line is only allowed to change state while the clock line is low. If the data line changes state while the clock line is high, it is either a START condition or a STOP condition. A START condition occurs when the clock line is high, and the data line goes from high to low. A STOP condition occurs when the clock line is high, and the data line goes from low to high.

After the master issues a START condition, the master sends a byte that indicates with which slave device to communicate. This byte is called the *address byte*. Each device on an I²C bus has a unique 7-bit address to which it responds. The master sends an address in the address byte, together with a bit that indicates whether the master wishes to read from or write to the slave device.

Every byte (address and data) transmitted on the I²C bus is acknowledged with an *acknowledge* bit. When the master finishes sending a byte (eight data bits) to a slave, the master stops driving SDA and waits for the slave to acknowledge the byte. The slave acknowledges the byte by pulling SDA low. The master then sends a clock pulse to clock the acknowledge bit. Similarly, when the master completes reading a byte, the master pulls SDA low to acknowledge this completion to the slave. The master then sends a clock pulse to clock the bit. The master always drives the clock line.

If a device is not present on the bus, and the master attempts to address it, it receives a *not-acknowledge* because no device is present at that address to pull the line low. A not-acknowledge is performed by simply leaving SDA high during an acknowledge cycle.

When the master has finished communicating with a slave, it may issue a STOP condition. When a STOP condition is issued, the bus becomes idle again. The master may also issue another START condition. When a START condition is issued while the bus is active, it is called a repeated start condition.

The [Timing Requirements](#) section shows a timing diagram for the ADS111x I²C communication.

Programming (continued)

9.5.1.1 I²C Address Selection

The ADS111x have one address pin, ADDR, that configures the I²C address of the device. This pin can be connected to GND, VDD, SDA, or SCL, allowing for four different addresses to be selected with one pin, as shown in Table 4. The state of address pin ADDR is sampled continuously. Use the GND, VDD and SCL addresses first. If SDA is used as the device address, hold the SDA line low for at least 100 ns after the SCL line goes low to make sure the device decodes the address correctly during I²C communication.

Table 4. ADDR Pin Connection and Corresponding Slave Address

ADDR PIN CONNECTION	SLAVE ADDRESS
GND	1001000
VDD	1001001
SDA	1001010
SCL	1001011

9.5.1.2 I²C General Call

The ADS111x respond to the I²C general call address (0000000) if the eighth bit is 0. The devices acknowledge the general call address and respond to commands in the second byte. If the second byte is 00000110 (06h), the ADS111x reset the internal registers and enter a power-down state.

9.5.1.3 I²C Speed Modes

The I²C bus operates at one of three speeds. Standard mode allows a clock frequency of up to 100 kHz; fast mode permits a clock frequency of up to 400 kHz; and high-speed mode (also called Hs mode) allows a clock frequency of up to 3.4 MHz. The ADS111x are fully compatible with all three modes.

No special action is required to use the ADS111x in standard or fast mode, but high-speed mode must be activated. To activate high-speed mode, send a special address byte of 00001xxx following the START condition, where xxx are bits unique to the Hs-capable master. This byte is called the Hs master code, and is different from normal address bytes; the eighth bit does not indicate read/write status. The ADS111x do not acknowledge this byte; the I²C specification prohibits acknowledgment of the Hs master code. Upon receiving a master code, the ADS111x switch on Hs mode filters, and communicate at up to 3.4 MHz. The ADS111x switch out of Hs mode with the next STOP condition.

For more information on high-speed mode, consult the I²C specification.

9.5.2 Slave Mode Operations

The ADS111x act as slave receivers or slave transmitters. The ADS111x cannot drive the SCL line as slave devices.

9.5.2.1 Receive Mode

In slave receive mode, the first byte transmitted from the master to the slave consists of the 7-bit device address followed by a low R/W bit. The next byte transmitted by the master is the [Address Pointer register](#). The ADS111x then acknowledge receipt of the Address Pointer register byte. The next two bytes are written to the address given by the register address pointer bits, P[1:0]. The ADS111x acknowledge each byte sent. Register bytes are sent with the most significant byte first, followed by the least significant byte.

9.5.2.2 Transmit Mode

In slave transmit mode, the first byte transmitted by the master is the 7-bit slave address followed by the high R/W bit. This byte places the slave into transmit mode and indicates that the ADS111x are being read from. The next byte transmitted by the slave is the most significant byte of the register that is indicated by the register address pointer bits, P[1:0]. This byte is followed by an acknowledgment from the master. The remaining least significant byte is then sent by the slave and is followed by an acknowledgment from the master. The master may terminate transmission after any byte by not acknowledging or issuing a START or STOP condition.

Appendix B.5. Connecting Multiple I²C Devices

ADS1113, ADS1114, ADS1115

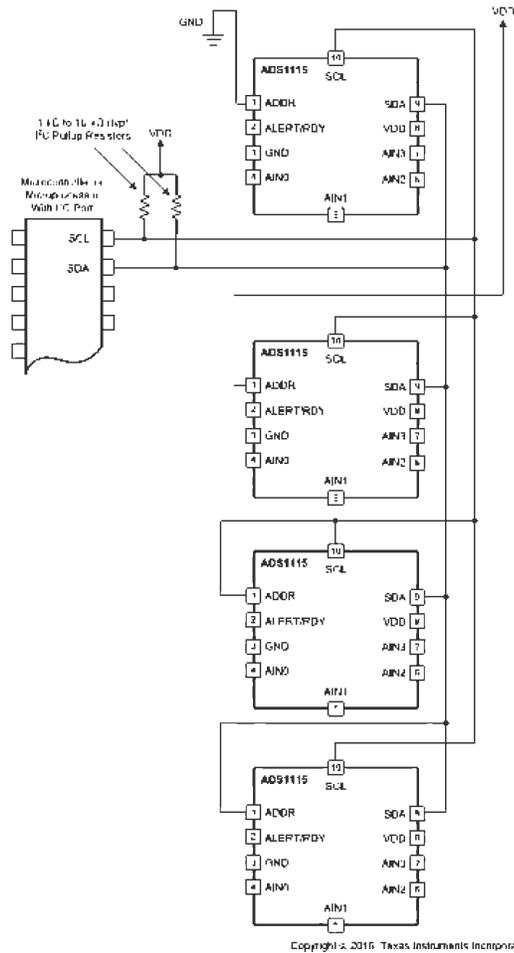
SBAS444D—MAY 2009—REVISED JANUARY 2018

www.ti.com

Application Information (continued)

10.1.6 Connecting Multiple Devices

It is possible to connect up to four ADS111x devices to a single I²C bus using different address pin configurations for each device. Use the address pin to set the ADS111x to one of four different I²C addresses. Use the GND, VDD and SCL addresses first. If SDA is used as the device address, hold the SDA line low for at least 100 ns after the SCL line goes low to make sure the device decodes the address correctly during I²C communication. An example showing four ADS111x devices on the same I²C bus is shown in Figure 42. One set of pullup resistors is required per bus. The pullup resistor values may need to be lowered to compensate for the additional bus capacitance presented by multiple devices and increased line length.



Copyright © 2016, Texas Instruments Incorporated

NOTE: ADS111x power and input connections omitted for clarity. The ADDR pin selects the I²C address.

Figure 42. Connecting Multiple ADS111x Devices

Appendix B.6. Description of Differential Comparator Function



www.ti.com

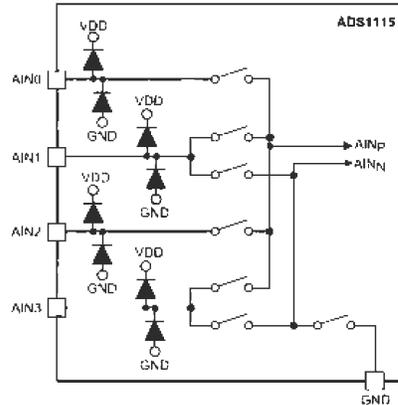
ADS1113, ADS1114, ADS1115

SBAS444D –MAY 2009–REVISED JANUARY 2018

9.3 Feature Description

9.3.1 Multiplexer

The ADS1115 contains an input multiplexer (MUX), as shown in Figure 25. Either four single-ended or two differential signals can be measured. Additionally, AIN0 and AIN1 may be measured differentially to AIN3. The multiplexer is configured by bits MUX[2:0] in the Config register. When single-ended signals are measured, the negative input of the ADC is internally connected to GND by a switch within the multiplexer.



Copyright © 2016 Texas Instruments Incorporated

Figure 25. Input Multiplexer

The ADS1113 and ADS1114 do not have an input multiplexer and can measure either one differential signal or one single-ended signal. For single-ended measurements, connect the AIN1 pin to GND externally. In subsequent sections of this data sheet, AIN_p refers to AIN0 and AIN_n refers to AIN1 for the ADS1113 and ADS1114.

Electrostatic discharge (ESD) diodes connected to VDD and GND protect the ADS111x analog inputs. Keep the absolute voltage of any input within the range shown in Equation 3 to prevent the ESD diodes from turning on.

$$\text{GND} - 0.3 \text{ V} < V_{\text{AIN}(X)} < \text{VDD} + 0.3 \text{ V} \quad (3)$$

If the voltages on the input pins can potentially violate these conditions, use external Schottky diodes and series resistors to limit the input current to safe values (see the *Absolute Maximum Ratings* table).

Appendix C. Dynamometer Plots

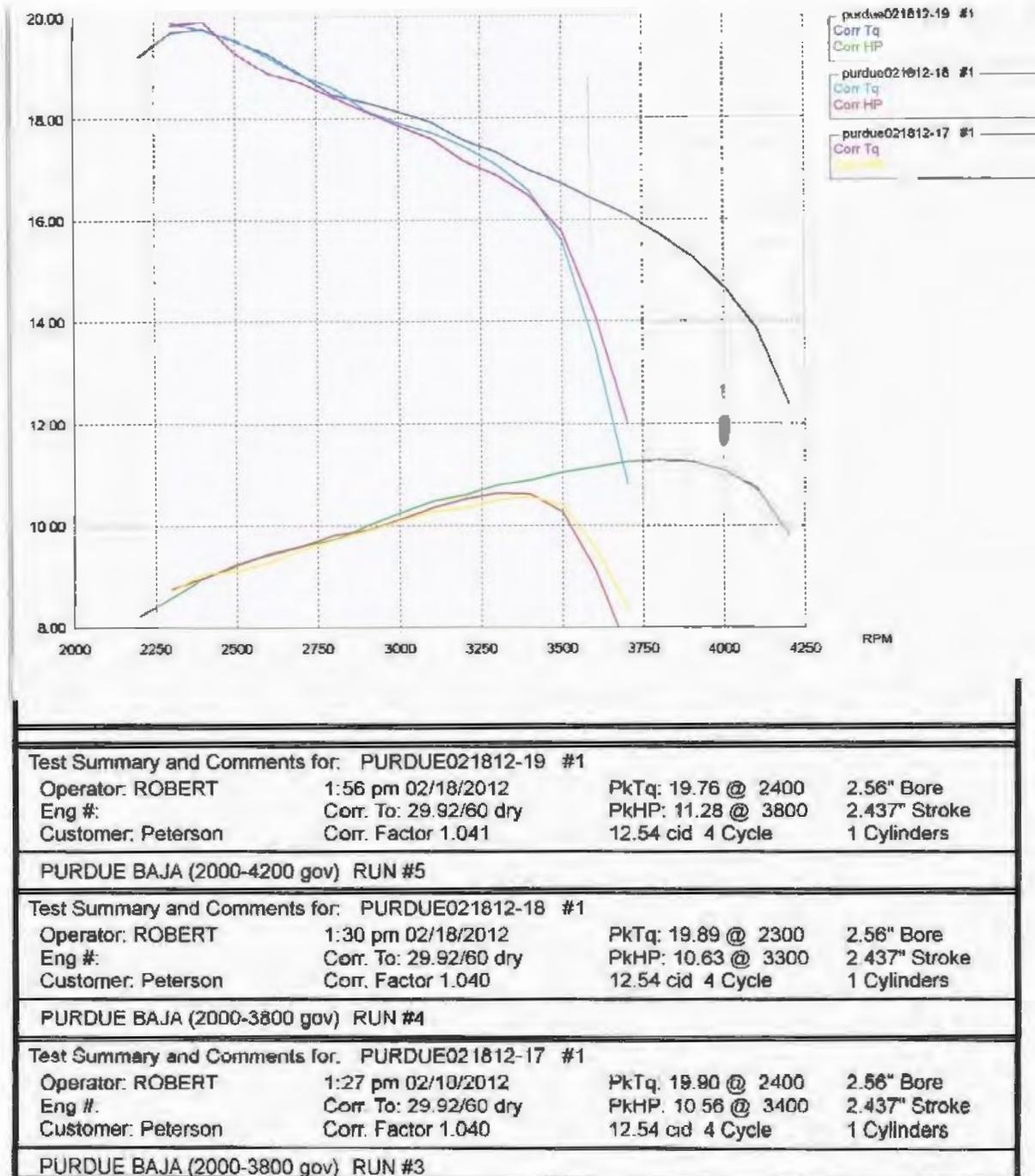
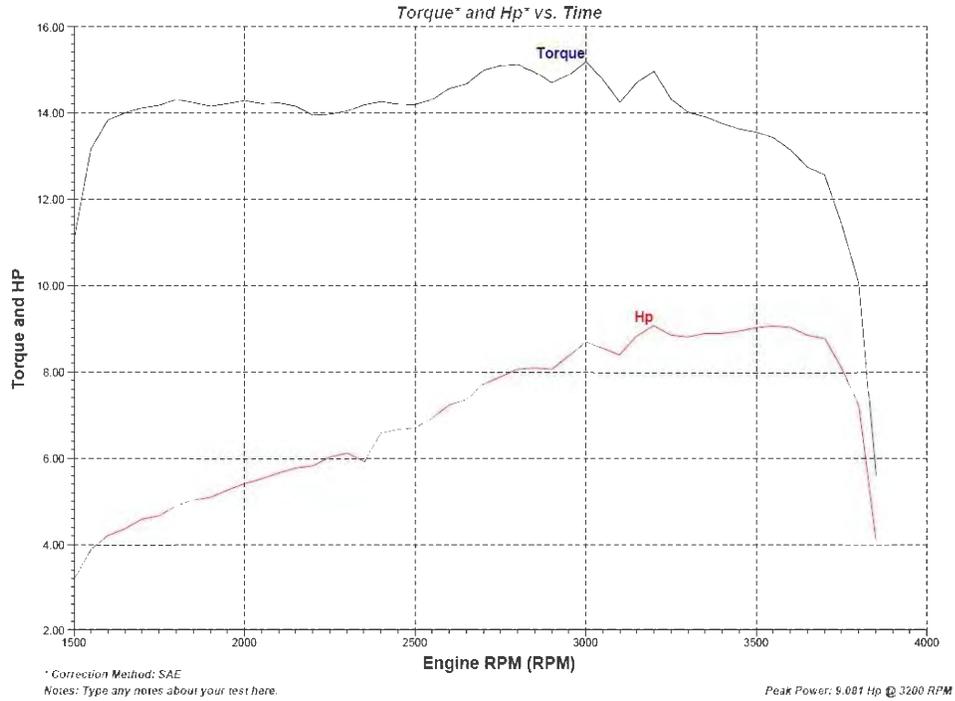


Figure 46: Purdue Dynamometer Reading, 2012 [11]



DYNomite test "My test name here on 2012-02-28 @ 22-02-35 " by University of Arkansas

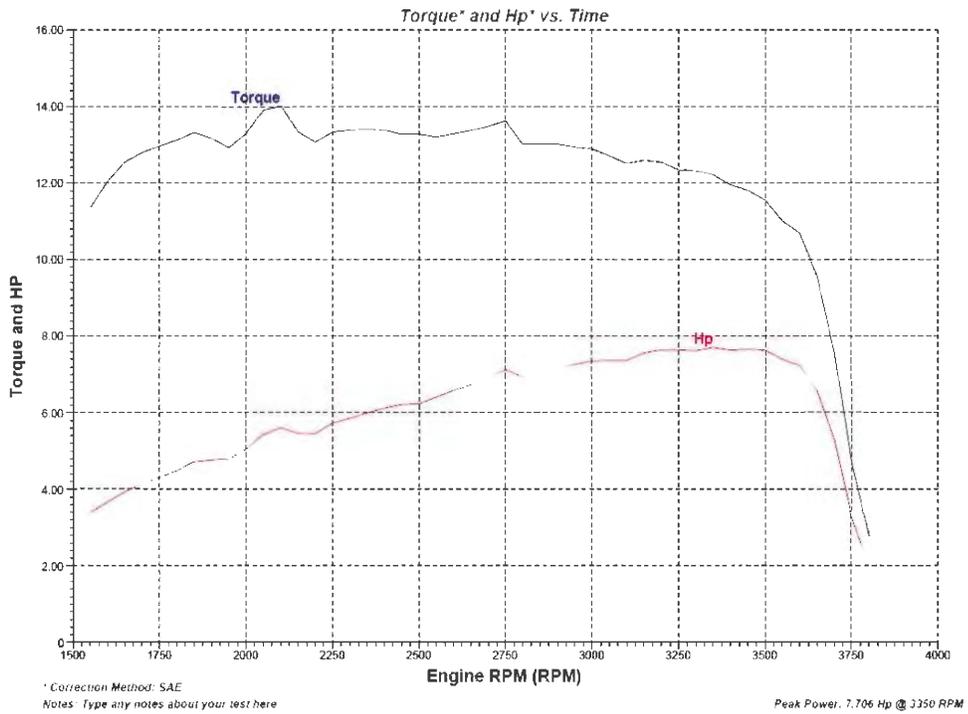


Figure 47: University of Arkansas Dynamometer Readings, 2012 [12]

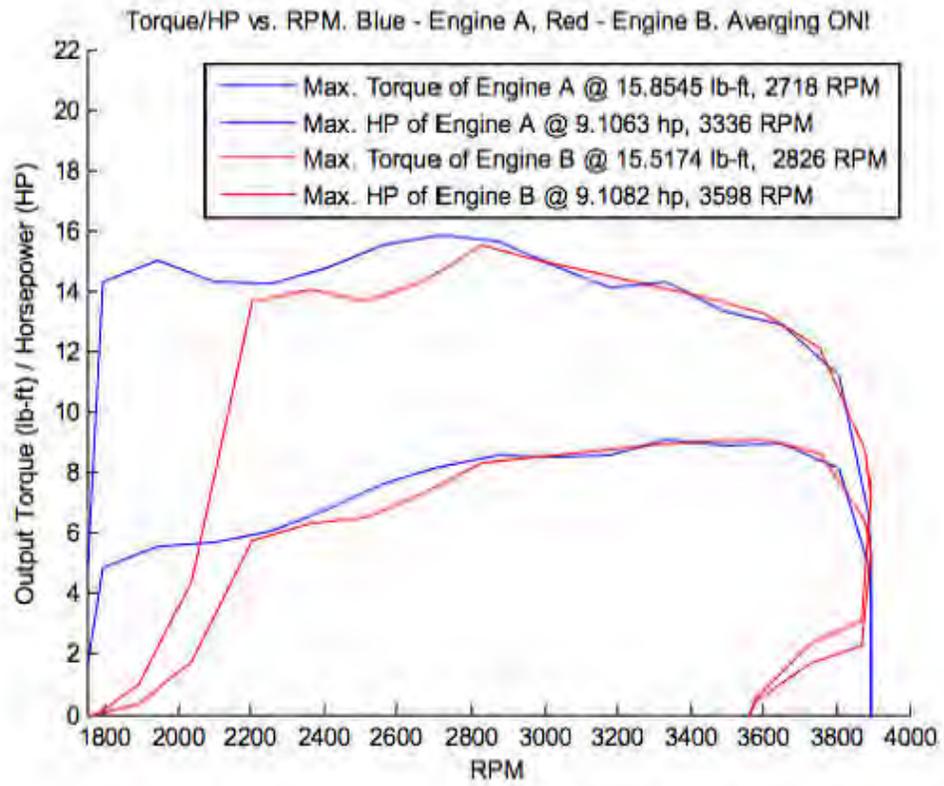


Figure 48: RIT Dynamometer Results [1]

Appendix D. ICM-20601 Data Sheets

Appendix D.1. Guide to Interfacing with ICM-20601 Breakout Board

The ICM-20601 uses standard I²C protocol, with a device address of 0x68. Use the following steps to interface with the board through the Arduino IDE. An example can be found in the file `Multiple_Data_String_Bluetooth_Test.ino` on the BCIT Racing GitHub.

Pin 20: SCL clock signal

Pin 22: SDA primary data line

1. Open I²C bus, set clock speed to 400000.
2. Ping ICM at 0x68, i.e.:

```
Wire.beginTransmission(0x68);
```

3. Choose the register you wish to interface with. For example, to write to the gyro command register at 0x1B:

```
Wire.write(byte(0x1B));
```

4. You now have selected the device, and the register of interest. Now, you can either read from or write to that register address. If we want to set the gyro to 4000 dps, we have to write to the configuration register, seen below in Appendix D.2. In this case, to select 4000 dps without selecting any other options, we want to write the binary value 00011000. Note that bit zero is on the right, bit 7 is on the left. This is equivalent to the hex value 0x18, so we command:

```
Wire.write(byte(0x18));
```

5. Close the communication stream to send/receive the data:

```
Wire.endTransmission();
```

Appendix D.2. Gyroscope Configuration

8.11 REGISTER 27 – GYROSCOPE CONFIGURATION

Register Name: GYRO_CONFIG

Register Type: READ/WRITE

Register Address: 27 (Decimal); 1B (Hex)

BIT	NAME	FUNCTION
[7]	XG_ST	X Gyro self-test
[6]	YG_ST	Y Gyro self-test
[5]	ZG_ST	Z Gyro self-test
[4:3]	FS_SEL[1:0]	Gyro Full Scale Select: 00 = ± 500 dps 01 = ± 1000 dps 10 = ± 2000 dps 11 = ± 4000 dps
[2]	-	Reserved
[1:0]	FCHOICE_B[1:0]	Used to bypass DLPF as shown in table 1 above.

Appendix D.3. Low-Power Mode Configuration

3.14 REGISTER 30 – LOW POWER MODE CONFIGURATION

Register Name: LP_MODE_CFG

Register Type: READ/WRITE

Register Address: 30 (Decimal); 1E (Hex)

Bit	Name	Function																												
[7]	GYRO_CYCLE	When set to '1' low power gyroscope mode is enabled. Default setting is '0'																												
[6:4]	G_AVGCFG[2:0]	Averaging filter configuration for low-power gyroscope mode. Default setting is '000'																												
[3:0]	LPOSC_CLKSEL	Sets the frequency of waking up the chip or take a sample of accel data (the low power accel Output Data Rate) <table border="1" data-bbox="868 541 1177 911"> <thead> <tr> <th>LPOSC_CLKSEL</th> <th>Output Frequency (Hz)</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.24</td></tr> <tr><td>1</td><td>0.49</td></tr> <tr><td>2</td><td>0.98</td></tr> <tr><td>3</td><td>1.95</td></tr> <tr><td>4</td><td>3.91</td></tr> <tr><td>5</td><td>7.81</td></tr> <tr><td>6</td><td>15.63</td></tr> <tr><td>7</td><td>31.25</td></tr> <tr><td>8</td><td>62.50</td></tr> <tr><td>9</td><td>125</td></tr> <tr><td>10</td><td>250</td></tr> <tr><td>11</td><td>500</td></tr> <tr><td>12-15</td><td>Reserved</td></tr> </tbody> </table>	LPOSC_CLKSEL	Output Frequency (Hz)	0	0.24	1	0.49	2	0.98	3	1.95	4	3.91	5	7.81	6	15.63	7	31.25	8	62.50	9	125	10	250	11	500	12-15	Reserved
LPOSC_CLKSEL	Output Frequency (Hz)																													
0	0.24																													
1	0.49																													
2	0.98																													
3	1.95																													
4	3.91																													
5	7.81																													
6	15.63																													
7	31.25																													
8	62.50																													
9	125																													
10	250																													
11	500																													
12-15	Reserved																													

To operate in gyroscope low-power mode or 6-axis low-power mode, GYRO_CYCLE should be set to '1'. Gyroscope filter configuration is determined by G_AVGCFG[2:0] that sets the averaging filter configuration. It is not dependent on DLPF_CFG[2:0].

The following table shows some example configurations for gyroscope low power mode:

FCHOICE_B	0	0	0	0	0	0	0	0	
G_AVGCFG	0	1	2	3	4	5	6	7	
Averages	1x	2x	4x	8x	16x	32x	64x	128x	
Tau (ms)	1.73	2.23	3.23	5.23	9.23	17.23	33.23	65.23	
Noise BW (Hz)	650.8	407.1	274.2	177.4	80.7	30.6	15.6	8.0	
3-dB BW (Hz)	622	391	211	108	54	27	14	7	
SAMPLRT_DIV	ODR (Hz)								
255	3.9	Valid							
99	10.0	Valid							
65	15.2	Valid							
64	15.4	Valid							
49	25.0	Valid							
33	29.4	Valid							
32	30.3	Valid							
19	50.0	Valid							
17	55.6	Valid							
16	58.8	Valid							
9	100.0	Valid							
8	111.1	Valid							
7	125.0	Valid							
4	200.0	Valid							
3	250.0	Valid							
2	333.3	Valid							
1	500.0	Valid							

Appendix D.4. Register Map

NOTE: Echo registers 66 through 72 to obtain gyroscope data. See Appendix D.1.

7 REGISTER MAP

The following table lists the register map for the ICM-20601.

Addr (Hex)	Addr (Dec)	Register Name	Size (b)	Reset Value (0x0000)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
00	00	SELF_TEST_0_GYRO	R/W	N	WB_ST_DATA[4:0]								
01	01	SELF_TEST_1_GYRO	R/W	N	YG_ST_DATA[4:0]								
02	02	SELF_TEST_2_GYRO	R/W	N	ZB_ST_DATA[4:0]								
03	03	SELF_TEST_0_ACCEL	R/W	N	XA_ST_DATA[7:0]								
04	04	SELF_TEST_1_ACCEL	R/W	N	YA_ST_DATA[7:0]								
05	05	SELF_TEST_2_ACCEL	R/W	N	ZA_ST_DATA[7:0]								
13	13	YG_OFFSET_USRH	R/W	N	Y_OFFSET_USR[13:0]								
14	14	YG_OFFSET_USRL	R/W	N	Y_OFFSET_USR[13:0]								
15	15	YG_OFFSET_USRH	R/W	N	Y_OFFSET_USR[13:0]								
16	16	YG_OFFSET_USRL	R/W	N	Y_OFFSET_USR[13:0]								
17	17	ZG_OFFSET_USRH	R/W	N	Z_OFFSET_USR[15:0]								
18	18	ZG_OFFSET_USRL	R/W	N	Z_OFFSET_USR[15:0]								
23	23	SAMPLRT_DIV	R/W	N	SAMPLRT_DIV[7:0]								
2A	26	CONFIG	R/W	N	-	FIFO_MODE	EXT_SYNC_SEL[2:0]			DLFF_CFG[2:0]			
3B	27	GYRO_CONFIG	R/W	N	YG_ST	YG_ST	ZB_ST	YS_SEL[1:0]			FCHOICE_B[1:0]		
3C	28	ACCEL_CONFIG	R/W	N	XA_ST	YA_ST	ZA_ST	ACCEL_FS_SEL[2:0]					
1D	29	ACCFI_CONFIG_2	R/W	N	-			DFC2_CFG		ACCEL_ROMB1_CE_B	A_D1_CFG		
3E	30	LP_MODE_CFG	R/W	N	GYRO_CYCLE	G_AVG_CFG[2:0]			-		IPOSEL_CFG		
3F	31	ACCEL_WDM_THR	R/W	N	WDM_THR[7:0]								
23	35	RF0_EN	R/W	N	RF0P_EN	XG_RF0_EN	YG_RF0_EN	ZG_RF0_EN	ACCEL_RF0_EN	-	-	-	
36	64	FSYNC_INT	R/W	N	FSYNC_INT	-						-	-
37	55	INT_PIN_CFG	R/W	Y	INT_PUFI	INT_OFFN	LATCH_INT_EN	INT_RD_CLEAR	FSYNC_INT_LVL	FSYNC_INT_MODF_EN	-	-	
38	56	INT_ENABLE	R/W	Y	WDM_INT_EN[7:5]			FIFO_OVERFLOW_INT	-	GDRIVE_INT_EN	-	DATA_RDY_INT_EN	
3A	58	INT_STATUS	R/C	N	WDM_INT[7:5]			FIFO_OVERFLOW_INT	-	GDRIVE_INT	-	DATA_RDY_INT	
3B	59	ACCEL_XOUT_H	R	N	ACCEL_XOUT_H[15:0]								
3C	60	ACCEL_XOUT_L	R	N	ACCEL_XOUT_L[15:0]								
3D	61	ACCEL_YOUT_H	R	N	ACCEL_YOUT_H[15:0]								
3E	62	ACCEL_YOUT_L	R	N	ACCEL_YOUT_L[15:0]								
3F	63	ACCEL_ZOUT_H	R	N	ACCEL_ZOUT_H[15:0]								
40	64	ACCEL_ZOUT_L	R	N	ACCEL_ZOUT_L[15:0]								
41	65	TEMP_OUT_H	R	N	TEMP_OUT[15:0]								
42	66	TEMP_OUT_L	R	N	TEMP_OUT[15:0]								
43	67	GYRO_XOUT_0	R	N	GYRO_XOUT[16:0]								
44	68	GYRO_XOUT_1	R	N	GYRO_XOUT[17:0]								
45	69	GYRO_XOUT_2	R	N	GYRO_XOUT[18:0]								
46	70	GYRO_YOUT_0	R	N	GYRO_YOUT[17:0]								
47	71	GYRO_YOUT_1	R	N	GYRO_YOUT[18:0]								
48	72	GYRO_YOUT_2	R	N	GYRO_YOUT[19:0]								
6A	104	SIGNAL_PATH_RESET	R/W	N	-	-	-	-	-	-	ACCEL_RST	TEMP_RST	
69	105	ACCEL_INTEL_CTR	R/W	N	ACCEL_INTEL_EN	ACCEL_INTEL_MODE	-						
6A	106	USER_CTR	R/W	N	-	FIFO_EN	-	ISEL_CFG	-	RF0_RST	-	SILENCE_RST	
6B	107	PWR_MGMT_1	R/W	Y	DEVICE_RESET	SLEEP	ACCEL_CYCLE	GYRO_STANDBY	TEMP_DIS	CLOSE[1:0]			

Addr (Hex)	Addr (Dec.)	Register Name	Serial ID	Access (Read/Write/Read/Write)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
02	106	INT_STATUS	R/W	R	INT_EN	-	INT_STA	INT_STA	INT_STA	INT_STA	INT_STA	INT_STA
03	107	INT_STATUS	R	R	-	-	-	INT_STATUS(3-8)				
04	108	INT_STATUS	R/W	R	-	-	-	INT_STATUS(9)				
05	109	WHO_AM_I	R	R	-	-	-	WHO_AM_I(0)				
06	110	INT_STATUS	R/W	R	-	-	-	INT_STATUS(4-7)				
07	111	INT_STATUS	R/W	R	-	-	-	INT_STATUS(8)				
08	112	INT_STATUS	R/W	R	-	-	-	INT_STATUS(9)				
09	113	INT_STATUS	R/W	R	-	-	-	INT_STATUS(10)				
0A	114	INT_STATUS	R/W	R	-	-	-	INT_STATUS(11)				
0B	115	INT_STATUS	R/W	R	-	-	-	INT_STATUS(12)				
0C	116	INT_STATUS	R/W	R	-	-	-	INT_STATUS(13)				
0D	117	INT_STATUS	R/W	R	-	-	-	INT_STATUS(14)				
0E	118	INT_STATUS	R/W	R	-	-	-	INT_STATUS(15)				

Note: Register Names ending in `_H` and `_L` contain the high and low bytes, respectively, of an internal register value.

In the detailed register tables that follow, register names are in capital letters, while register values are in capital letters and italicized. For example, the `ACCEL_XOUT_H` register (Register 59) contains the 8 most significant bits, `ACCEL_XOUT[15:8]`, of the 16-bit X-Axis accelerometer measurement, `ACCEL_XOUT`.

The reset value is 0x00 for all registers other than the registers below, also the self-test registers contain pre-programmed values and will not be 0x00 after reset:

- Register 107 (0x40) Power Management 1
- Register 117 (0xAC) WHO_AM_I

Appendix E. BCIT Racing Request for Proposal



Request for Proposals

Remote Torque and RPM Data Logger

Project Managers: Jeremiah Moreno, Russ Case
Telephone:
RFP Number: 1
Issue date: October 17, 2018
Closing Time: Proposals must be received **before 4:00 PM**, October 26, 2018.

DELIVERY OF PROPOSALS

Proposals must be submitted using one of the submission methods below.

Email Submission: Project proponents may submit electronic proposals by email. Emailed proposals must be submitted to procurement@bcitracing.com.

Hard Copy Submission: Proposals submitted by hard copy must be delivered to:

BCIT Racing Procurement
3700 Willingdon Avenue
Burnaby, B.C. Canada, V5G 3H2

Regardless of submission method, proposals must be received before Closing Time to be considered.

SUMMARY

BCIT Racing is in the process of prototyping and testing an off-road race vehicle, with the intention of finalizing the design process and beginning a limited production run of 4,000 units no later than Summer, 2020.

To achieve this goal while maintaining stringent performance and reliability standards, the vehicle design team requires testing of dynamic drivetrain load conditions - specifically, peak torque and power loads at the drive axles. These peak loads generally occur when the vehicle is being driven in race-like conditions, and so a need has been identified for the development of a robust and compact torque and RPM sensor, with remote data transmission capabilities.

As BCIT Racing's engineers are not specialists in the field of telemetry and data acquisition, the project team has elected to outsource the design of this torque sensor, and will be accepting proposals for the development of a working prototype.

BCIT RACING

PROJECT SCOPE AND REQUIREMENTS

The Data Logger design team will be responsible for conducting relevant research and planning, generating/presenting concepts, identifying required components, and creating a functional prototype.

The following baseline design parameters will be met:

- Must measure axle torque and rotation speed
- Must be removable with minimal effort, i.e., under 10 minutes
- Components must be contained in a watertight housing
- Must be calibrated to make accurate measurements upon delivery
- Must permit normal vehicle operation when installed
- Must transmit and store data for later analysis

TIMELINE AND KEY DATES

Proposal submission deadline:	October 26, 2018
Design concept review:	November 13, 2018
Design approval:	December 18, 2018
Prototype completion:	May 6, 2019

BUDGET

A budget has not been defined for this project. However, proposed costs should be included in proposal bids, and will be used as a selection criterion. A per-unit cost of less than \$1000 should be targeted.

PROPOSAL FORMAT

Project proposals should include the following:

- Short summary of key features/aspects of the proposal
- Breakdown and description of the proposed components
- Rough cost estimate, accounting for major off-the-shelf components
- Estimated project timeline

EVALUATION CRITERIA

Evaluation of proposals will be by a committee of BCIT Racing design engineers. Proposals will be scored based on design robustness and potential ease of use. Robust designs will receive higher scores, however total system weight should be reduced to a minimum. Controlled costs and realistic timeline estimates will also be taken into consideration.

Appendix F. Strain Gauge Application Guide



Instruction Bulletin B-127

Strain Gage Installations with M-Bond 200 Adhesive

INTRODUCTION

Micro-Measurements Certified M-Bond 200 is an excellent general-purpose laboratory adhesive because of its fast room-temperature cure and ease of application. When properly handled and used with the appropriate strain gage, M-Bond 200 can be used for high-elongation tests in excess of 60 000 microstrain, for fatigue studies, and for one-cycle proof tests to over +200 °F [+95 °C] or below -300 °F [-185°C]. The normal operating temperature range is -25° to +150°F [-30° to +65°C]. M-Bond 200 is compatible with all Micro-Measurements strain gages and most common structural materials. When bonding to plastics, it should be noted that for best performance the adhesive flowout should be kept to a minimum. For best reliability, it should be applied to surfaces between the temperatures of +70° and +85°F [+20° to +30°C], and in a relative humidity environment of 30% to 65%.

M-Bond 200 catalyst has been specially formulated to control the reactivity rate of this adhesive. The catalyst should be used sparingly for best results. Excessive catalyst can contribute many problems, e.g., poor bond strength, age-embrittlement of the adhesive, poor glue-line thickness control, extended solvent evaporation time requirements, etc.

Since M-Bond 200 bonds are weakened by exposure to high humidity, adequate protective coatings are essential. This adhesive will gradually become harder and more brittle with time, particularly if exposed to elevated temperatures. For these reasons, M-Bond 200 is not generally recommended for installations exceeding one or two years.

For proper results, the procedures and techniques presented here should be used with qualified Micro-Measurements installation accessory products. Those used in this procedure are:

- CSM Degreaser or GC-6 Isopropyl Alcohol
- Silicon Carbide Paper
- M-Prep Conditioner A
- M-Prep Neutralizer 5A
- GSP-1 Gauze Sponges
- CSP-1 Cotton Applicators
- PCT Gage Installation Tape

SHELF AND STORAGE LIFE

M-Bond 200 adhesive has a minimum pot life of three months at +75°F [+24°C] (not to exceed the date of expiration) after opening and with the cap placed back onto the bottle immediately after each use.

Note: To ensure the cap provides a proper seal, the bottle spout should be wiped clean and dry before replacing the cap.

Unopened M-Bond 200 adhesive may be stored up to nine months at +75°F [+24°C] or twelve months at +40°F [+5°C].

HANDLING PRECAUTIONS

M-Bond 200 is a modified alkyl cyanoacrylate compound. Immediate bonding of eye, skin or mouth may result upon contact. Causes irritation. The user is cautioned to: (1) avoid contact with skin; (2) avoid prolonged or repeated breathing of vapors; and (3) use with adequate ventilation. For additional health and safety information, consult the Safety Data Sheet, which is available upon request.

Note: Condensation will rapidly degrade adhesive performance and shelf life; after refrigeration the adhesive must be allowed to reach room temperature before opening, and refrigeration after opening is not recommended.

GAGE APPLICATION TECHNIQUES

The installation procedure presented on the following pages is somewhat abbreviated and is intended only as a guide in achieving proper gage installation with M-Bond 200. Micro-Measurements Instruction Bulletin B-129 presents recommended procedures for surface preparation, and lists specific considerations which are helpful when working with most common structural materials.

Strain Gage Installations with M-Bond 200 Adhesive

Document No : 11127
Revision 02-12-18

1 of 4
micro-measurements@vpgsensors.com

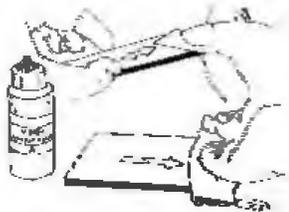
ID: BUL11127
www.micro-measurements.com

Step 1



Thoroughly degrease the gaging area with solvent, such as CSM Degreaser or GC-6 Isopropyl Alcohol. The former is preferred, but there are some materials (e.g., titanium and many plastics) that react with strong solvents. In these cases, GC-6 Isopropyl Alcohol should be considered. All degreasing should be done with uncontaminated solvents—thus the use of “one-way” containers, such as aerosol cans, is highly advisable.

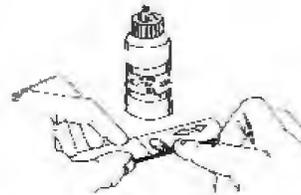
Step 2



Preliminary dry abrading with 220- or 320-grit silicon-carbide paper is generally required if there is any surface scale or oxide. Final abrading is done by using 320-grit silicon-carbide paper on surfaces thoroughly wetted with M-Prep Conditioner A; this is followed by wiping dry with a gauze sponge. Repeat this wet abrading process with 400-grit silicon-carbide paper, then dry by slowly wiping through with a gauze sponge.

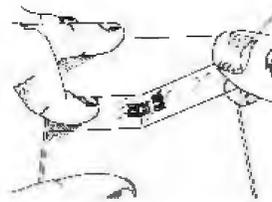
Using a 4H pencil (on aluminum) or a ballpoint pen (on steel), burnish (do not scribe) whatever alignment marks are needed on the specimen. Repeatedly apply M-Prep Conditioner A and scrub with cotton-tipped applicators until a clean tip is no longer discolored. Remove all residue and Conditioner by again slowly wiping through with a gauze sponge. Never allow any solution to dry on the surface because this invariably leaves a contaminating film and reduces chances of a good bond.

Step 3



Now apply a liberal amount of M-Prep Neutralizer 5A and scrub with a cotton-tipped applicator. With a single, slow wiping motion of a gauze sponge, carefully dry this surface. Do not wipe back and forth because this may allow contaminants to be redeposited.

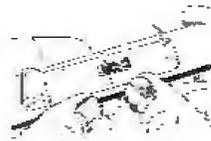
Step 4



Using tweezers to remove the gage from the transparent envelope, place the gage (bonding side down) on a chemically clean glass plate or gage box surface. If a solder terminal will be used, position it on the plate adjacent to the gage as shown. A space of approximately 1/16 in [1.6 mm] or more where space allows or application requires should be left between the gage backing and terminal. Place a 4- to 6-in [100- to 150-mm] piece of Micro-Measurements PCT gage installation tape over the gage and terminal. Take care to center the gage on the tape. Carefully lift the tape at a shallow angle (about 45 degrees to specimen surface), bringing the gage up with the tape as illustrated above.

Strain Gage Installations with M-Bond 200 Adhesive

Step 5



Position the gage/tape assembly so that the triangle alignment marks on the gage are over the layout lines on the specimen. If the assembly appears to be misaligned, lift one end of the tape at a shallow angle until the assembly is free of the specimen. Realign properly, and firmly anchor at least one end of the tape to the specimen. Realignment can be done without fear of contamination by the tape mastic if Micro-Measurements PCT gage installation tape is used, because this tape will retain its mastic when removed.

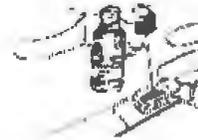
Step 6



Lift the gage end of the tape assembly at a shallow angle to the specimen surface (about 45 degrees) until the gage and terminal are free of the specimen surface. Continue lifting the tape until it is free from the specimen approximately 1/2 in [10 mm] beyond the terminal. Tuck the loose end of the tape under and press to the specimen surface so that the gage and terminal lie flat, with the bonding surface exposed.

Note: Micro-Measurements gages have been treated for optimum bonding conditions and require no pre-cleaning before use unless contaminated during handling. If contaminated, the back of any gage can be cleaned with a cotton-tipped applicator slightly moistened with M-Prep Neutralizer 5A.

Step 7



M-Bond 200 catalyst can now be applied to the bonding surface of the gage and terminal. M-Bond 200 adhesive will harden without the catalyst, but less quickly and reliably. Very little catalyst is needed, and it should be applied in a thin, uniform coat. Lift the brush-cap out of the catalyst bottle and wipe the brush approximately 10 strokes against the inside of the neck of the bottle to wring out most of the catalyst. Set the brush down on the gage and swab the gage backing. Do not stroke the brush in a painting style, but slide the brush over the entire gage surface and then the terminal. Move the brush to the adjacent tape area prior to lifting from the surface. Allow the catalyst to dry at least one minute under normal ambient conditions of +75°F [+24°C] and 30% to 65% relative humidity before proceeding.

Note: The next three steps must be completed in the sequence shown, within 3 to 5 seconds. Read Steps 8, 9, and 10 before proceeding.

Step 8



Lift the tucked-under tape end of the assembly, and, holding in the same position, apply one or two drops of M-Bond 200 adhesive at the fold formed by the junction of the tape and specimen surface. This adhesive application should be approximately 1/2 in [13 mm] outside the actual gage installation area. This will insure that local polymerization that takes place when the adhesive comes in contact with the specimen surface will not cause unevenness in the gage guideline.

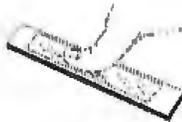
Strain Gage Installations with M-Bond 200 Adhesive

Step 9



Immediately rotate the tape to approximately a 30-degree angle so that the gage is bridged over the installation area. While holding the tape slightly taut, slowly and firmly make a single wiping stroke over the gage/tape assembly with a piece of gauze bringing the gage back down over the alignment marks on the specimen. Use a firm pressure with your fingers when wiping over the gage. A very thin, uniform layer of adhesive is desired for optimum bond performance.

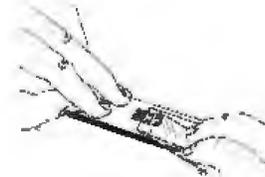
Step 10



Immediately upon completion of wipe-out of the adhesive, firm thumb pressure must be applied to the gage and terminal area. This pressure should be held for at least one minute. In low-humidity conditions (below 30%), or if the ambient temperature is below +70°F [+20°C], this pressure application time may have to be extended to several minutes.

Where large gages are involved, or where curved surfaces such as fillets are encountered, it may be advantageous to use preformed pressure padding during the operation. Pressure-application time should again be extended due to the lack of "thumb heat" which helps to speed adhesive polymerization. Wait two minutes before removing tape.

Step 11



The gage and terminal strip are now solidly bonded in place. It is not necessary to remove the tape immediately after gage installation. The tape will offer mechanical protection for the grid surface and may be left in place until it is removed for gage wiring. To remove the tape, pull it back directly over itself, peeling it slowly and steadily off the surface. This technique will prevent possible lifting of the foil on open-faced gages or other damage to the installation.

FINAL INSTALLATION PROCEDURE

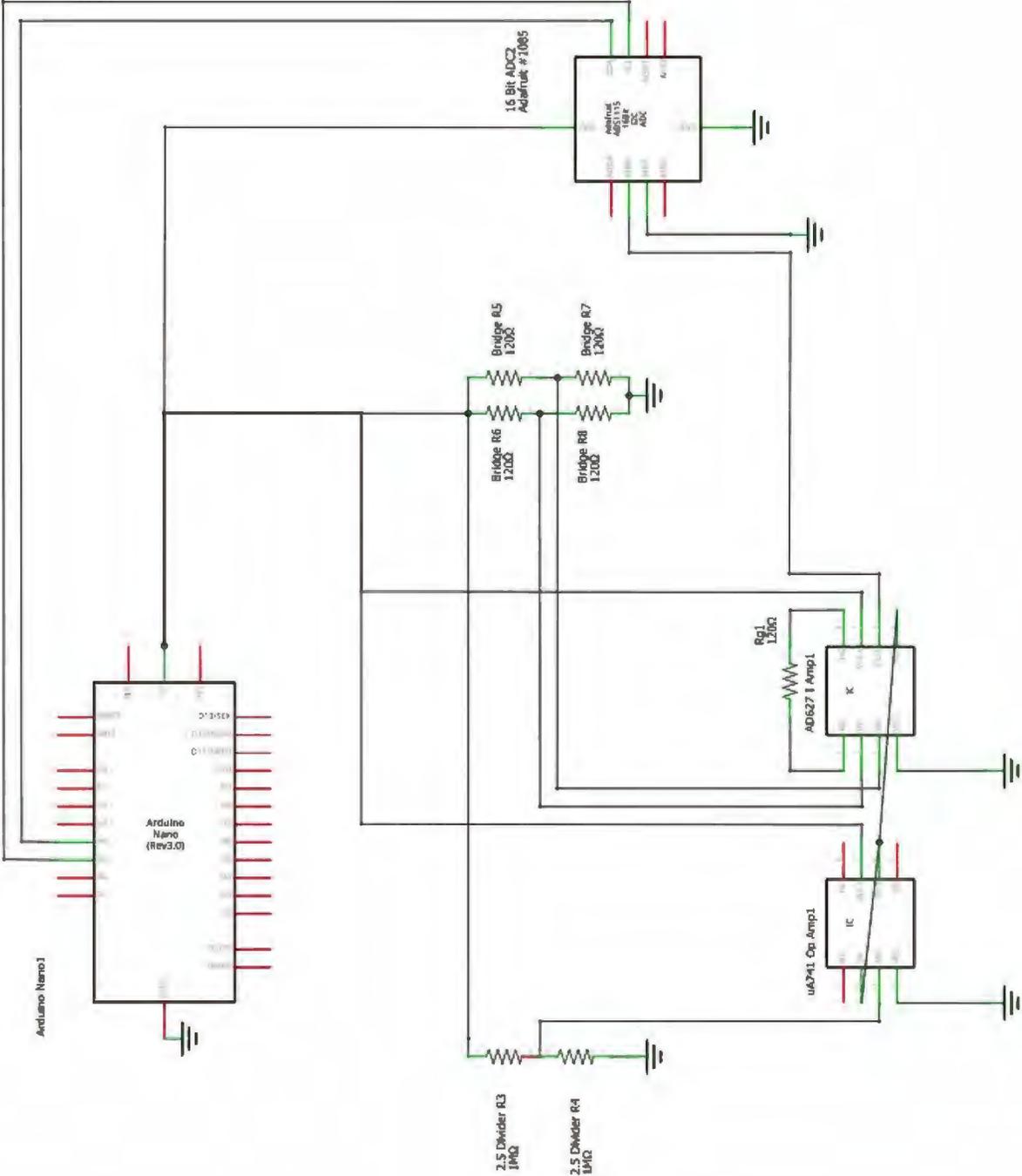
1. Select appropriate solder and attach leadwires. Prior to any soldering operations, open-faced gage grids should be masked with PDT drafting tape to prevent possible damage.
2. Remove the solder flux with Rosin Solvent, RSK-1.
3. Select and apply protective coating according to the protective coating selection chart found in the Micro Measurements Strain Gage Accessories Data Book found at <http://www.vishaypg.com/micro-measurements/databooks/>.

Strain Gage Installations with M-Bond 200 Adhesive

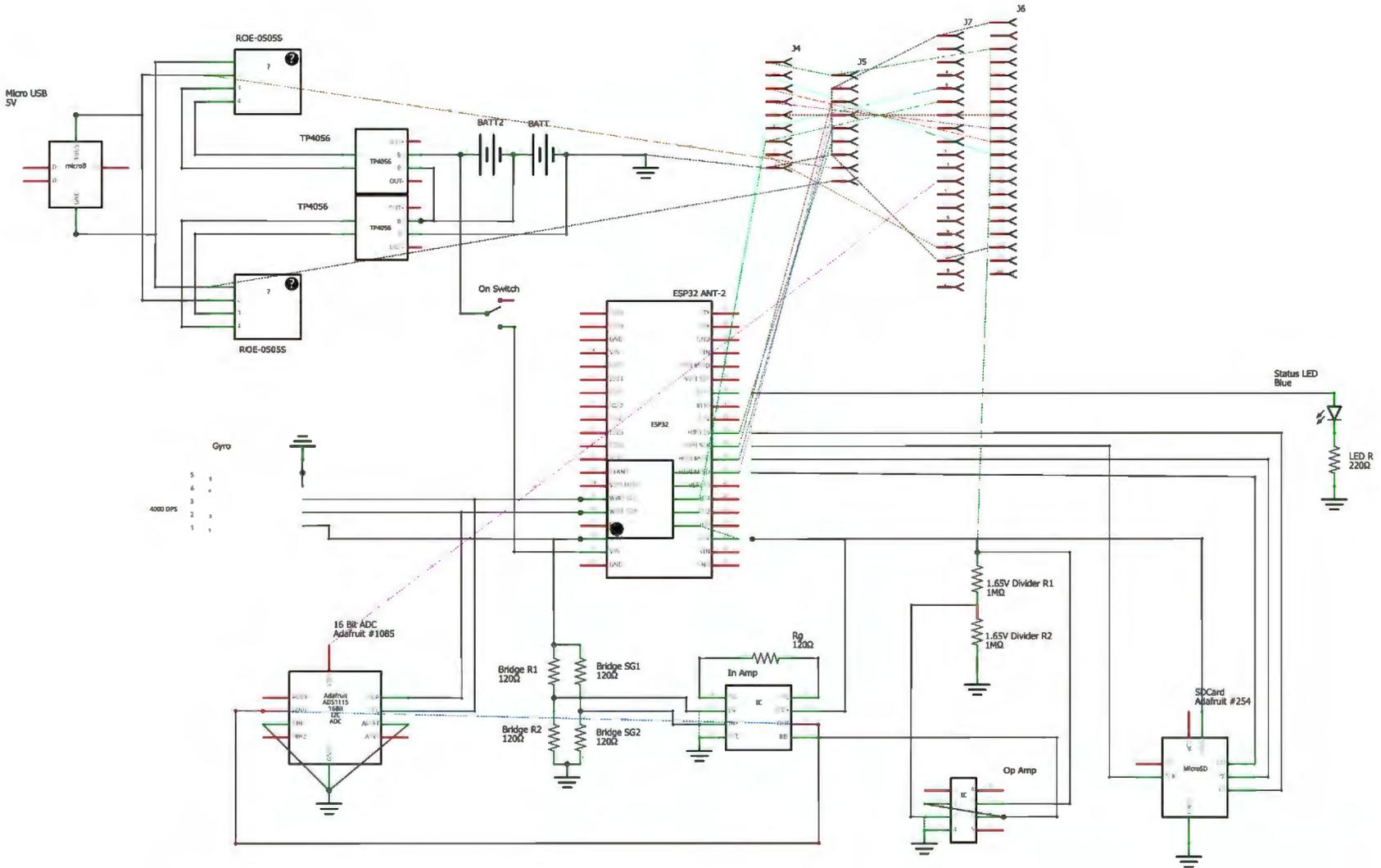
Appendix G. Wiring Diagrams

Note: Vector-format Wiring Diagrams can be found on the BCIT Racing GitHub:
https://github.com/BCITRacing/Storq/tree/master/Wiring_Diagrams

Appendix G.1. Torsion Test Rig

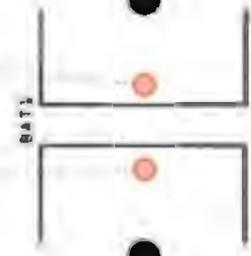
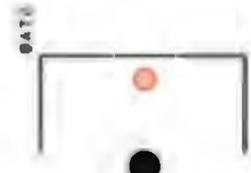
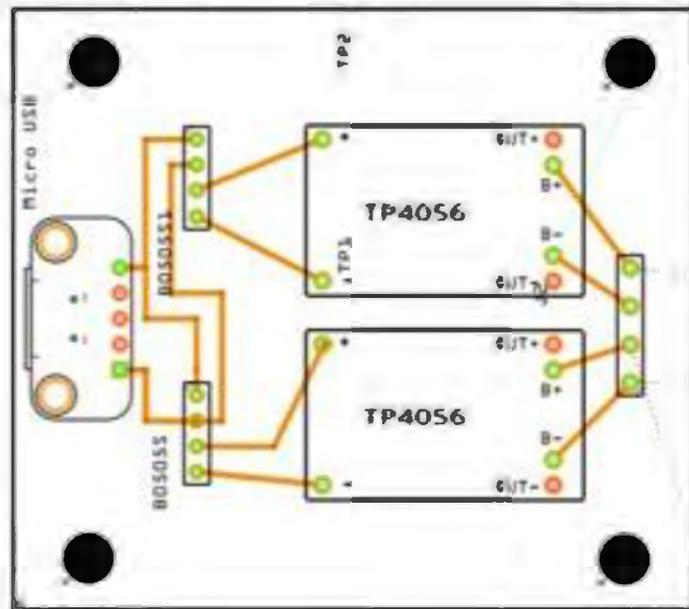


Appendix G.2. System Diagram



fritzing

Appendix G.3. Battery Charging Circuit

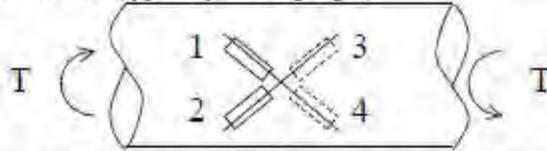


Appendix H. Torque Transducer Notes

Note: from Taco Niet's Force, Pressure Lecture Notes

Torque Transducers:

Basic Torque sensors are similar to load cells in that they have a simple mechanical element and a sensor (typically strain gauges):



The gauges are on two perpendicular 45° helixes with gauge 3 and 4 on the opposite side from 1 and 2.

$$\tau_{xy} = \frac{TD}{2J} = \frac{16T}{\pi D^3}$$

- D is diameter of the shaft

- J is the polar moment of inertia of the circular cross section.

For a shaft subject to pure torsion the normal stresses are:

$$\sigma_x = \sigma_y = \sigma_z = 0$$

We can therefore show that the stresses at the strain gauges are:

$$\sigma_1 = -\sigma_2 = \tau_{xy} = \frac{16T}{\pi D^3}$$

Hooke's law states that:

$$\varepsilon_1 = \frac{16T}{\pi D^3} \left(\frac{1+\nu}{E} \right) \quad \text{AND} \quad \varepsilon_2 = -\frac{16T}{\pi D^3} \left(\frac{1+\nu}{E} \right)$$

Mech 455 Lecture Notes

Combining with the sensitivity of the strain gauges we get:

$$\frac{\Delta R_1}{R_1} = \frac{\Delta R_2}{R_2} = \frac{\Delta R_3}{R_3} = -\frac{\Delta R_4}{R_4} = S_s \frac{16T}{\pi D^3} \left(\frac{1+\nu}{E} \right)$$

Using a full bridge for the gauges, we get:

$$E_v = S_s \frac{16T}{\pi D^3} \left(\frac{1+\nu}{E} \right) E_v$$

OR, solving for the torque:

$$T = \frac{\pi D^3 E}{16(1+\nu) S_s E_v} E_v = C E_v$$

The sensitivity is again 1/T and depends on the diameter, the material and the gauge factor.

Appendix I. 16 Bit ADC Resolution MATLAB Code

```
%-----  
--  
%Discretization Error and Resolution  
%-----  
--  
%Instrumentation amp Gain  
Rg = 330  
gain = 5+200000/Rg;  
  
% gain = 500.0  
% resistorYouNeed = 200000.0/(gain-5.0)  
  
% %Long Thin Beam  
% E = 68.9E9;  
% b = 0.02546;  
% h = 0.00642;  
% I = b*h^3/12;  
% x = 0.263;  
% SG = 2.1;  
% Vi = 5.0;  
%  
% %Resolution  
%  
% n = 10;%n bit ADC converter  
% ResolutionInVolts = (5-0)/(2^n-1);%V/step  
% ResolutionInSteps = 1/ResolutionInVolts;  
%  
% %Discretization Error  
%  
% dError = 1/2*ResolutionInVolts;%V/step  
%  
% %Smallest Error in terms of Force  
% Vo = ResolutionInVolts/gain;  
% dErrorForce = 4*Vo*E*I/(SG*x*h*Vi);%[N]  
  
%Rod Specs  
E = 27.5E6;  
nu = 0.31;  
D = 1.0;  
SG = 1.2; %1.3+-0.3  
Vi = 1.65; %half of 3.3  
  
%Instrumentation amp Gain  
sysVolt = 3.3; %Source voltage for all components  
Rg = 205;  
gain = 5+200000/Rg;  
%gain = gain*16.0 %ads1015 gain  
  
% gain = 500.0  
% resistorYouNeed = 200000.0/(gain-5.0)
```

```

%Resolution

n = 16;%n bit ADC converter
ResolutionInVolts = (Vi)/(2^n-1);%V/step
ResolutionInSteps = 1/ResolutionInVolts;

%Discretization Error

dError = 1/2*ResolutionInVolts;%V/step

%Smallest Error in terms of Force
Vo = ResolutionInVolts/gain;
dErrorTorque = Vo * pi * D^3 / SG / 8.0 * (E / (1.0 + nu)) / Vi /
12.0;%[ft*lbs]
dErrorTorquein = dErrorTorque*12.0;

Torque = 7920.0;

VoMax = Torque/pi/D^3*SG*8.0/(E/(1.0+nu))*Vi; %inch lbs

gainrequired = Vi/VoMax;

```

Code Output:

D	1
dError	1.2589e-05
dErrorTorque	0.0089
dErrorTorquein	0.1069
E	27500000
gain	980.6098
gainrequired	867.3909
n	16
nu	0.3100
ResolutionInSteps	3.9718e+04
ResolutionInVolts	2.5177e-05
Rg	205
SG	1.2000
sysVolt	3.3000
Torque	7920
Vi	1.6500
Vo	2.5675e-08
VoMax	0.0019

References

- [1] S. Iyer, "Power Transmission Design: BCIT Baja SAE 2015/17," Burnaby, BC, 2016.
- [2] A. Sekhon, A. Marciniak and A. Akhavan-Rezaei, "SAE Baja Drivetrain," Burnaby, BC, 2015.
- [3] S. Bitar, J. Probst and I. Garshelis, "Development of a Magnetoelastic Torque Sensor for Formula 1 and CHAMP Car Racing Applications," SAE International, Warrendale, PA, 2000.
- [4] Formula One World Championship Limited, "Rules & Regulations: Tyres and Wheels," Formula One World Championship Limited, 2019. [Online]. Available: <https://www.formula1.com/en/championship/inside-f1/rules-regs/tyres-and-wheels.html>. [Accessed 2019].
- [5] Honeywell, "Ways to Measure the Force Acting on a Rotating Shaft," Honeywell, 2019. [Online]. Available: https://measurementsensors.honeywell.com/techresources/appnotes/Pages/Ways_to_Measure_the_Force_Acting_on_a_Rotating_Shaft.aspx.
- [6] R. Hibbeler, Mechanics of Materials, Eighth Edition, Upper Saddle River: Pearson, 2011.
- [7] Vishay Precision Group, "Fatigue Characteristics of Micro-Measurements Strain Gages," Vishay Precision Group, 2010.
- [8] R. Case and J. Moreno, "Lab 4: Strain Measurement (MECH 8135)," The British Columbia Institute of Technology, Burnaby, BC, 2019.
- [9] Apec Braking, "ABS Sensors," Apec Braking, [Online]. Available: <https://www.apecbraking.co.uk/Resources/TechniciansGuides/ABS-Sensors.aspx>. [Accessed 2019].
- [10] J. Esfandyari, R. De Nuccio and G. Xu, "Introduction to MEMS gyroscopes," STMicroelectronics, November 2010. [Online]. Available: <https://electroiq.com/2010/11/introduction-to-mems-gyroscopes/>. [Accessed 2910].
- [11] Purdue Baja, "Baja SAE Forums," SAE International, May 2012. [Online]. Available: http://forums.bajasae.net/forum/engine-dyno-and-hp-variance_topic701.html. [Accessed 2019].

- [12] University of Arkansas Baja, "Baja SAE Forums," SAE International, March 2012. [Online]. Available: http://forums.bajasae.net/forum/briggs-dynos-if-your-interested_topic946.html. [Accessed 2019].
- [13] Texas Instruments, *ADS111x Ultra-Small, Low-Power, I2C-Compatible 860-SPS, 16-bit ADCs With Internal Reference, Oscillator, and Programmable Comparator*, Dallas, TX: Texas Instruments, 2018.
- [14] InvenSense, *World's First Wide-Range 6-Axis MotionTracking Device for High Speed Applications*, Invensense, 2016.