

AN ABSTRACT OF THE THESIS OF

Christian N. Dizon for the degree of Master of Science in Electrical and Computer Engineering presented on August 30, 2019.

Title: Mini-Data Acquisition System for Small-Scale Testing of Wave Energy Converters

Abstract approved: _____
Ted K. A. Brekken

Data acquisition (DAQ) systems are a necessary component for products whose outcomes depend on collected information throughout its operation. Current systems are directed towards users utilizing custom hardware, software, and requiring its user to have high-level skills in coding and system setup; but with the increasing interest in renewable energy, small corporations, university research departments, and independent companies are undergoing their own testing, requiring DAQ systems on a smaller scale.

The purpose of this project was to produce an easy-to-use EtherCAT server DAQ system alternative, aimed towards new developers conducting small scale testing in the wave energy field. This report serves to outline the product made in the Mini-Data Acquisition (MiniDAQ) project funded by Sandia National Laboratories.

Along with various sensors such as encoders, analog-to-digital converters, and an inertial measurement unit, the MiniDAQ contained an Arduino Mega 2560 Rev3 as the development board, equipped with AB&T's EasyCAT shield as a means to operate in real-time through the protocol EtherCAT when combined with real-time systems Speedgoat and Mathworks Simulink. This setup met the main requirements of real-time operation, 100 Hz sampling frequency, differential inputs/outputs, and two communication interfaces.

©Copyright by Christian N. Dizon
August 30, 2019
All Rights Reserved

Mini-Data Acquisition System for Small-Scale Testing of Wave Energy Converters
by
Christian N. Dizon

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented August 30, 2019
Commencement June 2020

Master of Science thesis of Christian N. Dizon presented on August 30, 2019

APPROVED:

Major Professor, representing Electrical and Computer Engineering

Head of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Christian N. Dizon, Author

ACKNOWLEDGEMENTS

I express sincere appreciation to those who helped the making of this thesis. This includes advisors, family, and friends. Special recognitions to:

Advisor Ted K A Brekken. Without his undertaking, patience and determination in finding a suitable project, all the while answering a curious student's constant questions, I would not have developed and confirmed my further passion I now find with wave power as a renewable energy.

Father Ed Dizon for providing support and reminding me of my enthusiasm for research throughout the years.

Friend Ali Haider who never failed to fulfill my desire to fully understand the smallest and largest of concepts.

Sandia National Laboratories for making this project possible.

TABLE OF CONTENTS

	<u>Page</u>
1 Introduction	1
2 Block Diagram	3
3 Proposed System	6
4 Requirements	11
5 Component Description and Testing	15
5.1 Arduino Mega 2560 Rev3 - Development Board.....	16
5.2 EasyCAT Shield - EtherCAT communication	17
5.3 DC682A - Analog to Digital Converter	18
5.4 DC2025A - Digital to Analog Converter	21
5.5 Load Cell Amplifier HX711 - Wheatstone Bridge.....	23
5.6 IMU & NS-RS232 communication	25
5.7 ADS1115 4-20 mA Current Loop Receiver.....	32
5.8 CUI AMT23 Absolute Encoder.....	34
5.9 MAX1484 - SSI protocol.....	37
5.10 Anti-Aliasing RC filter - Noise Testing.....	39
6 Software	41
6.1 Arduino Integrated Development Environment.....	41
6.2 EasyCAT Configurator.....	44
6.3 TwinCAT	45
6.4 Speedgoat/Simulink.....	47
7 System Testing	51
7.1 Excitation signal options.....	52
7.2 Frequency Response of Speedgoat	52
7.3 Frequency Response of ADC	56
7.4 Explanation of Phase Plot.....	58
7.5 Comparison to Chirp Signal	59
7.5.1 Magnitude.....	61
7.5.2 Phase Change.....	62
7.5.3 Sample Delay.....	63
8 Conclusion	65

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Block Diagram.....	5
2. Exploded View of Sensor System.....	7
3. Power Flow to System	8
4. Speedgoat Setup.....	9
5. Screenshot of SLRT model file.....	10
6. EasyCAT passing EtherCAT CTT.....	14
7. Arduino Mega 2560 Rev3.....	16
8. AB&T's EasyCAT shield	18
9. ADC Setup.....	19
10. Output of ADC vs Input of Function Generator through Simulink Data Inspector.....	20
11. DAC Setup	22
12. $\pm 10V$ Output of DAC, inputted by 0-65535 Integer Sawtooth Range.....	23
13. HX711 Load Cell Setup.....	24
14. Load Cell Output on Arduino Serial Monitor.....	25
15. IMU Setup.....	26
16. MT Manager Output Configuration Preferences.....	27
17. MT Manager Output Results.....	28
18. MT Message of Standard Length.....	28

LIST OF FIGURES (Continued)

<u>Figure</u>	<u>Page</u>
19. MT Message - Setting the Output Configuration; highlighted are the listed settings.....	29
20. Arduino Serial Monitor of IMU output; boxed in red is the float value of the roll variable	30
21. Arduino Serial Monitor of IMU output; boxed in red is the float value of the pitch variable.....	31
22. Circuit for Simulated Current as reading to ADS1115	32
23. ADS1115 4-20 mA Current Loop Receiver Setup... ..	33
24. Arduino Serial Monitor displaying Current Output in Integer and Float Form....	34
25. CUI AMT23 Setup.....	35
26. Integer Values of Encoder; begins from 16384 and decreases to 0	36
27. SSI Testing Setup.....	38
28. Anti-Aliasing RC filter for Differential Input.....	39
29. Anti-Aliasing RC Filter Setup... ..	40
30. Flow Chart of MiniDAQ Code	43
31. DAQ and Load Cell Section of MiniDAQ code with <i>BufferOut</i> and <i>BufferIn</i> functions boxed.....	44
32. EasyCAT Configurator Application with Custom Inputs and Outputs.....	45
33. TwinCAT software showing inputs in Task (green box) and those found on the EasyCAT shield (yellow box). Boxed in red is the 'Linked to' which connects the Task's variables to the current input and output variable on the EasyCAT shield.....	46
34. Network Settings in Simulink Real Time Explorer	48
35. Various Defined Inputs seen in the <i>EtherCAT PDO Receive</i> Block.....	49
36. Overall SLRT model.....	49
37. Sensor Outputs through Speedgoat Target Scopes... ..	50

LIST OF FIGURES (Continued)

<u>Figure</u>	<u>Page</u>
38. IMU Output through Speedgoat Target Scope	51
39. Generated PRBS Signal in Matlab.....	53
40. Speedgoat Frequency Response Setup.....	54
41. Matlab System Identification Toolbox.....	54
42. Speedgoat Frequency Response Test Magnitude and Phase Results.....	55
43. Speedgoat Frequency Response Test Step Response	55
44. ADC Frequency Response Setup – PRBS	56
45. ADC Frequency Response Test Magnitude and Phase Results... ..	57
46. ADC Frequency Response Test Step Response.....	57
47. Experimental vs. Theoretical Phase Response.....	59
48. Phase Plot of Figure 48 Wrapped to 180° and -180° to Determine Phase Change Points in Chirp Signal.....	60
49. ADC Frequency Response Setup – Chirp.....	60
50. Full Chirp Signal from 0 to 50 Hz	61
51. Zoomed in Chirp Signal from 0 to 3 Hz	61
52. 0° Phase Shift.....	62
53. 180° Phase Shift.....	63
54. Step response indicating 4 sample delay.....	64
55. 4 sample delay (0.04 s) between input and output.....	64

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Table of values showing input frequency and signal voltage, outputted signal voltage noiseless vs noise conditions.....	40

1 Introduction

Data acquisition systems (DAQ) are vital for accurate operation of any physical system where events in the outside world affect a product's outcome. These systems can be used from simple data logging, to aid in control systems for autonomous operation. At minimum, they contain sensors, a processing unit to condition the outputted signal, and a computer that interprets this signal to determine the events to follow. Without a DAQ, there is no way to gain information about the environment that a product is working in, also creating no events to act upon. The autonomous nature is since lost as operators would continuously need to monitor the state of a running product to make further decisions otherwise handled by a computer. It's clear that DAQs offer the highest advantage by removing a large sum of human error, slow response times, and decrease in accuracy. Additionally, they can be configured to operate in real-time, a requirement that is becoming more common in products and projects where live data is crucial for successful performance. DAQs can be found ranging from applications in a business office, self-running machines in the automotive industry, or renewable energies, specifically in this report, a wave energy converter (WEC).

WECs convert wave power into electrical power using a power take-off (PTO) device. This mechanism takes absorbed energy and transforms it into electricity. Forms of this include air turbines where the change in water level creates a direct change in air pressure in a chamber, pushing air back and forth to move a turbine connected to a generator. This would be the case of an Oscillating Water Column WEC. When considering the WEC developed by Pelamis, the waves motion creates changing hydraulic pressure in the moving joints, and when connected to a hydraulic motor, creates electricity. However, simply absorbing the wave's energy passively does not get closer to the goal of producing maximum power. The WEC itself must be in resonance with the changing environment, and while this is hard to achieve through physical features such as the size and weight of the converter, the PTO can be actively controlled to adjust how the WEC interacts with the waves [40]. To achieve this, the PTO needs to be governed by constant observation of the incoming wave's characteristics

such as height, period, direction, celerity, or pressure. Not only do these variables need to be logged, they must be available whenever requested, leading to a deterministic control scheme that DAQs can provide. Such a DAQ present on a WEC can now observe this data and provide a path towards the end goal.

Current systems available on the market supporting real-time include CompactRIO with LabVIEW by National Instruments, Beckhoff's ELx series using TwinCAT, and OPAL-RT OP series with RT-LAB. While these systems serve their purpose, they are costly, large in size, and directed towards complex practices that require tighter constraints. Although the DAQ could be present onshore or on a separate platform to remove the size restriction, the signals are at risk of being altered from the WEC's emitted electromagnetic interference (EMI). This creates a challenge for those interested in model-scale devices, leaving a need for a simple to use, low-cost, light weight, 'on-board' DAQ. Such a device would be useful for universities, independent researchers, or small companies that don't have the need for a high-level machine.

Today, such a DAQ doesn't exist. Two close examples include when Columbia Power Technologies developed a WEC tested at Oregon State University's O.H. Hinsdale Wave Research Laboratory (HWRL) that used three separate systems to acquire data: one provided by HWRL, PhaseSpace (an optical motion tracking system), and a custom built embedded controller containing various sensors [1]. Another is when Sandia National Laboratories (SNL) developed a design for a small DAQ in the Advanced WEC Dynamics and Controls Project. They too combined 5 collection systems to obtain all necessary data [2]. To reduce the amount of combinations of systems, our project aspires to provide a one-system DAQ that can be used in any situation where its inputs and outputs are relevant. This also means that, while the DAQ is directed towards WEC use, it can be used as a general-purpose tool for any application benefiting from its working design.

The following report describes the process involved in designing, producing, and testing the "MiniDAQ" project funded by SNL to be used on their WEC - the Floating Oscillating Surge WEC (FOSWEC). Commercial-off-the-shelf (COTS) hardware was used heavily to provide consumers readily

available assistance throughout the web and to minimize customization of the MiniDAQ. The final product uses the EtherCAT protocol to provide real-time communication with real-time systems Speedgoat and Simulink Real-Time (SLRT), contains various sensors found commonly on WECs, and satisfies the requirements by SNL as noted in section 5. Sections 6 and 7 validate the described hardware and software abilities in meeting these demands and section 8 describes the frequency response and time lag of the system.

2 Block Diagram

Figure 1 is a block diagram of the system to describe how the individual hardware and software interact with each other. It is divided into three boxes: grey representing the real-time system (SLRT and Speedgoat: control, configuration, and data records), light blue representing the entirety of the MiniDAQ (development board, EtherCAT transmission, sensors), and green representing the WEC (sensors, instruments, forces). The purpose of each box and flow of data is detailed next.

The job of the Arduino Mega is to configure the MiniDAQ sensors to the users desired settings and receive their outputted data or transmit data. To put this into a real-world perspective using notations in Figure 1, explained is how the MiniDAQ and WEC box work together: the 'sensor' that is providing the analog signal to the DC682A ADC could be a resistive wave gauge on the WEC, and the 'Interface Instrument' that is providing force readings to the HX711 Load Cell could be the stress on a mooring line attached to the WEC; the inertial measurement unit (IMU) does not contain an equivalent version as an IMU would be what is present on the WEC; for data being transmitted, the analog signal that is sent from the DC2025A DAC could be used to control the position of an actuator. Once data is ready to be received/sent, they are put into variables and sent to the EasyCAT shield to be prepared for transmission via EtherCAT to/from the Speedgoat which has been setup by SLRT through the host machine for, specifically in the MiniDAQ project, simply keeping data records. For other purposes, the Speedgoat has a higher computing power than the Mega alone, and therefore could be used to make complex control decisions

for the PTO to achieve maximum efficiency and power output of a WEC as described in the introduction. This is how the Real-Time System will be implemented on SNL's FOSWEC, to make control decisions such as directing how motors/generators operate.

3 Proposed System

DAQs are divided into three configurations which directly affect how a system is designed and will operate: distributed asynchronous, distributed synchronous, and centralized. The proposed system utilizes distributed synchronous as the other two could not satisfy both synchronization and close proximity of sensors while limiting the amount of wires and cables.

Hardware components consist of an Arduino Mega 2560 Rev3 chosen as the development board, equipped with AB&T's EasyCAT shield as a means to communicate through the EtherCAT protocol. EtherCAT was chosen because of the protocols ability to communicate in hard real-time, exceeding the performance of Sercos III, PROFINET, and Powerlink [1]. Sensors attached include two development boards from Linear Technology: DC682A, an analog-to-digital converter (ADC), and DC2025A, a digital-to-analog converter (DAC). Sparkfun's Load Cell Amplifier HX711, CUI's AMT23 absolute encoder, ADS1115 4-20 mA current loop receiver, and a passive RC filter to reduce EMI are also included. Additionally, an IMU from Xsens provided Euler angles and x, y, z accelerations. The communication protocols of specific sensors made it necessary to use RS-232 and SSI to TTL converters. Figure 2 is an exploded image of the sensors included in the MiniDAQ.

The power flow to the system starts from a wall outlet delivering 120 V AC connected to a DC power source, converting the AC to DC and bringing it down to 24 V DC. Using a DC-DC converter, this is then stepped down to 12 V DC which is supplied to necessary components, however, most required 5 V DC acquired by using a voltage regulator. In the final design, this will eventually be given by a cable fed to the FOSWEC. Figure 3 details the power flow from outlet to components.

For real-time operation of the systems inputs and outputs, Speedgoat and Simulink were used. Figure 4 displays the Speedgoat communication setup and Figure 5 is a screenshot of the model file in SLRT showing 'blocks' that represent data being received/transmitted to/from the EasyCAT and passed onto the Speedgoat. The interaction between these two will be explained in detail later.

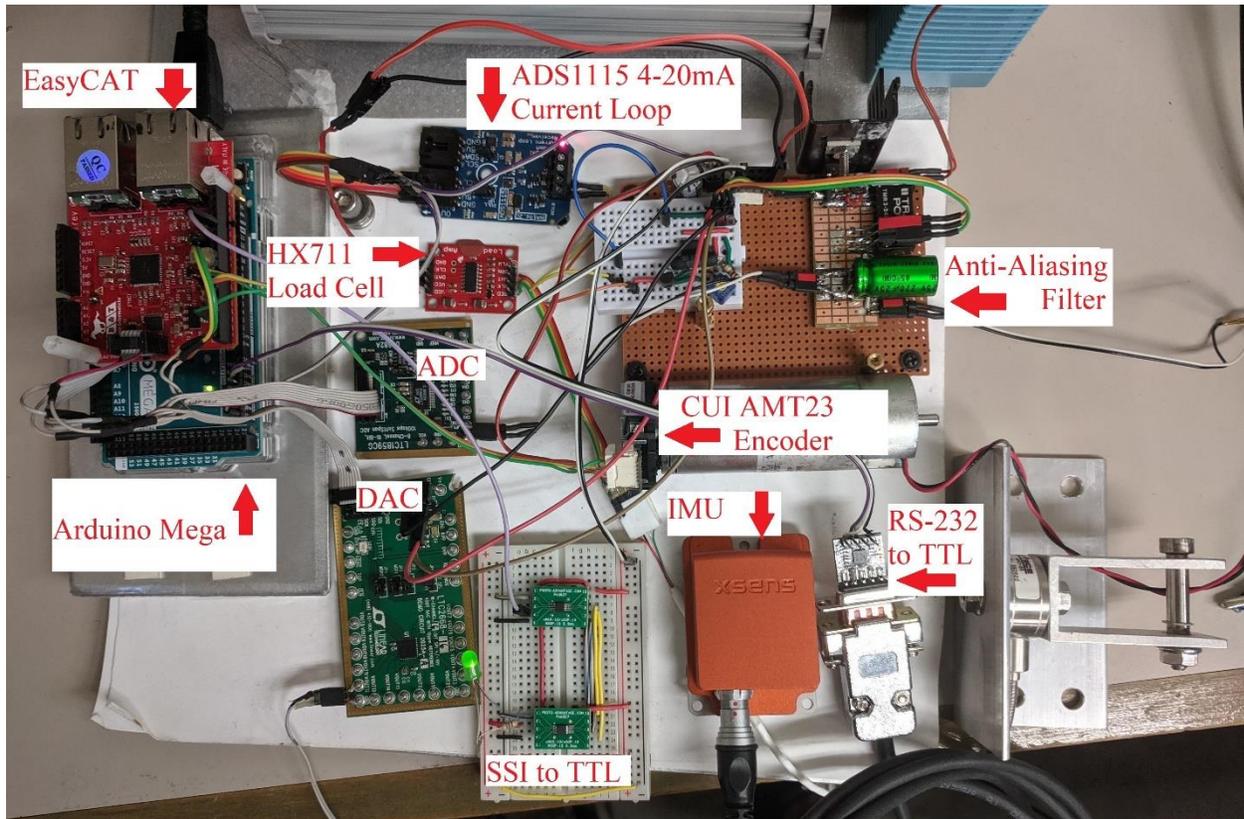


Figure 2: Exploded View of Sensor System

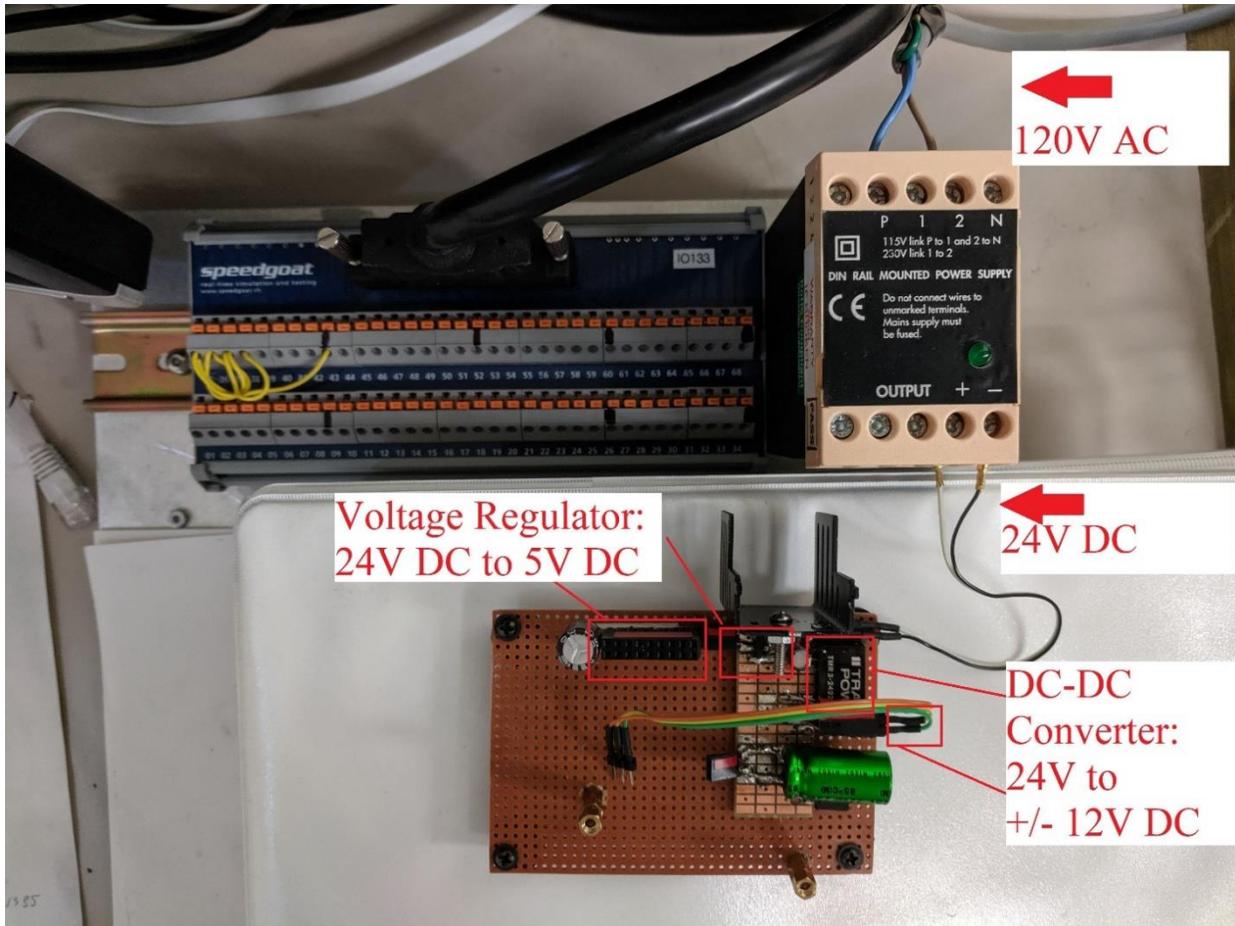


Figure 3: Power Flow to System

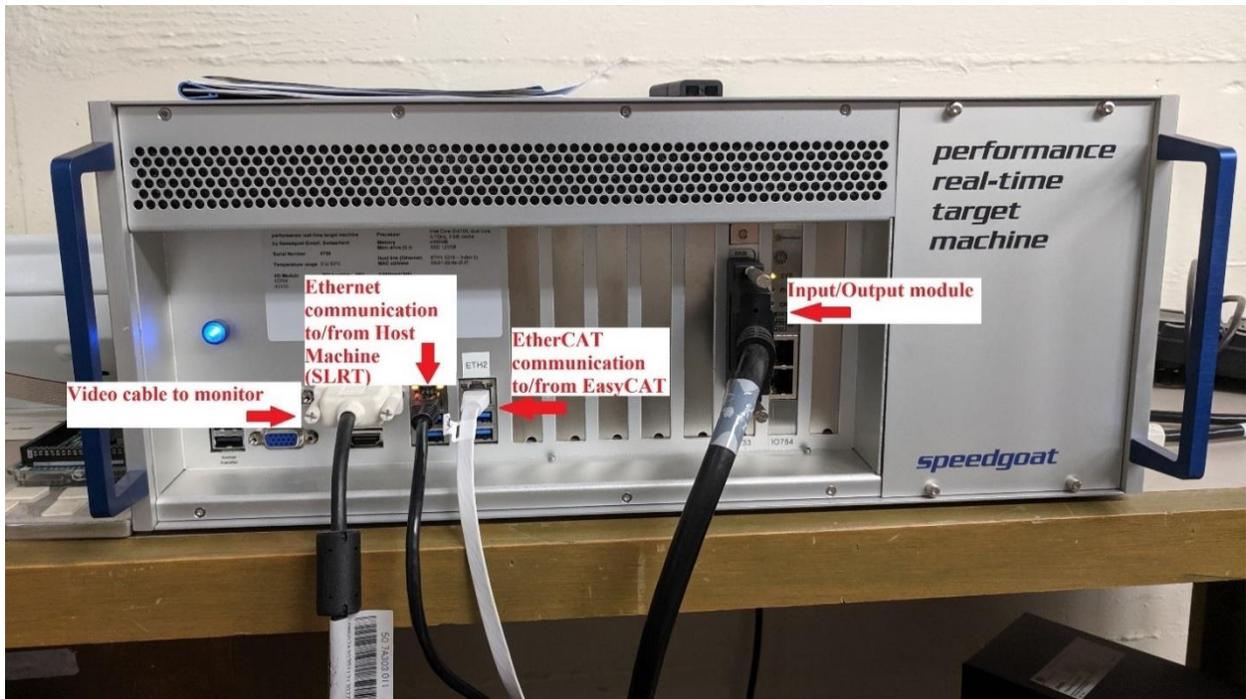
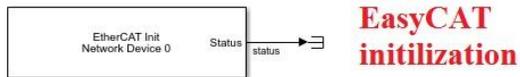
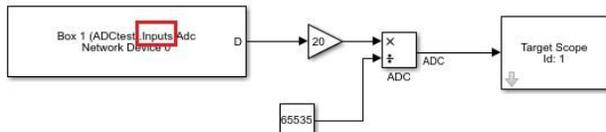


Figure 4: Speedgoat Setup

SETUP



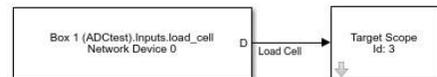
ADC Data Received Block



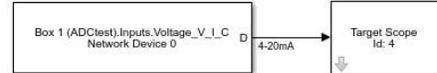
DAC Data Transmitted Block



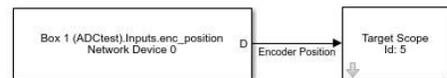
Load cell



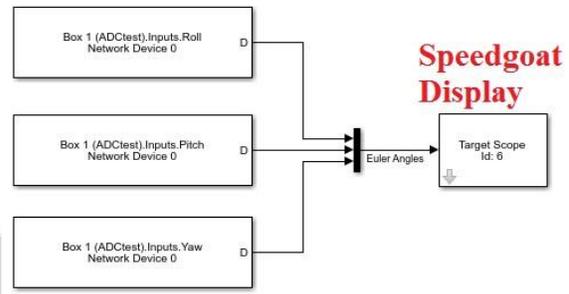
Voltage from ADS1115 4-20mA Current Loop Receiver



Encoder Position



Euler Angles



Speedgoat Display

Accelerations

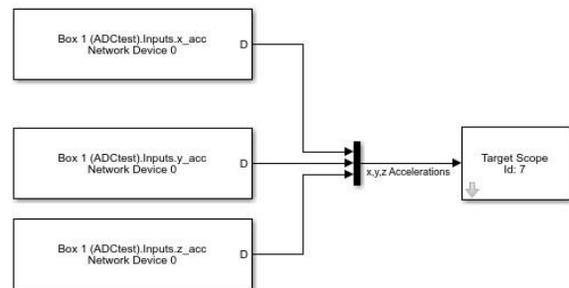


Figure 5: Screenshot of SLRT Model File

4 Requirements

Outlined in the project scope were the key and system requirements. Listed is a brief explanation of each requirement and why our system satisfies each.

1. Cost <\$5,000

To minimize the obstacle to small-scale testing, the cost should be low.

Arduino Mega 2560 (\$38.50[4])
 +EasyCAT(\$56.72[5])
 +DC682A(\$50.00[6])
 +DC2025A(\$100.00[7])
 +HX711(\$9.95[8])
 +NS-RS232(\$9.95[9])
 +MAX1484(2 x \$2.66[10])
 +ADS1115(\$45.95[11])
 +AMT23(\$50.96[12])
 +IMU MTI-20(\$1355.00[13])
 +Resistors(2 x \$0.19[14])
 +Capacitor(\$11.56[15])

= \$1734.29

2. 100 Hz Sampling Frequency

100 Hz was the maximum requested sampling frequency by SNL.

The current system when operating with all sensors yields a sampling time of just 10 milliseconds. With the requested final setup that uses one ADC, one 4-20 mA current sensor, and the Euler angles from the IMU, the system operates at 6-7 milliseconds.

3. 16 bit Resolution

16 bits was the requested resolution by SNL.

Both DC682A ADC and DC2025A DAC have output/inputs of 16 bits [16],[17].

4. Weight interference

To avoid interfering with the WEC's dynamics, the DAQ should be light compared to the total weight of the converter.

The total weight of our system is roughly 600 grams. This is negligible compared to the range of tens of pounds to hundreds of tons a small or large scale WEC weighs [19][20],[21],[22].

5. Volume/footprint

Space is limited on a WEC due to e.g. its physical design, motors, and cables, therefore the DAQ should be such a size so placement is the least restrictive.

The requested system has been fitted to a 6"x5"x4" box which is negligible compared to the range of up to tens of feet to meters in any dimension of a small or large scale WEC [18],[19],[20],[21],[22],[23].

6. Power/Data Cabling Minimized

The dynamics of a small-scale WEC can be influenced by the cables delivered from shore. These should be minimized in amount and size.

The proposed system currently runs on $\pm 12V$ DC. This will be supplied through a power cable already delivered to the WEC and the data is transmitted via a single standard Ethernet Cable. The addition from our system will not have any adverse effects.

7.EMI

EMI exists from the motors and electrical components present on a WEC. Limiting this and incorporating mitigation techniques will ensure signals are not disturbed.

Length of cable is dependent on location of WEC; keeping this short as possible is desired to minimize EMI. Use of differential inputs/outputs and passive RC filter also minimizes effects.

8.Deterministic Operation

WEC's utilize closed-loop control which calls for deterministic operation.

The system utilizes Speedgoat hardware and Simulink software that both support real-time operation [24], [25].

9.Multi-Platform Support

Considering the need for such a general DAQ system, it would be useful for it to support multiple real-time systems as to not constrain itself in the market.

Speedgoat was the only real-time system tested on. Justifications were made based on hardware availability and knowledge of software.

10.Scalability

To reduce cost and risk when moving to large-scale testing, the DAQ should be easily scalable for the change in power, robustness, and quality of measurement demands.

The system can easily be scalable to the designers intent. For example, on the Mega, one serial port and nine digital pins are used. This leaves a majority of the board for additions [4].

11. Support for Existing Hardware

Enabling COTS allows a larger audience to use the DAQ, provides available support, and offers online resources such as libraries and projects.

Our entire system is composed of COTS including those from Arduino, Spark- fun, and Linear Technology.

12. Passes EtherCAT Conformance Test Tool (CTT) evaluation

Ensures EtherCAT devices are operational.

AB&T has proven their devices are factory tested and pass the EtherCAT CTT with 0 errors and warnings as seen below.

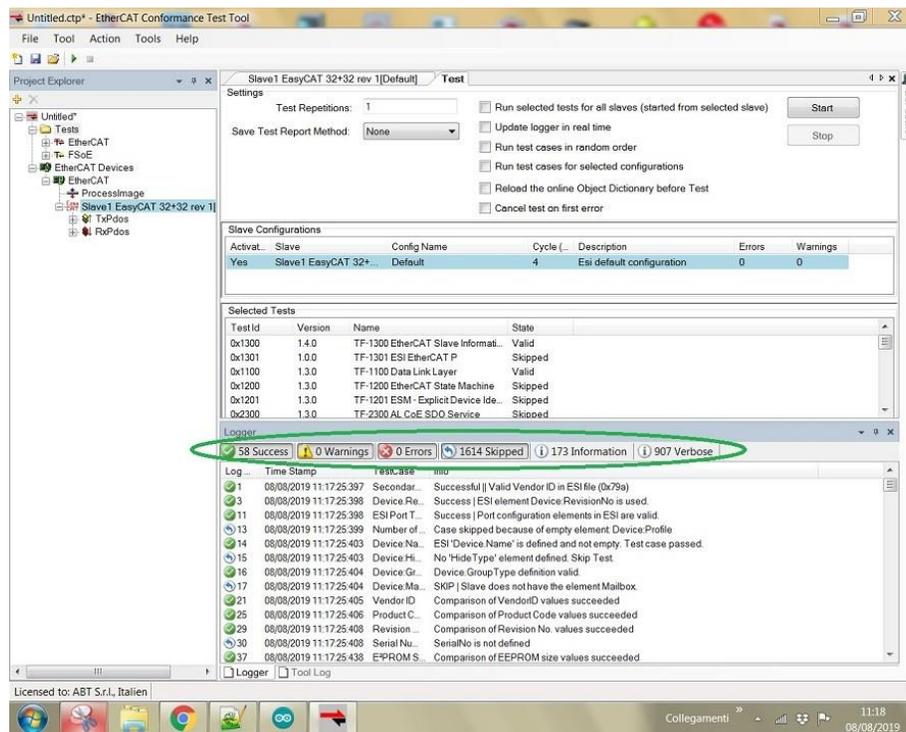


Figure 6: EasyCAT passing EtherCAT CTT

13. Inputs

Each item listed was a requirement by SNL.

(a) ± 10 V

Input to the entire system is ± 12 V DC which can be used to supply all components. 12 V is acceptable as the main condition was at minimum 10 V and to be differential.

(b) 4-20 mA

The ADS1115 4-20 mA current loop receiver reads these currents [11].

(c) Bridge

A load cell amplifier was provided that uses a Wheatstone Bridge [8].

(d) Synchronous Serial Interface (SSI)

CUI's AMT23 absolute encoder communicates through SSI [26]. SSI was also tested through the MAX1484 chips by converting TTL to SSI and back to TTL to control a LED [27].

(e) Serial Peripheral Interface (SPI)

Successful use of the ADC and DAC demonstrates that the system can communicate using SPI [16] [17].

14. ± 10 V Output

Requested output by SNL.

The DAC outputs ± 10 V DC [17].

5 Component Description and Testing

The purpose of this section is to describe the components present in the light blue MiniDAQ box from Figure 1 (Mega/EasyCAT/sensors). Each subsection will focus on an individual component, explaining what it is generally used for and its importance, along with figures detailing its setup and output. All components were chosen keeping in mind the target audience, i.e., they had to contain understandable coding and easy setup, all while fulfilling our requirements.

5.1 Arduino Mega 2560 Rev3 - Development Board

Development (microcontroller) boards are the brains to a system. They dictate how components connected to them operate by deciding what to do with their data and when to perform requested tasks. This directly impacts how a system acts, making the choice of the development board crucial. The MiniDAQ project called for a balance between optimal performance and simple use, giving way to a clear decision: the Arduino series. Arduino is an open-source electronics platform that provides hardware and software, while also offering a large helpful community. This platform is praised for its vast uses through a surprisingly easy setup and coding language, making it suitable for all skill levels.

Specifically, the Arduino Mega 2560 Rev3 is a popular microcontroller board among developers. It features 54 digital I/O pins, 16 analog inputs, runs on 5 V, supports SPI, and provides 4 serial communication ports. The Arduino Mega accomplishes the requirement of easy coding and setup, as well as providing multiple serial ports, an issue that became necessary for practical use with the IMU. Without this feature, the user would have to use the same serial port for uploading code to the board. If the IMU were wired here, it would interfere and make the upload fail. Along with these attributes, the Mega is arguably the most highly recognizable and familiar COTS development board. For these reasons it was chosen for the project. Figure 7 depicts this board.



Figure 7: Arduino Mega 2560 Rev3 [4]

5.2 EasyCAT Shield - EtherCAT communication

As mentioned earlier, real-time operation is becoming a common requirement for many upcoming products and projects. The availability of live data paves a path for more efficient operation because decisions can be made immediately instead of minutes later. For some uses, these minutes can have a large impact on the outcome of a situation. From a non-technical view, can you imagine the promise of a sunny day at the beach through a live camera that was actually 15 minutes off, and arriving to a rain cloud approaching? Looking at a situation tailored to WECs, closed-loop control governs a system by monitoring variables and using their value to determine what should occur next. If a large wave were to push a WECs PTO to dangerous limits, the hydraulic pumps, for example, that control this device could be turned off at certain pivot points to minimize the risk of damage. Of course, this will only be so effective if used at the correct timing, an issue that can be resolved through real-time operation.

EtherCAT is a new Ethernet based protocol developed by Beckhoff, a company providing open automation systems. The protocol's main feature is that it can be used for hard and soft real-time requirements. Unfortunately, the EtherCAT systems provided by Beckhoff come at a high price and are hard to implement for our development board, but AB&T's EasyCAT shield makes EtherCAT communication between itself and the Mega *easy* (pun intended). It also provides a simple way in both coding and setup to utilize the protocol in fulfilling our real-time requirement.

With a hardware layout that directly attaches to the Mega, the shield turns the development board into an EtherCAT server using a common EtherNET cable; CAT5 or CAT6 if better data performance and speed is needed. While EtherCAT was designed for uses in an automation industry application, AB&T's solution allows any coder to utilize the EtherCAT protocol without the commitment of high level hardware. They also provide example code to get started with which is easy to understand and modify to the users own self-defined variables fit for their application. As for availability of aid, this COTS customer support has proven beneficial in their quick responsiveness to any questions or issues and success in resolving these matters. The EasyCAT accomplishes the requirements of operating in real-time, passes the CTT, and is easy to use, making it

the perfect candidate for the project. Figure 8 depicts the shield.



Figure 8: AB&T's EasyCAT shield [5]

5.3 DC682A - Analog to Digital Converter

Analog signals are the typical output of a majority of sensors, but to measure and process them, they first need to be digitized because computers understand digital signals. ADCs are necessary for making use of information retrieved such as the change in pressure, temperature, or vibrations, as well as voltage fluctuations. The item to be aware of in ADCs is the bit resolution. This determines how well the original signal is captured. For our requirement, 16 bits means there are 0-65535 (2^{16}) values to represent a signal. The higher the bits, the more accurately represented the signal is. If instead the resolution were 8 bits, there would only be 256 values available. Normally, ADCs are already present within the sensor itself, but if not, an external one is needed.

Linear Technology offers an ADC development board - the DC682A. This helps break out the LTC1589 (an 8 channel, 16 bit, 100 ksps, Softspan ADC[16])

for easy use. To communicate with the converter, SPI protocol is used along with a unique chip select (a digital pin) which essentially enables the device to be used. The LTC1859 requires 8 bits to configure it for an input that can be either 0-5 V, 0-10 V, ± 5 V, or ± 10 V. However, since the output of the converter is 16 bits total, an additional empty 8 bits is sent after the configuration byte to receive the rest of the data sent from the converter. Figure 9 is the development board attached to the Mega/EasyCAT. Figure 10 depicts the 16 bit binary number converted to the equivalent float analog signal versus the actual input signal. As you can see it does an impeccable job. The DC682A not only accomplishes the requirement of differential input voltage and SPI communication, but makes converting digital to analog signals a now simple process.

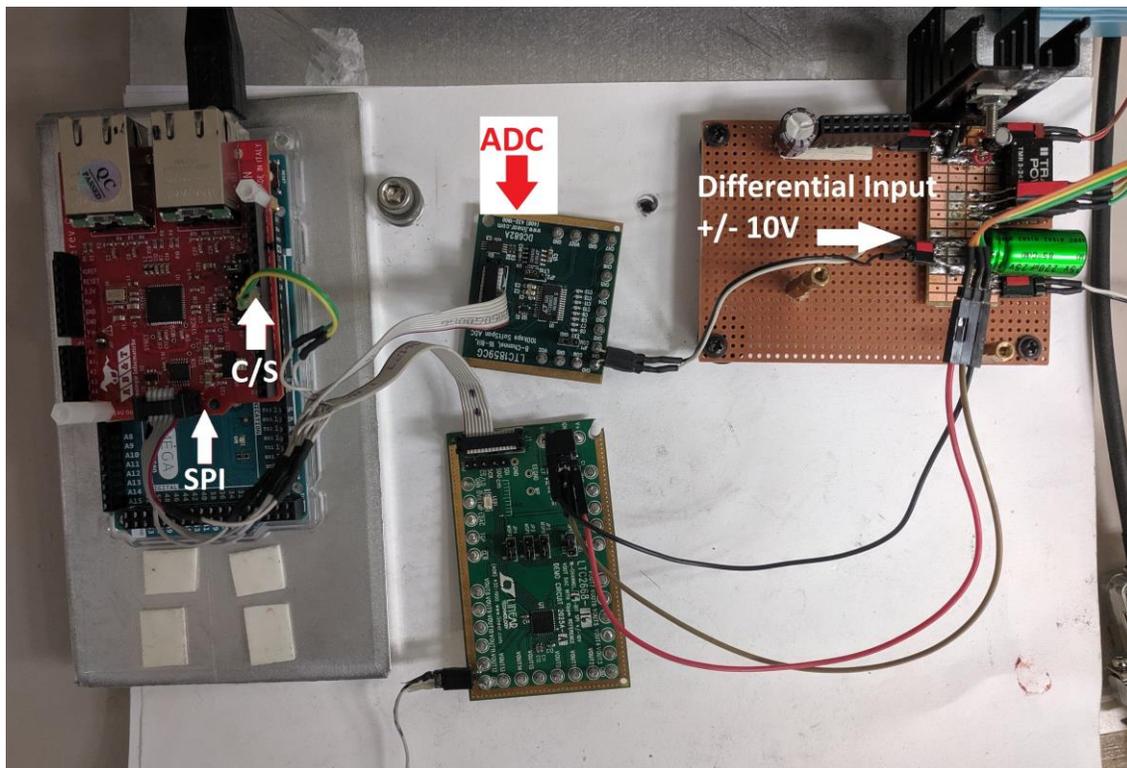


Figure 9: ADC Setup

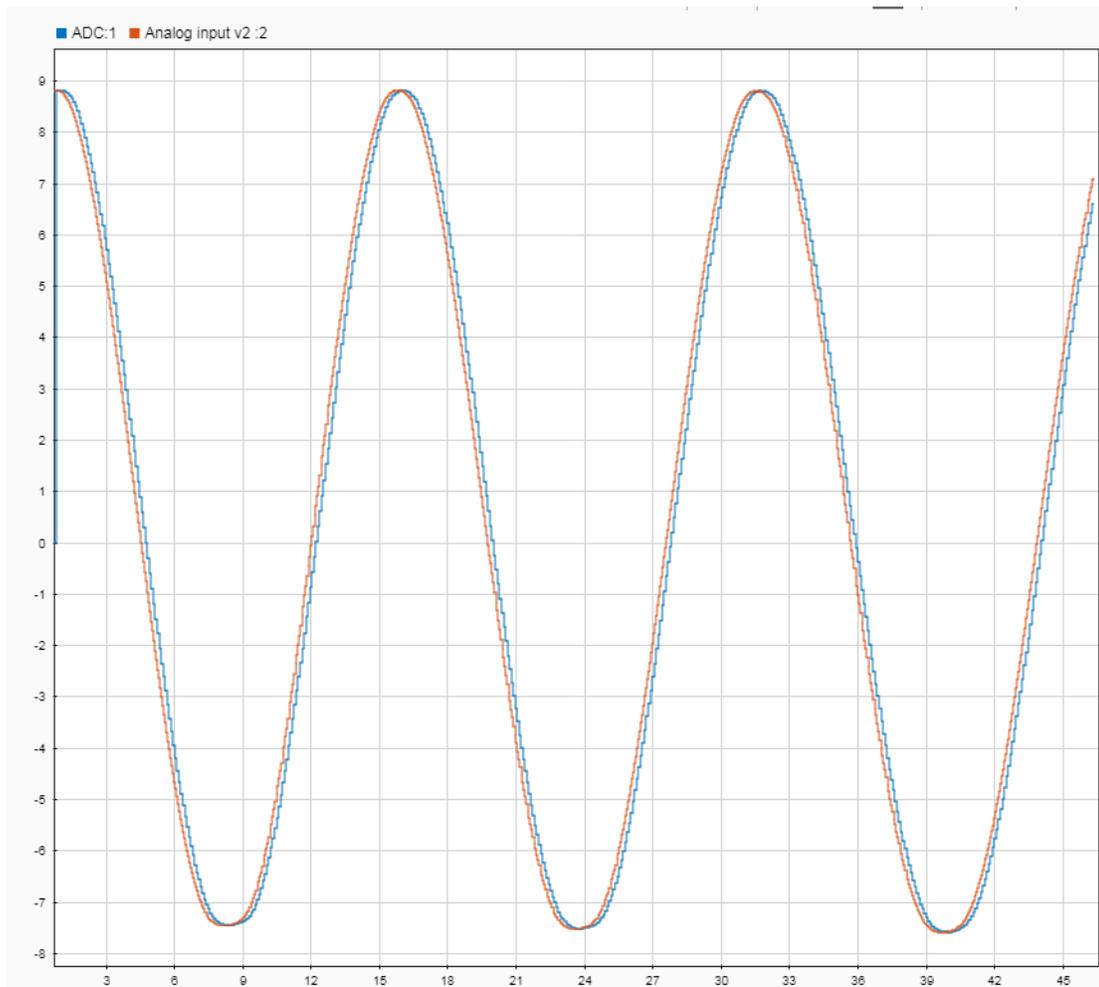


Figure 10: Output of ADC vs Input of Function Generator through Simulink Data Inspector

5.4 DC2025A - Digital to Analog Converter

Alternatively, if there is a need to output an analog signal using a digital signal from a computer, a DAC is needed. These are useful not just for converting digital audio files into an analog form to be heard, but also in a mechanical sense when wanting to control the position of an actuator to open or close a valve. The meaning of resolution is the same as that for an ADC; the higher the bits, the more values a DAC can output yielding a wider range of use.

Linear Technology's DAC equivalent development board is the DC2025A. It contains the LTC2668, a 16 channel, 16 bit, $\pm 10\text{V } V_{out}$ Softspan DAC [17]. The board is specified to run on $\pm 15\text{ V}$ to achieve full Softspan ranges, but for our use $\pm 12\text{ V}$ works. To communicate with the converter, SPI protocol is used along with a unique chip select. The LTC2668 requires 24 bits in total. The first byte is a combination of the command code (write/update all or one channel) and the address (which channel). The next 2 bytes is an unsigned 16 bit integer whose value is up to the user depending on the voltage level they want to send out. Figure 11 is the development board attached to the Mega/EasyCAT. An input sawtooth wave ranging from integer values 0 - 65535 was delivered to the DAC where Figure 12 shows the outputted sawtooth form from -10 V to +10 V. The DC2025A accomplishes the requirement of differential input/output voltage and SPI communication, as well as providing ease-of-use.

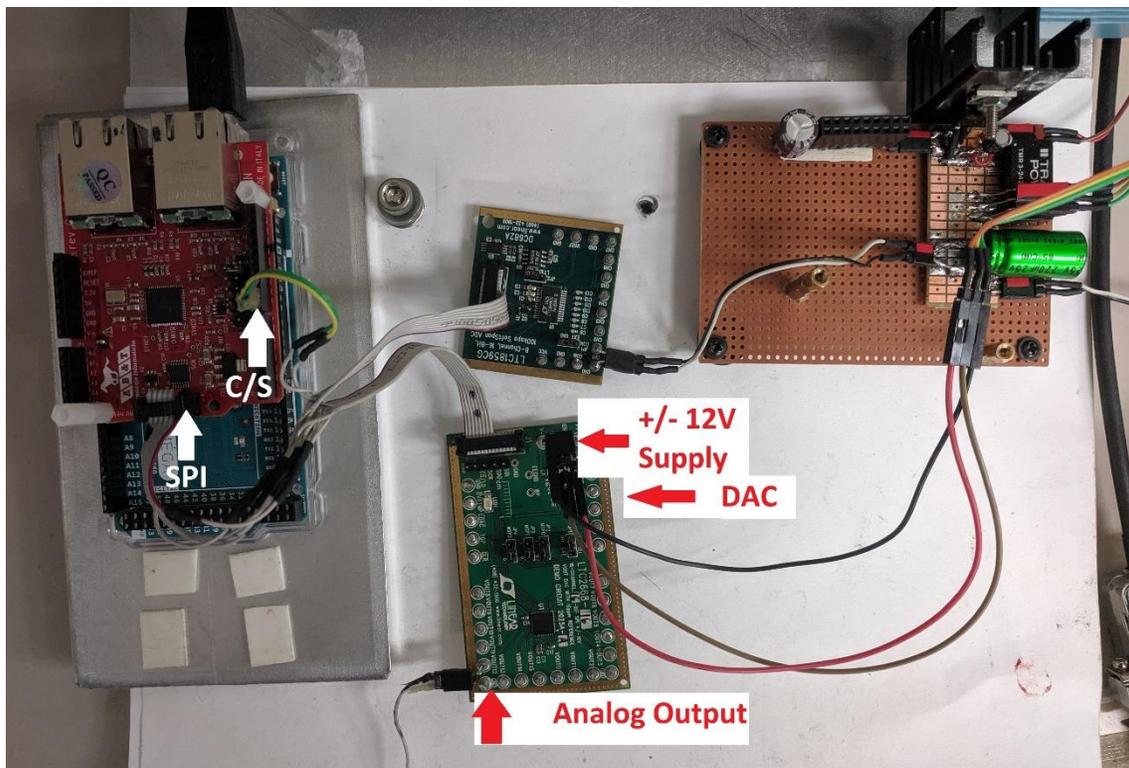


Figure 11: DAC Setup

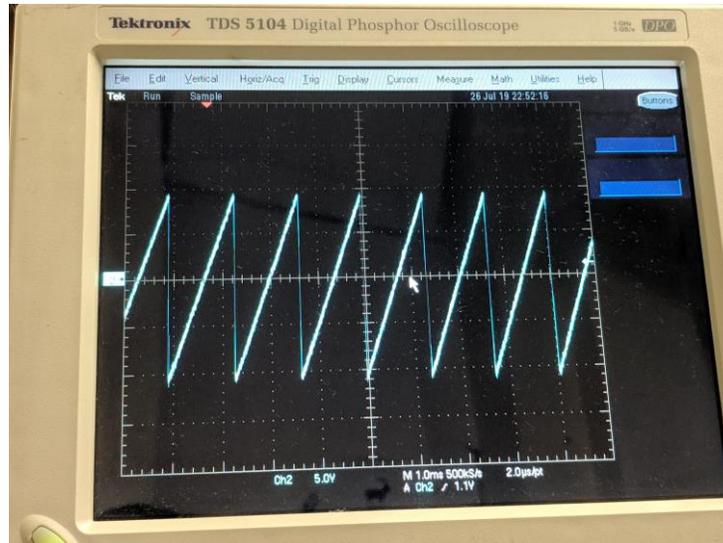


Figure 12: $\pm 10V$ Output of DAC, inputted by 0-65535 Integer Sawtooth Range

5.5 Load Cell Amplifier HX711 - Wheatstone Bridge

Bridge circuits serve many purposes. They can be used for detecting resistances/capacitances/inductances, rectifying signals, reversing the direction of a motor, or to measure a strain gauge. The last item is of interest when measuring force imposed on critical areas, and its configuration is that of the Wheatstone Bridge.

To test the system requirement bridge input, Sparkfun's Load Cell Amplifier HX711 [8] was chosen for its Wheatstone bridge use in measuring strain gauges. A product from Interface Force was used to measure the force in pounds imposed on itself. The output cable of this instrument contained 4 wires corresponding to the 4 strain gauges in the bridge. These directly wired to the same color arrangement on the HX711 - red to red, black to black, white to white, and green to green, with the option of grounding yellow to reduce any small EMI. The board runs on 5 V and outputs using a single data and clock pin. Calibration of the board is required allowing the user to determine the boards' specific offset,

and is factored into future work as *calibration factor*. Once found, all that is needed is one line of code to receive the outputted data. The HX711 was chosen, again, due to easy coding and setup. Figure 13 is the setup of the board with the Mega/EasyCAT and Figure 14 is the output on the Arduino Serial Monitor.

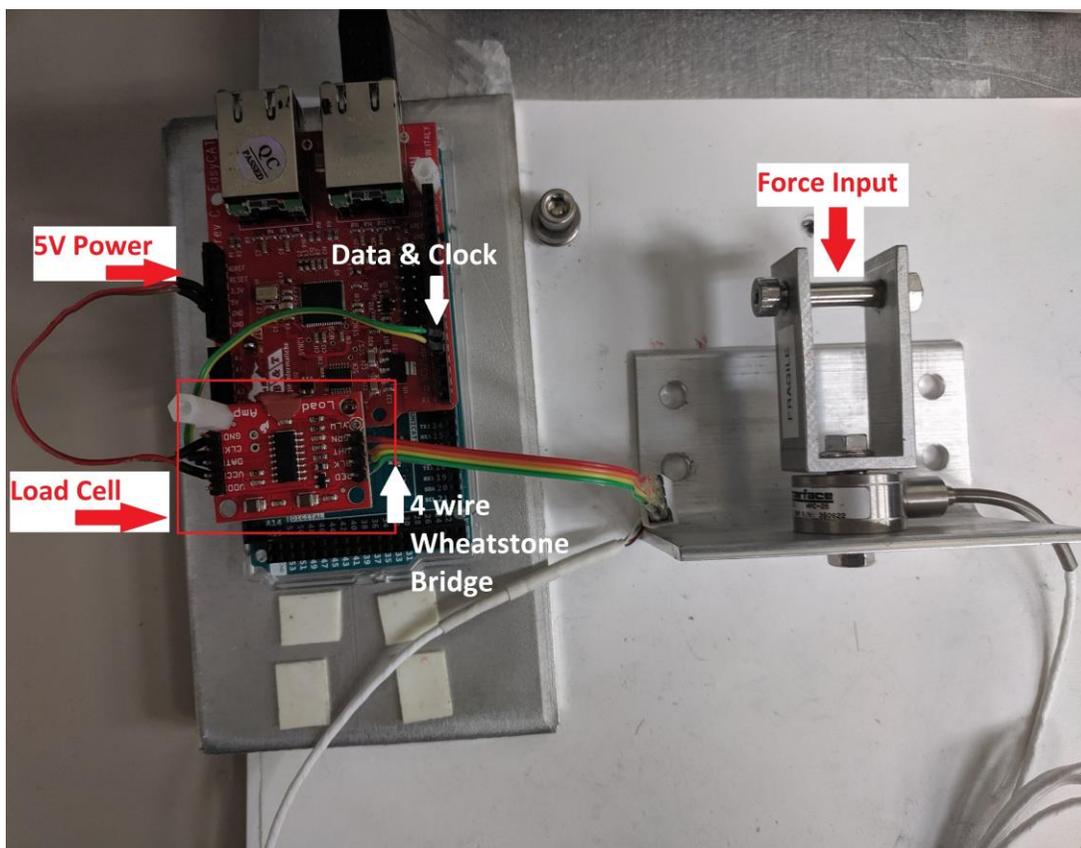
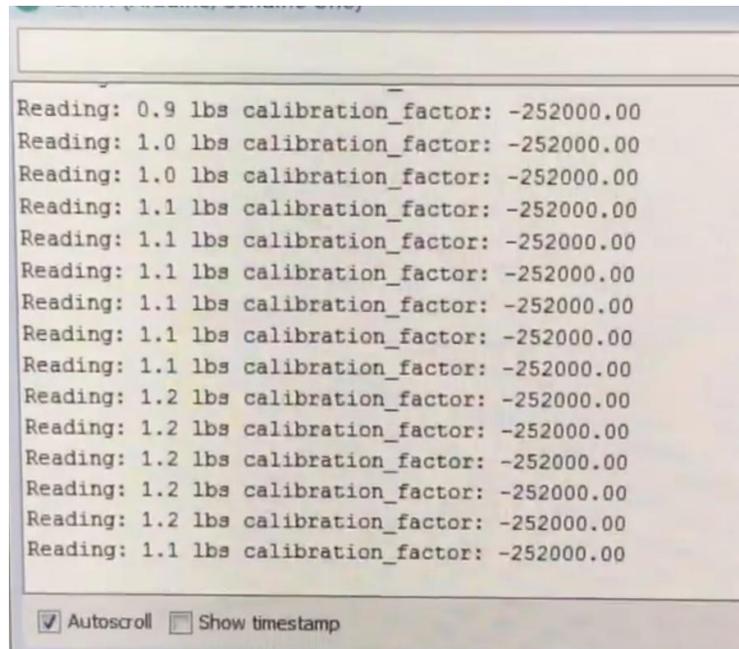


Figure 13: HX711 Load Cell Setup

The image shows a screenshot of an Arduino Serial Monitor window. The window has a title bar at the top and a text area below. The text area contains 15 lines of output data, each starting with "Reading:" followed by a weight value in lbs and a calibration factor. The weights range from 0.9 lbs to 1.2 lbs, with some values repeating. The calibration factor for all readings is -252000.00. At the bottom of the window, there are two checkboxes: "Autoscroll" which is checked, and "Show timestamp" which is unchecked.

```
Reading: 0.9 lbs calibration_factor: -252000.00
Reading: 1.0 lbs calibration_factor: -252000.00
Reading: 1.0 lbs calibration_factor: -252000.00
Reading: 1.1 lbs calibration_factor: -252000.00
Reading: 1.2 lbs calibration_factor: -252000.00
Reading: 1.1 lbs calibration_factor: -252000.00
```

Figure 14: Load Cell Output on Arduino Serial Monitor

5.6 IMU & NS-RS232 communication

Although not in our requirements, when surveying fellow WEC developers most stated an IMU was present in their design for obtaining the Euler angles and x, y, z accelerations of the WEC itself. Such information is valuable for observing the WEC in the current environment and in controlling how it should respond for the highest power absorption.

IMUs often communicate through the RS-232 standard. This happened to be the case with our IMU from Xsens; the Mti-20-2A864. Since the Mega operates its serial communication (RX & TX) through TTL (transistor-transistor logic), in order to transfer data between it and the IMU, a TTL to RS-232 converter was needed. Luckily these are readily available, of which we chose the NS-RS232 from NulSom Incorporated. This meant that the data output from the IMU was sent in RS-232 to the converter, and then read as TTL at the development board

and vice versa. The setup of the IMU and the Mega/EasyCAT is detailed in Figure 15.

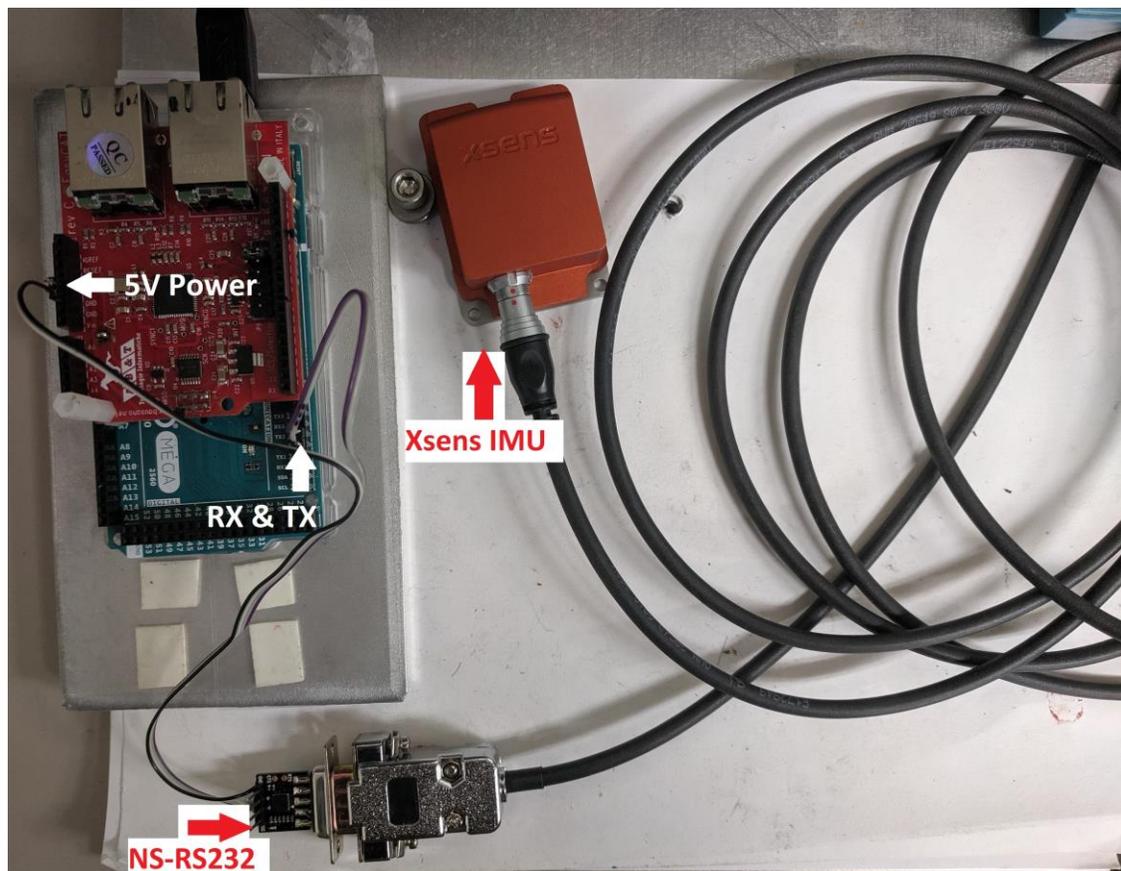


Figure 15: IMU Setup

To communicate with the IMU, Xsens provides their own downloadable MT Manager software. Its simple operation allows users to choose the output by checking boxes and clicking options in drop-down menus shown in Figure 16 and a typical output is seen in Figure 17. As the Mega does not support the Xsens device application program interface, another option they offer is to use the low-level communication method. This allows communication through

an MT message built with the following structure shown in Figure 18. The message is built starting with the pre-amble (start of the packet), bus identifier (which IMU to address), message identifier (which command), length of data, the actual data, and the checksum. In Figure 19 a message is built in the order previously stated. C0 is the message identifier for *SetOutputConfiguration* and 28 in hexadecimal indicates that there are 40 data points. Highlighted is the data listed by their name below the message.

Output Configuration			
Device Settings	<input checked="" type="radio"/> Normal mode <input type="radio"/> String report mode		
Synchronization Options	Preset: XDA Processing <input type="checkbox"/> Link Formats <input type="checkbox"/> Link Freqs		
Modeling Parameters			
Timestamp	<input checked="" type="checkbox"/> Packet Counter <input checked="" type="checkbox"/> Sample Time Fine <input type="checkbox"/> Sample Time Coarse <input type="checkbox"/> UTC Time		
Orientation	No Orientation	Floating Point 32-bit	400 Hz
Inertial Data	<input checked="" type="checkbox"/> Δq <input type="checkbox"/> Rate of Turn <input checked="" type="checkbox"/> Δv <input type="checkbox"/> Acceleration <input type="checkbox"/> Free Acceleration	Floating Point 32-bit	400 Hz
Magnetic Field	<input checked="" type="checkbox"/> Magnetic Field	Floating Point 32-bit	100 Hz
Temperature	<input type="checkbox"/> Temperature	Floating Point 32-bit	400 Hz
Pressure	<input checked="" type="checkbox"/> Barometric Pressure	Floating Point 32-bit	50 Hz
High-Rate Data	<input type="checkbox"/> Acceleration HR <input type="checkbox"/> Rate of Turn HR	1000 Hz	
Status	<input checked="" type="checkbox"/> Status Word <input type="checkbox"/> Status Byte		
Position and Velocity	<input type="checkbox"/> Latitude and Longitude <input type="checkbox"/> No Altitude <input type="checkbox"/> Velocity <input type="checkbox"/> GPS Age	Fixed Point 16.32	400 Hz
GNSS Data	<input checked="" type="checkbox"/> Pvt Data <input type="checkbox"/> SatInfo	4 Hz	

Figure 16: MT Manager Output Configuration Preferences [28]

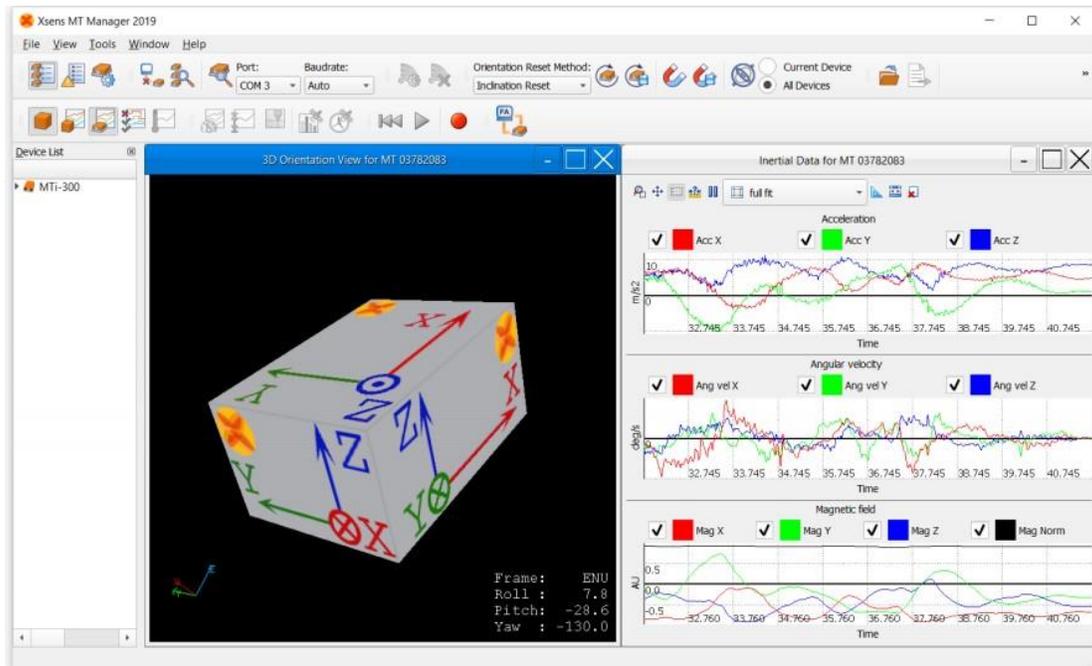


Figure 17: MT Manager Output Results [28]

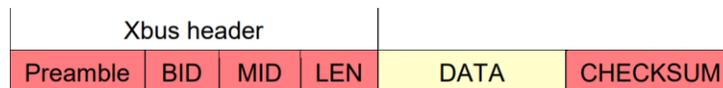


Figure 18: MT Message of Standard Length [29]

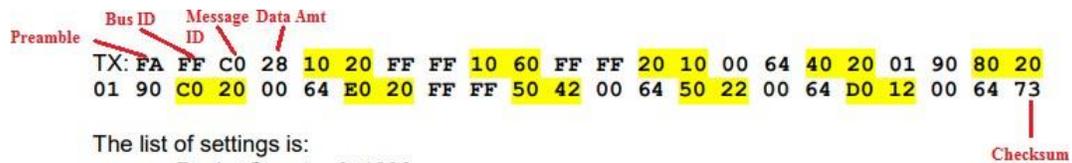


Figure 19: MT Message - Setting the Output Configuration; highlighted are the listed settings [29]

The code written for the IMU ensures that it is outputting Euler angles and x, y, z accelerations upon request. An example of roll and pitch outputs on the Arduino Serial Monitor is shown in Figures 20 and 21 where the IMU is positioned to the specified axis. From left to right is the roll, pitch, yaw, x, y, and z accelerations. The yaw column indicates close to 180 degrees instead of 0 because the IMU during testing was simply positioned a half turn off center.

COM6						
-66.21	4.09	172.72	-1.21	-9.71	3.29	13
-67.63	4.27	172.51	-0.60	-9.33	2.25	14
-68.66	4.31	172.34	-0.68	-9.49	2.71	13
-69.66	4.32	172.27	-0.52	-9.24	3.28	13
-70.37	4.32	172.43	-0.39	-9.44	4.25	14
-71.18	4.34	172.78	-0.73	-9.30	4.64	14
-71.80	4.36	173.11	-1.67	-9.04	4.88	13
-72.59	4.41	173.36	-1.59	-9.44	4.21	13
-73.71	4.42	173.36	-1.41	-9.40	3.07	14
-74.51	4.44	173.17	-0.91	-9.04	1.83	14
-75.12	4.47	172.84	-0.63	-9.48	0.63	13
-75.27	4.49	172.70	-0.27	-9.41	0.82	14
-75.12	4.52	172.73	-0.65	-9.08	1.61	13
-74.94	4.59	172.93	-0.46	-8.43	3.14	13
-74.98	4.52	172.33	-1.25	-8.12	3.81	15
-75.23	4.35	172.76	-0.65	-9.25	2.98	13
-75.57	4.20	172.52	-1.56	-11.39	1.52	13
-75.95	4.10	172.28	-1.35	-10.92	1.56	14
-76.17	4.15	172.75	-0.94	-9.85	1.80	13
-76.50	4.36	172.52	-0.56	-8.62	2.49	13
-76.93	4.56	172.28	-0.39	-8.14	3.07	14
-77.41	4.62	172.15	-0.32	-8.74	3.17	14
-77.91	4.43	172.13	-0.53	-10.02	2.90	13
-78.44	4.55	172.12	-0.96	-11.20	2.56	13
-79.24	3.91	172.09	-0.66	-11.55	2.21	14
-79.68	3.82	172.60	-0.95	-10.43	2.22	13
-80.08	4.14	171.96	-0.91	-8.66	1.81	13
-80.29	3.95	171.70	-0.63	-7.47	2.98	15
-81.44	3.76	171.67	-0.42	-7.54	2.14	13
-81.78	4.74	171.74	-0.93	-9.67	1.79	13
-81.89	4.56	171.31	-0.97	-11.43	1.51	14
-81.73	4.76	171.81	-1.06	-11.85	1.34	13
-81.76	4.70	171.76	-0.93	-10.67	1.55	13
-81.94	4.39	171.71	-0.67	-8.64	1.36	14
-82.73	4.05	171.70	-0.82	-7.82	1.41	13
-82.66	4.93	171.73	-1.15	-8.39	0.87	13
-82.34	4.09	171.72	-1.27	-10.22	0.55	14
-82.31	4.27	171.61	-1.28	-11.44	0.76	14
-82.62	4.32	171.42	-0.91	-11.32	1.12	14

Autoscroll Show timestamp

Figure 20: Arduino Serial Monitor of IMU output; boxed in red is the float value of the roll variable

COM6

5.44	-66.79	165.89	9.02	1.06	1.67	14
4.92	-67.11	165.89	9.11	1.52	1.90	13
14.17	-67.28	166.31	9.36	1.27	2.96	13
23.26	-67.74	166.91	9.51	1.96	2.18	14
22.88	-68.56	166.78	9.22	2.22	3.14	13
22.84	-69.56	166.01	9.32	2.27	5.35	13
23.43	-70.71	164.38	9.51	0.97	6.82	14
23.99	-71.40	168.06	9.22	0.35	5.02	13
24.08	-71.85	162.93	9.12	0.41	3.13	13
23.56	-72.25	162.95	8.83	1.33	1.17	15
22.99	-72.69	163.04	8.88	2.17	1.10	13
22.82	-73.37	162.55	9.11	2.46	2.02	13
23.26	-73.29	162.03	9.03	1.58	3.68	15
23.90	-71.19	161.29	9.76	0.31	4.24	13
23.84	-74.21	162.06	9.81	0.18	1.89	13
23.86	-74.25	162.01	9.03	0.93	0.72	13
23.68	-74.18	161.99	9.73	1.74	1.13	15
24.26	-74.25	161.00	9.60	1.64	2.80	13
25.27	-75.25	161.10	9.59	1.00	3.94	14
26.28	-75.63	160.96	9.57	0.42	4.05	14
26.21	-75.89	160.20	9.26	0.38	2.86	13
26.25	-75.32	158.96	9.38	0.94	1.82	13
26.25	-75.65	157.43	9.80	2.14	0.21	14
26.81	-75.31	156.59	10.01	2.52	1.01	14
26.22	-75.82	156.95	9.94	1.75	2.89	13
26.39	-76.95	157.41	9.60	0.72	3.54	14
25.89	-76.32	157.40	9.41	0.06	2.42	13
25.52	-76.72	156.48	9.01	0.42	1.88	13
26.19	-76.93	155.43	9.42	1.21	0.69	14
27.24	-77.02	154.61	9.75	1.65	1.11	13
27.83	-77.02	154.40	10.12	1.93	2.58	14
27.42	-77.12	154.54	9.76	1.47	3.15	14
27.23	-77.35	154.58	9.21	0.61	2.13	13
27.47	-77.65	154.22	9.15	0.31	0.92	13
27.93	-77.82	153.55	9.24	0.83	0.23	14
27.97	-77.75	153.38	9.57	1.40	0.82	14
27.33	-77.56	153.79	10.03	1.44	2.33	13
26.81	-77.40	154.34	9.85	0.86	3.22	14
26.91	-77.43	154.22	9.50	0.34	2.29	14

Autoscroll Show timestamp

Newline 115200 baud

Figure 21: Arduino Serial Monitor of IMU output; boxed in red is the float value of the pitch variable

5.7 ADS1115 4-20 mA Current Loop Receiver

Sensor signals come in two forms: voltage or current. Choosing which output is fit for an application depends on the degree of accuracy required and the layout of the circuit. Since data travelling in an industry application is commonly wired in series for serial communication, current is the preferred output because its value does not change whereas a voltage signal is more susceptible to interference and will lose the signals true value as the travel distance becomes longer. As mentioned before, voltage outputs use ADCs before sending the result to a computer. A current output uses a 4-20 mA current loop receiver to digitize the signal. To test this requirement a voltage controlled current source (VCCS) was designed since a sensor that outputs this current was not provided; its schematic is shown in Figure 22. The current was then read by the ADS1115 current loop receiver. Figure 23 shows the setup of this device with the Mega/EasyCAT.

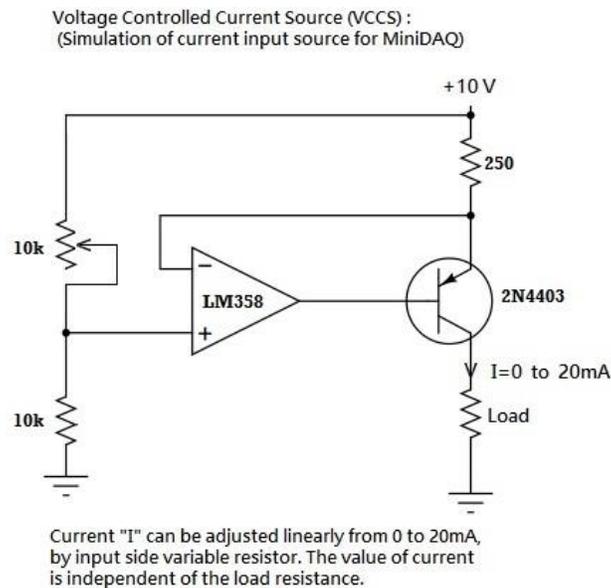


Figure 22: Circuit for Simulated Current as reading to ADS1115

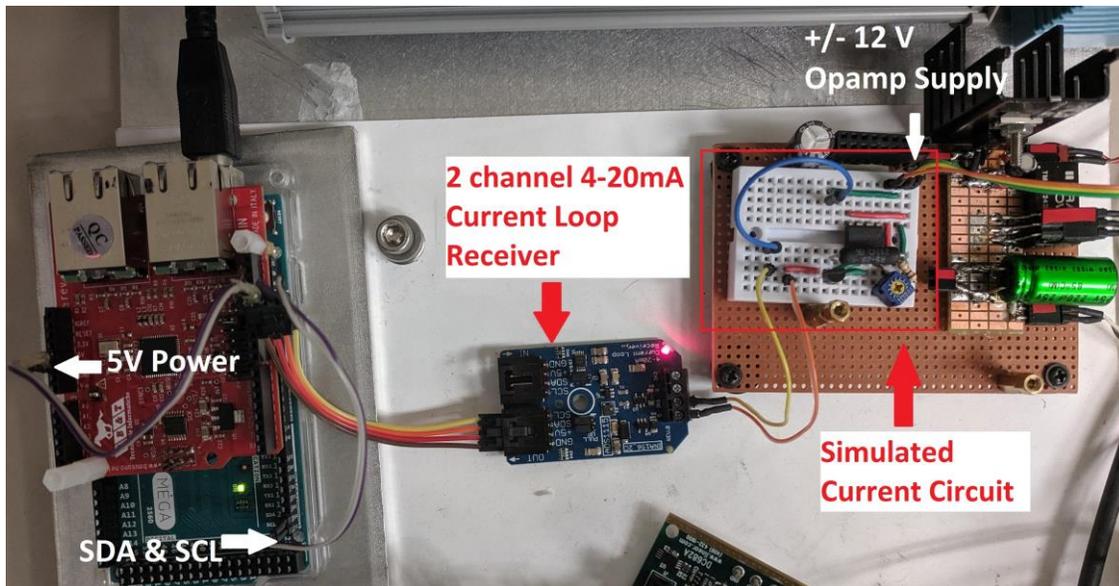


Figure 23: ADS1115 4-20 mA Current Loop Receiver Setup

Setup consists of defining the I2C address for the device, defaulted to the GND pin. After trial and error a gain of two yields results in our desired range. The ADS1115 is then set to continuously convert at a rate of 128 SPS (changing it made no significant difference in results). Fortunately, the device has a 1 LSB offset error yielding a 0.01% gain error, requiring no further calibration for our uses [30],[31],[32]. To receive values, the I2C register is read (SDA & SCL) which contains the data bits. Figure 24 is the output of the device through the Arduino Serial Monitor where the left column is the integer output and the right is this value converted to current in milli-amps.

31533	19.803
30955	19.440
30401	19.092
30131	18.922
29605	18.592
28942	18.176
28590	17.955
28254	17.744
27705	17.399
27244	17.109
26661	16.743
25774	16.186
25280	15.876
24283	15.250
23832	14.966
23162	14.546
22743	14.283
22380	14.055
21885	13.744
21260	13.351
20596	12.934
20083	12.612
19842	12.461
19339	12.145
18728	11.761
18321	11.506
17759	11.153
16831	10.570
15986	10.039
15465	9.712

Figure 24: Arduino Serial Monitor displaying Current Output in Integer and Float Form

5.8 CUI AMT23 Absolute Encoder

An encoder as a transducer is commonly found whenever measurements of a rotary or linear machine are needed. The outputted digital signal can then be used to determine speed, orientation, or position of a mechanical piece. This for example can be used to monitor the flaps on a WEC, giving the ability to alter its behavior and motion to incoming waves if its flaps are not in a correct state. This report focuses on rotary encoders that come as either incremental or absolute. Incremental encoders generate pulses typically seen as 2 square waves 90 degrees out of phase. As the pulses pass, they are counted and direction is determined by the order of the generated square waves. While

they are simpler

to use, the downfall of this type of encoder is that it must have a reference to adjust itself to every time it is turned on to yield accurate position readings. Absolute encoders, however, have distinct values for each point of rotation, removing the need for a reference and yielding more reliable and valid results [33].

The CUI AMT23 absolute encoder was provided for use. It communicates with the Mega through a single-ended 3 wire SSI (data, clock, chip select) and outputs a resolution of 2^{14} bits representing the position between values of 0-16383 with an accuracy within 0.2 degrees [34]. This was attached to an AMETEK Pittman brushed DC motor [35] to read its position. Figure 25 is the setup of the encoder on the motor, and the two wired to the Mega/EasyCAT. Figure 26 shows the integer values received from the encoder displayed through the Arduino Serial Monitor.

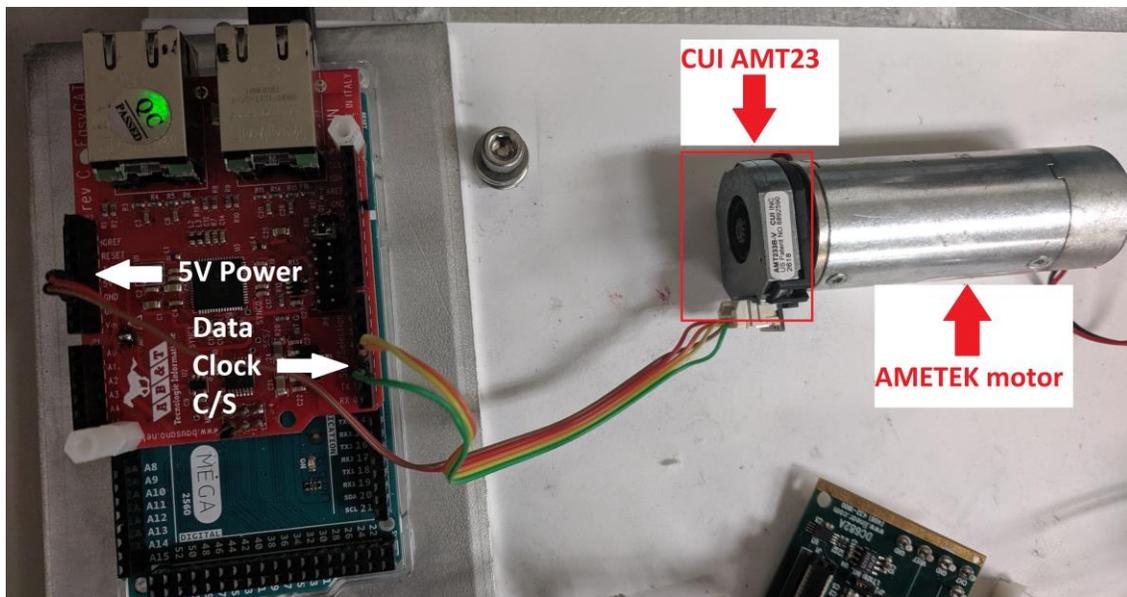
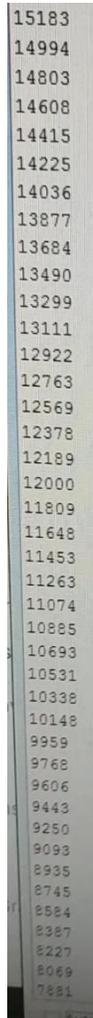


Figure 25: CUI AMT23 Setup



15183
14994
14803
14608
14415
14225
14036
13877
13684
13490
13299
13111
12922
12763
12569
12378
12189
12000
11809
11648
11453
11263
11074
10885
10693
10531
10338
10148
9959
9768
9606
9443
9250
9093
8935
8745
8584
8387
8227
8069
7881

Figure 26: Integer Values of Encoder; begins from 16384 and decreases to 0

5.9 MAX1484 - SSI protocol

As with the IMU, WEC developers also stated that the SSI communication standard is desirable when sending data across long distances because it operates on differential signals which are less influenced by EMI (so long as the lines are balanced). Also with small-scale testing, noise resistance needs to be considered because low voltage system [36] signals are at higher risk to being altered. This proves another advantage; the differential signals allow twice the protection compared to a single-ended signal simply as a result of the difference between the high and low voltage level (positive minus a negative is twice the positive value!). It is additionally more effective when the noise between the wires are uncorrelated to one another.

Knowing that SSI works on the RS-422 standard and the familiarity with MAXIM products, to fulfill this communication requirement the MAX1484 chip was chosen; a TTL to RS-422 converter. There was no provided hardware that communicated through the standard, so two MAX1484 chips were connected together to pass along a signal from TTL to RS-422 and back to TTL. This was tested on a LED. The Mega sent a digital signal to the driver input pin on the first chip, converted it to RS-422, read as RS-422 by the second chip, then converted back to TTL again by the second chip. The final signal was outputted at the receiver output where the LED was connected and blinking indicated communication was successful. The figure below represents the setup.

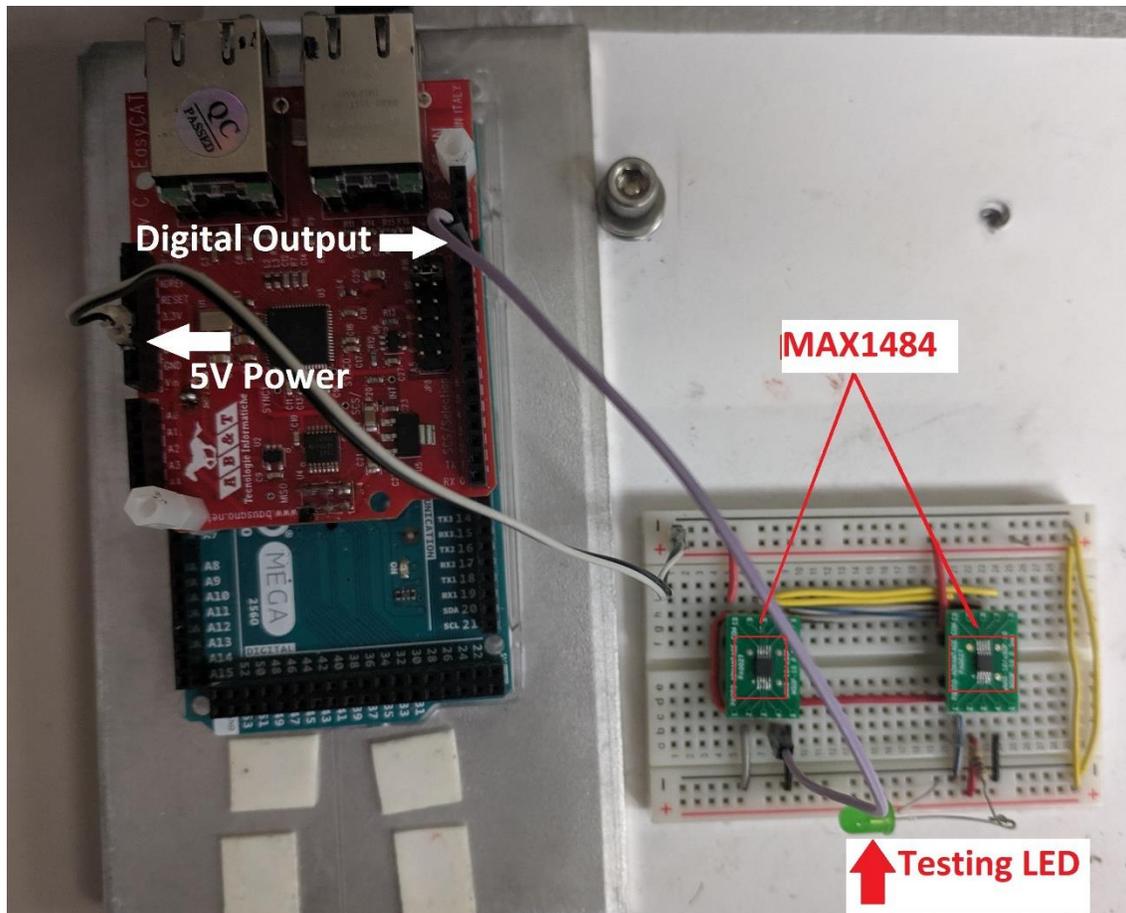


Figure 27: SSI Testing Setup

5.10 Anti-Aliasing RC filter - Noise Testing

When gathering data, it is critical to choose a suitable sampling frequency based on the intended bandwidth. If this is not chosen wisely, an aliasing effect can occur meaning that, when multiple signals are sampled, they appear to be identical to one another. We want to ensure that a signal, e.g. 80 Hz, is read by the computer as 80 Hz and not a different frequency. Aliasing can cause drastic problems such as false readings. Another issue to note is the presence of EMI due to the natural operation of a WEC as mentioned earlier.

To ensure that the MiniDAQ will provide accurate data for all frequencies in our bandwidth and to filter out any noise, an anti-aliasing RC filter was built and tested. The classic low pass filter design is changed for a differential input by adding a second resistor as seen below in Figure 28 [37]. Figure 29 displays the setup of the filter testing where the bandwidth range tested was given as 0.1 Hz (typical period of waves) to 50 Hz. As the sampling frequency of 100 Hz is twice the upper limit, the filter should hold true to the Nyquist theorem and avoid the aliasing effect.

Testing involved running the system under noiseless conditions and monitoring the input and output of the circuit up to the cutoff frequency of 50 Hz. This was repeated near a 230 V induction motor running at 60 Hz to simulate noise. Theoretically, as 50 Hz is approached, the amplitude of the output should steadily decrease, and reach a value 3 dB below the input at the cutoff (about 0.707 times the signal input), exponentially decreasing afterwards. The table in Table 1 shows that the circuit worked well in filtering out the unwanted distortion as both noiseless and noise values are similar. It also proves that the circuit was successfully built for a cutoff value of 50 Hz, shown by the highlighted row. Reading each row, all noiseless and noise outputs are above this 3dB value until 50 Hz, the first time it falls below which is the expectation. After 50 Hz, the difference between the input and output should be exponentially lower and the circuit presents this. The input and output frequencies were also monitored and were identical throughout all tests indicating that aliasing had not occurred.

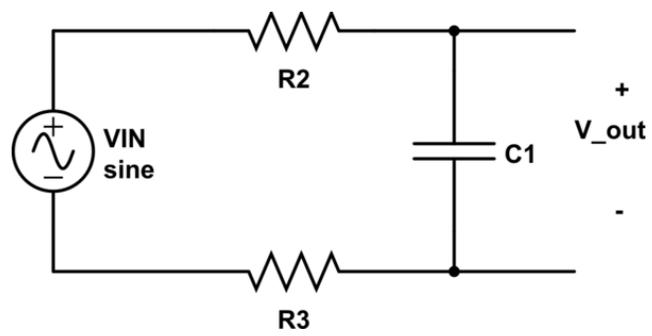


Figure 28: Anti-Aliasing RC filter for Differential Input [37]

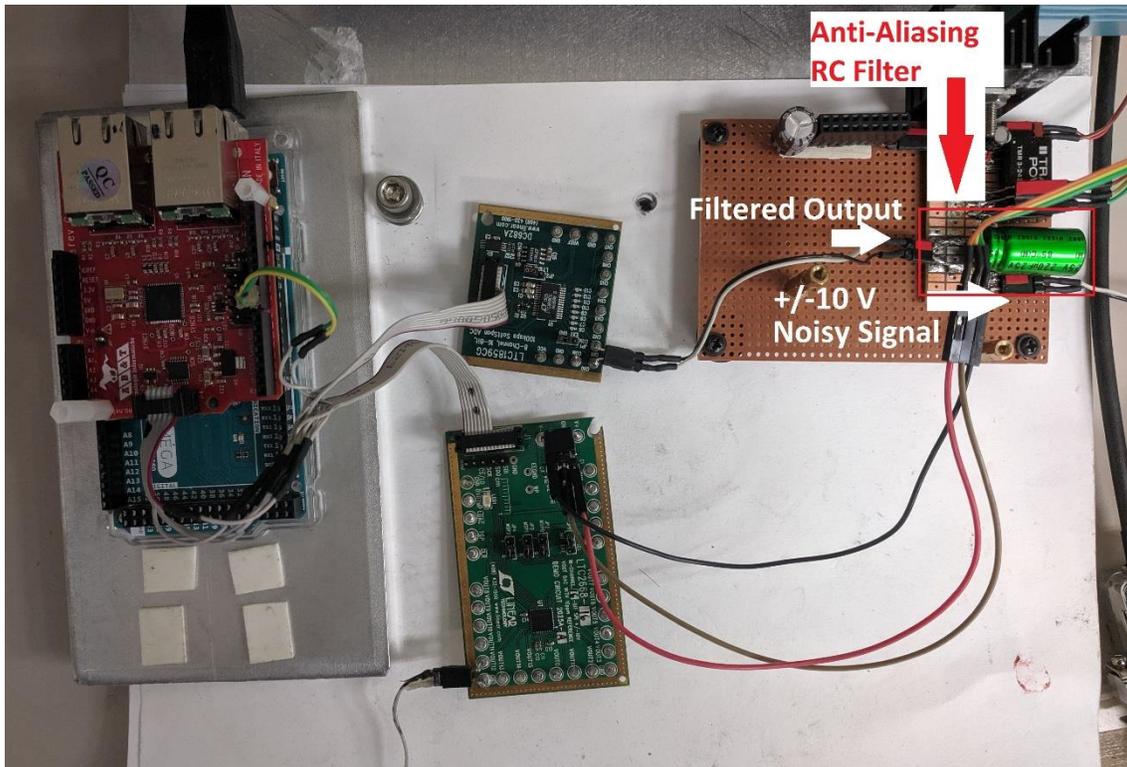


Figure 29: Anti-Aliasing RC Filter Setup

Frequency [Hz]	Signal Input [V]	Filter Output [V]		3dB below [V]
		Noiseless	Noise	
8.129	13.6	13.4	13.4	9.62
11.04	11.8	11.4	11.2	8.34
14.14	10.6	10.2	10	7.49
25.08	7.6	6.6	6.5	5.37
35.2	6.4	5	4.8	4.52
50.8	5.2	3.4	3.2	3.67
65.4	3.8	2.2	1.8	2.68
80.7	2.5	0.2	0.1	0.707

Table 1: Table of values showing input frequency and signal voltage, outputted signal voltage noiseless vs noise conditions

6 Software

The following subsections describe the softwares used to develop and operate the MiniDAQ project. This includes both the light blue MiniDAQ box and the grey Real-Time System box of Figure 5. Each subsection focuses on a software, describing what it is used for and its purpose in this project. These softwares were chosen in combination with being hardware related and knowledge of their operational environment.

6.1 Arduino Integrated Development Environment

The Arduino Integrated Development Environment (IDE) simplifies coding by a great deal. Not only does it provide understandable functions, it is structured in a straightforward manner allowing the user to quickly adjust to writing with the environment. Arduino's popularity also opens up the option to utilizing a mass amount of open source libraries available online, removing any customization that could have been required by the user. The simplicity and ease of use on the software side again, makes Arduino an ideal hardware and software candidate for our system.

Figure 30 is a flow chart detailing the MiniDAQ code explained next. The Arduino *.ino* file's purpose is defining variables, sending/receiving data, and processing this information into specified formats. The EasyCAT header file provided by AB&T configures the EasyCAT shield allowing communication between itself, the Mega, and Speedgoat through EtherCAT. Our unique header file containing custom inputs and outputs created through the EasyCAT Configurator is included to give the EasyCAT access to the sensors data. Once variables and pinmodes are defined, the EasyCAT shield and IMU are initialized, both able to end the code if a failure occurs. If successful, the code runs through a loop of receiving/sending data to/from the sensors, moving/retrieving them onto/from the EtherCAT frame, and printing the execution time. Data interacts with EtherCAT through the *BufferIn* or *BufferOut* function; an example of one such buffer is shown in Figure 31. Here, the code is checking to see if data is available on the EtherCAT frame variable *Dac*, and if so, to use that integer value to output an analog signal using the DC2025A. The Load

Cell code is checking if there is force data available, then it acquires this and pushes it onto the EtherCAT frame as *load cell*. To check our sample time, the code contains the function *millis()* between the start and end of the loop section. This outputs onto the serial monitor the execution time of the loop which should be less than 10 ms to meet the 100 Hz sampling requirement.

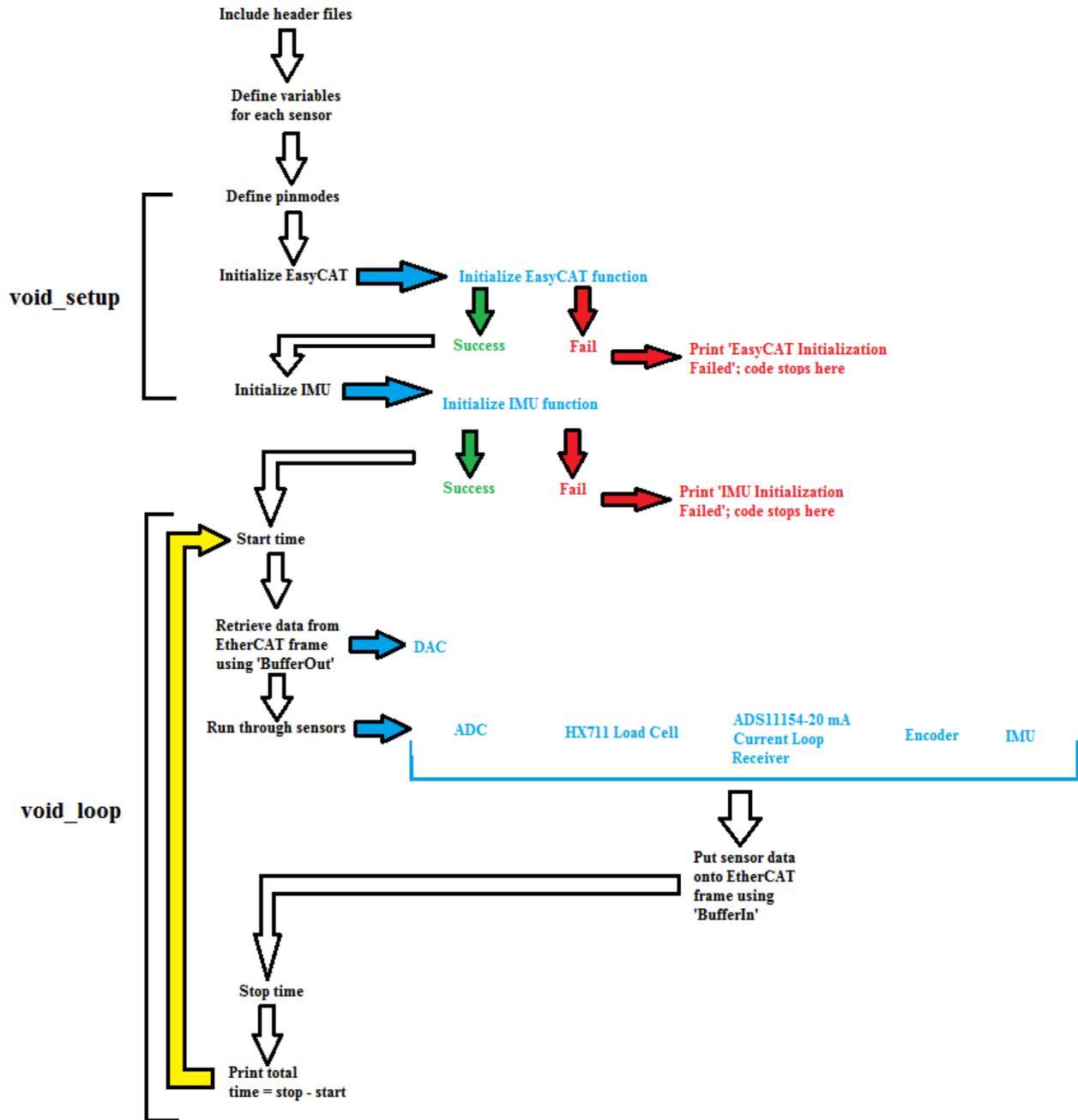


Figure 30: Flow Chart of MiniDAQ Code

```

//////////////////////////////////////      LTC_DAC      ////////////////////////////////////////
if(EASYCAT.BufferOut.Cust.Dac)
hi = highByte(EASYCAT.BufferOut.Cust.Dac);
lo = lowByte(EASYCAT.BufferOut.Cust.Dac);
write_DAC();

//////////////////////////////////////      HX Load Cell      ////////////////////////////////////////
if (scale.is_ready()){
load_cell = scale.get_units();
}
Serial.println(load_cell);
EASYCAT.BufferIn.Cust.load_cell = load_cell;

```

Figure 31: DAQ and Load Cell Section of MiniDAQ code with *BufferOut* and *BufferIn* functions boxed

6.2 EasyCAT Configurator

AB&T provides a configurator where users can define both custom names and variable types for their inputs and outputs. The application also creates a *.bin*, *.h*, and *.xml* file. The *.bin* file is written to the EasyCAT shield EEPROM from the application which configures the shield to recognize and obtain these inputs and outputs. The *.h* file needs to be moved to the same folder as the Arduino code in order for the IDE to identify these variables and the *.xml* file is placed to the correct TwinCAT location, detailed in the next section.

Figure 32 shows the setup of the configurator with one of each input and output available on the EtherCAT cable. You can see that the *Adc* inputs are signed 16 bits, *Roll* is a float, and *Dac* outputs are unsigned 16 bits. It is important to assign the data type correctly as it directly impacts how the sensors are communicated with.

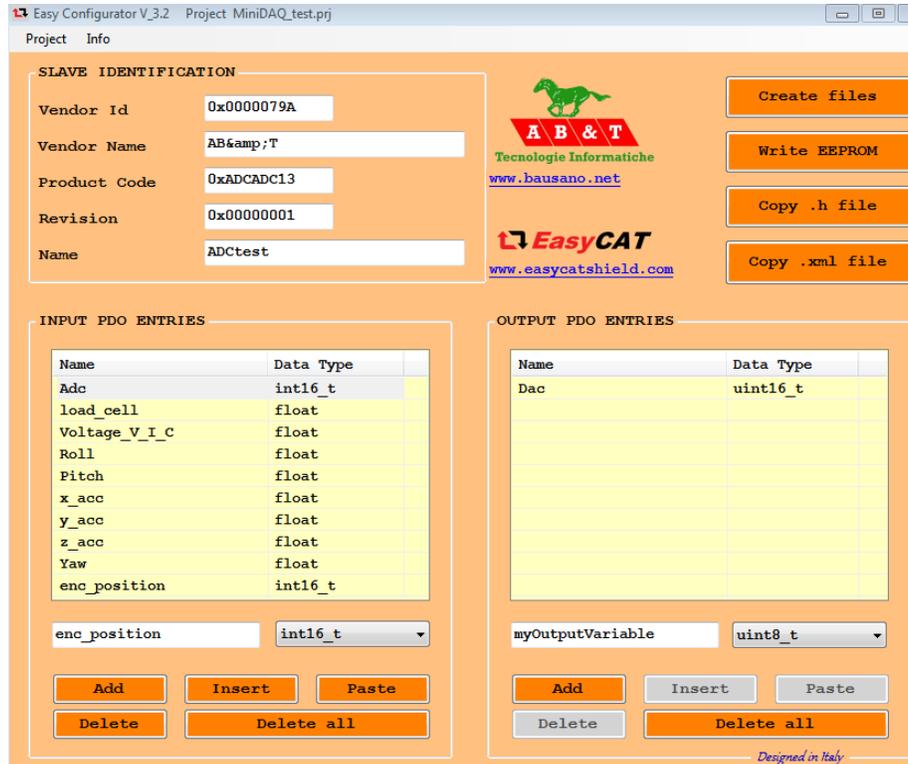


Figure 32: EasyCAT Configurator Application with Custom Inputs and Outputs

6.3 TwinCAT

TwinCAT is a software provided by Beckhoff that allows a PC to be a real-time controller, however, its only use for the MiniDAQ project was that the *.xml* file created by the EasyCAT Configurator is not compatible with the EtherCAT block in Simulink; this block supports *.xml* files made by TwinCAT meaning this had to be recreated for agreement among systems. Inside the software, the EtherCAT server is scanned and found - the EasyCAT shield on the Arduino shown as Box 1. Since the *.bin* and *.xml* files from the configurator have been written and moved to correct locations, TwinCAT finds the custom made inputs and outputs as shown in the yellow box. For Simulink to identify these, the

process from the configurator is repeated, but instead a Task is made. Inside TwinCAT shown in the figure below, inputs and outputs are again defined (green box), choosing the data type, and linking it to the inputs and outputs TwinCAT found earlier from the configurator (red box). This linking allows communication between the Speedgoat, SLRT, and data from the EtherCAT frame.

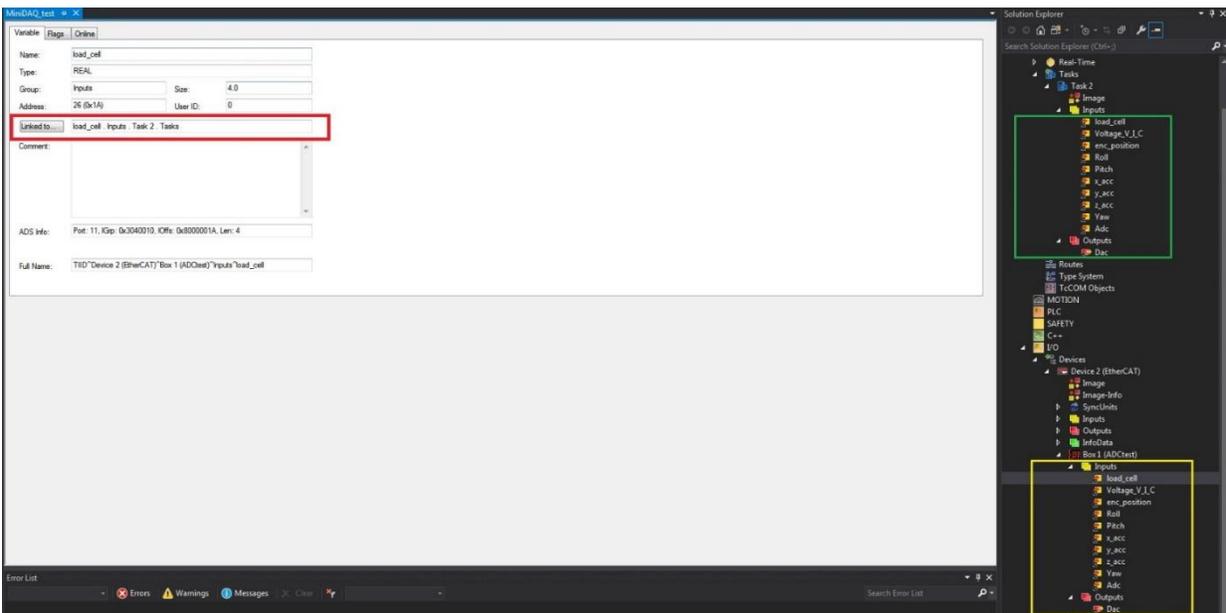


Figure 33: TwinCAT software showing inputs in Task (green box) and those found on the EasyCAT shield (yellow box). Boxed in red is the 'Linked to' which connects the Task's variables to the current input and output variable on the EasyCAT shield.

6.4 Speedgoat/Simulink

The real-time hardware chosen was Speedgoat (high performance target machine) which uses SLRT as its software. Connecting to the target machine requires one EtherNET cable from the host to target and configuring network settings such as the IP address, mask, and gateway. A successful connection is seen in Figure 34 below. After downloading the Speedgoat library, the input board IO133 is easily found. Inside is its initialization block that identifies what analog/digital input/output pins are being utilized along with their voltage levels. This would be used if inputting or outputting a voltage to the Speedgoat or sensor. As for EtherCAT communication, it has its own initialization block that takes the *.xml* file created by TwinCAT. Now the inputs and outputs made in TwinCAT are available for use in SLRT. The drop down menu in the *Ether-CAT PDO Receive* block displays the inputs seen in Figure 35. Similarly, the *EtherCAT PDO Transmit* block also allows the user to write to the outputs defined. Below this, Figures 36, 37, and 38 show the Simulink setup as well as the target scopes from Speedgoat showing the full system.

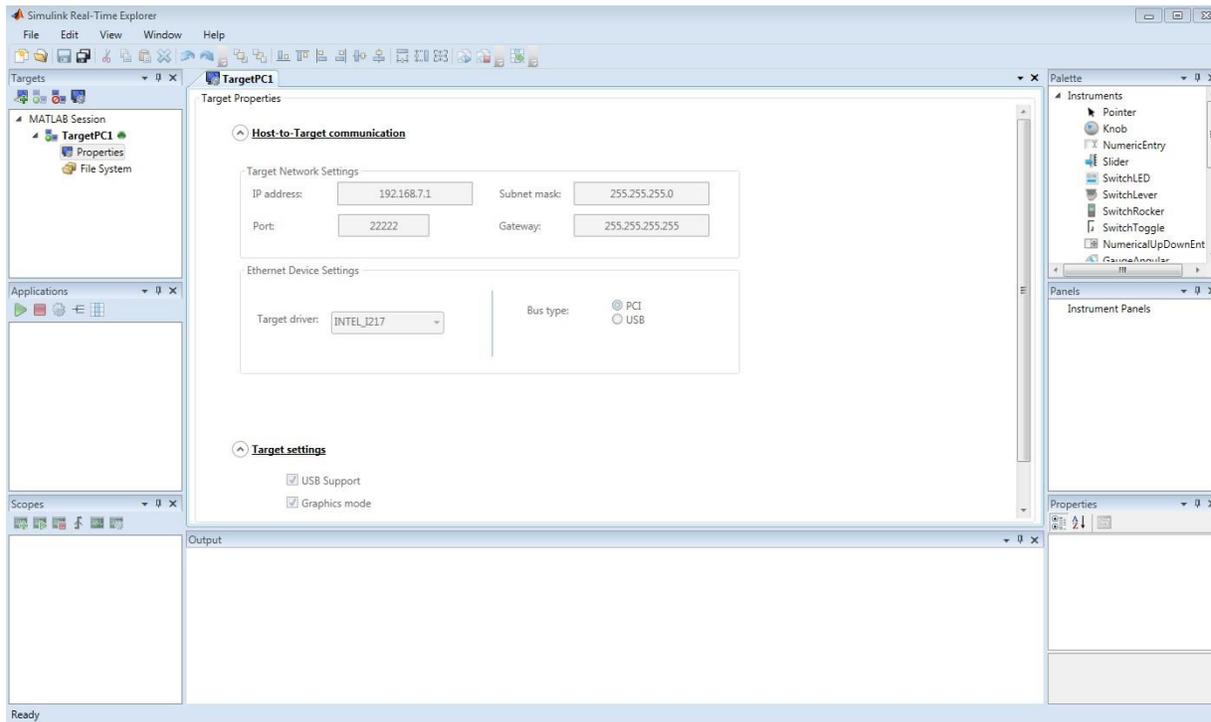


Figure 34: Network Settings in Simulink Real Time Explorer

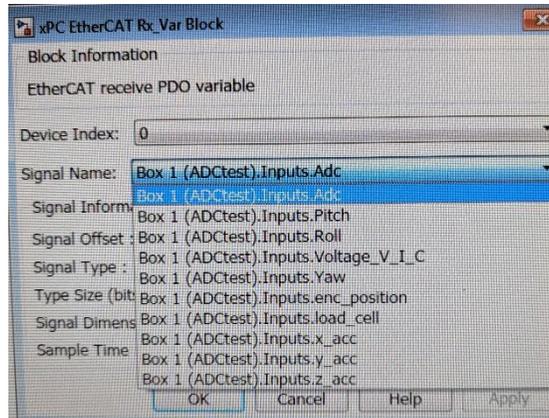


Figure 35: Various Defined Inputs seen in the *EtherCAT PDO Receive Block*

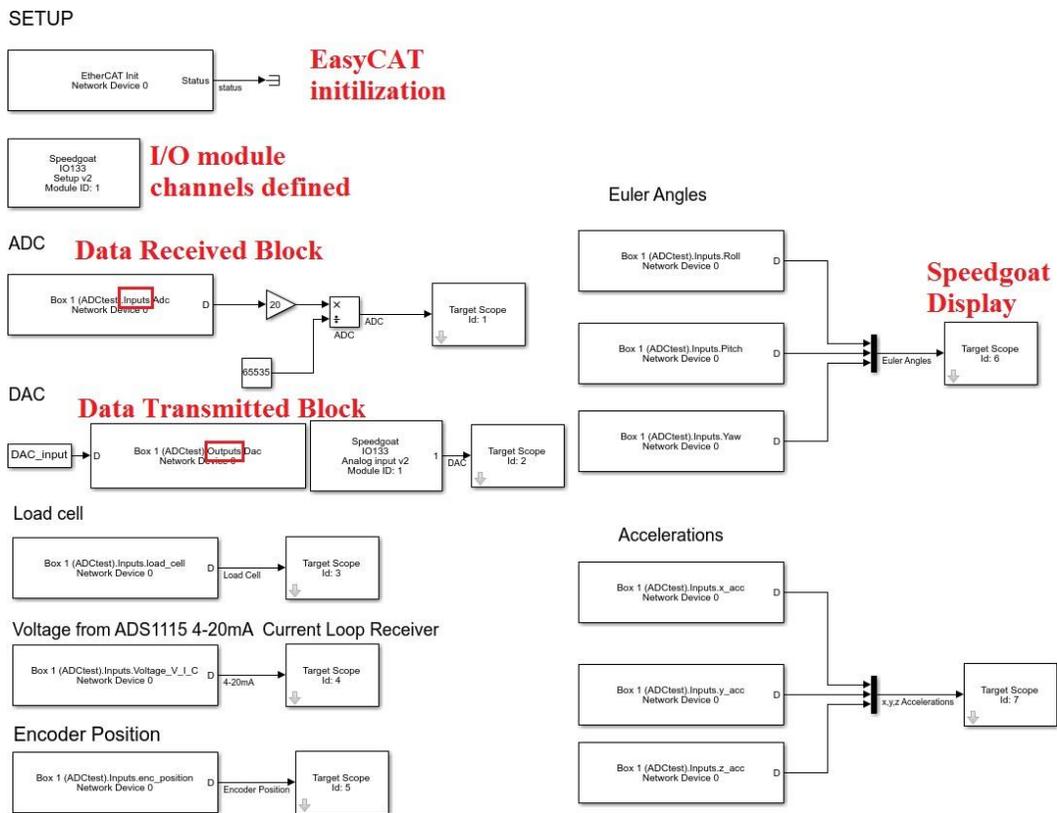


Figure 36: Overall SLRT model

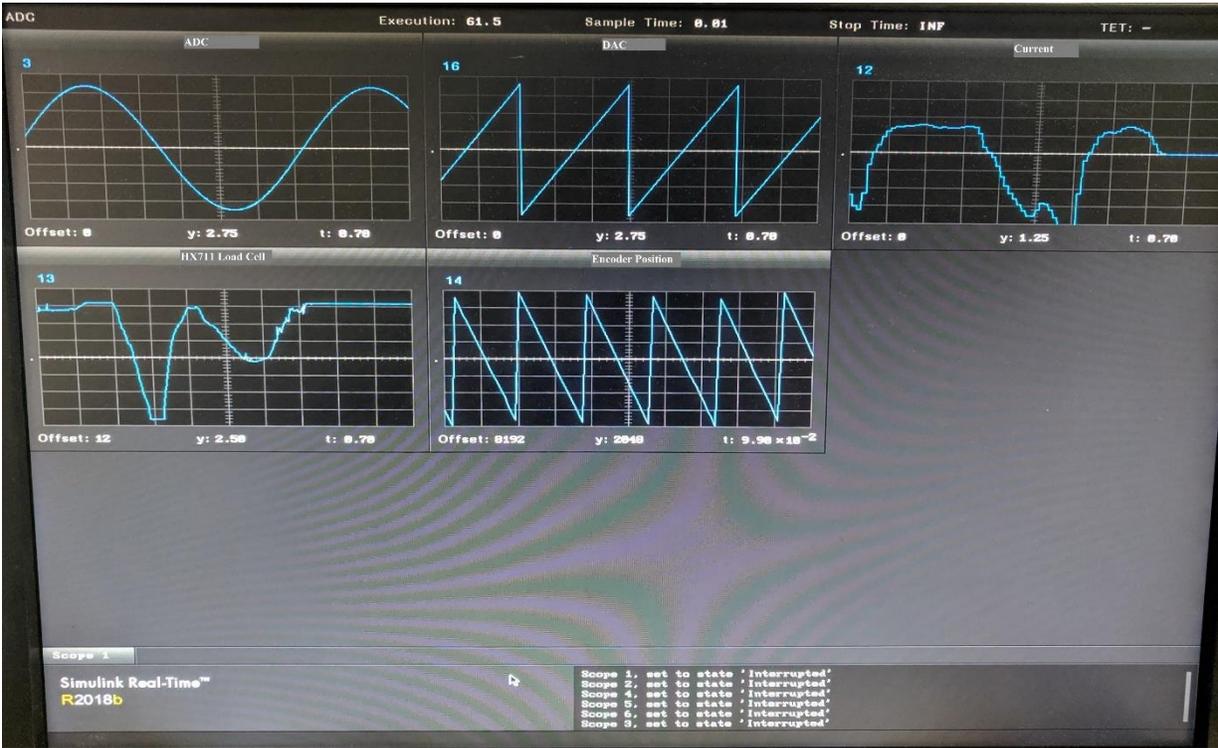


Figure 37: Sensor Outputs through Speedgoat Target Scopes

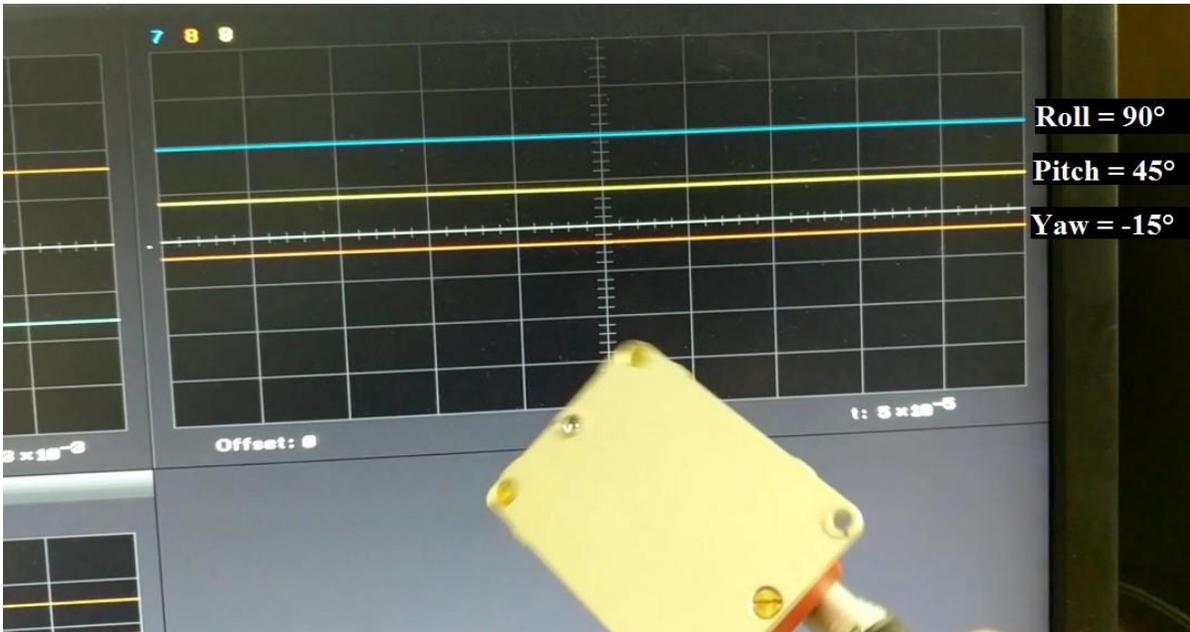


Figure 38: IMU Output through Speedgoat Target Scope

7 System Testing

Along with meeting the main requirements, SNL were interested in the frequency response (includes time lag) of the system. Frequency response is important in understanding how the output reflects the input when exposed to a certain signal, yielding the magnitude and phase of the output as a function of frequency. Ideally, the input and output should be equal indicating no interference was imposed when undergoing processes from the system, however, the MiniDAQ sensors introduce non-linearity, meaning there will be unwanted magnitude and phase change. Because ADC's are used in almost all sensors (including those present in here), the DC682A ADC was chosen as the output signal to characterize the entire MiniDAQ system.

7.1 Excitation signal options

To get a valuable frequency response, the excitation signal should be persistent. There are four common signals used to achieve this: pseudorandom binary sequence (PRBS), chirp, multisine, and stepped sine. These described next by Jouni Vuojolainen et al and Grauer et al detail when best to use each signal [38], [39]:

- PRBS signals are deterministic, periodic, and vary between two levels. They are easy to generate and have high spectral energy over a wide band range as well as an optimal spectrum for excitation signals.
- Chirp signals take the signal and sweep the frequency up or down in a period. They are appropriate when the frequency content needs to be controlled, i.e. when the power being concentrated over a specific frequency range is useful.
- Multisine signals are composed of multiple harmonic sinusoid components with different frequencies where the design of the signal is such to avoid the harmonics produced by the nonlinearities of the system. The amount of bands, starting and final frequencies, resolution, and amplitude can be chosen.
- Stepped sine contains all its power at one frequency, so in order to cover a certain range, different frequencies are to be individually excited. Useful when specific frequencies are focused on.

After weighing pros and cons to each signal and what would offer the most informative results, the PRBS signal was decided on as studies display this as yielding the most accurate results. These results from the PRBS frequency response will then be validated by a chirp signal.

7.2 Frequency Response of Speedgoat

Since function generators are unable to produce the PRBS signal, and the chirp sweep can be tedious, both signals were generated by the Speedgoat. By using the Speedgoat to produce these signals, its dynamics are now introduced into the

process. To rule out if these could influence the characterization of the DC682A ADC signal, a separate frequency response was tested on the Speedgoat's DAC to its ADC. Frequencies of interest are 0.1 Hz to 50 Hz (reasons the same as the low pass filter test).

Figure 39 displays the PRBS signal generated in Matlab and Figure 40 is the setup of this signal imported into SLRT where it was the input to Speedgoat's DAC. This analog output signal was then the input to Speedgoat's ADC. Both of these were logged to the workspace in Matlab as to make use of the system identification toolbox which provides quick evaluation of magnitude, phase, and step response between an input and output. Figure 41 is an overview of the toolbox. The *Import Data* window on the right is where the *in* and *out* data from Figure 40 were imported to.

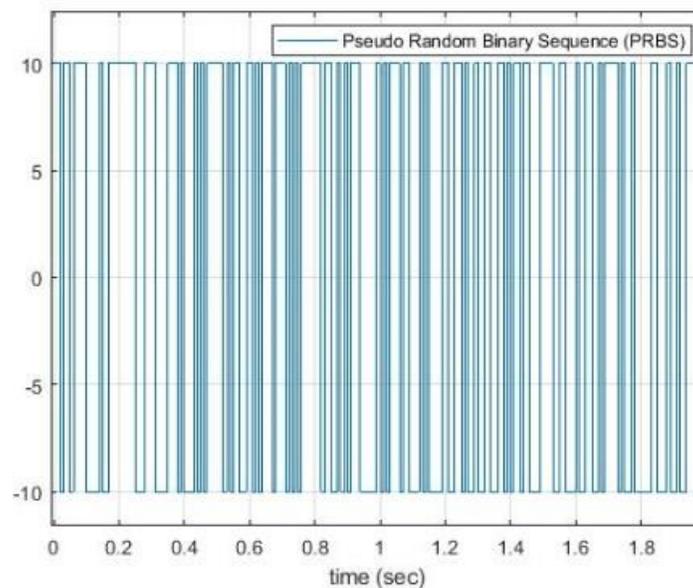


Figure 39: Generated PRBS Signal in Matlab

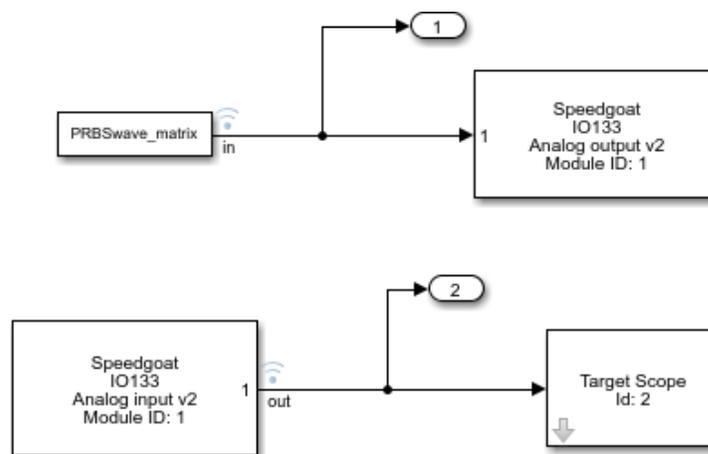


Figure 40: Speedgoat Frequency Response Setup

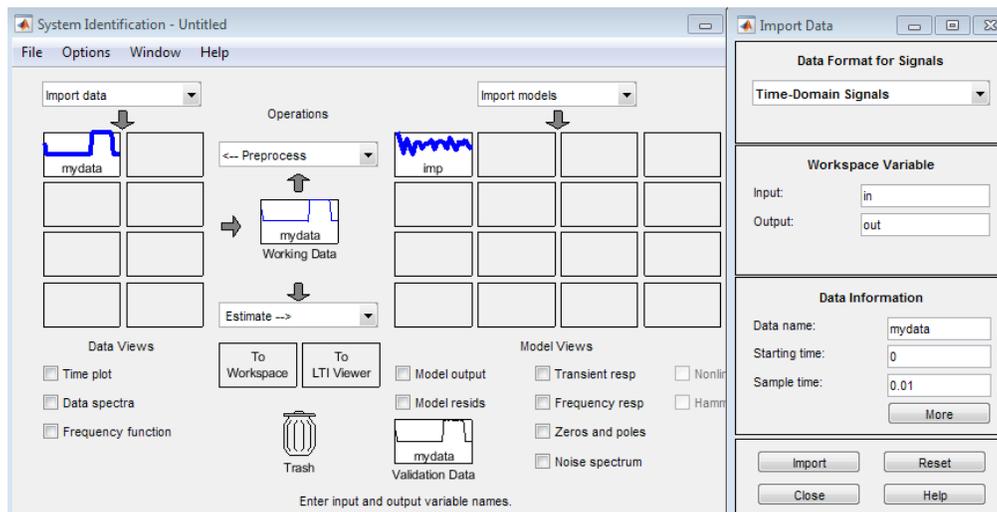


Figure 41: Matlab System Identification Toolbox

The transient and frequency response of this test are shown below. The magnitude is unaffected by the dynamics, staying at a constant value of one and the phase plot will be explained in a later section. Looking at the step response there is a delay of 0.01 seconds, equivalent to one sampling time. As this will not cause any negative effects in the ADC frequency response test, the generated signals by the Speedgoat are ruled acceptable to use.

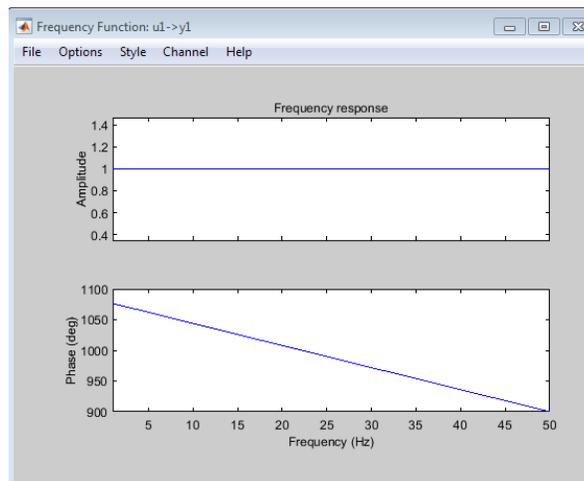


Figure 42: Speedgoat Frequency Response Test Magnitude and Phase Results

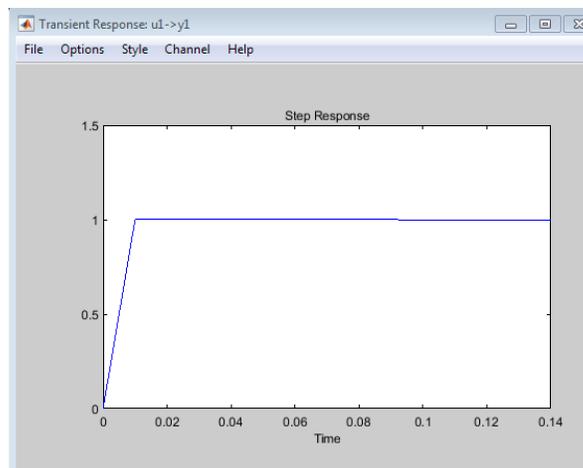


Figure 43: Speedgoat Frequency Response Test Step Response

7.3 Frequency Response of ADC

To continue on to testing the frequency response of the MiniDAQ ADC, the SLRT model was modified by including the DC682A ADC as producing the new *out* data for the system identification toolbox. Its physical input is the analog output from channel 1 of the IO133 module setup to output 10 V, meaning, the Speedgoat produces a signal to channel 1 accessible by 2 terminals on the IO133. Wires are connecting these terminals to the (+) and (-) terminals on the DC682A ADC.

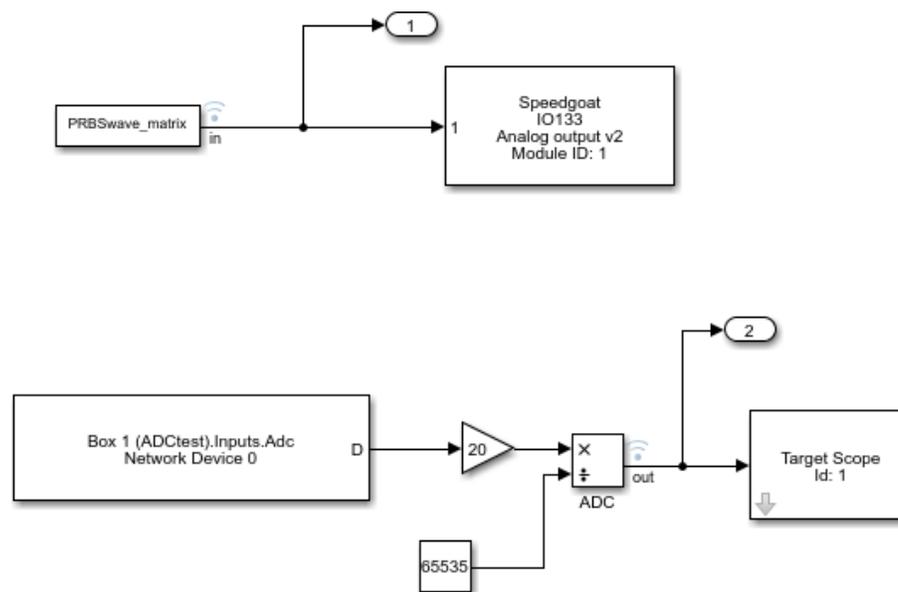


Figure 44: ADC Frequency Response Setup - PRBS

Running the system identification toolbox with the new *in* and *out* data produced by the setup in Figure 44 yields the following responses shown in Figure 45 and 46. The magnitude is wavering around 0.9 and 1 which is tolerable and the phase is exhibiting the same behavior as before. The system now has a 2 sample delay which is still within reason as confirmed by SNL.

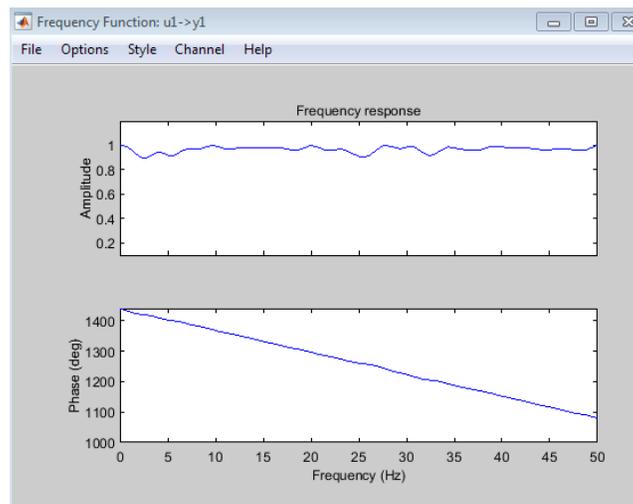


Figure 45: ADC Frequency Response Test Magnitude and Phase Results

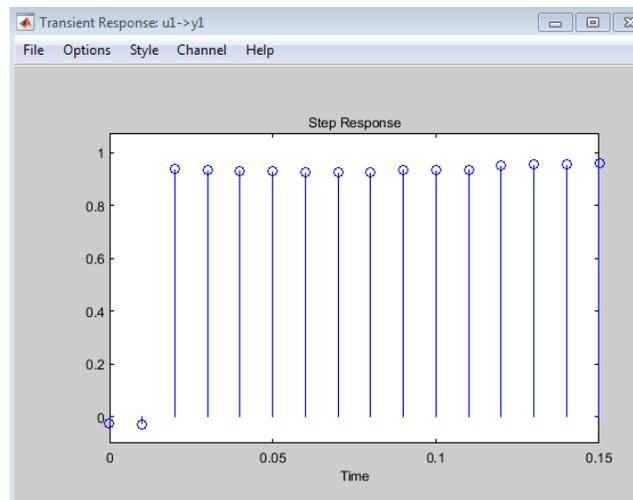


Figure 46: ADC Frequency Response Test Step Response

7.4 Explanation of Phase Plot

The phase plot is displaying the behaviour of a non-minimum phase system. To better understand this, a continuous sinusoid (typical input to the ADC) will be discretized at a sampling rate of $f_s = 1/T_s$ to get $u_1[nT_s]$:

$$u_1(t) = A \sin(2\pi f t)$$

$$u_1[nT_s] = A \sin(2\pi f nT_s)$$

Once processed by the ADC, $u_1[nT_s]$ appears as $y_1[nT_s]$ in the Speedgoat with a delay of 2:

$$\begin{aligned} y_1[nT_s] &= B \sin(2\pi f \cdot (n - 2) \cdot T_s) \\ &= B \sin(2\pi f nT_s - 4\pi f T_s) \\ &= B \sin(2\pi f nT_s + \varphi_y) \end{aligned}$$

The phase of $y_1[nT_s]$ as the function of input frequency f is:

$$\varphi_y = -4\pi f T_s \text{ [rad]}$$

$$\varphi_y = -4 \cdot (180) \cdot f T_s \text{ [deg]}$$

Knowing the desired sampling frequency of 100 Hz, the period is then $T_s = 10$ ms. The phase is now represented as:

$$\varphi_y = -7.2f \text{ [deg]}$$

Plotting the above theoretical equation and the experimental phase response of Figure 45 produces the figure below. This proves that the theoretical and experimental phase response are in agreement.

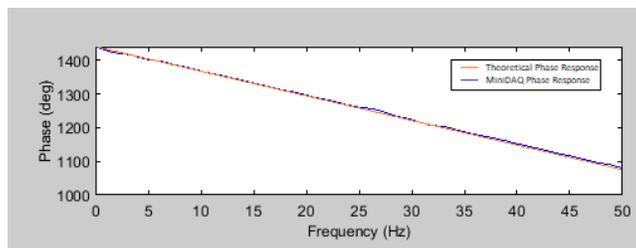


Figure 47: Experimental vs. Theoretical Phase Response.

7.5 Comparison to Chirp Signal

The SLRT model was again modified to Figure 49 below by switching out the PRBS signal for a chirp sweeping from 0 to 50 Hz as seen in Figure 50. It can be seen in this figure the initial cold start of the Speedgoat system resulting in a flat response at first, however during normal operation this is not present. This comparison test was conducted to validate the results from the frequency response test. Whenever the MiniDAQ is subject to any value between 0 – 50Hz, three features should be present as found in the previous sections:

1. The magnitude of 0.9 to 1 means the magnitude of both the input and output should be relatively the same
2. A 0° or 180° phase shift at specific frequencies will be seen
3. A 2 sample delay (0.02 s) will exist between the input and output

The specific frequencies feature 2 is referring to are determined by wrapping the phase plot of Figure 47 to 180° and -180° shown below. For example, at 0 Hz a 0° phase shift should be present, and at around 25 Hz, a 180° phase shift.

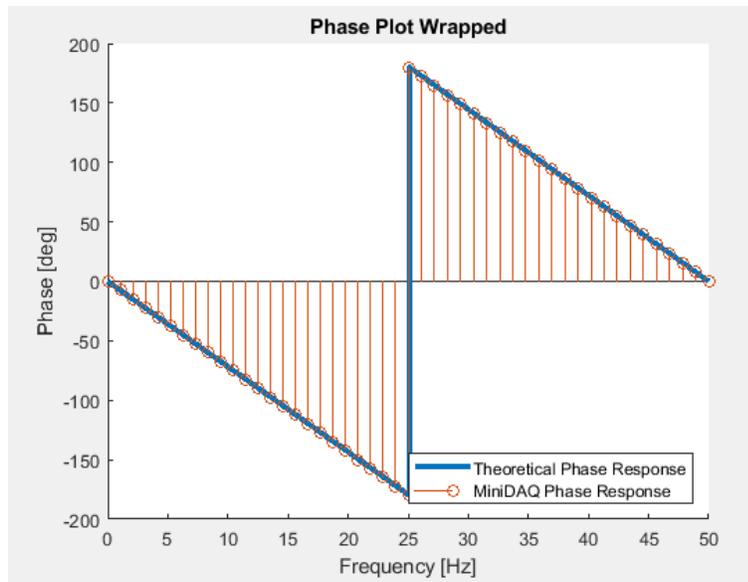


Figure 48: Phase Plot of Figure 48 wrapped to 180° and -180° to Determine Phase Change Points in Chirp Signal

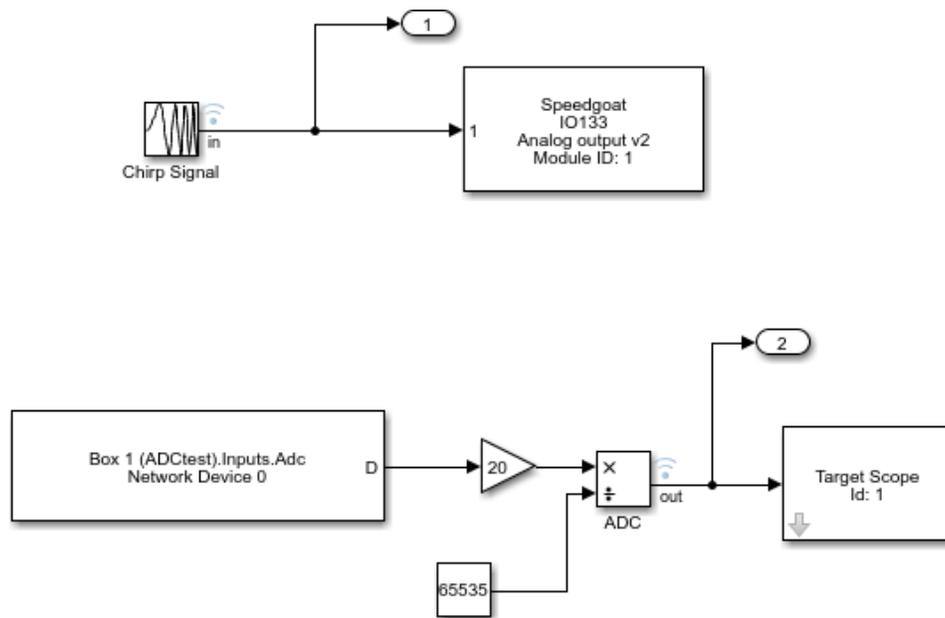


Figure 49: ADC Frequency Response Setup - Chirp

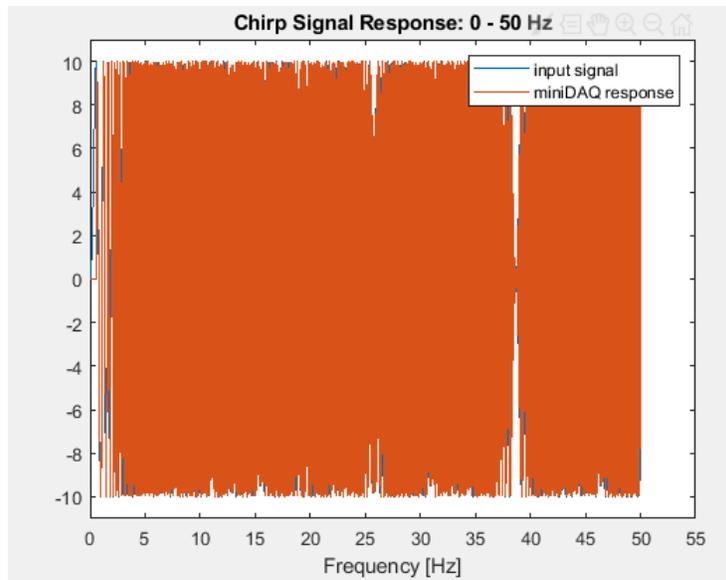


Figure 50: Full Chirp Signal from 0 to 50 Hz

7.5.1 Magnitude

Figure 51 displays the first feature of equivalent magnitudes between the input and output.

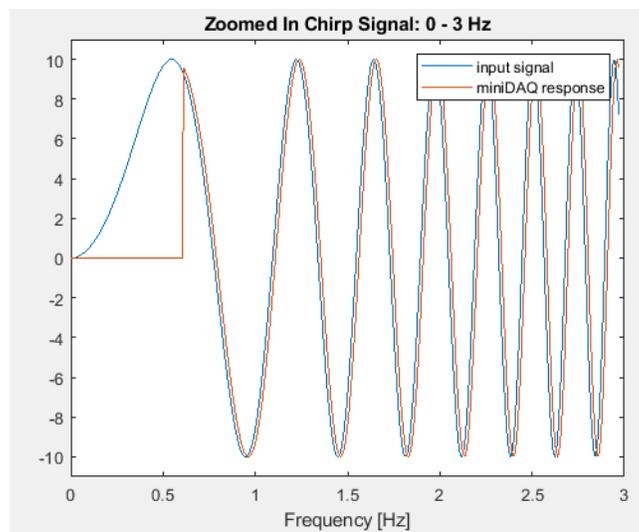


Figure 51: Zoomed in Chirp Signal from 0 to 3 Hz

7.5.2 Phase Change

To prove feature 2, the frequencies around 0 and 25 Hz are analyzed. Figure 52 displays almost 0° phase shift and an equivalent magnitude. Figure 53 shows the phase delay as being about 240° when we would expect 180° . Possible reasons for this value to not be exact are specifications when generating both excitation signals to be equivalent, but regardless, the results align well with what is expected from the PRBS response.

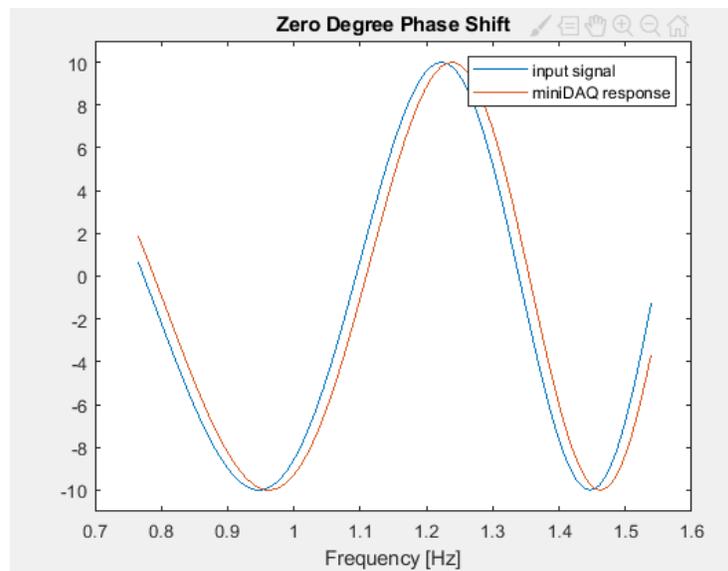


Figure 52: 0° Phase Shift

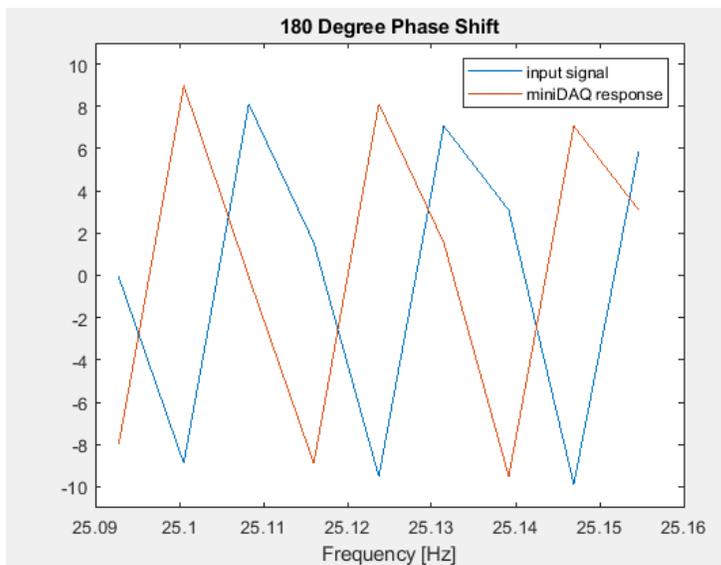


Figure 53: 180° Phase Shift

7.5.3 Sample Delay

At the time of testing, feature 3 indicated a 2 sample delay between input and output. The data taken, however, was plotted against the frequency axis, similar to the figures above. This cannot be used for determining sample delay as the response of the system to the chirp input must be plotted against time. Unfortunately when this was realized, part of the miniDAQ system had to be taken for the testing on the FOSWEC. This altered the systems' reaction to a step response, causing a less linear result seen in Figure 54. Due to this, retesting for sample delay was affected and now presented itself to be 4 samples.

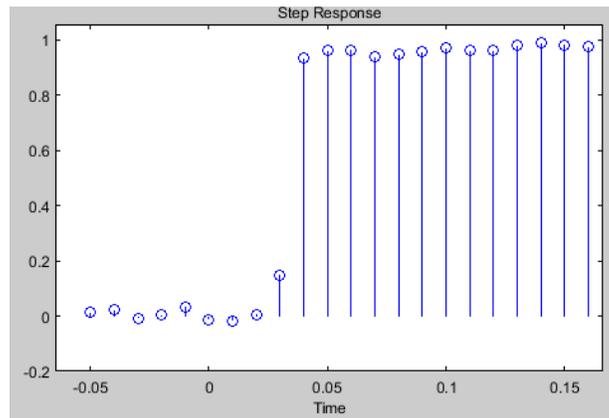


Figure 54: Step response indicating 4 sample delay

Still, feature 3 can be proven as, even with the modified system, the input and output should still contain this 4 sample delay. The figure below is a zoomed in portion of the chirp signal, with indicators to prove the delay against the time axis.

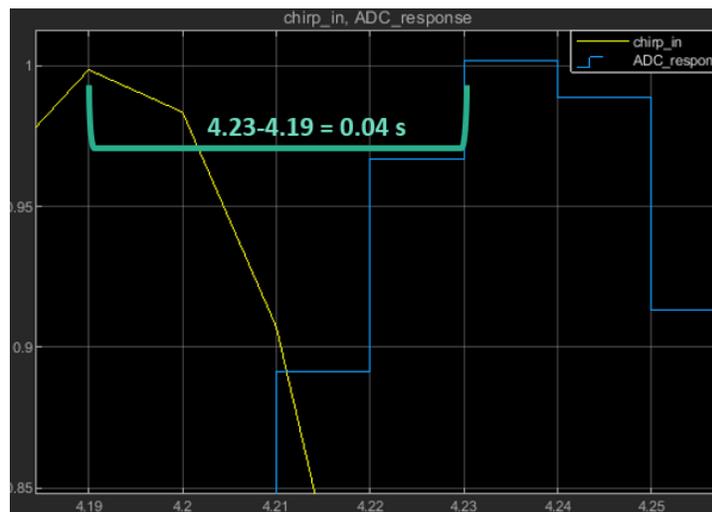


Figure 55: 4 sample delay (0.04 s) between input and output

8 Conclusion

The MiniDAQ project by SNL was intended to develop a DAQ for small-scale WEC testing as well as providing a general-purpose tool for any relevant application. This was motivated by the fact that a low cost, simple to use DAQ does not exist. Through design and testing, the MiniDAQ has confirmed that it has accomplished this goal. It has met the main technical requirements real-time operation, 100 Hz sampling frequency, differential inputs/outputs, and two communication interfaces. Along with that, the choices made when deciding on development boards and sensors were such that customization of the MiniDAQ was minimized, an important aspect of the design to allow any persons the ability to use this tool. Additionally, its small volume, light weight, and low cost make it an ideal device for any need in gathering data, whether that be a WEC, aid in autonomous operation for an automotive company, or weather reports for the beach...the MiniDAQ is there for you.

REFERENCES

- [1] Bacelli, Giorgio, et al. "Scoping and Requirements for a Data Acquisition System to Support Small-Scale Wave Energy Converter Testing." 22 Jan. 2019.
- [2] Rhinefrank, Ken, et al. "Numerical Analysis and Scaled High Resolution Tank Testing of a Novel Wave Energy Converter." *Journal of Off-shore Mechanics and Arctic Engineering*, vol. 135, no. 4, 2013, p. 041901., doi:10.1115/1.4024886.
- [3] Coe, Ryan, et al. Advanced WEC Dynamics & Controls. Oct. 2016, mhkdr.openei.org/files/151/AdvWecCntrl5 wtv01.pdf.
- [4] Arduino. (2019). *Arduino Mega 2560 Rev3* [online] Available at: <https://store.arduino.cc/usa/mega-2560-r3> [Accessed 19 June 2019].
- [5] AB&T. (2019). *EasyCAT* [online] Available at: <https://www.bausano.net/shop/en/home/1-arduino-ethercat.html> [Accessed 20 Mar. 2019].
- [6] Analog. (2019). *DC682A* [online] Available at: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/dc682a.html#eb-relatedhardware> [Accessed 20 Mar. 2019].
- [7] Analog. (2019). *DC2025A* [online] Available at: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/dc2025a-a.html#eb-relatedhardware> [Accessed 20 Mar. 2019].
- [8] Sparkfun. (2019). *Sparkfun Load Cell Amplifier - HX711* [online] Available at: <https://www.sparkfun.com/products/13879> [Accessed 20 Mar. 2019].
- [9] Amazon. (2019). *NulSom Inc. Ultra Compact RS232 to TTL Converter with Male DB9 (3.3V to 5V)* [online] Available at: <https://www.amazon.com/NulSom-Inc-Ultra-Compact-Converter/dp/B000PU2QJ4> [Accessed 20 Mar. 2019].

- [10] Digikey. (2019). *MAX1484EUB+* [online] Available at: <https://www.digikey.com/product-detail/en/maxim-integrated/MAX1484EUB/MAX1484EUB-ND/1474997> [Accessed 3 Aug. 2019]
- [11] NCD. (2019). *2-Channel 4-20 mA Current Loop Receiver 16-Bit ADS1115 I2C Mini Module* [online] Available at: <https://store.ncd.io/product/2-channel-4-20-ma-current-loop-receiver-16-bit-ads1115-i2c-mini-module/> [Accessed 3 Aug. 2019]
- [12] Digikey. (2019). *AMT232B-V* [online] Available at: https://www.digikey.com/product-detail/en/cui-inc/AMT232B-V/102-4485-ND/8031587?WT.zcid=ref_neda_dkc_buynow_cui&utm_source=ecia&utm_medium=aggregator&utm_campaign=cui
- [13] Digikey. (2019). *XSens Technologies BV MTI-20-2A8G4-DK* [online] Available at: <https://www.digikey.com/product-detail/en/xsens-technologies-bv/MTI-20-2A8G4-DK/MTI-20-2A8G4-DK-ND/6024013>
- [14] Mouser. (2019). *Yageo RC0201FR-078R2L* [online] Available at: <https://www.mouser.com/ProductDetail/Yageo/RC0201FR-078R2L?qs=sGAEpiMZZMtIubZbdhIBIOmt8RoBvXQUz8L7OkU3EE4%3D>
- [15] Amazon. (2019) *4 pcs Nichicon ES Muse Capacitors - 25V 220uf Audio* [online] Available at: <https://www.amazon.com/pcs-Nichicon-ES-MUSE-Capacitors/dp/B075K494YW>
- [16] Linear Technology. "LTC1857/LTC1858/LTC1859", LTC1859 datasheet. [online] Available at: <https://www.analog.com/media/en/technical-documentation/data-sheets/185789fb.pdf> [Accessed 20 Mar. 2019].
- [17] Linear Technology. "LTC2668", LTC2668 datasheet [online] Available at: <https://www.analog.com/media/en/technical-documentation/data-sheets/2668fa.pdf> [Accessed 20 Mar. 2019].

- [18] Digikey. (2019). *LMB Heeger Inc. TF-781 PL/UNPD* [online] Available at: <https://www.digikey.com/product-detail/en/TF-781+PL%2fUNPD/L105-ND/1936>
- [19] Paparella, Francesco, et al. "A Low-Cost Motion Capture System for Small-Scale Wave Energy Device Tank Testing." *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 3, 2018, p. 031008., doi:10.1115/1.4041608.
- [20] marineenergy.biz. (2019). *Oceantec WEC heads to bimep* [online] Available at: <https://marineenergy.biz/2016/10/12/oceantec-wec-heads-to-bimep/> [Revised 12 Oct. 2016] [Accessed 20 Mar. 2019].
- [21] en.wikipedia.org. (2019). *Pelamis Wave Energy Converter* [online] Available at: https://en.wikipedia.org/wiki/Pelamis_Wave_Energy_Converter [Revised 20 Feb. 2019] [Accessed 20 Mar. 2019].
- [22] Tethys (2019). *Wello Penguin at EMEC* [online] Available at: <https://tethys.pnnl.gov/annex-iv-sites/wello-penguin-emec> [Accessed 20 Mar. 2019].
- [23] coastalwiki.org (2019). *Wave Energy Converters* [online] Available at: http://www.coastalwiki.org/wiki/Wave_energy_converters [Revised 8 Nov. 2013] [Accessed 20 Mar. 2019].
- [24] Speedgoat (2019). *Speedgoat real-time simulation and testing* [online] Available at: <https://www.Speedgoat.com/> [Accessed 20 Mar. 2019].
- [25] mathworks.com (2019). *Simulink Real-Time* [online] Available at: <https://www.mathworks.com/products/simulink-real-time.html> [Accessed 20 Mar. 2019].
- [26] CUI. "AMT23 Modular Absolute Encoder", AMT23 Series datasheet. [online] Available at: <https://www.cui.com/product/resource/amt23.pdf> [Accessed 3 Aug. 2019].

- [27] Maxim Integrated. "Software-Selectable, Half-/Full-Duplex, Slew-Rate-Limited, 12Mbps, RS-485/RS-422 Transceivers in μ MAX Package", MAX1481-MAX1486 datasheet. [online] Available at: <https://datasheets.maximintegrated.com/en/ds/MAX1481-MAX1486.pdf> [Accessed 3 Aug. 2019].
- [28] *MT Manager User Manual*, Xsens, Culver City, CA, USA, 2019. [online] Available at: https://xsens.com/download/usermanual/ISM/MT_Manager_User_Manual.pdf [Accessed 1 Aug. 2019].
- [29] *MT Manager User Manual*, Xsens, Culver City, CA, USA, 2019. [online] Available at: https://documentation.xsens.com/mt_low-level_communication_protocol [Accessed 1 Aug. 2019].
- [30] Wu, Joseph. "ADS1115: Have Noise Issue and When and How to Calibration." Texas Instruments: E2E Support Forums, 16 Dec. 2018, e2e.ti.com/support/data-converters/f/73/t/758271?ADS1115-Have-noise-issue-and-when-and-how-to-calibration.
- [31] B.P., Jithin. "Calibrating the PSLab's Analog Features for Maximum Accuracy." Blog.fossasia.org, 7 Aug. 2017, blog.fossasia.org/calibrating-the-pslabs-analog-features-for-maximum-accuracy/.
- [32] Texas Instruments. "ADS111x Ultra-Small, Low-Power, I²C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator", ADS1113, ADS1114, ADS1115 datasheet. [online] Available at: <http://www.ti.com/lit/ds/symlink/ads1115.pdf> [Accessed 3 Aug. 2019]
- [33] Smoot, Jeff. "When Is an Absolute Encoder Right for Your Design?" CUI Inc, 20 Mar. 2018, www.cui.com/blog/when-is-an-absolute-encoder-right-for-your-design.

- [34] CUI Inc. "AMT23 Series Modular Absolute Encoder", AMT23 datasheet. [online] Available at: <https://www.cui.com/product/resource/amt23.pdf> [Accessed 1 Aug. 2019]
- [35] Ametek. (2019). *DC Gearmotor* [online] Available at: <https://prototypes.haydonkerk.com/ecatalog/brush-dc-motors/en/brush-dc-motors-GM8543S059-SP>
- [36] "Low Voltage." Wikipedia, Wikimedia Foundation, 24 Apr. 2019, en.wikipedia.org/wiki/Low_voltage.
- [37] Andy aka. (27 Feb. 2018). Using low pass filter for a differential ended receiver system? [Ans 1] Message posted to <https://electronics.stackexchange.com/questions/358696/using-low-pass-filter-for-a-differential-ended-receiver-system>
- [38] Vuojolainen, Jouni, et al. Comparison of Excitation Signals in Active Magnetic Bearing System Identification. 1st ed., vol. 38, Modeling, Identification and Control, 2017, pp. 1–11, Comparison of Excitation Signals in Active Magnetic Bearing System Identification.
- [39] Grauer, Jared, et al. Real-Time Frequency Response Estimation Using Joined-Wing SensorCraft Aeroelastic Wind-Tunnel Data. 2019, Real-Time Frequency Response Estimation Using Joined-Wing SensorCraft Aeroelastic Wind-Tunnel Data.
- [40] Têtu, Amélie. "Power Take-Off Systems for WECs." SpringerLink, Springer, Cham, 1 Jan. 1970, link.springer.com/chapter/10.1007/978-3-319-39889-1_8.