

# **Python Based Wind Turbine Data Acquisition Project (PCC V2)**

## **Final Report**

**Spencer Norton**

**Xuefeng Jiao**

**Summer 2022 – Fall 2022**



**Project Sponsor: Northern Arizona University Mechanical Engineering Department**

**Faculty Advisor, Sponsor Mentor, Instructor: Professor David Willy**

## **DISCLAIMER**

This report was prepared by students as part of a university course requirement. While considerable effort has been put into the project, it is not the work of licensed engineers and has not undergone the extensive verification that is common in the profession. The information, data, conclusions, and content of this report should not be relied on or utilized without thorough, independent testing and verification. University faculty members may have been associated with this project as advisors, sponsors, or course instructors, but as such they are not responsible for the accuracy of results or conclusions.

## EXECUTIVE SUMMARY

The python-based wind turbine data acquisition system project was brought about to improve on an existing design of a similar PCC (point of common coupling) data acquisition system. The team that was put together was two students, both Mechanical Engineering majors, to complete this project. The team met with the client, Professor David Willy, throughout the entirety of the project to ensure aligned views and to ensure the team was on track for completion. The beginning of the first semester, Summer 2022, the team began the project and established requirements and expectations with the client. These requirements were split up into customer and engineering requirements.

The team was required to meet all of the requirements and design and build a data acquisition system from the ground up. The engineering and customer requirements will be discussed in detail further on in this document, but in short, the team needed to improve on the previous PCC design by creating a smaller enclosure, utilize free software to analyze data, and improve the current sensing circuit. The team was able to satisfy these improvement requirements without ever seeing or reverse engineering any part of the previous design, but simply improved on the key aspects of the previous design through brainstorming and engineering knowledge gained through classes taken in the past.

The first semester was composed mostly of starting the project by establishing a firm understanding of what needed to be built. The acquisition system is complex, and needs to measure multiple values from many different sensors. Once the team had a good idea of what needed to be done, roles were established, and the work began. Xuefeng Jiao was established as the software engineer for this project, where his main goal was focused on writing Python code to analyze and plot data. Spencer Norton was established as the hardware engineer, where he was focused on creating, gathering, and building the enclosure and all the hardware parts to make the system work.

It was determined early on in the first semester that the project was going to be focused on the use of a Raspberry Pi, which is the central computer of the data acquisition system. This device was an ideal choice because of its ease of use, ability to run Python codes, and the flexibility it had with many other 3<sup>rd</sup> party devices. Using the Pi made writing python codes easier, as many libraries already existed for sensors, and these libraries only had to be adjusted to work together simultaneously. Once the Pi was decided to be used, the team moved on to determining the best kind of devices to use in order to accurately and correctly measure the values required for the project. These values were wind turbine voltage, wind turbine current, temperature, barometric pressure, and wind speed.

The team chose devices that were both very accurate, and worked well with the Raspberry Pi and Python language. As the first semester came to a close, the team had determined the sensors that were needed, and other hardware that would be needed to complete the project. The beginning of the second semester began with ordering parts and testing began to ensure everything worked. The Python languages were started as well, as these were imperative to the success of the project. The team determined a suitable electrical enclosure to house all the components, and found a good ADC (analog to digital) converter to read voltage. The python codes were created using nearly all the sensors at the same time. Code provided by the manufacturer of the ADC was used and modified to work as needed. The code was then modified to record data to a csv file, and also plot live data using matplotlib libraries.

By writing the Python code and physically building a system that could handle the requirements of the project, the team was able to successfully create a wind turbine data acquisition system that can measure voltage within a specific range, read current within two specific ranges, measure barometric pressure and temperature, and plot and record all the data all at the same time. Each requirement was met and in some cases exceeded by the hard work of the wind turbine data acquisition team. This document discusses in detail the process of completing this project.

## **ACKNOWLEDGEMENTS**

This project was completed with the help of Northern Arizona University, who is the sponsor of this project. Professor David Willy is the client for this project and has helped immensely from start to finish in providing knowledge and help in execution of engineering fundamentals and knowledge in the professional world when applicable to complete this project.

# TABLE OF CONTENTS

## Contents

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| DISCLAIMER                                                              | 2  |
| EXECUTIVE SUMMARY                                                       | 3  |
| ACKNOWLEDGEMENTS                                                        | 4  |
| TABLE OF CONTENTS                                                       | 5  |
| 1 BACKGROUND                                                            | 2  |
| 1.1 Introduction                                                        | 2  |
| 1.2 Project Description                                                 | 2  |
| 2 REQUIREMENTS                                                          | 3  |
| 2.1 Customer Requirements (CRs)                                         | 3  |
| 2.2 Engineering Requirements (ERs)                                      | 3  |
| 2.3 Functional Decomposition                                            | 4  |
| 2.3.1 Black Box Model                                                   | 4  |
| 2.3.2 Functional Model/Work-Process Diagram/Hierarchical Task Analysis  | 4  |
| 2.4 House of Quality (HoQ)                                              | 5  |
| 2.5 Standards, Codes, and Regulations                                   | 6  |
| 3 DESIGN SPACE RESEARCH                                                 | 8  |
| 3.1 Literature Review                                                   | 8  |
| 3.2 Benchmarking                                                        | 8  |
| 3.2.1 System Level Benchmarking                                         | 8  |
| 3.2.1.1 Existing Design #1: NETDAQ System                               | 9  |
| 3.2.1.2 Existing Design #2: WIFI Weather Station WS-2902C               | 9  |
| 3.2.1.3 Existing Design #3: Raspberry Pi Weather Station                | 9  |
| 3.2.2 Subsystem Level Benchmarking                                      | 10 |
| 3.2.2.1 Subsystem #1: Analog to Digital Converters                      | 10 |
| 3.2.2.1.1 Existing Design #1: U3 LabJack Converter.....                 | 10 |
| 3.2.2.1.2 Existing Design #2: ADS1015.....                              | 10 |
| 3.2.2.1.3 Existing Design #3: Pi-16AC Analog-Digital Converter Hat..... | 10 |
| 3.2.2.2 Subsystem #2: Anemometers                                       | 10 |
| 3.2.2.2.1 Existing Design #1: Post Connector Cup Anemometers.....       | 11 |
| 3.2.2.2.2 Existing Design #2: Vane Anemometer.....                      | 11 |
| 3.2.2.2.3 Existing Design #3: Analog Output Anemometer.....             | 11 |
| 3.2.2.3 Subsystem #3: Programming Layout Methods                        | 11 |
| 3.2.2.3.1 Existing Design #1: Numeric.....                              | 11 |
| 3.2.2.3.2 Existing Design #2: Case Structure in LabView.....            | 12 |
| 3.2.2.3.3 Existing Design #3: Raspberry Pi While Loops.....             | 13 |
| 4 CONCEPT GENERATION                                                    | 15 |
| 4.1 Full System Concepts                                                | 15 |
| 4.1.1 Full System Design #1: All-in-One Weather Station                 | 15 |
| 4.1.2 Full System Design #2: Touch Screen Weather Station               | 16 |
| 4.1.3 Full System Design #3: Standalone Weather Station                 | 16 |
| 4.2 Subsystem Concepts                                                  | 17 |

|         |                                                                 |    |
|---------|-----------------------------------------------------------------|----|
| 4.2.1   | <i>Subsystem #1: Boom Arm</i>                                   | 17 |
| 4.2.1.1 | Design #1: Double Measurement Arm                               | 17 |
| 4.2.1.2 | Design #2: Triple Measurement Arm                               | 17 |
| 4.2.1.3 | Design #3: Single Measurement Arm                               | 17 |
| 4.2.2   | <i>Subsystem #2: Enclosure Placement</i>                        | 17 |
| 4.2.2.1 | Design #1: Tall Enclosure                                       | 17 |
| 4.2.2.2 | Design #2: Flat Enclosure                                       | 18 |
| 4.2.2.3 | Design #3: Large enclosure with monitor                         | 18 |
| 4.2.3   | <i>Subsystem #3: Visual Layout</i>                              | 18 |
| 4.2.3.1 | Design #1: Tile layout                                          | 18 |
| 4.2.3.2 | Design #2: Tab layout                                           | 18 |
| 4.2.3.3 | Design #3: Touch screen swipe layout                            | 18 |
| 5       | DESIGN SELECTED – First Semester                                | 19 |
| 5.1     | Design Description                                              | 19 |
| 5.2     | Implementation Plan                                             | 20 |
| 6       | Project Management – Second Semester                            | 22 |
| 6.1     | Gantt Chart                                                     | 22 |
| 6.2     | Purchasing Plan                                                 | 22 |
| 6.3     | Manufacturing Plan                                              | 23 |
| 6.4     | Bonus/Substitution Sections – as needed                         | 25 |
| 7       | Final Hardware                                                  | 26 |
| 7.1     | Final Hardware Images and Descriptions                          | 26 |
| 7.2     | Design Changes in Second Semester                               | 26 |
| 8       | Testing                                                         | 28 |
| 8.1     | Testing Plan                                                    | 28 |
| 8.2     | Testing Results                                                 | 28 |
| 9       | RISK ANALYSIS AND MITIGATION                                    | 32 |
| 9.1     | Potential Failures Identified First Semester                    | 32 |
| 9.2     | Potential Failures Identified This Semester and Risk Mitigation | 32 |
| 10      | LOOKING FORWARD                                                 | 34 |
| 10.1    | Future Testing Procedures                                       | 34 |
| 10.2    | Future Iterations                                               | 34 |
| 11      | CONCLUSIONS                                                     | 35 |
| 11.1    | Reflection                                                      | 35 |
| 11.2    | Resource Wishlist                                               | 35 |
| 11.3    | Project Applicability                                           | 35 |
| 12      | REFERENCES                                                      | 37 |
| 13      | APPENDICES                                                      | 39 |
| 13.1    | Appendix A: House of Quality or Functional Decomposition        | 39 |
| 13.2    | Appendix B: Full Bill of Materials                              | 40 |
| 13.3    | Appendix C: FMEA                                                | 41 |
| 13.4    | Appendix D: Python Code                                         | 42 |
| 13.4.1  | Main Python File                                                | 42 |
| 13.4.2  | ADS1263                                                         | 43 |
| 13.4.3  | Config                                                          | 44 |
| 13.4.4  | Plotting                                                        | 45 |



# 1 BACKGROUND

## 1.1 Introduction

This project is called the Python based wind turbine data acquisition system, which was originally started to improve on an existing PCC (point of common coupling) data acquisition system. The team created a new wind turbine data acquisition system by the use of a Raspberry Pi and Python code to read, record, and plot live data coming from multiple sensors. The team was required to measure wind turbine output voltage, wind turbine current, temperature, pressure, and wind speed. This was achieved by using an analog to digital converter (ADC) to read voltage, current, and analog wind speed, a temperature probe was used to measure ambient temperature, a pressure sensing board was used to read barometric pressure, and a cup anemometer was used to measure wind speed. By reading all of these values, the team will have created a system that will in the future be able to read voltage and correlate this to wind speed to determine power curves of the wind turbine. With this data, it will be possible to determine how different wind speeds can output certain voltages to aid in a clean energy environment.

Through the use of Python code and many libraries provided by manufacturers, the team was able to modify and write new code to read from each sensor simultaneously. As these values were being read, the Raspberry Pi was able to record the data to a csv file, and plot the live data on graphs per the engineering and customer requirements established by the team and the client. This report discusses the project from beginning to end, including the entire design process and the final design built by the team.

## 1.2 Project Description

Based on the project description provided by the client, this project is always in need of constant improvement to find ways to collect data from the point of common coupling (PCC) between the turbine and the load to assess the performance of the turbine. This project will design and build a PCC data collection system out of inexpensive hardware and relatively free software.

Following is the original project description provided by the sponsor.

“The Collegiate Wind Competition is a project that we run every year for the fall-spring sequence. That project is always in need of ways to collect data from the point of common coupling (PCC) between the turbine and the load to assess the performance of the turbine. The current PCC system used by NAU is large and requires expensive software such as LabVIEW to run and an extra computer to collect the data. This project will design and build a PCC data collection system out of inexpensive hardware and relatively free software. Suggested hardware would be a raspberry pi and sensors for voltage, current, wind speed, temperature, rotor RPM, and pressure. Calculations that could be needed might be power, density perhaps. Suggested software would be python or any visual programming language comparable to LabVIEW but not as expensive.” [1]

## 2 REQUIREMENTS

The requirements of this project were determined through various staff meetings with the client and team members. Table 1 and Table 2 show customer requirements and engineering requirements with their associated units and values required by the client. Outside of these specific requirements, the team was required to use a Raspberry Pi to act as the main computer for this project. The team was also required to utilize free software to analyze and plot the data. This was something that needed to be improved from the previous PCC design, which used LabVIEW to analyze data, which has a hefty cost associated with it. The team was also required to build the system independent of an internet connection.

### 2.1 Customer Requirements (CRs)

**Table 1:** Customer Requirements

| Customer Requirement                                 | Description                                                                           | Units     |
|------------------------------------------------------|---------------------------------------------------------------------------------------|-----------|
| <b>CR1: Small Enclosure</b>                          | Must be under 6x10x12 inches                                                          | Inches    |
| <b>CR2: Display Gauges and Graphs</b>                | For ease of use and easy-to-read data                                                 | NA        |
| <b>CR3: Display Live Data, Track Historical Data</b> | Ensures ease of use, must be able to analyze data in external software                | NA        |
| <b>CR4: Downloadable to excel file</b>               | Allows the user to take data to analyze in external software                          | NA        |
| <b>CR5: Cost Within Budget</b>                       | Must create a system within the budget of \$1500                                      | US Dollar |
| <b>CR6: Durable and Robust Design</b>                | Must create a system that is durable and robust, including plastics, wood, and metal. | NA        |
| <b>CR7: Reliable Design</b>                          | Must Create a system that is reliable and functional                                  | NA        |
| <b>CR8: Safe to Operate</b>                          | Must create a system that is safe to operate for the user                             | NA        |

### 2.2 Engineering Requirements (ERs)

**Table 2:** Engineering Requirements

| Engineering Requirement         | Description                                                                                         | Units                     |
|---------------------------------|-----------------------------------------------------------------------------------------------------|---------------------------|
| <b>ER1: Measure Wind Speed</b>  | Measure Wind speed from 0-25 m/s and 0-8000 rpm                                                     | m/s, rotations per minute |
| <b>ER2: Measure Temperature</b> | Measure typical Flagstaff Arizona Ambient conditions, -25 to 100 degrees Fahrenheit                 | F                         |
| <b>ER3: Measure Pressure</b>    | Measure Atmospheric Pressure for typical Flagstaff Arizona conditions, 10-40 inHg, or 0.34-1.35 bar | Bar                       |
| <b>ER4: Turbine Voltage</b>     | Measure the range of voltage from the wind turbine, 0-48V, resolution range of <.05V                | Volts                     |

|                                        |                                                                                                               |      |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------|------|
| <b>ER5: Current</b>                    | Measure the Current coming from the wind turbine, in two ranges, 0-10 A and 0-30A, resolution range of <.01 A | Amps |
| <b>ER6: Reliability and Durability</b> | Design must be reliable and durable for everyday use                                                          | NA   |

## 2.3 Functional Decomposition

There are two main inputs in the Function Model: power input and programming input. The power input is used to provide stable power for the Raspberry Pi. The programming input is used to manipulate the sensors to acquire data from the wind turbine through the built-in function of Raspberry Pi using Python. The remote-control portion of the functional model is kept in the functional model as a feature the team would like to incorporate to allow remote access to the data acquisition system at a later date if needed.

### 2.3.1 Black Box Model

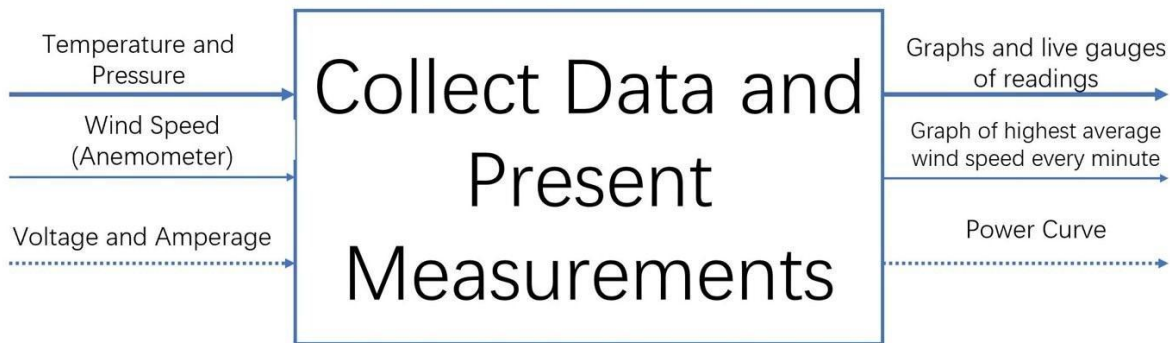


Figure 1: Black Box Model

The Black Box Model contains three inputs and three outputs. Three inputs are made up of data that the team needs to measure from the wind turbine and the environment. Three outputs are the results for the user, including the reading of graphs and multiple curves. The Black Box Model builds predictive models that exist in computer mode to help the team modify and clarify the input and output. This model helps the team follow the customer requirements and engineering requirements. There are multiple sensors that the team needs to use to collect data from the environment and the black box model visualizes the input as types of data collected from the environment.

### 2.3.2 Functional Model/Work-Process Diagram/Hierarchical Task Analysis

This Functional decomposition helps the team understand how each input and output is tied to the system as a whole. The "sensors" Section of the decomposition is an encompassing part of the model, which will include, temperature, pressure, humidity, wind speed, wind turbine amperage, and wind turbine voltage measuring devices. Being able to see how the system is broken down helps the team members in creating designs and in the future, wiring the system together, because the simplified version of the system is much easier to understand.

The functional model becomes more and more important as the team moves forward to the project as it clearly shows all the design requirements and goals. This model also guides the team to follow all the design requirements, including customer requirements and engineering requirements.

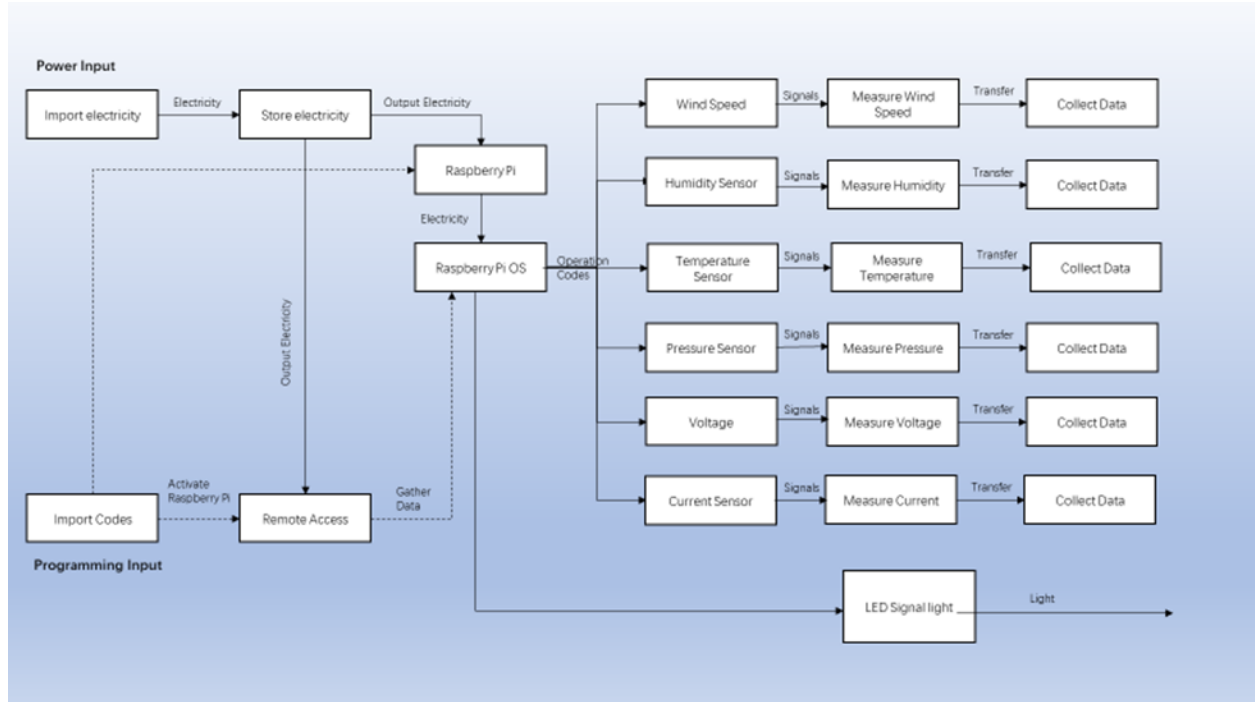


Figure 2: Functional Decomposition

## 2.4 House of Quality (HoQ)

In order to combine Engineering Requirements with Customer Needs and evaluate each of them to calculate the importance of each term of technical requirements, the team also generates a preliminary House of Quality. This model was built before second presentation, which means that the broad range of requirements and weights helped the team to have a basic and overall understanding of all potential design requirements. Finally, based on the results of technical importance, the team eliminates those unimportant requirements and chooses critical requirements into the future design process.

|                                          |                  | Engineering Requirements      |                                |                                             |                                |                                      |                                               |
|------------------------------------------|------------------|-------------------------------|--------------------------------|---------------------------------------------|--------------------------------|--------------------------------------|-----------------------------------------------|
| Customer Needs                           | Customer Weights | Measure Wind speeds (0-25m/s) | Measure Temperature (25-100 F) | Measure Barometric Pressure (0.34-1.35 bar) | Measure Current (0-30A, 0-10A) | Measure Wind Turbine Voltage (0-48V) | Meet Specific Resolutions for Measured Values |
| Display live data, track historical data | 5                |                               |                                |                                             |                                |                                      |                                               |
| Display gauges and graphs                | 5                |                               |                                | 9                                           | 9                              |                                      |                                               |
| Small enclosure                          | 3                | 1                             |                                |                                             |                                | 3                                    |                                               |
| Download historical data to excel file   | 4                |                               | 9                              | 9                                           |                                |                                      |                                               |
| Cost within budget                       | 3                | 9                             | 1                              |                                             | 9                              |                                      | 9                                             |
| Durable and robust design                | 3                | 9                             |                                |                                             | 1                              |                                      |                                               |
| Reliable design                          | 3                | 3                             |                                |                                             | 3                              |                                      | 1                                             |
| Safe to opera                            | 2                |                               |                                |                                             |                                |                                      | 9                                             |
| Technical Requirement Units              |                  | m/s                           | F                              | bar                                         | A                              | V                                    | NA                                            |
| Technical Requirement Targets            |                  | 0-25                          | 25-100                         | 0.34-1.35                                   | 0-30, 0-10                     | 0-48                                 | NA                                            |
| Absolute Technical Importance            |                  | 66                            | 60                             | 36                                          | 39                             | 99                                   | 48                                            |
| Relative Technical Importance            |                  | 3                             | 4                              | 9                                           | 8                              | 1                                    | 5                                             |

Figure 3: Updated QFD Model

## 2.5 Standards, Codes, and Regulations

This project has a few standards and codes that were determined needed to be followed in order to create a safe, reliable, and functional system. The main standard that is needed for this project is the standard for the electrical enclosure. This enclosure houses all the components of the data acquisition system and has multiple interface ports added into the side of the enclosure. These ports have been sealed with rubber gaskets to maintain a water tight seal as much as possible. Table 3 shows the standard for the enclosure and the other standards the team has applied to the project. All CAD drawings were made to meet GD&T standards, so all the drawings for the simple CAD models of the project meet these standards.

The next standard is for wire gauge, which is important for this project because of the high amount of electricity being used within the enclosure. The current sensing circuit is required to measure up to 30 amps, so per AWG standards, 10 gauge wire is used [2]. NIST standards apply to the temperature sensor used for this project, which in order to meet the standard, the probe needs to be constructed of a certain material in order to maintain a certain accuracy. The ITS-90 standard is applied to temperatures sensors that are calibrated using high accuracy standard temperature generating sources, which unfortunately the team did not have. The team tried to the best of their ability to calibrate the sensor using resources on hand. These calibration tests can be seen in the testing section of this document.

**Table 3:** Standards of Practice as Applied to this Project

| <b>Standard Number or Code</b> | <b>Title of Standard</b> | <b>How it applies to Project</b>                                                                                       |
|--------------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------|
| NEMA                           | NEMA 4X                  | Ensures enclosure protects inside components against dust or rain                                                      |
| GD&T                           | GD&T Y14.5               | Standard of all CAD drawings for this project                                                                          |
| AWG                            | AWG                      | This standard will ensure the correct gauge of wire is used for the circuits                                           |
| NIST                           | ITS-90                   | These standards will apply for each sensor used in the system: temperature probe, pressure sensor, anemometer, and ADC |

## 3 DESIGN SPACE RESEARCH

This section contains three parts: Literature Review, Benchmarking, and Functional Decomposition.

Literature Review describes what sources were used for benchmarking and design research.

Benchmarking involves on-site visits to organizations, observation, and interviews with employees to see how others have approached this type of design problem. Functional Decomposition introduces the main functions of the project and elaborates on the functional decomposition process.

### 3.1 Literature Review

This section describes what sources were used for benchmarking and design research. This is finished by examining similar systems, literature review, and web searches. There are five sources for each team member, and ten in total. Each member of the team was assigned a technical task based on their skillset, and their literature review sources are based on this designated title. Spencer was given the designation to work with hardware aspects of the project, while Xuefeng was designated as the Software specialist for this project. The following sources were used as literature reviews for this project.

*Book: Raspberry Pi Beginner's Guide*

This book will help the team to learn how to use the new mini-computer, such as setting up the operation system, programming in Python.

*Book: Python Basics*

This book is a self-learning introduction for the beginners of Python, which provides the team with basic functions of programming.

*Book: Data Acquisition Using LabVIEW*

The goal of this book is to transform physical phenomena into computer-acceptable data using a truly object-oriented language.

*Article: Raspberry PI Based Integrated Autonomous Vehicle Using LabVIEW*

This is a recent study that combines Raspberry PI, Python, and LabVIEW together and apply them into practice.

### 3.2 Benchmarking

For the team to properly design a wind turbine data acquisition system, many different existing designs had to be researched extensively to find advantages and disadvantages of these designs. The following sections go in to detail which products were benchmarked, including a NETDAQ system, a WIFI weather station, and a Raspberry Pi weather station. Then the team benchmarked sub systems to research parts of the system that had the most variability of change to effect the system as a whole. These included analog to digital converters, anemometers, and programming methods.

#### 1.1.1 System Level Benchmarking

This section of the benchmarking research includes systems that are similar to the weather station system as a whole. The purpose of the weather station is to receive data from numerous sensors and interpret that data to create metrics that show power curves, and other graphs to plot data. The following benchmarking items were seen as valuable research to do in order to find features that could benefit or not benefit the

future designed system.

#### **1.1.1.1      *Existing Design #1: NETDAQ System***

This system level benchmarking was done as a high level data collection comparison. The overall goal of the system being designed is to collect data and place that data in an excel sheet. The NETDAQ data acquisition system is a very outdated validated data collection system that can be used to collect data from numerous ports that can be collecting temperature, pressure, or other values. The research done with this system was for the place of employment of one of the team members, where the system was used to collect temperature measurements from multiple locations within a refrigerator. The system itself worked great at collecting the temperature data, with reasonable accuracy.

The main point of concern with this outdated system is that it runs on a Windows 7 operating system and an older version of excel, which is of course not ideal for the new design system. The system was chosen as a benchmark because of its ability to constantly record data to an actual excel sheet. Once the excel sheet was filled to the programmed limit, the sheet could be downloaded and used in other programs to utilize and analyze the data. This is something that the future design of the team's data acquisition system needs to incorporate. In the future, the NETDAQ will hopefully be researched more to see how the operating system interfaces with the user to receive inputs such as measurement rates, where to store data, how long to collect data, and other inputs like these.

#### **1.1.1.2      *Existing Design #2: WIFI Weather Station WS-2902C***

WIFI weather stations are one of the most popular weather station choices for families and anyone not wanting to really get into intense weather systems. Most of these wireless weather stations include a cup anemometer, potentially a wind direction sensor, and other sensors for temperature, humidity, and pressure. The reason this type of system was researched as a benchmark is because of the ease of use of the systems. These easy to acquire, WIFI stations are very easy to set up and very easy to use. For this reason, we wanted to look into what really makes them so user friendly.

After looking into many different types of stations, it was easy to see that most of them have a user interface that is laid out in a tile format. The touch screen display has all different types of tiles that each show different information such as pressure, temperature, wind speed and direction, weather to expect for the day, but none of them ever showed voltage or current readings from the anemometer [3]. This is not a big issue as most families wanting a weather station will not want to measure these things, but being able to look at the user interface gave us some great ideas of what to do when creating a visual user interface to display data from the various sensors.

#### **1.1.1.3      *Existing Design #3: Raspberry Pi Weather Station***

This particular weather station was created by a team for the Raspberry Pi company. The entire creation and operation of the weather station can be found online at the Raspberry Pi Foundation website [4]. This article was very interesting to read because not only does it go in depth for each component (i.e. anemometer, sense hat, temperature, and humidity), but it describes the Python code needed to make this system work properly.

We really wanted to use this system as a benchmark for the future design because it was so similar to what we need to create. Because this system is so simplified, it is great to see how different, simpler components work in similar ways. The system being researched utilized a cup anemometer that has an ethernet cable on one end instead of power pole connectors, which may not be the best solution for our design due to the need for multiple channels receiving input from our anemometer, but it was still interesting to see how this type of anemometer could work.

For humidity and temperature, the Pi foundation used external temperature and humidity sensing boards to measure these different values. They used the BME280 to measure pressure, and the DS18B20 to measure temperature. These board can interface well with a Raspberry Pi, and they will be looked into in the future as potential devices if the Raspberry Pi sense hat does not work for our design.

## **1.1.2 Subsystem Level Benchmarking**

This Chapter will discuss benchmarking for the subsystems of the wind turbine data acquisition system. It was decided by the team that the most important components of the system would be benchmarked. Not only were the most important components benchmarked, but the sub systems that had the most variability in choice that would affect the system.

### **1.1.2.1 Subsystem #1: Analog to Digital Converters**

Analog to digital converters will be crucial to the correct operation and longevity of the data acquisition system. In order for the cup anemometer to work correctly with the Raspberry Pi, an analog to digital converter has to be used to convert the electrical signals to readable digital signals. Once these analog signals are converted to digital, the Raspberry Pi can take the signals from the cup anemometer connected to multiple channels and the Python code will interpret that data to wind speed, voltage, and current.

#### **1.1.2.1.1 Existing Design #1: U3 LabJack Converter**

The U3-H3 Labjack Analog to digital converter is a great option to convert the voltage coming from the anemometer. This converter claims to have high voltage inputs, 12 flexible inputs and outputs, and 4 dedicated digital inputs and outputs [5]. Having this many inputs and outputs would not necessarily be needed for the future design phases or the final design, but it would still be nice to have them if more inputs were decided to be added.

#### **1.1.2.1.2 Existing Design #2: ADS1015**

The ADS1015 is an analog to digital converter that claims to work well with Raspberry Pi computers. The converter only has 4 channels, and comes in 12 or 16-bit models [6]. Because this converter works so well with Raspberry Pi computers, there is already libraries created that can be easily installed with Python code. This converter would be an easy to use, and very small solution to amplifying the voltage coming from the cup anemometer. There are no known downsides to using this converter for the data acquisition system.

#### **1.1.2.1.3 Existing Design #3: Pi-16AC Analog-Digital Converter Hat**

Raspberry Pi's are very customizable computers, and as such, hats are a very cool option in customizability. These hats are stackable, meaning there can be multiple hats on one Pi board, so the temperature and humidity hat can still function, and another can be added on top. This particular hat is made to convert analog to digital electricity, and would be very simple to use with the Raspberry Pi. This converter hat has 16 single ended, or 8 different analog to digital conversion ports, and is a 16-bit model [7]. This converter would be easy to use and have low power consumption, but on the downside, the heat generated by the board could effect the sense hat that is simultaneously measuring temperature and humidity. Because these hats are so inexpensive, the team will likely try one of the converter hats as part of the prototyping process.

### **1.1.2.2 Subsystem #2: Anemometers**

The team wanted to look into different types of anemometers to see which one would likely work best for the design. There are a number of different types of anemometers, the sponsor of the project provided two cup anemometers for the system, but did not require the use of them over other measurement devices.

#### **1.1.2.2.1 Existing Design #1: Post Connector Cup Anemometers**

The post connector style anemometer will likely be the best device for the future design of the weather station. This method of connection allows for the anemometer to be connected to a shunt resistor to allow the Raspberry Pi to measure current. Without the usage of a post style connecting anemometer, there would have to be an additional wind device to measure current and voltage separate from the cup anemometer. Having two different wind speed sensors would bring about difficulties in separation of streamlines so that one device is not interfering with the other.

#### **1.1.2.2.2 Existing Design #2: Vane Anemometer**

A Vane style anemometer was researched as another option to measure wind speed for this project. Due to the design of most vane anemometers, they can measure wind speed, temperature, and humidity all in one package. This would be beneficial to the design of the system because it would eliminate the need for the temperature and humidity sensors on the Raspberry Pi sense hat board. Using a vane anemometer and eliminating the sense hat would still create a need for a pressure sensor, and voltage/ current measurements. Vane anemometers are also difficult to interface with systems other than the velocity meters that they are made to work with. This made the vane anemometer become a likely component that would not work with the future design.

#### **1.1.2.2.3 Existing Design #3: Analog Output Anemometer**

There are multiple types of anemometers, and the team wanted to look into an analog output anemometer as a potential device to measure wind speed. Using an analog output device would allow the team to integrate the voltage output into the ADC and read the voltage similarly to how the turbine voltage output would be measured. This would simplify the amount of devices needed to read measurements. Being able to read voltage and convert to wind speed would greatly simplify the system as opposed to using a digital output anemometer that would require another ADC all together.

### **1.1.2.3 Subsystem #3: Programming Layout Methods**

LabVIEW is a system-design platform and development environment for a visual programming language from National Instruments, which uses the graphical language. For university students who have a programming basic, it is easier for the team to use Python. The following section will compare graphical language with Python language and reach the conclusion that the built-in programming function of Python in the Raspberry Pi works better than the graphical language of LabVIEW. There are four important data types of LabVIEW that the team will use in the design process: Numeric, Boolean, String, and project.

#### **1.1.2.3.1 Existing Design #1: Numeric**

Take the Numeric as example. Each element of the component is represented by a picture. This visual programming language makes the codes look more straightforward, but also increases the difficulty for coders if they do not have a solid understanding of these instruments. Figure 4 shows a simple final result of the programming. It is easy to read by users but involve a lot of electrical engineering knowledge for people to write the code. Considering the expensive cost of this software, the team decides to give up LabVIEW in the future design process.

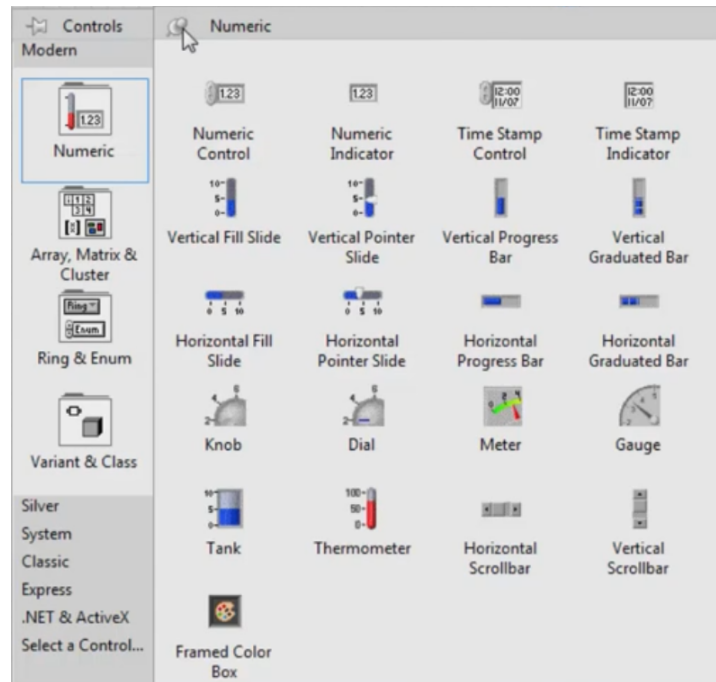


Figure 3: Front Panel

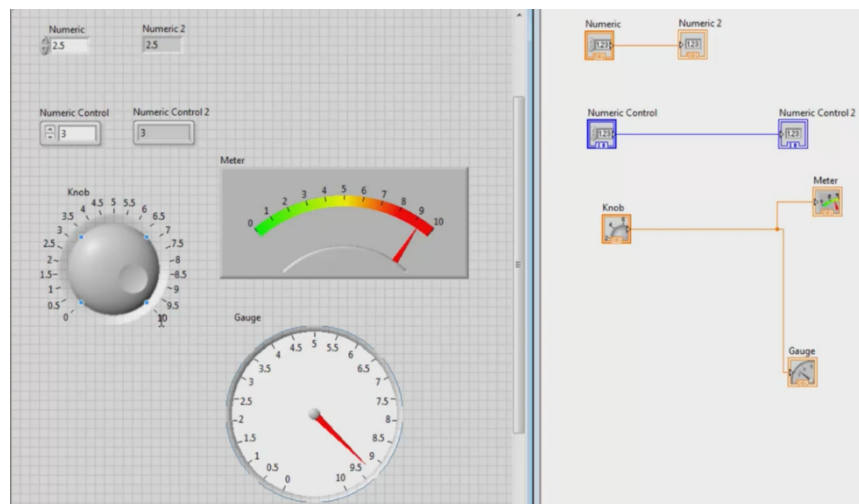


Figure 4: Block Diagram

#### 1.1.2.3.2 Existing Design #2: Case Structure in LabView

The enclosure system in the LabVIEW is called Case Structure. Take the thermometer as an example. For different amounts of temperature, the terms of low, high, or very high should be shown in the indicator. The first step is to insert a Knob, then change the amount from 1 to 100. The second step is to insert the Thermometer. The third step is to insert the Numeric Indicator. The fourth step is to insert the String Indicator. The last step is to open the Case Structure in the Block Diagram window, as shown in Figure 5.

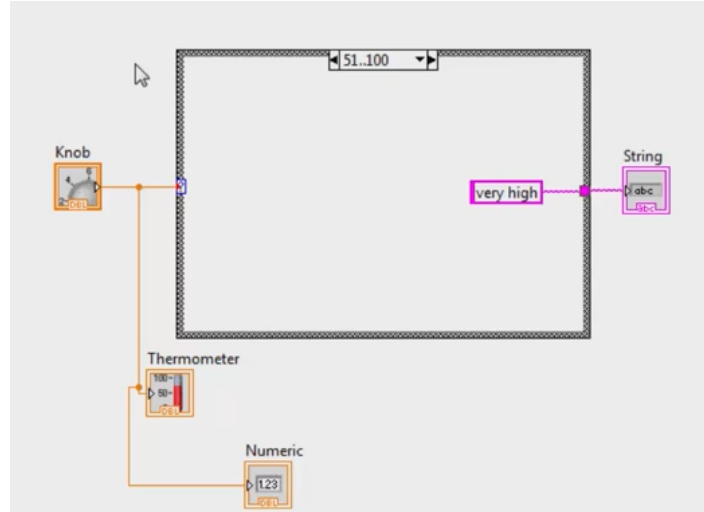


Figure 5: Case Structure

#### 1.1.2.3.3 Existing Design #3: Raspberry Pi While Loops

After implementing this process, it is true that the visual programming in LabVIEW is more straightforward than a Python code, but the problem is also obvious. There are too many similar components in the LabVIEW, students who lack relevant Electrical Engineering knowledge are easy to make mistakes when selecting the electrical component for the circuit.

While loop and For loop are two important programming functions in the design process of LabVIEW. In order to create a Loop, click the Structure and choose the corresponding Loop. Then the Structure containing Loop Conditions and Loop Iteration will be set up. The final step is to add electrical components to the structure and run the Loop to start the simulation.

The team then compares the LabVIEW programming of the Loop with the built-in Python programming of Raspberry Pi. The advantage of using Python to replace LabVIEW is that there are already plenty of Python codes on the internet, together with many projects based on the Raspberry Pi. In this case, the team can directly study these projects and use their codes. But there are less contents and discussion about LabVIEW on the internet because its programming language is only used by professional engineers. Therefore, be consistent with the Raspberry Pi will save the team members a lot of time and improve the accuracy of results.

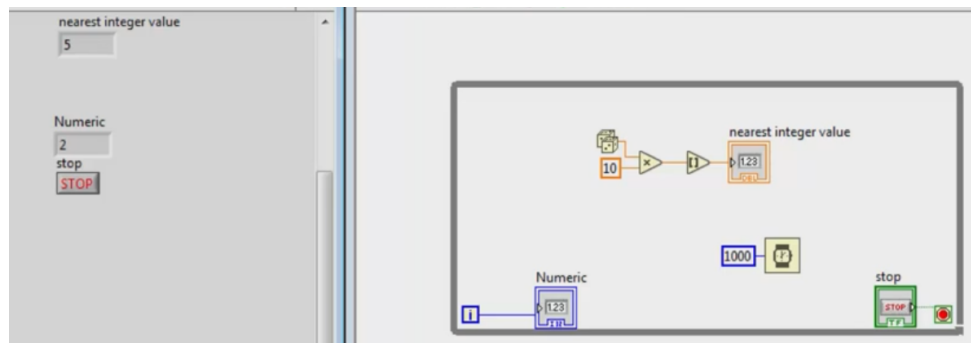


Figure 6: While Loop

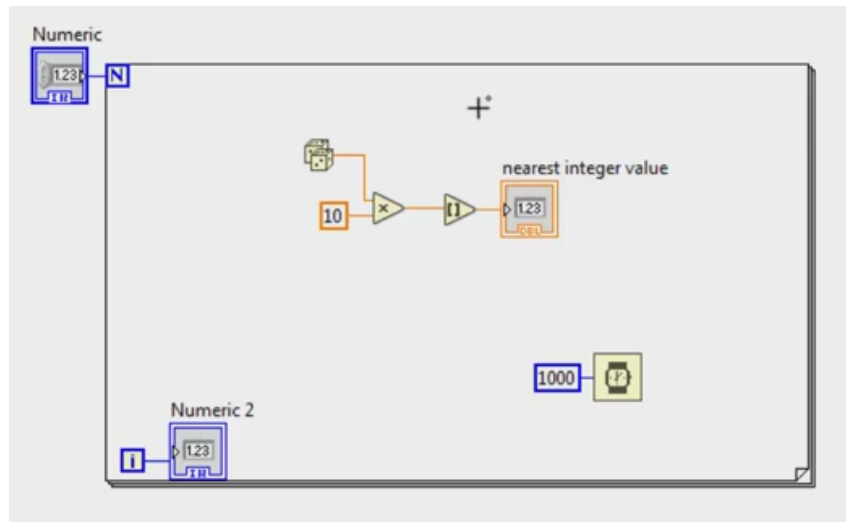


Figure 7: For Loop

## 2 CONCEPT GENERATION

This chapter shows the concept generation that the team went through to eventually find a final design. Because the number of team members are at a minimal number of two, the concept generation process was very simple. Each team member was asked to create two to three unique designs, where these designs then were discussed across the team members to decide which would make it to the final three full system concepts. Sub system concepts were broken up evenly across the two team members to find sub system designs for a few of the systems within the wind turbine data acquisition system.

### 2.1 Full System Concepts

The team came up with three full system concepts to start out the concept generation. Because the wind turbine data acquisition system as a whole is fairly simple and the design requirements are strict, the designs do not differ very much. The system is composed of a monitor, an enclosure, a boom arm containing the sensors, a keyboard, and a mouse. The largest design hurdle was to find the best way to combine these parts while minimizing cost and size, while still maintaining functionality.

#### 2.1.1 Full System Design #1: All-in-One Weather Station

The All in one weather station is one that was designed to keep all the components of the system very close together and not separable. The design is meant to have one single, large enclosure that houses the Raspberry pi, the sense hat, the analog to digital converter, the shunt resistor, and even the monitor, all in the same enclosure. There are water resistant monitor cases available on the market that would work for this design, and would allow for the electrical components to also fit inside with fans on the enclosure as well.

The only drawback to this design that the team thought of was the cost of the water resistant materials. Not only would the monitor enclosure need to be water resistant, but the keyboard and mouse located outside the monitor enclosure would also have to be water resistant, which will be more expensive.

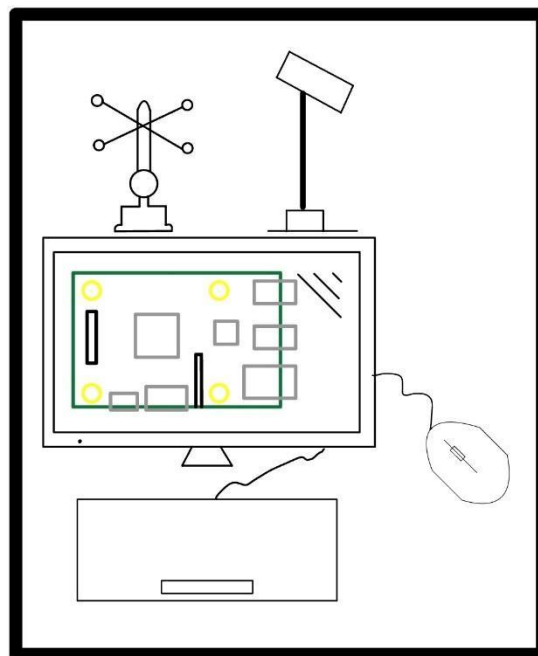


Figure 8: All-in-One Weather Station

### 2.1.2 Full System Design #2: Touch Screen Weather Station

To minimize the footprint of the weather station even more, a small touch screen was brought up as a replacement to the full size monitor screen. Having a small (7 inches) touch screen would allow the enclosure for the other electrical components to be much smaller than the all in one weather station. The touch screen and electronics enclosure would likely be kept on a tripod or table with the anemometer boom arm mounted to a wooded support surface. This design would also eliminate the need for a keyboard and mouse.

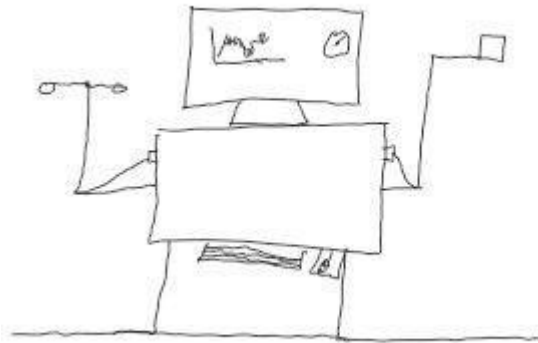


Figure 9: Touch Screen Weather Station

### 2.1.3 Full System Design #3: Standalone Weather Station

The standalone weather station was designed to look similar to most on the market weather stations. The cup anemometer and other sensors are kept on a tripod completely separate from the monitor, keyboard, and mouse. The monitor, keyboard, and mouse could be kept indoors or outdoors closer to the tripod of sensors. Keeping the monitor indoors would allow from access to the system without having to go outside, all the software and data would be inside away from the elements. This particular design would have the need for longer cables for both HDMI and USB going to the Raspberry Pi device, which is not necessarily a problem, but depending on the length, could cause issues with performance.

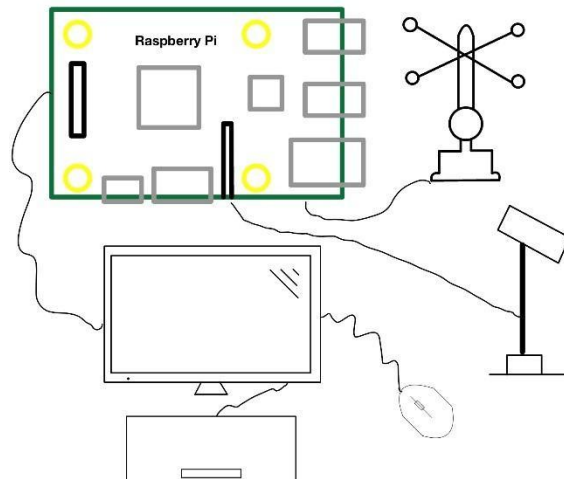


Figure 10: Standalone Weather Station

## **2.2 Subsystem Concepts**

In order to create better designs for separate parts of the weather station, the team broke up the system into three different subsystems. These subsystems were chosen over others because they had higher variability of change that could affect the system as a whole.

### **2.2.1 Subsystem #1: Boom Arm**

The boom arm of the weather station can have an interestingly difficult design to it depending on what the team chooses to put on the arm. As discussed earlier in the Benchmarking chapter, the team could have more than one component on the arm, causing the elevation of the different components to be very important to prevent inaccuracies in measurements in devices in the same turbulent streamlines.

#### **2.2.1.1 Design #1: Double Measurement Arm**

The Double measurement arm would have two sensors on the arm, one of which would be a cup anemometer, the other being a mini wind turbine. The functionality of these devices would be overall to gather wind speed, voltage, and current. The reasoning behind splitting them up is to decrease confusion in the wiring of the Raspberry Pi channels when only using one anemometer to measure all of these things. The elevation of the two different measuring devices is very important as to decrease the creation of more turbulent flow across the first sensor going to the second sensor. Using this design assumes the temperature, humidity, and pressure are kept within the enclosure.

#### **2.2.1.2 Design #2: Triple Measurement Arm**

The triple measurement arm design is a branch off from the double measurement arm, in that it still includes two anemometers, but also has a third arm for the temperature, humidity, and pressure sensors. In the event that the Raspberry Pi sense hat does not accurately measure ambient conditions, another approach will need to be made, and that is where this design will come into play. The ambient condition arm (third arm) would likely include a k-type thermocouple plugged into the Raspberry Pi, along with pressure sensors and humidity sensors.

#### **2.2.1.3 Design #3: Single Measurement Arm**

The single measurement arm was designed to only include one single cup anemometer on a boom arm. This would indicate that all the measurements of wind speed, voltage, and current are coming from the single anemometer. Temperature, humidity, and pressure are coming from the sense hat within the enclosure.

### **2.2.2 Subsystem #2: Enclosure Placement**

To create a better enclosure, the team decided to create designs for the placement of the enclosure itself, and the placement of the components inside the enclosure. The placement of the enclosure will be constrained by other component positions in the real life design of the system, but mostly, the placement of the components is what is important for this section.

#### **2.2.2.1 Design #1: Tall Enclosure**

The tall enclosure is designed to be just as the name say, tall, and skinny. This would allow for a very small design, which would likely go well with the touch screen full system design, because the touch screen could be mounted on top of the enclosure. The tall skinny enclosure would likely be placed on a wooden platform that the boom arm would be mounted to. This design would of course keep the footprint of the overall system at a minimum, but could also create difficulties in wiring space, and flow of cooling air from the fan to cool the Raspberry Pi.

### **2.2.2.2 Design #2: Flat Enclosure**

The Flat enclosure would have a similar functionality to the tall enclosure, but would allow for more of a flat design that could accommodate a monitor to sit on top of the flat enclosure. Placement of components would be fairly straightforward, with everything sitting directly in line with each other, and wires running across the length of the enclosure. Having a flat enclosure would bring about the need for likely more than one fan, because of the placement of the components, and the lack of air flow up and down the height of the box.

### **2.2.2.3 Design #3: Large enclosure with monitor**

This design would go best with the all in one weather station system design, because the components would be placed in a very large box, but it would also have a monitor. This enclosure would have plenty of room for spacing out the Raspberry Pi, digital to analog converter, and the shunt resistor to create less of a buildup of heat. Using this design would violate the customer restraint of dimensions for the enclosure which are 6x10x12 inches, but this could be an exception because of the monitor being kept in the enclosure with the electrical components.

## **2.2.3 Subsystem #3: Visual Layout**

This subsystem design is going to be important when programming the way the visual layout of graphs and gauges looks later in the project. As seen with at home weather stations, the screens that come with them do a great job at laying out the information in an easy to read tile layout so that all the information is in one accessible spot. This design phase aims to find different ways to lay out the interpreted information from the sensors.

### **2.2.3.1 Design #1: Tile layout**

This design aims to mimic the popular layout of most at home weather stations that have a tile layout of all of the information. This would be best on a touch screen and could include tiles that have the ability to move around. These tiles will include power curves, graphs for maximums and minimums of each channel, and tiles that are displaying live data from each channel.

### **2.2.3.2 Design #2: Tab layout**

This design will have tabs similar to a browser or an excel sheet, where each tab brings up a full screen window of whatever information the user is wanting to see. Tabs could include graphs of maximums and minimums, live data, and power curves. This design could be less than ideal because of the need to switch tabs to see different information. Users like to see all the information in an easy to see place all at once.

### **2.2.3.3 Design #3: Touch screen swipe layout**

This design utilizes the touch screen, and will be similar to the tab layout, but will only require a swipe to move between tabs. This design will show the same information as the tab layout, and has the same downsides as the tab layout. Another normal downside to touch screens is that they do not work well in the rain, which could be an issue for an outdoor weather station. Having a touch screen in the rain would render the screen practically useless to the user to choose what data to see.

### **3 DESIGN SELECTED – First Semester**

After all the design steps, analysis completed, and hardware tested, the team came up with a final design for the summer semester. This design tries to encapsulate all the results of testing and design phases to create a system that fully meets all of the customer needs and engineering requirements.

#### **3.1 Design Description**

The final design will include a water-resistant enclosure that is 6x8x4 inches, meeting the customer requirement for size. Within the enclosure, there will be a Raspberry pi to act as the main computer in the data acquisition system. Attached to the top of the Raspberry pi will be an analog to digital converter that will act as the hardware to convert the analog electricity coming from the wind turbine, into a digital signal for the Pi to measure. Within the enclosure there will also be two shunt resistors rated at 10 amps and 30 amps. These resistors will be wired in a way that the ADC can measure their voltage drops at each end.

Within the enclosure there will also be a breadboard that will have the resistance circuit wired to it to reduce the voltage coming from the wind turbine. This will all be kept in the base of the enclosure. The top of the enclosure, from the lid up, will be the 7" touch screen, mounted on a thin piece of plastic with holes for the ribbon connector going to the Raspberry Pi. The enclosure will also contain a cooling fan in one side, and cable glands on the opposite side to run wire out of the enclosure and prevent tugging of wires.

Connected to the Raspberry Pi will not only be an ADC, but also a temperature probe, an atmospheric pressure sensor, and the cup anemometer. These sensors were changed from the original design that utilized a sense hat to measure temperature and pressure. The sense hat was replaced because the temperature sensor did not measure air temperature, but instead measured the temperature of the board, which was much warmer than the air temperature.

The enclosure will be mounted to a wooden board or tripod that will have the anemometer arm mounted. The cup anemometer will be placed at a certain height depending on where the anemometer will be located in respect to the wind turbine to reduce wind measurement uncertainty. For the voltage reduction circuit, calculations were made to find what resistors needed to be placed in parallel and these calculations can be seen in Appendix C. To determine if the precision of the ADC was sufficient, calculations were made to see how precise of voltage measurements would be seen, and based off the engineering requirements the voltage had to have a resolution of  $<0.005$  volts, these calculations can also be seen in Appendix C. The rough CAD drawing of the final model enclosure can be seen below in Figure 11, and the actual enclosure as seen on Amazon can be seen in Figure 12.

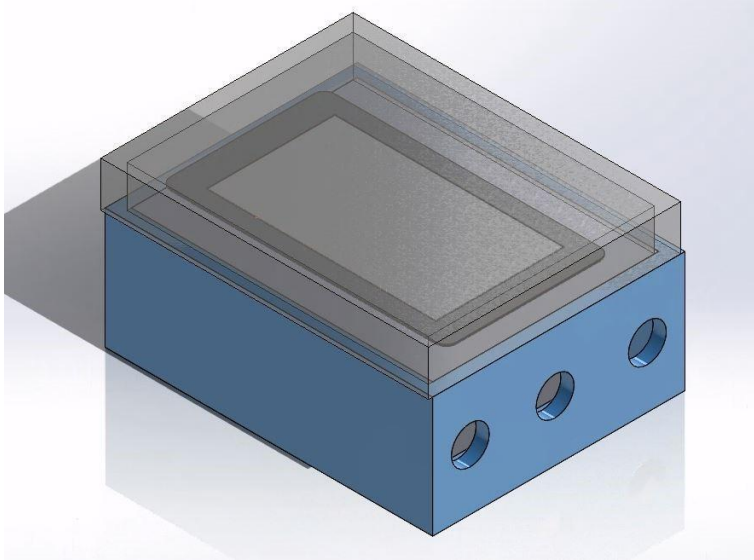


Figure 11: CAD Drawing of Enclosure



Figure 12: Amazon Enclosure Figure

### 3.2 Implementation Plan

To implement the team's design, an initial prototype was created where the Raspberry pi utilized the sense hat to measure temperature and pressure and recorded it in a google sheet. For the next prototype, the team will utilize the ADC, a temperature probe, a cup anemometer, and a simple voltage reduction circuit. The ADC will be measuring the voltage output of the 5v pin of the Pi, the temperature probe will be plugged into the top of the ADC measuring temperature, and the anemometer will be plugged into the GPIO pins to measure wind speed. All of these values will be recorded into a google sheet, but the team plans to remove the internet connection from the system and record data a different way. This final prototype was built and tested during the end of the Summer semester and was found to function properly.

At the beginning of the Fall semester, the team acquired the enclosure, as well as the shunt resistors. Once these items were acquired, the team started mounting components into the enclosure, and started testing the voltage drops across the resistors to measure current coming from the wind turbine. The expected enclosure appearance can be seen in the CAD drawings. An exploded view of the CAD was not necessary as the enclosure is a one-piece component and each item in the enclosure is very simple. Being able to measure the current has been the longest process in the creation of this system, along with the implementation of functional python files. By the end of the fall semester, the team will have created a fully functional wind turbine data acquisition system that will effectively and accurately measure temperature, pressure, wind speed, voltage, and current. The components of the system can be seen in the Bill of Materials as seen in Appendix B.

## 4 Project Management – Second Semester

### 4.1 Gantt Chart

For the fall semester, the team really needed to focus on finishing up the python coding and making sure the physical design worked as it should. With that goal in mind, the team worked tirelessly to develop functional code that would work with all sensors and not only record data, but also plot the live data. Figure # shows the Fall semester Gantt chart used by the team to track these goals and other assignments throughout the semester. There are goals on this Gantt chart such as programming goals, hardware goals, and presentation goals. Utilizing this schedule helped keep the team on track to finish the last semester effectively.

#### Wind Turbine Data Acquisition (PCC)

Spencer Norton and Xuefeng Jiao  
Project Sponsor: David Willy

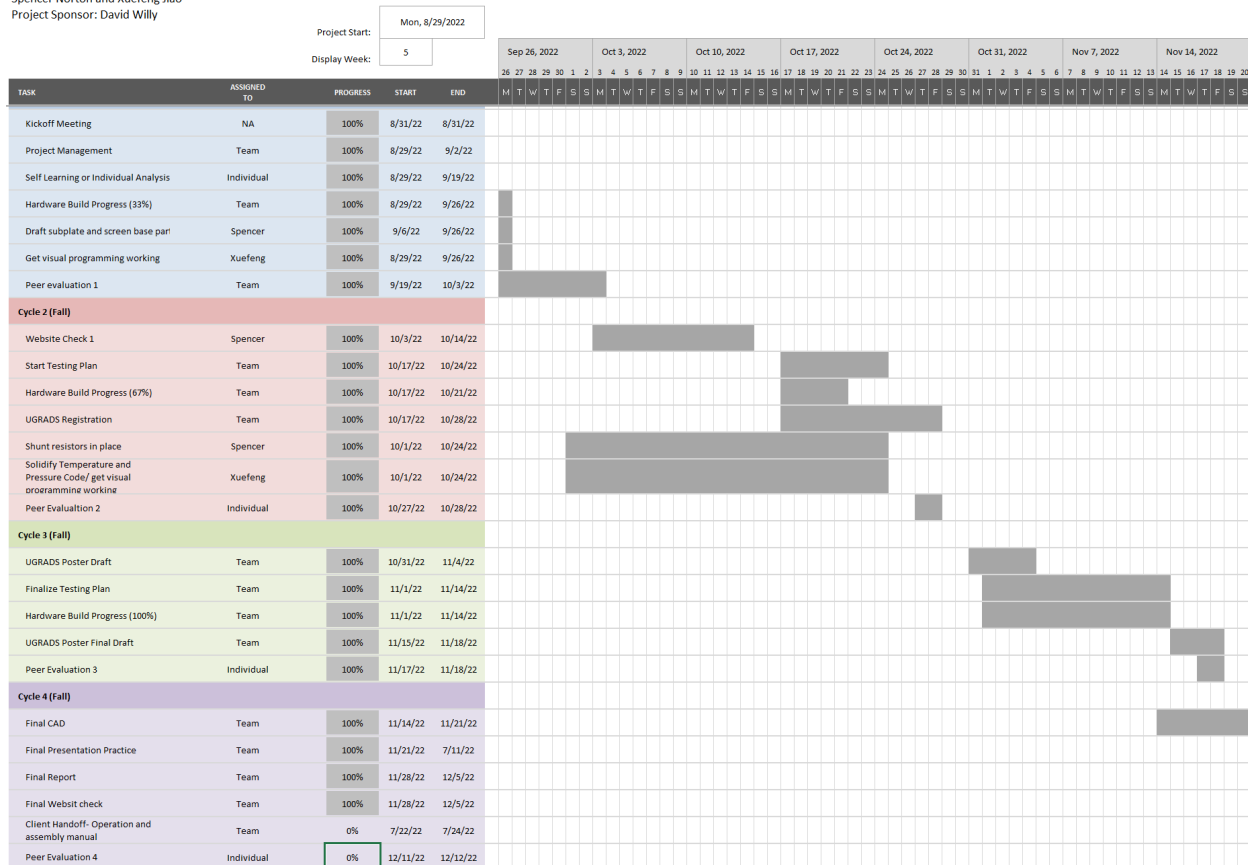


Figure 13: Gantt Chart

### 4.2 Purchasing Plan

For the team to adequately create a wind turbine data acquisition system, money had to be spent on many parts. The team was given a budget of \$1500 for the entirety of the project, so all expenses had to stay under this maximum limit. The team was able to create the fully functional data acquisition system by only spending about 35% of the overall budget. The most expensive parts were the extra Raspberry Pi, and some of the other computational equipment. The sensors, wiring, and enclosure were fairly inexpensive and really helped the team create a quality product. Some parts like the touch screen and metal for the component subplate and screen base were provided free of charge by either the university or from spare parts the team members had laying around. Appendix B shows the full bill of materials that includes all the parts that comprise the wind turbine data acquisition system. Figure # shows the

purchasing plan, or all the parts the team had to purchase as part of the budget provided by the university. This purchasing plan as well as the bill of materials drastically changed from purchasing plan and bill of materials from the summer semester. As the design process continued for the team throughout the fall semester, some parts didn't work, or the team realized other parts were needed, so these parts were purchased and added to the plans.

[illegible]

Figure 14: Purchasing Plan

### 4.3 Manufacturing Plan

The team was able to create a system that did not require much manufacturing. The system is comprised of an electrical enclosure that has user interface ports on the bottom. These ports had to be installed by drilling large holes into the enclosure. This allowed the team to create a water resistant, clean, and easy to use system by adding USB ports, USB C power in, HDMI adapter, Anderson Power pole voltage in and out, and a cable gland for the temperature sensor and anemometer wiring.

The screen base was created to allow the team to utilize a touch screen with the system. This was a big thing for the team, as it was desired that the system could work in multiple scenarios, either with the lid closed with the use of a keyboard and mouse, with the lid open as a touch screen, or with the lid closed utilizing a keyboard and mouse and the use of an external monitor. The screen base is simply a piece of steel with a hole cut into the middle to allow the processing board of the screen to have wires running to it and the ribbon cable for the screen itself. Also mounted to the screen base is the voltage amplifier for the anemometer. This device is mounted on the underside of the screen base to maintain a clean and organized look for the system.

The component subpanel is what houses all the main parts of the system. Figure 15 shows the component subpanel and all the devices mounted to it. The subpanel is made of painted steel, and has multiple holes drilled into it for the mounting of two shunt resistors, the raspberry pi, and perf boards. The shunt resistors are wired to the Anderson power pole connectors on the side of the enclosure. Four wires are going from the shunt resistors into the ADC channels (10 A: IN0 and IN1, 30A: IN2 and IN3). These wires allow the ADC to measure differential input signals from the resistors to measure current. Other wires are coming from the other sensors into the Raspberry Pi. The barometric pressure sensor utilizes the SCL and SDA pins of the Pi to send data, the temperature probe uses the D4 pin of the Pi to send data, and these two sensors also use 3V and ground pins to power them. These are all wired onto the perf boards for a cleaner

wiring system.

There are two perf boards in this system, the top one, or the one that is most clearly visible in the figure shown below is for the temperature and pressure sensing boards. The perf board underneath this board is for the voltage reduction circuit, which has resistors soldered onto the board with one wire coming in, which is receiving voltage from the Anderson Power pole connectors, and one wire going out to IN6 of the ADC. The Anemometer works through a voltage amplifier, that is specifically set to receive a 5V input, and output a 9V signal constantly. This allows the anemometer to work correctly and still output a signal that is small enough for the ADC to handle. The amplifier receives voltage in and ground from the perf board positive and negative sides and the amplifier output wires are running to the wiring for the anemometer that goes through the cable gland. The anemometer wire has three wires (brown=positive, black=negative, and blue=signal). The blue wire is connected to a brown extension wire that goes to IN7 of the ADC.

The last two parts of the system that were manufactured are the wiring for the screen, which are simply two wires coming from a 5V Pi pin, and a ground pin to power the screen. The cooling fan mount was 3D printed from a file online to allow the team to mount a cooling fan to keep the Pi running efficiently [8]. This fan requires a 3V signal and a ground pin, which are wired to the Raspberry Pi pins. Table 4 shows a summary of the manufacturing plan created by the team that includes the item being manufactured, who will make it, how long it will take, the material needed to create the part, and the location manufacturing will take place at.

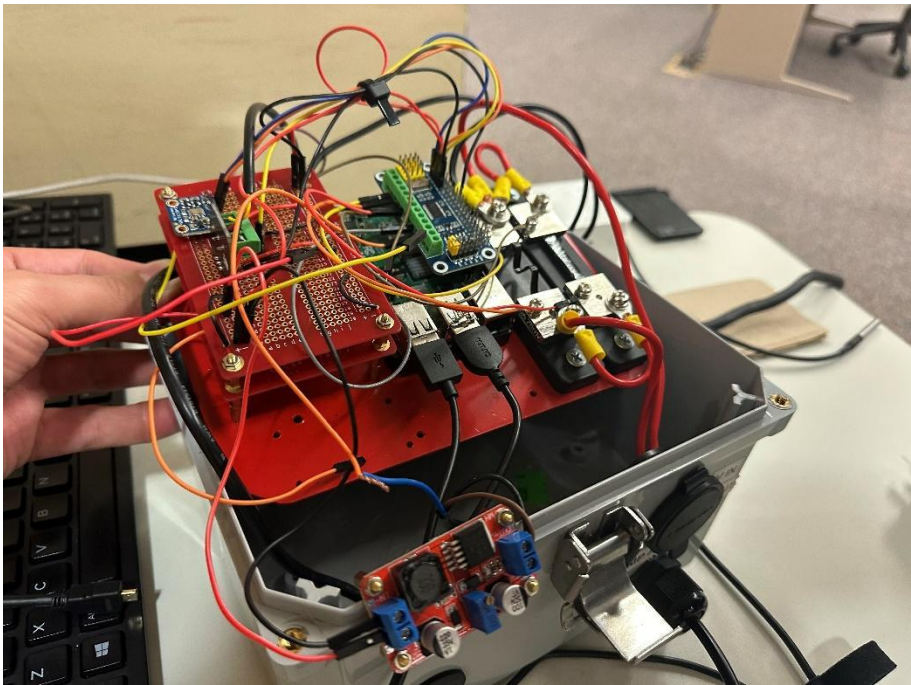


Figure 15: Component Subpanel

Table 4: Manufacturing Plan

| Item Name       | Who will Make | How long to make | material      | location of manufacturing |
|-----------------|---------------|------------------|---------------|---------------------------|
| Enclosure holes | Team          | 1 week           | Polycarbonate | Home                      |

|                             |      |         |       |                   |
|-----------------------------|------|---------|-------|-------------------|
| Screen Base                 | Team | 1 week  | Steel | Home              |
| Component sub plate         | Team | 2 weeks | Steel | Home              |
| Cooling Fan Mount           | Team | 1 week  | ABS   | Home (3D printer) |
| Perf Board Circuit building | Team | 2 weeks | NA    | Home              |
| Assembly of components      | Team | 1 week  | NA    | Home              |

#### ***4.4 Bonus/Substitution Sections – Heat Transfer Analysis***

In addition to the work done for this project, the team plans to take it a step further with a heat transfer analysis in another class. The team plans to utilize the Raspberry Pi, along with cooling heatsink fins and a cooling fan to analyze the effectiveness of the fan/heatsink combo in cooling the CPU. The Raspberry Pi will operate more effectively if the CPU is kept cool, so the team will do three different analysis and compare them to each other, one in Solid works, where a heat transfer analysis will be done to see what the temperature is at the top of the fins with and without a cooling fan. The second analysis will be done mathematically through extended fin analysis, where the team will perform calculations to determine the fin efficiency and the temperature expected to be seen at the fin tip with and without the fan (forced convection). Lastly, the team will utilize a data acquisition system to record temperatures at the CPU surface and the top of the fins with and without a cooling fan, to see how real world temperature changes happen with the use of the heatsink fins on the processors.

## 5 Final Hardware

### 5.1 Final Hardware Images and Descriptions

The final design and hardware that the team was able to create this semester can be seen below in Figure 16. This final design incorporates all the previously discussed parts and python code to operate correctly. Figure 17 shows the final Matplotlib output of the live plotting, these graphs show up on the screen of the Raspberry Pi and update about every second so that live data can be seen frequently. The final enclosure includes USB ports, a USB C power input, HDMI adapter, Anderson Power pole connectors for voltage in and out, and a cable gland for temperature probe and anemometer wiring. A carrying handle was added to the enclosure for easy transportation, and parts were added to the back of the enclosure to keep the temperature probe out of the way and provide a place for the anemometer wire to be stored when not in use.

### 5.2 Design Changes in Second Semester

The second semester brought about nearly all of the teams hardware building. Because the team only created a prototype the first semester, none of the actual hardware was built until the start of the Fall semester. At the end of the summer semester, the team had a fairly good idea of where the project was going to go. An electrical enclosure was already chosen, the manufacturer changed, but the overall shape and design of the enclosure stayed the same. The team decided to add more user interface ports, which was not something that was decided in the first semester. The team also decided to utilize a component subpanel, which was not decided in the first semester. The team was originally planning to mount all the components to the floor of the enclosure, which was found to possibly cause water leaking issues and other problems. With the use of a subpanel, the team not only was able to organize the components, but also add a common ground to the entire electrical system.



Figure 17: Final Hardware Design

The team also had a few sensor changes happen in the second semester. At the end of the summer semester, the team had acquired a sinusoidal output anemometer from the client. This anemometer was found to be very difficult to interface with the Raspberry Pi and the ADC, so the team decided to go a different direction. An Adafruit analog output anemometer was purchased instead and it worked great with the system. The device had to be calibrated and tested, but with the ADC that was used for the system, the Pi was easily able to read voltage input of the anemometer and convert it to wind speed. The

pressure sensor was also changed, where the original sensor purchased was found to not work well with python, so the team decided to keep shopping. An Adafruit barometric pressure sensor was purchased and worked flawlessly with Python, and was extremely accurate.

The team did not necessarily have very many hardware obstacles other than the troubleshooting of the current sensing circuit and the python code. Countless hours were spent on the development of the python code, although Waveshare Python libraries were used to interface with the ADC, many hours still had to be put in to achieve a successful python code that received accurate information from all the sensors. Not only did the python file have to be developed to read data, but plotting and recording to a csv was also a big roadblock that had to be worked around. The team tried multiple different plotting libraries like tkinter, matplotlib, pandas, and even some dashboards to attempt to display data in a graphical user interface. This graphical user interface was not entirely achieved this semester, but could be a very feasible option for future work if an internet connection were constantly provided to the Raspberry Pi. Instead of a GUI, the team utilized Matplotlib to create live graphs with the animation function. These graphs can be seen in Figure 18.

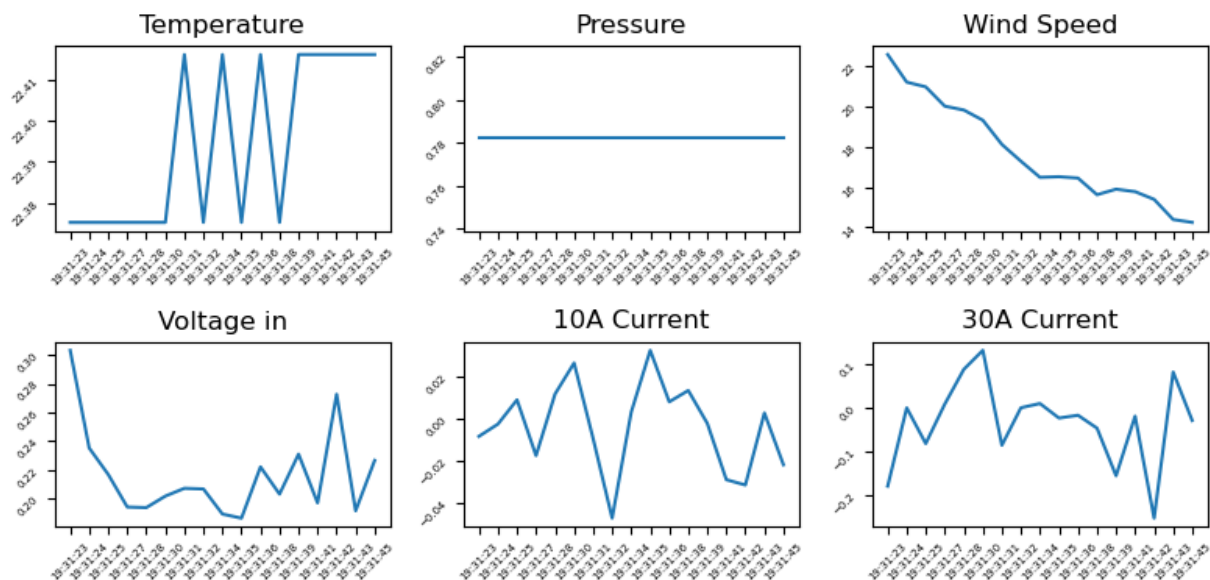


Figure 18: Matplotlib Live Graphs

The current sensing circuit was tested and developed in two ways. First the team attempted to create the circuit by the use of the positive side of the voltage signal. This was found to be tedious as each input would require a voltage reduction circuit, hence why there are still two other resistor circuits on the bottom perf board. After many hours of troubleshooting, the team decided to try the negative side of the voltage signal, and this worked. Using the negative side of the signal allowed the team to read very small voltages, which was possible due to the high accuracy of the ADC, and then develop the python file to read differential signals. The use of the negative signal also allowed the team to create a common ground, but caution had to be taken as to what was attached to this common ground, and it is charged higher than the normal ground provided by the Pi. Altogether, the team was able to create a safe and reliable system that meets all the specification requirements provided by the client.

## 6 Testing

This section of the report discusses the testing that the team performed to ensure the data acquisition system designed is able to meet the customer and engineering requirements. All requirements were met by the team, but a few requirements in specific required testing to ensure the devices within the system worked as they should, both accurately and correctly.

### 6.1 Testing Plan

In order for the team to represent that the customer requirements and engineering requirements have been met, the team has performed numerous tests. These tests have been done to prove the validity of the project, and come up with calibration values if any sensors used have deviated from manufacturer calibration tolerances. Table 5 shows the tests that were performed to determine the validity of the system. In this table, each testing experiment is referenced back to the corresponding engineering and customer requirements, and can also be seen in the QFD. Detailed testing plans and procedures can be seen in the “Final Testing Plan” document created by the team, this can be found on the team’s website.

Table 5: Testing Plan Summary

| Experiment/Testing                 | Relevant Design Requirements |
|------------------------------------|------------------------------|
| Ex 1 – Temperature Range Test      | ER2                          |
| Ex 2 – Pressure Range Test         | ER3                          |
| Ex 3 – Voltage/ Current Test       | ER4, ER5, CR8                |
| Ex 4 – Wind Speed Test             | ER1                          |
| EX 5 – Graphing and Recording Test | CR2, CR3, CR4                |

### 6.2 Testing Results

The team successfully completed each test by following the procedures outlined in the “Final Testing Plan” document. The results of these tests can be seen in this section. The team created specification sheets to help track the completion of testing plans and client acceptance for each test. This specification sheet shows the target value provided by the client, the measured/calculated value found in the tests, and whether or not the requirement was met by the team, and whether or not the client accepts the measured/calculated value found in the tests. Table 6 shows the specification sheet for the engineering requirements, and Table 7 shows the specification sheet for the customer requirements.

Table 6: Engineering Requirements Specification Sheet

| Engineering Requirement         | Target    | ER Tolerance | Measured/Calculated Value | Requirement Met? (Y/N) | Client Acceptable? (Y/N) |
|---------------------------------|-----------|--------------|---------------------------|------------------------|--------------------------|
| <b>ER1: Measure Wind Speed</b>  | 0-25 m/s  | NA           | 0-25 m/s                  | Y                      | Y                        |
| <b>ER2: Measure Temperature</b> | -25-100 F | NA           | -25-100 F                 | Y                      | Y                        |

|                              |                |                |                                  |   |   |
|------------------------------|----------------|----------------|----------------------------------|---|---|
| <b>ER3: Measure Pressure</b> | ~0.75 bar      | NA             | Flagstaff Conditions<br>0.79 bar | Y | Y |
| <b>ER4: Turbine Voltage</b>  | 0-48V          | < $\pm 0.05$ V | 0-48 V                           | Y | Y |
| <b>ER5: Current</b>          | 0-10A<br>0-30A | < $\pm 0.01$ A | 0-10A<br>0-30A                   | Y | Y |

Table 7: Customer Requirements Specification Sheet

| CUSTOMER REQUIREMENT                                 | REQUIREMENT MET? (Y/N) | CLIENT ACCEPTABLE? (Y/N) |
|------------------------------------------------------|------------------------|--------------------------|
| <b>CR1: Small Enclosure</b>                          | Y                      | Y                        |
| <b>CR2: Display Gauges and Graphs</b>                | Y                      | Y                        |
| <b>CR3: Display Live Data, Track Historical Data</b> | Y                      | Y                        |
| <b>CR4: Downloadable to excel file</b>               | Y                      | Y                        |
| <b>CR5: Cost Within Budget</b>                       | Y                      | Y                        |
| <b>CR6: Durable and Robust Design</b>                | Y                      | Y                        |
| <b>CR7: Reliable Design</b>                          | Y                      | Y                        |
| <b>CR8: Safe to Operate</b>                          | Y                      | Y                        |

To achieve such results from the testing plan, the team performed the wind speed test by utilizing a constant wind source, and compared the readings of a BT-846A anemometer to calibrate the device. Similar procedures were completed for the temperature sensor and the voltage reading ADC. For the temperature probe, the team used a DTT-1372 as the standard thermometer to calibrate the temperature sensor used in this project. Hot water and cold water were used, and a calibration curve was found. The voltage reading circuit utilized a variable power supply and load to maintain a constant voltage input to the ADC. This constant input was compared to the actual reading of the Raspberry Pi and a calibration curve was created for this device as well. In total, three calibration curves were created, and they can be seen in Figure 16, 17, and 18.

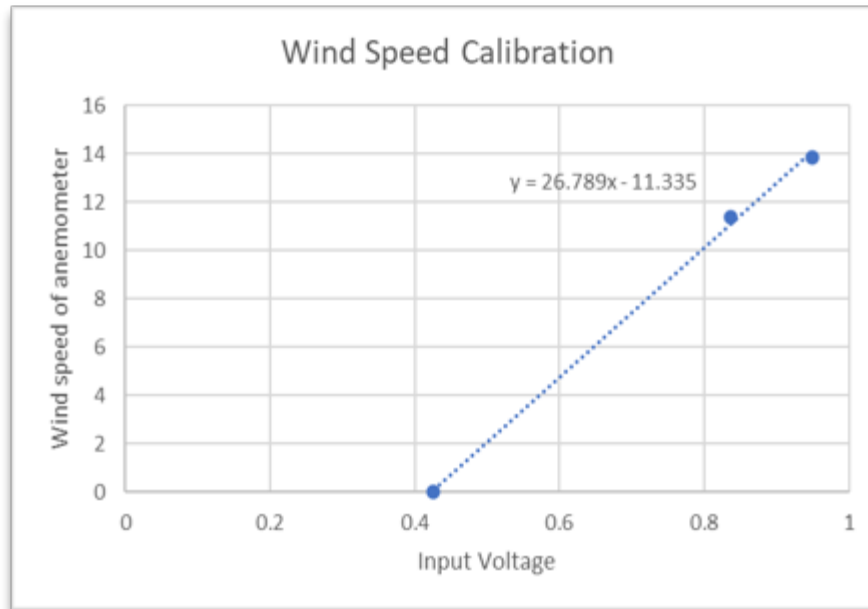


Figure 16: Adafruit Anemometer Calibration Curve

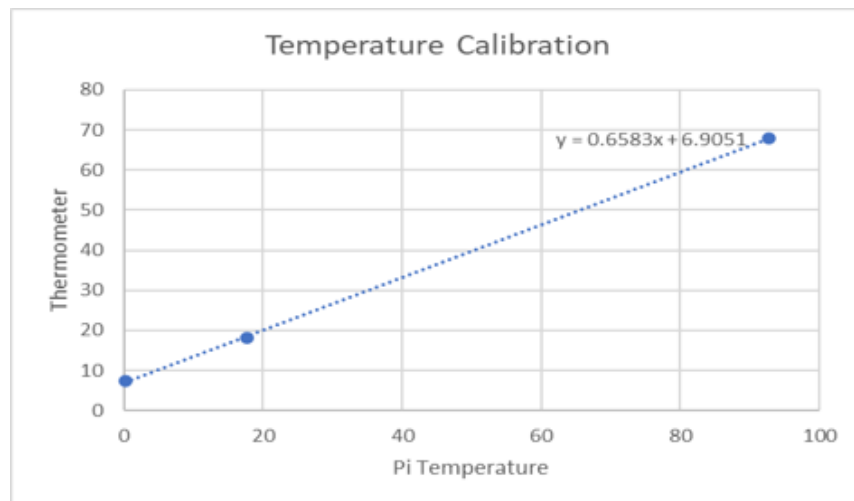


Figure 17: Temperature Sensor Calibration Curve

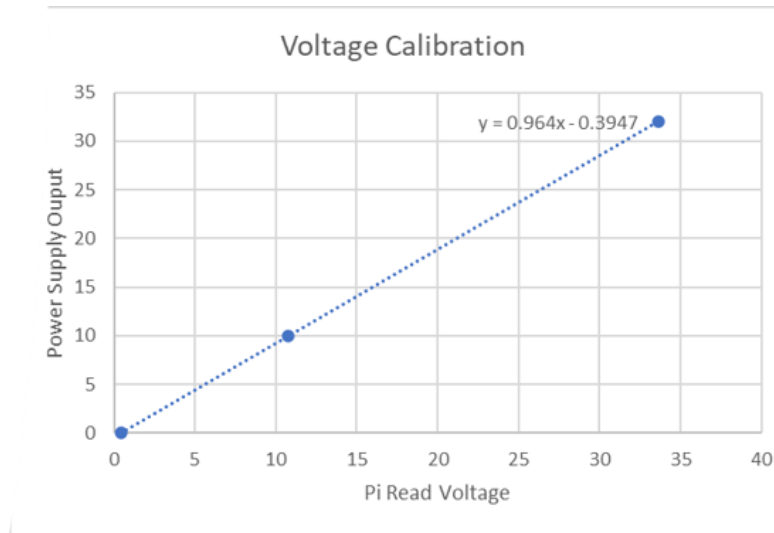


Figure 18: ADC Voltage Reading Calibration Curve

These calibration equations were added into the Python files for reading and recording data to ensure accurate and correct data was being recorded. These Python code files can be seen in Appendix D. Testing of the Barometric pressure sensor was not as in depth as some of the other sensors, but it was tested. To test the sensor, the team compared the readings to that of other nearby weather stations, and compared to calculations. Calculations for the testing of the pressure sensor can be seen in the “Final Testing Results Presentation” which can be found on the team’s website. The results of this calculation showed the pressure sensor was reading correctly, because the equation calculated pressure based off elevation, which was found to be about 2173 m, and the equation brought about an answer of 0.7863 bar, the pressure sensor read about 0.782255 bar.

The water resistance test was completed by using a spray bottle with water in it. Water was sprayed onto the top of the enclosure while it was closed, and the inside of the enclosure was inspected for any leakage of water. The enclosure did a good job of keeping all water out for this test. Although the test was successful, the enclosure would not handle complete submersion in water due to the user interface ports not being entirely sealed. This is an acceptable design aspect, and the team will still continue to try to seal the enclosure with the use of silicone gasket material to keep water out of the ports cut into the enclosure.

## 7 RISK ANALYSIS AND MITIGATION

This section discusses the failure modes and effected analysis (FMEA) that the team completed in the first semester, and the actions taken to avoid these potential failures in the design stage of the second semester. Design decisions were made to mitigate or remove the possibility of these failures happening, and changes were made to python code files to make readings as accurate as possible.

### ***7.1 Potential Failures Identified First Semester***

In order for the team to know the validity and sustainability of the design created, a failure modes and effected analysis (FMEA) needed to be created. During the first semester, the team created an FMEA of expected failures that could be seen within the designed system. The current and full FMEA can be seen in Appendix C, which shows all of the potential failures that the team expected to see in the design stage of the second semester. These failures include mostly measurement device failures such as the temperature probe, the barometric pressure sensor, the anemometer, the voltage sensing circuit, and the current sensing circuit. Other failures include the voltage reduction circuit, and the ADC not working correctly.

### ***7.2 Potential Failures Identified This Semester and Risk Mitigation***

The team encountered nearly every potential failure in one way or another during the second semester. Most of these failures were corrected and the system is expected to work properly in the future. For example, the temperature sensor was originally found to work properly until testing, where it was found the sensor was highly inaccurate at lower temperatures. The team calibrated the sensor and added the calibration equation into the Python code so that accurate temperatures would be given to the user. The first barometric pressure sensor purchased by the team was a good sensor, but it did not interface easily with Python. Because the sensors were inexpensive, the team purchased another sensor that worked better with the code. This sensor worked great and was highly accurate, so no actual failures were seen with this device other than simply converting sensor units to the required unit of bar.

The anemometer had a similar situation to the temperature sensor, where it was found to be fairly inaccurate. The team performed a calibration on this device that basically created a new equation to convert voltage to wind speed. The equation provided by the manufacturer was not working, and the new equation works correctly when compared to a known accurate wind speed sensor. Measuring voltage and current was found to be inaccurate with the ADC used by the team due to “ghosting” across channels. The team was unable to remove ghosting from the ADC, but a calibration was completed and mostly removed any inaccuracies caused by ghosting.

Being able to measure the voltage was a task that took the team a while to complete. During the building process, the team tried multiple resistor combinations until the correct one was found. The team was able to basically create a circuit that reduced voltage by dividing voltage by 100 (i.e. 32 v input is 0.32 v into the ADC), this was accomplished by the use of a 1 kilo Ohm resistor in parallel with a 10 ohm resistor. The current sensing circuit was another difficult task for the team where heavy testing was completed to ensure an accurate and safe current sensing circuit. The team tried both positive and negative flow through the shunt resistors to read voltage drops. It was found that the negative side was best because of the lower voltage read by the ADC in this situation. This ensured that the ADC never saw a voltage above the maximum input of 2.5 v. This was a potential critical failure for the system, and it was successfully mitigated by the use of hardware by the team.

Failure number 7 seen in the FMEA was not seen by the team because a good quality ADC was initially chosen. There were many different ADC devices on the market that could have been used, but the team chose the Waveshare high precision ADC as the device used in the project. This device had extremely high accuracy and code for the ADC was already developed. Although code was already given for this device, the team had to make heavy modifications to allow the code to work with all other devices. The

voltage division circuit previously discussed ensures that the ADC will not be overcharged, and the use of shunt resistors wired on the negative side of the input voltage ensured very small voltage signals seen by the ADC for current sensing. This potential failure was successfully mitigated by the use of these hardware components.

## 8 LOOKING FORWARD

The team is confident that they have created a good quality, highly accurate, and reliable data acquisition system. Any design can be improved upon, so this section will discuss what could possibly be done to the system to make it work even better.

### 8.1 *Future Testing Procedures*

For the Wind turbine data acquisition system to be improved, we think the best thing for it would be to develop a GUI. Most GUIs for Raspberry Pi's work great with online capabilities, which we were unable to develop due to the requirement of being operable without internet. An internet independent GUI was started and can be found on the Pi by following the path /home/pcc/Gui Tests. These three files were as far as the team was able to get with the GUI, and it can be further developed by another team.

The team also tried to calibrate the sensors to the best of their ability with the resources they had. This could be taken a step further by acquiring more accurate temperature standards, or a wind tunnel to calibrate the anemometer. The ADC is also not perfect, for some reason the device experiences ghosting, and this voltage gets distributed across all channels no matter what the voltage input of the individual channel is. This does have a solution, but the team did not have time to resolve it this semester. This device could be significantly more accurate if the ghosting was resolved.

### 8.2 *Future Iterations*

This project could be modified by adding GUIs like previously discussed, or by using different hardware. The Raspberry Pi is not made to work well with analog inputs, and an Arduino may be more suitable for measuring voltage and current. This could be a change that could make this system better. The team really wanted to add a tripod or some kind of mounting hardware to the electrical enclosure to make it more user friendly. In some of the design stage ideas, the team drew out a design that included a tripod, and a place for the keyboard and mouse to be placed. This would greatly enhance the user experience with the system if things like this were added. Other than these small changes, the system has been designed to do very specific work, and the enclosure chosen is so small and compact that adding any more hardware would be a difficult task, so the system is confined on space, and not many changes can be made to the overall hardware functionality of the system.

Although space within the enclosure is limited, a possible lithium ion battery could be added to make the system completely standalone and portable. Adding a battery could be an interesting capstone project for an Electrical engineering major to learn more about batteries and the wattage and voltage consumption analysis that goes into using such power sources.

## 9 CONCLUSIONS

In conclusion, the Design Requirements, updated Functional Decomposition, Standards and Codes, Testing Procedures, Risk Analysis and FMEA, Critical Failures, Risks and Trade-offs Analysis, Design Description, and Implementation Plan are presented in this report, and each of them is updated from time to time and thoroughly reviewed. So far, the team has successfully acquired data from the environment and stored data in the local files. Besides, all the sensors used for the future design process are collected. The final prototype, together with all the documents needed to build it are presented to the project client. The team has been working hard to achieve the goal of designing and building a PCC (point of common coupling) data collection system out of inexpensive hardware and relatively free software. The team has finished all the engineering requirements and customer needs as outlined in the course rubric and from weekly client meeting. The team has written a new code to measure the data of temperature, pressure, voltage, and wind speed at the same time, and plotting these data using Matplotlib. The team also finishes the testing procedure to ensure the accuracy of acquired data.

### 9.1 Reflection

The team was able to create a wind turbine data acquisition system that met all the requirements of the client. Not only did the team successfully meet all the requirements, but more importantly, the team created a system that could potentially help with creating a cleaner wind energy environment. With the creation of a power curve, which is wind speed vs power output, the data acquisition system could show the user how well the wind turbine can output power based off wind speed.

The team really wanted to ensure the system was safe to operate, and it is believed to have successfully accomplished that. The main concern for safety with the system would be associated with the voltage sensing and current sensing circuits. When creating the voltage division circuit, the team could have used resistors to divide the voltage by a small amount, while still staying beneath the ADC 2.5 volt maximum input. Instead, to maintain safety to both the user and the hardware components of the system, the team divided the circuit by 100 to ensure voltage would nearly never get above this maximum input. If the voltage did get above the maximum, the system would be operating outside of its required parameters. As previously discussed, the team utilized the negative signal for the current sensing circuit to maintain a low voltage for the ADC to measure, because the ADC is so accurate, this is possible. By choosing such hardware and wiring options, the team can ensure the electrical circuits within the machine are safe and reliable.

### 9.2 Resource Wishlist

This project definitely would have went better with some more resources. The project was constrained to be within a certain size and to use relatively inexpensive hardware. The team met the budget constraint very easily, but looking back, more expensive hardware could have been used. More specialized equipment would have been useful for calibrations and testing. The team was set up to work with only two students, which worked fairly well for this project, but it was a little heavy in the coding area, which was difficult because both students are ME majors and did not have a whole lot of experience with Python or similar coding types.

### 9.3 Project Applicability

#### Spencer:

This project has really helped me learn a lot more about programming, which was one of my weaknesses coming into this project and definitely one of the things I dreaded working on for the project. Learning how to interpret existing code and learning how to write code from scratch was interesting and sometimes fun. In my time with W.L. Gore, I have used a data acquisition system or two, which was interesting to see the background associated with such machines. I plan to continue my learning with programming, and

learn to apply it to my professional career and to my hobbies by working with Arduinos and Raspberry Pi's some more. I think with the way the engineering world is going, computer skills and programming skills are going to be absolutely vital to the success of engineering projects and engineering careers. I am excited to apply what I have learned and maybe even take more training/classes outside of my Bachelor's degree to learn more about computer programming and apply them to my personal career. I have thoroughly enjoyed learning new skills and learning through application of engineering fundamentals throughout this project, and I hope the next group is able to improve on our design and make it even better.

**Xuefeng:**

I have learned a lot from this project, both for professional knowledge and extracurricular abilities. This is the first time for me to learn and use Python to write codes. As a mechanical engineering major student, I have been thinking about learning and strengthening my coding abilities since computer science is the most popular major on the job market. Throughout this project, I successfully learned Python together with its libraries. I also learned how to use raspberry pi and multiple sensors to collector data from the environment. Raspberry pi is important in the engineering world for its convenience and functionality. In addition to the Python and raspberry pi, I also get the chance to work with a team under the guidance of a professor in an English environment. As an international student, it is important for me to strengthen my English communication skills. Besides, working as a team requires more than your individual abilities. How to successfully works as a team requires a strong communication ability. I am also proud of our final product, it looks good and functions well.

## 10 REFERENCES

- [1] Mechanical Engineering. 2022. Capstone projects | Mechanical Engineering. [online] Available at: <<https://nau.edu/mechanical-engineering/capstone/>> [Accessed 7 July 2022].
- [2] “Wire size guide: What size wire do I need?,” *What Size Wire for My Breaker Do I Need?* [Online]. Available: <https://www.totalhomesupply.com/wires-and-circuit-breakers#:~:text=For%20a%20maximum%20of%2030,is%20a%20central%20air%20conditioner.> [Accessed: 05-Dec-2022].
- [3] “What Size Wire for My Breaker Do I Need?,” [www.totalhomesupply.com](https://www.totalhomesupply.com/wires-and-circuit-breakers#:~:text=For%20a%20maximum%20of%2030). <https://www.totalhomesupply.com/wires-and-circuit-breakers#:~:text=For%20a%20maximum%20of%2030> [accessed Dec. 06, 2022].
- [4] Ambient Weather. 2022. Ambient Weather WS-2902C Smart Weather Station with WiFi Remote Monitoring and Alerts. [online] Available at: <[https://ambientweather.com/amws2902.html?utm\\_id=go\\_cmp-9255048604\\_adg-93600172293\\_ad-416614400595\\_pla-993418500170\\_dev-c\\_ext-prd-WS-2902C\\_mca-147779820\\_sig-Cj0KCQjw5ZSWBhCVARIsALERCvx2jPA6rdUa1tHBxoEzFEyBXSpXV-k2sNSfIE\\_ABHPPdM7ayWymJ-waAlWfEALw\\_wcB&utm\\_source=google&gclid=Cj0KCQjw5ZSWBhCVARIsALERCvx2jPA6rdUa1tHBxoEzFEyBXSpXV-k2sNSfIE\\_ABHPPdM7ayWymJ-waAlWfEALw\\_wcB](https://ambientweather.com/amws2902.html?utm_id=go_cmp-9255048604_adg-93600172293_ad-416614400595_pla-993418500170_dev-c_ext-prd-WS-2902C_mca-147779820_sig-Cj0KCQjw5ZSWBhCVARIsALERCvx2jPA6rdUa1tHBxoEzFEyBXSpXV-k2sNSfIE_ABHPPdM7ayWymJ-waAlWfEALw_wcB&utm_source=google&gclid=Cj0KCQjw5ZSWBhCVARIsALERCvx2jPA6rdUa1tHBxoEzFEyBXSpXV-k2sNSfIE_ABHPPdM7ayWymJ-waAlWfEALw_wcB)> [Accessed 7 July 2022].
- [5] Projects.raspberrypi.org. 2022. [online] Available at: <<https://projects.raspberrypi.org/en/projects/build-your-own-weather-station/5>> [Accessed 7 July 2022].
- [6] Labjack.com. 2022. U3 | LabJack. [online] Available at: <[https://labjack.com/products/u3?gclid=Cj0KCQjw5ZSWBhCVARIsALERCvxgt2qkQ0dxch77yVa\\_f0PIESNv6GAssg-ee0ixS8VXNqPNeI6yKx0aAoPNEALw\\_wcB](https://labjack.com/products/u3?gclid=Cj0KCQjw5ZSWBhCVARIsALERCvxgt2qkQ0dxch77yVa_f0PIESNv6GAssg-ee0ixS8VXNqPNeI6yKx0aAoPNEALw_wcB)> [Accessed 7 July 2022].
- [7] 2022. [online] Available at: <<https://www.digikey.com/en/maker/projects/raspberry-pi-analog-to-digital-converters/72388f5fla0843418130f56c53a1276c#:~:text=MCP3008%20Python%20library!-,ADS1015%20%2F%20ADS1115,bit%20ADC%20with%204%20channels.>> [Accessed 7 July 2022].
- [8] HAT, P. and Power, A., 2022. Pi-16ADC Analog-Digital Converter HAT. [online] PiShop.us. Available at: <<https://www.pishop.us/product/pi-16adc-analog-digital-converter-hat/>> [Accessed 7 July 2022].
- [9] Thingiverse.com, “Raspberry pi 4 fan mount by Hesi-Re,” *Thingiverse*. [Online]. Available: <https://www.thingiverse.com/thing:4097379>. [Accessed: 05-Dec-2022].
- [10] K. B. Swain, S. Dash and S. S. Gouda, "Raspberry PI based Integrated Autonomous Vehicle using LabVIEW," 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), 2017, pp. 69-73, doi: 10.1109/SSPS.2017.8071567.
- [11] Halfacree, G., 2020. THE OFFICIAL Raspberry Pi Beginner’s Guide. 4th ed. Cambridge: Raspberry Pi Trading Ltd.
- [12] Ehsani, B., 2016. Data Acquisition Using LabVIEW. 1st ed. Birmingham: Packt Publishing Ltd.
- [13] Bhasin, H., 2019. PYTHON BASICS. 1st ed. Dulles: MERCURY LEARNING AND INFORMATION LLC.



# 11 APPENDICES

## 11.1 Appendix A: House of Quality or Functional Decomposition

|                               |                                                              |                  |                                             |                              |                                  |                               |                                |                        |                           |                                  |                                    |
|-------------------------------|--------------------------------------------------------------|------------------|---------------------------------------------|------------------------------|----------------------------------|-------------------------------|--------------------------------|------------------------|---------------------------|----------------------------------|------------------------------------|
| System QFD                    |                                                              |                  | Project: 2022 Wind Turbine Data Acquisition |                              |                                  |                               |                                |                        |                           |                                  |                                    |
|                               |                                                              |                  | Date: 6-17-2022                             |                              |                                  |                               |                                |                        |                           |                                  |                                    |
| 1                             | Measure Wind speeds (Test 1)                                 |                  |                                             |                              |                                  |                               |                                |                        |                           |                                  |                                    |
| 2                             | Measure temperature (Test 2)                                 | ++               |                                             |                              |                                  |                               |                                |                        |                           |                                  |                                    |
| 3                             | Wind turbine voltage measurement (Test 4)                    | -                | +                                           |                              |                                  |                               |                                |                        |                           |                                  |                                    |
| 4                             | Wind turbine current (Test 5)                                | -                |                                             |                              |                                  |                               |                                |                        |                           |                                  |                                    |
| 5                             | PM measurement of wind turbine                               |                  | -                                           |                              |                                  |                               |                                |                        |                           |                                  |                                    |
| 6                             | Reduce production time                                       |                  | +                                           | ++                           |                                  |                               |                                |                        |                           |                                  |                                    |
| 7                             | Measure pressure (Test 3)                                    |                  |                                             | -                            |                                  |                               |                                |                        |                           |                                  |                                    |
| 8                             | Be able to measure data remotely                             | +                |                                             | ++                           |                                  |                               |                                |                        |                           |                                  |                                    |
| 9                             | Increase testing and measuring accurasion                    | ++               |                                             | +                            |                                  |                               |                                |                        |                           |                                  |                                    |
|                               |                                                              |                  | Technical Requirements                      |                              |                                  |                               |                                |                        |                           |                                  |                                    |
|                               |                                                              |                  | Measure Wind speeds (Test 1)                | Measure temperature (Test 2) | Wind turbine voltage measurement | Wind turbine current (Test 5) | PM measurement of wind turbine | Reduce production time | Measure pressure (Test 3) | Be able to measure data remotely | Increase testing and measuring acc |
|                               | Customer Needs                                               | Customer Weights |                                             |                              |                                  |                               |                                |                        |                           |                                  |                                    |
| 1                             | Meets all design requirements                                | 5                |                                             |                              |                                  |                               |                                |                        | 9                         |                                  |                                    |
| 2                             | Meets all safety requirements                                | 5                |                                             |                              |                                  |                               |                                |                        | 9                         |                                  |                                    |
| 3                             | Small Enclosure                                              | 3                | 1                                           | 3                            |                                  |                               |                                | 3                      | 3                         |                                  |                                    |
| 4                             | Low Cost                                                     | 4                |                                             | 9                            | 9                                |                               |                                | 9                      |                           |                                  |                                    |
| 5                             | Production Capable                                           | 4                |                                             | 3                            |                                  |                               |                                | 9                      |                           |                                  |                                    |
| 6                             | Easy to interpret display                                    | 3                | 9                                           | 1                            |                                  | 9                             | 3                              |                        |                           |                                  | 9                                  |
| 7                             | Store historical data and interpret live data                | 3                | 9                                           |                              |                                  | 1                             | 3                              |                        |                           | 9                                |                                    |
| 8                             | Take historical data and create other visual interpretations | 3                | 3                                           |                              |                                  | 3                             | 3                              |                        |                           | 1                                | 1                                  |
| 9                             | Adjustability                                                | 2                |                                             |                              |                                  |                               | 9                              |                        |                           | 9                                | 9                                  |
| Technical Requirement Units   |                                                              |                  | m/s                                         | °C                           | V                                | A                             | rpm                            | Hours                  | Pa                        | *                                | *                                  |
| Technical Requirement Targets |                                                              |                  | 15                                          | *                            | *                                | TBD                           | 5                              | 8                      | *                         | *                                | *                                  |
| Absolute Technical Importance |                                                              |                  | 66                                          | 60                           | 36                               | 39                            | 45                             | 81                     | 99                        | 48                               | 48                                 |
| Relative Technical Importance |                                                              |                  | 3                                           | 4                            | 9                                | 8                             | 7                              | 2                      | 1                         | 5                                | 5                                  |

## 11.2 Appendix B: Full Bill of Materials

| Bill of Materials                                                                                             |                                                                                                                                                                                                                       |                  |                               |                            |                          |        |                                             |                                      |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-------------------------------|----------------------------|--------------------------|--------|---------------------------------------------|--------------------------------------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Item Name                                                                                                     | Item Description                                                                                                                                                                                                      | Item Quanty      | Vendor                        | Cost                       | Make/B                   | Lead T | Past Status                                 | Size                                 | Part Number          | Link                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Enclosure                                                                                                     | Enclosure used to hold Raspberry Pi, ADC, and shunt resistors                                                                                                                                                         | 1                | NEMA Supply                   | 83.37                      | Buy/Make                 |        | Received                                    | 8x6x4 inches                         | H0064-CHLL           | <a href="https://www.nemasupply.com/products/integrus-premium-series-polycarbonate-enclosure-clear-backing-hatch-cover-mounting-feet?variant=155412440&amp;bvos=1">https://www.nemasupply.com/products/integrus-premium-series-polycarbonate-enclosure-clear-backing-hatch-cover-mounting-feet?variant=155412440&amp;bvos=1</a>                                                                                                                                                                                         |
| Raspberry Pi                                                                                                  | Main Computer to Collect Data<br>Used to convert analog signal electricity to digital signal for the Pi to read                                                                                                       | 1                | PiShop                        | 123.93                     | Buy                      |        | Received                                    | 3.34x2.2x0.67 inches                 | TGB-9002             | <a href="http://www.pishop.us/products/raspberri-pi-4-model-b-tbq/tcrrasrberripi">http://www.pishop.us/products/raspberri-pi-4-model-b-tbq/tcrrasrberripi</a>                                                                                                                                                                                                                                                                                                                                                           |
| Analog to Digital Converter (ADC)                                                                             |                                                                                                                                                                                                                       | 1                | WaveShare                     | 52.4                       | Buy                      |        | Received                                    | 2.55x1.13x0.72 inches                | NA                   | <a href="https://www.waveshae.com/dp/B00K391M4T?pcst=frtppgs_yg2ot_dtl_b_product_details">https://www.waveshae.com/dp/B00K391M4T?pcst=frtppgs_yg2ot_dtl_b_product_details</a>                                                                                                                                                                                                                                                                                                                                           |
| Shunt Resistor 10A                                                                                            | used to measure voltage drop and plot current                                                                                                                                                                         | 1                | Amazon                        | 10.73                      | Buy                      |        | Received                                    | 4.7x0.38 inches                      | NA                   | <a href="https://www.amazon.com/dp/B00K391M4T?pcst=frtppgs_yg2ot_dtl_b_product_details">https://www.amazon.com/dp/B00K391M4T?pcst=frtppgs_yg2ot_dtl_b_product_details</a>                                                                                                                                                                                                                                                                                                                                               |
| Shunt Resistor 30A Resistors                                                                                  | used to measure voltage drop and plot current<br>used to reduce voltage coming from turbine<br>Used to give a detachable connection between enclosure and turbine                                                     | 1<br>2           | Amazon<br>Amazon              | 10.73<br>NA                | Buy<br>Buy               |        | Received<br>Received                        | 4.7x0.38 inches<br>320 ohm, 5.1kohm  | NA<br>NA             | <a href="https://www.amazon.com/Riyipalib-Curtain-Shunt-Resistor-Ammeter/dp/B0WVVFZ5HJ/ref=sr_1_4?crid=PRMG2XIGTCY&amp;kgwpr_id=B0WVVFZ5HJ&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4">https://www.amazon.com/Riyipalib-Curtain-Shunt-Resistor-Ammeter/dp/B0WVVFZ5HJ/ref=sr_1_4?crid=PRMG2XIGTCY&amp;kgwpr_id=B0WVVFZ5HJ&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4</a>       |
| Power Pole Connectors                                                                                         |                                                                                                                                                                                                                       | 1                | Powerex                       | 26.36                      | Buy                      |        | Received                                    | 1-1/8 inch hole                      | 840128906057         | <a href="https://powerex.com/pnapole/pole-mount-motor-powerpole-black-dust-protector/dp/B07R3D7BFL/ref=sr_1_3?kgwpr_id=B07R3D7BFL&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4">https://powerex.com/pnapole/pole-mount-motor-powerpole-black-dust-protector/dp/B07R3D7BFL/ref=sr_1_3?kgwpr_id=B07R3D7BFL&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4</a>                         |
| Breadboard (soldable)                                                                                         | used as junction for all wiring going to ADC and Pi                                                                                                                                                                   | 1                | Amazon                        | 12.78                      | Buy                      |        | Received                                    | 3.2x2.1x0.3 inches                   | NA                   | <a href="https://www.amazon.com/Q3-Power-Terminid-Automotive-Amplifier/dp/B07V6E5N3H/ref=sr_1_L2_sps?cmrd=Q2TZGNZYASN&amp;kgwpr_id=B07V6E5N3H&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4">https://www.amazon.com/Q3-Power-Terminid-Automotive-Amplifier/dp/B07V6E5N3H/ref=sr_1_L2_sps?cmrd=Q2TZGNZYASN&amp;kgwpr_id=B07V6E5N3H&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4</a> |
| Wiring                                                                                                        | Used to connect wind turbine to shunt resistors<br>Touch screen display living within the enclosure to visualize data                                                                                                 | NA               | Amazon                        | 10.43                      | Buy                      |        | Received                                    | 12 gauge, 10 foot                    | NA                   | <a href="https://www.amazon.com/Rugbyber-Pi-1-Touchscreen-Display/dp/B0ISOR2AN">https://www.amazon.com/Rugbyber-Pi-1-Touchscreen-Display/dp/B0ISOR2AN</a>                                                                                                                                                                                                                                                                                                                                                               |
| Display<br>Keyboard<br>Mouse<br>Cup Anemometer                                                                | To interact with system<br>To interact with system<br>To Measure Wind Speed, 40 NRG                                                                                                                                   | 1<br>1<br>1<br>1 | Napulus<br>Surplus<br>Adafuit | NA<br>10.10<br>66.69       | Buy<br>Buy<br>Buy        |        | Received<br>Received<br>Ordered             | 7.62x5.4 x3.75x0.75 inch<br>NA<br>NA | NA<br>NA<br>NA       | <a href="https://www.amazon.com/dp/B001J06MCP?pcst=frtppgs_yg2ot_dtl_b_product_details">https://www.amazon.com/dp/B001J06MCP?pcst=frtppgs_yg2ot_dtl_b_product_details</a>                                                                                                                                                                                                                                                                                                                                               |
| Temperature Probe<br>Pressure Sensor (list)<br>Pressure Sensor                                                | Measure temperature<br>Measure Pressure<br>provides voltage and current for Raspberry Pi to measure                                                                                                                   | 1<br>1<br>1      | Amazon<br>Amazon<br>Amazon    | 8.39<br>9.02<br>17.61      | Buy<br>Buy<br>Buy        |        | Received<br>Received<br>Ordered             | NA<br>NA<br>GY-SHT30-D               | DS18B20<br>NA        | <a href="https://www.dfrobot.com/product/18933">https://www.dfrobot.com/product/18933</a>                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Wind Turbine                                                                                                  | 1/2in male NPT thread, nylon, used for keyboard/mouse, and temperature/humidity snesor                                                                                                                                | 1                | Amazon                        | NA                         | Buy                      |        | NA                                          | NA                                   | NA                   | <a href="https://www.ufotonicsdirect.com/dropshipping/cntrlng/groccss_controller/vs_microsensor/fvcl_vssensor_vs_controllers/fvcl_vssensor_vs_controllers_accsensorsic-100-1001">https://www.ufotonicsdirect.com/dropshipping/cntrlng/groccss_controller/vs_microsensor/fvcl_vssensor_vs_controllers/fvcl_vssensor_vs_controllers_accsensorsic-100-1001</a>                                                                                                                                                             |
| Cable Glands                                                                                                  |                                                                                                                                                                                                                       | 2                | AutomationDirect              | 17.36                      | Buy                      |        | Received                                    | 1/2 inch thread                      | LMS0-1001            | <a href="https://www.amazon.com/DIY-Prebuilt-Circuit-Curvature-Adjustable-Rugbyber/dp/B07S06CC3G/ref=sr_1_L2_sps?kgwpr_id=B07S06CC3G&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4">https://www.amazon.com/DIY-Prebuilt-Circuit-Curvature-Adjustable-Rugbyber/dp/B07S06CC3G/ref=sr_1_L2_sps?kgwpr_id=B07S06CC3G&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4</a>                   |
| Voltage Amplifier                                                                                             | amplify voltage from RPI for anemometer                                                                                                                                                                               | 1                | Amazon                        | 10.39                      | Buy                      |        | Received                                    | NA                                   | NA                   | <a href="https://www.amazon.com/VIN-3-D-Mount-Cable-Power-board/dp/B0WVVFZ5HJ/ref=sr_1_L4?crid=PRMG2XIGTCY&amp;kgwpr_id=B0WVVFZ5HJ&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4">https://www.amazon.com/VIN-3-D-Mount-Cable-Power-board/dp/B0WVVFZ5HJ/ref=sr_1_L4?crid=PRMG2XIGTCY&amp;kgwpr_id=B0WVVFZ5HJ&amp;pf_rd_p=552415428&amp;pf_rd_i=modular-kit-protection-device&amp;psc_rd_c=2&amp;adsource=AC2C114&amp;srt=1-4</a>                       |
| USB Pannel mount Ports<br>USB C enclosure power port<br>Raspberry Pi mounting screws<br>Shunt mounting screws | usb ports for keyboard and mouse or usb drive<br>used to easily plug in power for the raspberry pi<br>used to mount RPI to base<br>Used to raise the Pi above the enclosure floor<br>Used to mount shunts to subpanel | 1<br>4<br>4<br>4 | Amazon<br>Adafuit<br>NA<br>NA | 11.11<br>17.61<br>NA<br>NA | Buy<br>Buy<br>Buy<br>Buy |        | Received<br>Ordered<br>Received<br>Received | NA<br>NA<br>NA<br>NA                 | NA<br>NA<br>NA<br>NA | <a href="https://www.dfrobot.com/product/4218">https://www.dfrobot.com/product/4218</a>                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Screen Base<br>Sub plate<br>Tripod                                                                            | To mount screen and keep separetly from inside of enclosure<br>Used to mount all components in enclosure<br>used as additional part to enhance usability                                                              | 1<br>1<br>1      | N/A<br>NA<br>NA               | N/A<br>NA<br>NA            | Make<br>Make<br>NA       |        | Building (33%)<br>Building (65%)<br>NA      | 8x6<br>NA<br>NA                      | NA<br>NA<br>NA       | <a href="https://www.dfrobot.com/product/4218">https://www.dfrobot.com/product/4218</a>                                                                                                                                                                                                                                                                                                                                                                                                                                 |

### 11.3 Appendix C: FMEA

| FMEA (Failure Modes and Effects Analysis)           |                                  |                                 |                                                                  |                                                       |                                                                                    |                                                                                                     |                    |                                                                                   |
|-----------------------------------------------------|----------------------------------|---------------------------------|------------------------------------------------------------------|-------------------------------------------------------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|--------------------|-----------------------------------------------------------------------------------|
| Product: Wind Turbine Data Acquisition System (PCC) |                                  |                                 | Organization Name : Northern Arizona University Capstone Project |                                                       |                                                                                    |                                                                                                     |                    |                                                                                   |
| #                                                   | Function                         | Potential Failure Modes         | Potential Failure Effects                                        | Potential Causes of Failure                           | Current Process Controls                                                           | Recommend Actions                                                                                   | Responsible Person | Taken Actions                                                                     |
| 1                                                   | Measure Temperature              | Inaccurate Temperatures         | Loss of temperature reading ability                              | Heat flashes, water damage, sunlight on metal probe.  | Connections to RPI will be in enclosure. Calibration will be performed.            | Calibration of probe, keeping probe in buffer to prevent fluctuations.                              | Team               | NA                                                                                |
| 2                                                   | Measure Atmospheric Pressure     | Inaccurate Pressure             | Loss of Pressure reading ability                                 | Damaged or defective pressure sensor                  | Keep pressure sensing board in watertight enclosure                                | Calibration of pressure sensor                                                                      | Team               | NA                                                                                |
| 3                                                   | Measure Wind Speed               | Turbulence from wind turbine    | Inaccurate wind speed and rpm measurements                       | Turbulence in wind caused by wind turbine             | Fluid dynamics calculations to ensure proper height of anemometer                  | NA                                                                                                  | Team               | NA                                                                                |
| 4                                                   | Measure Voltage                  | Inaccurate voltage              | Damage to electrical components                                  | Incorrect or defective resistors                      | Parallel resistors to decrease input voltage                                       | Voltage resistance calculations, potentially fuses                                                  | Team               | Calculations to ensure voltage never reaches damaging voltages, consider fuse     |
| 5                                                   | Measure Current                  | Inaccurate Current              | Damage to electrical components                                  | Incorrect shunt resistor, voltage spike               | Raspberry Pi code will interpret voltage drops and alert high voltage and current. | NA                                                                                                  | Team               | Calculations to ensure correct shunt resistors                                    |
| 6                                                   | Reduce Input Voltage             | Failure to reduce voltage       | Damage to electrical components                                  | Voltage spikes, incorrect resistors, failed resistors | Raspberry Pi code will interpret voltage and alert high voltage.                   | NA                                                                                                  | Team               | Calculations for resistor choice, research into best resistor material for system |
| 7                                                   | Convert Analog to Digital Signal | Damage from overcharge (max 5v) | Overcharge from wind turbine voltage input                       | Incorrect resistor choice                             | Raspberry Pi code will interpret voltage and alert high voltage.                   | Use correct resistors, perform calculations for multiple combinations, test before charging circuit | Team               | Calculations for resistance                                                       |

## 11.4 Appendix D: Python Code

### 11.4.1 Main Python File

```
import csv
from time import gmtime, strftime
import time
from wlthermsensor import WlThermSensor
import board
import adafruit_mpl3115a2
import sys
import csv
import os
import ADS1263
import RPi.GPIO as GPIO

sensor = WlThermSensor()

fieldnames = ["time", "temperature", "pressure", "Wind Speed", "Voltage in", "10A
Current", "30A Current"]

i2c = board.I2C()
sensor2 = adafruit_mpl3115a2.MPL3115A2(i2c)
pressure = sensor2.pressure
i2c = board.I2C()
pressure = (sensor2.pressure)/1000
sensor2.sealevel_pressure = 1013

with open('Temperature_And_Pressure.csv', 'w') as csv_file:
    csv_writer = csv.DictWriter(csv_file, fieldnames = fieldnames)
    csv_writer.writeheader()

REF = 5
REF2 = 2.5
TEST_ADC2 = True

try:
    ADC = ADS1263.ADS1263()
    if (ADC.ADS1263_init_ADC1('ADS1263_7200SPS') == -1):
        exit()
    ADC.ADS1263_SetMode(0)
    if (TEST_ADC2):
        while(1):
            temperature = sensor.get_temperature()
```

```

        ADC_Value = ADC.ADS1263_GetAll()
        for i in range(6,7):
            if(ADC_Value[i]>>31 ==1):
                Vin_WindSpeed1 = (strftime("%H:%M:%S", gmtime()),",","speed%d =
%lf" %(7, (ADC_Value[7] * REF / 0x7fffffff)*100))

            else:
                Vin_WindSpeed2 = []

        with open('Temperature_And_Pressure.csv', 'a') as csv_file:
            csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
            info = {
                "time": strftime("%H:%M:%S",gmtime()),
                "temperature": 0.6583*temperature+6.9051,
                "pressure": pressure,
                "Wind Speed": (((26.789*(ADC_Value[7] * REF /
0x7fffffff)-11.335))),
                "Voltage in": ((ADC_Value[6] * REF / 0x7fffffff)*100),
                "10A Current": (((ADC_Value[1] * REF2 /
0x7fffffff)-(ADC_Value[0] * REF2 / 0x7fffffff))/0.0075),
                "30A Current": (((ADC_Value[3] * REF2 /
0x7fffffff)-(ADC_Value[2] * REF2 / 0x7fffffff))/0.0025)
            }
            csv_writer.writerow(info)
            print(strftime("%H:%M:%S",gmtime()),
0.6583*temperature+6.9051, pressure, (26.789*(ADC_Value[7] * REF /
0x7fffffff)-11.335), (0.964*(ADC_Value[6] * REF /
0x7fffffff)*100)-0.3947,(((ADC_Value[1] * REF2 / 0x7fffffff)-(ADC_Value[0] * REF2 /
0x7fffffff))/0.0075),(((ADC_Value[3] * REF2 / 0x7fffffff)-(ADC_Value[2] * REF2 /
0x7fffffff))/0.0025))

            time.sleep(0.25)

        ADC.ADS1263_Exit()

except IOError as e:
    print(e)

except KeyboardInterrupt:
    print("ctrl + c:")
    print("Program end")
    ADC.ADS1263_Exit()
    exit()

```

### 11.4.2 ADS1263

This code is very lengthy and will not be added to this document, this code was used from Waveshare's website and can be found by following the link in reference #

### 11.4.3 Config

```
import os
import sys
import time

class RaspberryPi:
    # Pin definition
    RST_PIN      = 18
    CS_PIN       = 22
    DRDY_PIN     = 17

    def __init__(self):
        # SPI device, bus = 0, device = 0
        import spidev
        import RPi.GPIO

        self.GPIO = RPi.GPIO
        self.SPI = spidev.SpiDev(0, 0)

    def digital_write(self, pin, value):
        self.GPIO.output(pin, value)

    def digital_read(self, pin):
        return self.GPIO.input(pin)

    def delay_ms(self, delaytime):
        time.sleep(delaytime / 1000.0)

    def spi_writebyte(self, data):
        self.SPI.writebytes(data)

    def spi_readbytes(self, reg):
        return self.SPI.readbytes(reg)

    def module_init(self):
        self.GPIO.setmode(self.GPIO.BCM)
        self.GPIO.setwarnings(False)
```

```

self.GPIO.setup(self.RST_PIN, self.GPIO.OUT)
self.GPIO.setup(self.CS_PIN, self.GPIO.OUT)

self.GPIO.setup(self.DRDY_PIN, self.GPIO.IN, pull_up_down=self.GPIO.PUD_UP)
self.SPI.max_speed_hz = 200
self.SPI.mode = 0b01
return 0;

def module_exit(self):
    self.SPI.close()
    self.GPIO.output(self.RST_PIN, 0)
    self.GPIO.output(self.CS_PIN, 0)

if os.path.exists('/sys/bus/platform/drivers/gpiomem-bcm2835'):
    implementation = RaspberryPi()

for func in [x for x in dir(implementation) if not x.startswith('_')]:
    setattr(sys.modules[__name__], func, getattr(implementation, func))

```

## 11.4.4 Plotting

```

from matplotlib.animation import FuncAnimation
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import csv
import pandas as pd
import numpy as np

def animate(i):
    data = pd.read_csv('Temperature_And_Pressure.csv')
    x = data['time']
    y1 = data['temperature']
    y2 = data['pressure']
    y3 = data['Wind Speed']
    y4 = data['Voltage in']
    y5 = data['10A Current']
    y6 = data['30A Current']

    ax.cla()
    ax.plot(x, y1)
    ax.tick_params(labelrotation=45)
    ax.set_title("Temperature")

```

```

ax.tick_params(labelsize = 5)

ax1.cla()
ax1.plot(x, y2)
ax1.tick_params(labelrotation=45)
ax1.set_title("Pressure")
ax1.tick_params(labelsize = 5)

ax2.cla()
ax2.plot(x, y3)
ax2.tick_params(labelrotation=45)
ax2.set_title("Wind Speed")
ax2.tick_params(labelsize = 5)

ax3.cla()
ax3.plot(x, y4)
ax3.tick_params(labelrotation=45)
ax3.set_title("Voltage in")
ax3.tick_params(labelsize = 5)

ax4.cla()
ax4.plot(x, y5)
ax4.tick_params(labelrotation=45)
ax4.set_title("10A Current")
ax4.tick_params(labelsize = 5)

ax5.cla()
ax5.plot(x, y6)
ax5.tick_params(labelrotation=45)
ax5.set_title("30A Current")
ax5.tick_params(labelsize = 5)

plt.tight_layout()

def animate1(frame):
    ax.set_xlim(left=0, right=frame)

fig = plt.figure()
ax = fig.add_subplot(2,3,1)

```

```
ax1 = fig.add_subplot(2,3,2)
ax2 = fig.add_subplot(2,3,3)
ax3 = fig.add_subplot(2,3,4)
ax4 = fig.add_subplot(2,3,5)
ax5 = fig.add_subplot(2,3,6)

ani = FuncAnimation(plt.gcf(), animate, interval=100)
plt.show()
```