**Wind Turbine Machine and Environmental Monitoring System**

**ECE 480 Senior Design Project Final Report: Team #3**

**Fall 2021**

Isabel Woelke, Davis Brown, Artan Tagani,

Shadman Chowdhury, Ryan Doyle, Devin Abb

Sponsor: Tecnix

Facilitator: Dr. Mahapatra

**Submission Date:** December 8th, 2021

**Table of Contents**

1. Introduction…………………………………………………………..3
   1. Project Description…………………………………………….3
   2. Customer Requirements……………………………………….3
2. Project Background…………………………………………………..3
   1. Prior Work and Shortcomings…………………………………3
3. Design Options…………………………………………………….....5
   1. Design Constraints…………………………………………….5
   2. Design Evaluations……………………………………………5
4. Final Design………………………………………………………….6
   1. System Overview……………………………………………...6
   2. Hardware Design……………………………………………...6
   3. Software Design……………………………………………....7
   4. Graphical User Interface……………………………………...7
   5. Block Diagram………………………………………………..7
   6. Operation Principles…………………………………………..9
5. Gantt Chart…………………………………………………………..9
   1. Gantt Chart Diagram………………………………………….9
   2. Assessment on Management and Coordination ……………...10
6. Budget……………………………………………………………….10
7. Team Roles…………………………………………………………..12
8. Project Conclusion…………………………………………………..13
   1. Results/Executive Summary...………………………………..13
   2. Suggestions for Future Work………………………………....13
9. References…………………………………………………………...14
10. Appendices………………………………………………………….14
11. Downloading Logs……………………………………………14
12. How to Move Files From PC to Pi using FTP………………..14
13. Voltage Divider & Shunt Configuration & Testing…………..15

**1 Introduction**

**1.1 Project Description**

Tecnix is a company that was created by an entrepreneur about ~10 years ago, who saw an opportunity to utilize the wind power in his community by building wind turbines. These wind turbines are unique in their size, which is ~7.5kW, and because they are locally produced in Tanzania, Africa. Recently Tecnix was purchased by Justin Heath, a Kenyan entrepreneur and businessman. Tecnix has working prototypes of the wind turbine system and the data loggers that are used to record important information like voltage, current, turbine speed, etc. These prototypes have some shortcomings and inefficiencies that Tecnix is hoping to improve to get better working prototypes, especially focused on the data loggers.

**1.2 Customer Requirements**

The customer requirements are organized by ‘3 Ds’ and ‘2 Cs.’ The customer requires *digestible data*, these data loggers can record data from the wind turbines at any time, but the customer requires useful data from the plethora of accessible data. The customer also requires the team to focus on *design cost.* In order to provide inexpensive services and to be a profitable business, the customer requires the data loggers to be at most $250/logger. The customer also requires that the team focuses on *design principles*. The customer requires that the team focuses on the differences between the high tech components and the low tech components to provide lasting improvements to the prototypes.

In terms of organizing the data, the customer requires separable *company data* and *customer data*. The customer requires the team to effectively gather and display results that are important for the company to use and important for the customer to know separately.

**2 Project Background**

**2.1 Prior Work and Shortcomings**

The wind turbines are fully functioning and are currently being sold and implemented in the target market of East Africa (Tanzania). They generate an average of ~7.5kW of power, and come equipped with a data logging system, dumpload, rectifier, and battery system depending on the needs of the customer. The turbines are built with a custom rotor and stator that is designed and produced in East Africa. Since the wind turbine design is complete and proven to be reliable, the main work is with the logger because it gathers information that is useful to the company for generating data to help sell the turbines, and also to the customers to view the performance of their turbine.

The data logging system features the Adafruit M0 + WiFi as its main board, and an Adalogger module which has an SD card with a RTC (real-time clock) for data storage. There is a small screen with 3 buttons that allows you to navigate the menu system of the data logger. The menu system allows you to view: the voltage meter, wind monitor, WiFi connection, dump load, I/V battery, I/V turbine, frequency, and firmware version. The data logging system creates its own WiFi hotspot, so you can wirelessly connect to it and download the information stored on the SD card. There is also an ethernet port that you can use to download the stored logs. For the data measurements, there is an ADC (analog-to-digital converter) that connects to the main board via I2C. There is a small transformer that converts the high input voltage down to the supply voltage, which allows it to take AC measurements of the turbine without destroying the modules.

The first version of the data logging system and firmware is already complete, however Tecnix wants to expand the features of the device to include; temperature, humidity, and blade/wind speed. Another shortcoming is the data accessibility and usage. Tecnix would like to have a user-friendly interface that presents digestible data so that the performance of the turbines can be measured and advertised for future sales. The interface for viewing the data needs to be easily accessible for both customers and Tecnix technicians. Another aspect of the system that needs to be improved is the voltage and current measurement accuracy using the ADC.

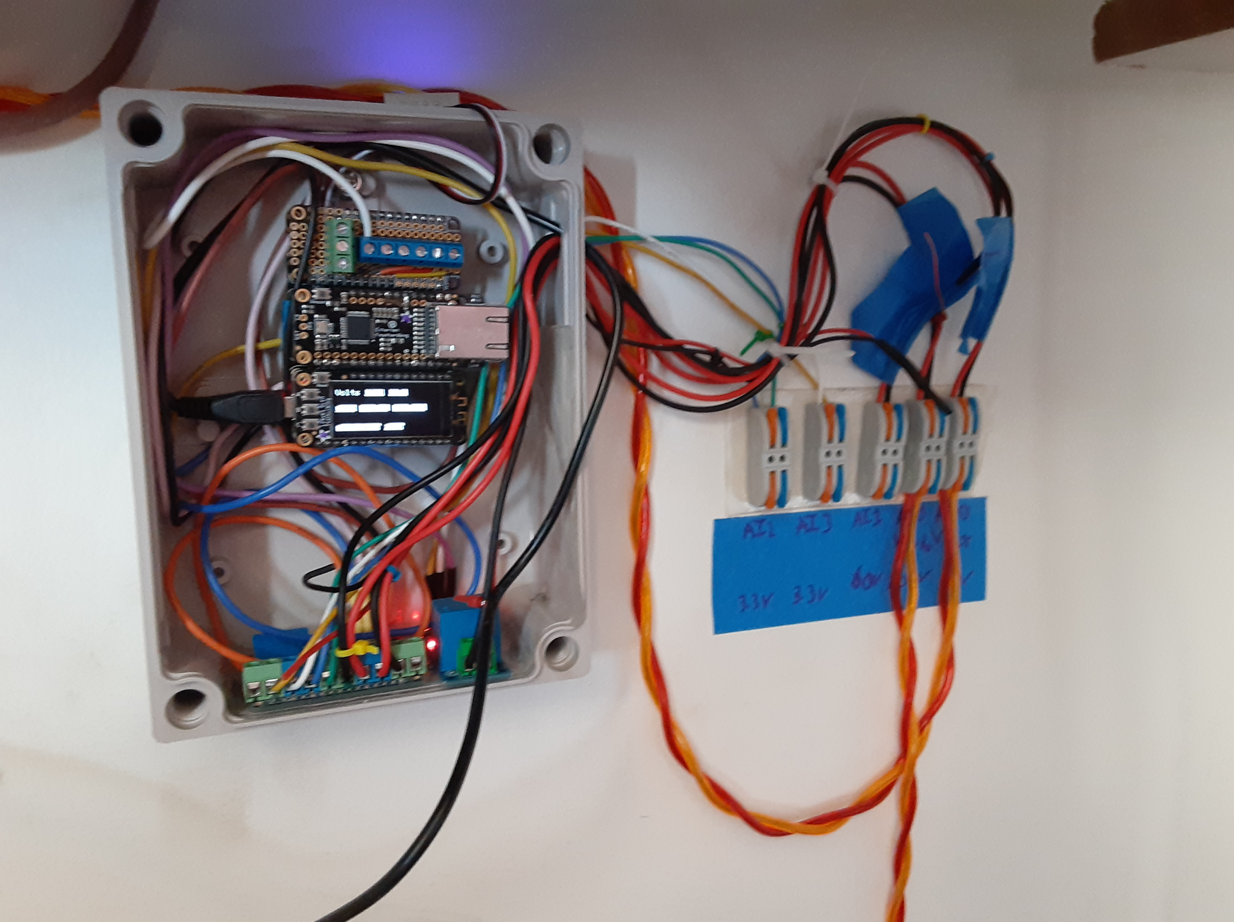


Figure 1: Installation of the Data Logger

**3 Design Options**

**3.1 Design Constraints**

There are a few constraints in place to support the current business model. For instance, the turbines need to be manufactured in East Africa. This means that any modules that we decide to implement need to be accessible to Africa and not have extreme shipping costs. With that being said, another design constraint is cost. A big selling point for Tecnix is that the turbines are affordable and cheaper than the competition, so we need to keep unnecessary costs to a minimum while maintaining quality and reliability. Within those main constraints, we can modify the system as much as necessary.

**3.2 Design Evaluations**

Tecnix provided the team with a data logger prototype which records, stores, and monitors wind turbine data such as battery voltage and current, turbine voltage output, dumpload current, and ideally monitoring weather conditions. The prototype model we were provided only outputs an array of hard to understand numbers and does not monitor any conditions. Tasks for this aspect of the project include implementing a new sensor to monitor humidity and temperature, and designing subsystems to measure voltage and current of the battery system. The team decided to use the temperature and humidity sensor (SHT30) to measure weather and box conditions which is the sensor that Tecnix suggested we use. The team decided to implement a voltage divider to bring the voltage from the battery to a safe level for the ADC to use. This was deemed the best solution to measure the voltage of the battery as the voltage of the battery without the voltage divider would be too large to pass through the ADC. The team decided to implement a shunt resistor to measure the current coming from the battery. This was deemed the best solution for this task as the current of the battery would be large and the shunt resistor is the most practical way to measure large currents.

Tecnix suggested that the team create a database for the history of data measurements to be stored in for future analysis and reference. We received and reviewed multiple options regarding how to go about this, and we settled upon using a Raspberry Pi-0 Microprocessor. We settled upon the Pi for a few reasons. One of them was that the company code could be modified to include a library for MySQL, which is what we used as a container and editor for the data. Another reason was that, at the time, they were relatively cheap. Another reason was that we needed something that could host a web server, and the Pi is capable of this when there is Internet connection. However, it appeared that the company M0, while being able to create its own hotspot, does not create Internet access, and our sponsor said this would be a good suggestion and responsibility of the company in the future. So, for our purposes, we did all of our Pi database/server testing with local Internet rather than connecting to the M0 hotspot. In order to simulate the files that would go into the Pi, we logged onto the hotspot of the M0 and captured actual data logs from it, so the data being put into our server would be fully accurate to the data the company is extracting.

Tecnix also asked the team to create a website that could be hooked up to the database to display the data in an easy to read format for analysis and for non-technical customers and employees to understand. We planned to achieve this originally using an admin capability on the RaspberryPi, but ended up doing it manually by creating a website using HTML and JavaScript, and pushing it onto our web server on the Pi.

**4 Final Design**

**4.1 System Overview**

The system includes voltage, current, temperature, and humidity data collection hardware and software that sends collected data (from turbine, battery, sensors) to a SQL database to be stored. The final element of the system is a web server that organizes the data into readable graphs for customers to understand.

**4.2 Hardware Design**

The system uses an ADC (ADS1115) to convert the analog voltage measurement to a usable digital value. The highest input the ADC can take is 5V, meanwhile the voltage of the battery that the system intends on measuring can be up to 100.0V. To solve this problem the team designed a voltage divider. The voltage divider is rated for 110V and brings it down to a range that is less than 3.3V so that we do not damage the ADC nor the main board (Adafruit Feather M0 is rated for maximum 3.3V) and can get accurate readings. In terms of the software for the ADC, we used the maximum gain that it could provide to use all available bits (15 bits) and yield results with high resolution.

The team added a temperature/humidity sensor to the system. The sensor used is the SHT30, and it outputs a digital value to the main board. The readings are then outputted to the systems logs, as well as the menu screen so that it can be viewed immediately by a user.

To measure the current from the battery the team had to design a shunt resistor system. The shunt is a known resistance connected on the battery’s negative leg. Terminals on the positive and negative poles of the shunt are each connected to an input pin of the ADC. Software on the M0 board takes the difference of the voltage measurements, and using the known resistance, calculates the current of the battery.

**4.3 Software Design**

The data logger firmware prototype was provided by the company. We improved upon the software by making key additions for our new sensor, our database, and for improving the run-time of the code as a whole. The less data we use, the more processing power the logger has. The software was uploaded to the M0 and is responsible for the behavior of the new sensor, hot-spot, digital interface, etc. As the logger saves data it is recording, we decided the best thing to do with the log files in the company’s future was to push them to a web server then display the data on a website within the server through a dynamic chart. All of this is hosted on a RaspberryPi-0.

Our server design implements a Linux, Apache, MySQL, PHP (LAMP) style web server for each individual data logger. We achieve this through using a RaspberryPi-Zero. Apache was first installed on the raspberry pi in order to host a web server to display the data. To make use of commonly used PHP applications on the web server, PHP 7.4 was also installed to the raspberry pi. For the server itself, we installed maria-db, an open source version of MySQL, because it was lighter than MySQL and therefore would run more efficiently on a raspberry pi. After creating a user and securing it, the MySQL connector for PHP was installed.

**4.4 Graphical User Interface**

The graphical user interface was created using Apache and Linux, and managed through PHPMyAdmin. The website on the server was created utilizing HTML5, CSS and Javascript. To add in additional graphing functionalities, we relied on chart.js, which is a free open-source data visualization library. HTML buttons have onclick functions that tell the pi to reconstruct the charts datasets, depending on which button is pressed.

**4.5 Block Diagram**

The following general connectivity diagram [Figure 2] shows major components of the turbine and logger system. The diagram includes relevant system connections and metering points as well as how these points interface with the data logger unit. This diagram represents the general operation of the system, a more detailed schematic with detailed wiring is shown in Figure 3.

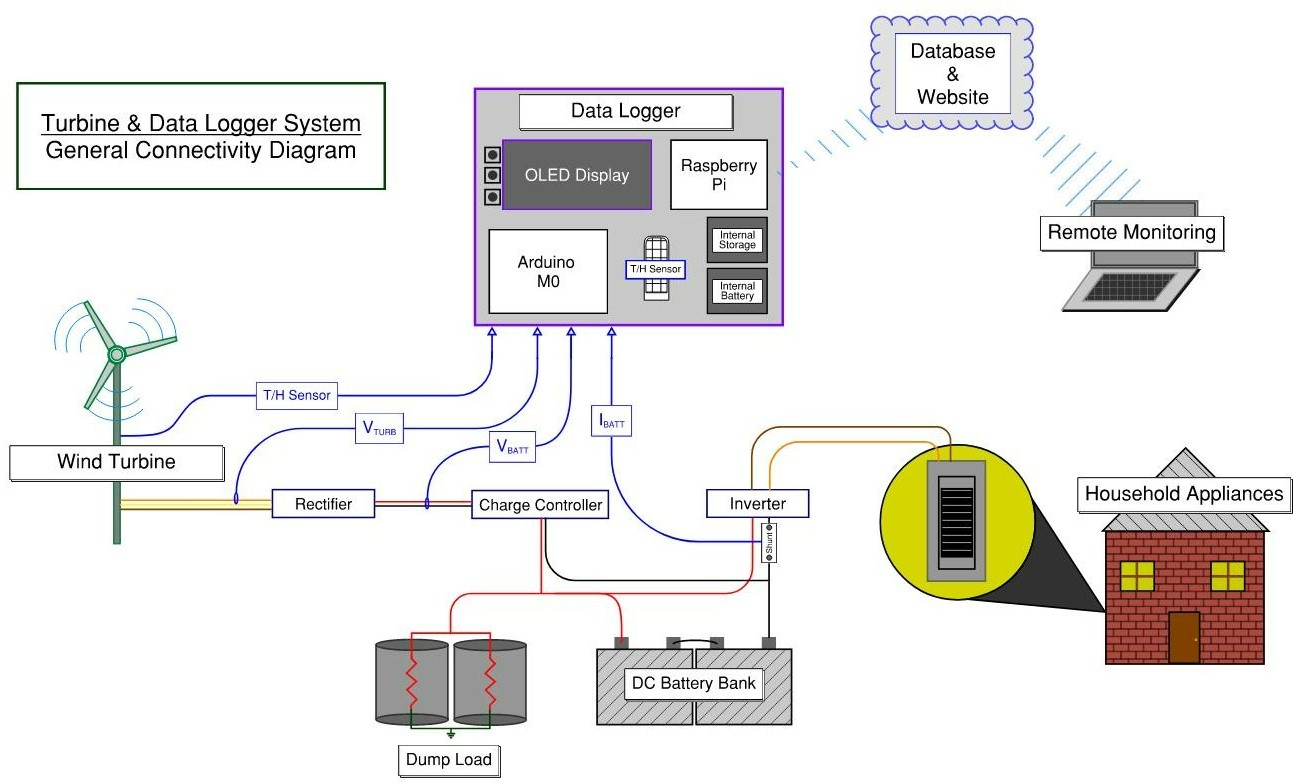


Figure 2: Turbine & Logger General Connectivity Diagram

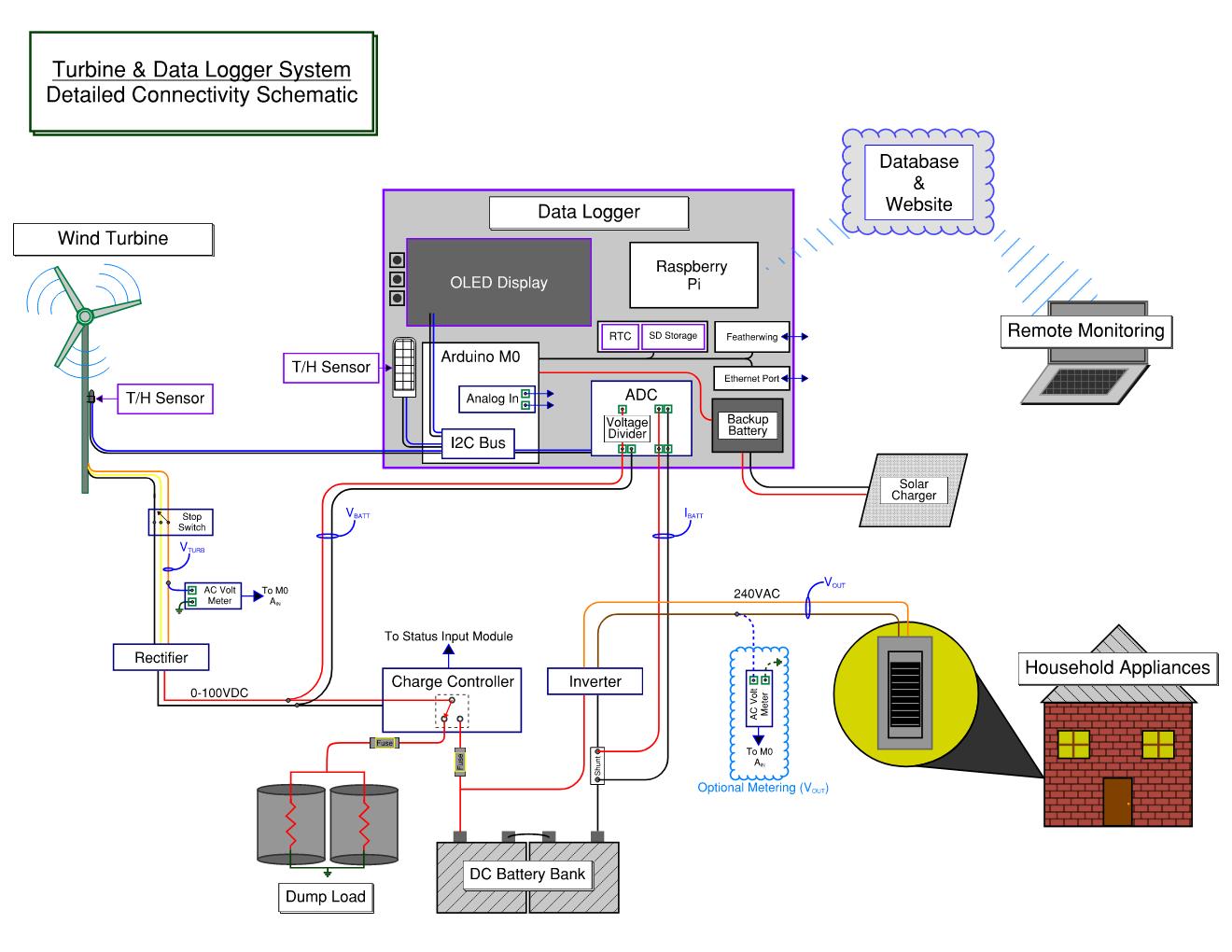


Figure 3: Detailed System Connectivity Schematic

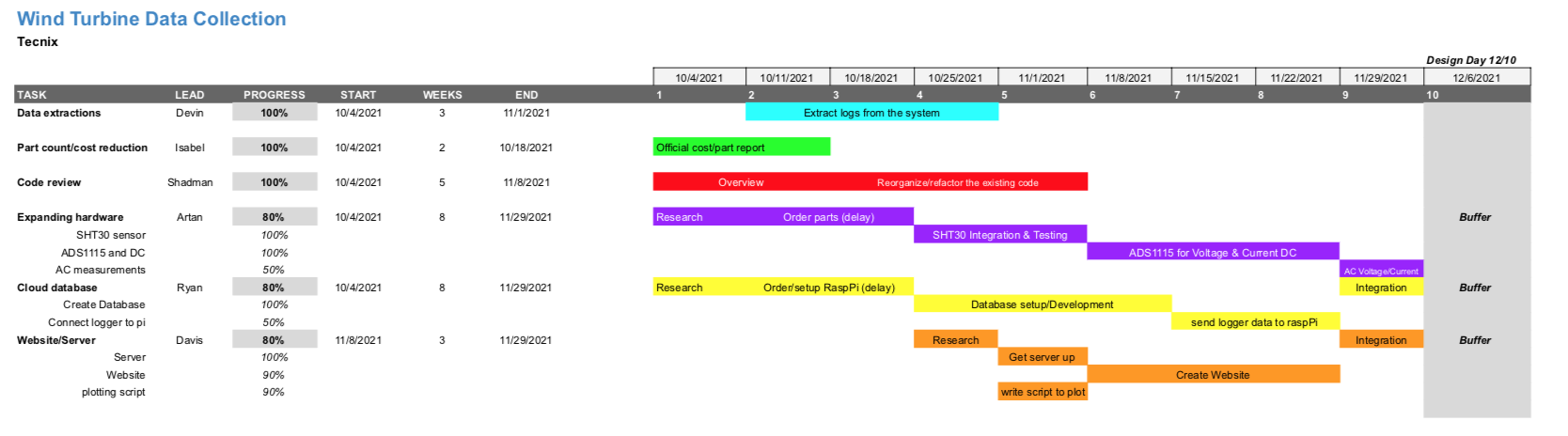
**4.6 Operation Principles**

Assuming favorable conditions, the turbine produces three-phase AC power which is passed through a rectifier. This rectifier converts the generated power to DC in order to charge the battery bank. Outgoing power from the DC battery bank passes through an inverter to create an AC power supply suitable for customer connections.

In cases where the turbine experiences unfavorable conditions, safe operation logic in the system protects both the customer and the turbine equipment. If the battery exceeds its rated voltage while the turbine is still generating power, the charge controller diverts the excess power to a resistive dump load. This high impedance load maintains the current draw of the system, protecting the turbine from overspeed. Although the dump load is a viable solution for most cases, a stop switch affords the turbine additional protection from overspeed. The switch helps to brake the turbine by shorting all three outgoing phases together.

**5 Gantt Chart**

**5.1 Gantt Chart Diagram**



**5.2 Assessment of Management and Coordination**

The management of this project was flexible and we made sure to maintain a good line of communication with each other. Due to this project having many aspects requiring different areas of expertise, we divided the team into two sub-teams; hardware and software. The duties of the two teams were primarily focused on their respective specialization but sometimes branched into the other. For example, the hardware team worked with new sensors and modules but also wrote the software for those. Meanwhile, the software team had a coding emphasis for the website/server/database but also worked with hardware in terms of setting up the raspberry pi and SD card, as well as with connecting the pi to the data logger. One thing that could be improved with our management would be to have assessed which tasks would be most difficult, and allocate more people to that task. For example, transferring data from the logger to the raspberry pi proved to be more of a challenge than anticipated.

**6 Budget**

Below is the budget and pricing information related to the monitoring system. The modules that are needed include ways to store data from the logger, connecting to wifi, and reading temperature and humidity.

| **ITEM** | **COST** | **DESCRIPTION** |
| --- | --- | --- |
| Raspberry Pi Pico RP2040 | $4.00 | Raspberry Pi with two I2C controllers, two SPI controllers and two UARTs. |
| SHT30 Temperature And Humidity Sensor - Wired Enclosed Shell | $8.95 | Temperature/Humidity sensor with ±2% humidity and ±0.5° celsius with I2C UI |
| AM2315 - Encased I2C Temperature/Humidity Sensor | $29.95 | Thermistor temperature sensor along with a humidity sensor with an I2C UI |
| Adalogger Featherwing & Battery | $9.90 | Allows data to be stored onto a micro SD and has an I2C real time clock |
| 128x64 OLED Display | $14.95 | Displays the information on the data logger |
| Panel Volt Meter | $7.95 | Reads the DC voltage |
| Ethernet Featherwing | $19.95 | Expands to allow for an ethernet connection port |
| Featherwing Tripler | $8.50 | Add-on for Featherwing board that allows connecting multiple Featherwing without stacking headers |
| Featherwing Single Proto | $4.95 | Add-on for the Featherwing board to duplicate pins |
| 10’ Ethernet Cable | $2.95 | Ethernet cable to connect to Ethernet FeatherBoard |
| Raspberry Pi Zero WH (Zero W with Headers) | $14.00 | Raspberry Pi to host a web server. (More powerful than Pico) |
| 4 Channel 16 Bit ADC Converter | $14.95 | Converts up to 4 analog signals to digital sent via I2C |
| **TOTAL:** | $174.61 |  |

**7 Team Roles**

| **Person** | **Non-Technical Role** | **Technical Role** |
| --- | --- | --- |
| Artan | Project management: Including charts, timelines, expenditures, managing budget and ensuring steady progress is being made weekly on the project | Expanding the features of the device to be able to record temperature/humidity conditions and DC voltage/current, assist software integration and refining |
| Davis | Presentation prep including creating PowerPoint presentations/posters, collecting data and presenting the project | Handling of the database; more specifically creating the webpage and affiliated code |
| Shadman | Presentation prep including creating PowerPoint presentations/posters, collecting data and presenting the project | Database design, web server administration, Microcontroller set-up, code refactoring and additions, scripting |
| Isabel | Document Preparation: writing the reports and peer editing. Also, in charge of taking notes during the building process and documenting progress | Part count and voltage/data measurements |
| Devin | Lab coordinator. Responsibilities include scheduling the lab times, ordering parts and managing the budget | Part count and cost reduction along with accurate data extraction |
| Ryan | Managed correspondence and scheduling. Point of contact with both the facilitator and sponsor; set up team meetings, company meetings, lab sessions, save contacts and meeting notes, etc. | Database design, web server administration, Microcontroller set-up, code refactoring and additions, scripting |

**8 Project Conclusion**

**8.1 Results/Executive Summary**

The team was able to make significant progress designing hardware subsystems to accurately measure and display the voltage and current of the battery system, creating a large database to store a history of data in, and designing a web server to display customer-based information. The team learned valuable lessons in subsystem design and testing, database creation and manipulation, web server creation and design. Tecnix gained for their company a working data logger and a system to store and display the data collected for their own records and analysis and their customer’s understanding of their products.

**8.2 Suggestions for Future Work**

There are a couple things that could be improved with this system, in regard to both hardware and software. For hardware, there would need to be the implementation of a new AC current sensor because the one that we attempted to use runs on a 5V supply but our board has a max rating of 3.3V. This adds unnecessary complexity to the system because we need to derive an external 5V supply for the sensor. After adding an AC current sensor, a desired deliverable would be to then calculate the power factor of the turbine (there is already an AC voltage sensor being implemented). Additionally, the implementation of another temperature and humidity sensor to measure the environmental conditions outside where the turbine is located would be useful because the weather conditions affect turbine performance.

Another aspect that will need to be addressed is the overall integration of the system. There are numerous components that need to be communicating with each other (M0, adalogger, raspberry pi, database, server, website), so there needs to be an efficient method of doing this so that the system performance isn't affected. We attempted to make a query to our MySQL server through the data logger software, but a connection could not be finalized as the updates caused the logger to freeze. Future work may also include implementing components from the whole data logging system onto a PCB to reduce the system’s cost and physical footprint. Honing in on a clear cut goal for where the product is headed and what is expected of the next team well beforehand is a crucial step for the company and the students.

**9 References**

1. Dexter Industries. 2021. *Transfer Files Between Your PC and Your Raspberry Pi*. [online].
2. Piggott, H. (2014). *A Wind Turbine Recipe Book* (English Units Edition). Hugh Piggott.

**10 Appendices**

**Appendix A**

How To Download Logs

Go to your laptop’s wifi settings and find the data logger wifi named “Turbine Sensor.” Enter the password when prompted and connect. Open a browser and enter the device IP address (192.168.1.1) You will then be able to select which logs you would like to download.

**Appendix B**

How to Move Files From PC to Pi using FTP

Install the program “[FileZilla](https://filezilla-project.org/)”. Then, make sure you’ve established a connection between your PC and RaspberryPi. Next, connect to your Pi using the top navigation bar in FileZilla, as shown in Figure A. You will need to know your credentials to login to the Pi.

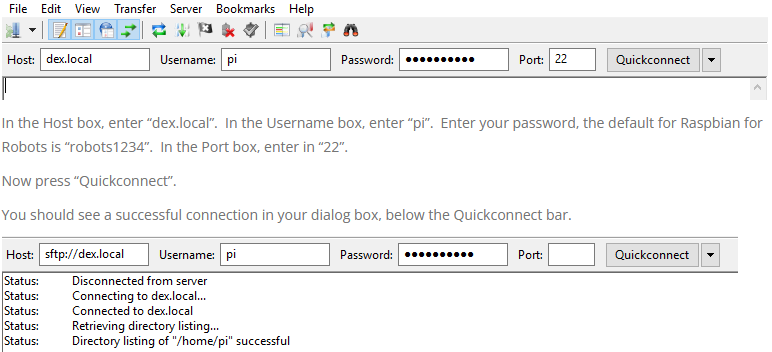


Figure A

Then, it is as simple as dragging and dropping files with your cursor to whichever destination you choose.

**Appendix C**

Voltage Divider & Shunt Resistor Configuration & Calculations

Due to the high power present in the system and the limited capabilities of available testing equipment our team had to scale and simulate the voltage divider and shunt resistor elements of the system.

1. Voltage Divider

The resistors that comprise the voltage divider are soldered to the ADC module and are configured to step voltages up to 100VDC down to below 3.3V. This configuration is accurately scalable from low testing voltages to the highest voltages of the system. The following diagram [Figure B1] shows the calculation process as well as the testing schematic.

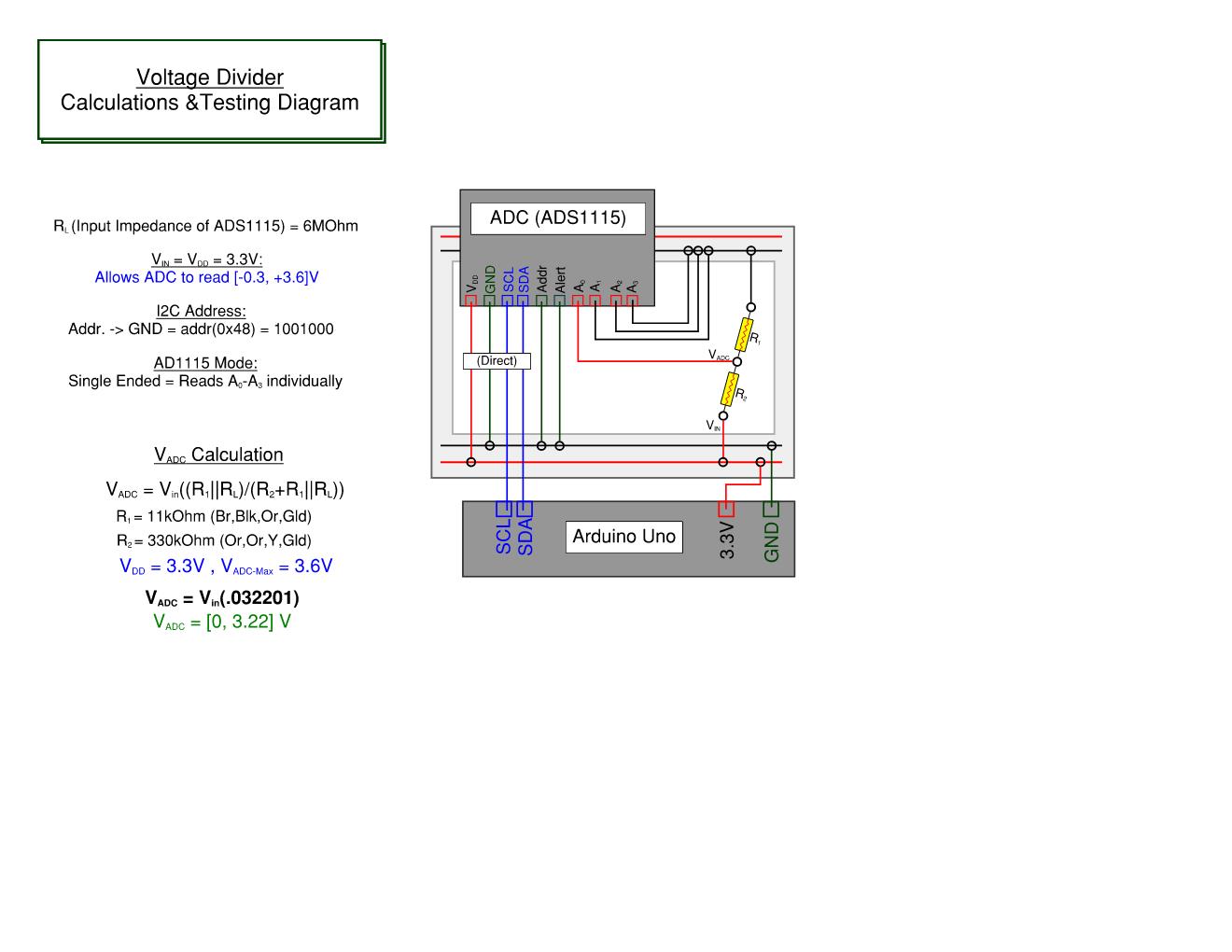


Figure B1: Voltage Divider Calculations & Testing Diagram

1. Shunt Resistor

In order to properly test the functionality of the shunt resistor system, the team simulated the battery load and shunt resistor using a similar configuration as the voltage divider. The testing power supply was current-limited at 100mA and represents 100A through the line. Subsequently the shunt resistor was increased from 0.1 Ohm to 10 Ohm. The figure below [Figure B2] shows the testing configuration with relevant calculations and notes.

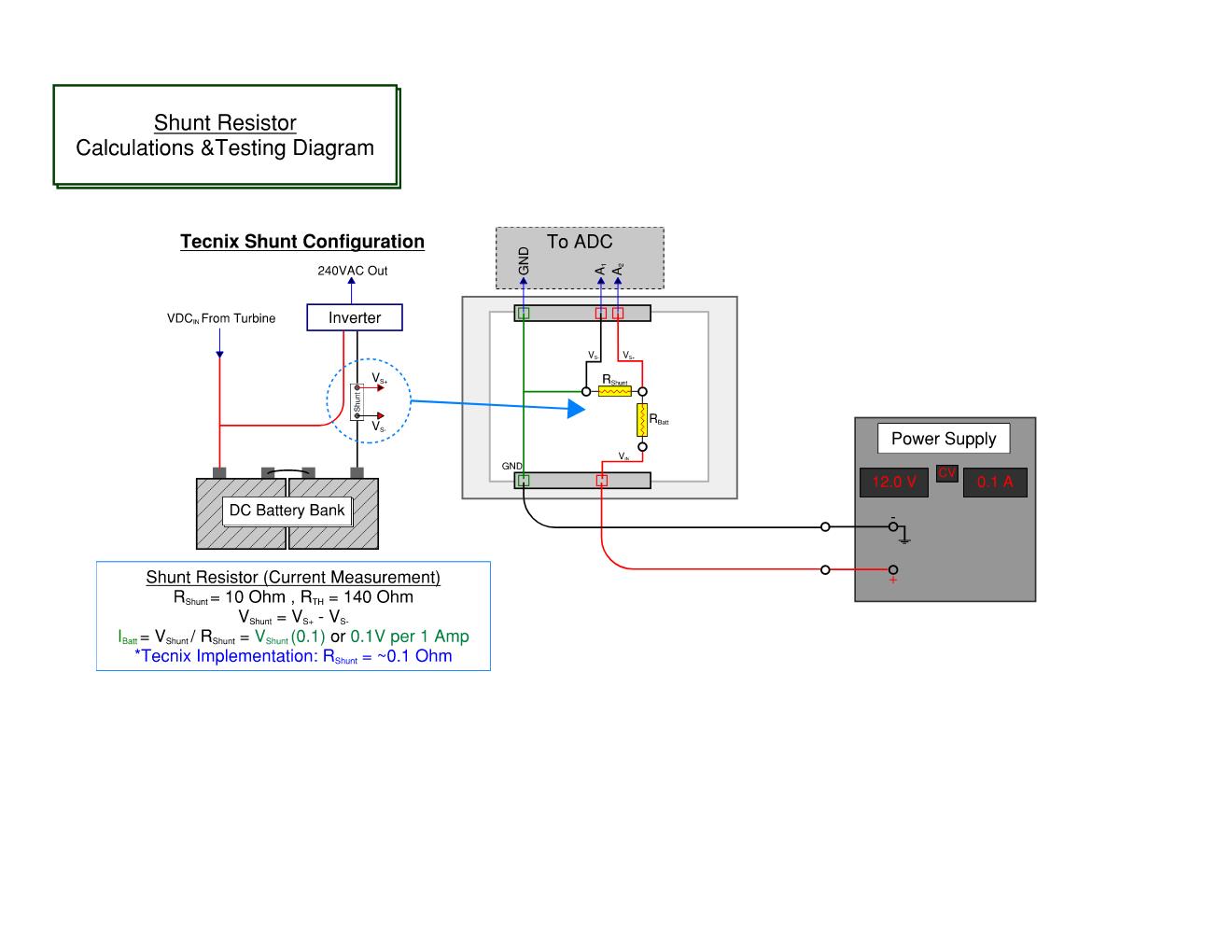


Figure B2: Shunt Resistor Calculations & Testing Diagram