University of Puerto Rico
Mayagüez Campus

Wind Tunnel Telemetry (WindTel)
Final Report

Prepared for:
Isidoro Couvertier, Nayda G. Santiago, Victor E. Lugo

Prepared by:
Luis O. Vega Maisonet (Project Manager)
Nelson G. Rodríguez Ortiz
Misael Valentín Feliciano (Quality Manager)
Kahlil J. Fonseca García

Customer:
Raúl Zapata López

Advisor:
Manuel Jiménez

Executive Summary <- *(Nelson)*

The WindTel system consists of an automated data acquisition system for the aerodynamic and environmental measurements in experiments performed on solid objects with the wind tunnel of the UPRM Civil Engineering and Surveying Department. Specifically, the system is capable of acquiring wind speed, pressure distribution, wind forces, temperature and humidity readings, and controlling the wind speed inside the wind tunnel. Additionally, the measurements obtained from the experiments can be stored in a database server and accessed by the user through a mobile/desktop application. The project deliverables include an embedded system hardware compacted in a 7 in x 11 in plastic enclosure and a mobile/desktop application that can be downloaded from Google Play Store to any Android device. In terms of problems and difficulties during the final phase of the project, the Empotrao' Solutions team overcame functionality problems in some hardware modules. The Control Module was redesigned completely due to problems when interfacing with the wind tunnel motor variable speed driver. The Wi-Fi module was consolidated with the Dynamic Module to minimize Master Module processing load. The Pressure Module was redesigned to acquire pressure measurements in 16 different points while reducing power consumption. The Balance Module was redesigned to obtain stable and accurate force measurements from the signal conditioning circuits installed in the mechanical balance. Furthermore, the team dealt with work environment problems in the Wind Tunnel Laboratory due to the lack of work space and equipment in the laboratory. The team managed to alleviate this difficulties by asking permission to borrow essential equipment from the Capstone and Microprocessing Interfacing laboratories. A more detailed description of each problem and the contingency measures carried out during the implementation and integration phases is presented in the Design Criteria and Specifications section.

In terms of expenditures, the WindTel project has incurred $17,857.59 in actual costs, in comparison to the $24,693.65 in expected costs which represents a difference of 26.88%. Some of the expenditures include electronic hardware components, printed circuit boards, enclosures and equipment used to install the hardware components in the Wind Tunnel Laboratory. For more information, please refer to Appendix H. In addition to Dr. Raul Zapata Lopez, other potential customers include UPRM students and professors from the Wind Engineering course (INCI6997) and researchers from UPRM or other universities and agencies. The AeroDesign team from the UPRM Mechanical Engineering Department, for example, are interested in testing and characterizing their airplane turbines under different airflow conditions inside the wind tunnel before their next competition next semester. For more on potential customers for the WindTel system, please refer to the Market Overview section.

Table of Contents *<- (Luis)*

1. Introduction <- *(Luis)*

A wind tunnel is a device used to perform aerodynamic characterization of objects that are placed in an airflow environment [39]. Engineers use wind tunnels to acquire aerodynamic measurements such as the forces exerted on an object by the wind, and pressure distributions to improve their designs before deploying them to their expected environment. The Empotra'o Solutions team's client, Dr. Raúl Zapata López, is in charge of the wind tunnel of the UPRM Department of Civil Engineering & Surveying. The wind tunnel represents the only resource capable of simulating the unpredictable conditions of nature which can significantly affect engineering designs.

The UPRM wind tunnel operates manually, resulting in low system usability and low accuracy of data measurements. The Empotra'o Solutions team presents the WindTel project as a solution for the lack of usability and accuracy of the Civil Engineering & Surveying Department's wind tunnel. The solution consists of developing an automated data acquisition system capable of acquiring wind forces and pressure distributions. The two major objectives that the Empotra'o Solutions team must fulfill are the following: to obtain a score of 70 in the System Usability Scale (SUS) [40], and to acquire data measurements within a 10% error margin with respect to baseline valufes established by manual data measuring instrumentation.

This report informs the reader about the completion of the data acquisition process automation and the WindTel system design. The project completion is measured by the design, the implementation, and the integration phases of each WindTel module. By completing the WindTel design, implementation and integration phase the Empotra'o Solutions team achieved a 100% of project completion. The data acquisition automation process is composed of four main components: an embedded system, a database server, and a desktop application. The progress report organization is based on the design of these four components which are divided into hardware modules and software modules. The embedded system is divided into three data acquisition modules: Balance, Dynamic, and Pressure, with one intermediary communication module named Master. The Wi-Fi component specified in the previous report was consolidated with the Dynamic Module to reduced the used of several microcontroller launchpads.

2. Design Criteria and Specifications <- *(All)*

Hardware Modules

1. Master Module <- *(Luis)*

    The wind tunnel sensors and actuators are distributed inside the wind tunnel laboratory approximately 7 ft apart from the researcher's main desk. The Master module which will be established in this desk will serve has a intermediary module where the data acquisition modules (the Balance, the Pressure and the Dynamic) will communicate via a serial communication protocol. The data acquisition modules will be in a master-slave relationship with the Master module. In the Embedded System Design course the Empotra'o Solutions team implemented such communication scheme using the Inter-Integrated Communication (I2C) with a bus network topology. The Empotra'o Solutions team faced challenges in troubleshooting the bus network topology and managing the traffic of information in the I2C bus. While the bus network topology operated adequately for acquiring a single data measurement it did not function properly for data sampling since more than one measurement produced trash in the I2C bus. The past bus network topology was change to a star network topology in conjunction with the Universal Asynchronous Receiver Transmitter (UART) communication.

2. Balance Module <- *(Nelson)*

    In 2013, a group of four computer engineering students developed an electronic balance system for their Embedded Systems Design and Capstone courses. They selected 2mV/V strain gauge-based 5-V load cells that can measure up to 20kg (45lb) of weight forces. They placed the load cells strategically to measure three axes: X, Y and Z. The X and Y axes require two load cells, while the Z axis require only one load cell [3][5]. They implemented a signal conditioning circuit for the load cells using the Texas Instrument (TI) INA129 amplifiers, but unfortunately, most of the circuits are not working as expected because of components fatigue and aging. Their electronic balance measured weight forces to calculate the drag and lift forces of the object under study in the wind tunnel. One of the goals of this project is to redesign Aerobal's electronic balance signal conditioning circuit. During the design phase, the Empotrao' Solutions team calibrated the balance with known weights and characterized the load cells transfer characteristic. For more information about the load cells transfer characteristic please refer to Appendix F. Two alternatives were considered for the Balance Module signal conditioning circuit, one that generates an output voltage swing from 0V to 3.3V and another that generates an output voltage swing from 0V to 5V. Both designs functionality were proved with simulations using PSpice tools and the best alternative was chosen considering complexity, budget and Analog to Digital Converter (ADC) resolution. Please refer to Appendix D for more information on the design of alternatives for the signal conditioning circuit.

3. Pressure Module <- *(Kahlil)*

    The pressure module is responsible of collecting the pressure measurements upon the object to be studied inside the wind tunnel test section. These pressure points will be measured through pressure sensors that will be located inside and outside the wind tunnel. The Arduino Mega 2560 is being used for the interfacing with the 16 barometric pressure sensors.

4. Dynamic Measurements Module <- *(Misael)*

The Dynamic Measurements Module is in charge of acquiring information about environmental variables that affect the measurements taken by the Balance Module and Pressure Module. It achieves this task by interfacing with a DHT22 Temperature & Humidity sensor and an Anemometer Wind Speed sensor. In addition, the Dynamic Measurements Module has been consolidated with the Wi-Fi module. The reason for this is to achieve a higher compactness and to reduce resource wastefulness. Given that the ESP32-DevKitC used for the Wi-Fi connection is in itself a microcontroller with features such as GPIOS, it is well suited to interface with the DHT22 Temperature & Humidity sensor as well as the Anemometer sensor.

Software Modules

1. WindTel Application <- *(Misael)*

The WindTel has an accompanying software application, providing an alternative way to use the system such the design of experiments is more efficient. The WindTel application is cross-platform, and is available to use on desktop, Android smartphone, and Android tablet. The technology of choice was Ionic Framework[42]. Ionic is a framework that facilitates the development of native cross-platform applications, under the philosophy of “Write once, deploy everywhere.”

2. Statistics and Data Visualization <- *(Misael)*

The WindTel Application allows the user to perform statistics and data visualization upon the acquired data measurements. These data visualizations consist of graphs that allow the user to visually inspect relationships among the acquired data. The technology of choice for the data visualization is Chart.js. This allows for the data visualization to be performed client-side[41].

3. Database Server <- *(Kahlil)*

A database server serves as means to store, in this case, the information of the users and the experiment results they wish to store. It has been implemented utilizing Flask, which is a Python Framework for building web apps and REST APIs. Ubuntu linux operating system was used to setup the backend application environment.

3. Methods and approach to the solution <- *(Luis)*

Team Organization

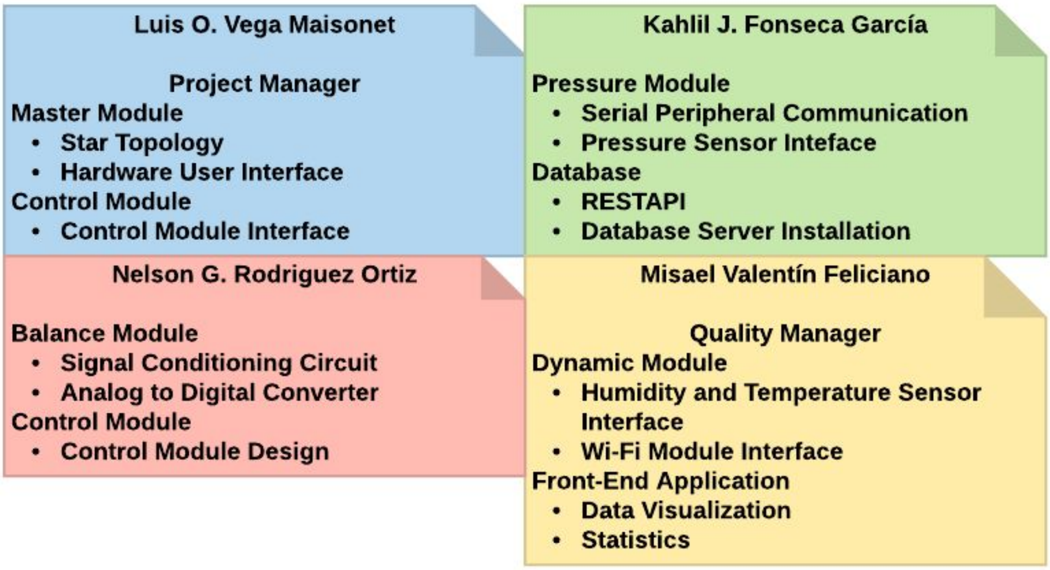


Fig. 1. Team Organization

Account of the Activities of the team in the Project

January

1. Planification and Design
2. System Ideation and Problem Definition

February

1. Design Completion
2. Preliminary Implementation

March

1. Implementation of Master and Dynamic Modules
2. Started Balance and Pressure Modules Implementation
3. Database Implemented and Tested

April

1. Continued Implementation Phase
2. Started Integration Phase
3. Software Application was Developed

May

1. Software and Hardware Integration
2. Product Delivery

Specification Summary

A wind tunnel instrument known as a pitot tube is used to validate the wind tunnel wind speed. Temperature and relative humidity measurements were compared using the Google search engine. The wind exerted forces were validate throw the used of 2.5lb weights. The pressure sensor measurements were compare with the atmospheric pressure.

Changes in Schedule

Due to the various problem the Empotra'o Solutions team was forced to reschedule task. The two main problems that affected the schedule. First the wind tunnel chamber got stuck because of this we were forced to shift the emphasis from the control module to the other modules. Lastly, the control module MCU interface malfunction before the Oral Exam #3, forcing the Empotra'o Solutions to redesign the MCU interface to another more secure and simplistic.

4. Market Overview <- *(Nelson)*

WindTel project is an automated data acquisition and wind tunnel operation system that provides aerodynamic and environmental measurements in experiments done on small scaled objects. The main goal of this project is to increase the usability of the wind tunnel by improving the accuracy of the data acquisition scheme. The users of the wind tunnel will take advantage of the automated system through an intuitive user interface supported by a mobile/desktop application.

<table>
<tr><th colspan="4">WindTel</th></tr>
<tr><td colspan="4">Potential Users:
● UPRM researchers: INCI and INME students and professors
● High school students and teachers (Science Fair projects)</td></tr>
<tr><td>Competitors</td><td>UPRM Wind Tunnel with WindTel</td><td>Juracán Energy Wind Tunnel</td><td>Aeronautical and Aerospace Institute of Puerto Rico</td></tr>
<tr><td>Cross-Sectional Area of Test Area (ft^2)</td><td>9</td><td>4</td><td>1</td></tr>
<tr><td>Maximum Wind Speed (mph)</td><td>50</td><td>~50</td><td>~89.5</td></tr>
<tr><td>Data Acquisition using Sensors and Actuators</td><td>Available</td><td>Not Available</td><td>Available</td></tr>
<tr><td>Blower Fan</td><td>Centrifugal type with 30 hp motor</td><td>Not Available</td><td>Computer controlled with a 2 hp DC Motor</td></tr>
</table>

Data Visualization Software	Available	Not Available	Available

Table 1: Market Analysis

The WindTel data acquisition system will provide the following competitive advantages to the users of the UPRM wind tunnel:

- System automation
- Measurements accuracy
- Data visualization
- Experiments results storage and portability
- Sustainable design

5. Results and Impact of the Project <- *(Misael)*

The WindTel system has led to a more usable and accurate wind-tunnel application. The integration of the wind-tunnel fan control with the supporting sensors into one compact system with a convenient user-interface facilitates the use of the wind-tunnel as a research tool, thereby bolstering the Department and University's prestige.

6. Budget Analysis <- *(Misael)*

Fig. 2 and Fig. 3 below present a graphical summary of the actual project costs in comparison to the expected project costs. These are categorized as Human Resources Costs, Hardware Costs, Software Costs, Facilities Costs, and Miscellaneous Costs. For all cost categories, with exception of software costs, actual costs are lower than the expected costs. The greatest source of this discrepancy stems from the facilities costs category. We had initially assumed database server maintenance costs of $1,500/month; however, these costs were not necessary as the Civil Engineering & Surveying Department of the UPRM has provided us with access to their servers, thereby eliminating the need for the Empotra'o Solutions team to set-up its own servers.

In the human resources costs, actual costs are lower than expected costs due to the two days in which the team was not able to work due to the power and water outages. In the hardware costs, actual costs were lower than expected costs due to the number of desktop computer requisitions being reduced from 4 to 2. For software costs, actual costs were exactly equal to the expected costs. This is because the software costs represent the software licence costs, all of which were bought in the initial stages of the project. Actual facilities costs were 50.51% lower than expected. As mentioned above, this is due to the database server maintenance costs no longer being necessary. In the miscellaneous costs category, $1,680 were used in the activation of risk contingency measures during the two separate occasions in which access to the Capstone Laboratory was restricted due to power and water outages in the UPRM. However, these costs were below the expected $3,000 in risk contingency measures, due to the Dynamic Measurements Module re-design phase being completed ahead of schedule. Moreover, because the team did not travel, the expected $850 travel costs were reduced to 0 actual costs. We present a detailed discussion of these expenditures in Appendix H.

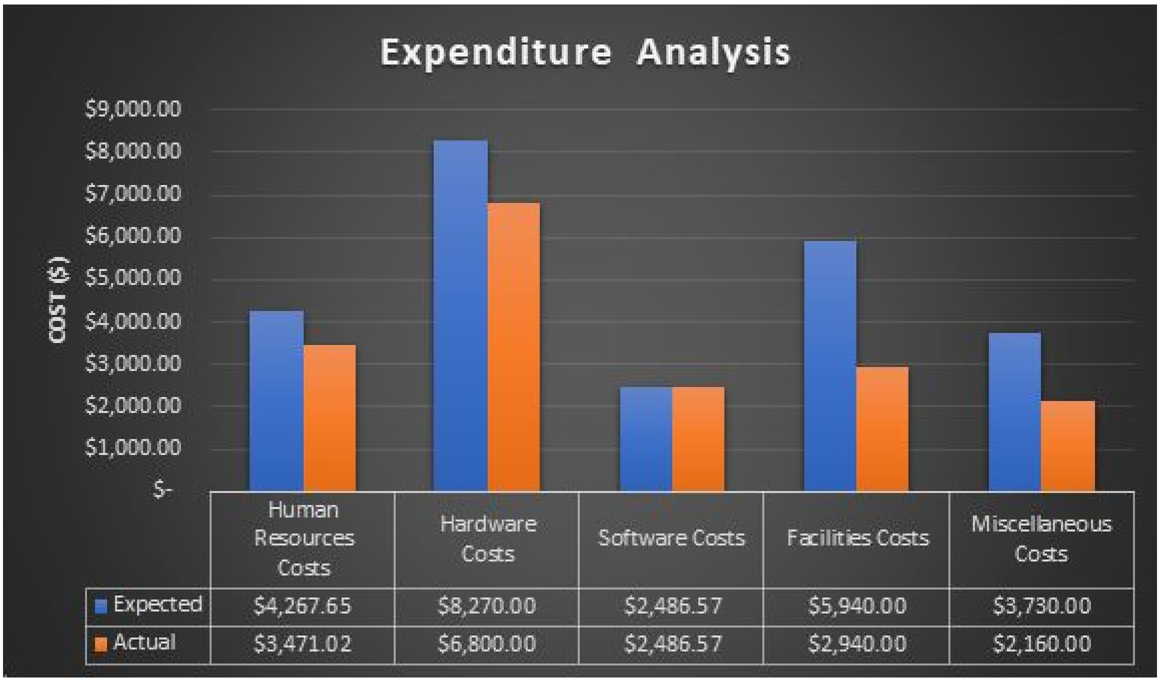


Fig. 2. Categorized summary of actual costs vs. expected costs

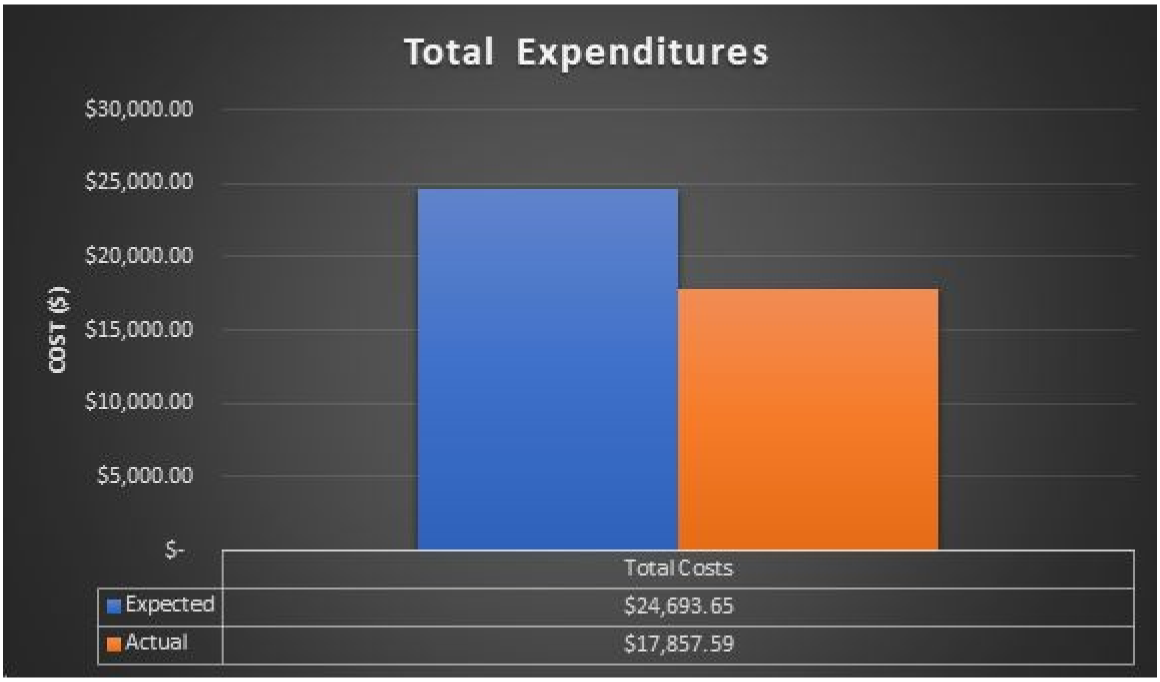


Fig. 3. Actual costs vs. expected costs

7. Conclusions and Future Work <- *(All)*

We have succeeded in completing the essential requirements proposed at the beginning of this semester (January 2019). Our hardware system is capable of permitting the user to vary the velocity of the wind motor, as well as perform experiments and acquire the aerodynamic and environmental measurements. The most recent results will be shown on the graphic Liquid Crystal Display (LCD) of the hardware system.

The WindTel system can also be operated remotely through a mobile/desktop application where results can viewed in a graphical manner and parameters can be set as desired. It is required that the administrator gives the correct permission the new users in order to use the WindTel system. The user may then view the results in a graphical manner, making it easier for the user to see the behaviour of the object being studied in the wind tunnel test section. The user may then store the results, if desired, in a remote database server that the application will be connected to.

For future work, we have that pressure module may be made to be expandible in order to increase the amount of pressure points that can be obtained, the balance module may be verified so that its damaged load cells can be replaced, and that the database server may be implemented in the local Civil Engineering Department, having the operating system of Windows Server 2016.

Bibliographic References <- *(All)*

[1] M. Jae, Signal Conditioning Circuit Design, 1st ed. East Lansing: Michigan State University, 2011, pp. 2,3.

1.

[2] Signal Conditioning & PC-Based Data Acquisition Handbook*, 3rd Ed., 2004, ed. John R. Gyorki, available from IOtech, Inc.*

[3] A. Llanos, J. Lebron, J. Luzon and J. Mendez, "Aerobal", *Aerobalmicro2.blogspot.com*, 2013. [Online]. Available: http://aerobalmicro2.blogspot.com. [Accessed: 10- Mar- 2019].

[4] GreatScott!, *Electronic Basics #33: Strain Gauge/Load Cell and how to use them to measure weight*. 2017.

[5] A. Llanos, J. Lebron, J. Luzon and J. Mendez, "allanos1/ICOM5217", *GitHub*, 2014. [Online]. Available: https://github.com/allanos1/ICOM5217/blob/master/Project/Phase%2005/Report/latex/sections/systemSpecifications.tex. [Accessed: 10- Mar- 2019].

[6] "Parallel Beam Load Cell Sensor 20kg/45lb Weight Sensor with Shielding Cable CE | eBay", *eBay*, 2019. [Online]. Available: https://www.ebay.com/itm/Parallel-Beam-Load-Cell-Sensor-20kg-45lb-Weight-Sensor-with-Shielding-Cable-CE-/181118193076. [Accessed: 10- Mar- 2019].

[7] *Fundamental Signal Conditioning*. 2019, pp. 1-19.

[8] "ADC Acquisition Time - Developer Help", *Microchipdeveloper.com*, 2019. [Online]. Available: http://microchipdeveloper.com/adc:adc-acquisition-time. [Accessed: 10- Mar- 2019].

[9] "Load cell electrical circuit. | Utilcell", *Utilcell.com*, 2019. [Online]. Available: https://www.utilcell.com/en/load-cell-electrical-circuit-/. [Accessed: 10- Mar- 2019].

[10] "Hardware Test Plan – for complex or mission-critical products - Viewpoint Systems", *Viewpoint Systems*, 2019. [Online]. Available: https://www.viewpointusa.com/TM/ar/hardware-test-plan-for-complex-or-mission-critical-products/. [Accessed: 10- Mar- 2019].

[11] "Product Testing Methods – for industrial hardware products - Viewpoint Systems", *Viewpoint Systems*, 2019. [Online]. Available: https://www.viewpointusa.com/TM/wp/product-testing-methods-industrial-hardware-products/. [Accessed: 10- Mar- 2019].

[12] "Hardware Product Testing Strategy – for complex or mission-critical parts & systems - Viewpoint Systems", *Viewpoint Systems*, 2019. [Online]. Available: https://www.viewpointusa.com/TM/wp/hardware-product-testing-strategy-complex-mission-critical-parts/. [Accessed: 10- Mar- 2019].

[13] "Hardware Verification, Testing and Maintenance —", *Aceproject.org*, 2019. [Online]. Available: http://aceproject.org/main/english/et/ete05a.htm. [Accessed: 10- Mar- 2019].

[14] "RS-232, RS-422, RS-485 Serial Communication General Concepts - National Instruments", *Ni.com*, 2019. [Online]. Available: http://www.ni.com/white-paper/11390/es/. [Accessed: 10- Mar- 2019].

[15] E. Williams, "What Could Go Wrong: SPI", *Hackaday*, 2019. [Online]. Available: https://hackaday.com/2016/07/01/what-could-go-wrong-spi/. [Accessed: 10- Mar- 2019].

[16] "3-wire SPI (Corelis chip)", 2019. [Online]. Available: https://m.blog.naver.com/PostView.nhn?blogId=wi3er&logNo=221008438362&proxyReferer=https%3A%2F%2Fwww.google.com%2F. [Accessed: 10- Mar- 2019].

[17] "Network Topologies: Advantages, Disadvantages and Diagrams of Each", *Comparitech*, 2019. [Online]. Available: https://www.comparitech.com/net-admin/network-topologies-advantages-disadvantages/. [Accessed: 10- Mar- 2019].

[18] "BoosterPack Checker - TI Cloud Tools", *Dev.ti.com*, 2019. [Online]. Available: https://dev.ti.com/bpchecker/#/. [Accessed: 10- Mar- 2019].

[19] "Types of Network Topology in Computer Networks | Studytonight", Studytonight.com, 2019. [Online]. Available: https://www.studytonight.com/computer-networks/network-topology-types. [Accessed: 10- Mar- 2019].

[20] "What is network topology? - Definition from WhatIs.com", *WhatIs.com*, 2019. [Online]. Available: https://whatis.techtarget.com/definition/network-topology. [Accessed: 10- Mar- 2019].

[21] "What is Twisted-pair Cable?", *Computerhope.com*, 2019. [Online]. Available: https://www.computerhope.com/jargon/t/twispair.htm. [Accessed: 10- Mar- 2019].

[22] J. Vasquez, "Taking the Leap Off Board: An Introduction to UART Over Long Wires", *Hackaday*, 2019. [Online]. Available: https://hackaday.com/2017/02/08/taking-the-leap-off-board-an-introduction-to-UART-over-long-wires/. [Accessed: 10- Mar- 2019].

[23] *P82B715*, 8th ed. 2009.

[24] "Benefits of Twisted Pair Cable Construction", *Sopto.com*, 2019. [Online]. Available: http://www.sopto.com/st/twinax-cable-knowledge/benefits-of-twisted-pair-cable-construction. [Accessed: 10- Mar- 2019].

[25] "Lesson 12: UART Basics – Simply Embedded", *Simplyembedded.org*, 2019. [Online]. Available: http://www.simplyembedded.org/tutorials/msp430-UART-basics/. [Accessed: 10- Mar- 2019].

[26] "UART Distance - Page 2", *Forum.arduino.cc*, 2019. [Online]. Available: https://forum.arduino.cc/index.php?topic=57604.15. [Accessed: 10- Mar- 2019].

[27] Software Testing Fundamentals. (2019). *Black Box Testing - Software Testing Fundamentals*. [online] Available at: http://softwaretestingfundamentals.com/black-box-testing/ [Accessed 10 Mar. 2019].

[28] The Internet Centre. (2019). *Ethernet Cable Color Coding Diagram - The Internet Centre*. [online] Available at: https://incentre.net/ethernet-cable-color-coding-diagram/ [Accessed 10 Mar. 2019].

[29] Forum.arduino.cc. (2019). *UART Distance - Page 2*. [online] Available at: https://forum.arduino.cc/index.php?topic=57604.15 [Accessed 10 Mar. 2019].

[30] Mysql.com. (2019). *MySQL :: How to Buy MySQL Products and Services*. [online] Available at: https://www.mysql.com/buy-mysql/ [Accessed 10 Mar. 2019].

[31] Postgresql.org. (2019). *PostgreSQL: The world's most advanced open source database*. [online] Available at: https://www.postgresql.org/ [Accessed 10 Mar. 2019].

[32] Flask.pocoo.org. (2019). *Welcome | Flask (A Python Microframework)*. [online] Available at: http://flask.pocoo.org/ [Accessed 10 Mar. 2019].

[33] Study.com. (2019). *What is an Object-Oriented Database? | Study.com*. [online] Available at: https://study.com/academy/lesson/what-is-an-object-oriented-database.html [Accessed 10 Mar. 2019].

[34] SearchDataManagement. (2019). *What is NoSQL (Not Only SQL database)? - Definition from WhatIs.com*. [online] Available at: https://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL [Accessed 10 Mar. 2019].

[35] SearchDataManagement. (2019). *What is RDBMS (relational database management system)? - Definition from WhatIs.com*. [online] Available at: https://searchdatamanagement.techtarget.com/definition/RDBMS-relational-database-management-system [Accessed 10 Mar. 2019].

[36] SearchMicroservices. (2019). *What is RESTful API? - Definition from WhatIs.com*. [online] Available at: https://searchmicroservices.techtarget.com/definition/RESTful-API [Accessed 10 Mar. 2019].

[37] Techopedia.com. (2019). *What is Twisted Pair Ethernet? - Definition from Techopedia*. [online] Available at: https://www.techopedia.com/definition/25741/twisted-pair-ethernet [Accessed 10 Mar. 2019].

[38] Software Testing Fundamentals. (2019). *White Box Testing - Software Testing Fundamentals*. [online] Available at: http://softwaretestingfundamentals.com/white-box-testing/ [Accessed 10 Mar. 2019].

[39] Wild, F. (2015, April 29). What Are Wind Tunnels? Retrieved from https://www.nasa.gov/audience/forstudents/k-4/stories/nasa-knows/what-are-wind-tunnels-k4.html

[40] A. S. for P. Affairs, "System Usability Scale (SUS)," *Usability.gov*, 06-Sep-2013. [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html. [Accessed: 12-May-2019].

[41] Chartjs, "Chart.js," *Chart.js · Chart.js documentation*. [Online]. Available: https://www.chartjs.org/docs/latest/. [Accessed: 12-May-2019].

[42] Ionicframework, "Ionic - Cross-Platform Mobile App Development," *Ionic*

Framework. [Online]. Available: https://ionicframework.com/. [Accessed: 12-May-2019].

[43] "Desarrollo de aplicaciones de Xamarin con Visual Studio | Visual Studio", *Visual Studio*, 2019. [Online]. Available: https://visualstudio.microsoft.com/es/xamarin/?rr=https%3A%2F%2Fwww.google.com%2F. [Accessed: 12- May- 2019].

[44] "Free JavaScript training, resources and examples for the community", *Javascript.com*, 2019. [Online]. Available: https://www.javascript.com/. [Accessed: 12- May- 2019].

[45] "Angular Blog", *Angular Blog*, 2019. [Online]. Available: https://blog.angular.io/. [Accessed: 12- May- 2019].

[46] "R: The R Project for Statistical Computing", *R-project.org*, 2019. [Online]. Available: https://www.r-project.org/. [Accessed: 12- May- 2019].

[47] SimpleLink™ MSP432P401R high-precision ADC LaunchPad™ Development Kit. [Online]. Available: http://www.ti.com/tool/MSP-EXP432P401R. [Accessed: 12-May-2019].

[48] "ESP32-DevKitC V4 Getting Started Guide¶," *ESP*. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/get-started/get-started-devkitc.html. [Accessed: 12-May-2019].

[49] "MATLAB - MathWorks", *Mathworks.com*, 2019. [Online]. Available: https://www.mathworks.com/products/matlab.html. [Accessed: 12- May- 2019].

[50] ARM® Cortex®-M4F Based MCU TM4C123G LaunchPad™ Evaluation Kit. [Online]. Available: http://www.ti.com/tool/EK-TM4C123GXL. [Accessed: 12- May- 2019].

[51] "ESR Value for MSP430F5418A - MSP low-power microcontroller forum - MSP low-power microcontrollers - TI E2E Community", *E2e.ti.com*, 2019. [Online]. Available: https://e2e.ti.com/support/microcontrollers/msp430/f/166/t/438980?ESR-Value-for-MSP430F5418A. [Accessed: 12- May- 2019].

[52] "The Web framework for perfectionists with deadlines | Django", *Djangoproject.com*, 2019. [Online]. Available: https://www.djangoproject.com/. [Accessed: 12- May- 2019].

[53] M. Jiménez, R. Palomera and I. Couvertier, *Introduction to Embedded Systems*. New York, NY: Springer, 2014.

[54] "What is the difference between Bluetooth and Wi-Fi?", *Techopedia.com*, 2019.

[Online]. Available:
https://www.techopedia.com/2/27881/networks/wireless/what-is-the-difference-between-bluetooth-and-wi-fi. [Accessed: 12- May- 2019

[55] MSP432P401R, MSP432P401MSimpleLinkTMMixed-Signal Microcontrollers. [Online]. Available: http://www.ti.com/lit/ds/symlink/msp432p401r.pdf. [Accessed: 12- May- 2019].

[56] INA333 Micro-Power (50μA), Zerø-Drift, Rail-to-Rail Out Instrumentation Amplifier. [Online]. Available: http://www.ti.com/lit/ds/symlink/ina333.pdf. [Accessed: 12- May- 2019].

8. Appendices

A. Glossary <- *(Nelson)*

1. **Accuracy:** Refers to the closeness of a measured value to its true value.
2. **Aerodynamic Characterization:** Refers to the process by which an object's structure and properties are probed when it moves through the air.
3. **Automate:** Operation with minimal human intervention.
4. **Black-Box Testing:** Also known as Behavioral Testing. A software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.
5. **Database:** A collection of information that is organized so that it can be easily accessed, managed and updated.
6. **Embedded System:** A device that contains tightly coupled hardware and software components to perform a single function, forms part of a larger system, is not intended to be independently programmable by the user and is expected to work with minimal or no human interaction.
7. **Force:** Any action that tends to maintain or alter the motion of a body or to distort it.
8. **Hardware:** Refers to the physical elements that make up a computer or electronic system and everything else involved that is physically tangible.
9. **Heuristic:** A problem solving method that uses shortcuts to produce good-enough solutions given a limited time frame or deadline.
10. **Manual Testing:** A type of Software Testing where Testers manually execute test cases without using any automation tools. Manual Testing is the most primitive of all testing types and helps find bugs in the software system.
11. **Modular:** Design approach that consists of organizing a complex system as a set of distinct components that can be developed independently and then plugged together.
12. **Pair Programming:** Is a cooperative way to program working side by side, which increases the quality of the code without delaying its deadline.
13. **Pressure:** The perpendicular force per unit area, or the stress at a point within a confined fluid.
14. **Prototype:** An original model, form or an instance that serves as a basis for other

processes.

15. **Quality:** A measure of excellence or a state of being free from defects, deficiencies and significant variations.
16. **Relative Humidity:** The ratio of the amount of water vapor present in the air to the greatest amount possible at the same temperature.
17. **Scrum:** Framework of a set of sequential events or meetings that are performed on a regular basis.
18. **Signal Conditioning Circuit:** Circuit that manipulates an analog signal in such a way that it is optimized for further processing.
19. **Software:** Set of instructions or programs instructing a computer to do specific tasks.
20. **System Usability Scale (SUS):** A reliable tool for measuring usability quantitatively. It consists of a 10-item questionnaire with five response options for respondents; from Strongly agree to Strongly disagree.[40]
21. **Temperature:** A measure of how hot or cold something is.
22. **Troubleshooting:** The process that determines the cause of the problem in the electronic circuit by examining the affected area of it, and then by taking appropriate action.
23. **Unit Testing:** A testing technique in which individual modules are tested to determine if there are any issues by the developer himself.
24. **Usability:** Refers to the ease of access and/or use of a product or website.
25. **User Interface:** Visual part of a computer application or operating system through which a user interacts with a computer or software of a system.
26. **V Model:** A method that is followed in a sequential order where each phase is carefully tested and validated.
27. **White-Box Testing:** A software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs.
28. **Wi-Fi:** A wireless networking technology that allows computers and other devices to communicate over a wireless signal.
29. **Wind Speed:** The rate of the movement of wind in distance per unit of time.
30. **Wind Tunnel:** Device for producing a controlled stream of air in order to study the effects of movement through air or resistance to moving air on models of aircraft and other machines and objects.

B. User Requirements

B. User Requirements

The following list of requirements has been discussed with the Empotra'o Solutions team and the client Dr. Raúl Zapata López. The completion of this list signifies the project success and closure.

Luis O. Vega Maisonet
WindTel Project Manager
Luis O. Vega Maisonet

Raúl E. Zapata López
Client
Dr. Raúl Zapata López

1. Functional
   a. Essential
      - The embedded system will consist of a master module exchanging data with three slave modules.
      - The slave modules will acquire data from environmental and aerodynamic sensors.
      - The embedded system shall contain a user interface for manual operation of the wind tunnel.
      - The embedded system user interface shall be composed of a graphic Liquid Crystal Display (LCD) and pushbuttons.
      - The embedded system shall be able to acquire the following aerodynamic and environmental measurements from the wind tunnel test section:
        - object weight forces
        - the distribution of pressure upon the object
        - the temperature of the test section
        - the humidity relative to the amount of water vapor present in the air
        - the wind speed
      - The embedded system shall be able to acquire aerodynamic environmental data from an object with less than 9 inches of length, less than 3 inches of width and less than 3 inches of height.
      - The embedded system shall be able to acquire the weight exerted forces from an object which is placed inside the wind tunnel test section in five directions: up, drag front, drag back, side left and side right.
      - The embedded system will be capable of acquiring a maximum object weight of 20 kg (~45 lb).
      - Analog to Digital Converters (ADCs) of the embedded system will convert electrical pulses obtained from force measurements to digital data with a resolution of at least 12 bits.
      - The embedded system shall acquire the distribution of pressure upon the test object from 16 barometric pressure sensors.
      - The embedded system shall present the aerodynamic and environmental data measurements in the embedded system user interface.
      - The system shall present the aerodynamic and environmental measurements in the desktop application or in the embedded system user interfaces using the International System of Units or the English System.

- The embedded system shall be capable of managing the wind tunnel blower fan through a closed-loop control system submodule.
- The embedded system shall be able to turn on or off the wind tunnel blower fan.
- The control scheme of the system will provide increments or decrements of the wind tunnel motor of approximately 2mph to 3mph.
- The embedded system will acquire a maximum wind speed of approximately 50 mph.
- The user will be able to monitor the speed of the wind inside the wind tunnel in the embedded system or desktop application user interfaces.
- The experiment duration is established by the user in the embedded system or the desktop application user interfaces.
- The experiment time will range from 5 seconds to 5 minutes.
- The experiment will start after the user finishes the specification of the aerodynamic and environmental measurements and the experiment time duration.
- The experiment is limited to acquiring a maximum of 5 weight forces measurements, 16 barometric pressure measurements, 1 relative humidity measurement, 1 temperature measurement and 1 wind speed measurement per a minimum instance of time of 5 seconds.
- The experiment process can be stopped if the user desires by selecting the stop option in the embedded system or desktop application user interface.
- If the experiment process is stopped by external sources, it will result in the deletion of the data at both ends of the transaction.
- The embedded system shall be able to transfer the acquired aerodynamic and environmental data measurements to a desktop application through Wi-Fi communication.
- The desktop application shall be able to acquired aerodynamic and environmental data measurements from the embedded system through Wi-Fi communication.
- The desktop application must have a login screen for users to sign in or sign up.
- The user shall be able to retrieve the aerodynamic and environmental data from a database server only through the desktop application.
- The user shall be able to delete the aerodynamic and environmental data from a database server only through the desktop application.
- The user shall be able to store the aerodynamic and environmental data to a database server only through the desktop application.
- The user shall be able to design experiments using the embedded system or the desktop application user interfaces where the user will specify the following:
    - the length of the rod where the object will be placed in the wind tunnel test section
    - the sensors to be used
    - the aerodynamic and environmental data measurements
    - an experiment time duration
    - the number of data measurements to be taken in the experiment time duration

- The user shall be able to perform statistical operations on the acquired data measurements in the desktop application such as:
    - mean
    - median
    - standard deviation
    - variance
    - probability distribution
    - max value
    - min value
    - skewness
    - kurtosis
    - order of magnitude
    - confidence intervals
    - quartiles
    - outliers
- The desktop application will be able to create line, bar and dot plots between any two aerodynamic or environmental measurements the user desires.

b. Desirable
- The embedded system will contain a sealed enclosure for the barometric pressure sensors.
- The user will be able to calculate pressure related metrics in the desktop applications such as:
    - pressure coefficient
    - drag force
    - lift force
- The embedded system's module containing the barometric pressure sensors shall have expandability properties of adding more pressure modules to increase the number of pressure sensors that can be connected.
- The desktop application shall be migratable to an Android Tablet which will operate the embedded system inside the wind tunnel laboratory.

2. Non-Functional

a. Usability
- The WindTel system must satisfy Jakob Nielsen's usability heuristics.
- The WindTel system must obtain a score of at least 70 in the System Usability Scale (SUS).
- A user manual detailing the proper usage of the WindTel system must be delivered to the client.
- A wind tunnel administrator manual must be delivered to the client containing all the information in the user manual plus an electronic balance calibration procedure and a module repair procedure.

b. Reliability
- The WindTel system must be able to detect faults during communication between hardware modules and display them on the embedded system user interface.

c. Performance
- The embedded system will acquire aerodynamic and environmental measurements within an accuracy of 10% of the baseline values

established with manual instruments.

d. Communication Interfaces
    - Serial Peripheral Interface (SPI) communication will be used for the pressure module circuitry.

C. System Specifications <- *(Luis)*

MSP432P401R	
Characteristic	**Value**
Operating Voltage	3.3V
Operating Current	3.9mA
Power Consumption	12.87mW
VOH	2.7V
VOL	0.6V
VIH	2.25V
VIL	0.55V
Maximum Current Per Pin	±2mA
Maximum Current Per Port	±20 mA

Table 2: Master Module MSP432P401R Specifications

Kentec QVGA Display Booster Pack	
Characteristic	**Value**
Operating Voltage	3.3V
Operating Current	300µA
Power Consumption	0.990mW
Pin Minimum Logic Input Voltage	1.4V
Input Current	Input Current < 10mA
Luminance	270 cd/m2

Table 3: Control Panel Module Kentec QVGA Display Specifications

Variable Speed Driver (Schneider Electric Altivar 212)	
Characteristic	**Value**
Operating Voltage	10V

Operating Current	10mA
Power Consumption	100mW
F (Multifunctional programming input)	24VDC with Input Current < 5mA
PP (Voltage Supply for reference potentiometer)	10VDC with Input Current < 10mA
P24 (Power Supply Output)	24VDC with 50mA

Table 4: Closed Loop Control System Variable Speed Driver Specifications

Voltage Regulator (L7805CV)	
Characteristic	**Value**
Operating Voltage	10V
Operating Current	500mA
Power Consumption	5000mW
Output Voltage	5V
Input Current	Input Current < 6mA

Table 5: Closed Loop Control System Voltage Regulator Specifications

Digital Potentiometer (DS1809-010)	
Characteristic	**Value**
Input Voltage	5V
Input Current	0.5mA
Power Consumption	2.5mW
Logic Input High	2.0 V
Logic Input Low	0.8V

Table 6: Closed Loop Control System Digital Potentiometer Specifications

Relay Module (SRD-05VDC-SL-C)

Characteristic	**Value**
Operating Voltage	5V
Operating Current	85mA
Power Consumption	425mW
IN1 (Active Low Input Voltage)	Input Voltage < 2.0V
IN2 (Active Low Input Voltage)	Input Voltage < 2.0V

Table 7: Closed Loop Control System Relay Module Specifications

Wi-Fi Submodule (ESP32-DevkitC V4)	
Characteristic	**Value**
Operating Voltage	3.3V
Operating Current	50nA
Power Consumption	0.165µW
VIH	2.48V
VIL	0.825V
VOH	2.64V
VOL	0.33V
Input Low Current	Input Low Current < 50 nA
Input High Current	Input High Current< 50 nA
Output Low Current	Output Low Current < 28 mA
Output High Current	Output Low Current < 20 mA

Table 8: Wi-Fi submodule Specifications

Instrumentation Amplifiers (INA333)	
Characteristic	**Value**
Operating Voltage	5V
Operating Current	50µA

Power Consumption	250µW

Table 9. INA333 Amplifiers Power Specifications

High Performance Operational Amplifiers (OPA363)	
Characteristic	**Value**
Operating Voltage	5V
Operating Current	1.1mA
Power Consumption	5.5mW
Output Voltage	$0 \leq Output\ Voltage \leq 3.3V$

Table 10. High Performance Operational Amplifiers Specifications

High Speed Operational Amplifiers (OPA743)	
Characteristic	**Value**
Operating Voltage	5V
Operating Current	1.1mA
Power Consumption	5.5mW
Common Mode Rejection Ratio (CMRR)	84dB

Table 11. High Speed Operational Amplifiers Specifications

Humidity and Temperature Sensor (DHT22)	
Characteristic	**Value**
Operating Voltage	5V
Operating Current	2.1mA
Power Consumption	10.5mW
Sensing Period	2 seconds
Operating Range	0% - 100%
Resolution	0.1%

Table 12: Dynamic Measurement Module Humidity and Temperature Sensor Specifications

Wind Speed Sensor (Young 05103 Wind Monitor)	
Characteristic	**Value**
Operating Voltage	3.3V
Operating Current	<2.0mA
Power Consumption	<6.6mW
Sensing Range	0 mph to 224 mph
Accuracy	0.6 mph
Output Frequency	3 cycles per revolution (0.098m/s per Hz)
Output Voltage	3.3V

Table 13: Dynamic Measurement Module Wind Speed Sensor Specifications

Amazon Web Services (AWS) Database Server	
Characteristic	**Value**
Storage	20 GB
Operating System	Ubuntu Linux
IP	http://54.159.138.94:80

Table 14: Database Server Specifications

Android Tablet	
Characteristic	**Value**
Hard Drive Storage	16 GB
RAM	1 GB SDRAM
Operating System	Android 8.1 Oreo
Item weight	1.8 pounds

Table 15: Android Tablet Specifications

D. Analysis of Alternatives <- *(All)*

The design choices for the project are divided into two broad categories: software choices and hardware choices. The former is comprised of the choices for the front-end, REST API, database management, and statistics and data visualization features for the WindTel application. The latter is comprised of the choices for the microcontroller development platform, and the interfaces for inter-modular as well as external communication. Tables 16 and 17 below present a summary of the analysis of design alternatives for all software and hardware design areas.

Design Area	**Chosen Technology**	**Alternatives**	**Justification**
Front-end	Ionic	Xamarin, React	Cross-platform
Statistics and Data Visualization	Chart.js	R, MATLAB, Python	Client-side processing
REST API	Flask	Django	Fast performance; employee experience
Database Management	PostgreSQL	MySQL	Relational database; migratable; employee experience

Table 16: Summary of software alternatives analysis

Design Area	**Chosen Technology**	**Alternatives**	**Justification**
Inter-module Communication Configuration	Star Network Topology	Bus Network Topology	Easy to troubleshoot; reduced information traffic
Inter-module Communication Interface	UART	SPI, UART	Easy to implement; less susceptible to noise; transmission range
External Communication Interface	Wi-Fi	Bluetooth	Faster transfer rate; transmission range
MCU Development	TI MSP432	TI TM4C123G, TI	Large number of pins,

Platform		MSP430F5418A	high resolution ADC, low power device
Balance Module ADC Reference Voltage	3.3V	5V	Power consumption; simpler and better ADC resolution
BMP280 Inter-sensor Communication	Star Network Topology	Ring Network Topology	Unable to address more than 2 different sensors with UART

Table 17: Summary of hardware alternatives analysis

Software Design Alternatives Analysis

A. Front-end <- *(Misael)*

Chosen technology: Ionic and Electron

Ionic is an open-source framework for hybrid mobile application development with Angular integration [42]. This technology was chosen for its 'Write once, deploy everywhere' approach to cross-platform development, which will facilitate the deployment to desktop [41]. Electron is a framework that allows for the development of desktop GUI applications using web technologies with access to native APIs. This framework, in conjunction with the Ionic framework, will facilitate the deployment of the WindTel application to a desktop platform. Due to its cross-compatibility, AngularJS integration, ease-of-use, and extensive documentation [42].

Alternative: Xamarin

Xamarin is another framework for cross-platform mobile development. However, Xamarin is not an open-source alternative, and its language of use is C#, with which none of the team members have prior experience [43].

Alternative: React

Javascript library for building user interfaces. Ease of use and documentation make this an attractive alternative. However, this library is solely for building user interfaces, and has no additional tools for application development [44].

B. Statistics and Data Visualization <- *(Misael)*

Chosen technology: Angular Chart

Angular Chart [45].

Alternative: MATLAB

An industry standard for statistical analysis, data visualization, and general-purpose scientific computing. However, MATLAB is proprietary software, as well as expensive. Moreover, the learning curve is greater compared to other alternatives[49].

Alternative: R

A programming language widely-used by the scientific community for statistical analysis and data visualization. The main disadvantage is the higher learning curve [46].

C. Database Management <- *(Luis)*

Chosen technology: PostgreSQL

A comparison between the RDBMS, the Object Oriented Database (OODB) and the No SQL database models have been performed [33,34]. RDBMS follow the Entity Relational model and is related to the Create, the Read, the Update and the Delete (CRUD) operations [35]. The OODB treats the entities has objects in a object oriented programming language and insert the concept of classes. No SQL manage different kind of data models such document and keys [34]. The RDBMS is the database model selected for this database server because of it follows a Entity Relational model which is the best suited model to store the wind tunnel experiment data samples [35]. PostgreSQL is a open source database management system with a significant amount of documentation and resources from its official site and its community [31].

Alternative: MySQL

MySQL is similar to PostgreSQL in syntax, in having a significant amount of documentation and resources but it is not a open source database management system. A license from Oracle Corporation must be bought to use MySQL[30].

D. REST API <- *(Luis)*

Chosen technology: Flask

Flask is a web microframework for Python and a technology to develop REST API in conjunction with a library named Psycopg2 [32]. It has a extensive documentation and its prefer because of the Empotra'o Solutions experience with such technology [32].

Alternative: DJango

DJango is web framework similar to Flask. It provides a vast documentation and also works with the Python programming language. The main disadvantage is the higher learning curve [52].

Hardware Design Alternatives Analysis

A. Inter-module Communication Configuration <- *(Luis)*

Chosen technology: Star Network Topology

In the star network topology every node has a dedicated connection to the MCU. Past components from the Embedded Design course project can be reused. It's more flexible to troubleshoot [53].

Alternative: Bus Network Topology

In the bus network topology the transmission of data is in one direction. Every device is connected into a single cable. The performance of the network decreases if the bus traffic increase [53].

B. Inter-module Communication Interface <-*(Luis)*

Chosen technology: UART

The UART is serial communication protocol supports board-level interconnections of IC modules and peripherals. The protocol uses two lines to establish a full duplex [53].

Alternative: SPI

The Serial Peripheral Interface (SPI) is a synchronous bus standard with full duplex capability. Establish a mechanism to relay information between a master device and one or more slave devices [53]. This alternative is a optimum selection for this project but UART buffer components have already been bought in the first WindTel implementation during the Embedded System Design course. This alternative would result expensive to the Empotra'o Solutions team.

C. External Communication Interface <-(Misael)

Chosen technology: Wi-Fi

Wi-Fi is a wireless communication technology. Wi-Fi was chosen for its high speed, range, and security, all of which are essential in the WindTel project [54].

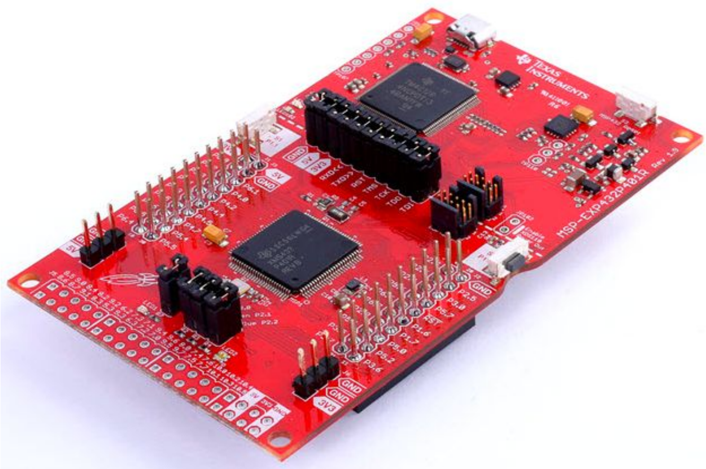
Alternative: Bluetooth

Bluetooth is also a wireless communication technology. However, Bluetooth's low speed, low range, and low security force us to discard it as an alternative [54].

D. MCU Development Platform <-(Nelson)

Chosen Technology: TI MSP432P401R

The MSP432P401R has a 48MHz ARM 32-bit Cortex-M4F Central Processing Unit (CPU) with 256KB Flash and 64KB RAM memory (Fig. 4). It has a high-precision 16-bit 24 channels ADC which enables high resolution of analog data taken from the outside world. In addition, it has an ultra-low-power mode that can be enabled using software. Another feature is that it includes four programmable serial communication modules which can be used to transfer data using UART, SPI and/or UART protocols.. Because of its high-resolution ADC, this MCU was chosen to reduce the error of the telemetry in the wind tunnel. The MCU is also compatible with Bluetooth modules and Wi-fi wireless connectivity, which will facilitate interfacing with the WLAN peripheral. It can be programmed using Integrated Development Environments (IDEs) such as IAR Embedded Workbench and Code Composer Studio. Finally, a Launchpad kit is available in Texas Instruments, Inc. website for $12.99 [47].



Courtesy of Texas Instruments, Inc.

Fig. 4. TI MSP432P401R Launchpad

Alternative: TI TM4C123G

The TM4C123G, member of the Tiva C Series family of MCUs, has an 80MHz 32-bit ARM Cortex-M4-based microcontrollers CPU with 256KB Flash, 32KB SRAM and a 2KB EEPROM (Fig. 5). It has a 12-bit 12-channel ADC with single-ended and differential-input configurations and an on-chip internal temperature sensor. In addition, it has 8 UART, 6 UART and 4 SPI ports which can be used for serial communication with the other external devices. This MCU was also considered because of the resolution of the ADC, the memory size and the number of serial communication modules which makes it suitable for the application. It can be programmed with IDEs such as IAR Embedded Workbench and Code Composer Studio. As the MSP432P401R, the TM4C123G comes in a Launchpad kit for $12.99 [50].

Courtesy of Texas Instruments, Inc.

Fig. 5. Tiva C Series Launchpad

Alternative: TI MSP430F5418A

The MSP430F5418A is a 16-bit ultra-low power microcontroller with a 25MHz Reduced Instruction Set Computing (RISC) architecture and a memory of 128KB and a 16KB RAM (Fig. 6). The MCU has a 12-bit 16-channel ADC and four universal serial communication interfaces which can be programmed to use has UART, SPI and UART protocol ports. Although the memory size is limited compared to the other alternatives, the resolution of the ADC and the number of communication ports makes this MCU a good option. The main disadvantage is that it does not come in a development kit or in a Launchpad kit which makes its usage more difficult. The cost of the MCU is $4.63 [51].

Courtesy of Texas Instruments, Inc.

Fig. 6. The TI MSP430F5418A MCU

E. Balance Module ADC Reference Voltage <- *(Nelson)*

Chosen technology: 3.3V

The signal conditioning circuit is simpler, economic and provides a higher resolution for the ADC.

Alternative: 5V

Although the signal conditioning circuit is not affected by the loading effect produced by the input resistance of the ADC, the circuit is more complex, expensive and the output voltage headroom is constrained by the 5V power supply.

F. BMP280 Inter-sensor Communication <- *(Kahlil)*

Chosen technology: Star Network Topology

In the star network topology, every node has a dedicated connection to the MCU. Past components from the Embedded Design course project can be reused. It's more flexible to troubleshoot [X].

Alternative: Ring Network Topology

In the ring network topology the MCU has connection to the first and last device, while each device is connected to each other having the transmission of information in a unidirectional manner. Past components from the Embedded Design course project can be reused.

E. System Architecture and Interfaces <- *(All)*

WindTel System Block Diagram <- *(Luis)*

The WindTel embedded system consists of four microcontroller-based hardware modules partitioned according to their functionality. These modules are categorized as: Master Module, Pressure Module, Balance Module and Dynamic Measurements Module. Each of the architectures of these modules are explained in more detail in the following sections. A Universal Serial Bus (USB) charger connected to a power outlet will serve has the system power supply. Fig. 7 and Fig. 8 illustrate how each of these hardware modules interface with each other.

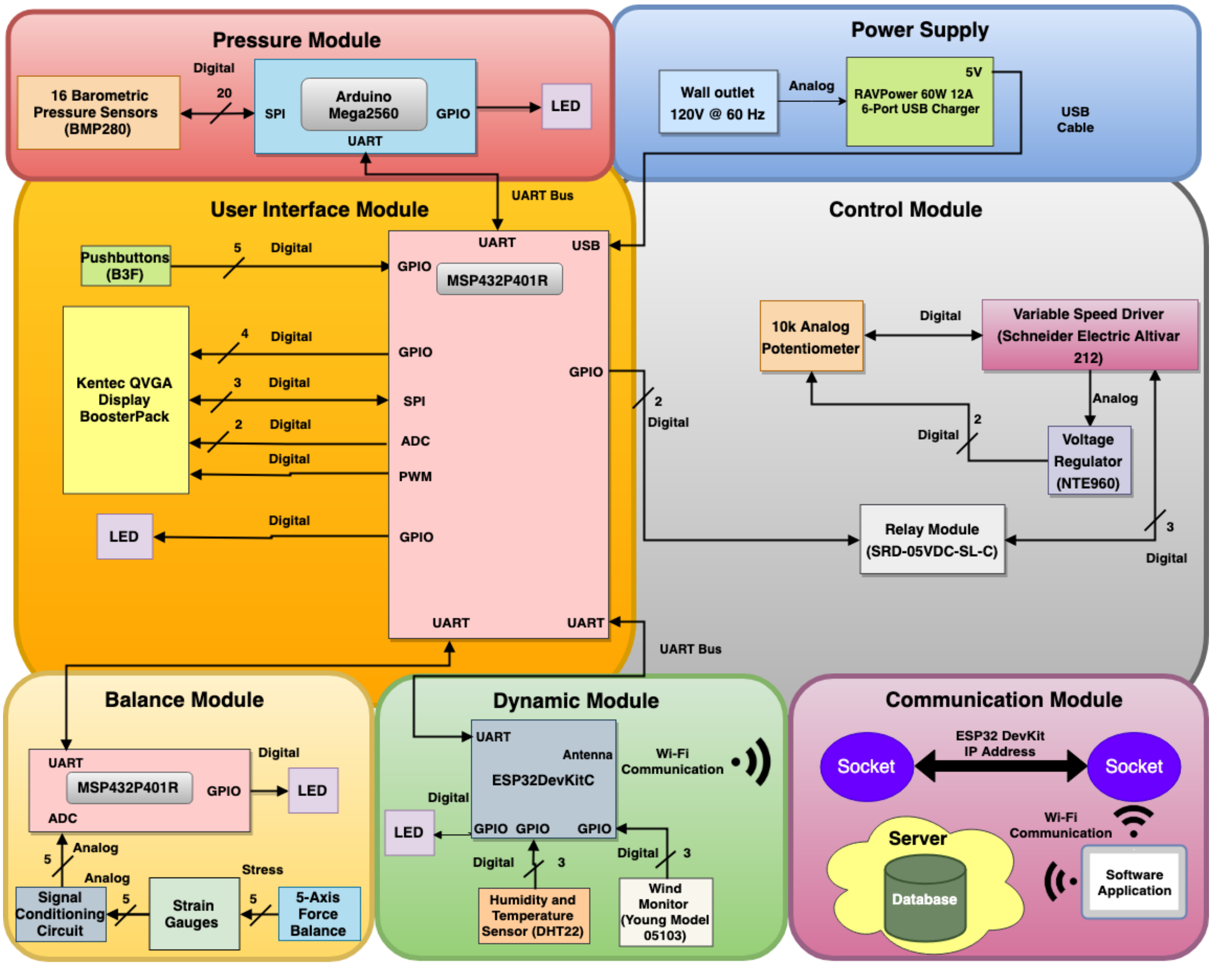


Fig. 7. WindTel Hardware Block Diagram

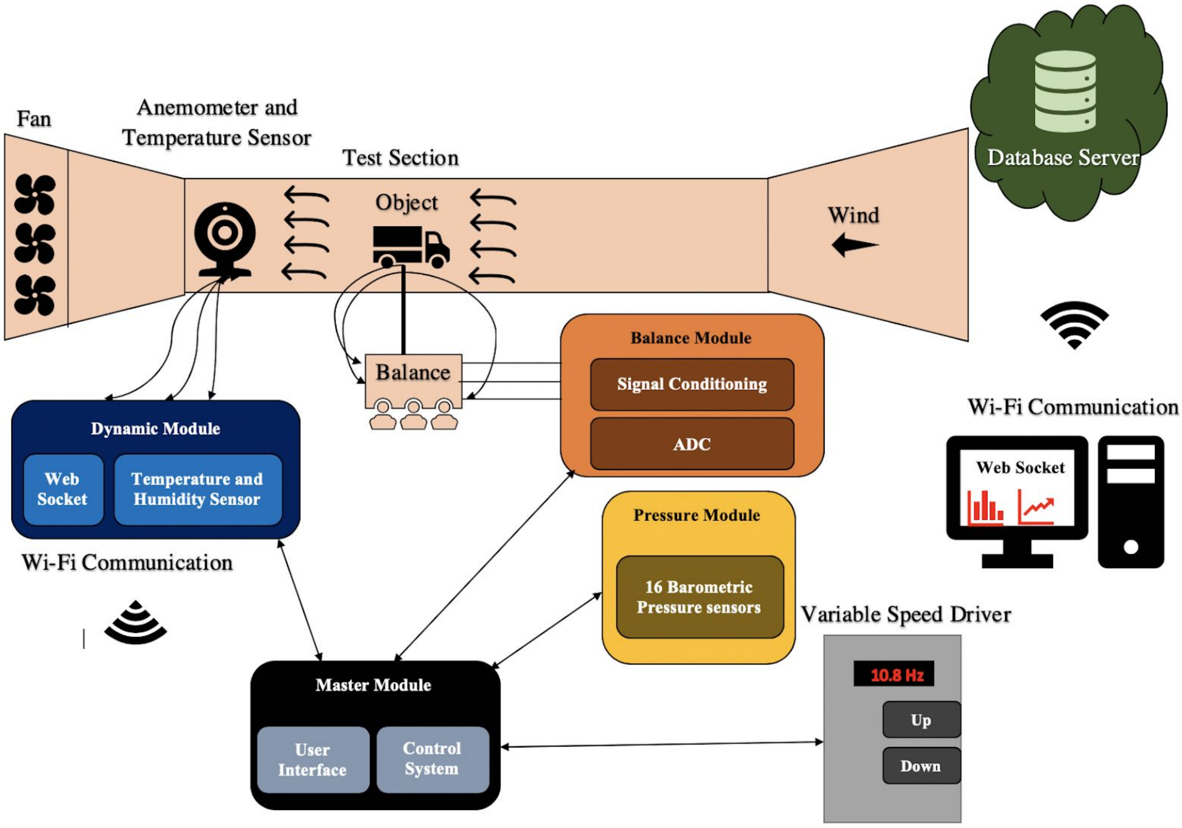


Fig. 8. WindTel System Conception Diagram

1. Hardware Modules <- *(All)*

    a. Master Module <- *(Luis)*

    The Master Module has the purpose of serving as an intermediary module, where the Balance, Pressure and Dynamic Measurements modules communicate with the Master Module via their own respective UART bus. The data acquisition process will begin when the Master Module sends a request to a respective slave modules, and then the slave module sends an acknowledgement to the Master Module for the transaction to begin. Fig. 9 illustrate a block diagram of the Master module interface with its slave modules.

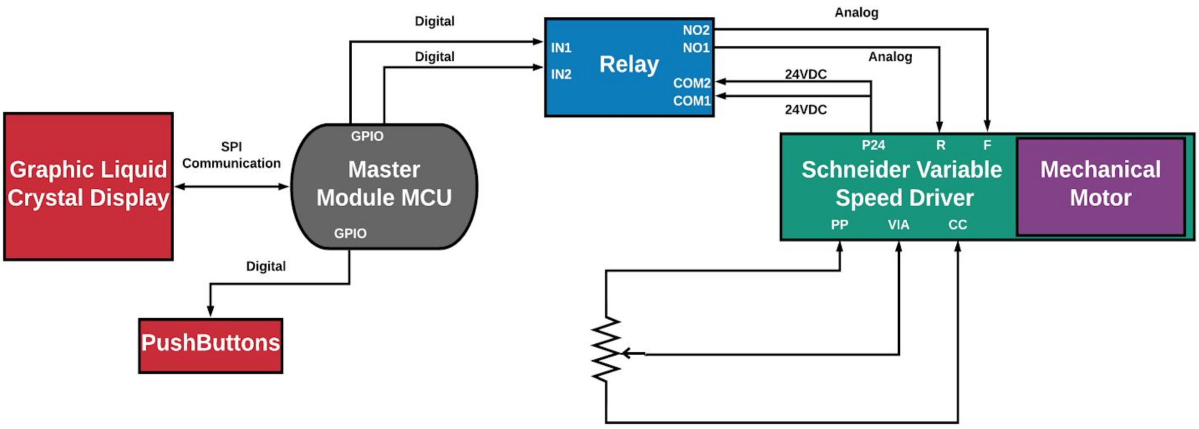


Fig. 9. Master module and slave modules interface

b. Pressure Module <- *(Kahlil)*

The Pressure Module will collect the distribution of pressure measurements upon the object to be studied.. Each of the sensors will be in a 4x4 capsule matrix, resulting in resolution of 16 pressure points. Fig. 10 shows the hardware architecture of the pressure module.

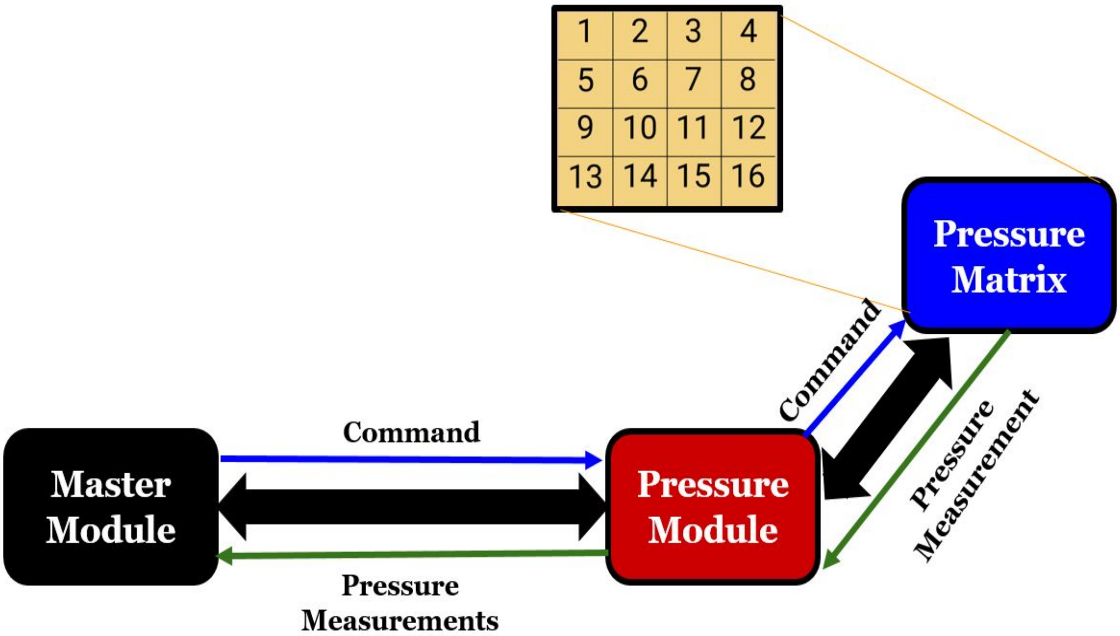


Fig. 10. Pressure Module hardware architecture

c. Balance Module <- *(Nelson)*

Fig. 11 shows the proposed hardware architecture for the WindTel Balance Module. The balance of the wind tunnel was implemented by computer engineering students during their Capstone course in 2013. They attached five strain gauge-based load cells to measure the drag and lift forces that the wind exerts to the object under study inside the wind tunnel [3]. The most important part of this module is the signal conditioning circuit which amplifies the analog signals from the load cells and filters the undesired noise at high frequencies. The conditioned analog signal is digitized using an Analog to Digital Converter (ADC) to permit the MCU of the module to process the data.

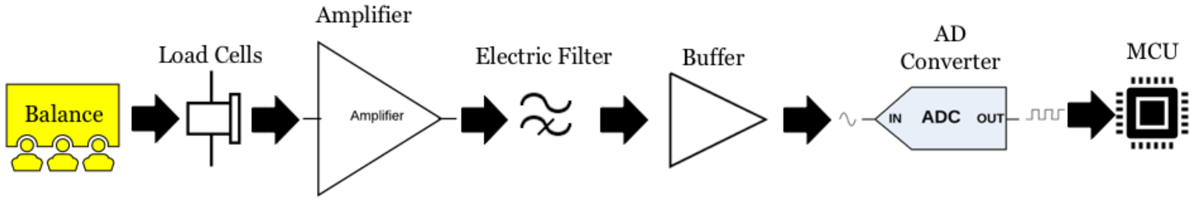


Fig. 11. Balance Module Hardware Architecture

d. Dynamic Measurements Module <- *(Misael)*

The Dynamic Measurements Module is composed of an anemometer which functions as a wind-speed sensor, and a temperature and humidity sensor. These sensors provide the environmental conditions inside the wind tunnel. A block diagram of this module is provided in Fig. 12 below.

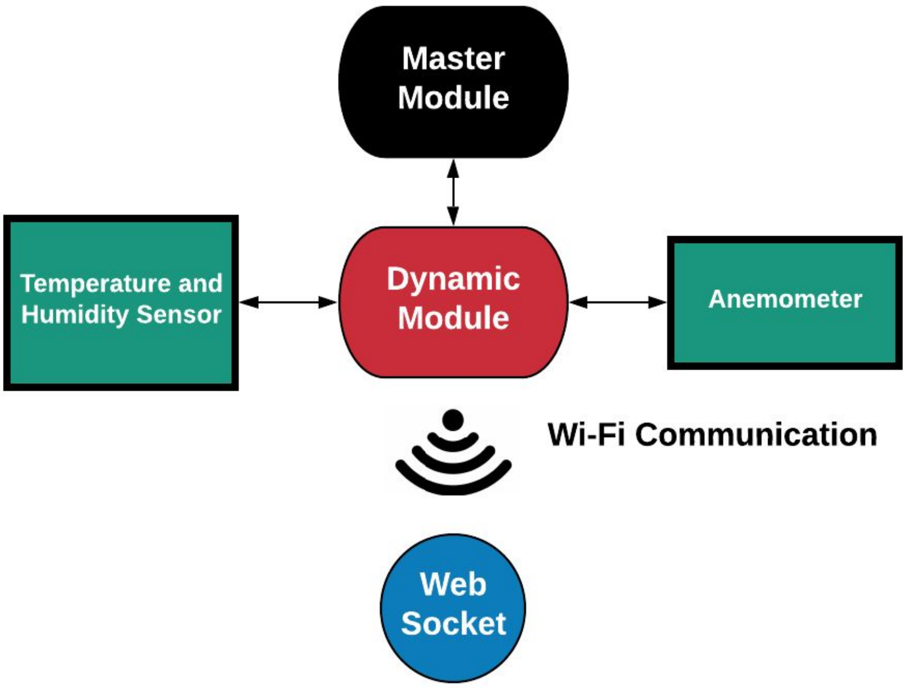


Fig. 12. Dynamic Measurements Module block diagram

2. Software Architecture <- *(Misael)*

The software architecture is divided into three layers: User Interface, REST API, and Database. The User Interface layer is what the user sees when he interacts with the WindTel application. It is through this layer that he may log in or register, design experiments and experiment templates, perform experiments, and carry out statistics and data visualizations upon the acquired results.

The storage of experiment results in the Database layer is mediated by a REST API. This REST API layer performs CRUD operations that allow for the storage and subsequent retrieval or modification of experimental results. Fig. 13 below presents a layered illustration of this architecture.

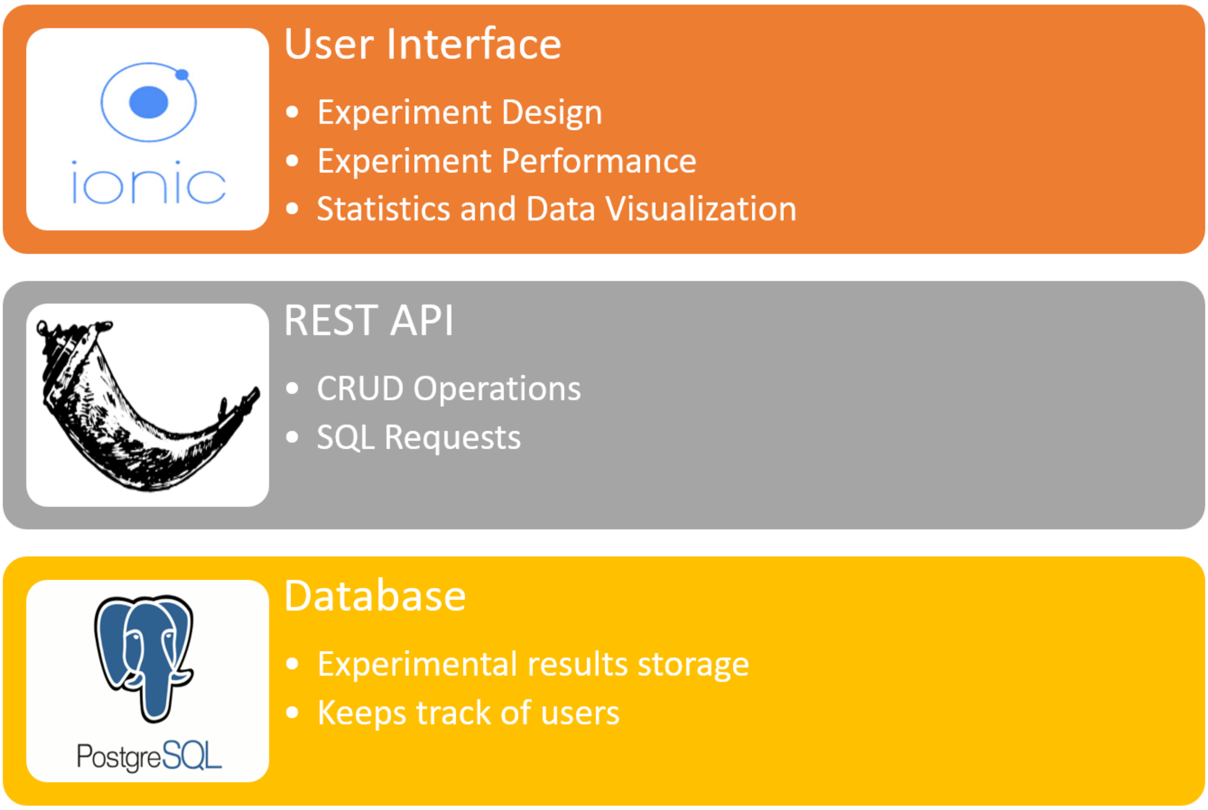


Fig. 13. Software Architecture

F. Design Documentation <- *(All)*

1. Hardware Documentation<- *(All)*

- Master Module <- *(Luis)*

  The Master module is divided into 2 submodules: a control system and a user interface.

  - Control System

    The control system hardware has already been implemented as part of the Embedded Systems Design course. Unfortunately it did not satisfy the wind tunnel functionality. This sub-module was redesign to use a 10k ohm analog potentiometer instead of a 10k ohm digital potentiometer that was proposed. This redesign was perform due to the digital potentiometer consuming more power than it can handle. A dual relay module is used to turn the wind tunnel operation on or off. The control system is interfaced via two General Purpose Input Output (GPIO) pins of the Master module microcontroller (MCU). The MCU sends two active low digital signal to the relay module IN1 and IN2 pins. The IN1 pin serve as a stop signal and IN2 serve has start signal to the variable speed driver the wind tunnel. The two channel relay selection will indicate the wind motor direction.

- User Interface

The user interface hardware as already being implemented as part of the Embedded System Design course. This submodule is compose of five hardware debouncing circuits and a graphic Liquid Crystal Display (LCD) BoosterPack named Kentec QVGA Display BoosterPack. Each of these circuit takes approximately 43.7ms to charge and approximately 39ms to discharge its respective capacitor. The time charge and discharge time values were chosen using has a guide the Embedded System Design Laboratory Manual. The LCD BoosterPack is a much more simpler approach to implement due to the quick interface integration with the MCU. A TI hardware compatibility verification tool was used to determine the LCD and MSP432P401R compatibility. The LCD BoosterPack communicates directly with the MCU through SPI communication. TI provide an extensive library named DriverLib for BoosterPack source code development. Fig. 14 display the BoosterPack/MCU compatibility checker tool.

Fig. 14. Front Side of the User Interface Including the Analog Potentiometer in the Upper right corner of the enclosure

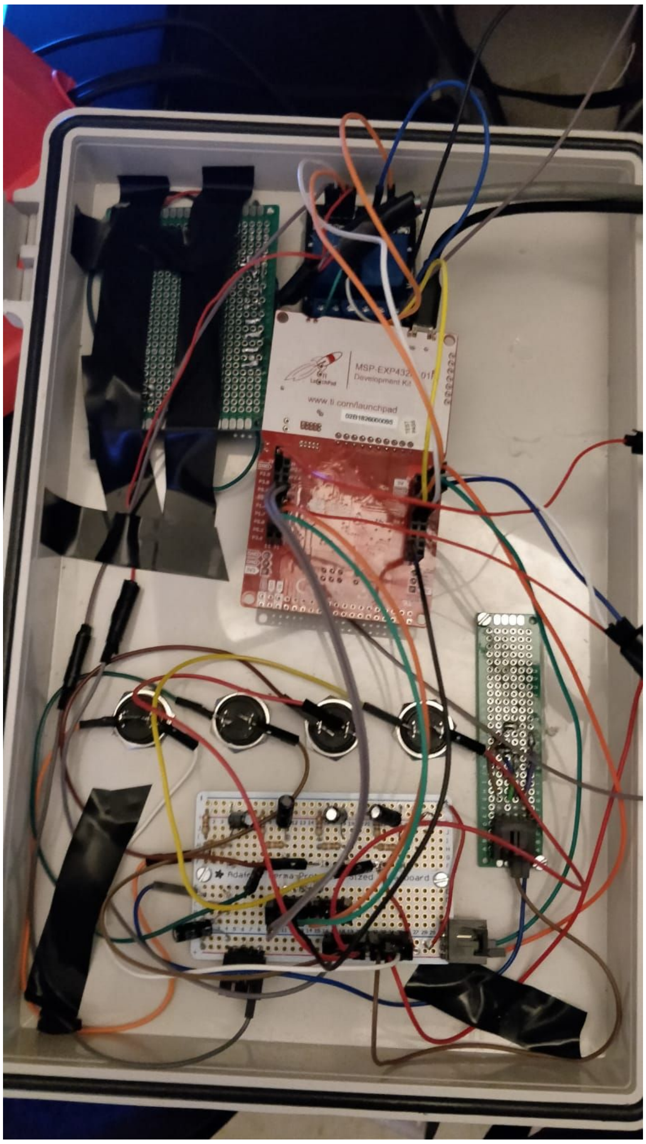


Fig. 15. Back Side of the User Interface

MSP432P401R	
Characteristic	**Value**
Operating Voltage	3.3V
Operating Current	3.9mA
Power Consumption	12.87mW

Table 18: Master Module MSP432P401R Power Consumption

Kentec QVGA Display Booster Pack	
Characteristic	**Value**
Operating Voltage	3.3V
Operating Current	300µA
Power Consumption	0.990mW

Table 19: Control Panel Module Kentec QVGA Display Power Consumption

Variable Speed Driver (Schneider Electric Altivar 212)	
Characteristic	**Value**
Operating Voltage	10V
Operating Current	10mA
Power Consumption	100mW

Table 20: Closed Loop Control System Variable Speed Driver Power Consumption

Voltage Regulator (L7805CV)	
Characteristic	**Value**
Operating Voltage	10V
Operating Current	500mA
Power Consumption	5000mW

Table 21: Closed Loop Control System Voltage Regulator Power Consumption

Relay Module (SRD-05VDC-SL-C)	
Characteristic	**Value**
Operating Voltage	5V
Operating Current	85mA
Power Consumption	425mW

Table 22: Closed Loop Control System Relay Module Power Consumption

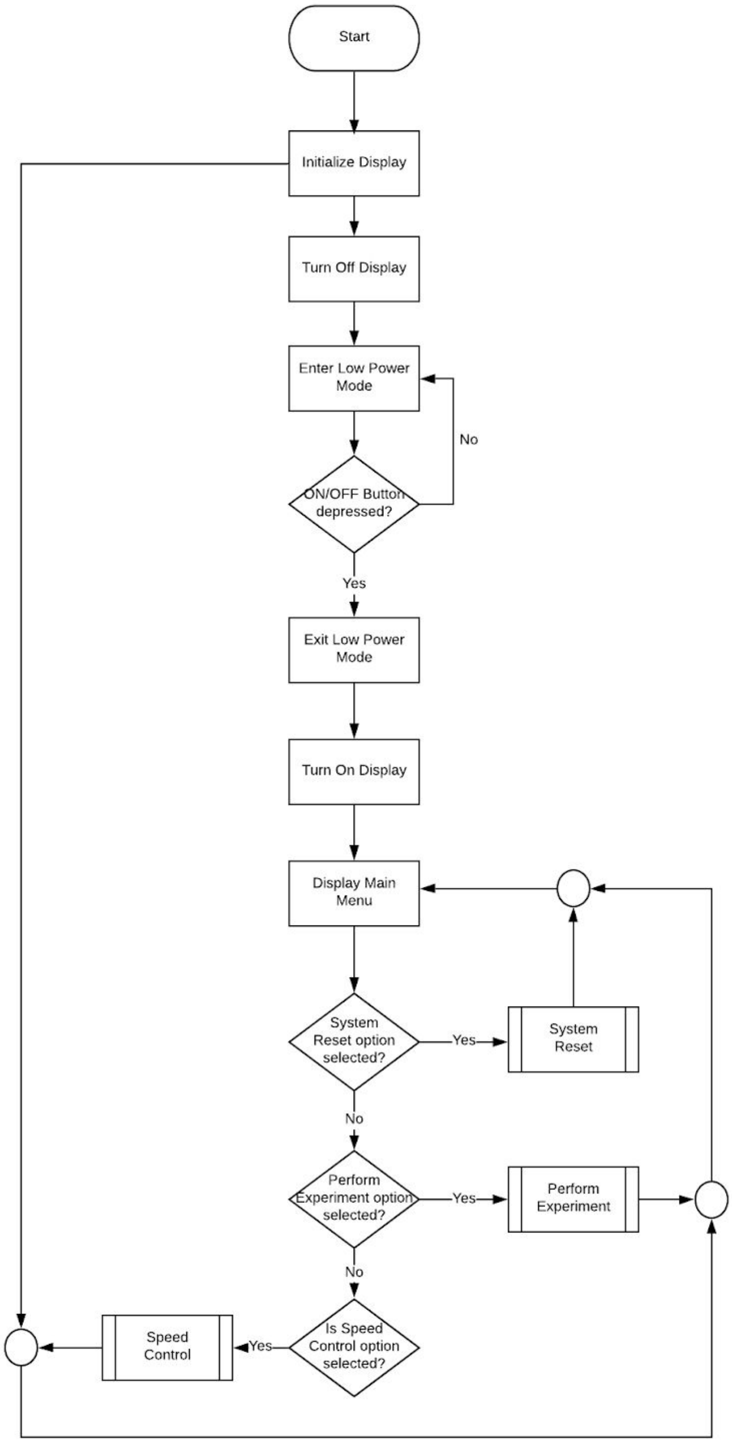


Fig. 16. Master module main flowchart

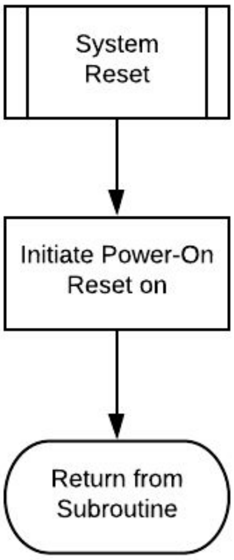


Fig 17. Master module System Reset subroutine flowchart

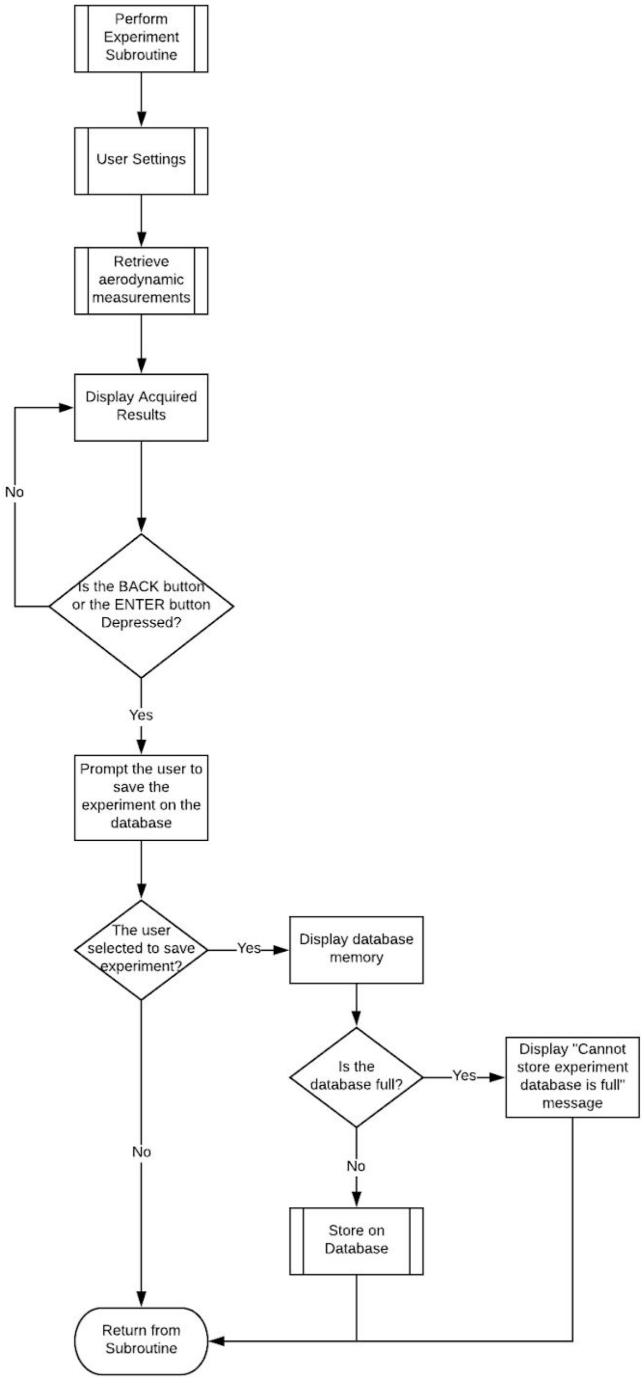


Fig. 18. Master module perform experiment subroutine flowchart

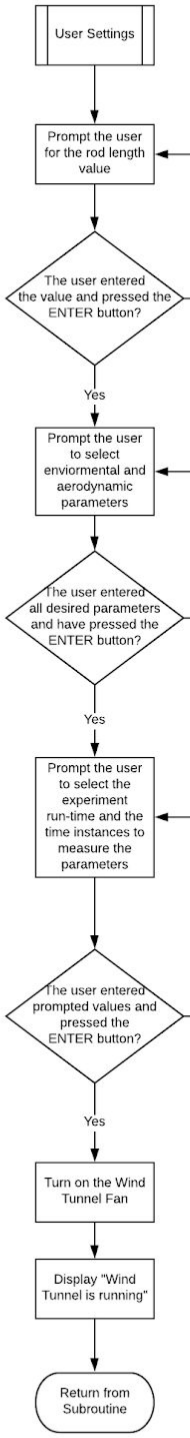


Fig. 19. Master module user setting subroutine flowchart

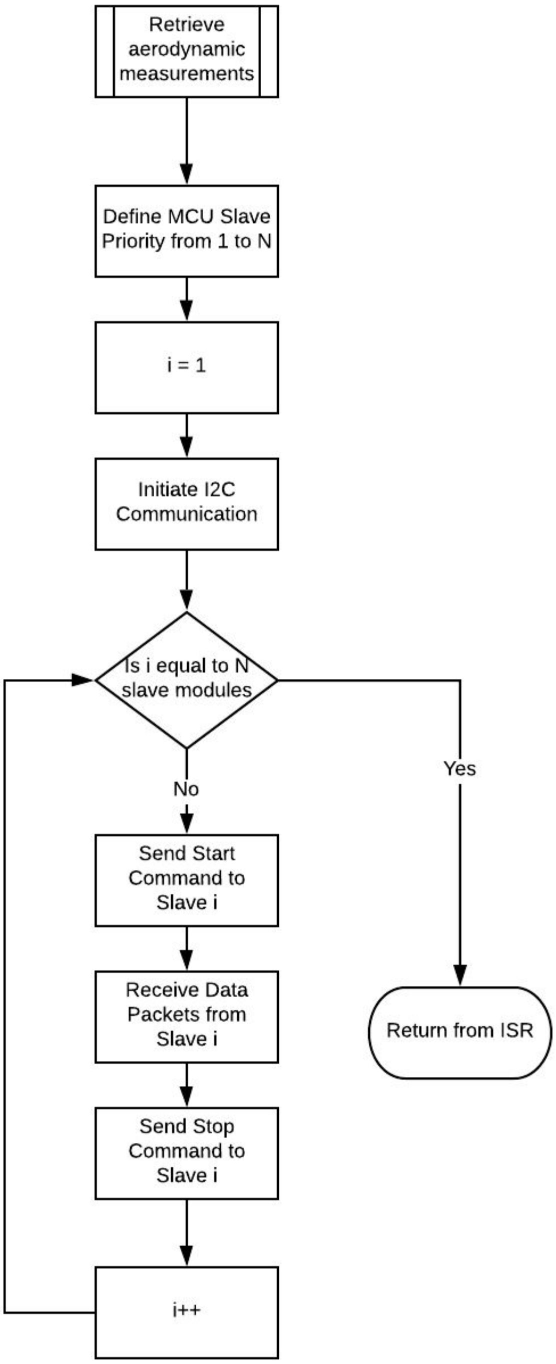


Fig. 20. Master module retrieve aerodynamic measurement subroutine flowchart

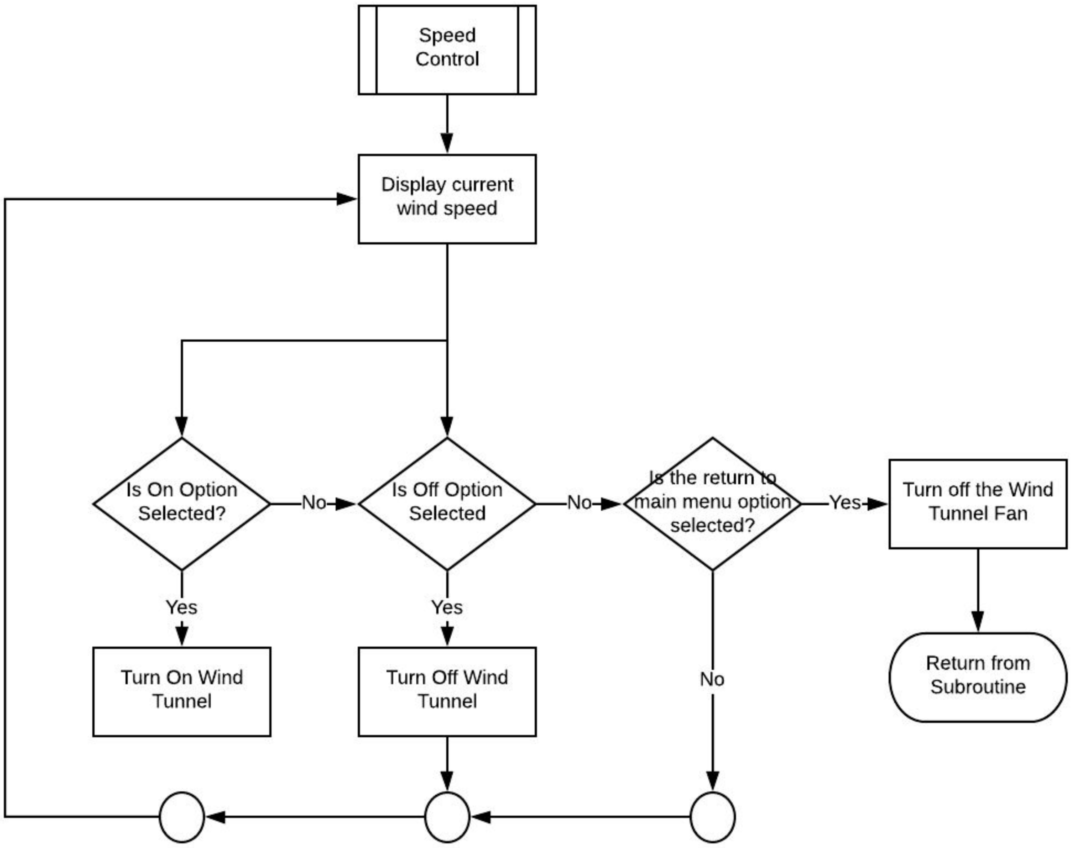


Fig. 21. Master module Speed Control Subroutione subroutine flowchart

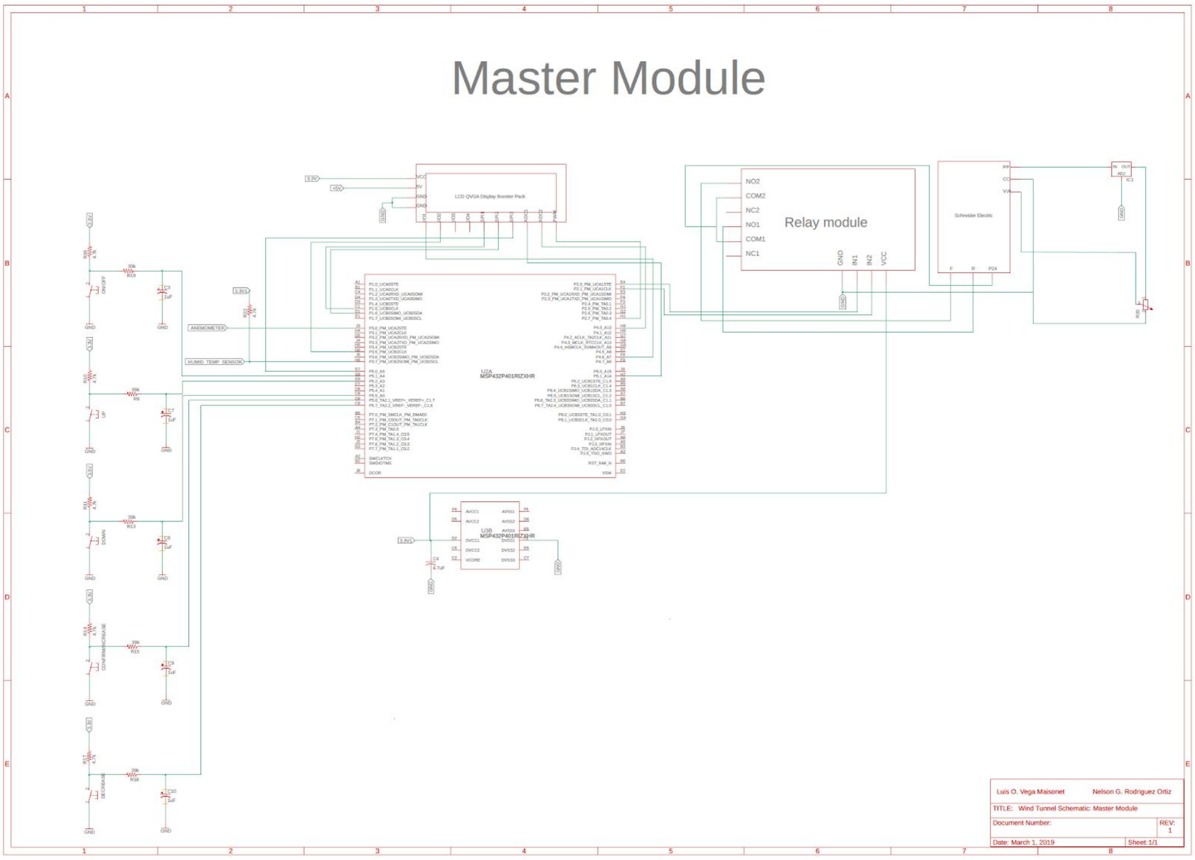


Fig. 22. Master module schematic

- Pressure Module <- *(Kahlil)*
  The pressure module measures the distribution of pressure upon the object to be tested inside the wind tunnel test section, using 16 barometric pressure sensors. The pressure sensors will be interfaced with a microcontroller, in which the measurements will be gathered via a communication protocol. The microcontroller used for this module is the Arduino Mega 2560, where it has the Adafruit libraries for the BMP280 barometric pressure sensors installed and makes the interfacing easier due to the specific timings of the communication between the microcontroller and the BMP280 barometric pressure sensors.

  The BMP280 pressure sensors have the option to be interfaced using an I2C or SPI communication. The code provided by Adafruit Industries is made for an I2C communication between an Arduino and a BMP280 barometric pressure sensor, but it lacks the ability to connect more than 2 barometric pressure sensors due to that the I2C for these pressure sensors has only 1 bit for their address and therefore only 2 possible addresses. With 2 different addresses, it is not recommended to connect more than 2 different barometric pressure sensors because then you won't have control of each pressure sensor individually which is the main idea. Since there are 16 BMP280 pressure sensors, the option of using the I2C protocol seems complex unless we use external additional components such as multiplexers. Therefore, we chose the second and last communication protocol that these BMP280 pressure sensors are compatible with, which is the SPI communication protocol. Using SPI communication protocol requires the use of 4 GPIO pins, for SDI, SDO, CS, and Clk, which are Serial Data In, Serial Data Out, Chip Select, and Clock respectively. In order to make an interface with 16 barometric pressure sensors using the SPI communication with the Arduino microcontroller, a strategy or topology is needed since it would require a lot of GPIO pins if treated individually. Two of these were analyzed, which are: Ring Topology and the Star Topology.

1) Ring Topology

   The ring topology involves connecting the SIMO of the microcontroller to the SDI of a pressure sensor in this case, and the SDO of this sensor to the SDI of the next sensor, and so on. The SDO of the second to last sensor will be connected to the SDI of the last sensor, and the SDO of the last sensor connected to the SOMI of the microcontroller. This way we will have a connection similar to the one shown on Figure 23.

   a) Advantages:
      1. Data flows in a unidirectional way, minimizing the amount of collisions between packets. [19]
      2. Data can move through bus at high speed. [19]
   b) Disadvantages:
      1. Failure of one node damage the whole bus. [19]
      2. Communication delay increases as amount of devices connected increases. [19]

2) Star Topology

   The star topology involves connecting the SIMO of the microcontroller to the same node of the SDI of all pressure sensors, and the SOMI to the same node of the SDO of all pressure sensors. The only way to distinguish or communicate with a pressure sensor is via the CS, which in this case would be 16 different GPIO pins. A similar connection of this topology is shown on Figure 24.

   a) Advantages:
      1. Main microcontroller has access to each device. [19]
      2. A device won't damage the communication between the main microcontroller and the other devices. [19]
   b) Disadvantages

1. The whole connection goes down when the main microcontroller stops working. [19]
2. The performance of the communication depends on the performance of the main microcontroller.

The Star topology was chosen since the Ring topology, because of the way the pressure sensors were made by the Adafruit Industries, is not viable. This is because when a CS is disabled, when trying to gather the measurement of the sensor following that one, it will transmit trash. According to their datasheet and some testings made at the laboratory, whenever the bus is active and the CS is disabled, the SDO will contain the hexadecimal byte value of 0xFF. Even though the Ring topology would be more suitable for a design such as our Pressure Module design, this particularity has led us to choose the Star topology instead.

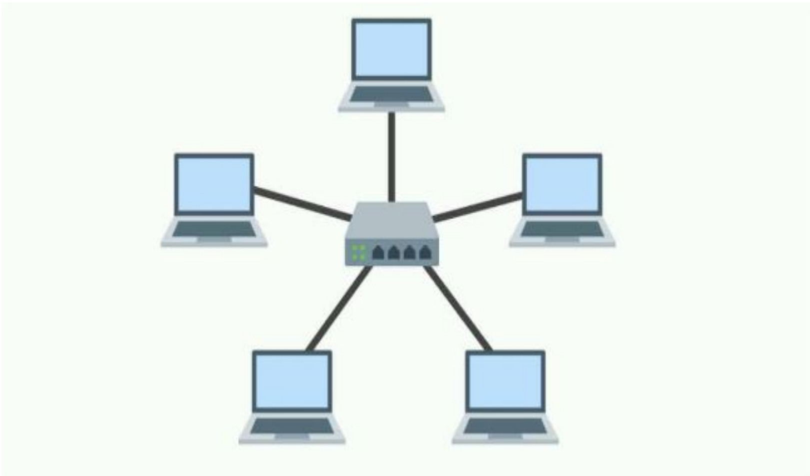


Figure 23: Ring topology representation [17]

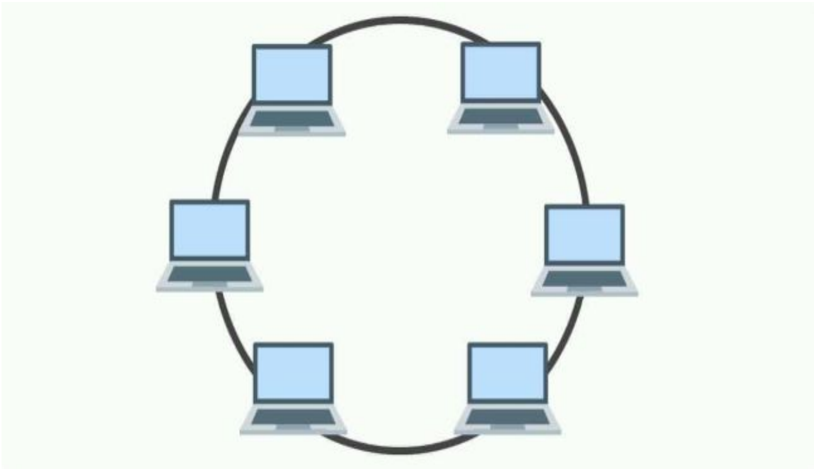


Figure 24: Star topology representation [17]

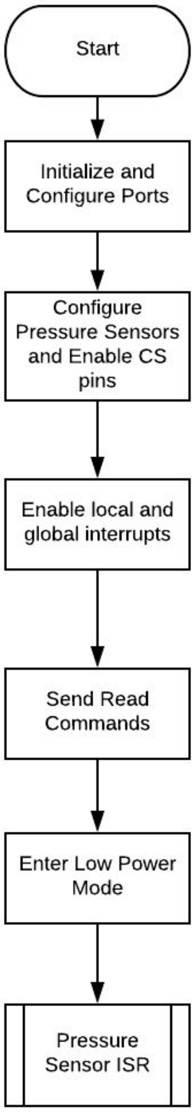


Fig. 25. Pressure Module Main Flowchart

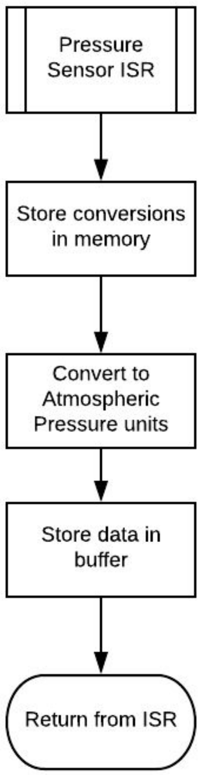


Fig. 26. Pressure Module ISR Flowchart

The enclosure currently being used for the barometric pressure sensors is that of a pill organizer which can be found in nearby pharmacies. It has been set up in a 2 x 8 matrix array for the 16 barometric pressure sensors.

A picture of the Pressure Module with the enclosure for the 16 barometric pressure sensors are provided below.

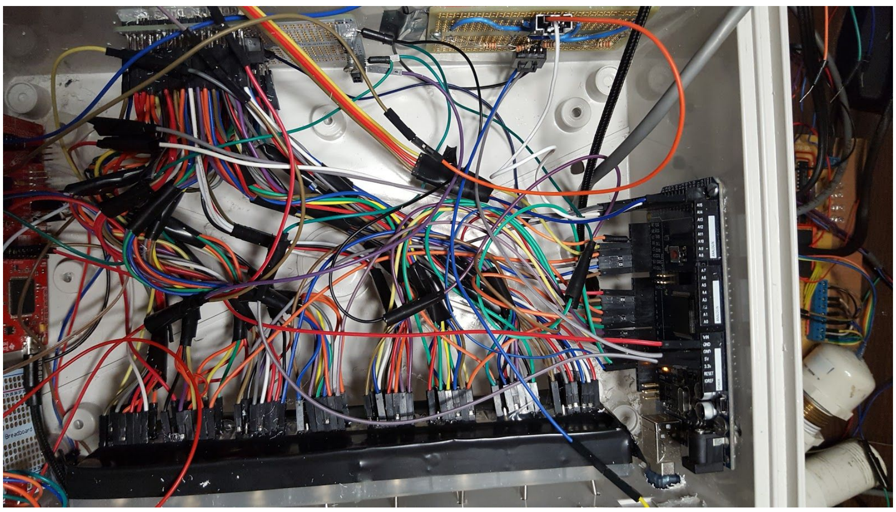


Fig. 27. Top view of the pressure module inside system enclosure

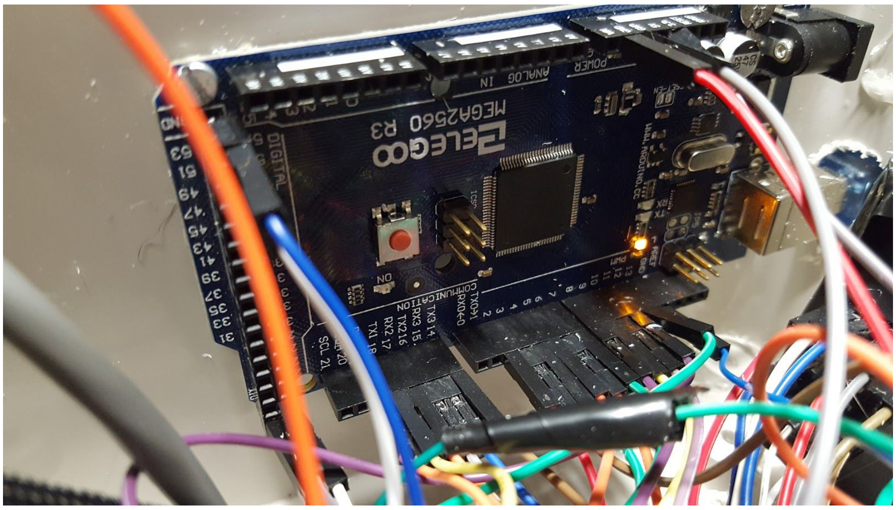


Fig. 28. Arduino Mega 2560

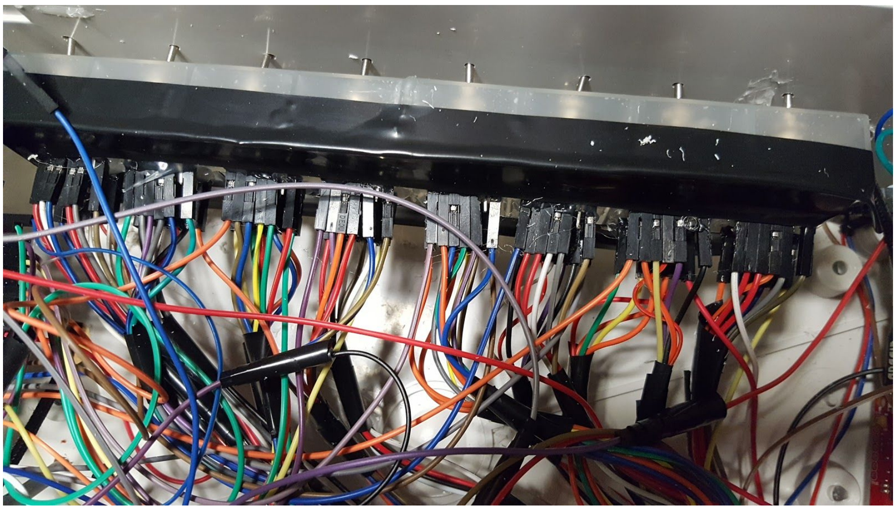


Fig. 29. BMP280 barometric pressure sensors pill organizer enclosure

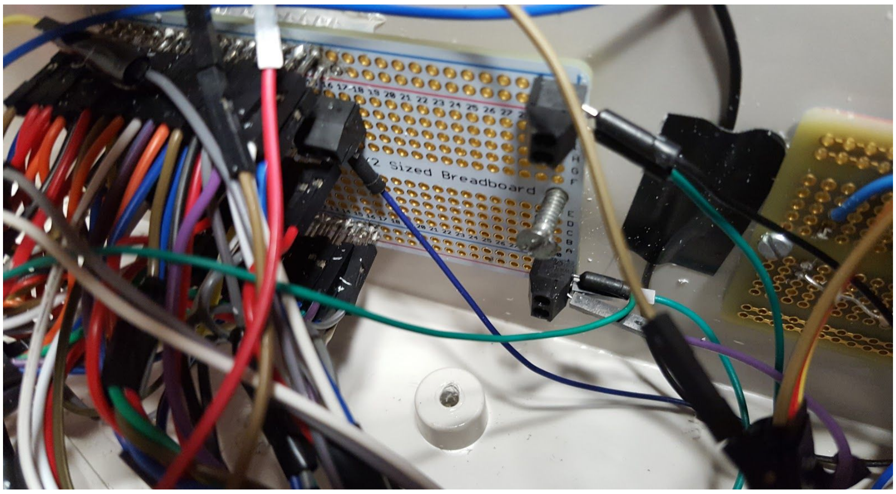


Fig. 30. Adafruit protoboard used for the pressure module

The wiring of the BMP280 barometric pressure sensors is provided below, where an Adafruit Protoboard was used. Each sensor share the same nodes of VIN, GND, SCK, SDO, and SDI, but not their Chip Select. The pins for the chip selects connected on the Arduino Mega 2560 are as follows:

Sensor 1: Pin 2
Sensor 2: Pin 3
Sensor 3: Pin 4
Sensor 4: Pin 5
Sensor 5: Pin 6
Sensor 6: Pin 7
Sensor 7: Pin 8
Sensor 8: Pin 9
Sensor 9: Pin 10
Sensor 10: Pin 14
Sensor 11: Pin 15
Sensor 12: Pin 16
Sensor 13: Pin 17
Sensor 14: Pin 22
Sensor 15: Pin 23
Sensor 16: Pin 24

The following figures show the Adafruit protoboard with the wiring for the common nodes and the BMP280 barometric pressure sensor.

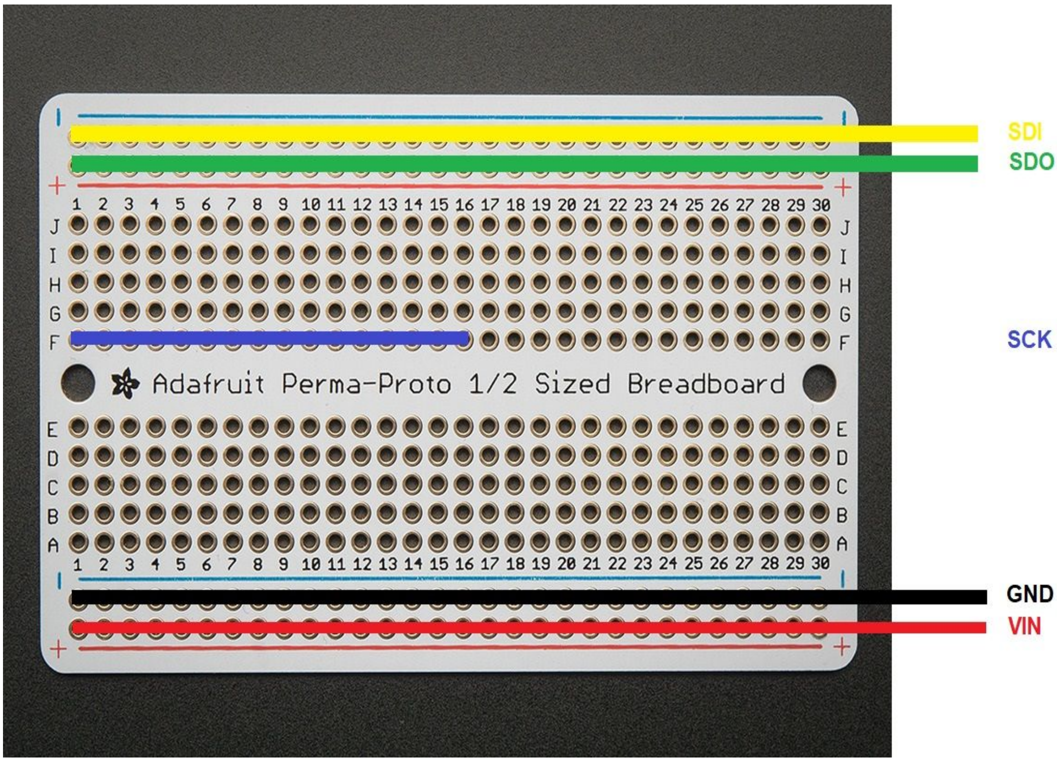


Fig. 31. Adafruit protoboard wiring for the BMP280 barometric pressure sensors

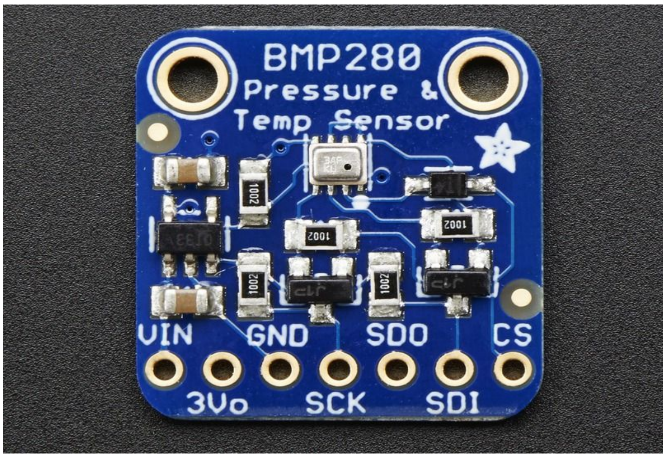


Fig. 32. BMP280 barometric pressure sensor

The VIN is driven by the 3.3V pin.
The GND can be connected to any GND pin.

The SCK is connected to Pin 13.
The SDO is connected to Pin 12.
The SDI is connected to Pin 11.

The UART pins used to interface with the master module are: TX1 (18) to the RX of the master module and RX1 (19) to the TX of the master module.

Pin 52 is connected to an LED that will turn on when results are being transmitted to the master module.

- Balance Module <- *(Nelson)*

The function of the module is to determine the drag and lift forces that the wind exerts to the object under study inside the wind tunnel. The emphasis of this module is in the signal conditioning circuit that amplifies the analog signals obtained from the wind tunnel balance load cells and filters the undesired noise produced at high frequencies. Refer to Appendix E for a block diagram of the Balance Module.

Before designing the Balance Module hardware, the Empotra'o Solutions team calibrated the balance with known weights (Fig. 33 and 34) to obtain the load cells transfer characteristic (Fig. 35). The resulted plot was a straight line with a slope of approximately 0.2073 mV/lb. From the results obtained, it was concluded that the relation between weight forces and the load cells output voltages is linear. Moreover, it was found that the maximum weight that the load cells can measure was 20 kg (45 lb) which means that the maximum load cells output voltage is approximately 10 mV according to the linear equation that characterize the load cells force-voltage relation.

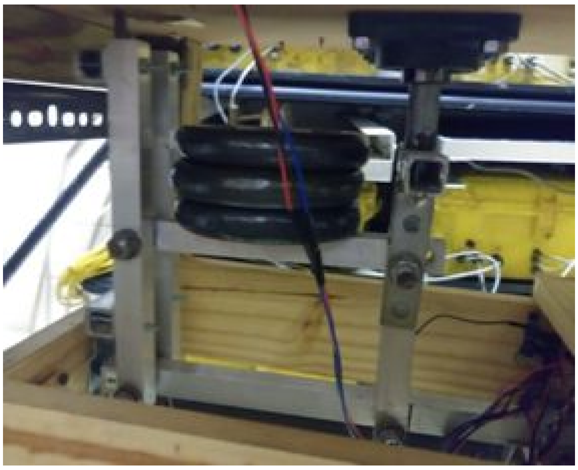


Fig. 33. Aerobal electronic balance calibration

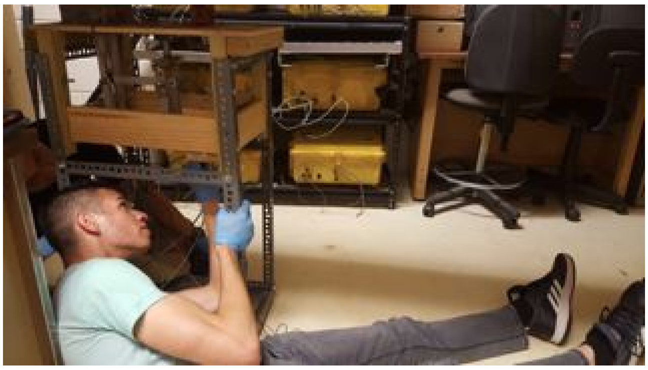


Fig. 34. Acquisition of load cells output voltages

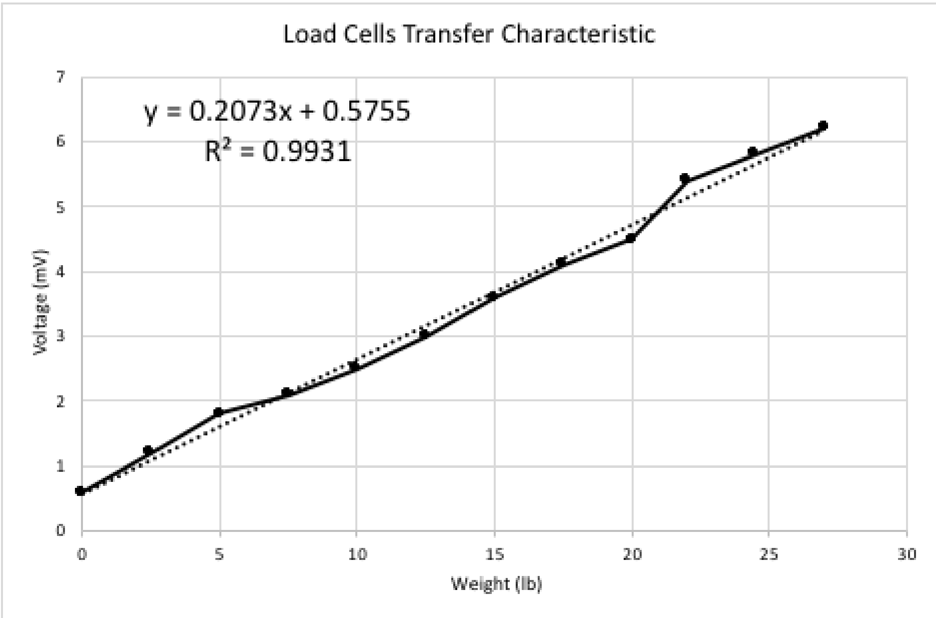


Fig. 35. Load Cells Transfer Characteristic

Two alternatives were considered for the signal conditioning circuit for the Balance Module. The difference between these alternatives is the Vref used for the ADC, specifically, Alternative 1 was designed using a Vref of 3.3V and Alternative 2 was designed using a Vref of 5V. The instrumentation amplifiers chosen were the TI INA333 because they have a Common Mode Rejection Ratio (CMRR) of 100 dB which is sufficient to acquire an accurate measurement of the output voltage of the load cells. According to the datasheet [56], the gain is calculated using the relation

$$Gain = 1 + (100k\Omega/RG)$$

Where RG is a resistance used to set the INA333 desired gain.

For an ADC Vref of 3.3V, the gain should be 330V/V with a $RG = 303.03\Omega$. For an ADC Vref of 5V, the gain should be 500V/V with a $RG = 200\Omega$. Commercial values of 284.7Ω and 220Ω were chosen for the design.

Two filtering circuits were considered: a simple RC low-pass filter circuit for Alternative 1 and an active op-amp circuit for Alternative 2. According to the MSP432P401R datasheet [55], the filter is constrained by

$$C \geq 20C_I = 20(15pF) = 0.3nF$$

Where C_I is the input capacitance of the MSP432P401R ADC. Therefore, $C = 0.47nF$ was chosen.

Also, the RC constant is constrained by the acquisition time by

$$t_{samp} \geq (n+1)ln(2)RC$$

Where n is the number of bits of the ADC. In the case of the WindTel project, the ADC has a 14-bit resolution.

Solving (4) by R we get

$$R \leq 41.54\Omega$$

Therefore, $R = 39\Omega$ was chosen.

Similar approach was used to calculate the components values for the active op-amp filter using the TI OPA743. The values chosen were $R_1 = 10k\Omega$, $R_2 = 11k\Omega$ and $C = 1.5pF$ which provides an approximated unity gain (1.1V/V). Finally, a buffer stage was implemented using the TI OPA363 which is an amplifier with a CMRR of 90 dB used in data acquisition applications.

In addition, a power analysis was performed to ensure that the components operate within the recommended conditions. Most of the power consumed by the Balance Module is determined by the amplifier blocks and the MCU. The procedure consists in calculating the power by considering the operating voltage and current of each component in active mode. The analysis was performed for both Alternative 1 and Alternative 2 components.

MSP432P401R	
Characteristic	**Value**
Operating Voltage	3.3V
Operating Current	3.9mA
Power Consumption	12.87mW

Table 23. MSP432P401R Power Consumption

Instrumentation Amplifiers (INA333)	
Characteristic	**Value**
Operating Voltage	5V
Operating Current	50µA
Power Consumption	250µW

Table 24. INA333 Amplifiers Power Consumption

High Performance Operational Amplifiers (OPA363)

Characteristic	Value
Operating Voltage	5V
Operating Current	1.1mA
Power Consumption	5.5mW

Table 25. High Performance Operational Amplifiers Power Consumption

High Speed Operational Amplifiers (OPA743)	
Characteristic	**Value**
Operating Voltage	5V
Operating Current	1.1mA
Power Consumption	5.5mW

Table 26. High Speed Operational Amplifiers Power Consumption

Fig. 36 and 38 show the schematic design alternatives for the Balance Module Signal Conditioning Circuit. A Parametric Analysis using the load resistance as the parameter was performed in PSpice (Fig. 37 and 39) to demonstrate circuit functionality. Table 27 and 28 provides the percentages of error between the calculated and simulation results of different load cells output voltage values for the Alternative 1 conditioning circuit and Alternative 2 conditioning circuit, respectively, when $R_L = 1k\Omega$.

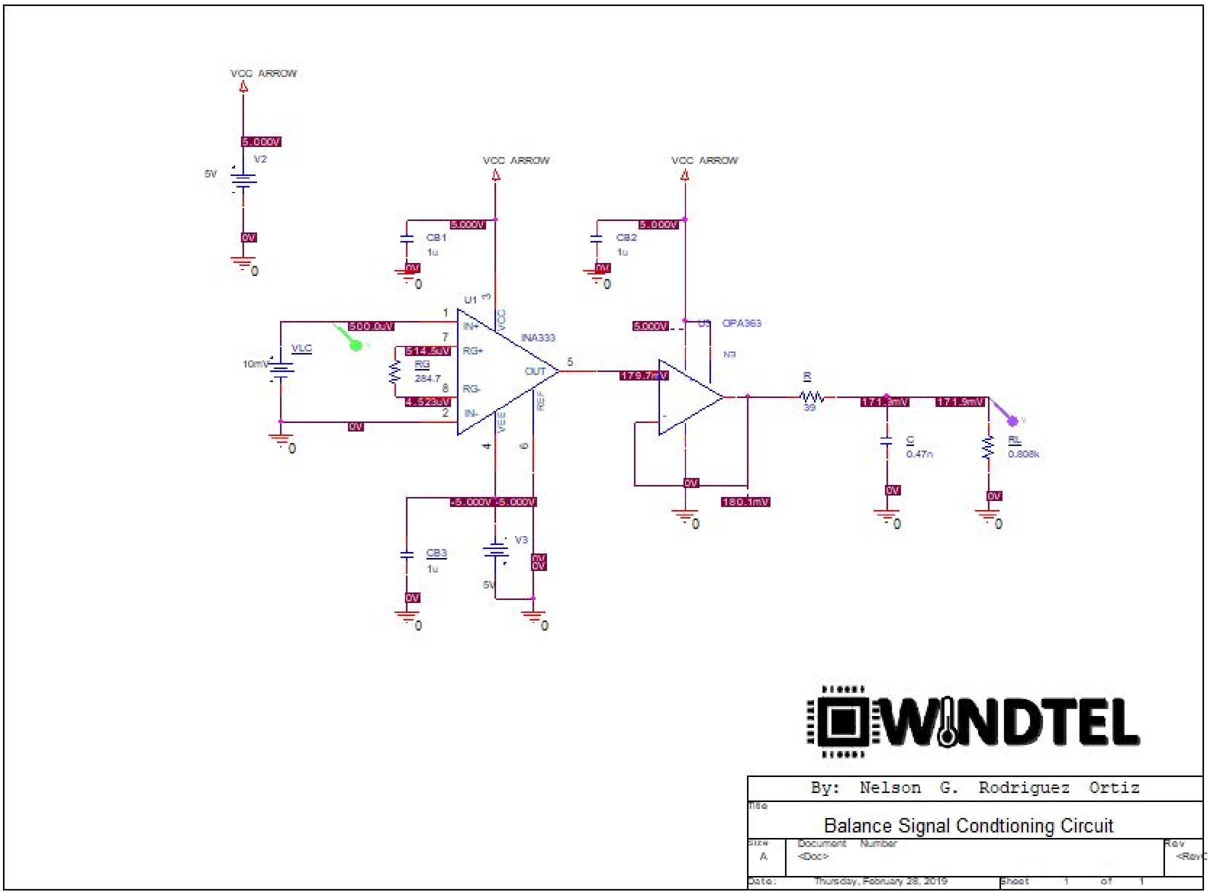


Fig. 36. Balance Module Signal Conditioning Circuit Alternative 1 Schematic

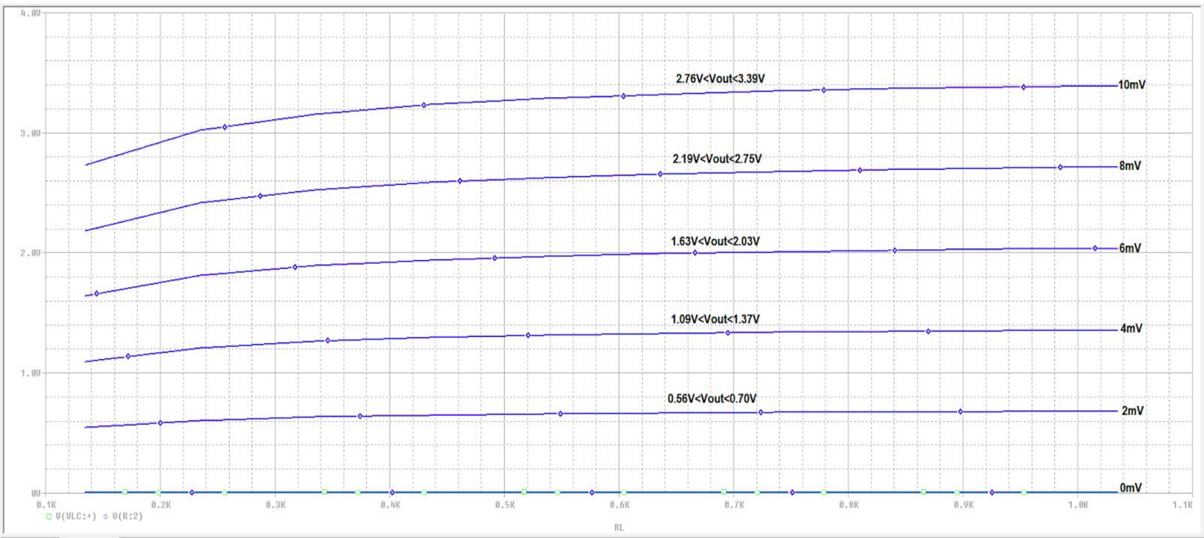


Fig. 37. Balance Module Signal Conditioning Circuit Alternative 1 Parametric Analysis Results

Load Cells Output Voltage	**Calculated Signal Conditioning Circuit Output Voltage**	**Simulation Signal Conditioning Circuit Output Voltage**	**Percentage of Error**
0mV	0V	0V	0%
2mV	0.70V	0.70V	0%
4mV	1.41V	1.37V	2.83%
6mV	2.11V	2.03V	3.79%
8mV	2.82V	2.76V	2.13%
10mV	3.3V	3.39V	-2.72%

Table 27: Comparison between calculated and simulated Alternative 1 signal conditioning circuit values when $R_L = 1k\Omega$

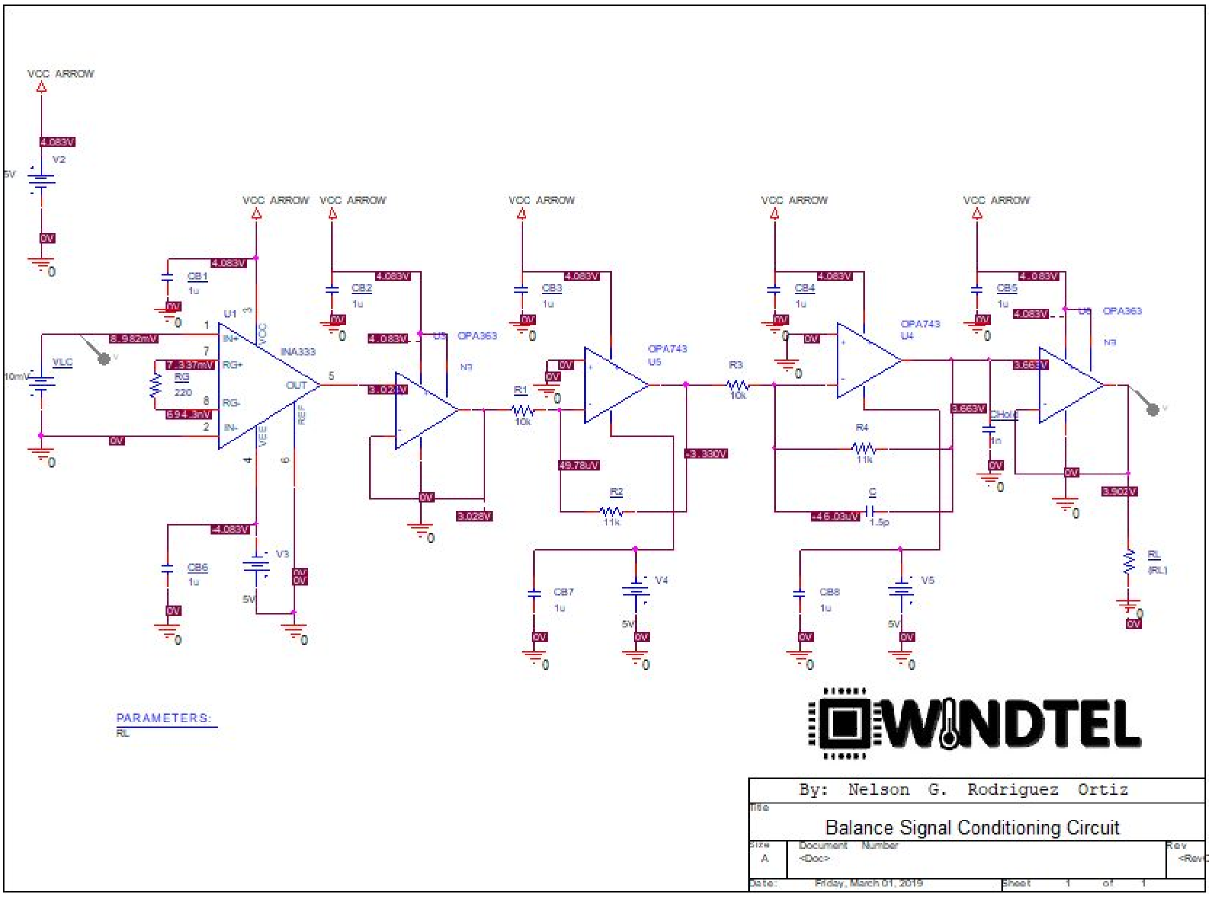


Fig. 38. Balance Module Signal Conditioning Circuit Alternative 2 Schematic

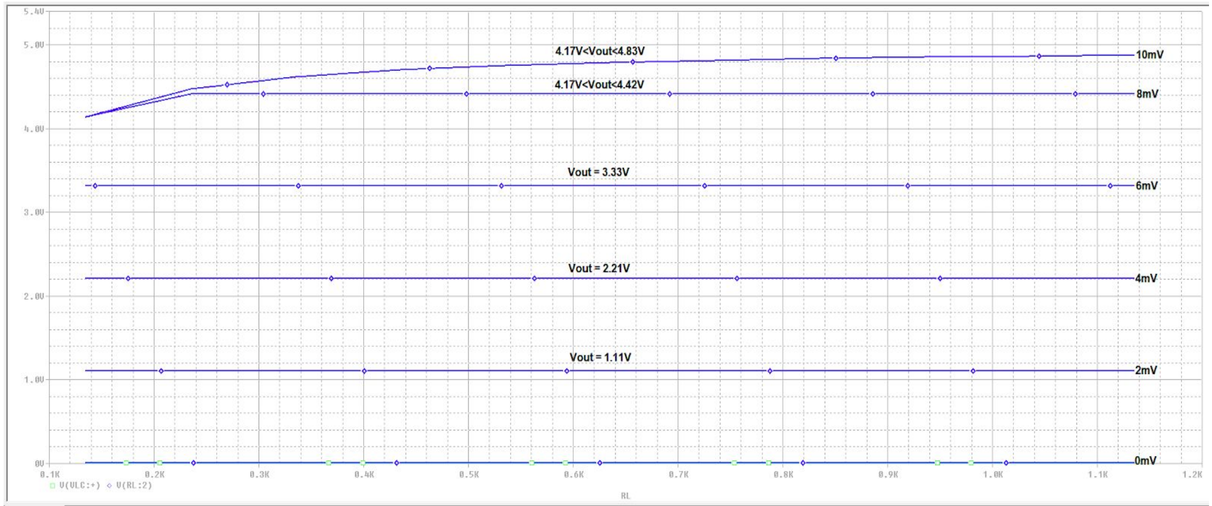


Fig. 39. Balance Module Signal Conditioning Circuit Alternative 2 Parametric Analysis Results

Load Cells Output Voltage	**Calculated Signal Conditioning Circuit Output Voltage**	**Simulation Signal Conditioning Circuit Output Voltage**	**Percentage of Error**
0mV	0V	0V	0%
2mV	1.09V	1.11V	-1.83%
4mV	2.19V	2.21V	-0.91%
6mV	3.28V	3.33V	-1.52%
8mV	4.37V	4.42V	-1.14%
10mV	5V	4.83V	3.40%

Table 28: Comparison between calculated and simulated Alternative 2 signal conditioning circuit values when $R_L = 1k\Omega$

After the design phase, a re-design of the signal conditioning circuit was necessary due to simulation of wrong INA333 SPICE model. The re-design only affected the amplification stage of the circuit where the INA333 is being operated in single-supply operation with a dc voltage of 5V with Vref of 2.5V in the new design. The new INA333 output voltage is denoted by

$$V_{out} = (1 + 100k\Omega/R_G)V_{in} + 2.5V$$

Solving for RG it is obtained that for Vin = 10mV, RG = 1.3kΩ.

The correct SPICE model of the INA333 was downloaded and the new circuit was simulated using PSpice (Fig. 40). Load cells where simulated in this case using the Wheatstone Bridge configuration with their nominal resistance of 280Ω. The 100Ω resistor was added between ground and the Wheatstone Bridge to keep VCC and VEE 0.1V above ground for linear operation. A parametric analysis was performed in one of the resistances of the bridge to obtain the output voltage vs differential input voltage characteristic (Fig. 41).

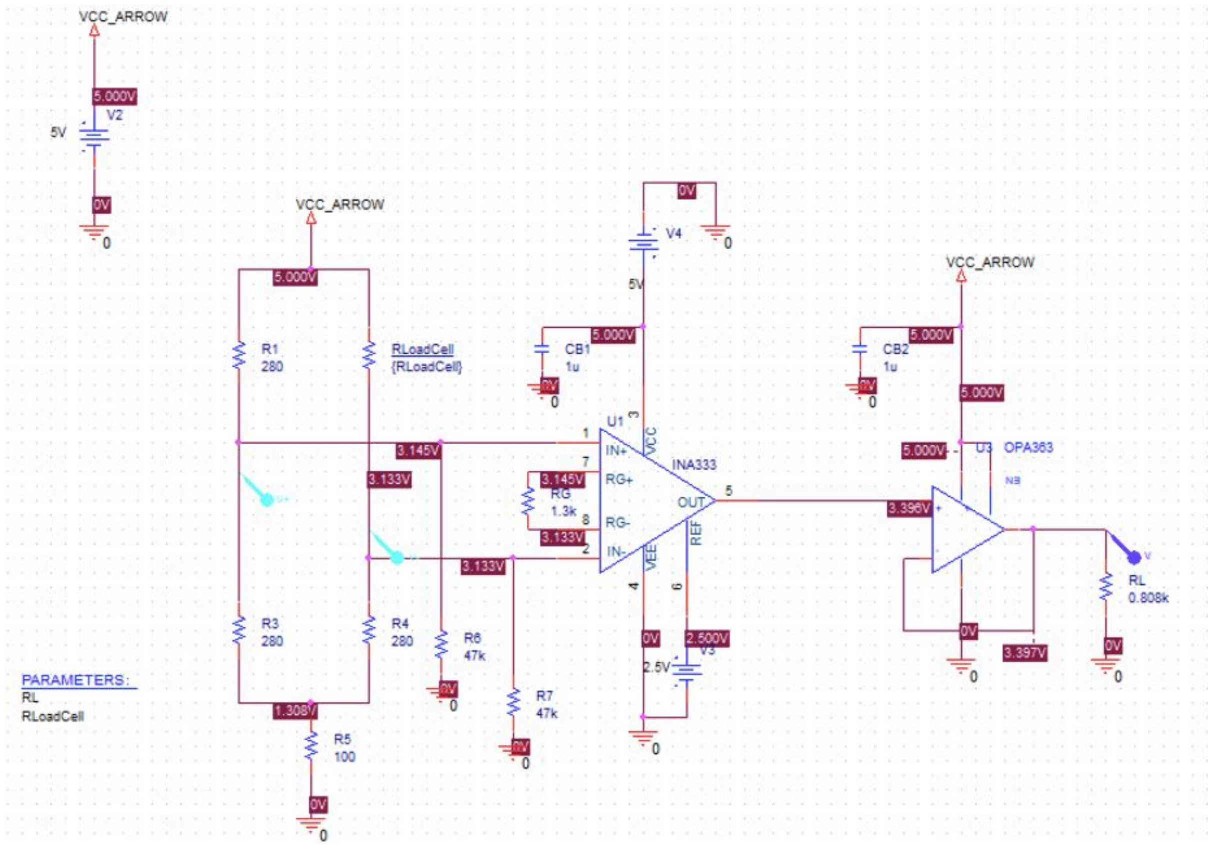


Fig. 40. Balance Module Signal Conditioning Circuit Redesign Schematic

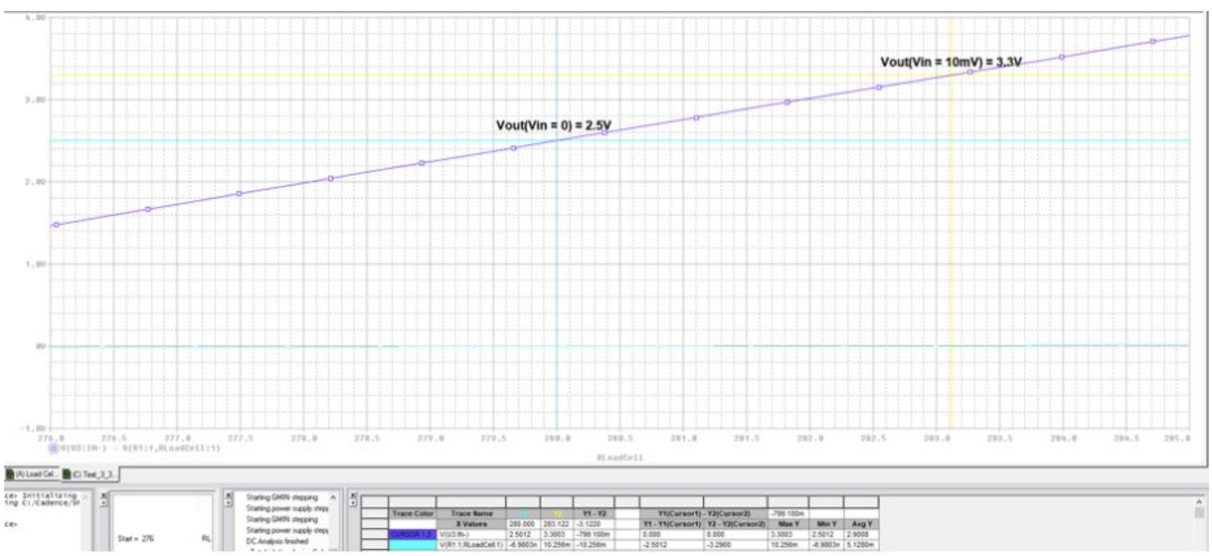


Fig. 41. Balance Module Signal Conditioning Circuit Redesign Parametric Analysis Results

In terms of the Balance Module firmware, the process consists in configuring the MCU ADC to store data in the ADC module memory and calculate the drag and lift forces by using the following formulas:

$$F_{Lift} = W_{Lift}(L_2/L_1)$$

$$F_{Drag} = W_{Drag}(L_4/L_3)$$

where L_1, L_2, L_3 and L_4 are the lengths shown in Fig. 42. The lengths L_1, L_2 and L_4 are 14.60cm, 16.75cm and 18.97cm, therefore, they will be defined in the module's firmware while L_3 depends on the rod length defined by the wind tunnel operator and, thus, it is specified by the operator. The force measurements will be stored in the UART buffer waiting for the Master Module request to send the data. The flowchart for the Balance Module is shown in Fig. 43.

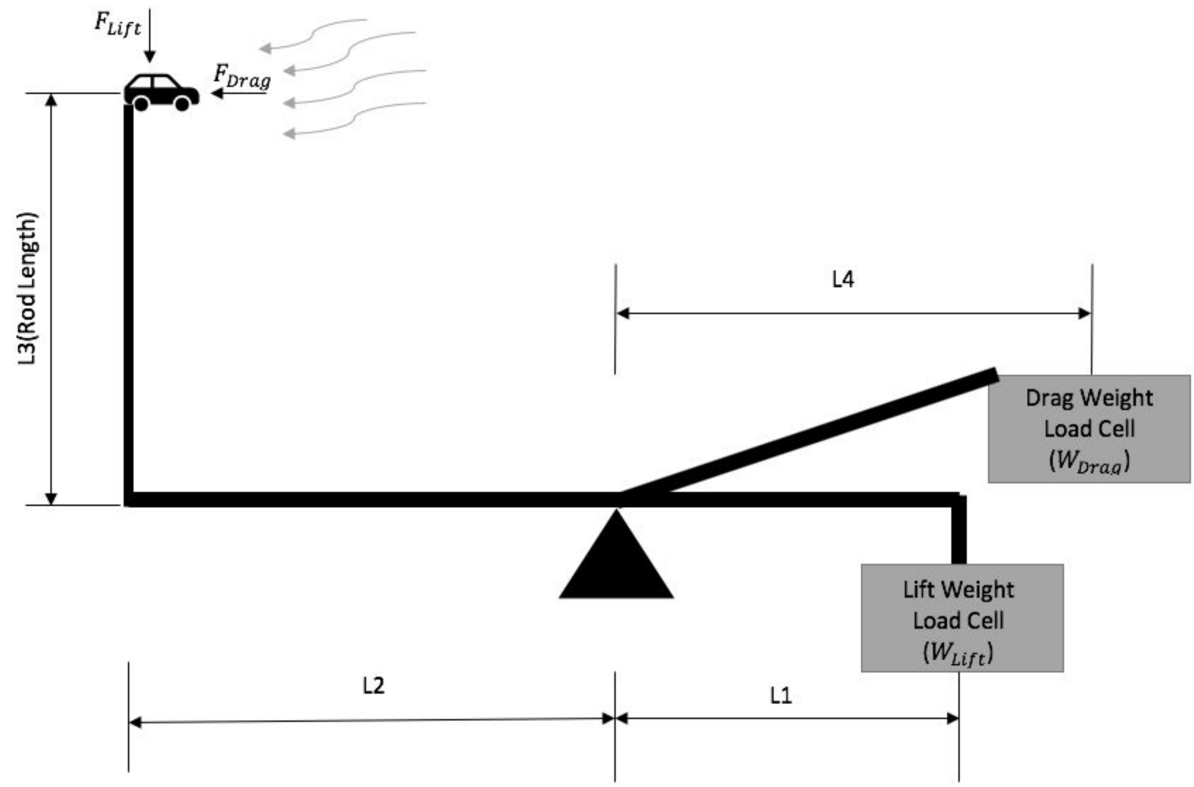


Fig. 42. Free Body Diagram with lengths definition

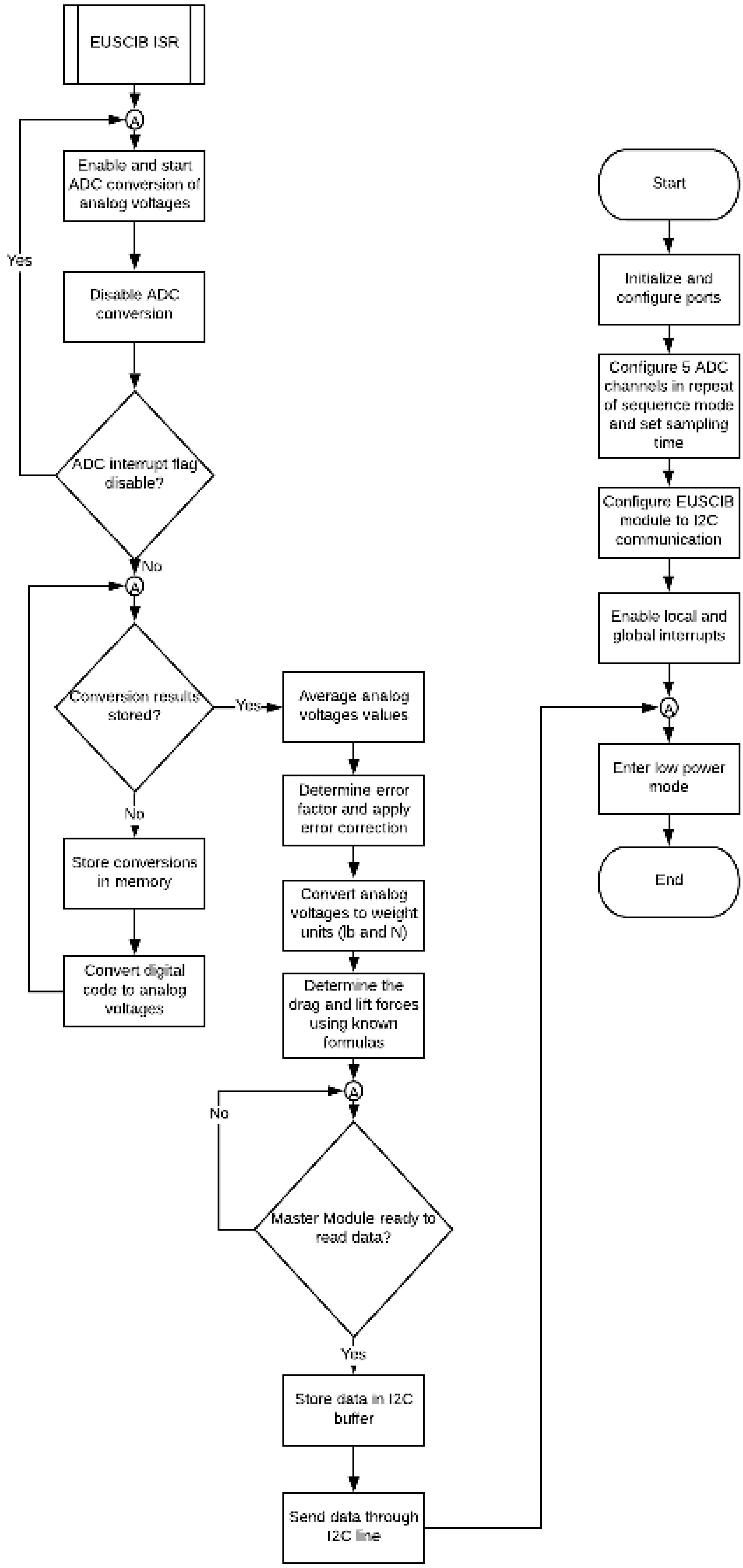

EUSCIB ISR
A
Enable and start ADC conversion of analog voltages
Yes
Disable ADC conversion
ADC interrupt flag disable?
No
A
Conversion results stored?
Yes
Average analog voltages values
No
Store conversions in memory
Convert digital code to analog voltages
Determine error factor and apply error correction
Convert analog voltages to weight units (lb and N)
Determine the drag and lift forces using known formulas
A
No
Master Module ready to read data?
Yes
Store data in I2C buffer
Send data through I2C line
Start
Initialize and configure ports
Configure 5 ADC channels in repeat of sequence mode and set sampling time
Configure EUSCIB module to I2C communication
Enable local and global interrupts
A
Enter low power mode
End

Fig. 43. Balance Module Firmware Flowchart

Fig. 44-49 show the implementation and testing of the signal conditioning circuit and its integration with the balance module. The C source code for the MSP432 can be downloaded in the following link: https://github.com/yldas/windtel/tree/master/Balance%20Module. Weights values can have a variability up to when weight is less than 5lb, but less variability when weight is greater than 5lb. This error is due to the initial strain that the load cells have because of the mechanical system of the wind tunnel balance. In addition, sensors readings are not constant and vary depending on the temperature of the room, therefore, the voltage must be averaged before converting it to forces.

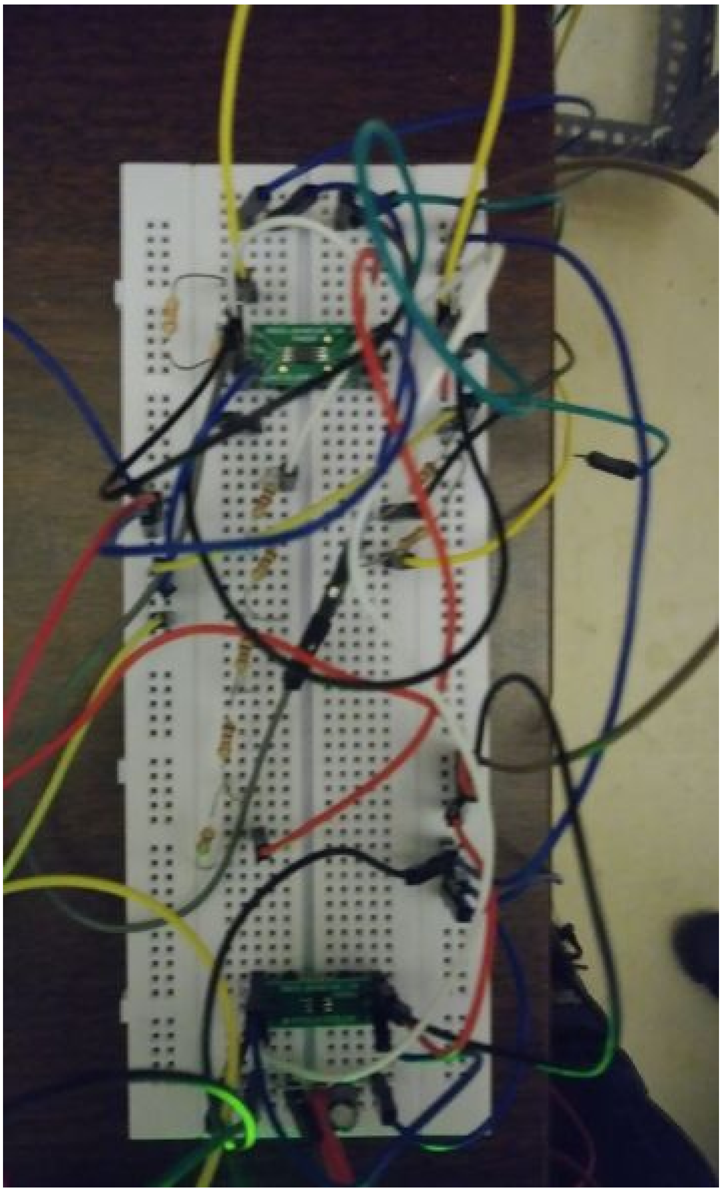


Fig. 44. Balance Module Signal Conditioning Circuit Prototype

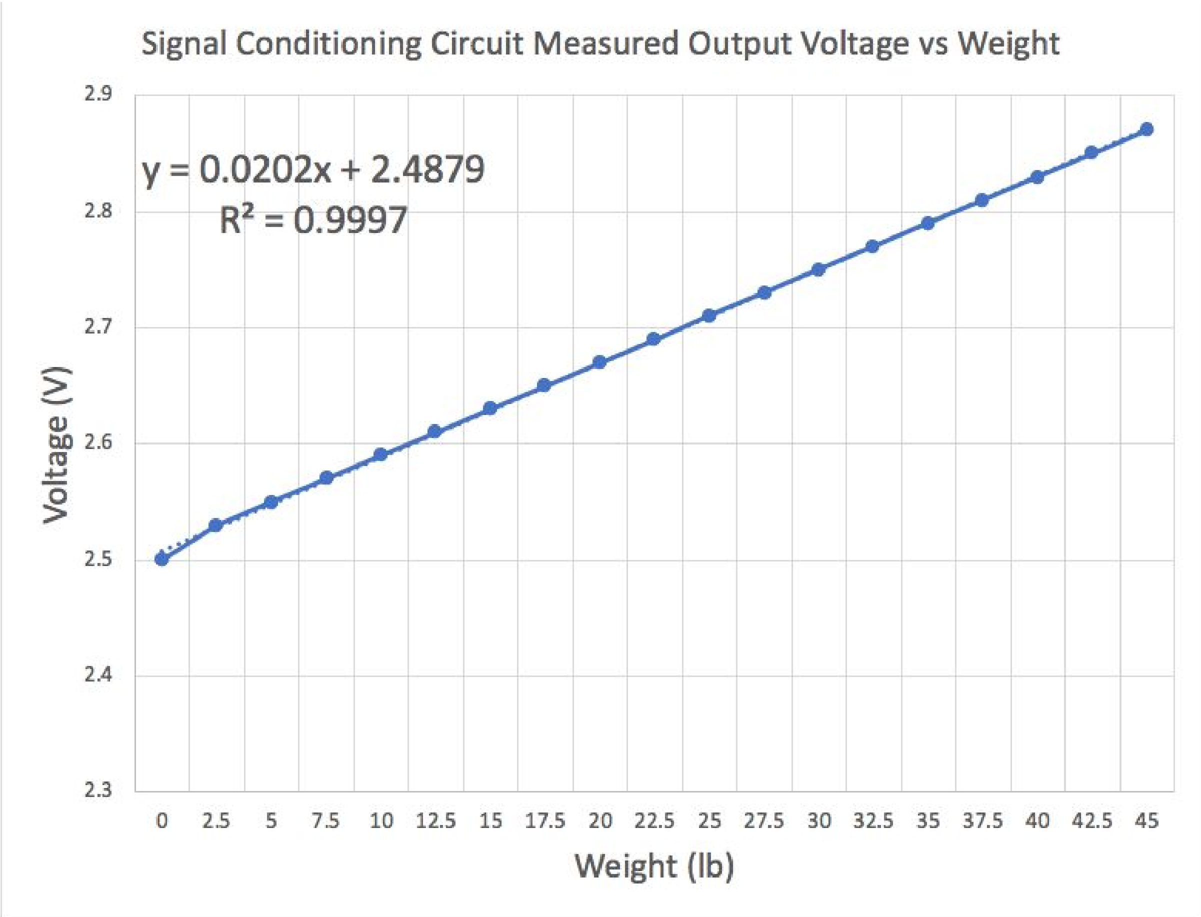


Fig. 45. Signal Conditioning Output Voltage vs Weight Transfer Characteristic

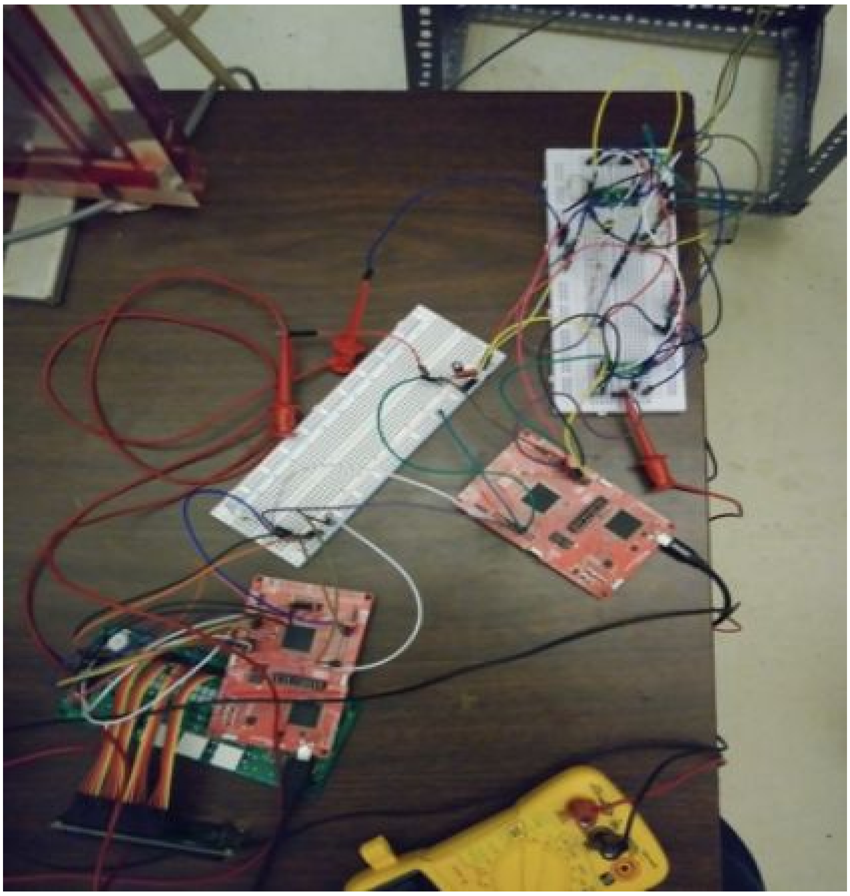


Fig. 46. Balance Module Top View

Fig. 47. Balance Module Test with No Weight

Fig. 48. Balance Module Test with 2.5lb Weight

Fig. 49. Balance Module Test with 7.5lb Weight

The Signal Conditioning Circuit and Balance Module printed circuit boards (PCBs) were designed (Fig. 50 and 51) using Autodesk Eagle software and sent to fabrication at the beginning of the integration phase of the WindTel project. In addition, a protoboard was designed for load cells output signal reinforcement (Fig. 52). The protoboard consists of a JBtek Breadboard Power Supply Module 3.3V/5V, two 100Ω resistors and two 47kΩ resistors. The purpose of the power supply is to apply regulated 5V to each load cell and its corresponding signal conditioning circuit. The 100Ω resistors keep the load cells differential outputs 0.1V above ground to ensure that the INA333 amplifier operates in linear operation. Finally, the 47kΩ resistors provide a current path for the input current of the INA333 amplifier. In sum, a total of 11 PCBs were assembled for the wind tunnel balance system.

Fig. 50. Signal Conditioning Circuit PCB

Fig. 51. Balance Module PCB

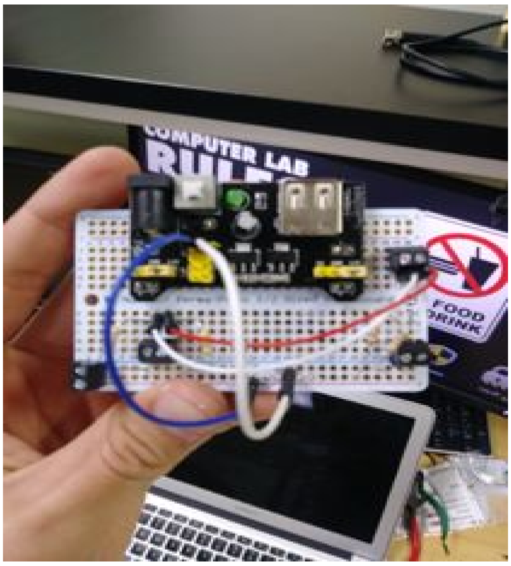


Fig. 52. Load cells output signal reinforcement circuit

The WindTel balance module and signal conditioning circuit PCBs were installed in the mechanical balance of the wind tunnel in the Wind Tunnel Laboratory (Fig. 53). The wires used to connect the load cells with their corresponding signal reinforcement circuit and signal conditioning circuit follow a color code (Fig. 54). It is important to mention that the -Output Signal wire is white instead of gray. Similarly, the output conditioned signal for each force is transmitted to the balance module PCB with wires in a shielded cable according to the color code shown in Fig. 55 and Fig. 56. The shielded cable transmits every output conditioned signal and ground (GND) to the balance module with minimum signal loss. Although the GND wire is not shown, GND is shared within the five signal conditioning circuits and with the balance module through a silver wire inside the shielded cable.

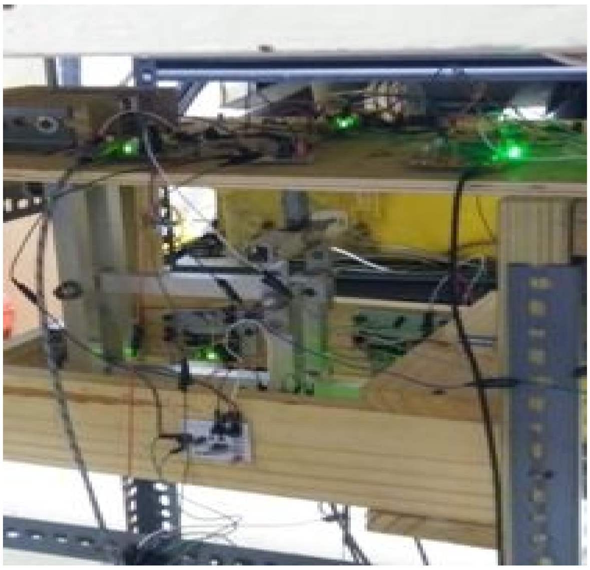


Fig. 53. Balance module and signal conditioning circuits installation

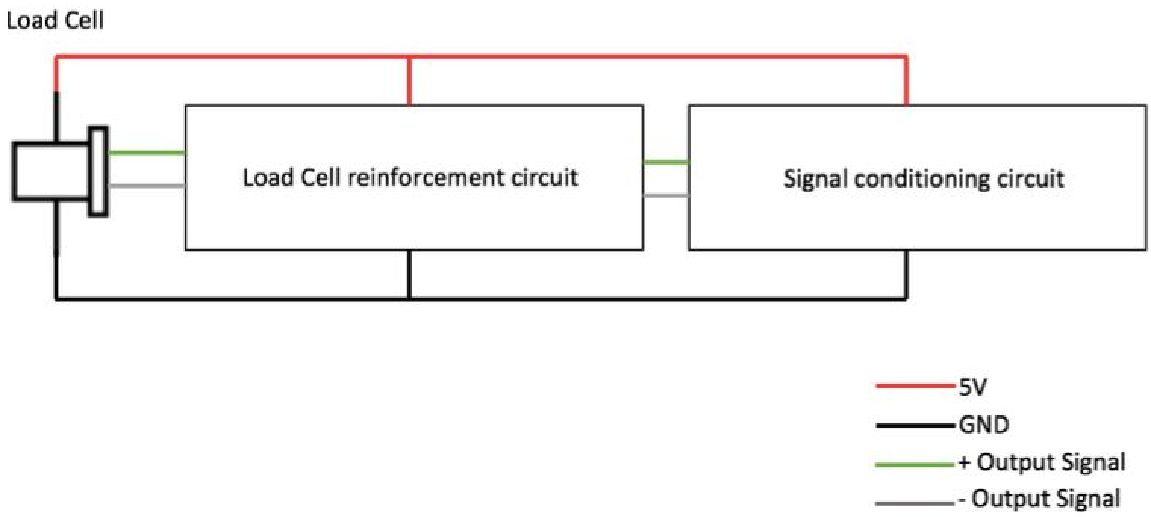
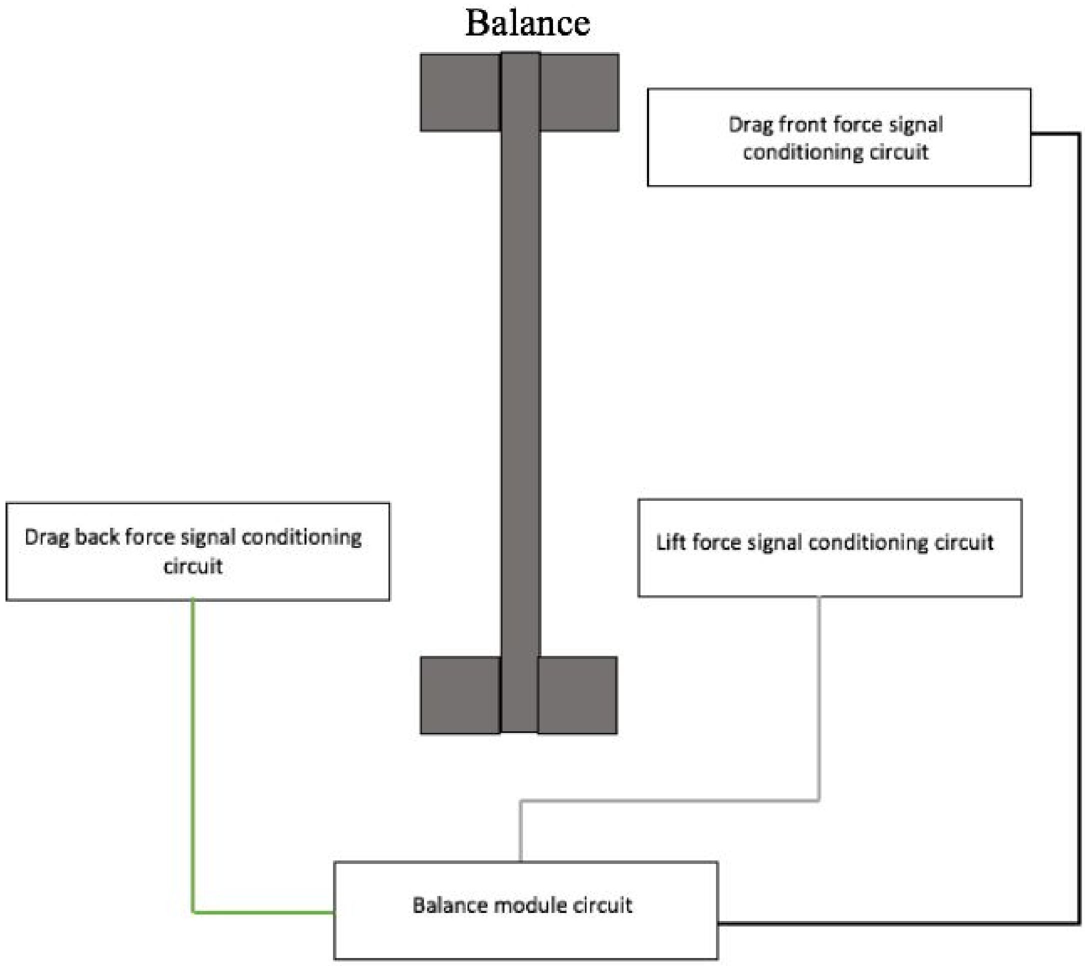


Fig. 54. Load cells interface with their corresponding reinforcement circuit and signal conditioning circuit



Note: The lift force signal conditioning circuit output wire is white.

Fig. 55. Top view of balance module interface with signal conditioning circuits

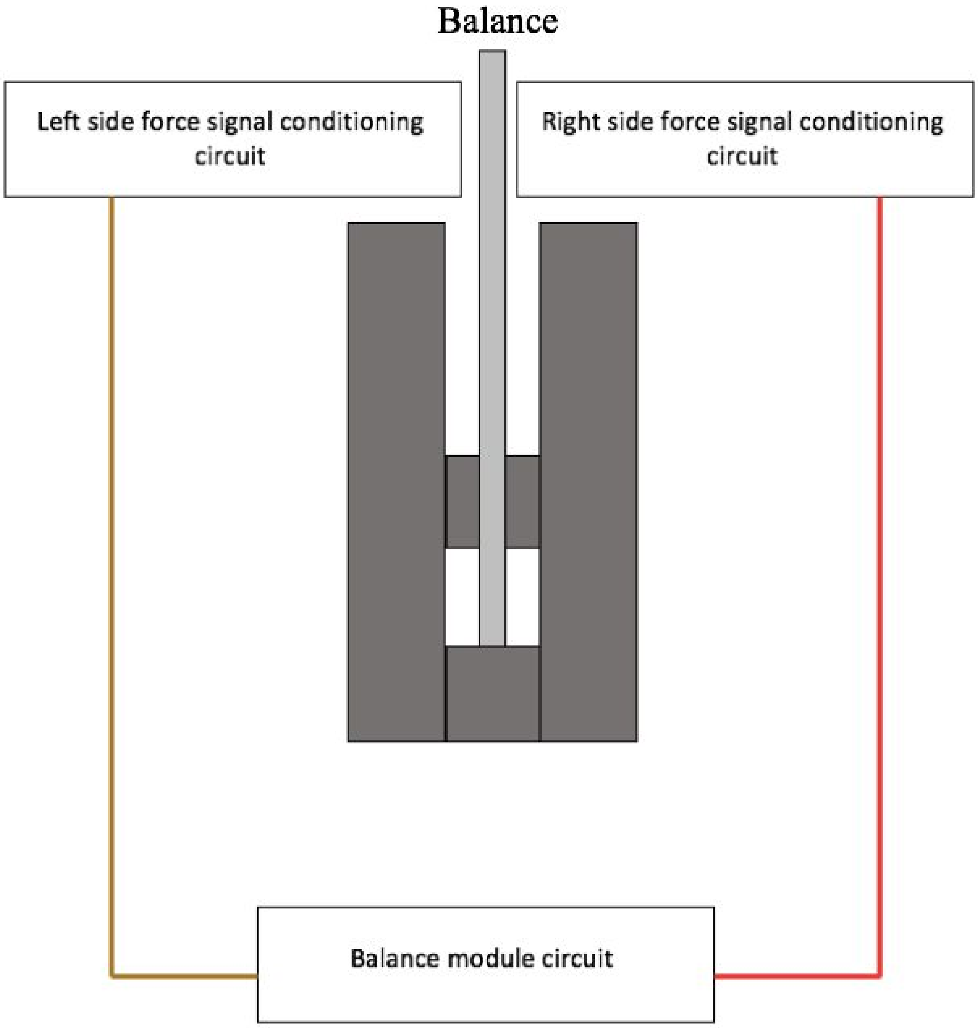


Fig. 56 Front view of balance module interface with signal conditioning circuits

The Balance Module communicates with the Master Module using Universal Asynchronous Receiver-Transmitter (UART) protocol. The communication starts when the Master Module sends a command to the Balance Module in the format shown in Fig. 57. When the Balance Module receives the command, the ADC starts conversion of the conditioned analog voltages from the Signal Conditioning Circuit every 2 seconds until the Master Module is ready to receive the measurements results. When the Master Module is ready to receive the results, the Balance Module sends XXX samples of each force (drag front, drag back, right side, left side and lift forces) and the transaction is completed.

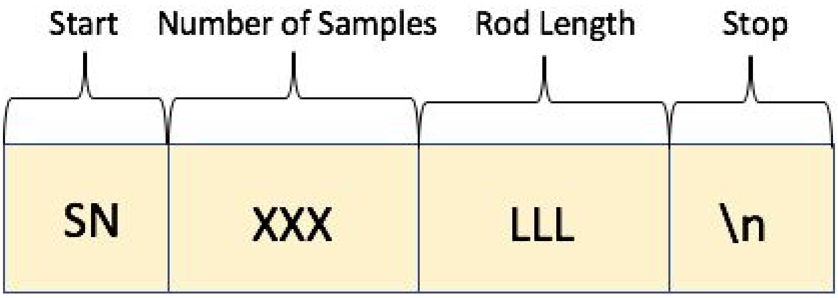


Fig. 57. Master Module commands to the Balance Module through UART

- Dynamic Module <- *(Misael)*

Fig. 60, Fig. 61, Fig. 62 and Fig. 63 below illustrate the data acquisition process for the Dynamic Measurements Module and the communication process between it and the Master Module. The Dynamic Measurements Module sends a start signal to the DHT22 Temperature and Humidity Sensor. Fig. 59 below shows the timing for the communication with this sensor. The acquisition process for the wind speed sensor consists of configuring a timer to count the number of pulses that the MCU GPIO receives in a given period of time. Every pulse triggers an IRS. The device datasheet then defines the relation between pulse frequency and wind speed. Figure 58 below illustrates the relation between wind speed and potentiometer voltage. This was derived by increasing the wind speed by increments of 5 mph and measuring the voltage at the output of the potentiometer.

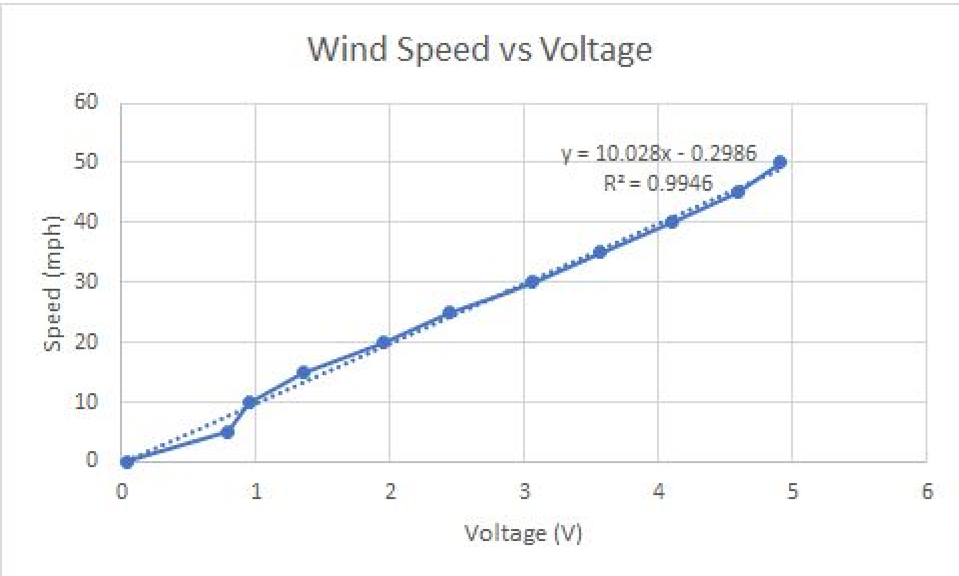


Fig. 58: Wind Speed vs Voltage curve

Wi-Fi Submodule (ESP32-DevKit V4)	
Characteristic	**Value**
Operating Voltage	3.3V
Operating Current	50nA
Power Consumption	0.165µW

Table 29: Wi-Fi submodule Power Consumption

Wind Speed	Output Frequency
m/s	0.0980 x Hz
knots	0.1904 x Hz
mph	0.2192 x Hz
km/h	0.3528 x Hz

Table 30: Young Wind Monitor Calibration Formulas

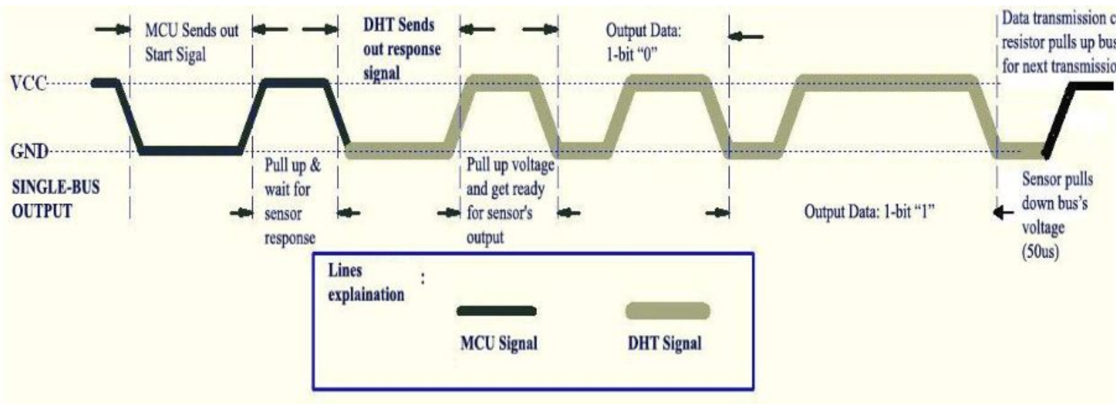


Fig. 59: DHT22 Temperature and Humidity Sensor Timing Diagram

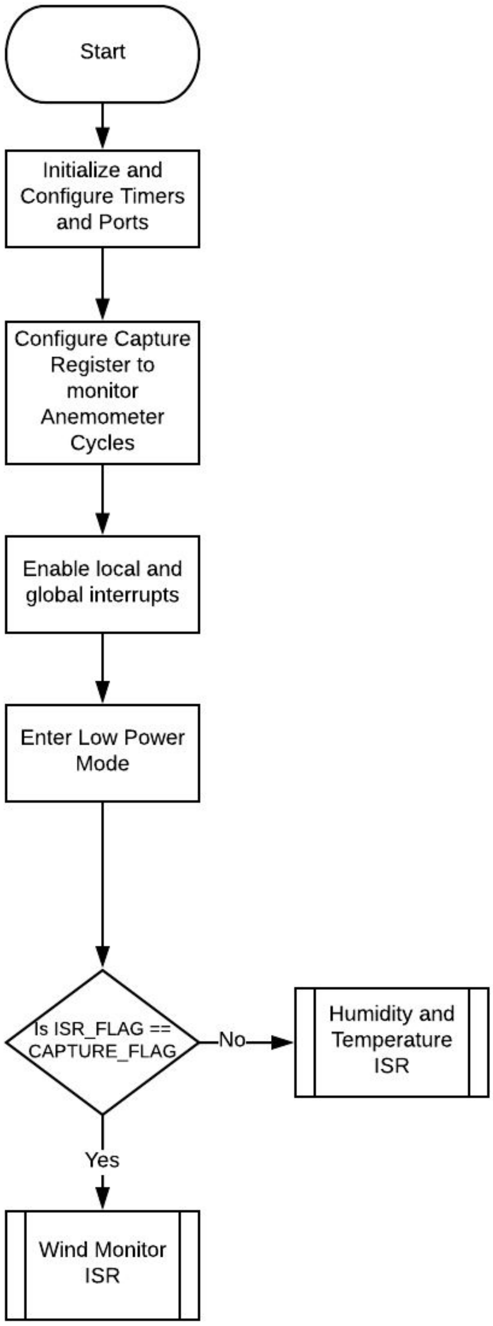


Fig. 60. WindTel Dynamic Measurement Module Wind Monitor Interrupt Service Routine Flowchart

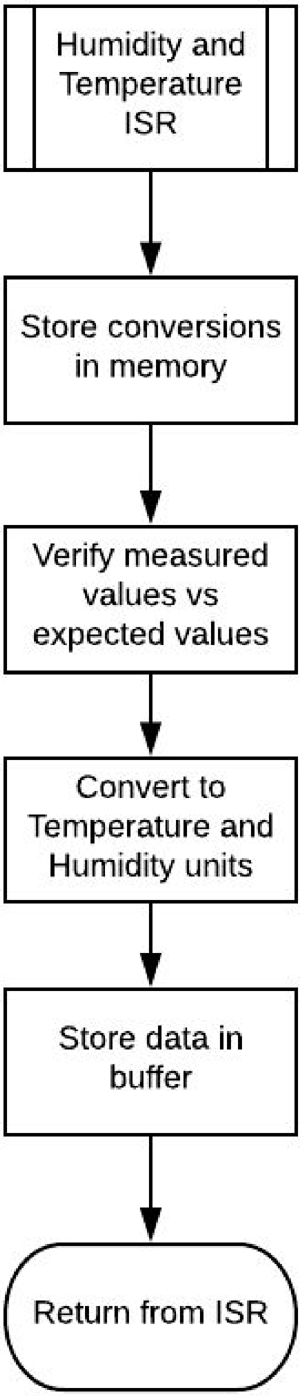


Fig. 61. WindTel Dynamic Measurement Module Humidity and Temperature Interrupt Service Routine Flowchart

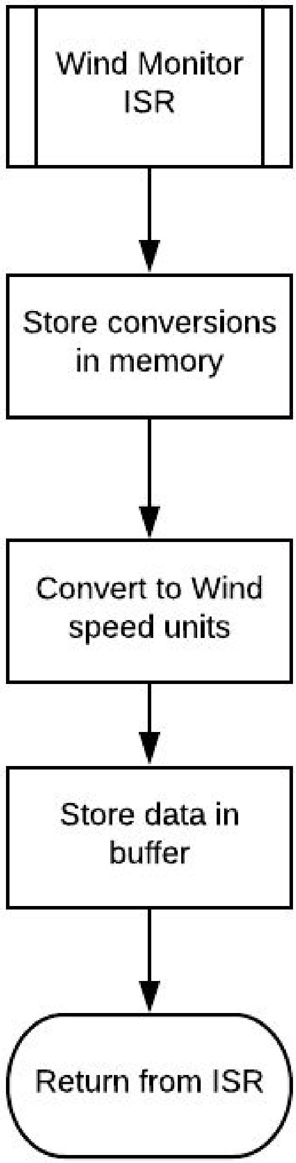


Fig. 62: WindTel Dynamic Measurement Module Wind Monitor Interrupt Service Routine Flowchart

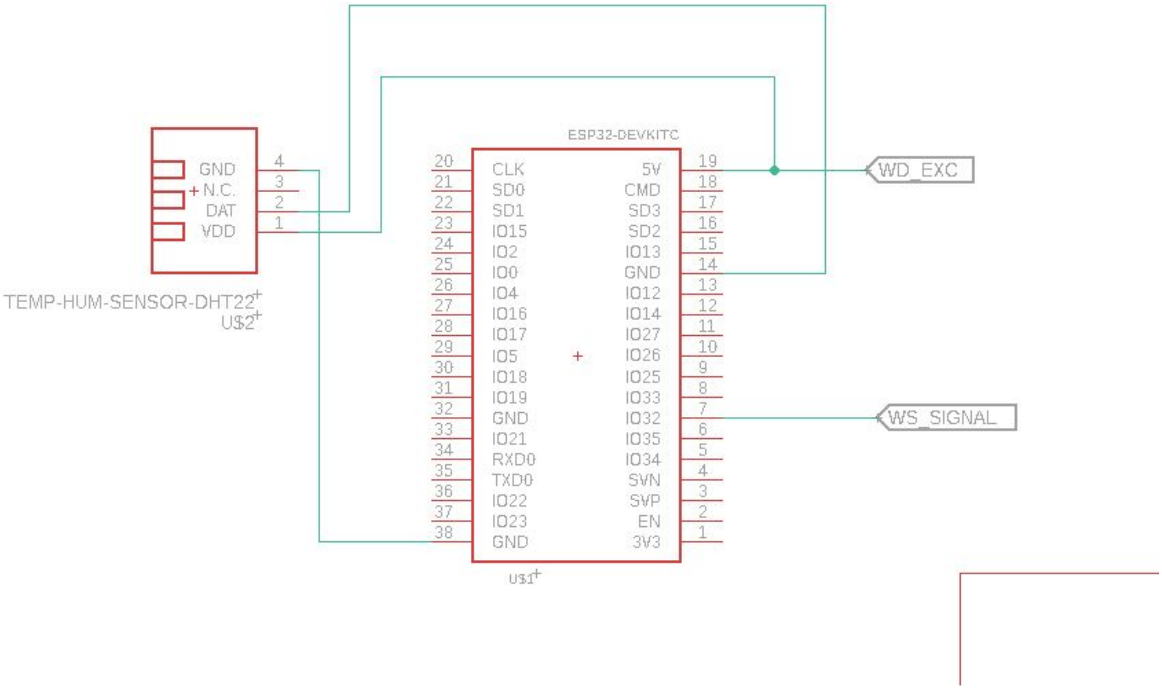


Fig. 63: Dynamic Measurements Module Schematic

2. Software Documentation <- *(Kahlil, Misael)*
    a. Components and modules

- Android tablet application <- *(Kahlil)*

The Android tablet application will serve as a way to operate the wind tunnel remotely. This will give the user the flexibility to move around with the tablet while connected to a wi-fi connection to operate the system.

- Windows desktop application <- *(Misael)*

The WindTel application will be deployed on desktop through the Electron framework. This is a framework that allows for the development of desktop applications using web technologies.

- Statistics and Data Visualization Tool <- *(Kahlil)*

The statistics and data visualization tool is a software that will reside inside the application of both the Android tablet and the desktop application. This tool will take the results of the data acquisition and display via graphs and charts as the user desires. The user will be able to select the any parameter of aerodynamic and environmental measurements and perform different statistical analysis from the variety of options that can be selected on the software.

- Backend Application

A) Setting up the database server

A database server serves as means to store, in this case, the information of the users and the experiment results they which to store. It has been implemented utilizing Flask, which is a Python Framework for building web apps and REST APIs. Ubuntu linux operating system was used to setup the backend application environment.

There were several steps needed to follow in order to have the necessary tools for the installation of a database server as well as the application that will contain the routes and the definition of the requests that will be done to the database server.

1) Install postgreSQL and pgadmin3
   The first step is to install postgreSQL, which is an open source database, and pgadmin3 which serves to administer. This can be done with the following commands on an ubuntu linux terminal:
   - sudo apt-get update
   - sudo apt-get install postgresql postgresql-contrib
   - sudo apt-get install pgadmin3

   The first command will look for newer packages and update the current ones. The second command will install the Postgres package along with a -contrib package that adds some additional utilities and functionality. The third command installs pgadmin3 which is an open source administration and development platform for PostgreSQL.
2) Create a role for the application
   The second step would be to create a role for the postgres user using the following commands:
   - sudo -i -u postgres
   - createuser appusr -e -P

   The first command will put you at the root of the account of postgres, and the second will create a user called ‘appusr’. It will prompt to set a password for ‘appusr’ afterwards.
3) Create the database
   Now we create the database as follows:
   - createdb appdb

   Creates the database named ‘appdb’.
4) Grant privileges of the database to user
   We now make the user ‘appusr’ the owner of the ‘appdb’ database.
   - psql appdb

   The psql command connects to the ‘appdb’ database through the terminal, and you will see something like this:
   appdb=#
5) Configuration of Postgres for password connections
   The configuration will make the remote connection to the database possible. You will need to go to the following path:

/etc/postgresql/10/main and open the postgresql.conf file. Here you will edit the following:

listen_addresses = ‘localhost’

to

listen_addresses = ‘*’

You may then exit from the postgres account with the following command:

- sudo etc/init.d/postgres restart

This will restart and update the changes to postgres.

6) Login to database

You may then login with the following command:

- psql ‘appdb’ -U ‘appusr’ -W

The command will login to the database ‘appdb’ as user ‘appusr’. The flag -U
specifies user and -W ask for password.

After these steps are done, you may then be able to run queries for the requests to the database. As an example, if you want to add a table named researcher such as the one done for the WindTel system, you can perform the following query:

- appdb=> create table researcher
- appdb-> (rid serial primary key, rname varchar(20), rlast_name varchar(20),
- appdb(> remail varchar(30), rvocation varchar(20), rdepartment varchar(20),
- appdb(> rinstitution varchar(20), rcity varchar(20), rpassword varchar(20),
- appdb(> rpermissions varchar(15));

After these are inserted you will see the following response: CREATE TABLE, denoting that the table has been created successfully.

B) Creating the backend application

The backend application was written in Python utilizing the Flask framework. The backend application has been divided in three modules which are:

- app.py - main of the backend application. It has the routes through which the requests will flow and the classification: POST, PUT, GET, DELETE.
- handler - the handler is divided into 4: researcher.py, experiment.py, measurement.py, and pressurepoint.py. Each handles the methods that correspond to the tables of their naming respectively as well as the dictionaries for the JSON outputs.
- dao (data access object) - there are 4 daos, one for each handler. The daos contain the queries that will be done to the database server in order perform the corresponding requests.

C) Access to a private server

An Amazon Web Services (AWS) was used to provide a remote database server for the WindTel system. This was done by purchasing the free 1 year service that Amazon can provide to its users and the credentials of one of the members was enlisted for the purchase. After the year passes, another remote server will be needed in order to provide the system a remote database server. A local database server will be provided in case no other subscription is bought from AWS or any other remote server provider.

The steps to setup the AWS can be looked up on the Amazon page and the steps for the database server and postgres installation are the same as for local installation.

b. User Interface <- *(Kahlil)*

The WindTel system will be composed of three user interfaces, which are:

1) The graphic LCD contained along the hardware system
2) The Android tablet software application
3) The Windows desktop software application

The hardware user interface can be operated without a wifi internet connection, but will be unable to store information to the database since there is no connection to the internet. The Android tablet application and the desktop application on the other hand, can't be used without connectivity with the internet since it is required for the user to provide their credentials and login through the application. Using the application will give the user the option to not just execute experiments and store the results, but to be able to visualize data in statistical terms. A mock up of what our vision of how the software will look like as required by the client and brainstorms we have conducted, are presented in the following figures.

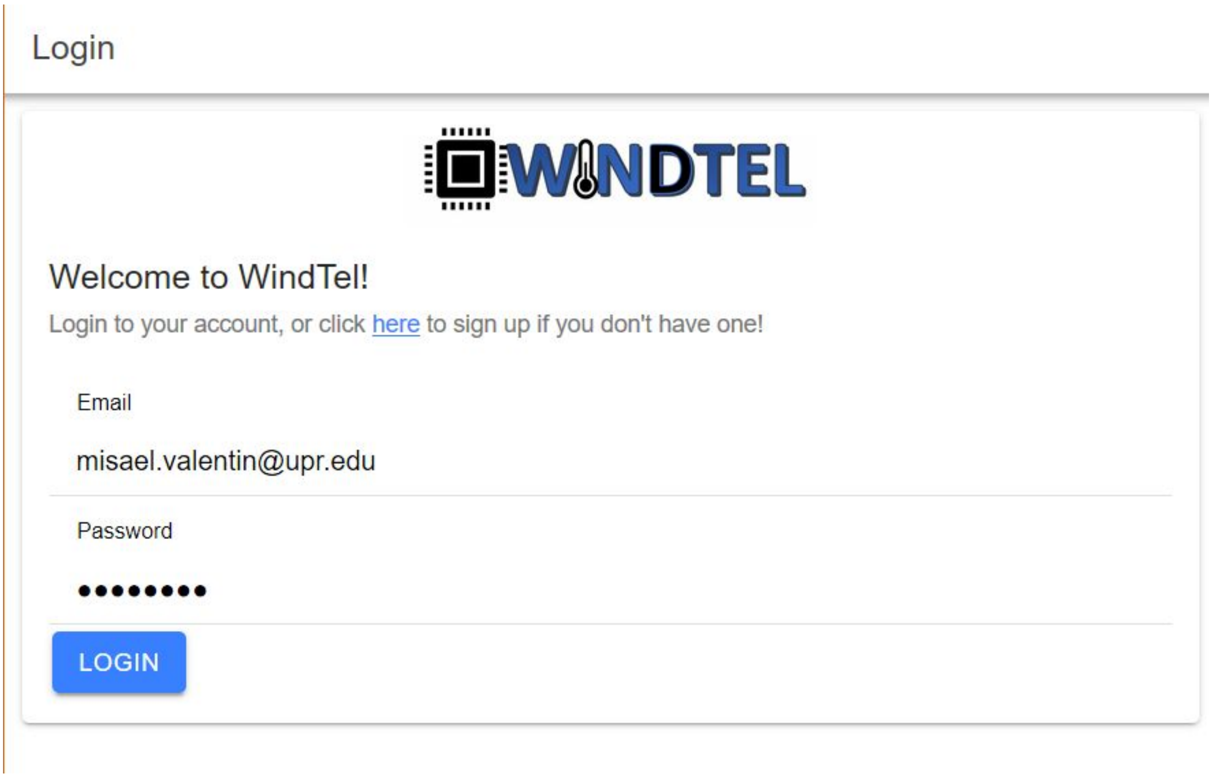


Fig. 64. Home View

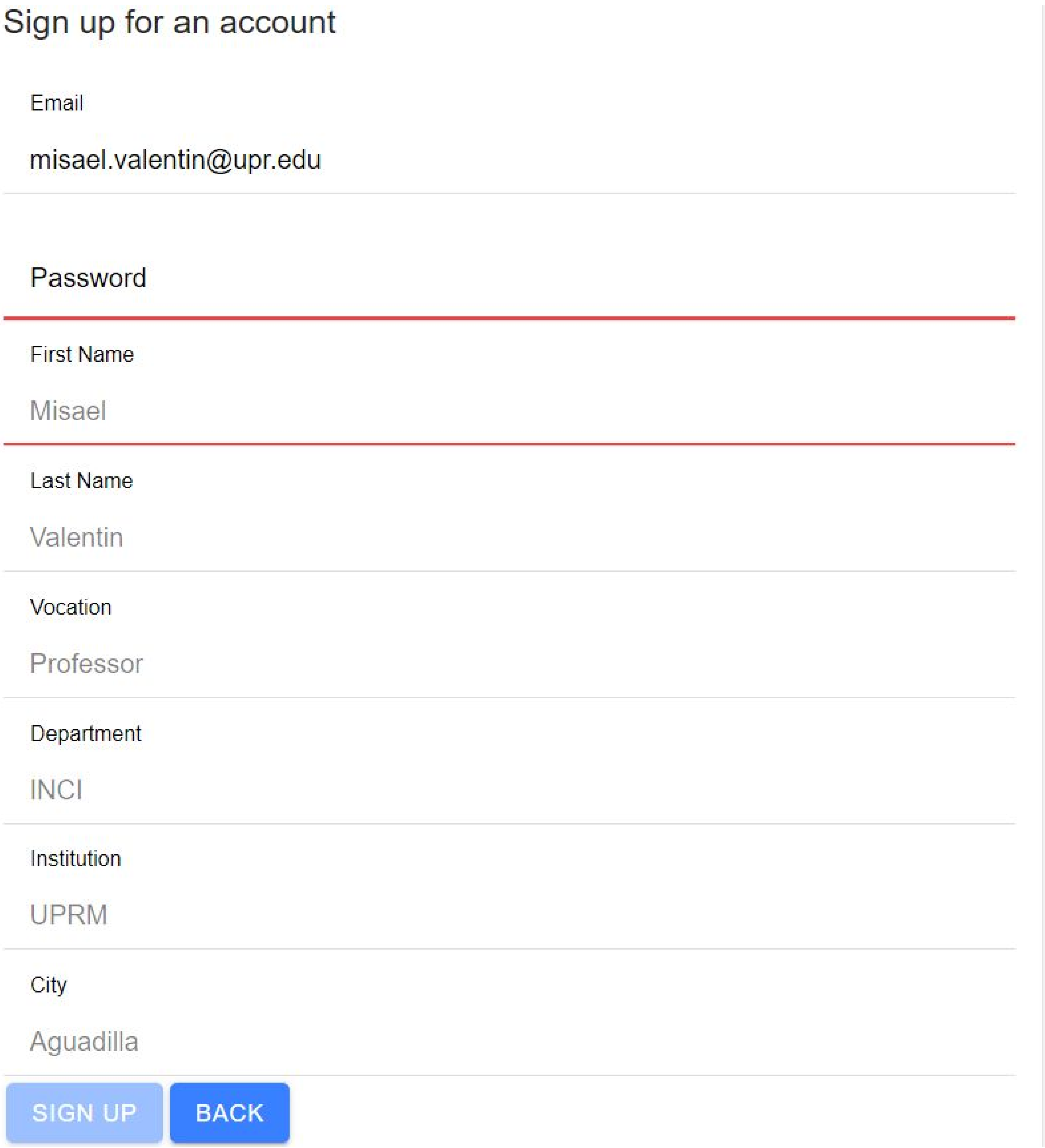


Fig. 47. Sign Up View

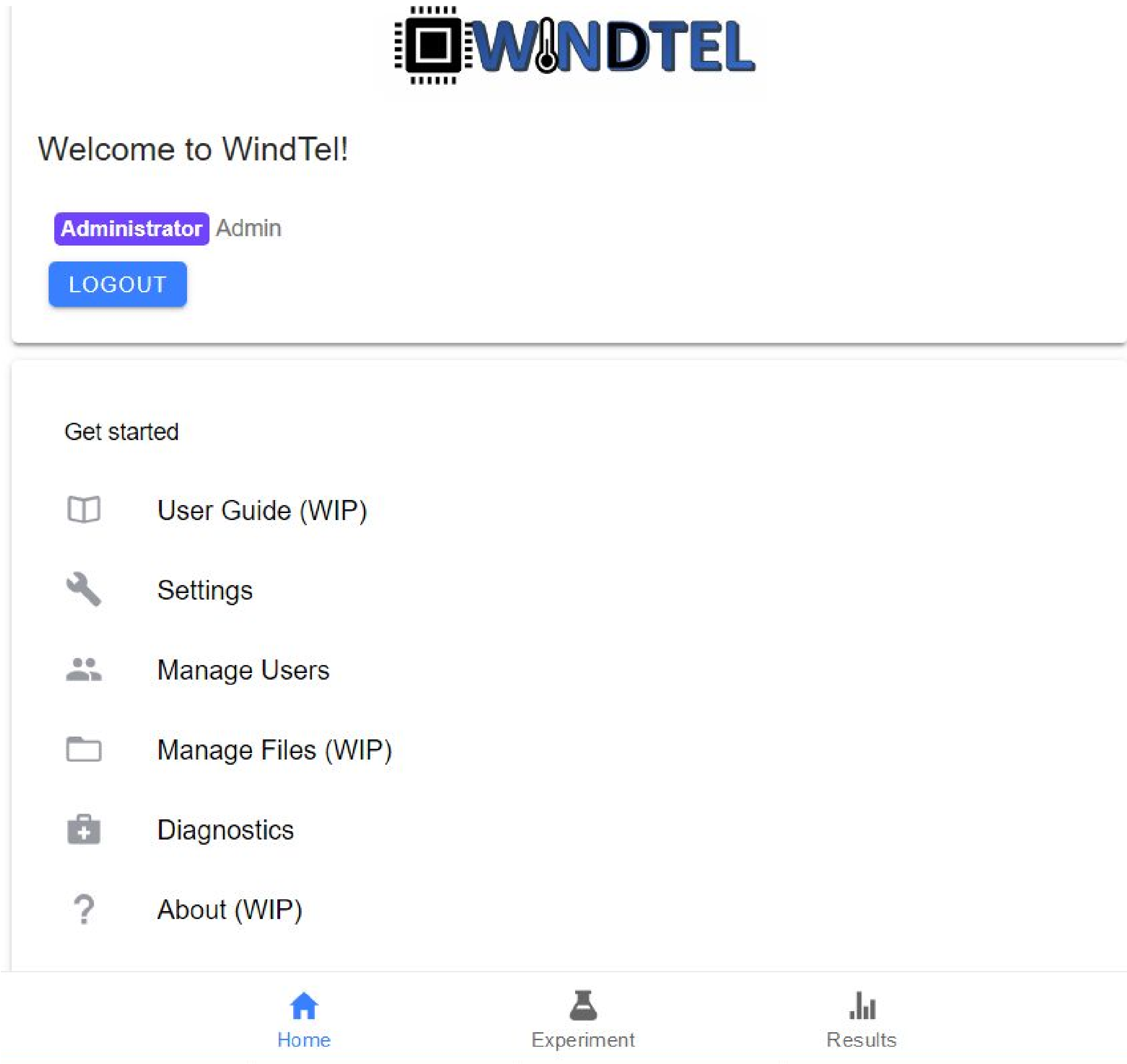


Fig. 65. Main Menu View

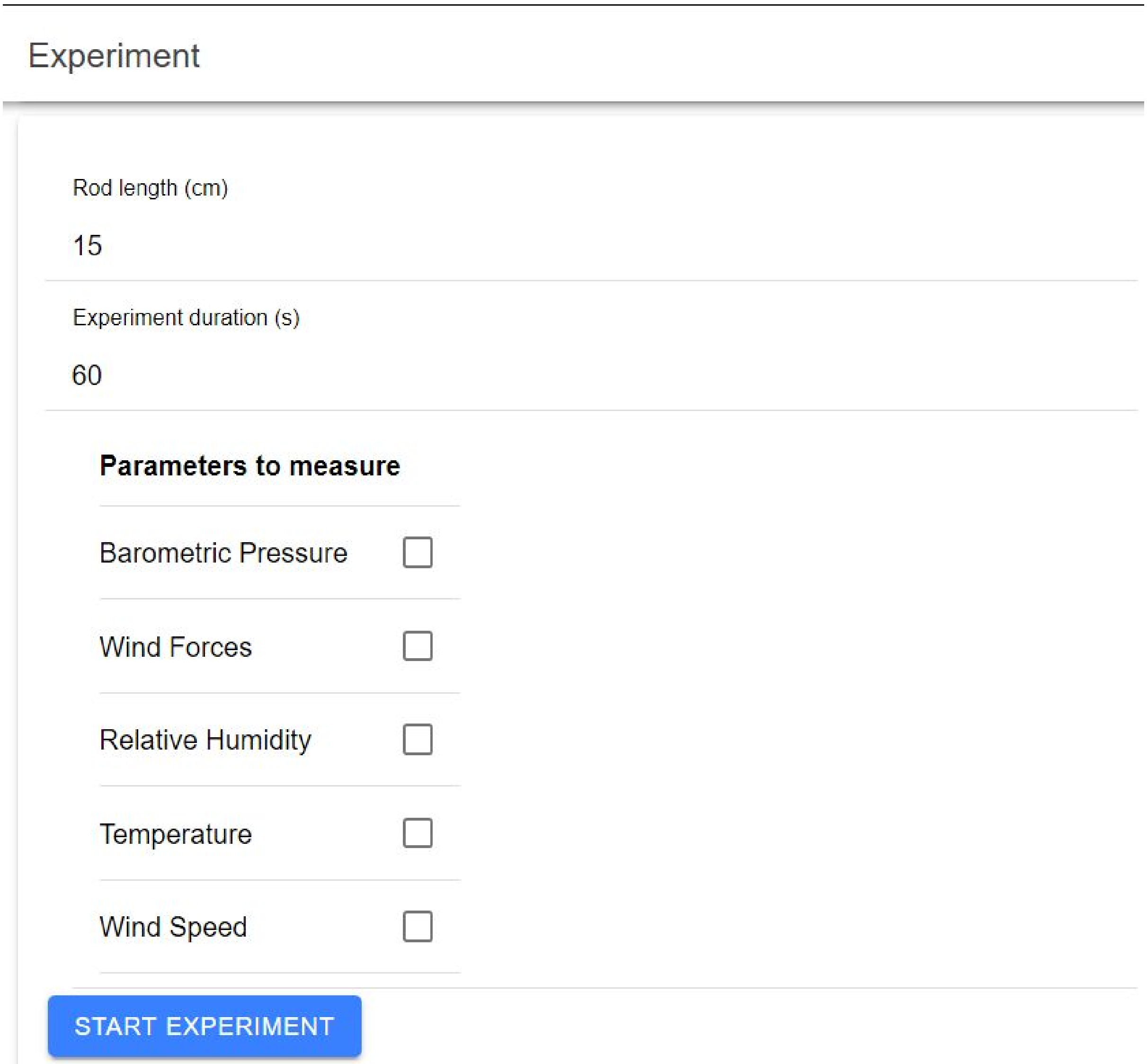


Fig. 66. Design Experiment View

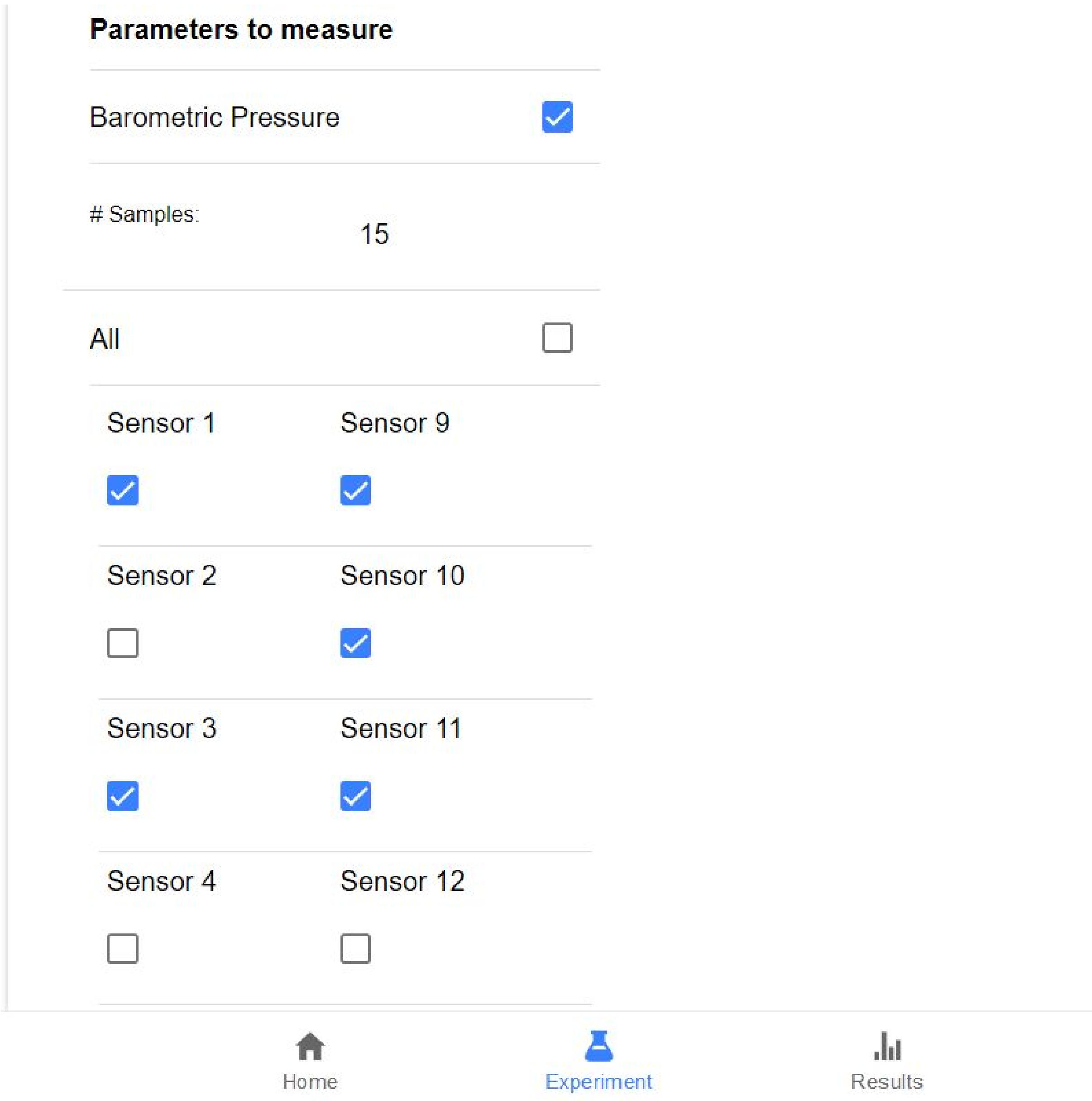


Fig. 67. Design Experiment - Pressure View

Experiment Results

Sample #	Wind speed	Temperature	Humidity
1	21.32 mph	25.02 C	85.09%
2	20.32 mph	25.02 C	85.09%
3	19.41 mph	25.22 C	85.09%
4	22.22 mph	26.03 C	85.09%
5	21.00 mph	26.03 C	85.10%
6	25.50 mph	26.03 C	85.10%
7	29.04 mph	26.03 C	88.99%
8	37.32 mph	26.03 C	88.99%
9		26.03 C	88.99%
10			88.99%
11			88.99%
			88.99%

Fig. 68. Experiment Results View

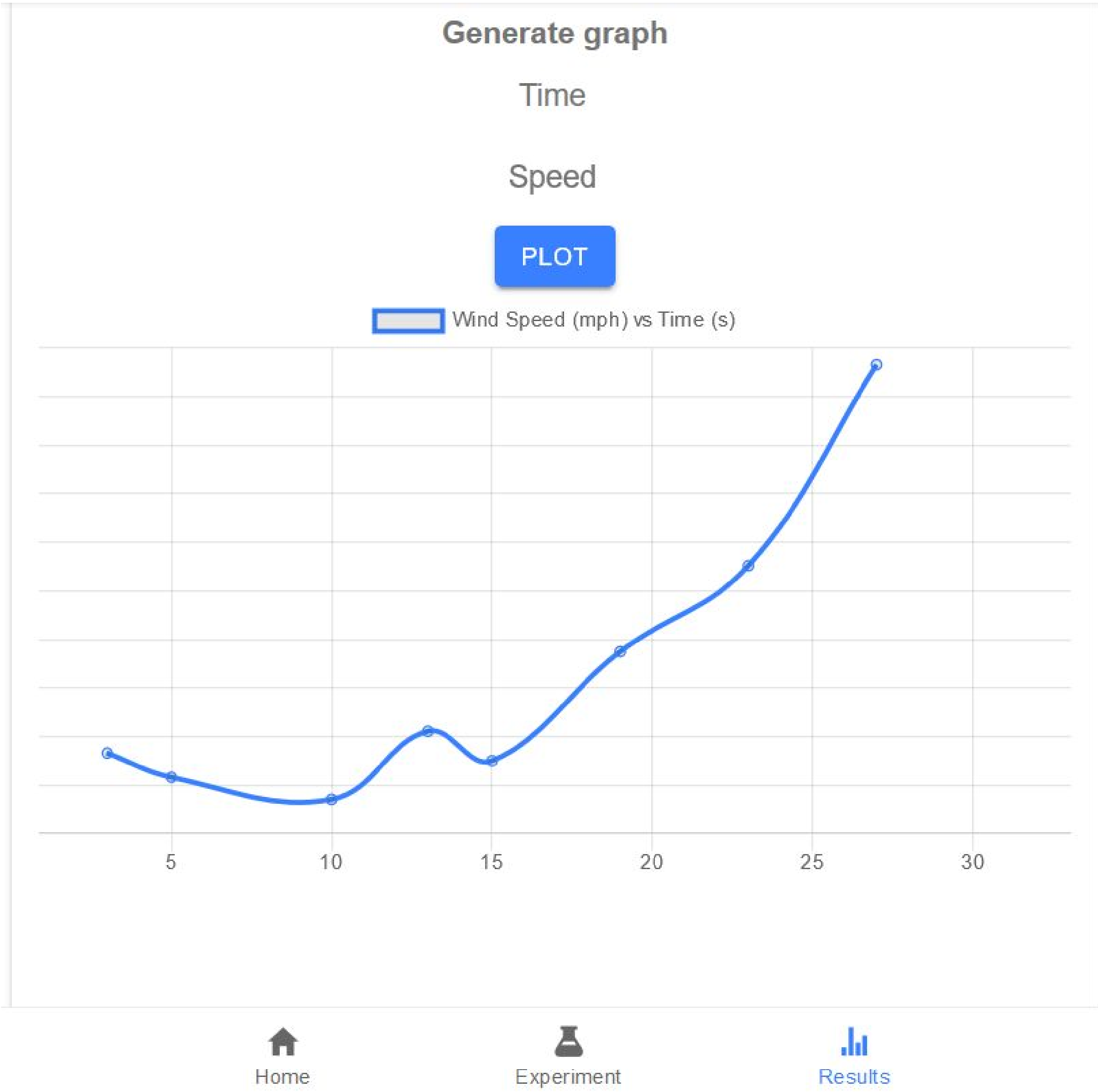


Fig. 69: Data Visualization View

c. Use Case Diagrams <- *(Kahlil)*

We have identified three different types of users that the WindTel system may have: researcher, technician, and administrator. Out of these three, there must be an administrator in order to give users access to the WindTel software application. Although the administrator won't be necessary when using the hardware user interface, it will be necessary to manage the system in terms of storage capacity, removing users, and other important operations. The use case of each type of user are shown in the next sections.

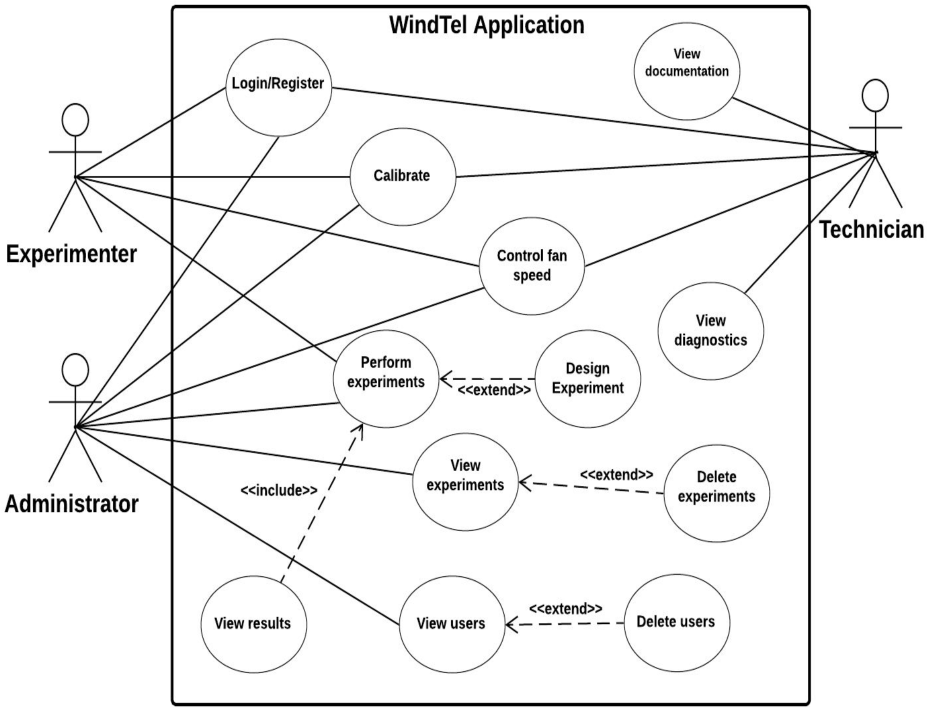


Fig. 70. Use-case diagram

d. Sequence Diagrams <- *(Misael)*

Fig. 71 illustrates a typical message sequence for a Researcher use case. This sequence starts with the Researcher turning the WindTel application on, attempting to log in, which will trigger the WindTel application to send an authentication request to the Database. The Database's reply will determine whether the user credentials are valid. If they are valid, then the Researcher may choose to start an experiment. Upon choosing to start an experiment, the WindTel application communicates with the Master Module through the Wi-Fi Sub-Module. The Master module then relays to the data acquisition modules to begin the data acquisition process. This data acquisition process is illustrated with the Sequence Diagram in Fig. 72. After each data acquisition module has indicated that they are finished acquiring data, the Master Module communicates to the Wi-Fi Submodule to begin the process of storing the acquired results in the Database. Finally, the Master Module communicates to the WindTel application through the Wi-Fi Submodule that the experiment and data store have finished, and the WindTel application in turn indicates the same to the Researcher through the graphical user interface (GUI).

Fig. 73 illustrates the communication between the Master Module and the Data Acquisition Modules. This process starts by the Master Module communicating to each module, in a sequential manner, to initiate their data acquisition. Each module will begin to acquire data concurrently; the relative timing will depend on each module's sampling rate. The Master Module will then ask for each data acquisition module's status. When all modules have indicated that they have finished acquiring data, the Master Module communicates to the Wi-Fi module to begin the data storage process.

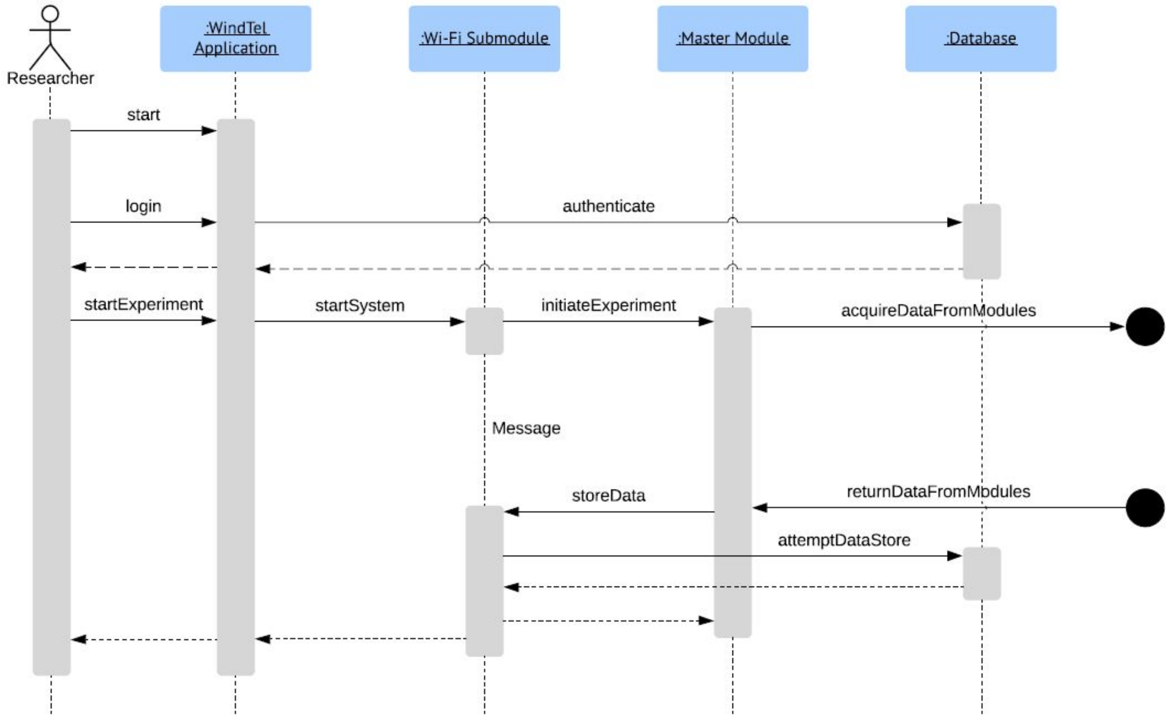


Fig. 71. System Sequence Diagram

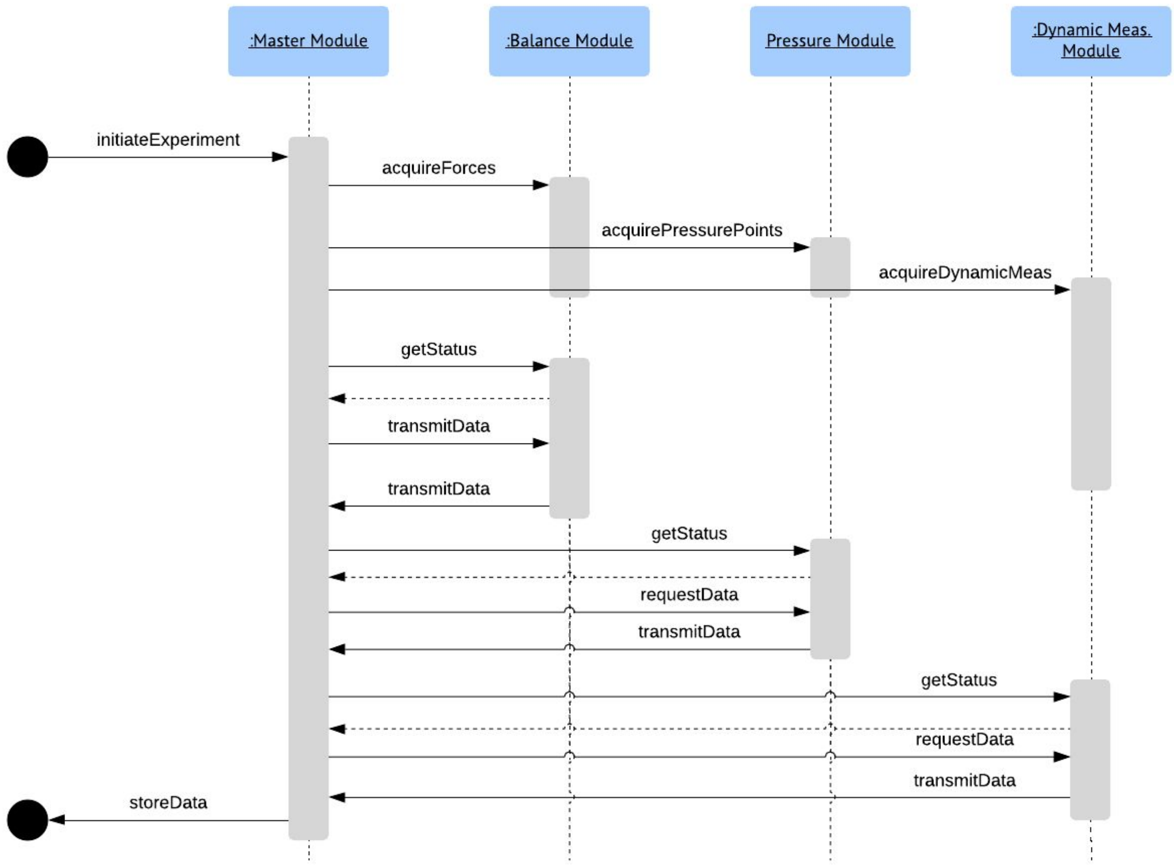


Fig. 72. Data Acquisition Sequence Diagram

e. Entity Relational Diagram *<- (Kahlil)*

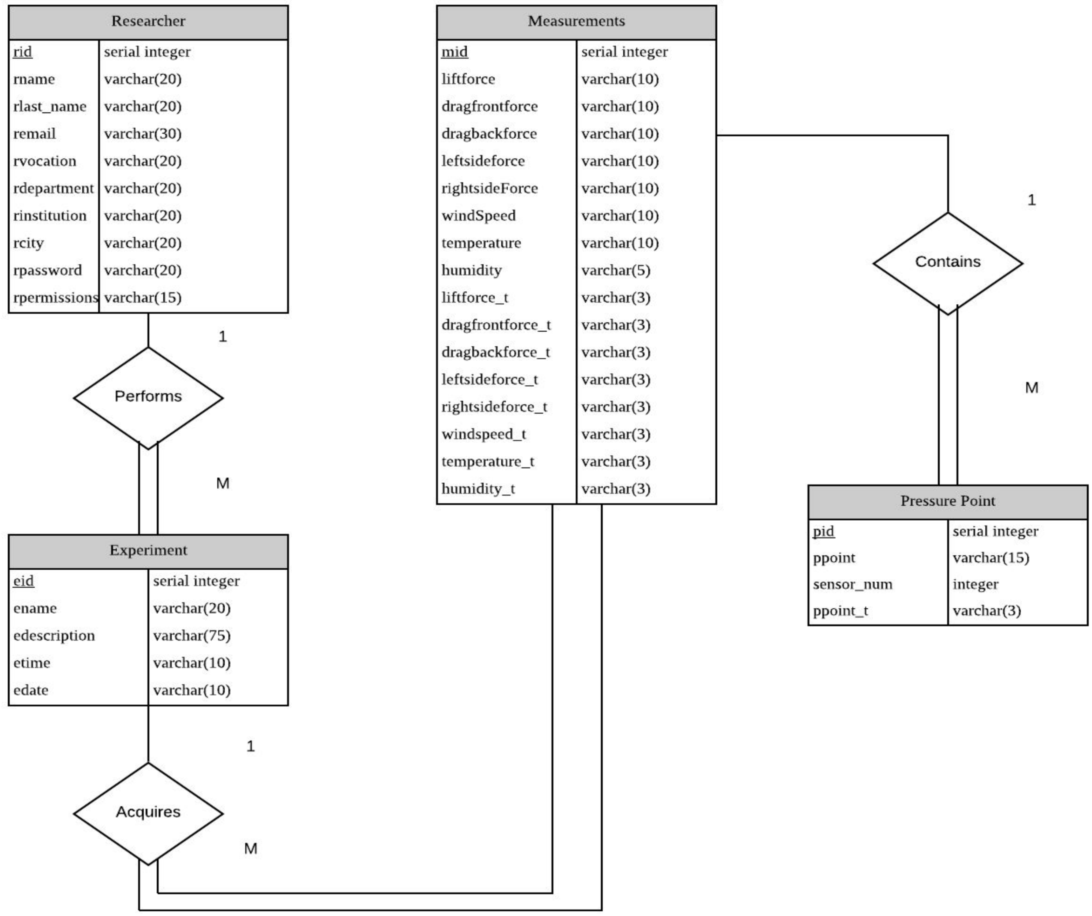


Fig. 73 Database Server Entity Relational Diagram

f. Backend Class Diagram <- *(Kahlil)*

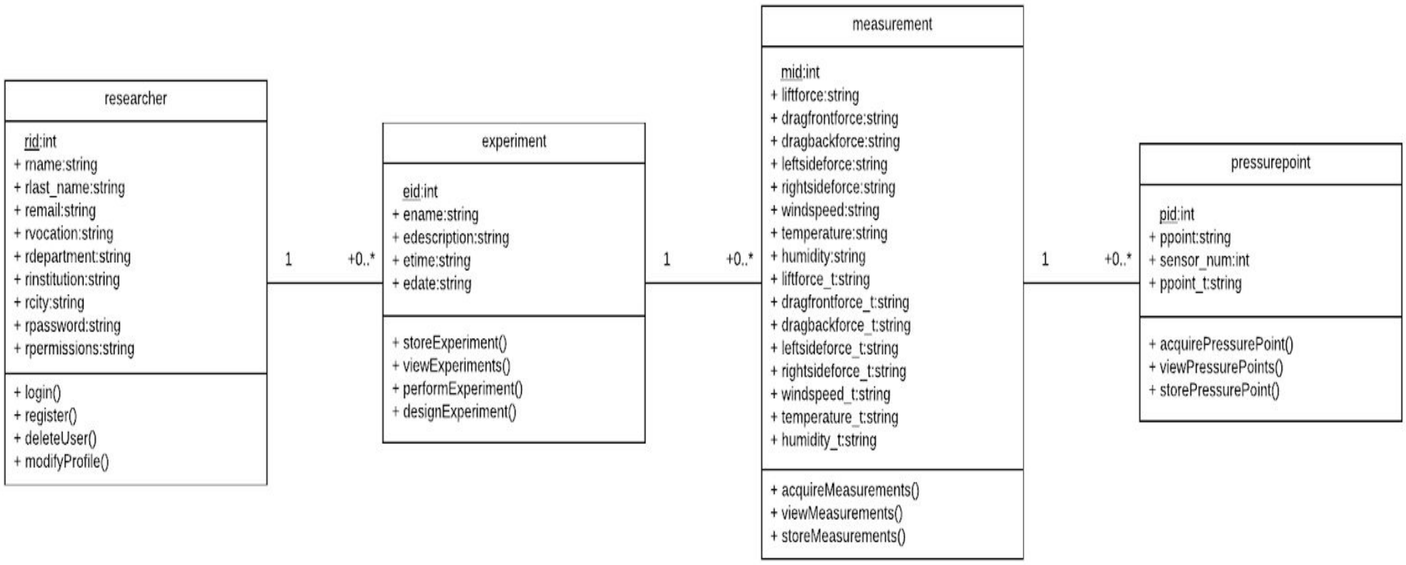


Fig. 74 Database Class Diagram

f. Frontend Class Diagram <-*(Misael)*

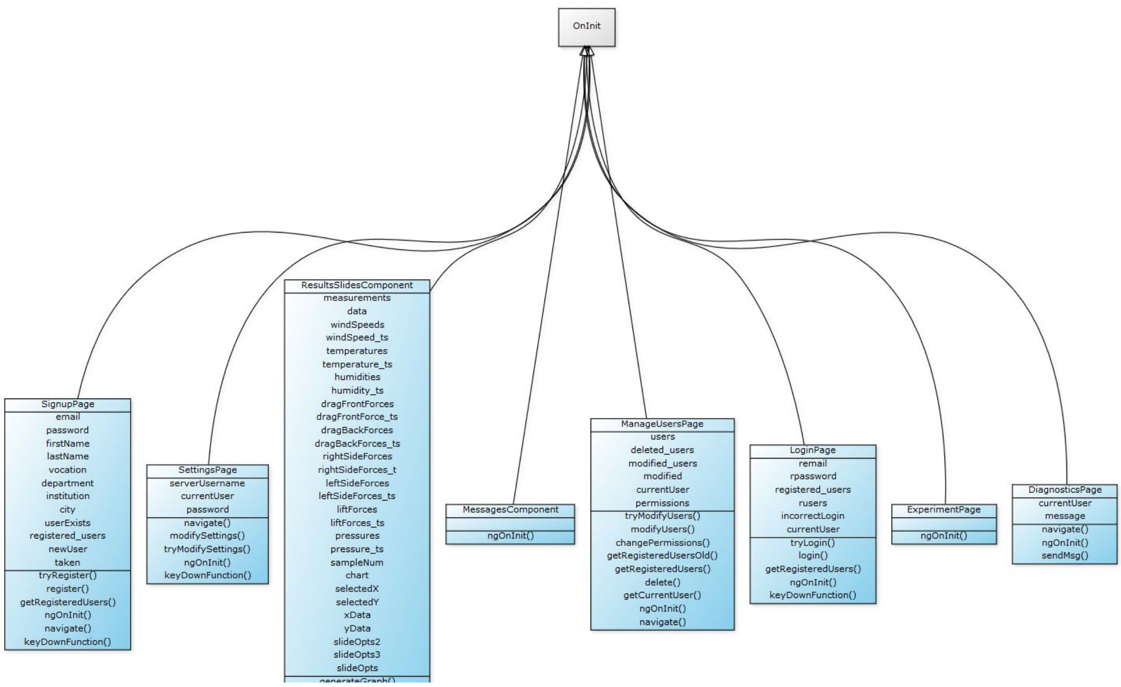


Fig. 75. WindTel Application class diagram

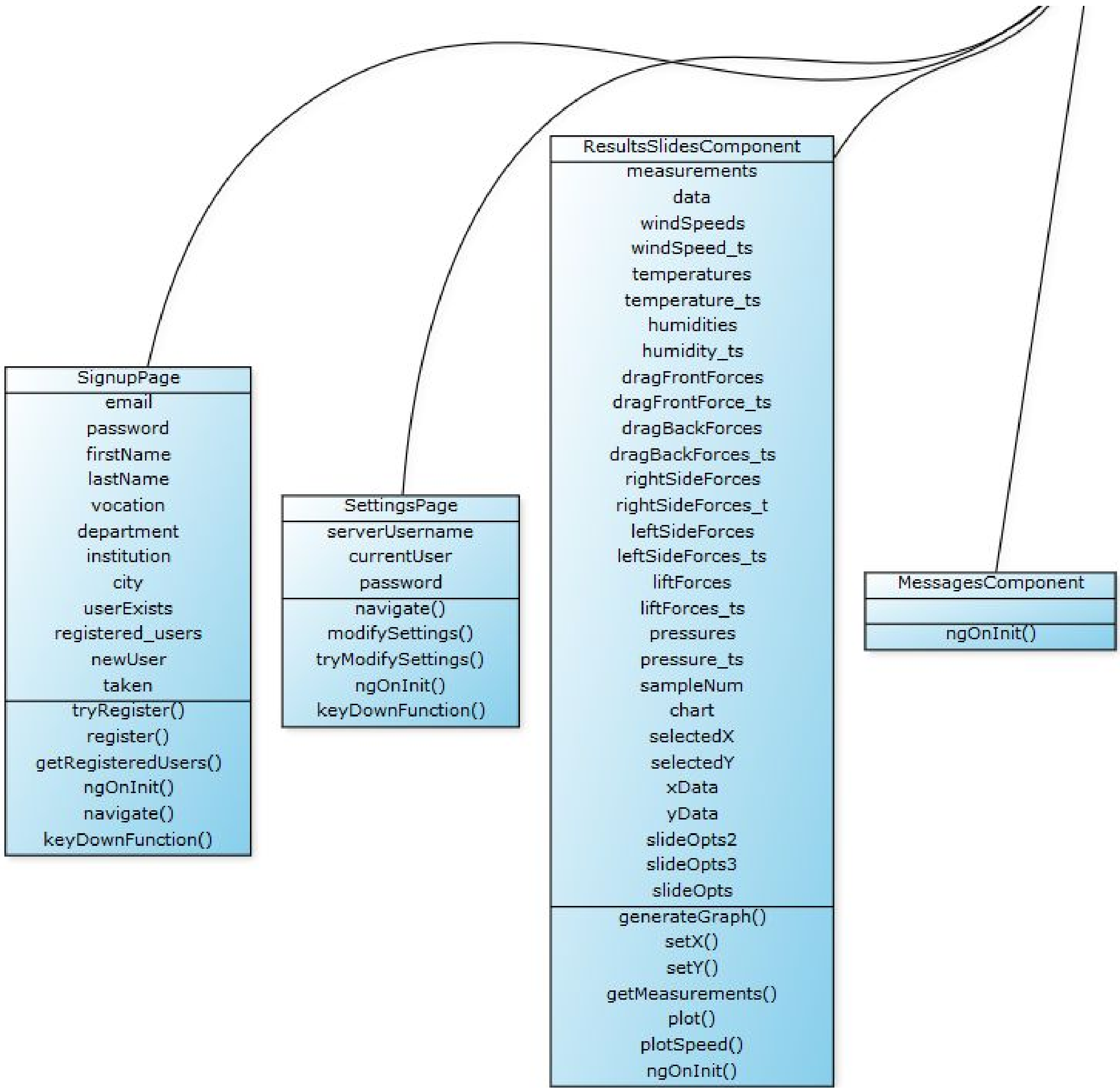


Fig 76. WindTel Application class diagram detail

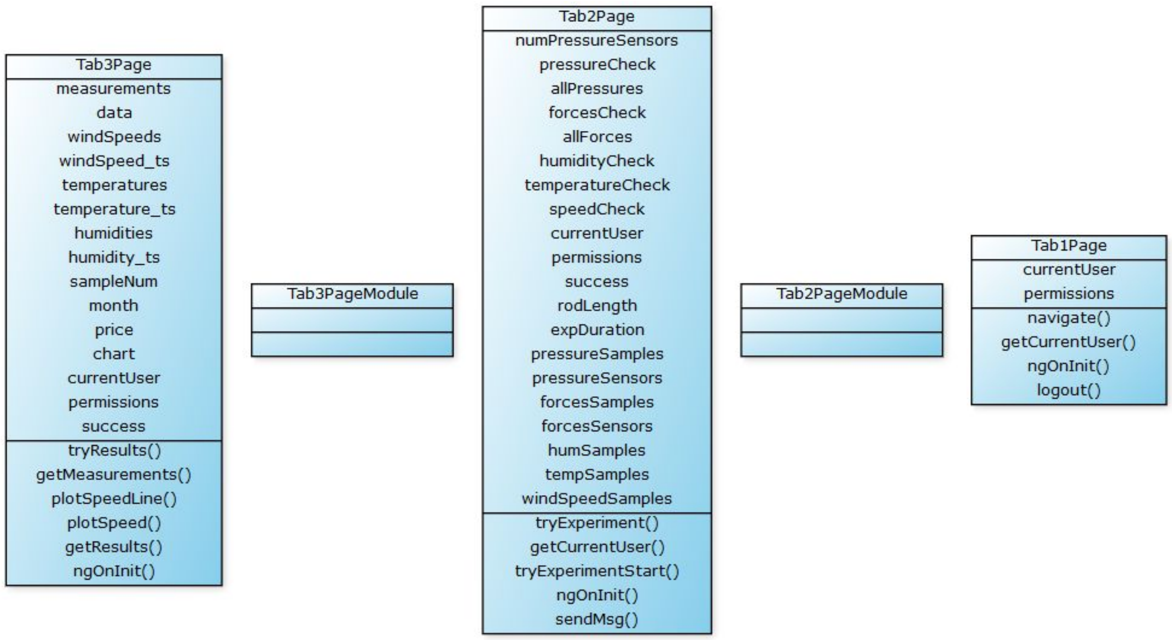


Fig. 77. WindTel Application class diagram detail

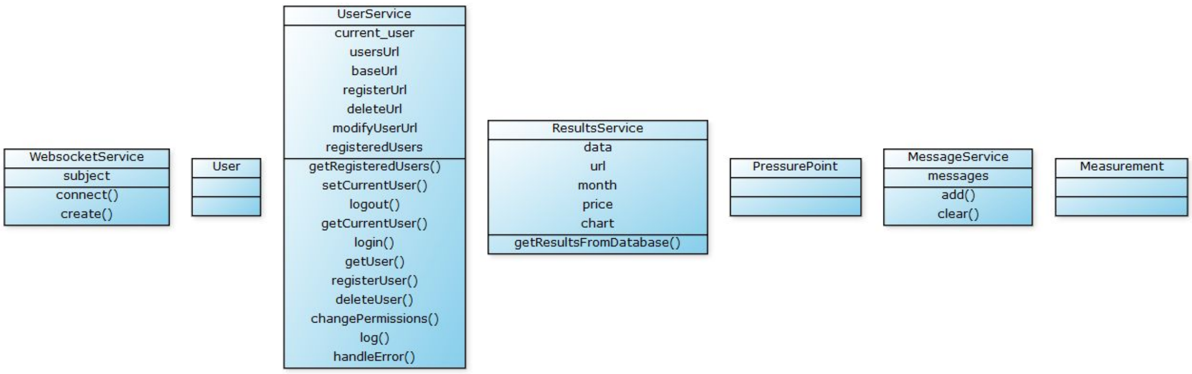


Fig. 78: WindTel Application class diagram detail

Fig. 79: WindTel Application class diagram detail

G. Testing Plan <- *(All)*

1. Hardware Testing Plan <- *(Nelson)*

The WindTel system hardware consists of a master microcontroller exchanging data with three slave microcontrollers. The slave microcontrollers acquire aerodynamic and environmental measurements from sensors while the master microcontroller communicates with a variable speed driver to control the blower fan of the wind tunnel. The hardware modules characteristics that will be tested are the following:

a. Load cells output voltage variation in function of the weight forces

Load cells are transducers that convert weight forces to analog voltages in the milivolts range. A linear relation between the weight forces and the load cells output voltage was obtained according to the calibration made during the design phase of the project. In the implementation phase, the load cells output voltage will be tested with the input resistance of the signal conditioning circuit as the load (Fig 78). A voltmeter will be used to measure the voltage at the load cells output and will be verified using the linear relation between the load cells forces and output voltages. It is expected that the load cells output voltage follows a similar linear relation in function of the force since the input impedance of the amplifier stage of the signal conditioning circuit is high.

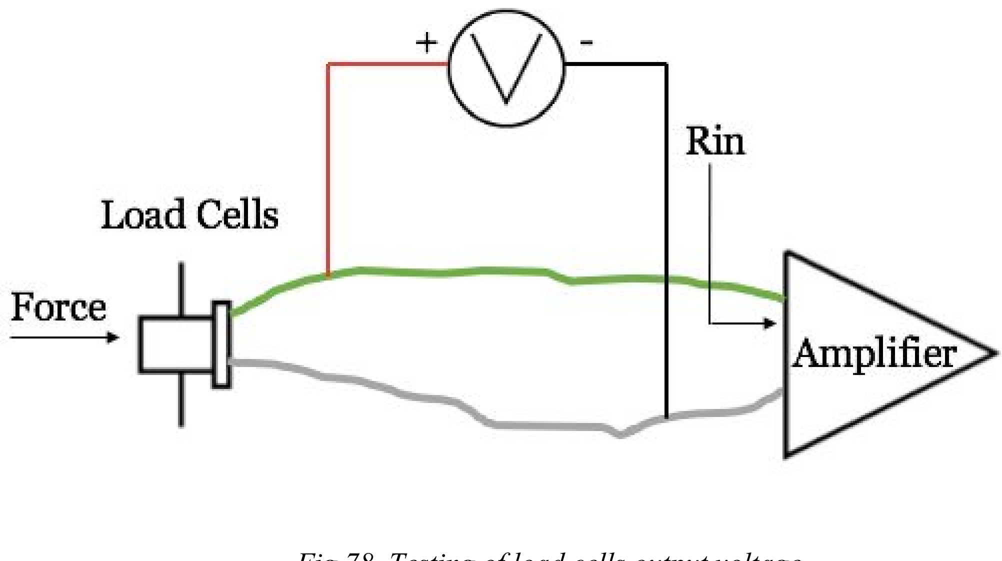


Fig 78. Testing of load cells output voltage

b. Balance Module signal conditioning circuit output voltage variation with ADC input resistance

During the design phase, a parametric analysis varying the signal conditioning circuit load resistance from 0.135kΩ to 1kΩ was performed to determine the effect of the ADC input resistance in the circuit behavior. A similar approach will be used to test the circuit implementation in breadboard and printed circuit board. The difference is that instead of connecting a resistance to the load, the ADC will be connected as the load. According to the MCU ADC, the input resistance varies from 0.135kΩ to 1kΩ for a voltage supply from 1.8V to 3.7V. A voltmeter will be used to measure the signal conditioning output voltage (Fig. 79) and compare it to the family of curves presented in Appendix F. The signal conditioning circuit is expected to maintain similar operating voltages and currents so that the output voltage is not affected significantly.

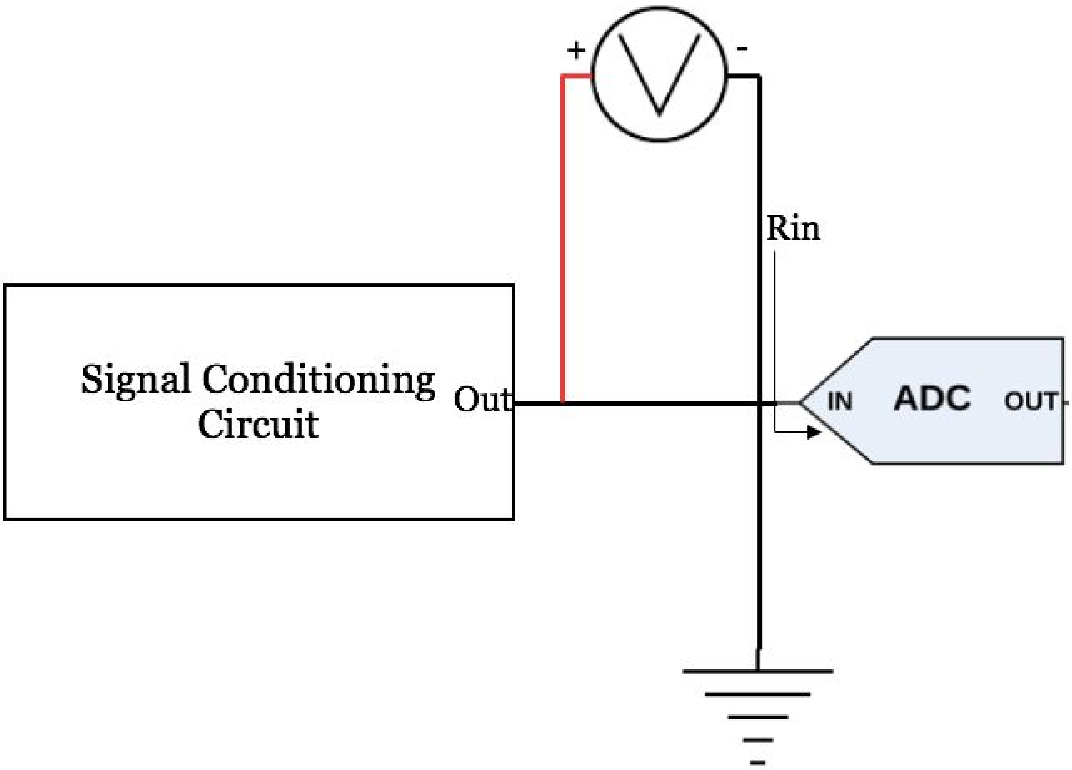


Fig 79. Testing of Balance Module signal conditioning circuit output voltage

c. ADC digital code generation according to the weight forces

The complete Balance Module will be tested by applying several known weight forces to the load cells and obtaining the digital code of the ADC in a Liquid Crystal Display (LCD). The weights will be measured using a scale for weighing and the output digital code will be converted to its corresponding analog value to obtain the forces using the linear relation between load cells forces and output voltage.

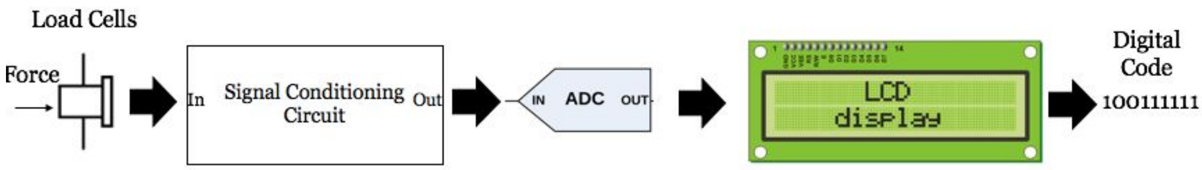


Fig 80. Testing of Balance Module

d. Data transfer speed in SPI communication line between the Pressure

Module and the pressure sensor matrix

The speed in which commands from the Pressure Module are sent and received by the pressure sensors is essential in the WindTel project. If for some reason a delay occurs when the Pressure Module communicates with the pressure sensor matrix, the pressure sensors may read incorrect measurements and the total wind pressure distribution may not be consistent with the expected results. The data transfer speed in the SPI line will be tested by using an oscilloscope (Fig. 81). During the test, the frequency of the signal in the serial line will be measured and compared with the expected measurement acquisition frequency of the pressure sensors (87.0 Hz).

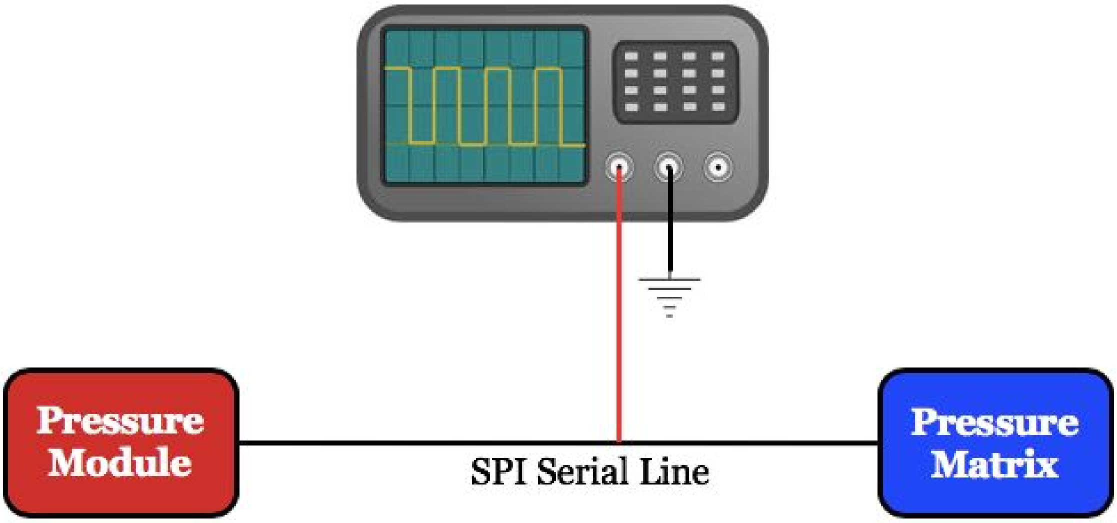


Fig. 81. Testing of data transfer speed in SPI line between Pressure Module and pressure sensor matrix

e. Pressure sensor selection with chip select signal sent by the Pressure Module

The Pressure Module MCU assigns a slave address to each pressure sensor of the pressure matrix to facilitate the selection of one sensor using the chip select pin of the sensors. However, the tester must ensure that the data acquired in the Pressure Module is from the selected sensor and not from the other unselected sensors. If the reading is not from the selected sensor, then the pressure reading will be incorrect because pressure varies with the wind density in the cross-sectional area and the acquired data will not be from the point of interest for the wind tunnel user. The test will be conducted by using an oscilloscope in the SDO pin of the selected pressure sensor (Fig. 82). The test results should be a square waveform in which the digital code represents the pressure taken in a certain point of interest inside the wind tunnel.

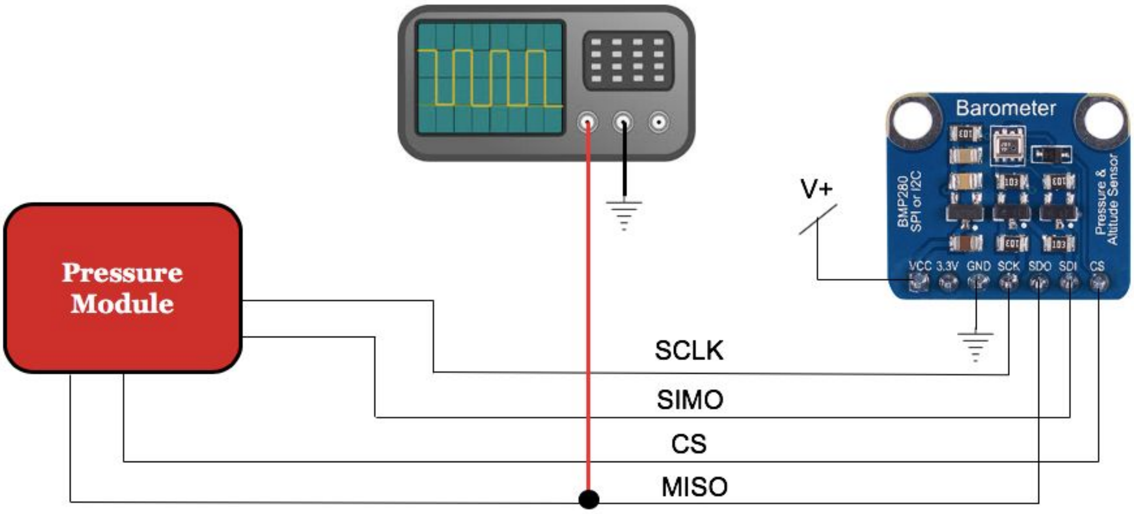


Fig. 82. Testing of pressure sensor selection with chip select signal from Pressure Module

f. Communication error in UART communication line between the Master Module and the other system's hardware modules

The communication between the Master Module and the rest of the system's hardware modules is the most important part of the WindTel system. Errors in the UART communication can lead to incorrect data measurements been displayed in the user interface. The UART line will be tested by identifying an error between the Master Module and each hardware module using the MSP432P401R watchdog timer. The test consists of setting fixed times for each module to send their data to the Master Module. If the data is not sent in the specified time, then the watchdog timer will reset the module with the error and start the acquisition of the module's data again.

g. Pressure sensors readings variation with height and temperature

The pressure sensors will be tested at different heights and temperature environments. Pressure data from the sensors will be acquired on the 7th floor of Stefani building to verify the pressure dependence on the height. For the pressure variation with temperature, the test will be conducted with the sensors inside the wind tunnel since the wind flow cools the wind tunnel test area environment. It is expected that the pressure decreases when the height increases and the temperature increases.

h. Wind speed generated by the blower fan comparison with anemometer wind speed readings

It is important to ensure that the wind speed inside the wind tunnel is the wind speed specified by the user in the user interface of the Master Module. The test will be conducted using the anemometer readings (Fig. 83). After the tester selects the wind speed to test, the anemometer will acquire the wind speed data and send it to the Dynamic Module. The Master

Module will acquire the measurement from the Dynamic Module using the UART protocol and it will be displayed in the user interface. The wind speed reading will be compared with the specified wind speed to determine discrepancy. If the wind speed reading is not consistent with the specified wind speed, then the Master Module's Control System submodule will be verified and refined.

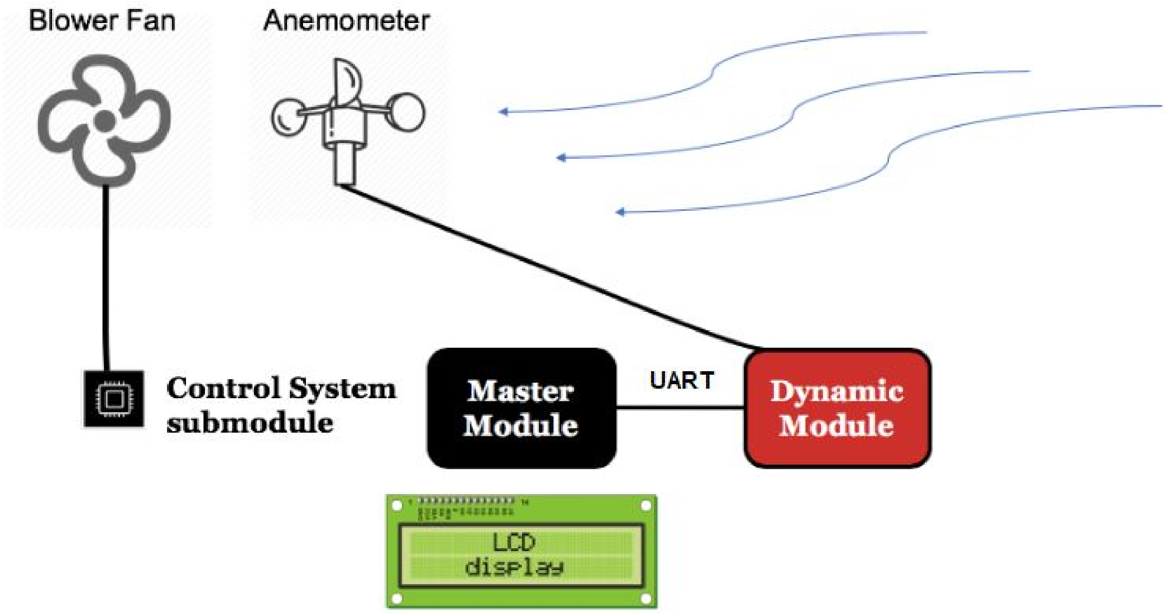


Fig. 83. Testing of the blower fan generated wind speed with the anemometer readings

i. Wind speed variation with Control System submodule reference voltage

The wind speed that the blower fan generates depends on the Control System submodule. This submodule contains a digital potentiometer in which the voltage drop is varied using pulses generated by the Master Module MCU. The wind speed variation respect to the submodule reference voltage will be conducted using a voltmeter to measure the voltage drop in the potentiometer and the wind speed specified by the tester (Fig. 84). The test results will be compared with the expected results obtained from the relation between the wind speed and the voltage drop in the potentiometer obtained by calibration of the Control System submodule.

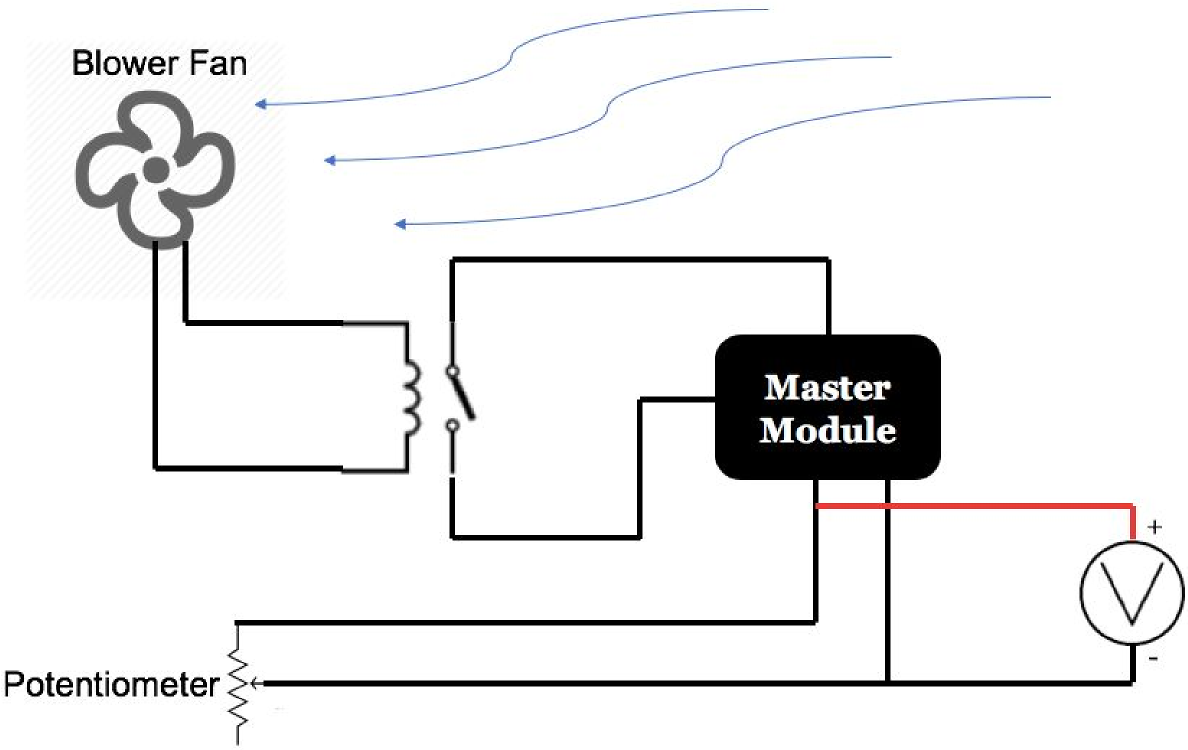


Fig. 84. Testing of wind speed variation with Control System submodule reference voltage

j. Wind speed step size variation with Control System submodule reference voltage

The wind speed step size is another important parameter to verify before delivering the WindTel project. This will be tested by incrementing and decrementing the reference voltage of the Control System submodule step by step and measuring the wind speed in each step using the anemometer. It is expected that the step size is equal between steps to prevent inconsistency with the relation between the reference voltage and the wind speed obtained from calibration. If this is not the case, then, the Control System submodule needs to be refined.

k. Data transaction from Wi-Fi submodule to database server

The storage of data from the Wi-Fi submodule to the database server will be tested by sending commands from the Wi-Fi submodule to the database. After each transaction, the database sends an acknowledgement to the Wi-Fi submodule indicating that the transaction was completed. It is expected that each data sent has a corresponding tuple allocation in the database. If this is not the case, troubleshooting in the hardware should be made to identify the error during the data transaction.

l. Drag and lift forces variation with rod length

As explained in Appendix F, the drag and lift forces varies with the rod length. This variation will be tested by acquiring the forces from the Balance Module and comparing them with the calculations made using the formulas in Appendix F. It is expected that the drag and lift forces approximate the calculated forces. If the results are not the expected, then the Balance Module needs to be verify and refined to obtain more accurate results.

m. Wind maximum speed with Control System submodule

The maximum wind speed of the wind tunnel with the Control System submodule will be tested using the anemometer readings. The test consists in specifying the maximum wind speed that the wind tunnel can produce (50 mph) in the user interface and acquiring the Dynamic Module measurement to compare it with the maximum wind speed. If the maximum wind speed is not obtained, then, the Control System submodule will be verified and refined.

2. Software Test Plan <- *(Kahlil, Misael)*

The Empotra'o Solutions' team will test and verify the software modules of the WindTel system with the following methods:

A) Software applications

Unit Testing - unit testing must be applied to each function created in the software. This will provide us a way to verify the functionality of each function individually.

White Box Testing - this method will be applied to test the internal structure of an item. This item is known to the tester and the tester may then choose the inputs to evaluate the paths through the code and analyze the outputs. It is necessary to conduct this method of testing at the beginning of the implementation since it will help with the progress of the software development. An example of use would be to inject a fault error to the internal structure and determine if the paths taken are correct or need verification.

Black Box Testing - black box testing will be used to test a set of functions to send a set of hardwired values as inputs and compare them with the expected results. This way we can partition the software by parts and provide a set of test cases to each partition to verify their correct functionality.

Security Testing - when signing up for the first time, the user will be asked to insert a password with three main restrictions: 8 characters minimum, at least 1 symbol, and at least 1 capital case. The password will show as asterisks while the user is typing, and combined with the stipulated restrictions, it will ensure that the password is not easily decoded with an external software.

B) Database server

Black Box Testing - it is necessary to test the queries to which requests to the database server will carry out. For this, we must use a method of testing such as this one in order to test the 4 methods of the Representational State Transfer Application Programming Interface (REST API), which are: Get, Post, Put, and Delete. Each of these methods are essential for the interaction with the database server.

Security Testing - we will provide a means to secure the access to the database server by restricting its access through the hardware. A user will be able to store results into the database if and only if the administrator gave the user permission to do so. The system will first verify that the user is permitted to store results and the system will then proceed to store results or prompt the user with a warning message stating that access is needed. To accomplish this, we will create dummy accounts and verify that they won't have permission unless an administrator gives permission beforehand.

Volume Testing - the database must be capable to collect multiple data and store them correctly without errors. To test this, a code will be provided to submit multiple data simultaneously to verify if it's managed properly in the database server.

C) Data visualization tool

White Box Testing - we will submit hardwired faults to test the functionality of the internal structure of the statistical analysis. This will ensure that if some values are null or empty, the user won't see a damaged graph on the user interface.

Black Box Testing - sample inputs must be tested and validated with the outputs. In this case, the inputs may be the parameters to be seen on the graphs, and the outputs would be the plots.

3. System Test Plan

Usability Testing - Due to the fact that system usability is one of the design goals for the WindTel project, the Empotra'o Solutions team will employ system usability testing metrics in the development of the project. In particular, the results obtained from these tests will be considered in the development of new iterations of the User Interface.
The team will use Jakob Nielsen's usability heuristics as well as the System Usability Scale (SUS)[40].

Below we present Jakob Nielsen's usability heuristics.

1. **Visibility of system status:** The system should inform and provide feedback to the user about the current status of the system.

2. **Match between system and the real world:** The system should be presented with terminology that matches the user's context and knowledge rather than technical or system-specific jargon.

3. **User control and freedom:** Users should be given the freedom to exit out of unwanted states.

4. **Consistency and standards:** The system should follow established conventions.

5. **Error prevention:** Prevent errors from occurring in the first place by avoiding error-prone conditions.

6. **Recognition rather than recall:** The user should be presented with ample guidance and instructions for how to navigate and accomplish tasks within the system.

7. **Flexibility and efficiency of use:** Provide for accelerators such that experienced or so-called "power users" can speed up system interaction.

8. **Aesthetic and minimalist design:** System dialogues should contain only as much information as is needed; strike balance between information density and aesthetics.

9. **Help users recognize, diagnose, and recover from errors:** System should provide clear error messages that express the issue concisely and suggest a solution.

10. **Help and documentation:** Provide system documentation that is easy to search and contains all relevant system information.

Below we present a sample survey that can be administered as part of the SUS method.

1. I think that I would like to use this system frequently.

2. I found the system unnecessarily complex.

3. I thought the system was easy to use.

4. I think that I would need the support of a technical person to be able to use this system.

5. I found the various functions in this system were well integrated.

6. I thought there was too much inconsistency in this system.

7. I would imagine that most people would learn to use this system very quickly.

8. I found the system very cumbersome to use.

9. I felt very confident using the system.

10. I needed to learn a lot of things before I could get going with this system.

Users answer these questions on a scale from “Strongly Disagree” to “Strongly Agree”. Based on this, a score on a scale from 1 to 100 can derived. Our goal is to achieve a score of at least 70 on this scale.

H. Economic Analysis <- *(Misael)*

A detailed summary for all cost categories is presented in the tables below. For Human Resources Costs, expected costs were calculated as the sum of all employee wages and benefits for the first two months of the project. The actual costs are equal to expected costs, minus the sum of all employee salaries and benefits for the two days in which the Empotra’o Solutions team could not work due to power and water outages.

Hardware costs were lower than expected due to the decision to only purchase 2 desktop computers.

Software costs were exactly the same as expected, as all software licences were purchased at the beginning of the project.

Facilities costs were lower than expected due to the database server maintenance being provided by the Department of Civil Engineering & Surveying.

For miscellaneous costs, travel costs were lower than expected, due to the team not having to travel during the first two months of the project. Risk contingency costs were lower than initially expected; the only two instances in which risk contingency measures had to be activated were during the power and water outages.

Employee	Work hours	Wages	Benefits
Misael Valentín	25.71	$844.71	$236.52
Luis Vega	25.71	$844.71	$236.52
Kahlil Fonseca	25.71	$844.71	$236.52
Nelson Rodríguez	25.71	$799.97	$223.99
Total		**$4,267.65**	

Table 31: Expected Human Resources Costs

Employee	Work hours	Wages	Benefits

Misael Valentín	25.71	$687.03	$192.37
Luis Vega	25.71	$687.03	$192.37
Kahlil Fonseca	25.71	$687.03	$192.37
Nelson Rodríguez	25.71	$650.64	$182.18
Total		**$3,471.02**	

Table 32: Actual Human Resources Costs

Equipment	Quantity	Cost
Oscilloscope	1	$1,600.00
Digital Multimeter	1	$1,000.00
Waveform Generator	1	$3,000.00
Soldering kit	1	$20.00
Desktop computer	4	$2,400.00
Android tablet	1	$250.00
Total		$8,270

Table 33: Expected Hardware Costs

Equipment	Quantity	Cost
Oscilloscope	1	$1,600.00
Digital Multimeter	1	$1,000.00
Waveform Generator	1	$3,000.00
Desktop computer	2	$1,200.00
Total		**$6,800**

Table 34: Actual Hardware Costs

Equipment	Cost	Period	Total cost

Android Development License	$25	2 months	$25.00
Microsoft Windows 10 Enterprise License	$85/year	2 months	$85.00
Microsoft Office Business Premium License	$12/month	2 months	$24.00
Microsoft Azure SQL Database License	$1,160/month	2 months	$2,320.00
Web hosting	$8.79/month	2 months	$17.58
Domain name	$14.99/year	2 months	$14.99
Total		**$2486.57**	

Table 35: Expected Software Costs

Equipment	Cost	Period	Total cost
Android Development License	$25	2 months	$25.00
Microsoft Windows 10 Enterprise License	$85/year	2 months	$85.00
Microsoft Office Business Premium License	$12/month	2 months	$24.00
Microsoft Azure SQL Database License	$1,160/month	2 months	$2,320.00
Web hosting	$8.79/month	2 months	$17.58
Domain name	$14.99/year	2 months	$14.99
Total		**$2486.57**	

Table 36: Actual Software Costs

Equipment	Cost	Period	Total cost

Capstone Laboratory	$100	2 months	$100.00
Wind Tunnel Laboratory	$100	2 months	$100.00
Office rent	$1,200/month	2 months	$2,400.00
Electricity	$60/month	2 months	$120.00
Water	$30/month	2 months	$60.00
Internet Service Provider	$80/month	2 months	$160.00
Database Server Maintenance	$1,500/month	2 months	$3,000.00
Total		**$5,940.00**	

Table 37: Expected Facilities Costs

Equipment	Cost	Period	Total cost
Capstone Laboratory	$100	2 months	$100.00
Wind Tunnel Laboratory	$100	2 months	$100.00
Office rent	$1,200/month	2 months	$2,400.00
Electricity	$60/month	2 months	$120.00
Water	$30/month	2 months	$60.00
Internet Service Provider	$80/month	2 months	$160.00
Total		**$2,940.00**	

Table 38: Actual Facilities Costs

Transportation	Cost
Food	$480.00
Travel	$850.00
Risk contingency	$2,400.00
Total	**$3,730.00**

Table 39: Expected Miscellaneous Costs

Transportation	Cost

Food	$480.00
Risk contingency	$1,680.00
Total	**$2,160.00**

Table 40: Actual Miscellaneous Costs

Category	Expected	Actual	% Difference
Human Resources Costs	$4,267.65	$3,471.02	**-18.67%**
Hardware Costs	$8,270.00	$6,800.00	**-17.78%**
Software Costs	$2,486.57	$2,486.57	**0%**
Facilities Costs	$5,940.00	$2,940.00	**-50.51%**
Miscellaneous Costs	$3,730.00	$2,160.00	**-42.09%**
Total	**$24,693.65**	**$17,857.59**	**-26.68%**

Table 41: Total Costs

I. Task Progress and Gantt Chart <- *(All)*

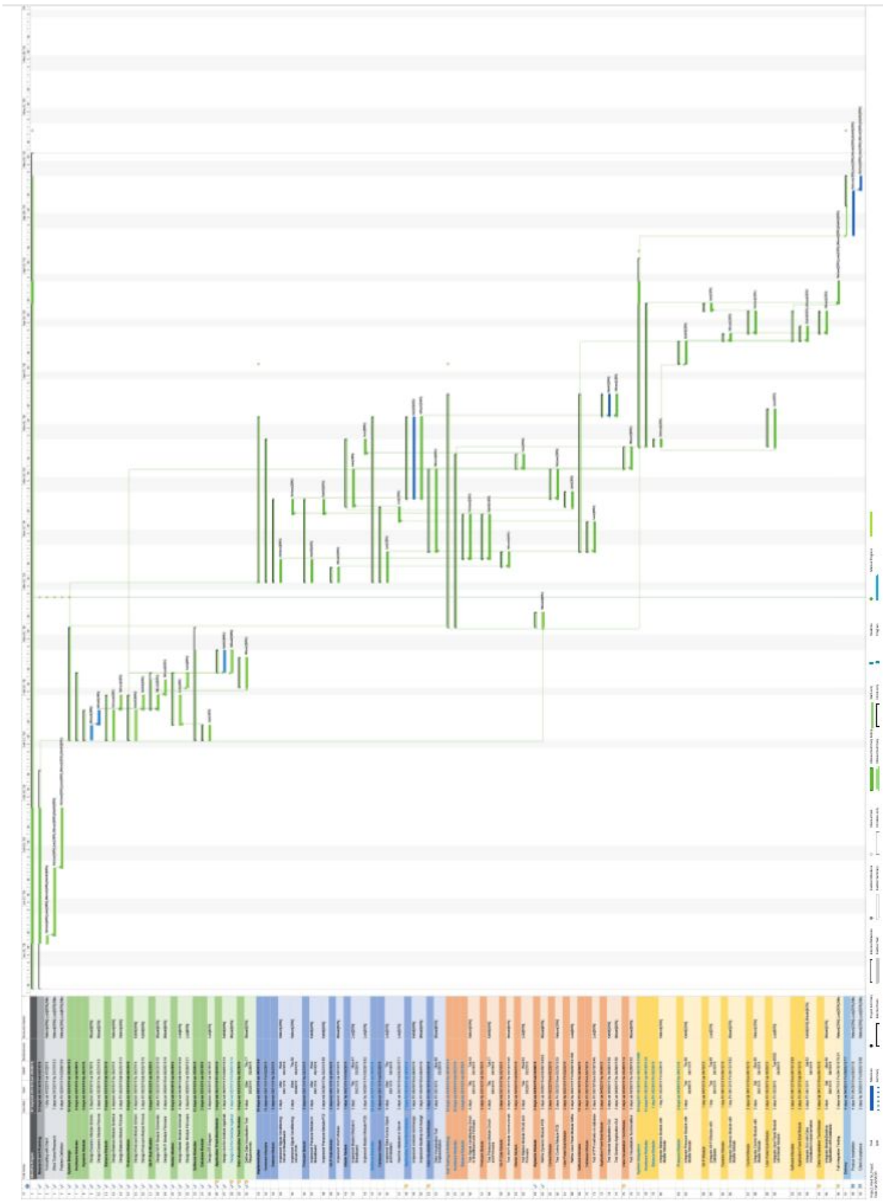


Fig. 85. WindTel Project Gantt Chart

For a more information regarding the WindTel project Gantt Chart visit:
https://drive.google.com/drive/folders/1Us_UJZKFMHVj1JF8lDRlFCawCjwxGcQD

J. Ethical Analysis of the Project <- *(Kahlil)*

Ethical Conflicts and Parties Affected

Parties like the Aerodesign team from the Mechanic Engineering Department from UPRM and the University of Turabo student can use the wind tunnel. This causes that the wind tunnel services act as a queue and thus wind tunnel user access is limited.

Solutions using harm, reversibility, and publicity techniques

The wind tunnel administrator should establish a priority when it comes to operating the UPRM wind tunnel system. In a passive manner the user should reserve the use of the system throw email notification.

Conflict

The wind tunnel administrator restrict the access of the wind tunnel to authorize personal. Each wind tunnel user will be inform of the wind tunnel operation procedure and the operation will be supervise by the wind tunnel administrator or wind tunnel electronic technician.

K. Environmental Impact Assessment of project <- *(Luis)*

Natural resources or environmental factors, and the actions of the project that cause an impact

Part of the actions that can be carried out would be to promote our product in the university so that both teachers and students can make use of this resource for free.

Environmental, cultural, or social impacts - positive or negative - of the project, and assesses the most significant impact

1. With the WindTel system you can have a resource to conduct research in the areas specialized mainly in aerodynamics.

2. You can acquire data autonomously and save the results in real time on the remote database.

3. A negative aspect is that it will cause noise when conducting an experiment, besides that it will be sucking air from the environment that may affect, in a way, nature.

Environmental management measure

The impact that this will have will be to be accessible and usable the wind tunnel located in the civil engineering department, in addition to obtaining exact measurements within a range of 10%. This impact is called an environmental and cultural impact because it is accessible
for the university community, will cause it to be used more frequently and therefore There will be a higher frequency of noise caused by the wind motor when it is used. Laws that can against this type of noise is the Law Number 131 of August 9, 1995 that talks about noise unnecessary and the penalties that these can entail, and Law Number 71 of April 26, 1940 that talks about the distortion of Public Peace. Although it is a public resource for the community university, these laws must be taken into account since it is possible that charges will be awarded for Noises caused by the engine.