UNIVERSITY OF PUERTO RICO AT MAYAGÜEZ

MAYAGÜEZ, PUERTO RICO

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

AeroBal: An Automated Wind Tunnel

FINAL REPORT SUBMITTED AS A REQUIREMENT OF ICOM 5217 - MICROPROCESSOR INTERFACING

by

Anthony Llanos - Leader
Jesús Luzón
Juan Lebrón
Jean C. Méndez

For: Dr. M. Jiménez
Course: ICOM 5217, Section 098
Date: December 9, 2013

Abstract

Wind tunnels are used for researching aerodynamic characteristics which are the effects of moving air on solid objects. One of the characteristics studied is the force that air causes on object. To do this, the wind tunnel of the University of Puerto Rico, Mayagüez Campus uses technologies that are dated and have to manually be performed, which introduces complexity and error into experiments conducted. In this report, a system that would improve the one previously described is presented. The system would improve the current wind tunnel in an important way by adding sensors connected to a system that performs the procedure of obtaining the measurements automatically. The main sensor used in this system are strain gauges which allow the measurement of force electronically.These are integrated with a device that allows the forces to be transferred to the strain gauges. Additionally, different parameters of the experiment must be studied such as temperature, pressure, humidity, wind direction and speed, for which a sensor for each one is included in the system. To allow easier interaction, the system provides both a hardware LCD interface and a Bluetooth connected software mobile application that additionally performs storage of the data for later processing. To connect all of these components together, a Tiva TM4C123GH6PM Microcontroller is used as the controlling component of the system. AeroBal would make recording data much simpler than it currently is, and could help make research easier in the future for constant users of the tunnel, such as mechanical and civil engineering students.

Table of Contents

List of Figures

1. Introduction

Wind tunnels [1] are structures where an object is placed to study its aerodynamic characteristics. The tunnel pulls air into it to achieve laminar flow, and the motion of this air acts on the object, causing forces (drag, lift, side) and moments (pitch, roll, yaw) to act on it. To study large objects, scaled down versions of these models are placed inside the tunnel and the values are then scaled to their counterparts.

Numerous institutions around the world use wind tunnels. Some examples are NASA, the University of Maryland [2], and the University of Southampton [3]. The Civil Engineering Department of the University of Puerto Rico, Mayaguez Campus [4] has a small 2-room wind tunnel used for research. In the past, this tunnel has been a central part of analyzing the aerodynamics of numerous projects, such as a Solar Car, ailerons, and tanks of storage for fluids.It was constructed in 1983 and since then it has not been improved much.

The technology used in the tunnel is purely mechanical and requires manual intervention by the experimenter. To measure force the tunnel uses a balance that allows free movement in the horizontal and vertical axis, using a design that connects various steel bars together. Below we can see an image of the design implemented in the Civil Department Wind Tunnel:

Figure 1: Current Mechanical Balance used in the wind tunnel.

Whenever an object is placed in the tunnel and the wind tunnel is started, the object tilts the balance out of equilibrium. An example of the tilted balance can be seen in Figure 2. The direction of the tilt is caused by the drag and lift forces acting on the model. The data from the balance is obtained by pouring sand on whichever cup needs it in order to balance the device until it in equilibrium again. Then the user takes the plastic cup and weights the poured sand to obtain the force magnitudes.

Figure 2: Mechanical Balance Tilted.

The process of obtaining the data using the mechanical balance of this particular wind tunnel can be time consuming and error prone due to this manual intervention. We propose an electronic system to renew the mechanical balance, having a microprocessor as a central point, in order to improve wind tunnel as a whole. Our model would modernize the current tunnel in a meaningful way without having the economic implications of buying an entire new system.

2. Theoretical Background

2.1 Brief Historical Background

Wind tunnels date back to 1871. These first tunnels were mostly small and enclosed, used to test aerodynamic characteristics of objects. One of the most popular wind tunnels from the 1900's was made by the Wright Brothers in their development of their Wright Flyer [8] as can be seen on Figure 3.

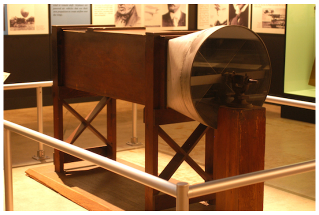


Figure 3: Wright Brother's Wind Tunnel [9]

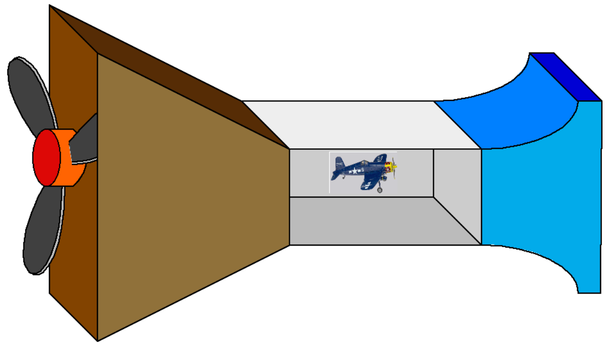


Figure 4: Example of a simple wind tunnel.

Progress and importance of wind tunnels increased during the Second World War. One of the biggest tunnels ever constructed was made in Wright Field in Dayton, Ohio in 1941. [10] This tunnel had measurements of 45 ft in height and around 20 feet in width. Large aircraft models were tested here at speeds of around 400 mph. By the end of World War Two, the United States had built various tunnels, including an even bigger one in Moffett Field near Sunnyvale, California. Since then, a great number of institutions around the world have built their own wind tunnels, including the University of Puerto Rico.

2.2 How It Works

A wind tunnel consists of three main parts: (1) the wind source, (2) the studied object, (3) the sensors to obtain the measurements.

2.2.1 Wind Source

The wind source of a wind tunnel can come from different devices. Small tunnels like AeroBal use a simple conventional fan found in any discount store. It's speed does not exceed 25 mph. Larger wind tunnels used in major organizations use fans with speeds higher than 100 mph. Special types of wind tunnels, called Supersonic Wind Tunnels, can produce supersonic speeds (around 345 mph).

2.2.2 Studied Object

Arguably the most important element of a wind tunnel is the object being studied in it. In small wind tunnels, the objects are usually replicas of larger versions of that object. Objects such as cars, helmets, rockets, and airplanes, and which are normally under the effect and dependent of the motion of air, are studied.

Figure 5: Miniature Airplane used inside a NASA Wind Tunnel. [12]

2.2.3 Sensors

The main sensor used in the tunnel is the weight sensor that measures the forces. To measure the force of drag and lift, only the force while under the effect of the motion of air must be calculated. There are two stages for an experiment: 1) calibrate the instrument to equilibrium with no wind, 2) with the wind turned on and causing the instrument to not be in equilibrium, return the instrument to equilibrium using some form of weight which can be measured. Therefore, the weight used to return to equilibrium represents the measurements of drag and lift, depending on where they are placed in the instrument, to counter the components of them:

$$Component_{object} = Weight_{Wind} - Weight_{NoWind}$$

Should the object use a *base* for the object which is considerably affected by the wind, then the base's components (drag and lift) must be measured using the same procedure, such that these are substracted from the end results which contain a combined measurement of both the base and the object.

$Component_{object} = Component_{Combined} - Component_{Base}$

The weight sensor at the UPRM Wind Tunnel is implemented by manually pouring sand, and then using a weight scale to measure its weight. Other factors measured with sensors are temperature, pressure, relative humidity, wind direction, and wind speed. These can be implemented with different sensors of each type and are measure before and during the wind tunnel's operation.

3. System Overview

AeroBal consists of balance (scale) device to act as the sensor for force, sensors to read experiment environment data, and a user interface. The scale can be connected to a prototype wind tunnel custom-built by the team to test the implementation of the sensors. In our case, we made our own small wind tunnel for testing, as shown below in the top level view of the system:

Figure 6: System Top Level View.

One of the key components of the system are the *strain gauges* which measure the amount of force exerted from the wind on to the object being measured. The strain gauge is a device whose electrical resistance varies in proportion to the amount of strain being caused on it *[NI reference]*. The strain gauges are normally used in load cells which are a combination of a mechanical arrangement and a strain gauge. The mechanical arrangement is deformed by an external force and the strain gauge tied or glued to the arrangement is indirectly deformed as well.

To implement the strain gauge component of the system, load cells were used, and calibrated using a postal scale which contains a load cell as well. Multiple values were used to the order of 2.3 grams or the weight of a dime. Different amount of coins were placed both on a scale and on the load cell such that the load cell could be characterized by obtaining multiple values and then obtaining the equations of each one.

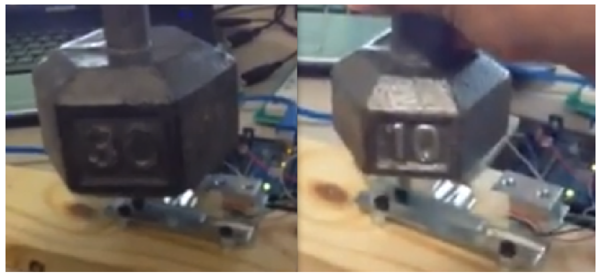


Figure 7: Calibration using weights.

A total of 4 strain gauges were used such that the both positive and negative components of the drag and lift forces were measured. A balance was designed based on the same concept provided by Dr. Zapata that provided free movement in the 3 axes. For this the designed however, the balance was enclosed between the strain gauges such that it did not move as depicted below. The object is attached to a base that is suspended from a rod that is directly connected to the balance, transferring all forces to it. The design of the balance is such that the forces are separated correctly into the component because of its movement.

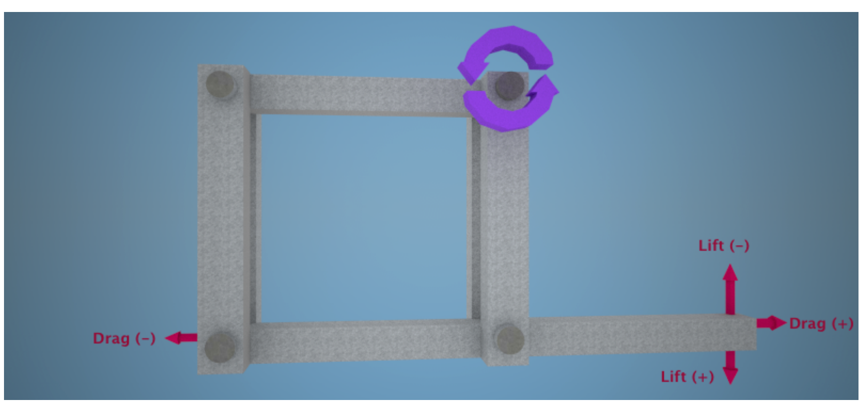


Figure 8: Balance Concept and its force detection diagram. [12]

Besides the strain gauges, there are sensors to detect temperature, wind direction, humidity, pressure, and wind speed. A servo is used to control windows that allow the passage of air in the tunnel depending on their opening, and thus controlling wind speed. The user interface consists of an LCD screen and LEDs for output, and buttons for input, as well as a tablet application that allows remote control of the system from small distances using Bluetooth.

All of these things are interfaced using a Tiva C Series Cortex-M4 microcontroller (TM4C123GH6PM) from Texas Instruments. The main purpose of our system is to automate as much as possible the research conducted in the current wind tunnel at UPRM.

4. System Block Diagram

The following image presents the final system block diagram of the system:

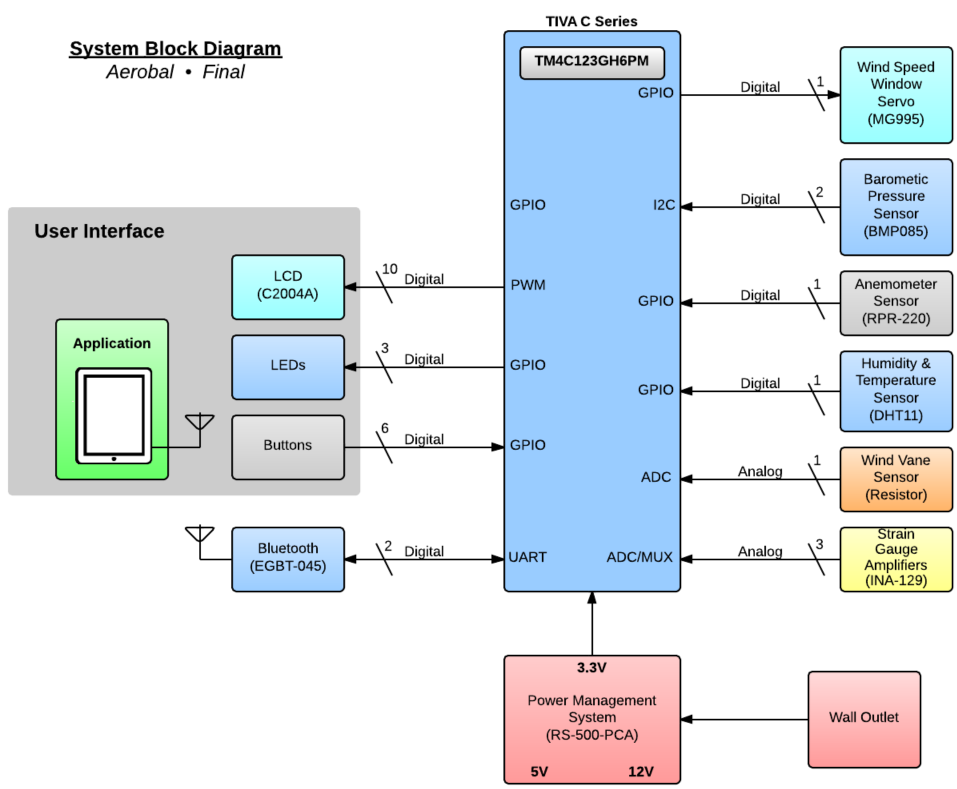


Figure 9: The final version of the AeroBal system block diagram.

5. Power Analysis

Power analysis requires that we consider the logic compability of the signals, the driving compability of the components, the design of the power supply and the load on the microprocessor in use.

5.1 Logic Compatibility

The following listing and table shows all of the major components of AeroBal and their voltages:

- **DHT11**: Humidity and Temperature Sensor
- **BMP085**: Barometric Pressure and Temperature Sensor
- **MG995**: Servo
- **C2004A**: LCD Interface
- **INA129**: Instrumental Amplifier
- **EGBT-046S**: Bluetooth
- **Relay**
- **TM4C123GH6PM**: Tiva MCU

Component	V_{DDRec}	V_{IH}	V_{IL}	V_{OH}	V_{OL}	I_{Drive}	I_{Source}
TM4C123GH6PM	3.3V	2.48V	1.16V	2.4V	0.4V	*	31.5mA
MG995	5V	*	*	*	*	*	450mA
BMP085	3.3V	2.15V	0.66V	1.32V	0.3V	3mA	5uA
DHT11	5V	*	*	*	*	*	2.5mA
EGBT-046S	3.3V	2V	0.66V	2.64V	0.66V	*	40mA
C2004A	5.0V	2.2V	0.6V	2.4V	0.4V	0.8mA	4.0mA
INA129	5V	2V	0.8 V	1.4V	0.8 V	*	750uA
Relay	3.3V	*	*	*	*	*	30mA
RPR-220	5V	*	*	*	*	*	10.3mA
	Safe $\pm V_{IN}$	$+V_O$	$-V_O$	P.Supply			
INA129	±40V	+V - 0.9	−V + 0.8	±15V	-	-	-

NOTE: The extra two analog components (Wind Vane and Anemometer) each work at 3.3V and 5V, respectively. Thus, there are no compatibility issues with those two components either.

* The information is not specified in the datasheets or is not applicable (not digital).

5.2 Driving Capability

Driving Capability testing ensures that components are provided the required current to function properly. The Tiva Microprocessor can provide from each GPIO port a maximum of 8mA. All components are therefore drivable if needed by GPIO ports. For components whose current drive was not found, the Tiva was tested and confirmed to be working with the component. Power supply cannot be provided by a single Tiva MCU, as in testing it was found that both the servo and the relay did not properly work without interfacing with an external power supply therefore these components have made the system require an external power supply.

5.3 Power Supply Design

For AeroBal we chose the LPK 230 power supply which follows the ATX specification. It is a simple computer power supply that we found and preferred because it has an output of +12V, +5V, and +3.3V as specified in the standard. Computer power supplies by definition regulate the voltage for use in a computer , and some of them contain other components like short circuit, overload, overvoltage, undervoltage, overcurrent, and overtemperature protection.

5.4 Port Loading

The following table shows the current capabilities of our power supply:

Source	Voltage	Current
1	+3.3V	25A
2	+5V	25A
3	+12V 1	16A

Here we show the calculations for current needed from each port:

- **+5V connector** (Capacity: 25A):

  This connector will have four components, the instrumentational amplifiers, LCD, Humidity and Temperature Sensor, Servo, and the Anemometer Sensor :

$$750uA + 4.0mA + 2.5mA + 450mA + 10.3mA = \mathbf{0.46355A} << \mathbf{25A}$$

- **+3.3V connector** (Capacity: 25A):

  This connector will have four components: the relay, Bluetooth Module, Pressure Sensor, and the MCU:

$$30mA + 40mA + 5uA + 31.5mA = 101.505mA = \mathbf{0.101505A} << \mathbf{25A}$$

Hance the device implementation will consume a *total current* of:

565.055mA

Power consumption for the system:

$$(0.46355A * 5V) + (0.101505 * 3.3V) = \mathbf{2.6527W}$$

5.5 Thermal Analysis

On thermal analysis we need to determine the maximum power dissipation the chip can handle. Using function 8.11 from the book:

$$P_{diss(max)} = (T_{J(max)} - T_A)/\theta_{JA}$$

According to the datasheets:

$$T_{J(max)} = 150C$$

$$T_A = 25C$$

$$\theta_{JA} = 54.8C/W$$

$$P_{diss(max)} = (150C - 25C)/54.8 = 2.28W$$

Hence, 2.28 W is the maximum power dissipation of the microprocessor.

The current required by the processor while running at the speed of 16MHz with V_{DD} = 3.3V and ambient temperature 25C is 31.5mA. The current required from GPIO devices should they be both powered and interfaced by the MCU would be (With the exception of the Relay and the Servo which have been confirmed to need an external power supply):

$$3mA + 2.5mA + 40mA + 4.0mA + 10.3mA = 60mA$$

$$I_{DD(avg)} + \sum_{all pins} c_i = 45.1mA + 60mA = 105.1mA$$

Multiply by voltage level V_{DD} = 3.3V:

$$P_{diss} = 3.3V * 105.1mA = 346.83mW < P_{diss(max)} = 2.28W$$

Therefore our system complies with the thermal limitations of the chip and it is safe to use for the application.

5.5.1 Fan Relay

The Fan Relay datasheet specifies that the power consumption of the relay coil is 0.36W. This component is not directly supplied power by the microprocessor but from an external source and therefore this dissipation does not affect the microprocessor thermal analysis.

5.5.2 Servo Motor

The datasheet for the servo does not specify power consumption however an auxiliary version of the datasheet was found in [13] and using this information is how we state the current consumption which is 450mA.

However, while researching about how to calculate the power dissipation of a motor, it was found in [14] that the resistance of the motor windings of the motor is needed to be able to calculate the power dissipation. The datasheet of the motor being used does not provide such parameter, and after consultation from a student it was found that special equipment would need to be used to determine such parameter. Once obtained, the procedure highlighted in the T́hermal Calculationsśection of the webpage may be used to calculate its dissipation.

6. Timing Analysis

Various components of AeroBal need timers to work. We had to determine the base times of each one with the Tiva C's provided clock source and determine if it was compatible. After testing, it was found that all components worked with the default clocking in tests which is from the 16Mhz Internal Oscillator of the MCU. Some components require clock switching in order to function properly such as the ADC. These details are provided below.

6.1 Time Bases

The following table shows the minimum pulse widths of all the components in the system that require a clock source. This will give us an idea of the time bases for each module.

Component	Min. PW	Timer/Clock Parameters *
Barometric Pressure Sensor	0.29μs	Internal I2C Timer (100Kbps) (**)
Custom anemometer (Rot. Encoder)	0.5s	Timer: SysCtlClockGet()/2
Bluetooth Module	125ns	Internal UART Timer (9600bps)**
Humidity and Temperature Sensor	1μs	Timer: SysCtlClockGet()/1000000
	-	Timer: SysCtlClockGet()/55 For 18 ms
	-	Timer: SysCtlClockGet()/25000 For 40us
	-	Timer: SysCtlClockGet()/14900 For 67us
ADC	(1/16Mhz)	**SysCtlDiv10** Divider on Clock ***

* For the Tiva API, timer parameters are provided by invoking the SysCtlClockGet() function which returns a timing base of 1 second. This value is then divided by inverse of the period required for the timer to obtain its frequency.

** Since the Tiva contains three internal oscillators (Pg 217 TM4C123GH6PM datasheet) clocking may be provided for modules who are performing clocking dependent operations without affecting the main operation of the processor.

*** A divider is applied to the whole clock operation of the Tiva. This means that during the operation of the component that requires it, operations of all the components are applied this clocking scheme if interfaced. However in software, clocking is only set during the operation of the component that requires it.

6.2 Analysis

A **functional** minimum pulse width: **Source:** Tiva w/ Bluetooth Module @ 9.6 Kbps Baud requires 8 Mhz Frequency. This forces the processor to have a minimum frequency of:

$$8Mhz < 16Mhz = ChosenTivaFrequency$$

Tiva gives us the benefit of having up to 80 Mhz to work with. We have chosen the default 16Mhz clocking scheme used in examples and tutorials and found that components work at the frequency.

7. Hardware Reliability

7.1 Precision and Convenience

Our system was designed specifically to make the user's usage of the wind tunnel convenient, while keeping the precision previously had. The strain gauges measure what is probably the most important parameter of the system, force. They currently measure in ounces and pounds, with a precision of around 0.01 oz. The method used to obtain their measurement was to characterize the gauges using different weights as mentioned earlier. This may present inconsistencies especially if they are deformed. The strain guages can deform if too much force is applied to them specially if well beyond their capability. As per the functional requirements of the system at the initial stage, professor Zapata stated that the heaviest component that would be measured in the system would be of 15lb and hence we have chosen strain gauges with a capability of 30lb to well exceed this range. Should they accidentally exceed this range, the strain gauges must be replaced.

The Android Application serves as an extension to the hardware User Interface of the system, providing remote control of the system components and operation and an important feature of the system which is storage of data. The data compiled from each experiment is stored in the system with the date and time of the experiment to allow the user to access this data for later analysis if desired. Currently the Android application has the capability to run custom built analysis tools for the experiment but these have been not implemented yet. The operating system application interface allows for many possible ways to handle and manipulate the data received from the system, giving this component great importance for expansion in future work.

The pressure and temperature sensor provides data presicion up to 0.1 C and 1 Pa. The Humidity and temperature sensor were programmed to provide data up to 1% and 1C of the respective parameters.

The wind vane angles were mapped using a protractor. The anemometer was tested by putting it in the top of one of our cars, and we compare the speed given by it with the miles-per-hour count given by the car. This test gave us an error of less than 10%. For example, when the car was driving at 25mph, the anemometer displayed around 28mph, and when the car slowed down to around 10mph, the anemometer displayed 11 mph. The wind vane is characterized to a range of 28 degrees to 290 degrees due to the potentiometer not being able to rotate 360 degrees completely.An alternative would need to be considered if it is required to rotate completely although in the wind tunnel the potentiometer was placed such that the direction of the wind was in the middle if this range and if there was rotation then the wind vane would had $(290-28)/2 = 131$ degrees to move.

As of the current system implementation, noise reduction for the AC components has not yet been implemented in hardware but only in software. Averaging and multiple sampling methods are used in software. This hardware noise reduction component is planned for future work.

8. Memory Usage

The following diagrams show the memory usage of our software in our Tiva C MCU. It was found directly from our IDE, Code Composer Studio while executing the latest version of the software and having run a few experiments on it.

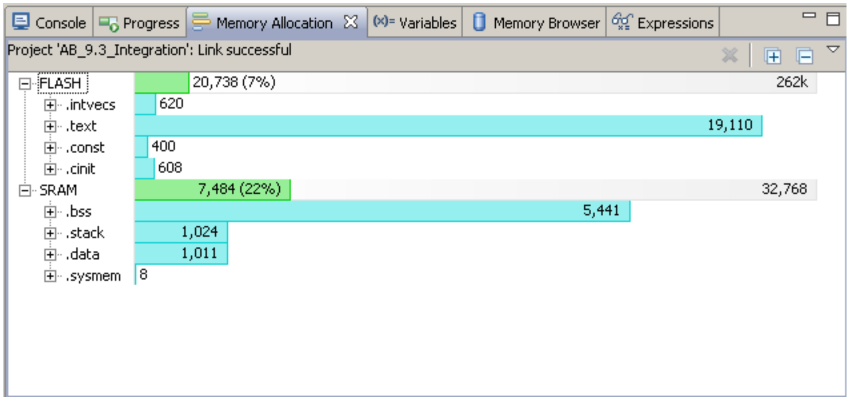


Figure 10: Diagram of the memory usage of all of our software, according to Code Composer Studio, during a particular execution.

The **.text** section of the FLASH memory has the program and it states that the program is 19.11kB in size and this includes all of the libraries that are used for programming the ARM microcontroller using the Tiva ARM API. Due to the size availability of the Tiva we can see that the system uses less than 10% of the flash memory provided by the MCU. This provides a lot of opportunity for the expansion of the system software without having to introduce memory components. Should the elements be permanently stored in the MCU (they are not, but in the Android application they are as previously stated) then depending on the amount of data to store then the system might need to have an additional memory component added.

9. Level Of Completion (Hardware)

The following table represents the completion of each of the components of the Aerobal System as divided in the team.

Module	Completion	Notes
Servo (Windows)	100%	
Humidity & Temperature	100%	
Strain Gauges	100%	
Bluetooth	100%	
Anemometer	100%	
Wind Vane	100%	
Pressure Sensor	100%	
LCD	100%	

10. Software Plan

We include the flowcharts for the software in our system. We give one page for each one for clear understanding to the reader.

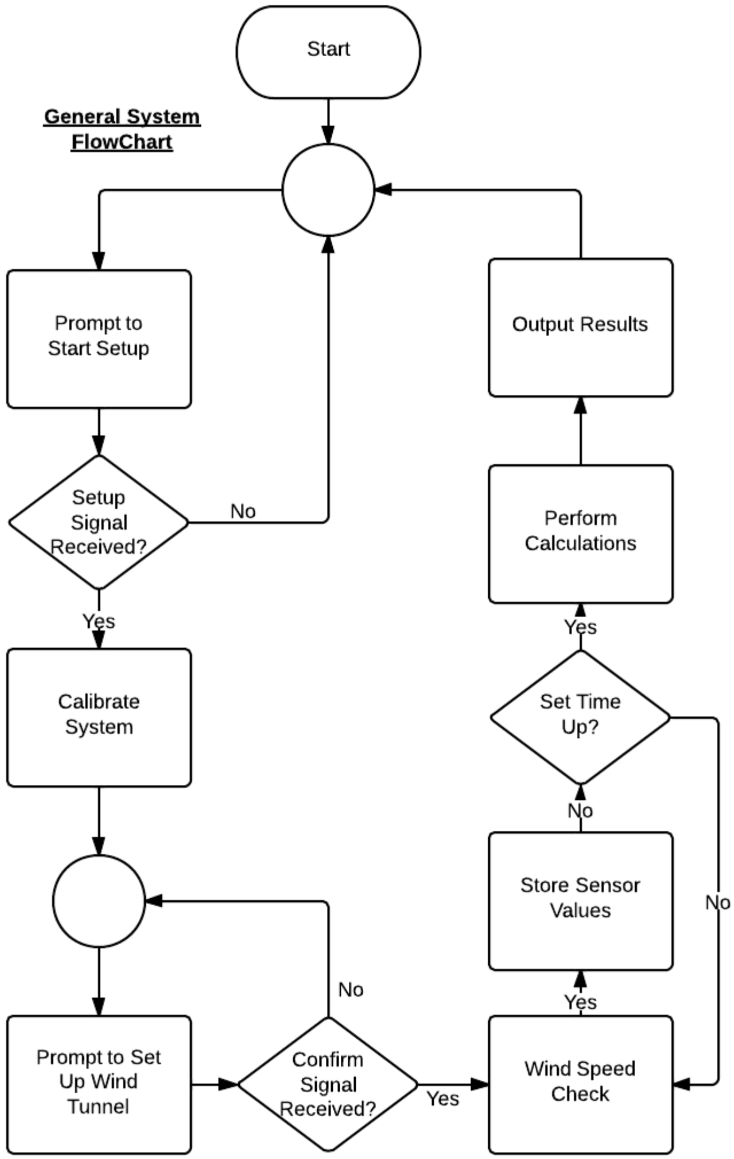


Figure 11: The general/main system procedure flowchart.

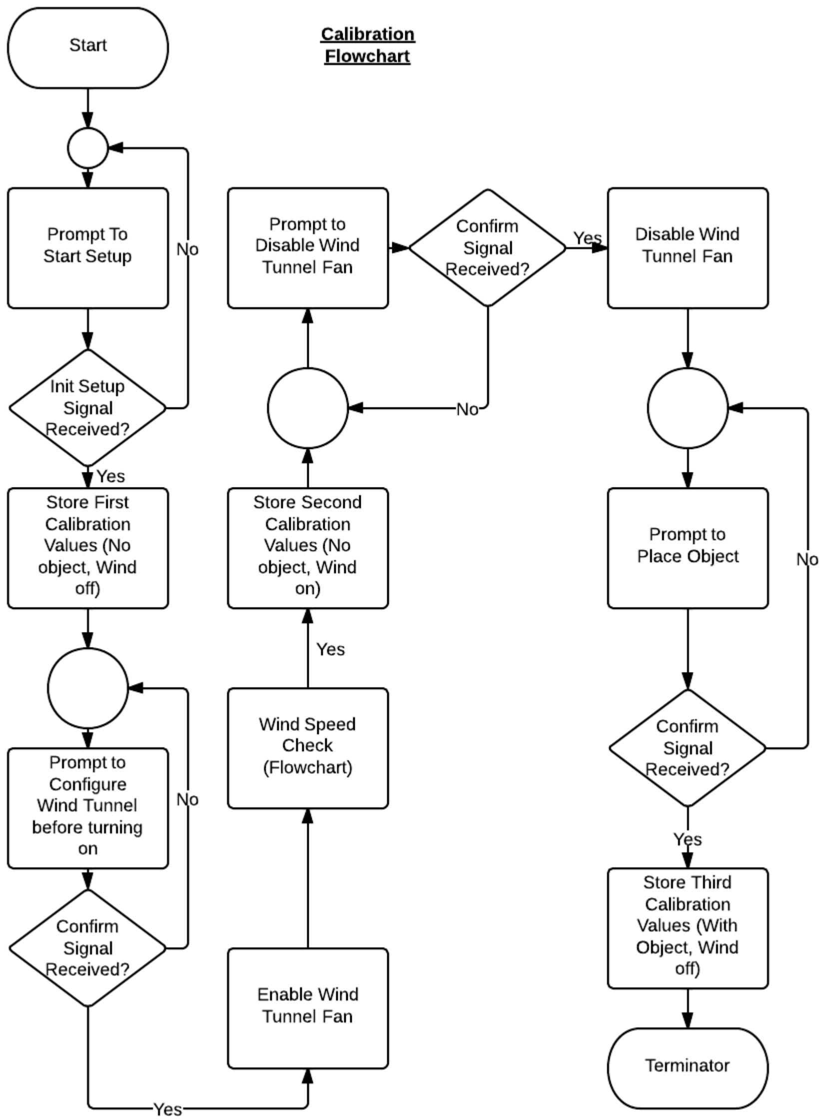


Figure 12: Calibration flowchart to calibrate the strain gauges.

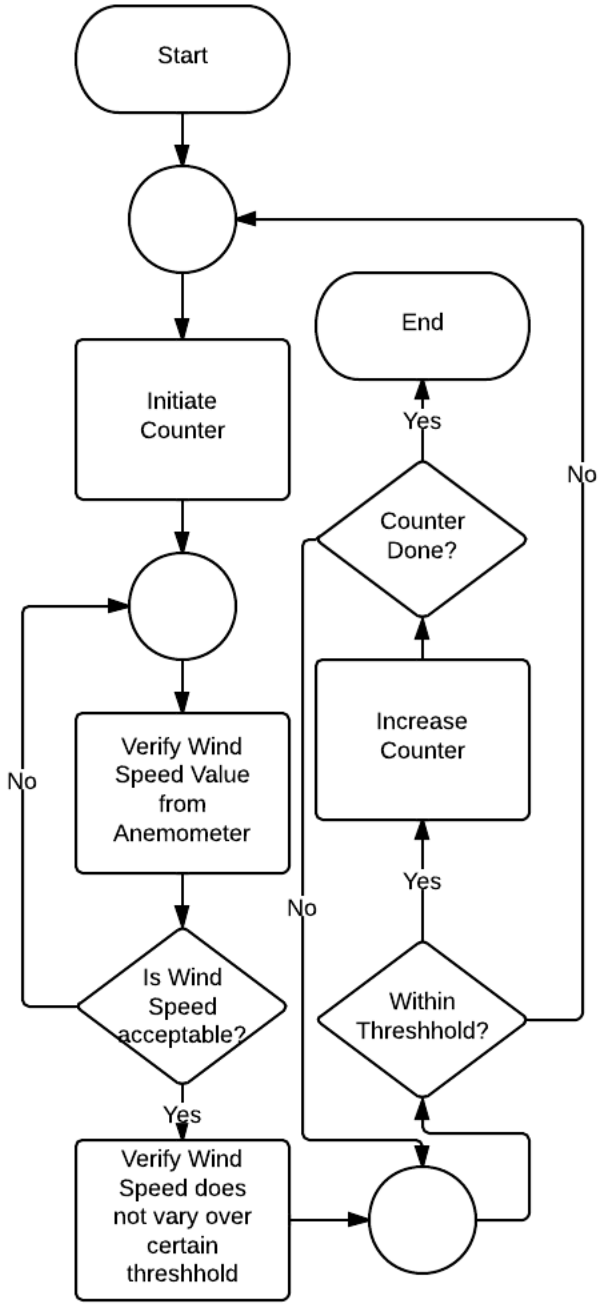


Figure 13: Wind Speed Checker flowchart.

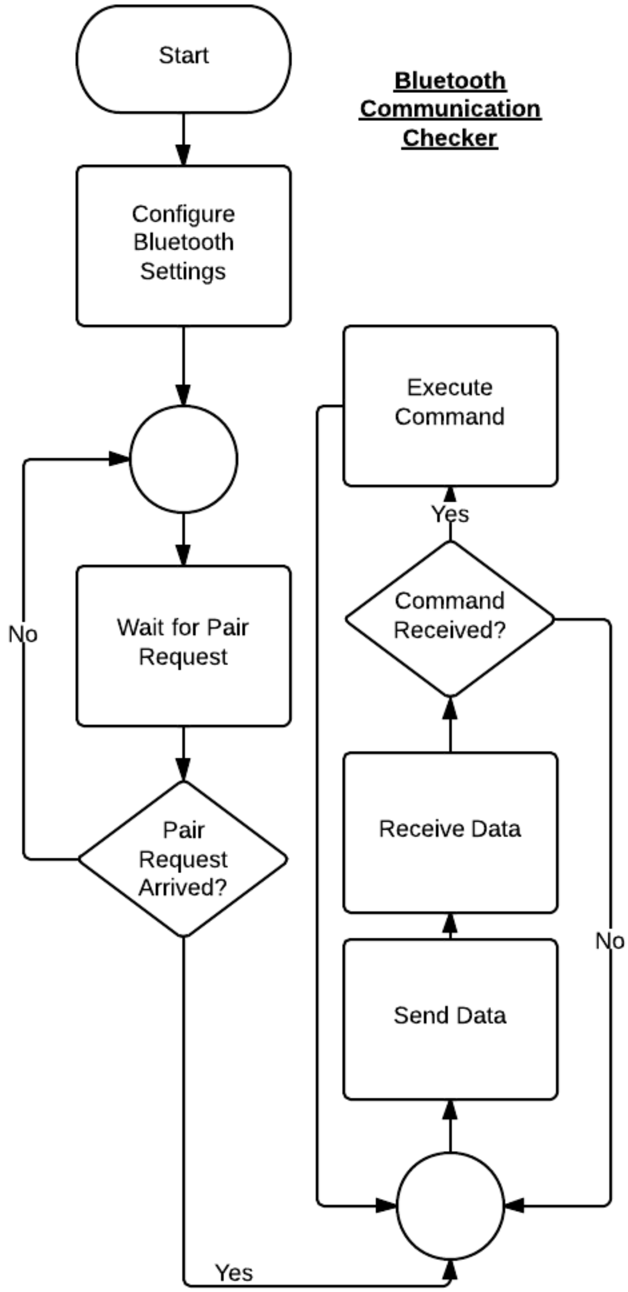


Figure 14: Bluetooth Communication checker.

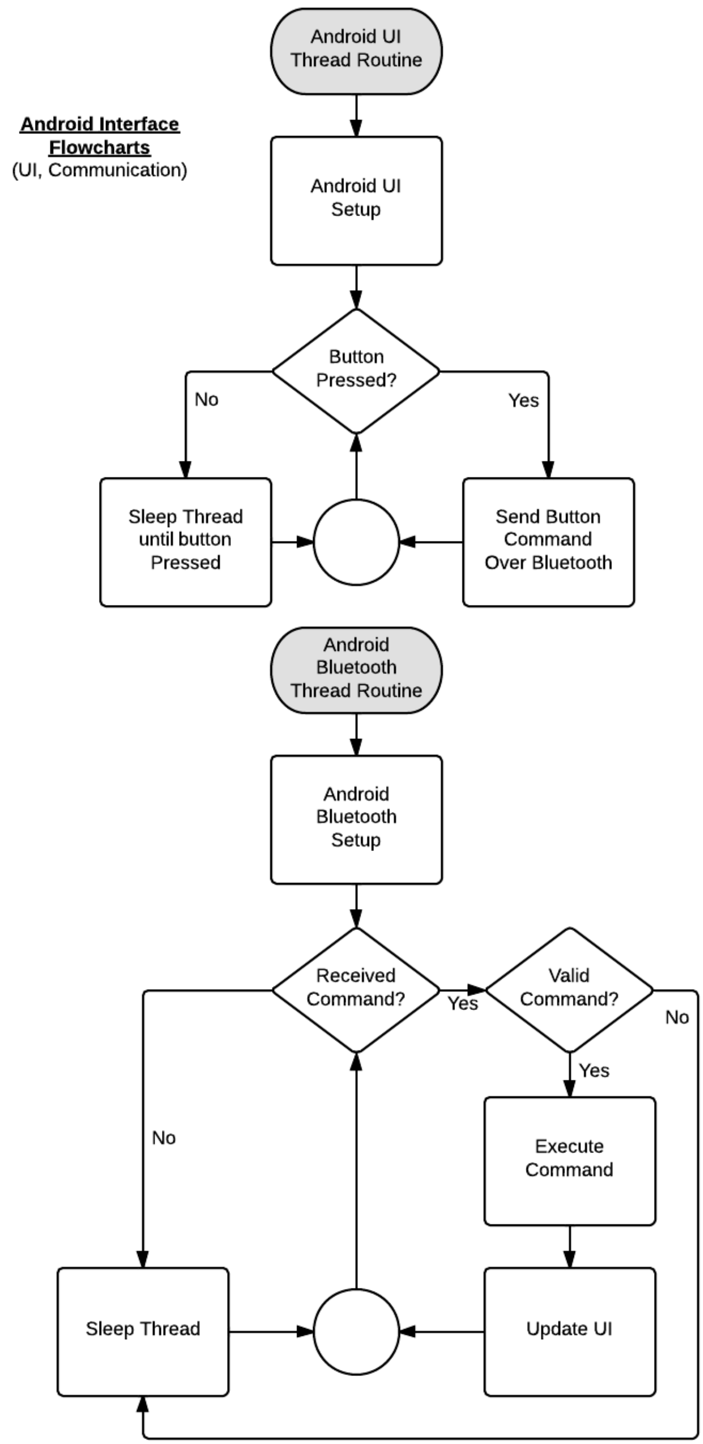


Figure 15: Bluetooth Communication checker.

11. Multi-Unit Software Organization

A software framework was developed for the implementation of this project as a means of abstracting the processes and replicating the flowchart to the maximum extent possible. Each component integrated in the project was modularized with self-made libraries, some with simple functions, others with propietary software libraries from TI. to make their interfacing process easier. Each sensor has a library with instructions on how to install the library and how to assign their ports and interrupts. As for the structure of this programming. In the following image, a description of the software organization:

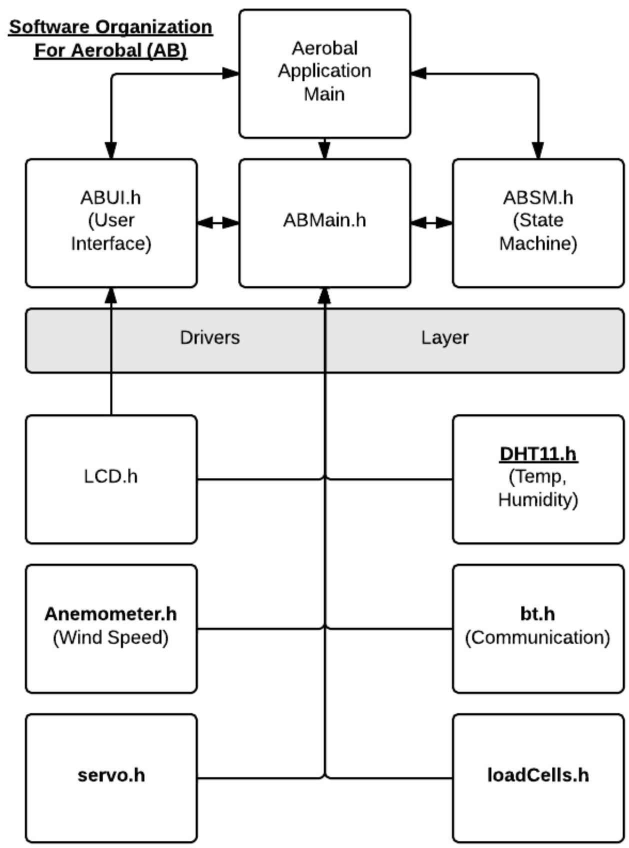


Figure 16: Software Organization Chart of our system.

A state machine for the main application was developed such that it replicates the flowcharts planned for the application. This state machine can be transitioned from state by both hardware and software interface.

12. Software Reliability

12.1 Overall Reliability and Software Expansion

Our software consists of a robust state machine with each component added through a library. Due to this, it works in a modular way, by calling functions from each library in the main method and running them through the state machine, one component at a time. Its design allows for additional functionality to be integrated to the system by adding only states and conditions allowing easy expansion of the software. A copy of the data will be saved in the android application. This maintains data integrity and allows the usage of data in both interfaces.

12.2 Data Integrity

Analog components have noise by nature and therefore need methods to eliminate it. In an effort to reduce noise and obtain measurements correctly, the values obtained from the sensors that are analog are averaged. The ADC of the MCU has the ability to take multiple samples of a measurement and average those samples to the magnitude of 64 samples. These are averaged as well aside from the software averaging previously mentioned of each value obtained.

13. Level of Completion (Software)

The following table represents the completion of each of the components of the Aerobal System as divided in the team.

Module	Completion	Notes
Servo (Windows)	100%	
Humidity & Temperature	100%	
Strain Gauges	100%	
Bluetooth	100%	
Application	100%	
Anemometer	100%	
Wind Vane	100%	
Pressure Sensor	100%	
LCD	100%	

14. Conclusion

In conclusion, we have successfully developed an embedded system that can make the research conducted in the Civil Engineering Department more productive by providing an system that automates the procedure being performed. The main objectives which were to measure the aerodynamic forces were achieved by the use of the strain gauge sensors which provide a very high presicion measurement of the forces to the order of 0.01 oz which is well below the required 0.01lb specified by Professor Zapata. The measurement of the forces is therefore more accurate and the balance designed even achieves adding a third component for measurement named side which measures the horizontal force perpendicular to the drag force thus providing complete measurement in free space. The use of strain gauges and its interface with the microprocessor eliminates most of the manual intervention by the experimenters. The design of the balance makes it portable and adaptable to the wind tunnel already present in the UPRM facility. Additionally, sensors to provide environmental data of the experiment, were interfaced such that all of this data which had to be manually measured could be provided automatically.

The user interface provided through the Android application adds function to the system by allowing more real state to show and organize experiment data. It provides a base for extensions that can automate procedures normally performed by experimenters such as scaling the results to their life like components.

15. Future Work

The next step in the development on our system is to implement it where the current weighting scale in the Civil Engineering Department at UPRM. For the implementation of the system, the current noise cancellation method is software averaging and hardware averaging from the ADC, however hardware methods of cancelation may be explored to be able to obtain more reliable values. The method for obtaining the wind direction uses a potentiometer that causes friction to movement when other wind vanes use a magnetic implementation such that the friction is very close to none which could be adapted to our implementation. The wind speed calculation is done with an anemometer that does not have a reference or a confirmation value and for such reason another system to obtain wind speed could be implemented. Using two barometric pressure sensors could provide a differential measurement of pressure that allows the computation of the speed of the wind inside the tunnel.

In software, the data provided is just the raw data obtained from the measurements of the procedure. There are additional parameters and procedures used for the data experimented such as averaging, determining outliers, graphing the data, comparison of experiments, visualization of data, experiment scaling, and others that can be implemented. The application developed in Android allows the use of a component that has the ability to expand its computing power for analysis. Hence the system software can be expanded to provide more features to the experimenters.

Appendices

A. References

[1] NASA.gov (2010) What are wind tunnels? [Online] http://www.nasa.gov/audience/forstudents/k-4/stories/what-are-wind-tunnels-k4.html

[2] University of Maryland AeroSpace Engineering (2010), Glenn L Martin Wind Tunnel [Online] http://windvane.umd.edu/

[3] University of Southhampton (2013) University of Southhampton Wind Tunnels [Online], http://www.southampton.ac.uk/windtunnels

[4] Departamento de Ingenieria Civil y Agrimensura, (2013) Laboratorio de Tunel de Viento, [Online], http://ingenieria.uprm.edu/inci/mod/page/view.php?id=33

[5] Electric Power Around The World, [Online], http://www.kropla.com/electric2.htm

[6] Baud, Wikipedia, 2013 [online], http://en.wikipedia.org/wiki/Baud

[7] Datasheet Storage, Aerobal, 2013, online, https://www.dropbox.com/sh/4dh73m9yoqclbrs/7CbMi-WmmH

[8] NASA Wright Brothers History, http://wright.nasa.gov/airplane/tunnel.html

[9] National Museum Data Archives, http://www.nationalmuseum.af.mil/shared/media/photodb/photos/051019-F-1234P-003.jpg

[10] ASME.org Datasheets, https://www.asme.org/getmedia/5fe3daaf-75a3-4eb8-b5a7-da95fdc2413e/183-Wright-Field-5-Foot-Wind-Tunnel.aspx

[11] Volvo Club Wind Tunnels, http://drivingspirit.com/storage/2012/05/volvo-s60-tta-wind-tunnel.jpg

[12] NASA Wind Tunnel Images, http://www.windtunnels.arc.nasa.gov/pics/12FT/12t2

[13] http://www.rcgroups.com/forums/showthread.php?t=650962

[14] http://www.micromo.com/motor-calculations.aspx

University of Puerto Rico at Mayagüez

Mayagüez, Puerto Rico

Department of Electrical and Computer Engineering

AeroBal: Users Guide

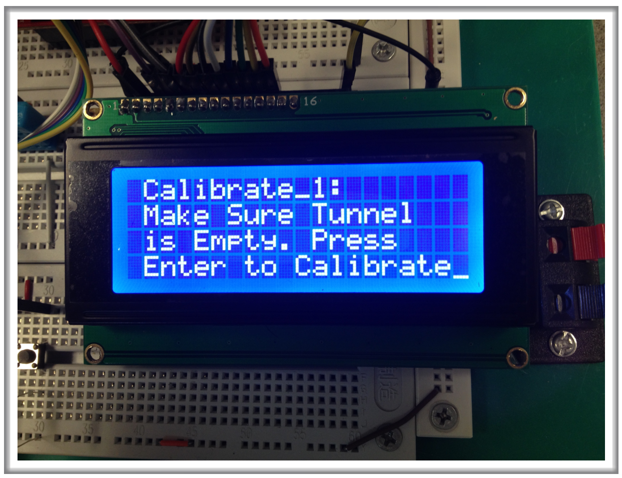
Users Guide for Final Report of a project for ICOM 5217: Microprocessor Interfacing

by

Anthony Llanos - Leader
Jesús Luzón
Juan Lebrón
Jean C. Méndez

For: Dr. M. Jiménez
Course: ICOM 5217, Section 098
Date: December 9, 2013

To start operating the wind tunnel, the user needs only to press the Enter button to start the sequence.



Once calibration has started the user must confirm that there are no objects placed on balance and tunnel. Only the base of the object must be placed.

After confirming, the sequence starts in which the user must wait until the software can store referential values.

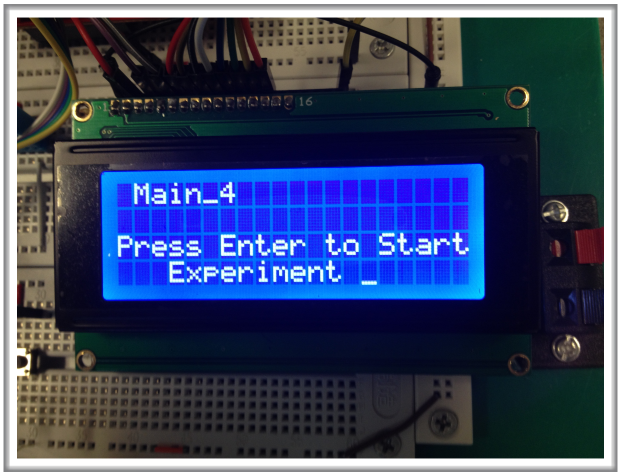
Once initial values without fan have been stored the user must close the tunnel such that the air can flow correctly.

The software will then read referential values which will use for deltas in force calculation of drag and lift.

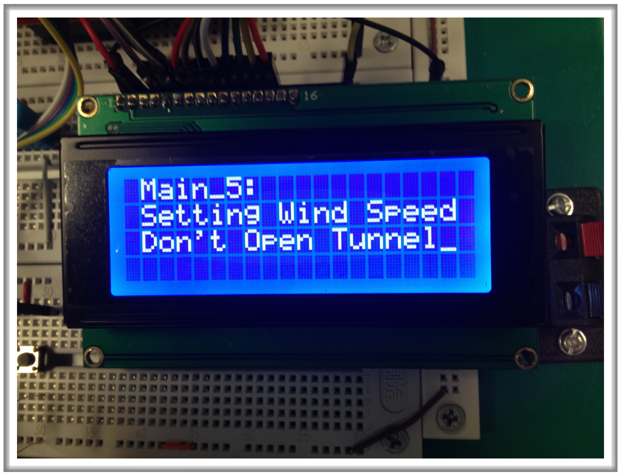
Here these values are being calculated and stored.

Now the object to be measured is placed inside the tunnel.

Initial Values for object and base are stored.

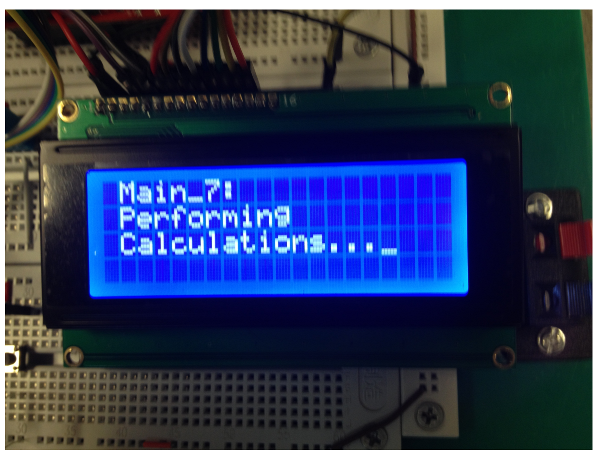


Once all referential inital values for base and base with object are obtained, then the user presses enter which will start the tunnel and the measurements.



Before starting experiment however, the speed must be calibrated by software

After the speed is reached to desired one, the software will then start to reflect those results in the screen.



After obtaning data, the system will calculate the resulting forces and display them to the user. Pressing enter will return the user to the starting sequence.

To configure the software before experiments, the user need only press the Menu button such that parameters for experiment can be entered.

Information will be stored permanently on the Tablet version, and temporarily on the MCU version.

This is the first version of the User Guide.

B. System Specifications

In this section we provide some of the specifications of all of the major components in **AeroBal**. They are included an interfaced in a strategic way to provide the user the best possible experience when conducting research on their various testing models.

B.1 MCU: Tiva C Series TM4C123G

Our selected MCU is the TM4C123G from Texas Instruments, widely known as the Tiva C Series launchpad. Our two biggest reasons for selecting it were the number of available pins (40) and processing power, both at a low price ($12.99). Figure 8 shows an image of our MCU.

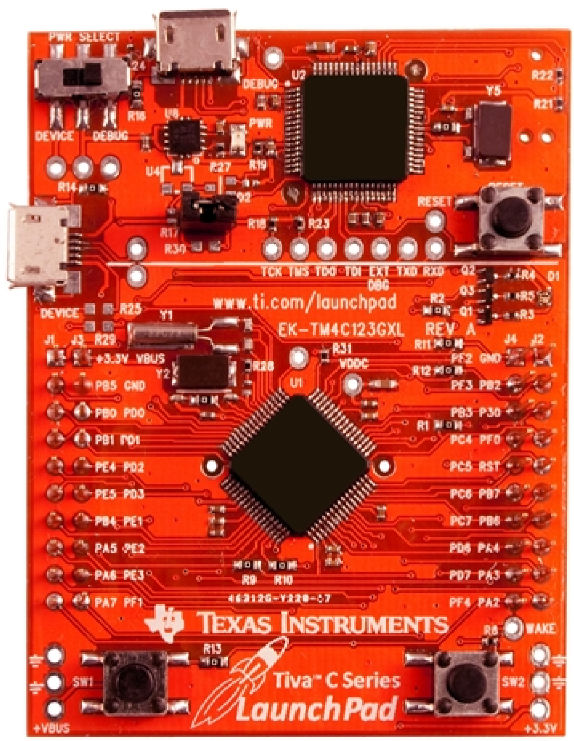


Figure 17: The TM4C123G Tiva C Series Launchpad

Some of the notable features of our the Tiva are the following:

- ARM Cortex-M4F processor core.
- 256 KB flash memory and 2 KB of EEPROM.
- 8 UART Ports, 4 SSI, and 4 I2C Ports.
- 2 PWM modules.
- 2 12-bit ADC modules.
- 1 3.3 V Voltage Source.
- 1 5 V Voltage Source.

Given our large amount of interfaced sensors and devices this microcontroller provided us with the ability to interface multiple components at the same time. The ARM Cortex processor core has also given us great flexibility for our large number of computations with all the data that we process through the system.

B.2 Strain Gauges

The strain gauges are the most important part of the system besides the Tiva C. They provide the user the data of how much force is being exerted to the testing model. Naturally, these forces will vary depending on the aerodynamic characteristics of the object being tested.

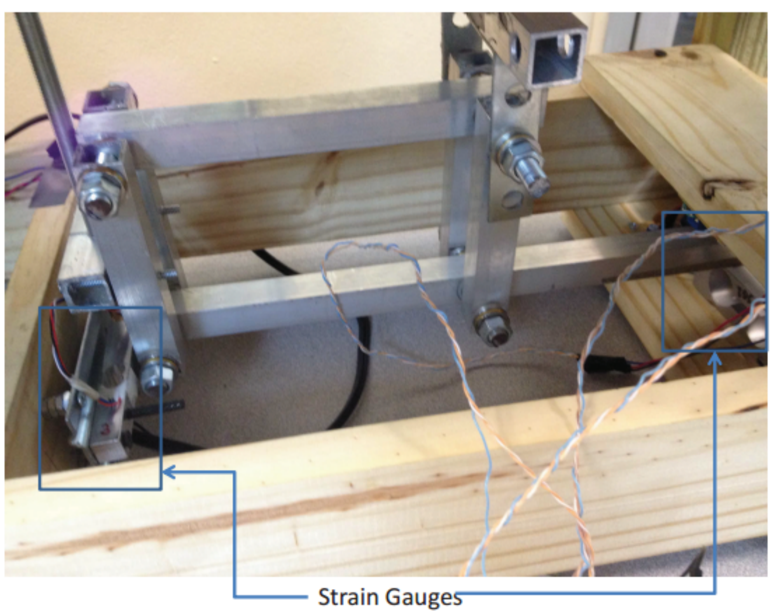


Figure 18: Strain Gauges inside the tunnel

For our system, we selected gauges that can measure up to 20 Kg of forces. These are surplus to the requirements and should provide a user testing small objects (such as toy cars) more than enough flexibility to test their devices.

These strain gauges were placed strategically to measure 3 axis: X, Y, and Z. The X and Y axis require two strain gauges, while the Z axis only requires one.

The gauges require 5 V each, and their data is read using the ADC in the Tiva. Initially, the signal provided is not high enough. Therefore, they go through an amplification process using INA 129 Operational Amplifiers from Texas Instruments. These amplifiers are extremely cheap, and make the strain gauges work fine after building a small and simple circuit amplifying their signals.

B.3 Bluetooth Connection

Bluetooth connection to external devices, such as cell phones and tablets, is a feature that users should find more than handy. Being able to see the data they want through a wireless connection using their tablets or cell phones is one of the big steps we wanted to bring the user more comfort.

For AeroBal, we used the EGBT Bluetooth Serial Port Module. It has UART Connectivity and programmable baud rate. The main idea of this bluetooth module was to allow wireless connectivity from the embedded system to a tablet or cell phone with the Android application.

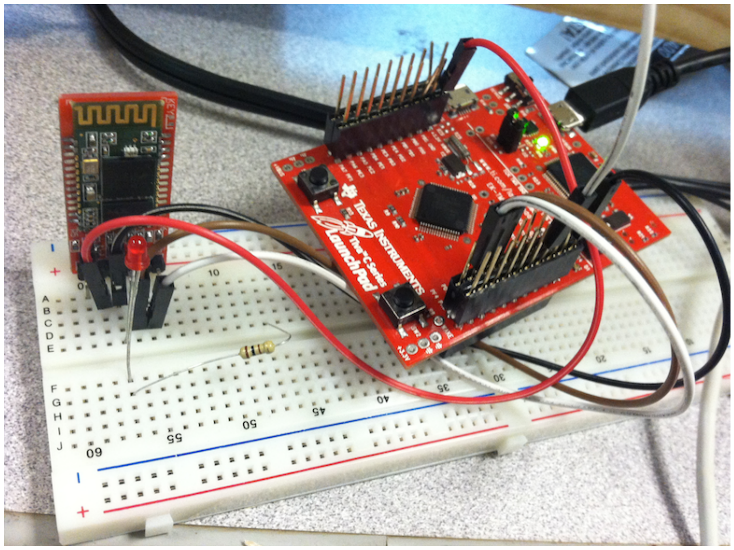


Figure 19: The HC-05 (left) interfaced with the Tiva C (right)

Some of the main hardware features in the HC-05 are:

- -80dBm sensitivity
- Low Power 1.8V Operation
- UART Interface with programmable baud rate
- Integrated antenna

The integrated antenna has also eased us of working with that problem as well.

B.4 Temperature and Humidity Sensor

The pressure and humidity sensors are included to give more useful data to the users about the conditions in the current environment. In our system, we used the DHT11. It connects to an 8-bit microcontroller and delivers high-precision measurements. All of this in a very cost-effective package.

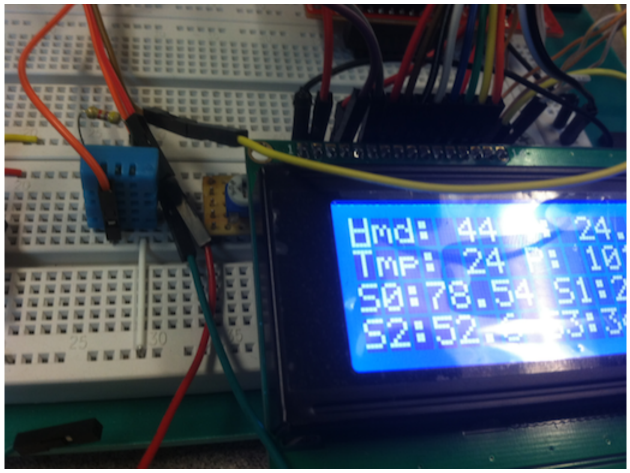


Figure 20: DHT11 sensor (blue) interfaced with Tiva C and LCD Screen

The DHT11 runs with a voltage of 5V using the Tiva C's integrated 5V source.

B.5 LCD Screen

A good LCD screen was a vital part of our system to provide output of experiments. Although an Android Application is a great feature for the user, an LCD screen provides a hardware interface for the application. For our system we selected a C2004A LCD screen. It is identical to the Hitachi HD44780 screen used in the experiments. However, it has a bright, blue screen and allows more lines to be programmed. A library was created to interface with it quickly.

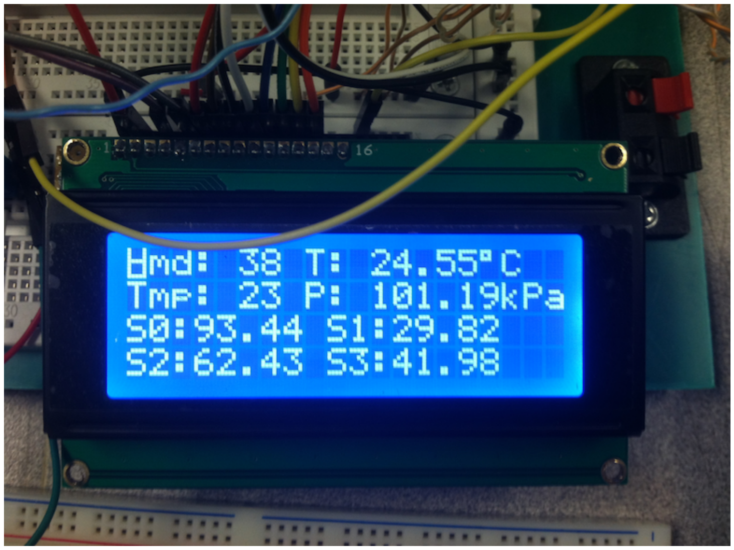


Figure 21: C2004A LCD Screen Interfaced with the Tiva C, DHT11, and Strain Gauges.

B.6 Wind Vane

The Wind vane is used to show the user the direction of the current air flow inside the tunnel. This component was not in our original design, but added later to the project by Dr. Jiménez. The main purpose of the wind vane is to make AeroBal look more and more like a real wind tunnel with all of the major features found not only in other wind tunnels, but also in other places involving high levels of wind, such as farms. The wind vane is a rotating potentiometer moved by the vane tied to the rotating part begin hit by the wind. The wind coming from the fan should move the potentiometer's resistance value which can be measured with the ADC. One big disadvantage was that the smooth potentiometer we found did not behave in a linear way. This caused a need to characterize it in a specific way depending on the values the ADC read on the angles we wanted.

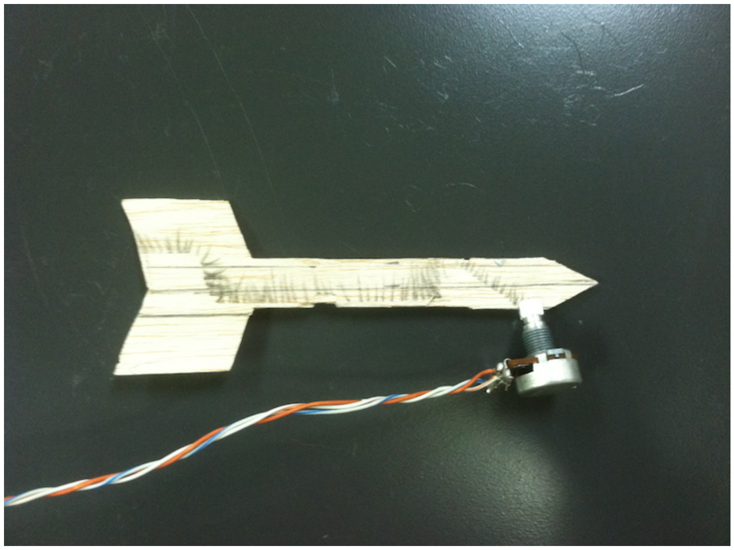


Figure 22: Wind Vane sensor used in AeroBal.

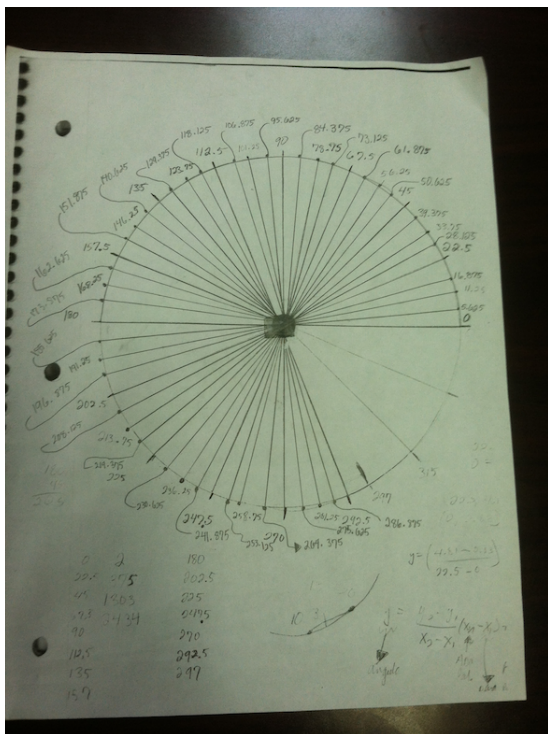


Figure 23: Paper with angles and values used to characterize the sensor.

In the software, the ADC values shown by the Tiva on each of the wanted angles were stored in an array in the sense of a lookup table. Whenever the wind vane is moved, the software searches through the lookup table to find the value in the array nearest to the one provided by the ADC and interpolates using the value in the position previous to the one found in the lookup table.

The wind vane uses a 3.3V source. This is also provided by the Tiva, so interfacing with it was fairly simple as well in a hardware sense. The process of recording all the values inside an array, however, was long and tedious.

B.7 Servo Motor

The servo motor in our system is used to control how much air is going inside the tunnel from the fan we provided. It is expected that the flow of air into the tunnel from the fan is controlled by the voltage applied to the fan. However, the wind tunnel in the Civil Engineering at UPRM uses a different method.
Instead of controlling the voltage applied to the fan, they instead have windows inside the tunnel which they control instead. These windows control the air flow going into the tunnel fairly easily, without the hassle of working with the high voltages of the fan.
In order to emulate the original tunnel as much as possible, we made our own windows. These windows can be controlled by hand, but to make it automatized we use a servo motor to control them instead.
The servo motor we selected is the MG995 Standard Servo Motor. This motor works with a 5V source and can move at a speed of 0.20 seconds per 60 degrees. It's single function is to move the mechanical parts connected to the windows.

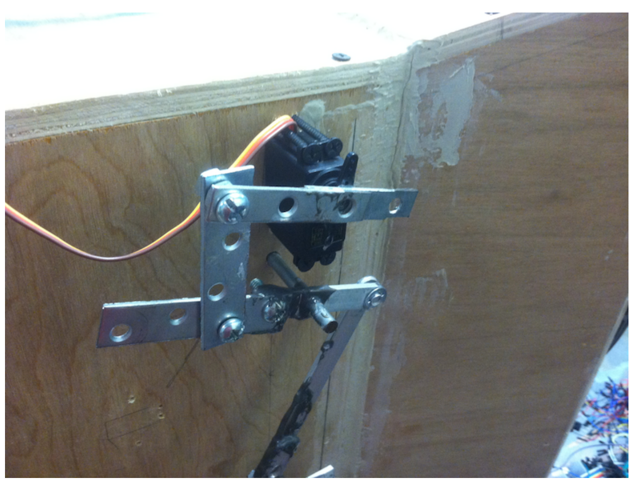


Figure 24: MG995 connected to windows in wind tunnel.

B.8 Android Application

One of the features of AeroBal that most other wind tunnels do not have is a personal Android application used for controlling various aspects of the tunnel and obtaning the data from the experiment. With the Android application, the user may turn the fan on or off as desired, and view the measurements also displayed on the LCD screen in their own device. If extended, the tablet can do more analysis of the system and provide more visual data.

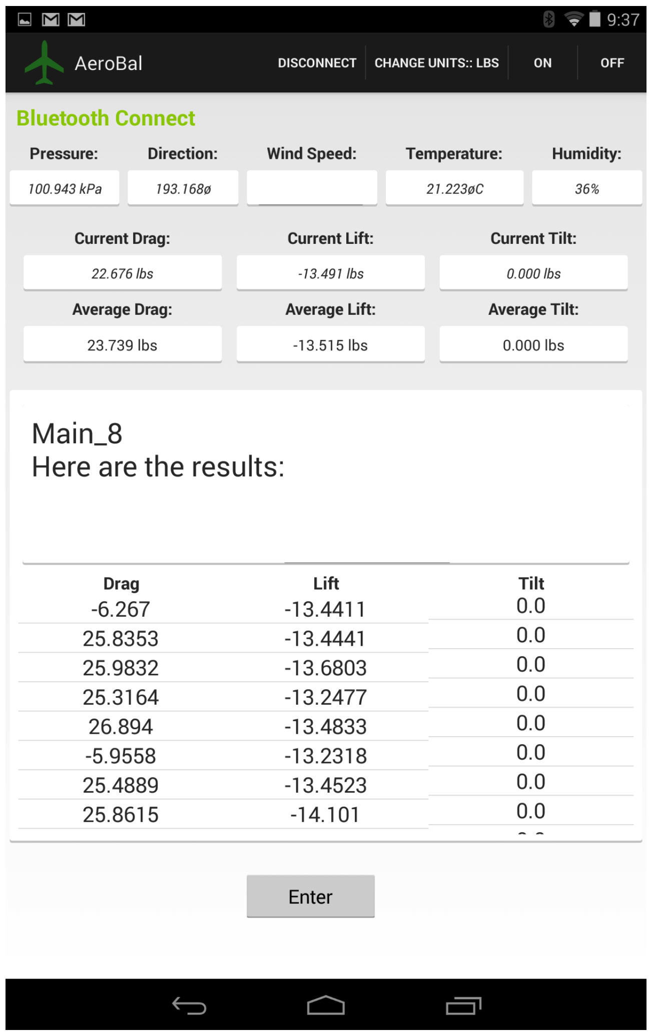


Figure 25: User interface for Android Application.

Although not a necessity, an Android application is a good way to show how up to date we want to make the wind tunnel, using the newest technologies available.

B.9 Miniature Wind Tunnel

A miniature wind tunnel had to be made in order to test our interfaced devices, given that we don't have access to the real-sized wind tunnel in the department of Civil Engineering at UPRM. This miniature tunnel was made entirely of wood, and we integrated a fan into it as the wind source.

Figure 26: First finished version of the wind tunnel, alongside all 4 group members.

The tunnel itself took around 6 weekends to complete plus planning, polishing/conditioning sessions, and not counting the electronic parts added to it later. After the first version was completed, windows were added to the tunnel. These windows are meant to be operated with a servo motor.

Figure 27: Windows inside tunnel; controlled by servo motor.

After we included all the external physical components (windows and strain gauge bases), we moved the wind tunnel to the Microprocessor Interfacing Laboratory (MIL). Since then, all our testing has been conducted inside the lab and other components have been slowly integrated into it.

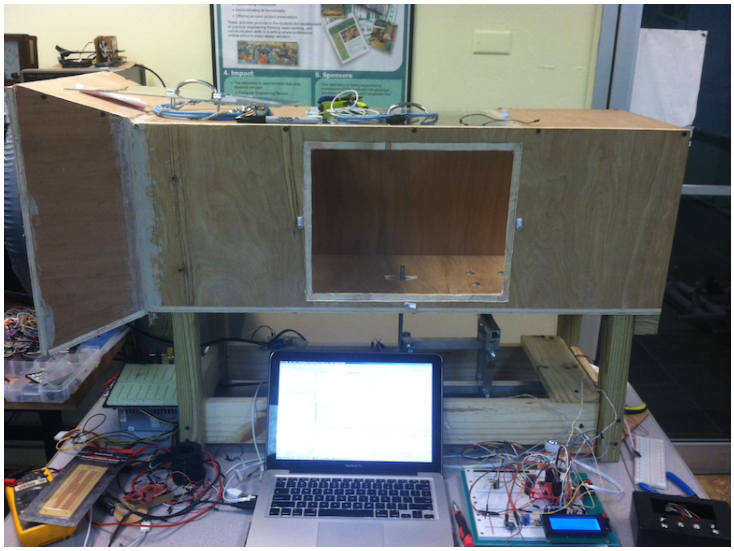
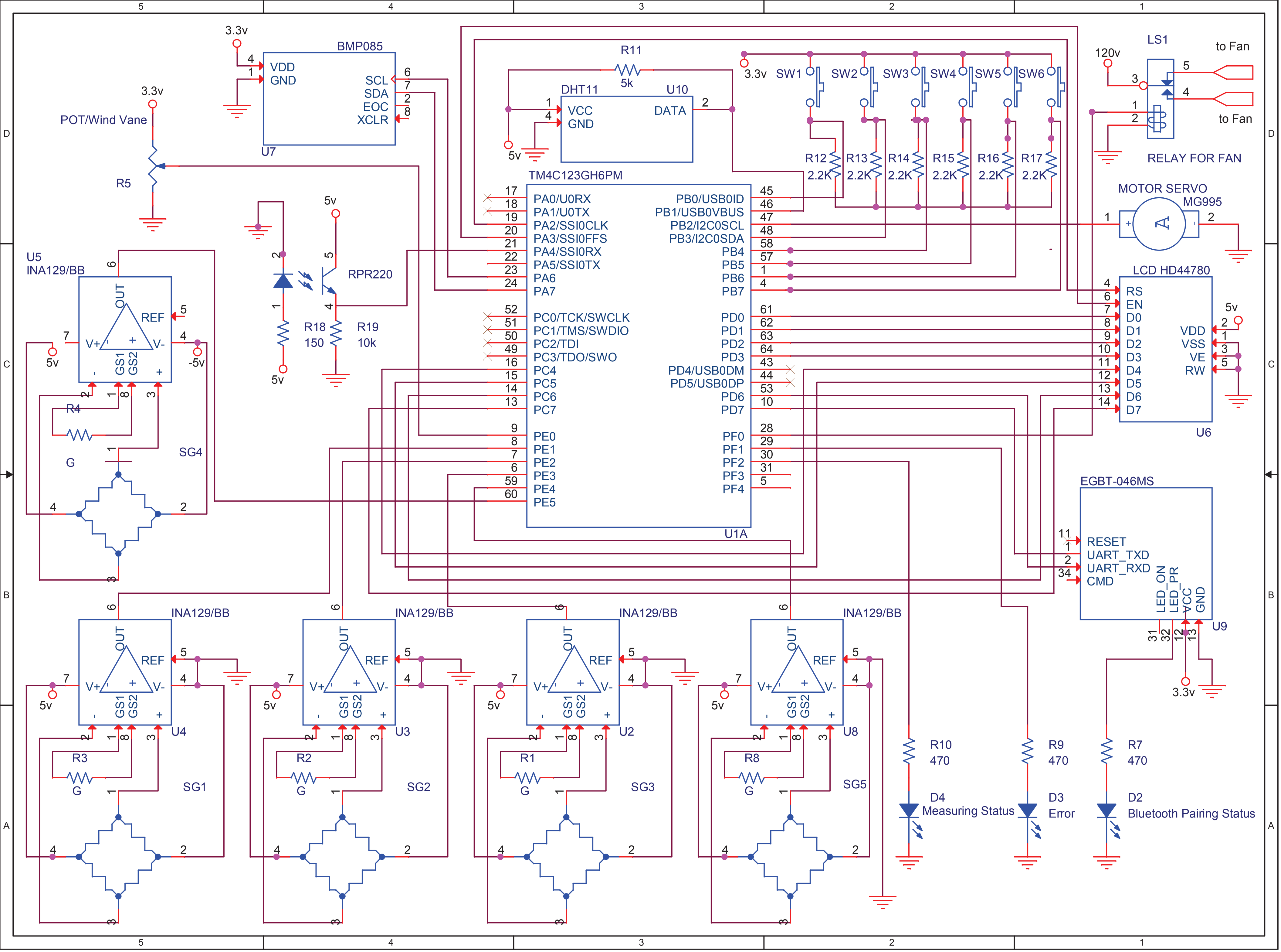


Figure 28: Wind tunnel inside MIL; integration around 65% done by then.

This tunnel is expected to be painted and refined to look presentable to a big crowd by the time the project is 100% done. Although it is not the most important part of the project, it is the showcase for great presentations of it.

C. Schematics

In the following page we present the system schematics in full-page view.


BMP085
U7
VDD
GND
SCL
SDA
EOC
XCLR
POT/Wind Vane
R5
3.3v
5v
-5v
U5
INA129/BB
OUT
REF
V+
V-
GS1
GS2
SG4
R4
G
RPR220
R18
150
R19
10k
R11
5k
DHT11
U10
VCC
GND
DATA
TM4C123GH6PM
U1A
PA0/U0RX
PA1/U0TX
PA2/SSI0CLK
PA3/SSI0FFS
PA4/SSI0RX
PA5/SSI0TX
PA6
PA7
PC0/TCK/SWCLK
PC1/TMS/SWDIO
PC2/TDI
PC3/TDO/SWO
PC4
PC5
PC6
PC7
PE0
PE1
PE2
PE3
PE4
PE5
PB0/USB0ID
PB1/USB0VBUS
PB2/I2C0SCL
PB3/I2C0SDA
PB4
PB5
PB6
PB7
PD0
PD1
PD2
PD3
PD4/USB0DM
PD5/USB0DP
PD6
PD7
PF0
PF1
PF2
PF3
PF4
SW1
SW2
SW3
SW4
SW5
SW6
R12
R13
R14
R15
R16
R17
2.2K
LS1
120v
to Fan
RELAY FOR FAN
MOTOR SERVO
MG995
LCD HD44780
U6
RS
EN
D0
D1
D2
D3
D4
D5
D6
D7
VDD
VSS
VE
RW
EGBT-046MS
U9
RESET
UART_TXD
UART_RXD
CMD
LED_ON
LED_PR
VCC
GND
INA129/BB
U4
U3
U2
U8
SG1
SG2
SG3
SG5
R3
R2
R1
R8
R10
470
R9
470
R7
470
D4
Measuring Status
D3
Error
D2
Bluetooth Pairing Status

D. Component Layout

In the following page we show an image of our MCU and the interfaced components at the time of writing this report. The components are named.

componentlayout

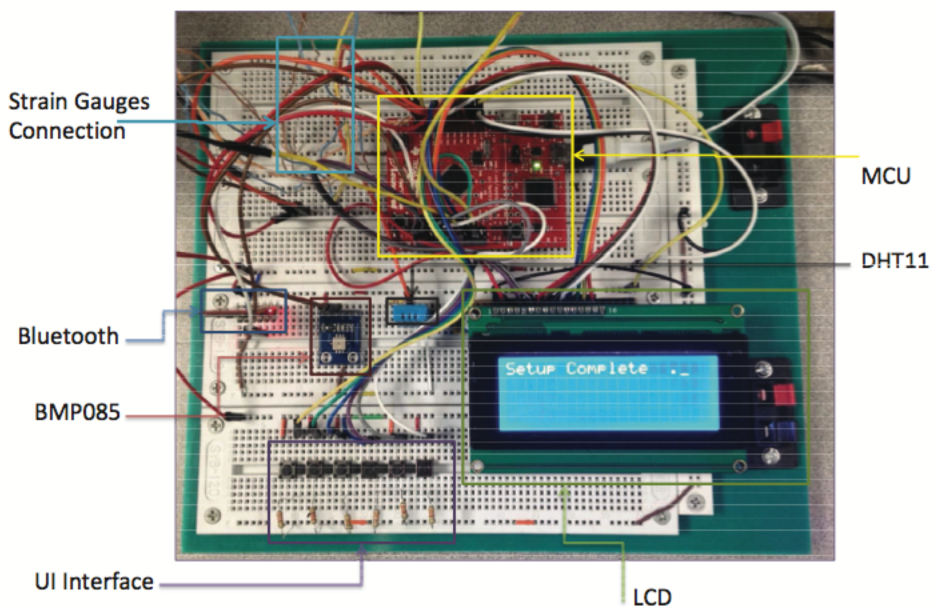


Figure 29: Component Layout of the Aerobal System

E. Bill Of Materials

Listed below are the materials that were and will be used for both the electrical and mechanical components of the tunnel and balance.

Materials	Vendor	Quantity	Price	Total Price
Electrical Materials				
Micro Controller - Tiva Launchpad	Texas Instruments	1	12.99	12.99
Instrumentation Amplifier INA129	Digikey	4	09.18	36.72
Barometric Pressure Sensor BMP085	eBay	1	08.36	08.36
LCD 20x4 - HD44780	eBay	1	09.50	09.50
2 Channel Relay Module	eBay	1	04.74	04.74
Humidity and Pressure Sensor DHT11	eBay	1	03.25	03.25
Load Cell 30lbs	eBay	4	18.79	75.16
Bluetooth 2.0 Module - EGBT-046MS	eBay	1	10.78	10.78
Cables	eBay	1	06.80	06.80
Fan	eBay	1	40.65	40.65
PC Power Supply	eBay	1	21.00	21.00
Resistor Package	eBay	1	10.00	10.00
LED	Sparkfun	5	00.35	01.75
Total				241.70
Balance Materials				
Thrust Bearings	DHGATE	4	07.93	31.72
Pillow Block Bearing 3/4”	eBay	1	09.18	09.18
Balance Materials (Pipes, Tools)	Home Depot	1	55.56	55.56
Machinery Rent (Drill Bench)	Home Depot	1	50.00	50.00
Total				146.46
Tunnel Prototype Materials				
Wood and Tools	Home Depot	1	68.44	68.44
Materials (Scres, Bolts, Drills)	El Palacio Hardware	1	42.50	42.50
Total				130.44
Project Total				518.60

F. Work Distribution

In the following page we present a table with the work distribution of the team throughout the semester.

	Jesus	Anthony	Juan	Jean
Define Specifications	✗	✗	✗	✗
Introduction				✗
Abstract				✗
System Architecture			✗	
System Conception		✗		
Design Criteria		✗		
Define Hardware and Software Aspects	✗	✗	✗	✗
Block Diagram		✗		
Graphical Aspects		✗	✗	
Expert Opinion		✗		✗
MCU Selection	✗	✗	✗	✗
Project Journal				✗
Cost Analysis	✗	✗	✗	✗
Project Time Table	✗			✗
Work Distribution Table	✗			✗
Parts Selection	✗	✗	✗	✗
Market Description	✗			
Proofreading				✗
Design	✗	✗	✗	✗
Proposal and Reports	✗	✗	✗	✗
Testbench Prototype			✗	✗
Tilt Sensor	✗	✗		
Control System				✗
User Interface - Computer	✗	✗		
User Interface - Micro			✗	✗
Communication Protocol	✗			✗
Soldering			✗	✗
Motor Systems			✗	✗
LCD		✗	✗	
User Interface Buttons	✗			✗
Implement Control System				✗
Motor/Control System Integration		✗		✗
Communication Integration	✗		✗	