Instrumentation of the Atwater Kent Wind Turbine

A Major Qualifying Project Report
Submitted to the Faculty
of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in Electrical and Computer Engineering
by

____

Thomas Izzo

____

Carmel Kozlov

____

Steven Fanara

Approved:

____

Professor McNeill, Advisor

____

Professor Emanuel, Co-Advisor

April 28th 2011

Abstract

This report represents the design and implementation of a system used with the Wind Turbine on the Atwater Kent building at Worcester Polytechnic Institute. The system measures the current and voltage outputs from the Wind Turbine, performs calculations, and displays this information graphically online. Finally, a charging circuit was also designed to harvest the energy from the Wind Turbine and charge a battery with it.

Executive Summary

The development of alternative energy has been very important in the past few decades, as new technologies are being developed and incorporated with the hope to eventually replace fuel sources. Worcester Polytechnic Institute (WPI) has been involved in the research and development of alternative energy, and encouraging their students to be engage in alternative energy related projects. In 2008 a SWIFT brand wind turbine was donated by National Grid to the Electrical and Engineering department at WPI, in order to create more clean wind energy projects within the department.

Prior to this project, the energy produced by the SWIFT wind turbine was configured to harness the wind energy and distribute the energy produced back to the power infrastructure of the Atwater Kent building at WPI. After examining the configuration of the SWIFT wind turbine, three objectives were set out for this project. The first objective was to build a measuring circuit that would provide feedback regarding the voltage and current of the wind turbine. The second objective was to build an online portal that would interface with the measuring circuit and display real-time data of the wind turbine on a website. The third objective was to build a battery charging circuit that would use the low-power output from the wind turbine to charge a 12V battery.

The three design objectives set forth in the beginning of the project were all met. The measuring circuit was designed and built; the online portal was interfaced with the circuit to display the real-time data from the wind turbine; and the battery charging circuit was designed and built.

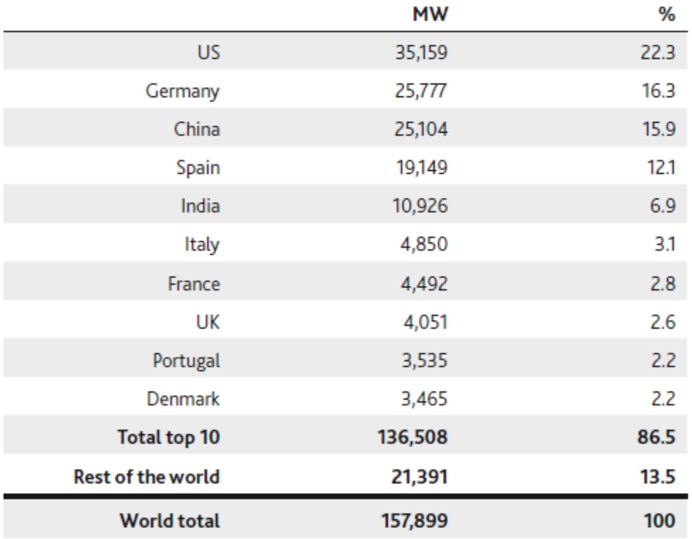
Contents

1.0 Introduction

The development of alternative energy has been very important in the past few decades, as new technologies are being developed and incorporated with the hope to eventually replace fuel sources. Alternative energy refers to sources of energy which do not use up natural resources or harm the environment.

Wind turbines are rotary devices which generate energy by converting the kinetic energy of the wind to electrical energy. The rotating blades of the turbine spin a shaft connected to a generator which then generates electricity. Thus, the generation of electricity relies on the amount of wind that is present.

Currently, many countries produce large amounts of electrical power from wind. From 1990-2002, wind has been the fastest-growing power source worldwide on a percentage basis, with an annual average growth rate exceeding 30% (Wind Industry Statistics, 2009).



	MW	%
US	35,159	22.3
Germany	25,777	16.3
China	25,104	15.9
Spain	19,149	12.1
India	10,926	6.9
Italy	4,850	3.1
France	4,492	2.8
UK	4,051	2.6
Portugal	3,535	2.2
Denmark	3,465	2.2
Total top 10	**136,508**	**86.5**
Rest of the world	**21,391**	**13.5**
World total	**157,899**	**100**

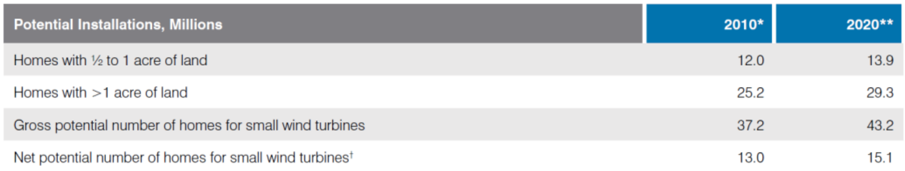
Figure 1 - Top 10 Cumulative Wind Energy Capacity 2009 (EWEA)

The European Wind Energy Association reported that in 2009 the US was the global leader in cumulative wind energy capacity at 22.3%, as can be seen in Figure 1, and that the growth rate of the wind turbine industry was 39.3% in the US at the end of 2009 (World Wind Energy Report 2009).

However, these statistics cover the larger wind turbines, which produce large amounts of power and are used to supply the national grid with energy. In order to evaluate the possibility of successfully using a smaller wind turbine as a secondary source of power, or as a household emergency power source, the statistics for small wind turbines must be examined.

Despite the economic downturn that the US is currently experiencing, the American Wind Energy Association (AWEA) reported that the market for small wind turbines (rated at 100 kilowatts or less) grew 15% in 2009. They further explain that the growth equates to around 10,000 new units installed, with a total capacity of 100MW across the countries (AWEA Small Wind Turbine Global Market Survey 2009).

AWEA explains that there is a niche market that would have the potential for smaller wind turbines installations, as in shown in Figure 2.



Potential Installations, Millions	2010*	2020**
Homes with ½ to 1 acre of land	12.0	13.9
Homes with >1 acre of land	25.2	29.3
Gross potential number of homes for small wind turbines	37.2	43.2
Net potential number of homes for small wind turbines†	13.0	15.1

Figure 2 - Potential Installations of Small Wind Turbines in the United States (AWEA Small Wind Turbine Global Market Survey 2009)

The new potential number of homes for small wind turbine installations depends on the size of the home and lot. The meet the electrical needs of a typical US home, AWEA reports that for a house with "medium wind regime" the wind turbine must have a rotor diameter of 16 to 25 feet and be installed on a tower 60 to 150 feet tall. AWEA explains that out of the net potential number of homes for small wind turbines presented in the table, 35% have a sufficient wind resource, that is, the wind power class is at least a 2 as rated by US Department of Energy (average of 9.8 mp/h at 10m height).

In 2008 a SWIFT wind turbine was donated by National Grid to the Electrical and Computer Engineering department at Worcester Polytechnic Institute. The current installation of the wind turbine is configured to harvest the wind energy and to supply it back to the electrical grid. Unfortunately, the wind turbine was not being used for any particular application, and was only supplying energy back to the grid. Due to the potential of smaller wind turbines in the market, it was important to take advantage of this alternative energy source, which was the motivation for this project.

In the following subsection - 1.1 Project Objectives, the project objectives are laid out and explained. Furthermore, Chapter 2 describes pertinent background information regarding the current installation of the SWIFT wind turbine. In Chapter 3 the initial design concepts and problems are defined. Finally, Chapter 4 includes that detailed description of the final system design including problems that were encountered. In Chapter 5, the results of the entire system are described, and lastly, Chapter 6 reviews the conclusion and outlines any further work that could be done in the future.

1.1 Project Objectives

This project was concerned with finding and implementing an alternative application of the wind turbine that is currently placed on the roof of Atwater Kent at Worcester Polytechnic Institute. Throughout the first few weeks of the projects, different options were discussed and considered. For example, one option was to focus heavily on a home power backup system, focusing on providing a home with a source of power in the state of an emergency using the wind turbine to charge a battery. Another option was to build an inverter to convert the DC coming out of the wind turbine to AC. After considering these different options, three main objectives were defined. These objectives were defined as they matched the different skill set of the different project members, and were concluded to be achievable in the time frame that was available.

The first goal was to build a system that would measure the voltage and current produced by the turbine. The data would then be used to analyze the performance of the wind turbine. Due to the high voltage that can be produced by the wind turbine, it was important to design a system that would measure the voltage and current with galvanic isolation to ensure the protection of all of the instrumentation involved.

The second goal of this project was to create a battery charging circuit. This involved designing and building a DC-to-DC converter that would not only charge the battery but also report its state of charge. Additionally, consideration to the extra power produced by the wind turbine beyond the capacity of the battery was made.

Finally, the third goal of this project was to create an online portal that would display the real time data measured from the wind turbine as well as data from a weather station. Using the voltage and current measurements, additional inferred data could also be displayed online, as well as the state of charge of the battery. The final measurements to be displayed on the on the online portal would have to have percent in reading error of less than or equal to 1%.

2.0 Background

In this chapter background information regarding the Atwater Kent wind turbine is discussed. The current configuration of the SWIFT wind turbine installation is explained in detail. Addition information regarding the specification of the wind turbine is overviewed. Finally, the state of the wind turbine online portal is discussed.

2.1 Current Configuration

The wind turbine donated by National Grid sits on top of the Atwater Kent building in Worcester Polytechnic Institute as seen in Figure 3.

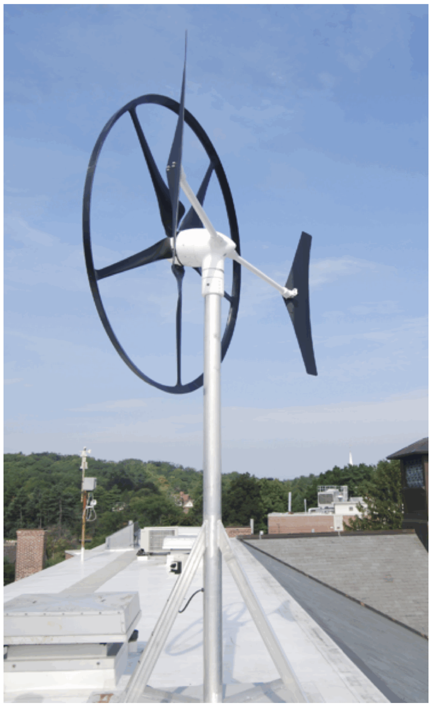


Figure 3 - Wind Turbine on the roof of Atwater Kent Laboratories

The wind turbine mast is 16 feet tall, with a rotor diameter of 7 feet. The block diagram of the wind turbine, presented in Figure 4, shows its current setup in Atwater Kent.

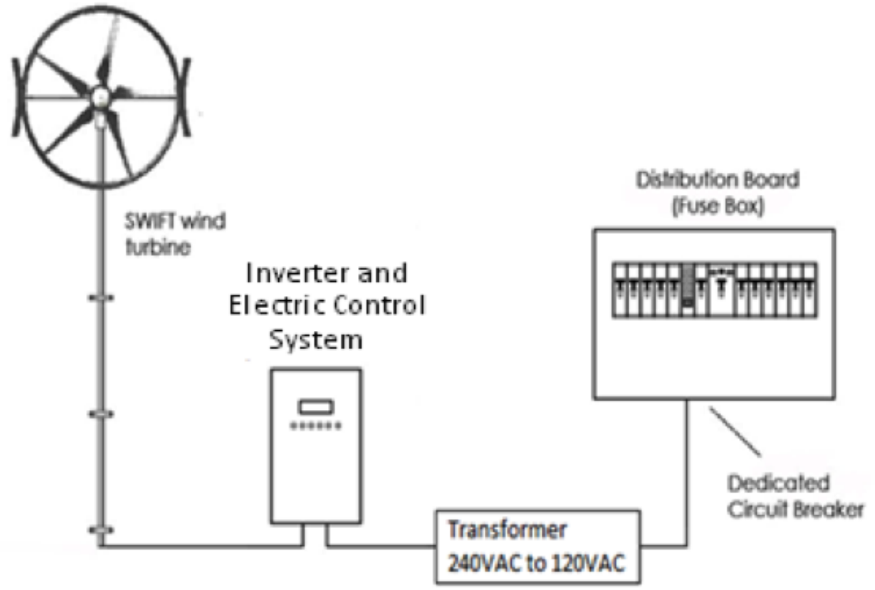

SWIFT wind turbine
Inverter and Electric Control System
Distribution Board (Fuse Box)
Dedicated Circuit Breaker
Transformer 240VAC to 120VAC

Figure 4 - SWIFT Turbine Block Diagram (Adapted from SWIFT Data Sheet)

The turbine generator consists initially of a three phase, 8-pole generator. The signals produced from the three-phase generator are then rectified with a six diode full bridge wave rectifier, producing a DC output signal as shown in Figure 5. This generator is located in the first part of the wind turbine installation, labeled as SWIFT wind turbine in Figure 4.

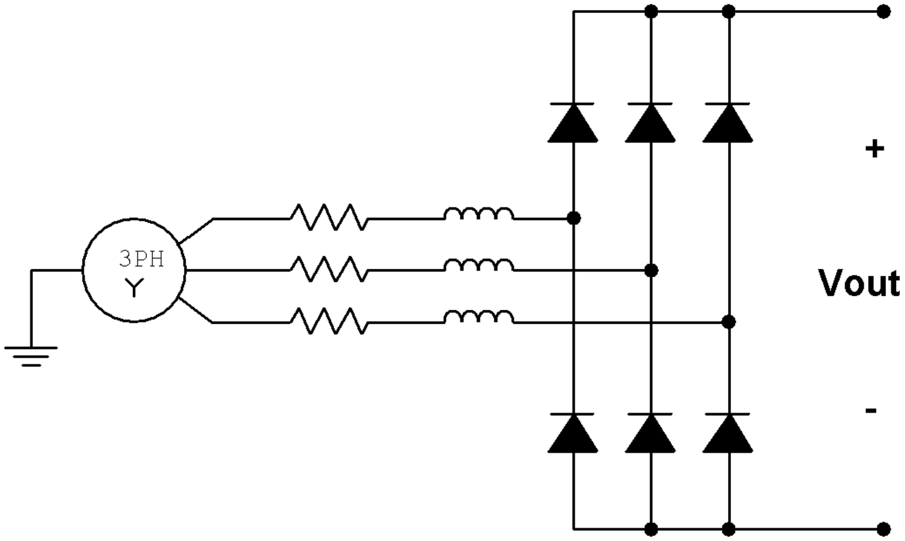


Figure 5 - Six Diode Full Bridge Wave Rectifier

This output signal is sent down into the building from the roof where it arrives initially at the SWIFT inverter and an electronic control system. The SWIFT inverter includes a filter to remove stray noise, and has a fuse for over current and over voltage protection. Next, a DC to AC converter converts the signal from DC to 240V 60Hz.

From the SWIFT inverter the AC signal is then sent to the isolation transformer. This extra step is needed as Atwater Kent runs on 120VAC, not 240VAC. The isolation transformer transforms the 240VAC to 120VAC and then ties the signal to the grid.

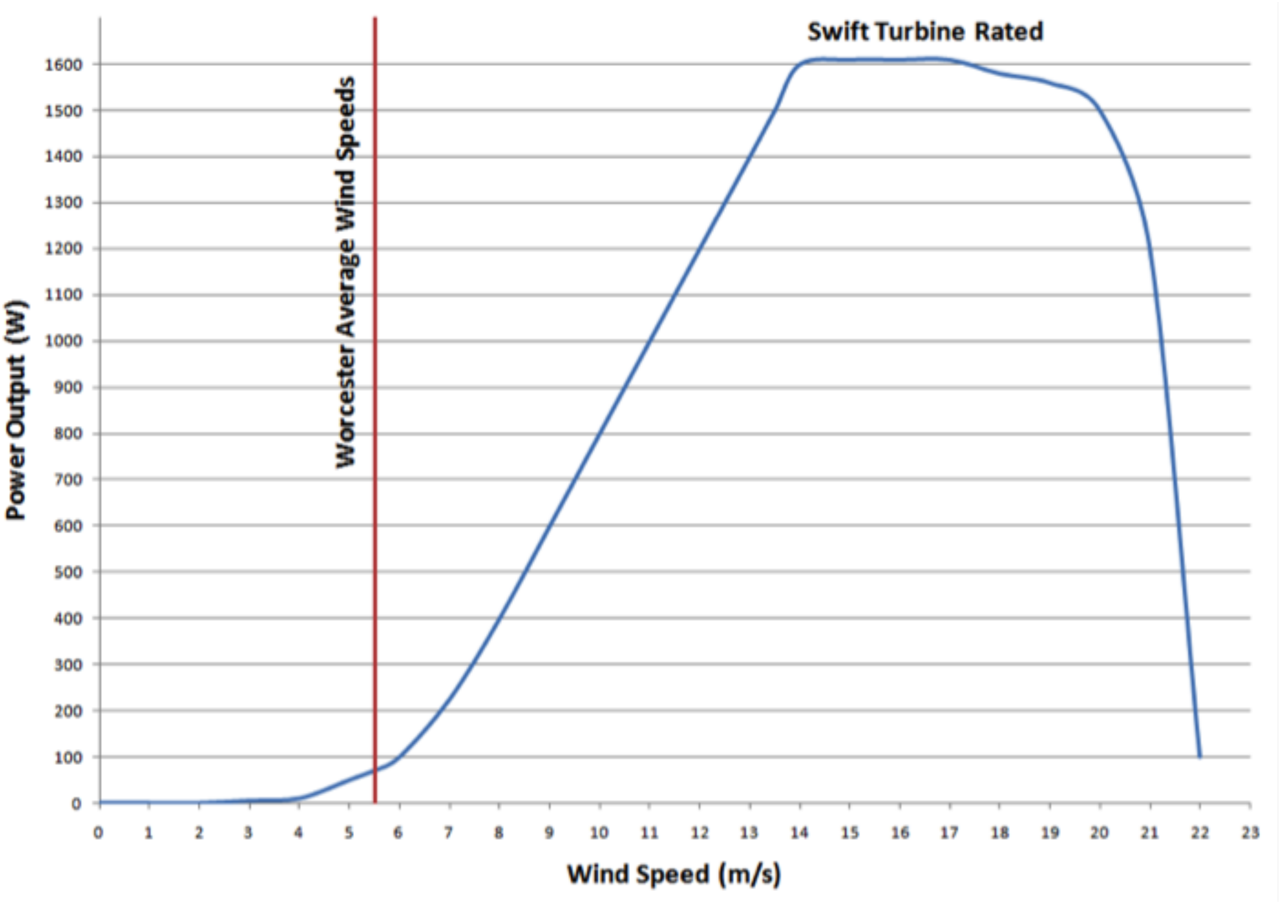


Figure 6 - Power Output of SWIFT Turbine versus Wind Speeds

Finally, Figure 6 shows a graph of the power production of the SWIFT turbine due to the wind speeds. The turbine is able to produce a maximum amount of 1600W at wind speeds between 13m/s (29.1mph) and 17m/s (38mph), at a maximum of 400VDC. Also shown in Figure 6 is the average wind speed of 5.5m/s (12mph) in Worcester (National Weather Service, 2010), indicating that on average the SWIFT turbine will be generating less than 100 Watts.

2.2 Efficiency of Atwater Kent Wind Turbine

Wind turbines aren't able to convert all of the available kinetic energy to electrical energy. Betz' law states that the maximum energy extraction from wind energy is 59.3% of the total kinetic energy available (Spachos, 2003).

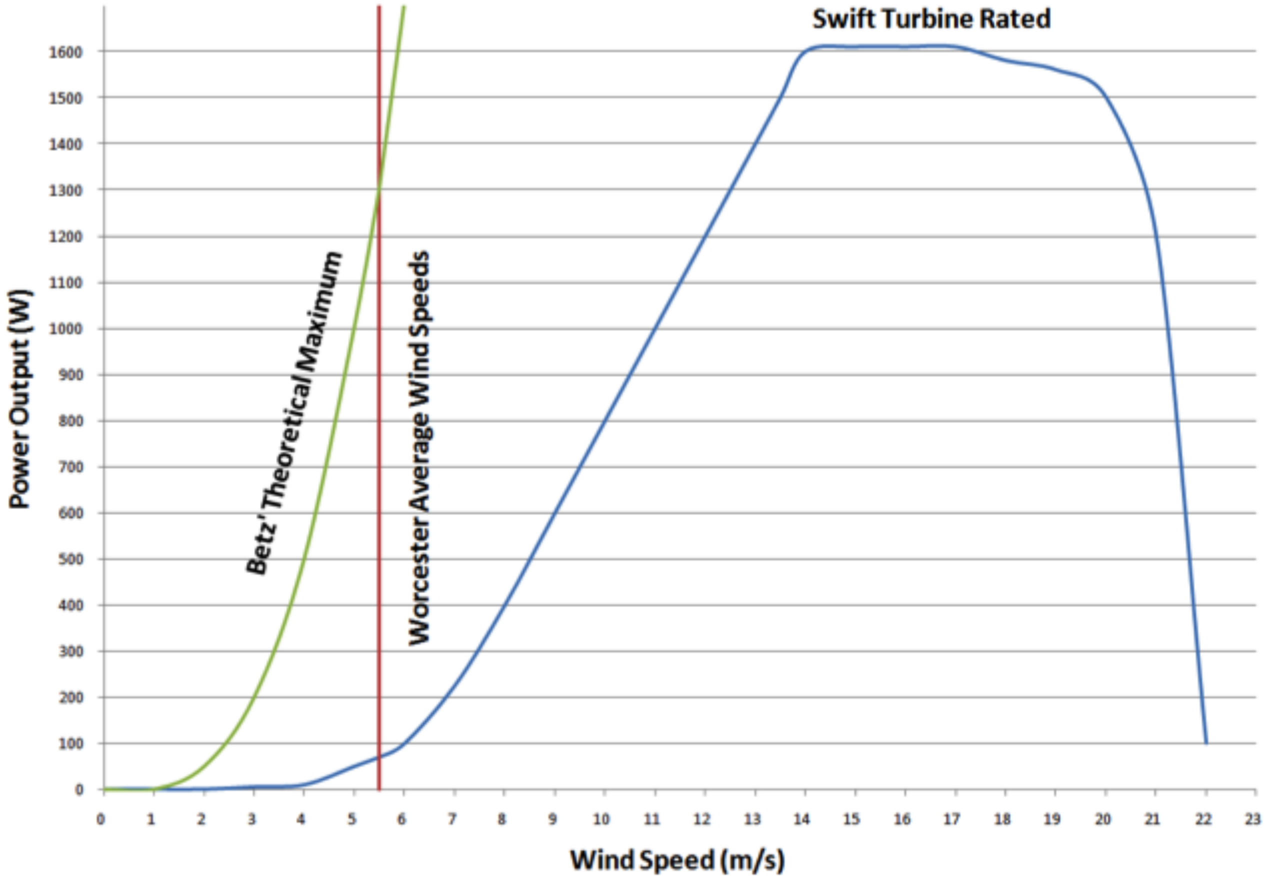


Figure 7 - Betz' Law Theoretical Maximum

Figure 7 shows Betz' law theoretical maximum power extraction calculated for the wind speeds given (green line) and the power output from the SWIFT turbine. Additionally, the red line represents the average wind speed in Worcester, indicating that the average power output expected from the turbine is smaller than the maximum power it can generate. It is apparent from the graph that the SWIFT wind turbine does not reach Betz' Theoretical Maximum. However, most wind turbines' wind efficiency is at only about 40% for high wind speeds (Spachos, 2003). The extra energy loss is due to energy losses in the wind turbine, lost in heat due to friction in between the rotor, shaft and bearings, as well as electrical losses in the conversion of the energy from wind to electrical (Spachos, 2003). This explains why the SWIFT wind turbine does not reach Betz' theoretical maximum.

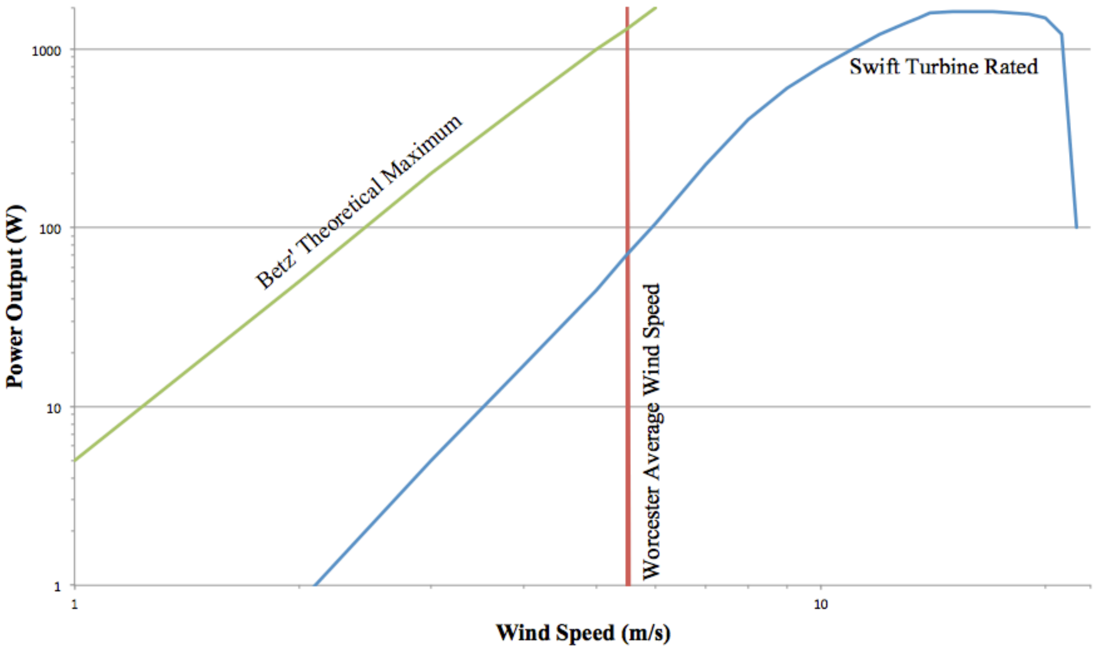


Figure 8 - Log Scale of Betz' Theoretical Maximum

Figure 8 shows the log-log scale of Betz’ Theoretical maximum power output for the SWIFT turbine. It is easier to see the large difference between the theoretical maximum power output and the actual rated power output. It appears that Betz’ theoretical maximum is a magnitude larger than the rated output for the wind turbine. The inefficiency of the wind turbine is caused by different factors. Electrical power losses happen at the rectifier, the inverter and the transformer. Since the wind turbine was donated to the Electrical and Computer Engineering department, there was no way for the project team to access the wind turbine schematics to see where these losses may happen. Schematics that were available through the manufacturer were only indicative of the turbine’s configuration in the building. This was another factor that helped shape the goals of the project – since there was no access to the detailed schematics, and there was no option to open up the wind turbine to inspect the circuits presented there. The project had to focus on the output of the wind turbine, as there was no other way to interact with the wind turbine.

2.3 Weather Instrumentation and Online Portal

One goal of the project is to have an online portal allowing visitors to view pertinent data on the current weather and the operation of the wind turbine. Currently, a Honeywell TE923W as shown in Figure 9 is installed on the roof of Atwater Kent, which was operational, as well as the main focus of the online portal.

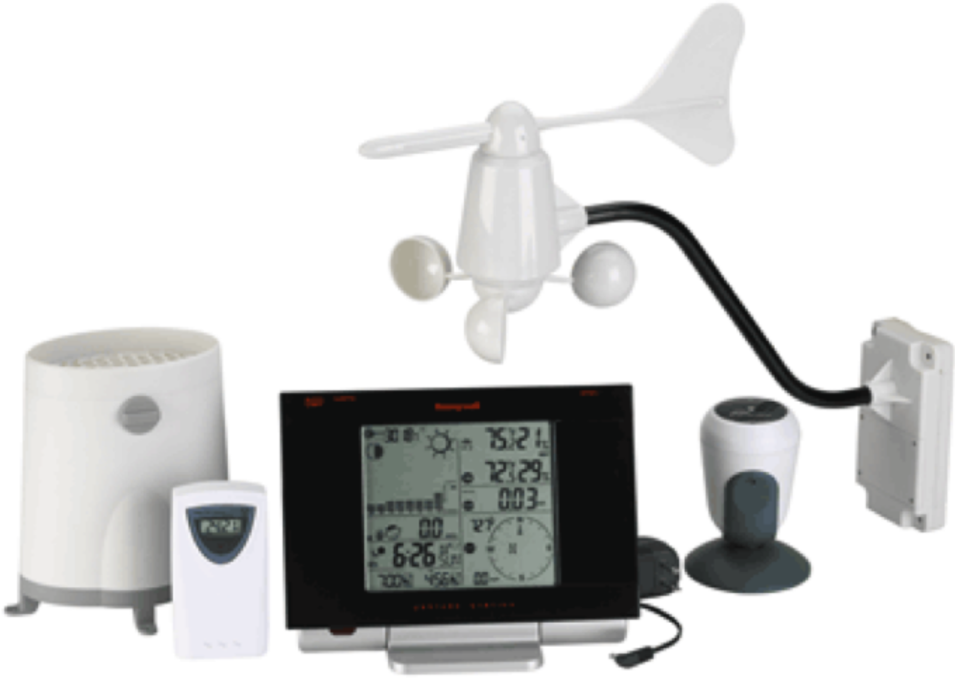
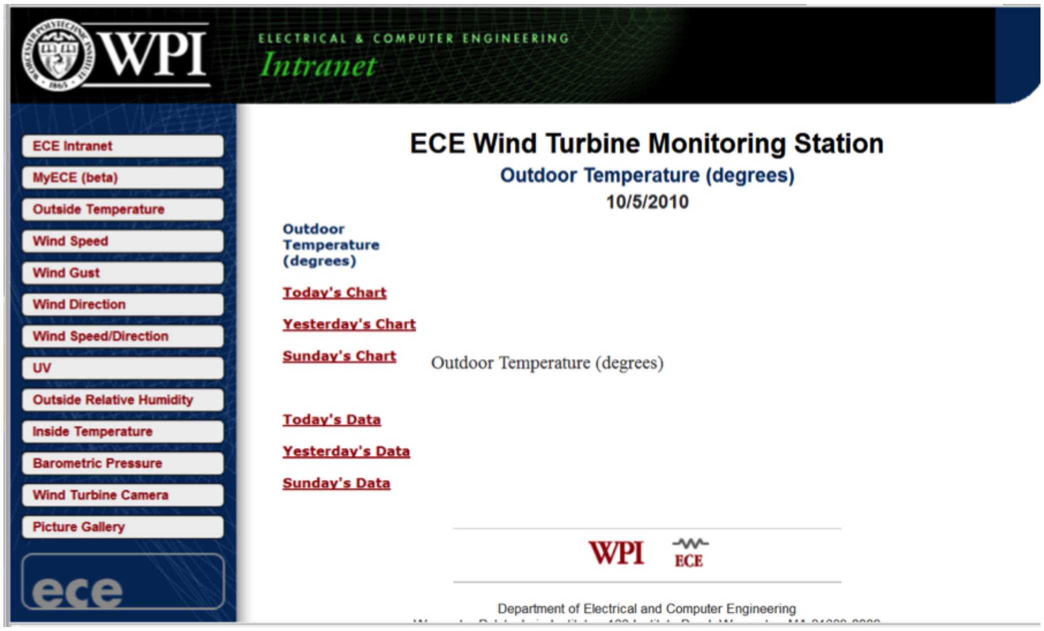


Figure 9 - Honeywell TE932W

The Honeywell TE932W is a wireless home weather station that includes an anemometer, rain gauge meter, UV sensor and a thermo-hygrometer. The base unit sits in a room on the third floor of Atwater Kent. The main unit and the weather station communicate over a wireless link. Additionally, the main unit connects to a computer running Honeywell software.

However, there have been quite a few issues with this weather station. It has not been functional for a few years. There is interference between the main unit and the weather station on the roof, and therefore the weather station isn’t able to send complete data packets. When this happens, a manual reset of the station is required. As the weather station is mounted on the roof of Atwater Kent, this is not easily done. Additionally, the unit tends to stop working in extreme weather conditions, such as when a storm passes by.

In the past, the online portal displayed information directly from the weather station. Figure 10 shows the current online portal. The only functioning page on the online portal is the wind turbine camera.



ECE Wind Turbine Monitoring Station

Outdoor Temperature (degrees)

10/5/2010

Outdoor Temperature (degrees)

Today's Chart

Yesterday's Chart

Sunday's Chart

Outdoor Temperature (degrees)

Today's Data

Yesterday's Data

Sunday's Data

Department of Electrical and Computer Engineering

Figure 10 – Old Online Portal

The other pages load properly, but not do display any information or data. In the past, the computer connected to the main unit of the weather station would import the data from the Honeywell software. This was imported to a Linux machine that would plot the data using Gnuplot and update the portal every few minutes with graphs.

3.0 System Prototyping

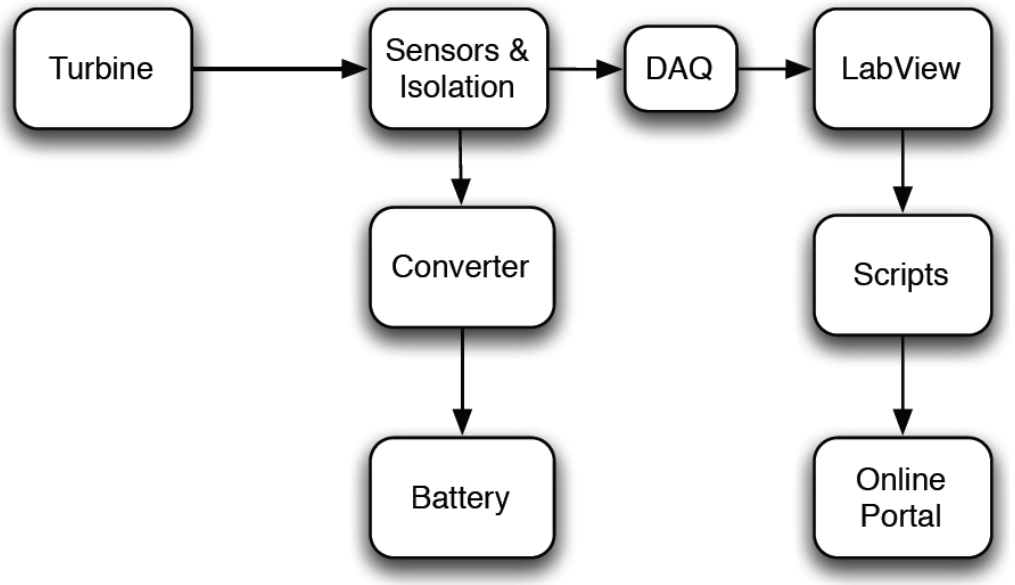


Figure 11 - System Prototype Block Diagram

After background research and defining goals for the project, the initial system block diagram was formed shown in Figure 11. In the subsequent subsections, the initial design process as well as initial problems for each block is described. Section 4 describes the complete design process and the solutions that were used to solve the problems that arose during the design of the system.

3.1 Measuring Outputs from Wind Turbine

As the only parameters available to the group were the maximum voltage rating, as well as the power output versus the wind speeds produced by the wind turbine, it was necessary to be able to measure the important parameters of the wind turbine. By measuring both the voltage and the current from the wind turbine, other parameters could be extracted, such as the power being produced and in future work, the RPM of the turbine as well as wind speeds could be deduced.

Therefore, a sensor circuit was designed to measure the current and voltage being generated by the wind turbine. Since the maximum voltage generated by the wind turbine can reach 400VDC and can produce up to 1600 Watts, it was important to build a circuit that would be galvanically isolated from the wind turbine to ensure the safety of circuit components.

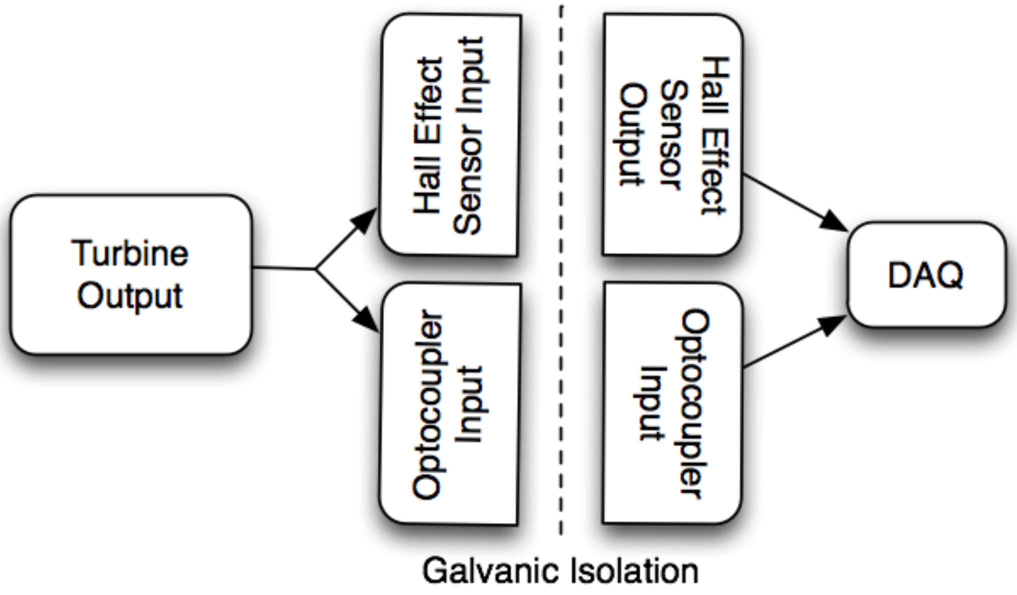


Figure 12 - Block Diagram for Measurement Circuit

Figure 12 shows the initial block diagram for the measurement circuit. To measure the current being produced by the wind turbine, a Hall Effect sensor was used. Hall Effect sensors supply a voltage perpendicular to the magnetic field that they sense. Since a current carrying wire has a magnetic field encircling it, a Hall Effect sensor can be used to measure the strength of the current in the wire.

The voltage measurements were done using an optocoupler, which consists of a LED and two phototransistors, sealed inside a package. As the voltage supplied to the chip increases and decreases the light emitted from the LED changes. These changes are then captured by one of the phototransistic diodes causing a change in the voltage that is allowed to flow through it. The change of voltage through the phototransistic diode can then be measured, giving a reading of voltage from the wind turbine through a correlation between the voltage changes in the LED to the phototransistic diode. The second phototransistic diode is used in the feedback loop to the LED to remove the non-linear properties of the LED. Since the relationship is measured through changes in light there is no physical connection between the components, this in return provides the desired galvanic isolation. All of these components are sealed within the silicone chip to remove any disturbance from ambient light.

3.2 Acquiring the Data using a Data Acquisition System and LabVIEW

The output from the sensors needed to be acquired so that the data gathered could be stored for analysis and processing. After researching options for various data acquisition, the project received a data acquisition system (DAQ) from the Electrical Engineering department at WPI. This DAQ was a NI-USB6008, which has 8 analog inputs and 2 analog outputs. With multiple analog inputs, data from both the Hall Effect Sensor as well as the optocoupler could be gathered at the same time. As this is a National Instrument DAQ, LabVIEW could be used to gather and analyze the data as needed without needing to design an interface between the DAQ and a computer program.

As with any numeric conversion from analog to digital, there is some data loss due to quantization levels and the sampling rate. The NI-USB6008 is able to sample at different frequencies, and has a minimum and maximum voltage range acquiring accuracy, which depends on how the DAQ is configured, as well as the type of software used to interface with it. To be able to calculate the loss of precision in the sampling of the data, it was first necessary to know how often to sample the data.

Once the data is sampled, it is processed through LabVIEW. This is where there was additional loss of data integrity due to the various calculations performed on the data. The current and voltage readings were used to calculate power, as well as the RPM of the turbine.

Finally, it was necessary to find a way to save the data acquired from the sensors, as well as the data that was calculated in LabVIEW. The data would be used for both future analysis as well as creating graphs for the online portal.

3.3 Displaying the Data in Real Time

The online portal allows users to view real time data from the wind turbine online. To do this, data measurements from the wind turbine are needed. Figure 13 shows the block diagram of the process used to generate graphs for the online portal.

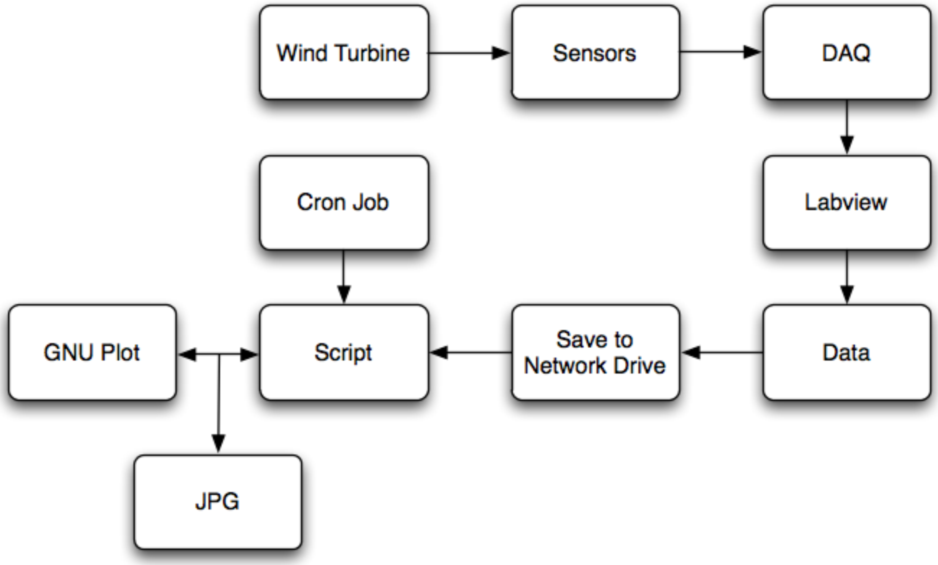


Figure 13 - Online Portal Block Diagram

The data is saved from LabVIEW, and a script runs on the data to parse it. Using a cron job in the UNIX environment, the script can be scheduled to run at any wanted time interval. The script uses Gnuplot in the UNIX environment to graph the data. When the script runs, a new image is created and saved, which is then updated online.

3.4 Battery Charging Circuit Design

The design for the battery charging circuit in this project was specific to the input coming from the wind turbine. The battery charging circuit must be able to potentially handle up to 400VDC at currents of up to 4A. During the initial design of the battery charging circuit, various issues arose that were specific to the design of the circuit as it was interfacing with the turbine. As the voltage coming from the turbine isn't constant, but the output from the charging circuit had to be a constant voltage. In the following subsections the initial characteristics of the circuit are described.

3.4.1 High Voltage Source

Because the turbine outputs a DC voltage of up to 400V, a charging circuit must be designed that can handle those voltages without breaking down. This was accomplished by modifying a standard buck converter design. A Power MOSFET was used instead of a standard MOSFET because the switch needed to be able to withstand voltages up to 400V without breaking down. A switch-mode power supply was chosen over a linear power supply because while there are diodes that can withstand reverse-bias voltages up to 400V, a second power MOSFET will increase the efficiency over that of a diode. The high side MOSFET then needed to be bootstrapped to float the gate signal above the source voltage so that the gate-to-source voltage V_{GS} is always at zero or at +5V.

3.4.2 Variable Voltage Input

As the wind speed varies over time, so does the DC voltage output by the turbine. The charging circuit was designed to be able to handle any DC voltage between 12 and 400V and always output 12V with a voltage ripple of at most 1% of the ideal 12V output. This was accomplished by building in a voltage controlled feedback loop that will increase or decrease the duty of the switching elements if the voltage gets too high or low.

4.0 Detailed System Design

In Chapter 3.0, an initial system block diagram was introduced and the general approach for each block was explained. Figure 14 shows the final system block diagram, and a detailed description of the processes taken to engineer and build the blocks is explained. After the initial problem definition, the different blocks of the system were built and put together over the course of a few months to form the final, working system.

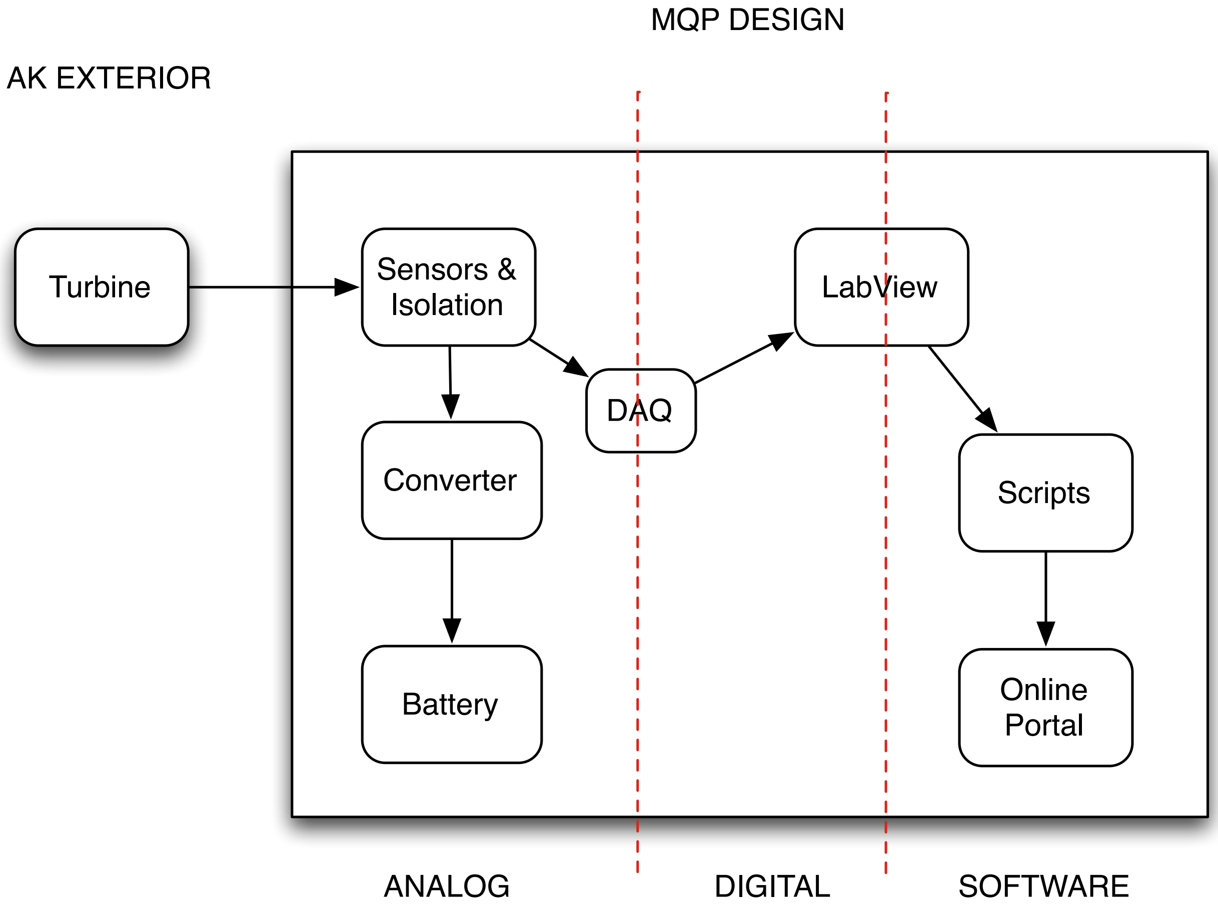


Figure 14 – Final System Block Diagram

Figure 14 shows how different parts of the system cross over different disciplines within Electrical and Computer Engineering. The wind turbine is located in a physical location, and was not a part of the system design, which is why it is placed outside the rectangular block representing the MQP design. The sensors and isolation block, as well as the converter and battery block are in the analog discipline. The DAQ block contains both analog as well as digital components, and lies in between the analog and digital discipline. Using LabVIEW allowed the project to bridge the gap between the digital and software world as will be described in detail in the following subsections. Finally, the scripts and online portal all lay within the software discipline.

4.1 Measuring Outputs from Wind Turbine

As mentioned in Section 3.1, an optocoupler and Hall Effect sensor were chosen for measuring the outputs from the wind turbine. In addition to the voltage and the current, the power generated by the wind turbine was displayed on the online portal. The power can be calculated by multiplying the voltage and the current. Section 4.2 and 4.3 discuss the measurements and circuit configurations used for the Hall Effect sensor and the optocoupler.

4.2 Hall Effect Sensor

The LEM LA 100-P current transducer was used. This is a Hall Effect sensor that was built for the purpose used in the wind turbine measuring circuit. It measures both DC and AC, and has galvanic isolation between the primary circuit (high power) and the secondary circuit (low power electronic circuit). Additionally, it can read a current of up to 160A and it is low power - consumes 10mA, and runs on 12V to 15V. Lastly, it can withstand high voltages and has no additional circuit configuration, besides a measuring resistor.

4.2.1 Circuit configuration

The LEM LA100-P/SP13 is a closed loop (compensated) current transducer producing a current, I_R, and a voltage V_R, which are proportional to the current being measured, I_{IN}. This proportionality is called the conversion ratio, K_N, and it is 1:1000 (I_R:I_{IN}). Measurements were done in order to investigate the operation of the Hall Effect sensor, to ensure that it is working properly and to calculate the error factor that can be expected when using this device.

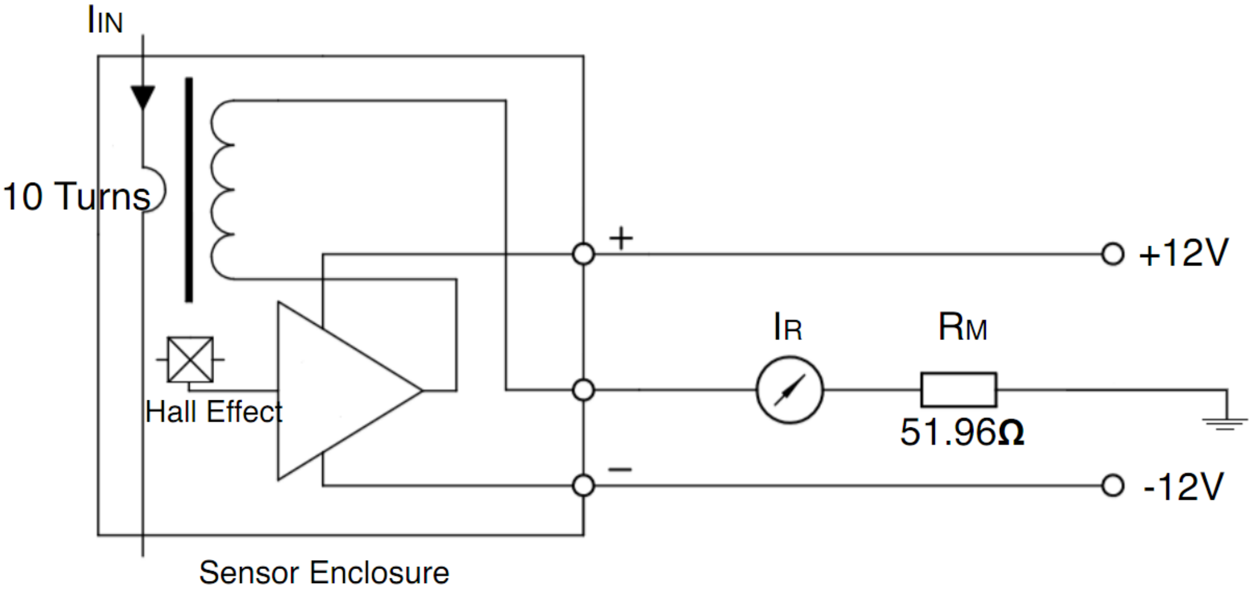


Figure 15 - Circuit Setup for Hall Effect Measurements

The circuit used for the measurements is shown in Figure 15. The controlled input current, I_{IN}, was wrapped around the Hall Effect sensor with 10 turns to achieve high magnetic coupling. The Hall Effect sensor sits within an enclosure that includes a core to increase the magnetic field, as well as an amplification circuit which amplifies the signal from the sensor. A measuring resistor was then connected to the measuring pin of the transducer, whose values matching those recommended by the manufacturer. The current and voltage flowing through this resistor were then measured and used to calculate the error expected for the device.

4.2.2 Testing and Error Characterization

To verify the conversion ratio and to calculate the error of the Hall Effect measurements were taken and then analyzed.

4.2.2.1 First Set of Measurements

In the following sets of measurements, the voltage across the resistor was measured. This was done due to the fact that the Hall Effect sensor will be connected to a NI USB 6008 DAQ which can only measure voltage levels. The DAQ will measure the voltage drop over the resistance in order to calculate the current of the Wind Turbine. To use the voltage measurement to estimate the current generated by the wind turbine, an error needs to be found for this procedure. For the this set of measurements, 20 turns were used on the Hall Effect and a 1% 1W resistor rated at 51Ω, and was measured in the lab with the ohmmeter to give a value of 51.862 Ω.

Table 1 - First Set of Measurements for Current Error Characterization

I_{IN} (A)	I_{IN}×N (A)	V_R (V)	I_R (A)	V_R/R (A)
0	0.00	0.0096	2.2000E-04	0.000185107
0.008	0.16	0.0235	4.5900E-04	0.000453126
0.016	0.32	0.0350	6.6100E-04	0.000674868
0.028	0.56	0.0465	8.6100E-04	0.000896610
0.038	0.76	0.0580	1.0640E-03	0.001118353
0.050	1.00	0.0695	1.2650E-03	0.001340095
0.060	1.20	0.0810	1.4650E-03	0.001561837
0.070	1.40	0.0925	1.6650E-03	0.001783579
0.080	1.60	0.1040	1.8660E-03	0.002005322
0.090	1.80	0.1154	2.0670E-03	0.002225136
0.098	1.96	0.1248	2.2290E-03	0.002406386
0.196	3.92	0.2376	4.2050E-03	0.004581389
0.300	6.00	0.3506	6.1850E-03	0.006760248
0.400	8.00	0.4636	8.1630E-03	0.008939108

0.500	10.00	0.5765	1.0141E-02	0.011116039
0.598	11.96	0.6895	1.2120E-02	0.013294898
0.698	13.96	0.8004	1.4060E-02	0.015433265
0.798	15.96	0.9135	1.6041E-02	0.017614053
0.900	18.00	1.0266	1.8022E-02	0.019794840
1.000	20.00	1.1396	2.0001E-02	0.021973699
1.500	30.00	1.7032	2.9869E-02	0.032841001
2.000	40.00	2.2672	3.9748E-02	0.043716016
2.498	49.96	2.8346	4.9653E-02	0.054656589
2.998	59.96	3.4002	5.9526E-02	0.065562454
3.498	69.96	3.9668	6.9399E-02	0.076487602
4.000	80.00	4.5235	7.9271E-02	0.087221858
4.502	90.04	5.0994	8.9140E-02	0.098326328
5.000	100.00	5.6679	9.9041E-02	0.109288111

Figure 16 shows the measured results obtained in the blue circles, and the least square fit line in green calculated using the polyfit function. The line of least square fit appears to be going through all of the real data points in this plot, indicating a linear relationship.

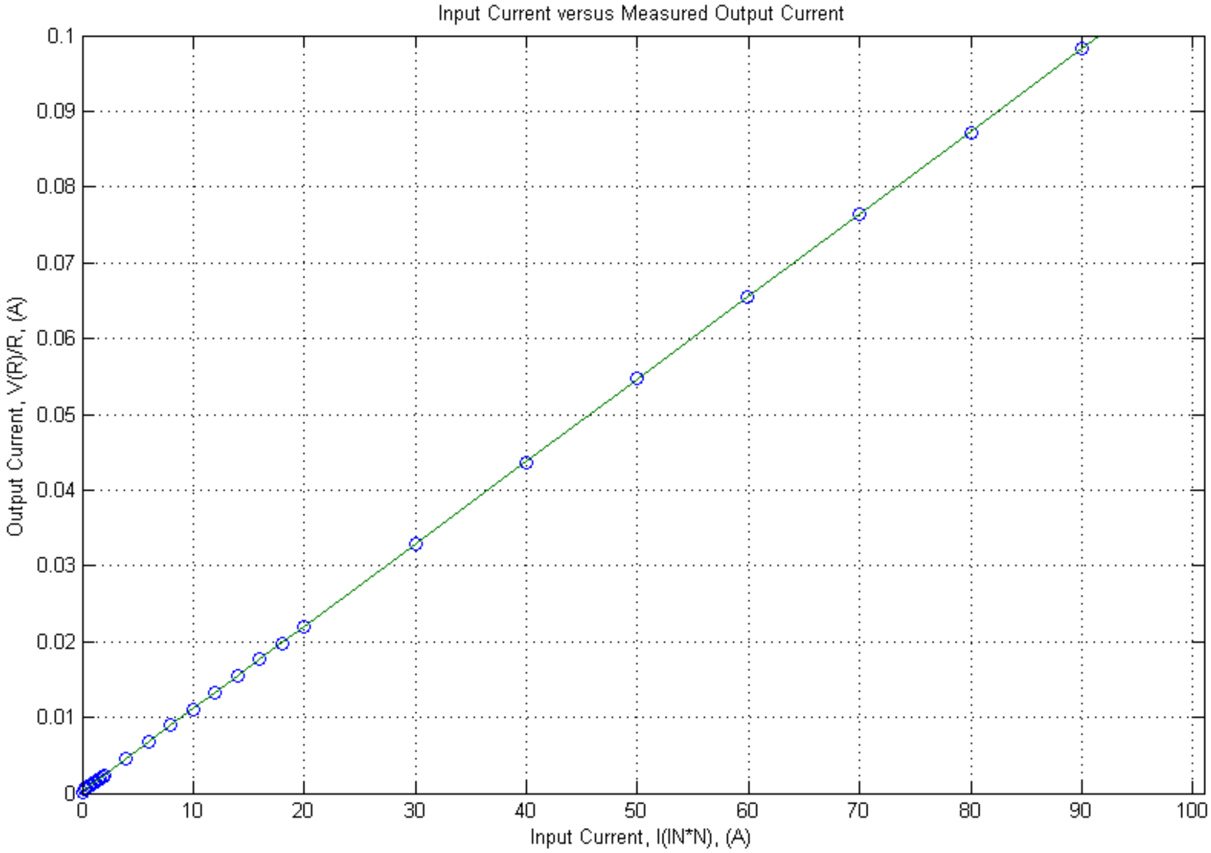


Figure 16 – Polyfit for I_R versus I_{IN}*N for N=20 turns

The line of least square fit for the voltage measured versus the input current was found, as the voltage will be measured by a different current as the output value from the transducer. The multimeter used was set on Manual for the range of measurement display, so all measured values had the same amount of decimal places. This was done in order to ensure minimal error from the multimeter. The voltage value that was measured was used for the plots, since the DAQ that will be used will measure the voltage across the resistor.

To plot the input current versus the output current, the voltage measured was divided by the value of the resistor (51.862Ω). Figure 4 shows the residual as well as the line of least square fit for the voltage measured versus the current.

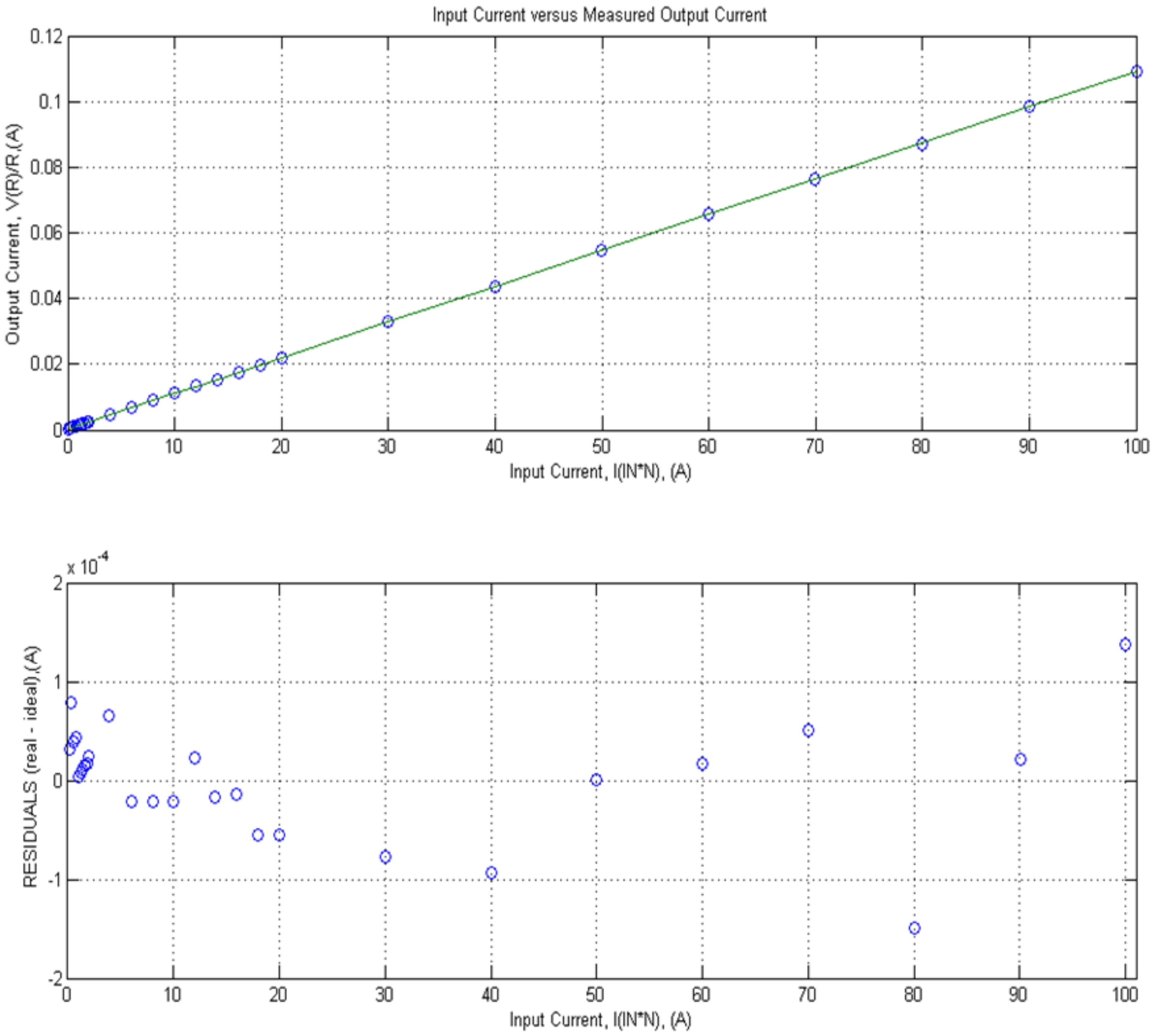


Figure 17 - Residual for First Set of Measurements

The residual plot in Figure 17 shows that the residual of the data is less than an order of 10-4 from a straight line indicating that the assumption of linearity may be satisfied by this data obtained. Additionally, these residuals are on the order of 10-4 - an order small enough to further indicate that the linearity assumption may be satisfied. When obtaining the polyfit of the data with Matlab, the slope and offset of the line of least square fit are acquired. The slope and the offset of the line of least square fit can be used to calculate the average error under which the results lie when making measurements using the Hall Effect. The line of least square fit for this set of measurements was found to be $y = 0.0011x + 0.000247248$.

4.2.2.2 Second Set of Measurements

For the second set of measurements, 20 turns were used together with the previous resistor used. As with the first set of measurements, the voltage was plotted against the input current, and the measurement display was set to manual. Table 2 shows the results obtained for the second set of measurements.

Table 2 - Second set of Measurements for Current Error Characterization

I_{IN} (A)	**I_{IN}×N (A)**	**V_R (V)**	**I_R (A)**	**V_R/R (A)**
0	0.00	0.0090	1.8600E-04	0.000173537
0.008	0.16	0.0227	4.2700E-04	0.000437700
0.016	0.32	0.0342	6.3000E-04	0.000659442
0.028	0.56	0.0457	8.3100E-04	0.000881185
0.038	0.76	0.0573	1.0350E-03	0.001104855
0.050	1.00	0.0688	1.2360E-03	0.001326598
0.058	1.16	0.0804	1.4380E-03	0.001550268
0.070	1.40	0.0919	1.6390E-03	0.001772010
0.080	1.60	0.1034	1.8410E-03	0.001993753
0.090	1.80	0.1150	2.0420E-03	0.002217423
0.098	1.96	0.1243	2.2060E-03	0.002396745
0.196	3.92	0.2372	4.1830E-03	0.004573676
0.300	6.00	0.3503	6.1640E-03	0.006754464
0.400	8.00	0.4633	8.1430E-03	0.008933323
0.500	10.00	0.5762	1.0122E-02	0.011110254
0.598	11.96	0.6893	1.2100E-02	0.013291042
0.696	13.92	0.8001	1.4041E-02	0.015427481
0.798	15.96	0.9132	1.6022E-02	0.017608268
0.900	18.00	1.0265	1.8004E-02	0.019792912
1.000	20.00	1.1395	1.9983E-02	0.021971771
1.500	30.00	1.7030	2.9851E-02	0.032837145
2.000	40.00	2.2668	3.9734E-02	0.043708303
2.498	49.96	2.8340	4.9631E-02	0.054645019
2.998	59.96	3.3988	5.9499E-02	0.065535459
3.498	69.96	3.9642	6.9366E-02	0.076437469
4.000	80.00	4.5306	7.9234E-02	0.087358760
4.502	90.04	5.0971	8.9102E-02	0.098281979
5.000	100.00	5.6669	9.9009E-02	0.109268829

Figure 18 shows the residual plot for the fifth set of measurements, showing the small residuals that. The line of least square fit was found to be $y = 0.0011x + 0.00023724$.

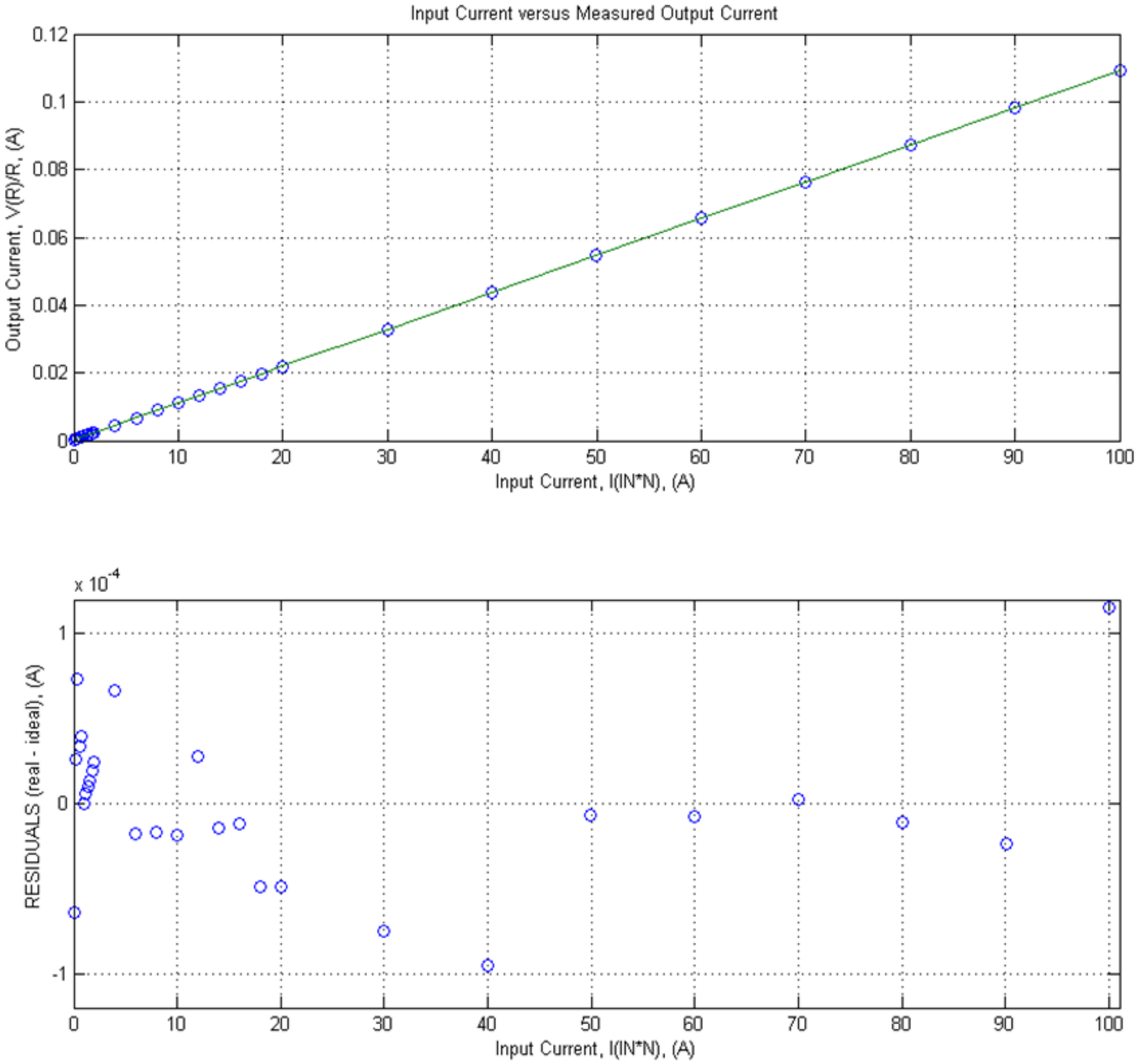


Figure 18- Residual of the Second set of Measurements

4.2.2.3 Third Set of Measurements

Finally, the last set of measurements was taken. Just as with the first and the second sets, the same setup of 20 turns and a 1W 1% 51Ω resistor was used. The voltage output over the measuring resistor was measured and can be seen in Table 3.

Table 3- Result set for third Set of Measurements for Current Error Characterization

I_{IN} (A)	I_{IN}×N (A)	V_R (V)	I_R (A)	V_R/R (A)
0	0.00	0.0090	1.8400E-04	0.000173537
0.008	0.16	0.0230	4.2400E-04	0.000443485
0.016	0.32	0.0346	6.2400E-04	0.000667155
0.028	0.56	0.0461	8.2500E-04	0.000888897
0.038	0.76	0.0577	1.0270E-03	0.001112568
0.050	1.00	0.0692	1.2280E-03	0.001334310
0.060	1.20	0.0807	1.4270E-03	0.001556053
0.070	1.40	0.0923	1.6300E-03	0.001779723
0.080	1.60	0.1038	1.8290E-03	0.002001465
0.090	1.80	0.1152	2.0300E-03	0.002221280
0.098	1.96	0.1246	2.1930E-03	0.002402530
0.196	3.92	0.2375	4.1610E-03	0.004579461
0.300	6.00	0.3505	6.1330E-03	0.006758320
0.400	8.00	0.4634	8.1040E-03	0.008935251
0.500	10.00	0.5764	1.0072E-02	0.011114111
0.598	11.96	0.6894	1.2044E-02	0.013292970
0.698	13.96	0.8002	1.3977E-02	0.015429409
0.798	15.96	0.9133	1.5959E-02	0.017610196
0.900	18.00	1.0265	1.7930E-02	0.019792912
1.000	20.00	1.1395	1.9892E-02	0.021971771
1.500	30.00	1.7030	2.9710E-02	0.032837145
2.000	40.00	2.2668	3.9573E-02	0.043708303
2.498	49.96	2.8334	4.9446E-02	0.054633450
2.998	59.96	3.3982	5.9247E-02	0.065523890
3.498	69.96	3.9637	6.9069E-02	0.076427828
4.000	80.00	4.5304	7.8908E-02	0.087354903
4.502	90.04	5.0967	8.8609E-02	0.098274266
5.000	100.00	5.6667	9.8413E-02	0.109264972

Figure 19 shows the residual plot for the third set of measurements. The line of least square fit was found to be y=0.0011x + 0.00024235.

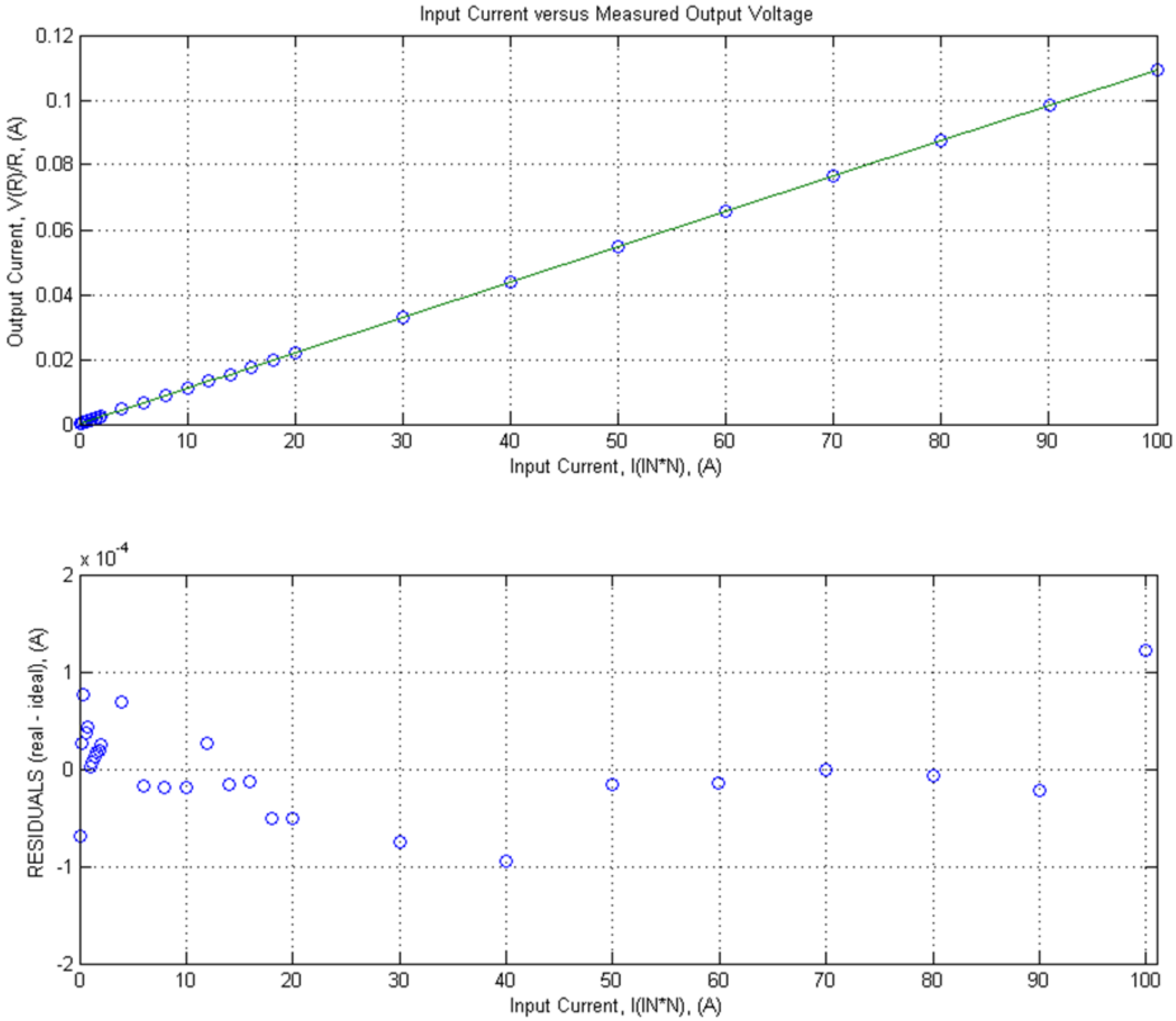


Figure 19- Residual of the third Measurement

For these sets of measurements, all of the least square fit lines had the same slope and the same offset. The slope error for these lines turned out to be 0.0011 and the offset was found to be 0.2x10-3. Since these were three different measurements with the setup that will be used to measure the current using the DAQ, one could conclude that these are error results that may be correct. Next, the error estimation from the data sheet must be found to see if the error estimation using these results is correct or not.

4.2.2.7 Error Calculation

It was known that the relationship between the unknown current, IIN and the output current from the Hall Effect sensor is as follows:

$$I_{IN} = KI_{OUT}$$

However, for the last three sets of measurements, the voltage that was measured was used which was then divided by the resistor value:

$$I_{IN} = K\frac{V_{measure}}{R_M}$$

It was known that the following properties contribute to an error in the result:

- An offset current, I_{Offset}
- Error for K_N, ε_K
- Measuring error for the Resistor, ε_{RM}
- Linearity error, ε_L

Next, the error was found with the following format:

$$I_{OUT} = I_{IN}K_N(error) + I_{Offset}$$

The error was then found using the following format:

$$I_{OUT} = I_{IN}K_N[(1 + \varepsilon_K)(1 + \varepsilon_L)(1 + \varepsilon_{RM})] + I_{Offset}$$

Since multiplying two error factors by each other yields very small errors, these were negligible. The following error format was then found:

$$I_{OUT} = I_{IN}K_N(\varepsilon_K + \varepsilon_L + \varepsilon_{RM}) + I_{Offset}$$

Then, using the manufacturer's data sheet, it was found that for the Hall Effect sensor:

- An offset current, I_{Offset} = 0.2mA
- Error for K_N, ε_K = 0.0045
- Measuring error for the Resistor, ε_{RM} = 0.0015
- Linearity error, ε_L = 0.010 (retrieved from the multimeter manual).

Finally, it was observed that the line of the error would be y=0.016x+0.0002. As the last three measurements all have the line of y=0.0011x+0.0002, and it was concluded that this error measurement is sufficient and can be used in future analysis with the Hall Effect sensor.

4.3 Optocoupler

The following sections discuss the circuit configuration for the optocoupler as well as the error calculations that were done.

4.3.1 Circuit configuration

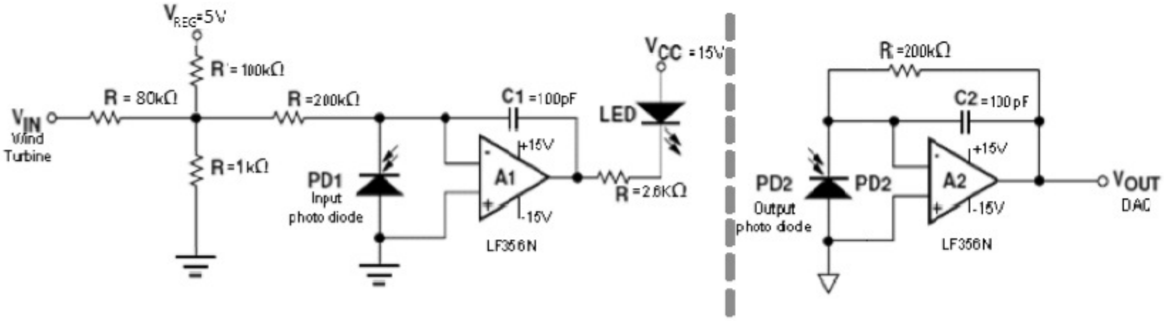
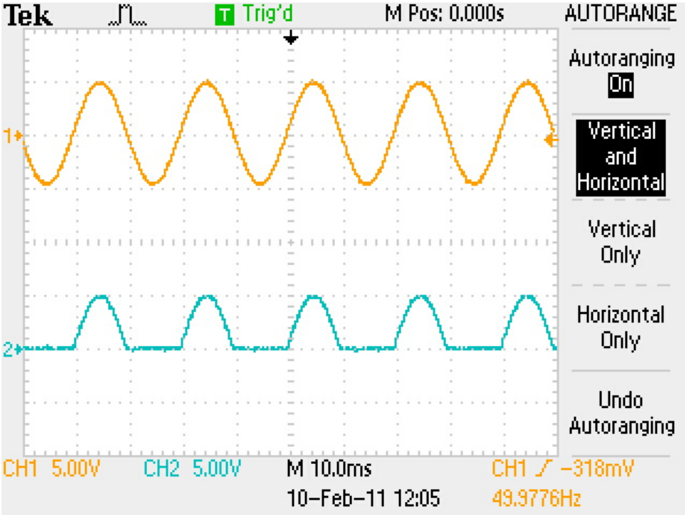


Figure 20 - Optocoupler Circuit Configuration

The optocoupler used in the voltage measuring circuit is the Avago HCNR201 high-linearity analog optocoupler. This optocoupler is made up of an AlGaAS LED that shines on two photodiodes with nearly identical characteristics. LED typically are nonlinear and have drift characteristics, which is why one of the photodiodes was used as input photodiode that monitors the light output of the LED and then stabilizes it. The second photodiode produces a photo current that is linearly equal to the light output of the LED (Technologies, 2008). The three parts are sealed within a silicon chip to protect against any ambient light interference. The reason for choosing this product was as it contained galvanic isolation. However, the optocoupler was unable to read values larger than 5V. For this reason, a voltage divider was set up, using two resistors as seen in Figure 20, right at the input voltage from the wind turbine. This voltage divider was set up to step down the input voltage by a factor of 80, as to allow the potential of up to 400V to be read by the optocoupler.

4.3.2 Measurements & Test Results

The first step in verifying the functionality of the circuit was to check for its input and output waveforms. An input function, V_{IN} was generated with a 1 kHz 5V peak to peak signal. An oscilloscope was connected to the system, with channel 1 attached to V_{IN} as seen in Figure 21, and channel 2 attached to V_{OUT}, shown on channel 2. Figure 21 was an initial positive response, as it demonstrated that there was feedback from the optocoupler; however, it was still not as accurate as it was able to be. This problem was fixed by changing the values of the capacitors in the circuit. The original 1000pF capacitors were too large and by replacing them with 10pF capacitors the system then was operating properly to read the correct response.


Tek Trig'd M Pos: 0.000s

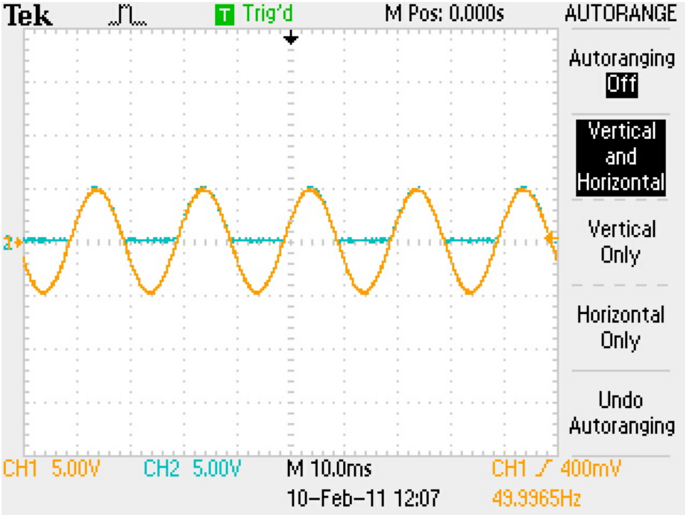
CH1 5.00V CH2 5.00V M 10.0ms CH1 ⁄ −318mV

10−Feb−11 12:05 49.9776Hz

Autoranging On

Figure 21 – Input Signal and Output Signal from the Optocoupler

The operational circuit produced in Figure 21, shows the two signals separated from each other, the input voltage, V_{IN} is the generated signal from the function generator and V_{OUT} was the response of the optocoupler. Figure 21 shows how the optocoupler responded only to the positive input of the system, as there is no way to indicate a negative voltage with photodiodes. This was not an issue for the system, as negative voltage levels weren't expected to be produced by the wind turbine.


Tek Trig'd M Pos: 0.000s

CH1 5.00V CH2 5.00V M 10.0ms CH1 ⁄ 400mV

10−Feb−11 12:07 49.9965Hz

Autoranging Off

Figure 22 – Overlapping the Input and Output Signal from the Optocoupler

Finally, Figure 22 demonstrates how V_{IN} and V_{OUT} overlap, signifying how the optocoupler follows the input signal. To further verify the accuracy to the optocoupler, measurements were done to specific the error within the output.

4.3.2.1 First Set of Measurements

After demonstrating functionality of the circuit, measurements were made to determine the precision of the system. The function generator was removed from the input and a voltage source was connected in its place. Table 4 shows the first set of measurements that were done with a voltage source of V_{CC} = ± 12V

Table 4 – Optocoupler Measurements for Voltage Error Characterization

Voltage (V)	V_{IN} (V)	V_{OUT} (V)
0.00	0.00010	0.001327
0.01	0.00682	0.001327
0.02	0.01682	0.001330
0.03	0.02690	0.006400
0.04	0.03690	0.018500
0.05	0.04680	0.028900
0.10	0.09690	0.079600
0.20	0.19710	0.177800
0.30	0.29730	0.282400
0.40	0.39740	0.384400
0.50	0.49750	0.484100
1.00	0.99810	0.992900
1.50	1.49860	1.497900
2.00	1.99900	2.003900
2.50	2.50010	2.514800
3.00	3.00070	3.021500
3.50	3.50120	3.526400
4.00	4.00160	4.035800
4.50	4.50200	4.540400
5.00	5.00240	5.048400
5.50	5.50270	5.525300
6.00	6.00300	5.566300
6.50	6.50330	5.566300

Using the data obtained in Table 4, further analysis was taken.

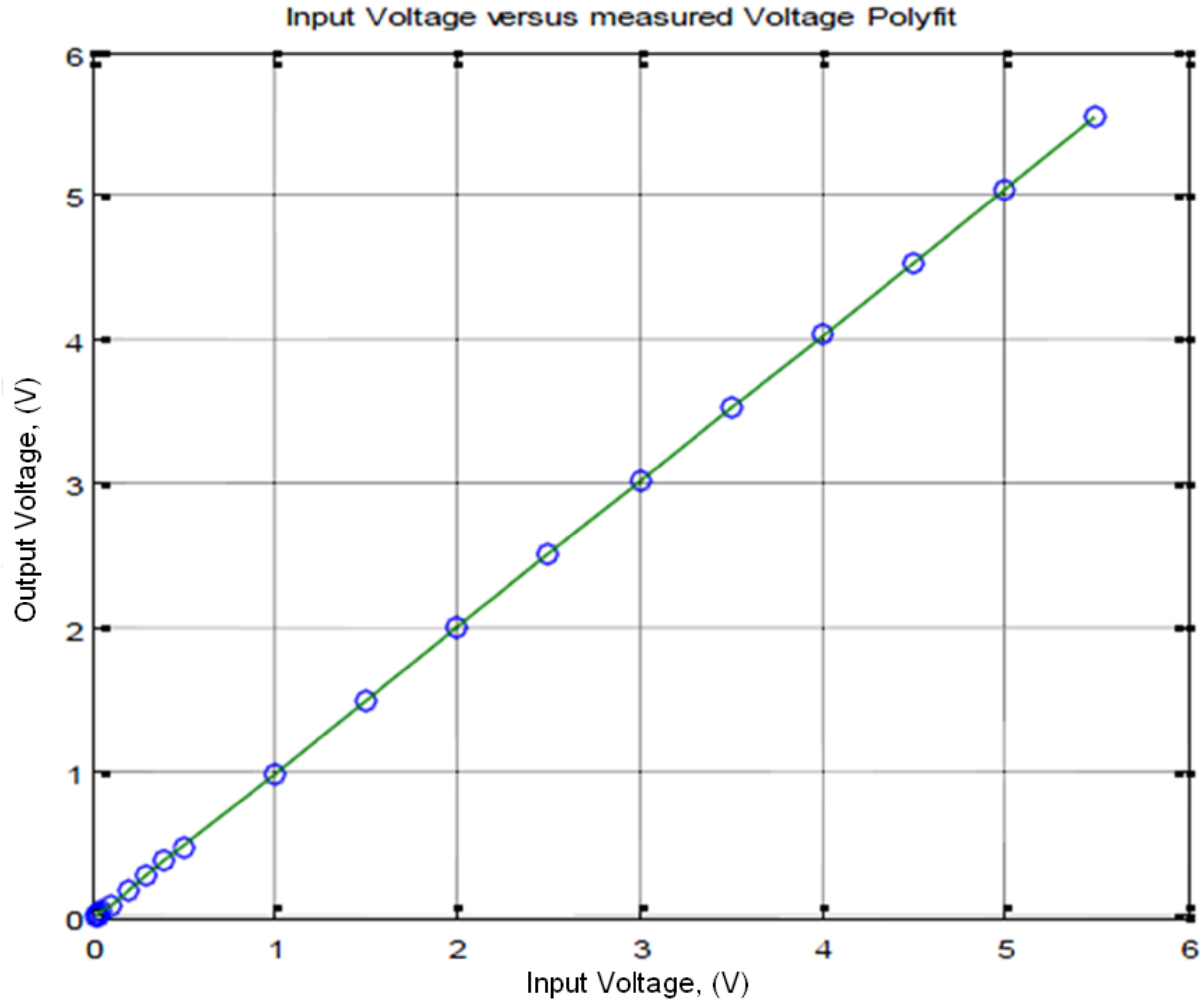


Figure 23- Optocoupler Polyfit

Figure 23 shows the graph of the input voltage versus the measured output voltage of the system. To graph this data a polyfit function was used to determine the error estimates in respect the line of best fit. The line of best fit depicted on the graph has a constant slope of 1.0128, which was an appropriate result because it resembled the desired linearity of the system. The y-intercept of the line of best fit was -0.0187, resulting in a small offset from the system. These results were as expected and desired because they demonstrated the linearity that was desired, which was the reason for choosing the HCNR201 optocoupler.

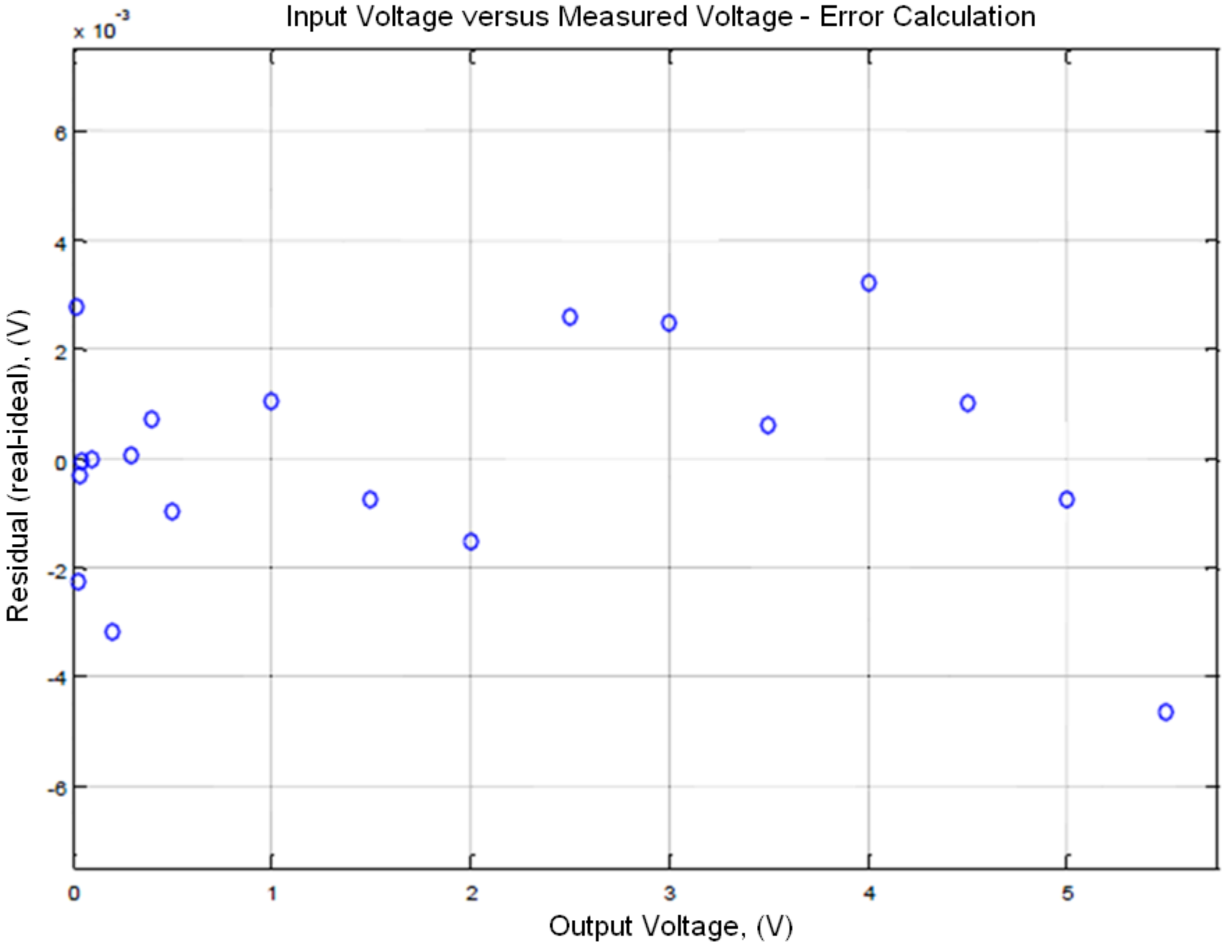


Figure 24 - Optocoupler Residuals

The purpose of plotting residuals was to show the distance of a point from the line of best fit. If a residual is positive the point is above the best fit line and negative when the point is below the line of best fit. Figure 24 shows the residuals of the measurements taken in respect to the line of best fit. The line of best fit is centered at zero and the points follow a similar trend of following that line. This demonstrates that even though the measurements were not perfectly linear with a slope of one, they were within an accurate region of measurement.

4.3.2.2 Second Set of Measurements

A second set of measurements was taken to have more precise results.

Table 5 - Second Measurements Optocoupler for Voltage Error Characterization

Voltage (V)	V_{IN} (V)	V_{OUT} (V)
0.0	0.00010	0.00140
0.5	0.49724	0.49841
1.0	0.99784	1.00475
1.5	1.49820	1.51010
2.0	1.99850	2.01680
2.5	2.49960	2.52340
3.0	3.00000	3.02980
3.5	3.50050	3.53600
4.0	4.00080	4.02680
4.5	4.50120	4.54860
5.0	5.00140	5.05440
5.5	5.50170	5.56120
6.0	6.00180	6.06680
6.5	6.50200	6.57340

Using the measurements that were found in Table 5, the data was plotted.

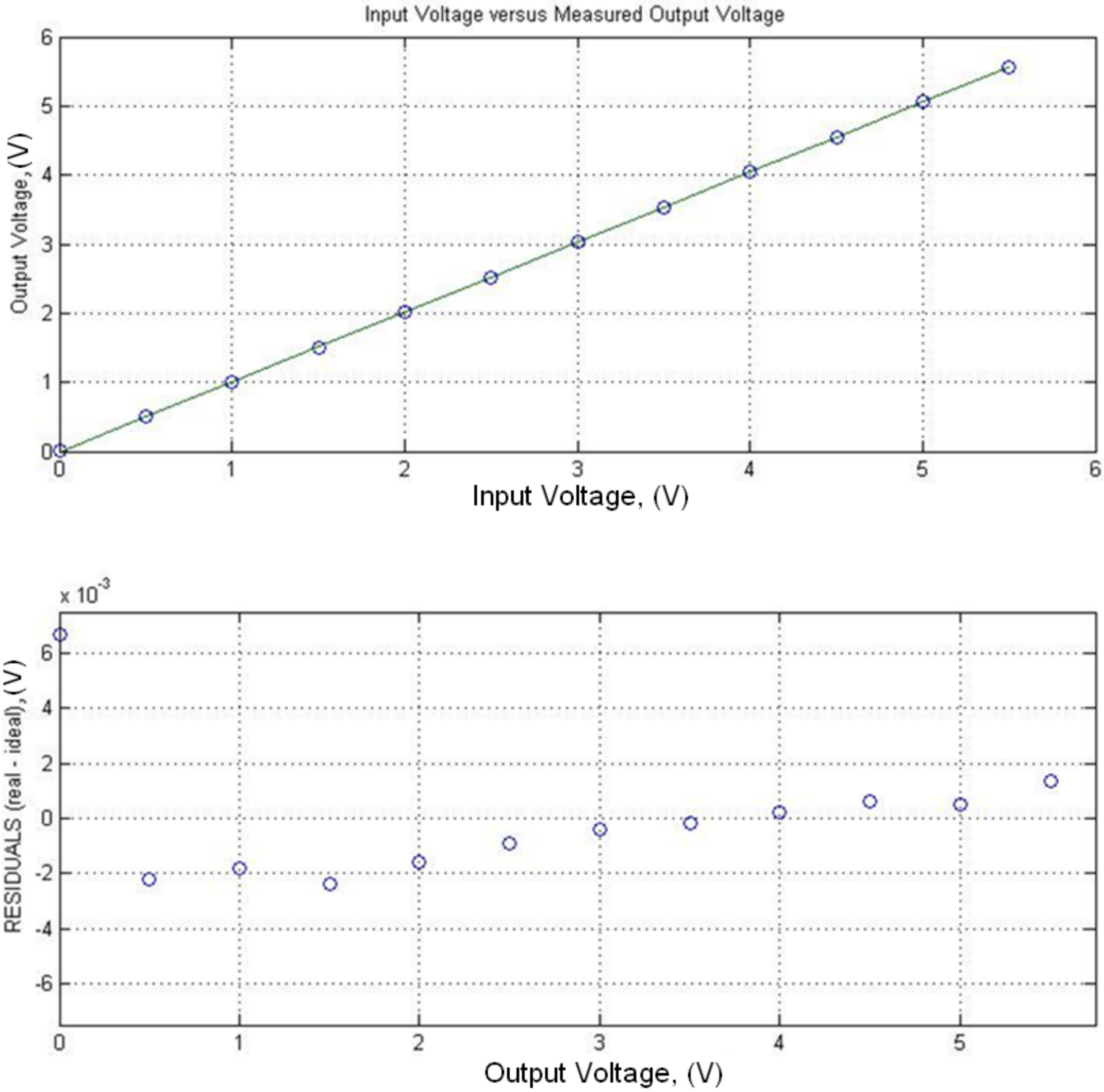


Figure 25 - Optocoupler Measurements Residuals

To insure the accuracy of the results and to make sure the proper function of the system, measurements were taken at a different point in time using different lab equipment. From the measured results and graphs produced from Matlab, it was evident that this was a proper functioning circuit and met the needs of the project goals. The data point at zero does not follow the rest of the data points. This irregularity is due to the system cannot completely measure zero because of noise and offset. The least square fit of the line was $y = 1.0118x - 0.0053$. This value was comparable to the least squares fit of $= 1.0128x - 0.0187$ obtained from the first set of measurements. This demonstrates consistency in the measurement circuit, which is valuable when determining the error of the optocoupler.

4.3.2.2 Error Calculation

The circuit was now functional and reading the voltage correctly, however to ensure that the measurements were coming within the manufacturer's specifications the total error of the system was calculated. To calculate the error, three parts of the circuits were analyzed separately first, and then combined to produce a total error output.

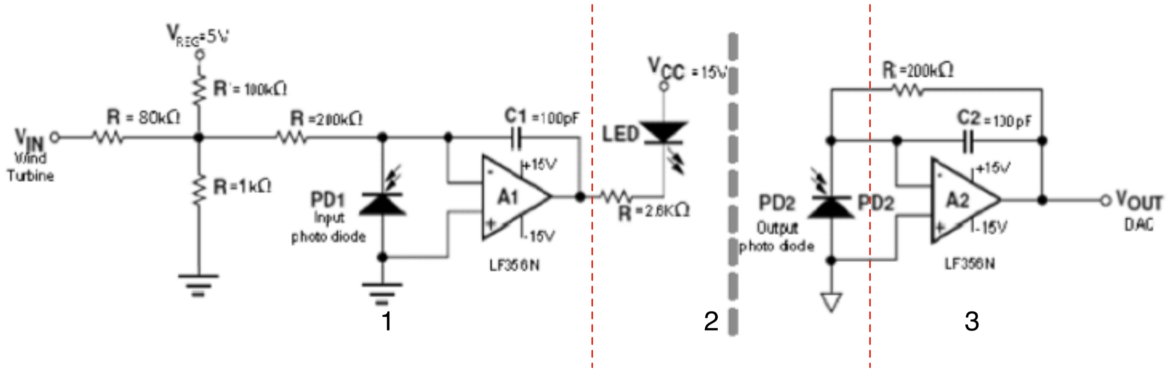


Figure 26 - Breakdown of Optocoupler Circuit for Error Calculation

Figure 26 shows the different sections of the optocoupler circuit that were broken down and analyzed separately as follows to calculate the error:

V_{IN}	Voltage supplied by the wall transformer
PD_1	Photo Diode 1: Stabilizes the LED
PD_2	Photo Diode 2: Measures the light produced from the LED
V_{OS1}	Voltage offset 1: $\pm .003V(typical) \pm .01V(Max)[From\ data\ sheet\ LF356N]$
V_{OS2}	Voltage offset 2: $\pm .003V(typical) \pm .01V(Max)[From\ data\ sheet\ LF356N]$
I_{BIAS1}	Current Bias 1: $\pm 30pA(typical) \pm 100pA(Max)[From\ data\ sheet\ LF356N]$
I_{BIAS2}	Current Bias 2: $\pm 30pA(typical) \pm 100pA(Max)[From\ data\ sheet\ LF356N]$
ε_{NL}	$Error\ Non\ Linearity = 0.0001$
ε_{TG}	$Error\ Transfer\ Gain = 0.05$
R1,R2	200kΩ

To calculate the error, the different sections of the circuit were analyzed. The first section, section 1 of the optocoupler circuit in Figure 26, analyzes the error from the first photodiode along with offsets from the first op amp:

$$I_{PD1} = \frac{(V_{in} - V_{OS1})}{R1} - I_{BIAS1}$$

Section 2 of Figure 26, analyzes the error due to the LED emitting on the photodiodes:

$$I_{PD2} = (1 + \varepsilon_{TG} + \varepsilon_{NL}) * I_{PD1}$$

Section 3 of Figure 26, analyzes the error due to the output of Photo Diode 2 and the second op amp:

$$V_{out} = V_{OS2} + (I_{PD2} - I_{BIAS2}) * R_2$$

The equivalent output with combing all of the circuit component equations is as follows:

$$V_{out} = V_{OS2} + R_2\,[((1 + \varepsilon_{TG} + \varepsilon_{NL})\left(\frac{(V_{in} - V_{OS1})}{R1} - I_{BIAS1}\right)) - I_{BIAS2}]$$

$$V_{out} = \left(V_{in} * \frac{R_2(1 + \varepsilon_{TG} + \varepsilon_{NL})}{R_1}\right) + \left[V_{OS2} - V_{OS1}\left(\frac{R_2(1 + \varepsilon_{TG} + \varepsilon_{NL})}{R_1}\right) - I_{BIAS1}\big(R_2(1 + \varepsilon_{TG} + \varepsilon_{NL})\big) - I_{BIAS2} * R_2\right]$$

$$V_{out} = 1.0501 * V_{in} - 0.02054$$

The slope and offset of the line of best fit which were 1.0128 and .0187, came from the least square fitting of the measurements take. The value found for the error measurements met the allowable error determined from the specifications, meaning the system is operating within specification.

4.4 Hall Effect and Optocoupler Circuit

Once the optocoupler circuit was finalized and the Hall Effect sensor error was calculated, the two were put into one circuit. To do this, it was necessary to find a source of power to power the circuit. The optocoupler needs electrical isolation, and to achieve this DC to DC converters for isolating power used. Specifically, two Remcom RB05012D chips were used in order to achieve isolated DC power rails. These DC to DC converters take in 5V DC and produce isolated DC power rails at 12V. It was possible to run both the Hall Effect and the optocoupler off of 12V, so no further voltage conversion was needed.

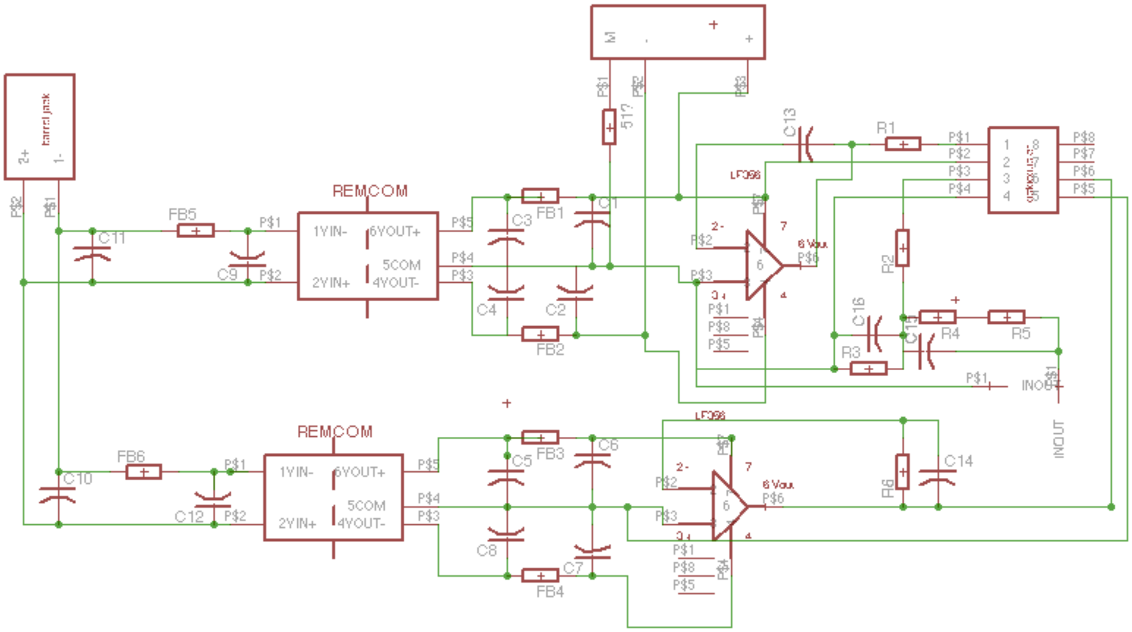


Figure 27 - Optocoupler and Hall Effect Sensor Circuit

Figure 27 shows the finalized circuit that was built for the Hall Effect and optocoupler. To power the Remcom DC/DC converters, 5V was needed. To do this, an AC wall adapter was bought, along with a DC barrel power jack. The AC adapter was a CUI EPS050100 that had a voltage output of 5V DC, and the barrel power jack was a CUI PJ-037B that had the necessary dimensions to fit with the purchased AC adapter.

4.5 LabVIEW and DAQ

As previously mentioned, LabVIEW was used to acquire the data produced from the sensor circuit using a data acquisition (DAQ) unit. Although there are many programs that allow the acquisition and manipulation of data LabVIEW was due to the fact that there was a National Instrument (NI) DAQ available in the WPI Electrical and Engineering for the project to use for no additional cost. As this DAQ turned out to be a NI DAQ, it was only natural to choose LabVIEW as the program to use, as it is easy to interface with NI devices. LabVIEW also has powerful tools allowing manipulating, analyzing, and saving data in different ways. That factor was important for this project as the data was used outside the LabVIEW environment for the online portal, and it was necessary to ensure that it would be easy to use the data LabVIEW produced. This section details the use of the DAQ and the program written in LabVIEW for the project.

4.5.1 DAQ

The DAQ was donated to the project by the Electrical and Engineering department at WPI. It is a NI USB 6008 DAQ, which has 8 analog inputs and 2 analog outputs; it has a 12 bit resolution for both the inputs and outputs. To sample the signals coming from the Hall Effect sensor and the optocoupler the DAQ was used in differential mode for measurements, sampling in a range of ±10V which allowed an input error of up to 7.73mV per measurement for the Hall Effect measurement. For the optcoupler measurement, the sampling range was to ±5V, and at that setting the DAQ had an input error of up to 4.28mV per measurement.

4.5.2 LabVIEW

The data from the DAQ connected directly to a LabVIEW program running on a computer. The signal acquired was then split so that the data coming from the DAQ could be used to make voltage and current calculations.

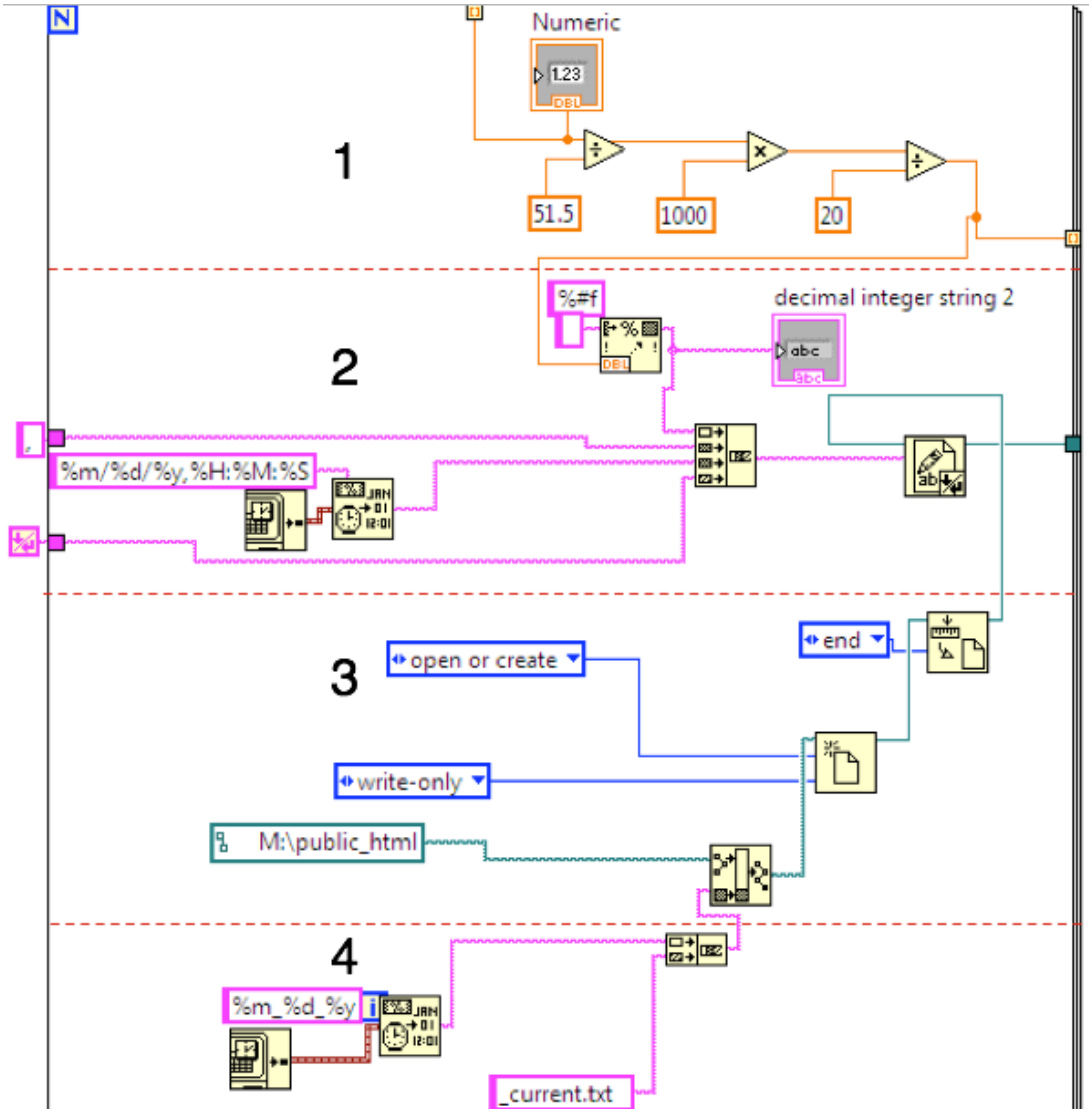


Figure 28 - Hall Effect Sensor Calculation LabVIEW

Figure 28 shows the part of the LabVIEW program that calculates and outputs the data from the Hall Effect sensor. The top area in orange indicated by the number 1 has an input coming in directly from the DAQ. These are the numeric data points that are coming directly from the DAQ. The signal coming in from the DAQ was the voltage drop over the resistor and some calculations were needed to convert the voltage to a current. First, the numeric data points coming from the DAQ indicated by the orange line were divided by 51.5. This was the value of the resistor used with the circuit. By using Ohm's law, the current was found. However, due to the conversion ratio K_N (1:1000) of the Hall Effect, the numeric value was then multiplied by 1000. Finally, since 20 turns were used to wrap the coil carrying the current being measured around the Hall Effect sensor, the numeric value was divided by 10 to provide with the correct measured current.

This numeric value was then needed to be saved in a specific format. This was accomplished by the area marked with a 2 in Figure 28. Here, the system timestamp is being continuously retrieved, indicated by timestamp function. Then, using the timestamp conversion function, the timestamp format is dissected in a specific way, indicated by %m%d%y,%H:%M:%S. This created the timestamp that would be saved together with the numeric data point being acquired at that specific time. Using the numeric conversion function, the numeric value being acquired was being converted into a string. Finally, using the concatenation function, the string containing the numeric value was concatenated together with a comma, the timestamp, and an end of line character.

The third section in the program showed in Figure 28 is in charge of creating or opening a document. The create document function was used to create a new document every day. To settings indicate that if there isn't a document already with the name that is being given, then create a new one. Section 4 is in charge of creating a timestamp and obtaining the date from it, and generating a name for the file from that timestamp. Therefore, every day at midnight, a new document was created. To distinguish from other files, the type name of the file was concatenated at the end. An example of this would be 01_12_2011_current.txt as a name file for saving the current data points on a specific day. Additionally, section 3 shows that the settings for creating a new file were such that the data would be added to the end of the already existing date in order for all the data to be saved correctly.

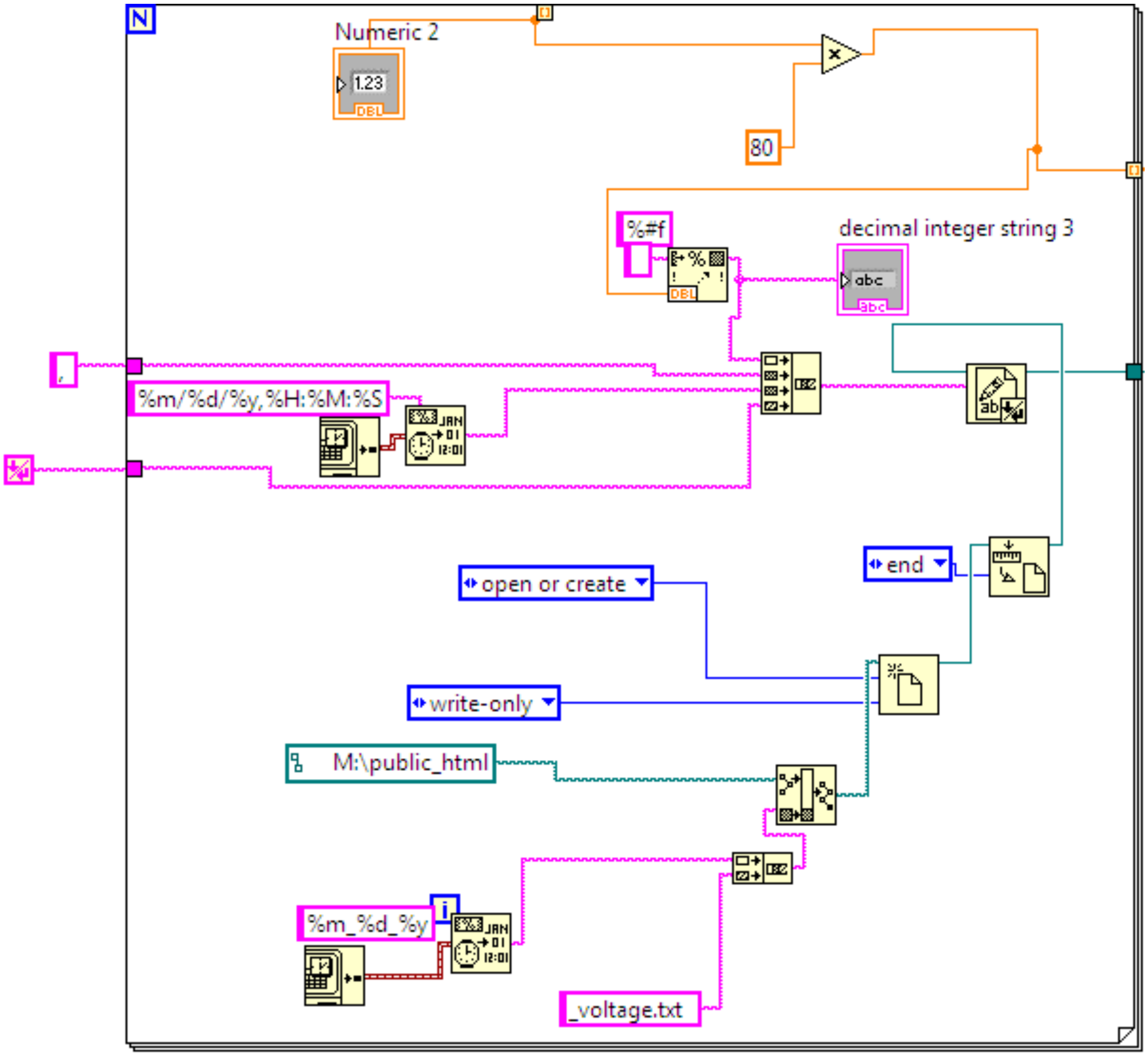


Figure 29 - Optocoupler Calculation LabVIEW

Figure 29 shows the part of the LabVIEW program that calculated and outputs the data from the Hall Effect sensor and the optocoupler. This setup was identical to the one used for the Hall Effect, except for two differences. The first is that the only type of numeric arithmetic that the data points were going through was to multiply the data points by 80 to compensate for the voltage divider and to obtain the correct voltage reading. The second difference was the word voltage was being used to concatenate on to the date when creating a new file, thus saving the file name under voltage and not current. However, using this setup, the output from the calculation had a large error bound. During testing of the data calculated, Gnuplot was used to plot the data in order to see what the data being produced looks like. The power was calculated simply by multiplying the data from the voltage and current by each other, and saving it in the same manner shown for the voltage and current calculations.

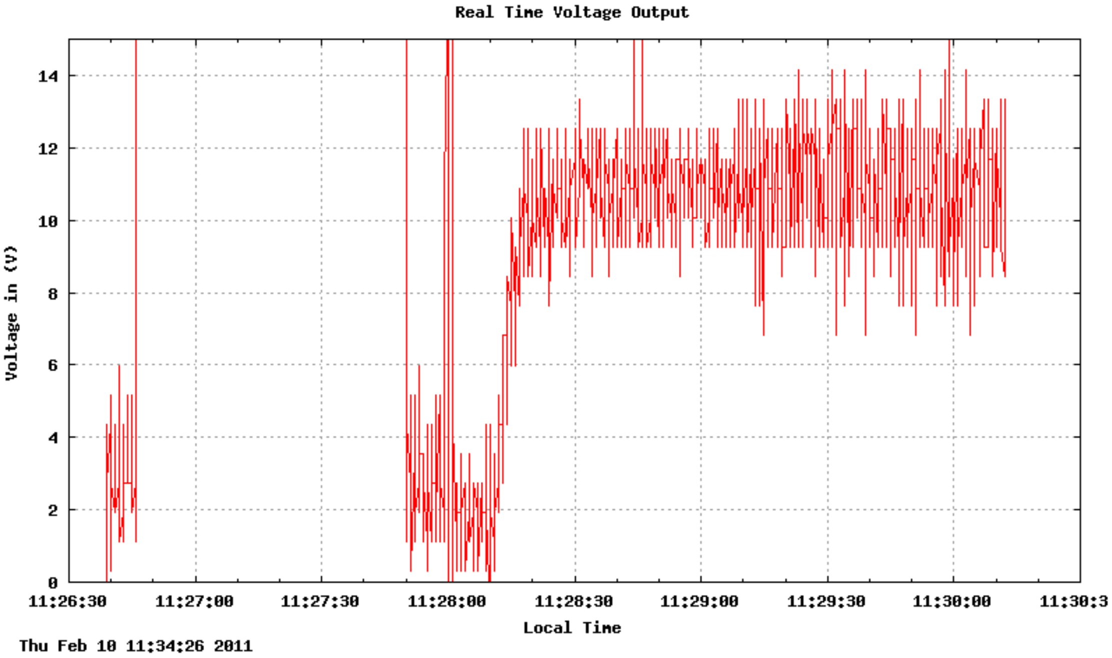


Figure 30 - Optocoupler Data Output with Noise

Figure 30 shows random data created in LabVIEW and acquired through the DAQ and optocoupler, that was then plotted using Gnuplot. From the 11:28:30 mark until the end of the data plot at around 11:30:00, the input to the optocoupler was set at 11V. However, this produced a large variation in error, ranging from around 7.5V to 12.5V. This was determined to happen due to a few factors. The voltage divider had to be large enough to reduce a potential signal of up to 400V, therefore it had a large factor of 1:80. This meant that a small signal went through the optocoupler – up to 5V, and the optocoupler then generated a small signal that was acquired by the DAQ. The small signal that was acquired by the DAQ was no larger than 5V, so the DAQ was set to acquire a signal of up to 5V for the optocoupler acquisition. At that setting the DAQ had an error of ± 4.28mV. This was a significant error as the signal coming through the DAQ was quite small. The signal was then multiplied by 80 to compensate for the voltage divider, which caused the large variation in error as seen in Figure 30.

To fix this issue, numerous troubleshooting steps were taken. First, the averaging function in LabVIEW was used to average the input and to reduce the noise, and a low pass filter was implemented into the circuit. Figure 31 shows data that was sampled before the use and then after the use of the averaging function.

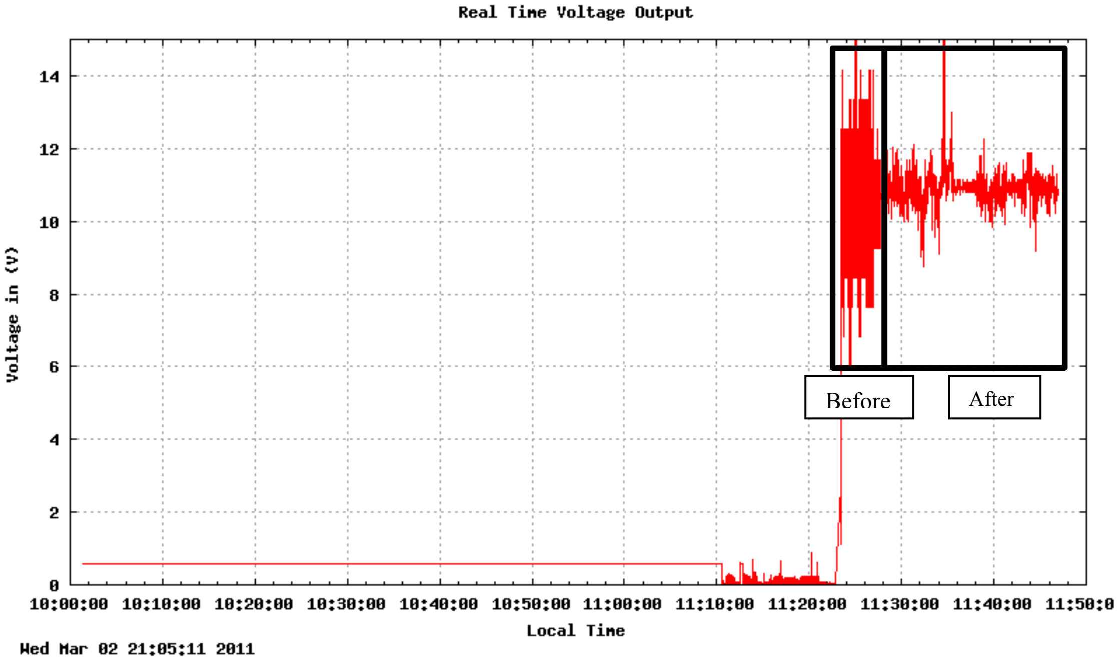


Figure 31 - Before and after the use of the Averaging Function

In Figure 31, the red block before the 11:30:00 minute mark is the data that was sampled before the use of the averaging function, and the more distinguishable noisy data after the block is the data being sampled after the use of the averaging function. In order to verify how many samples were being averaged, histograms of the samples were made.

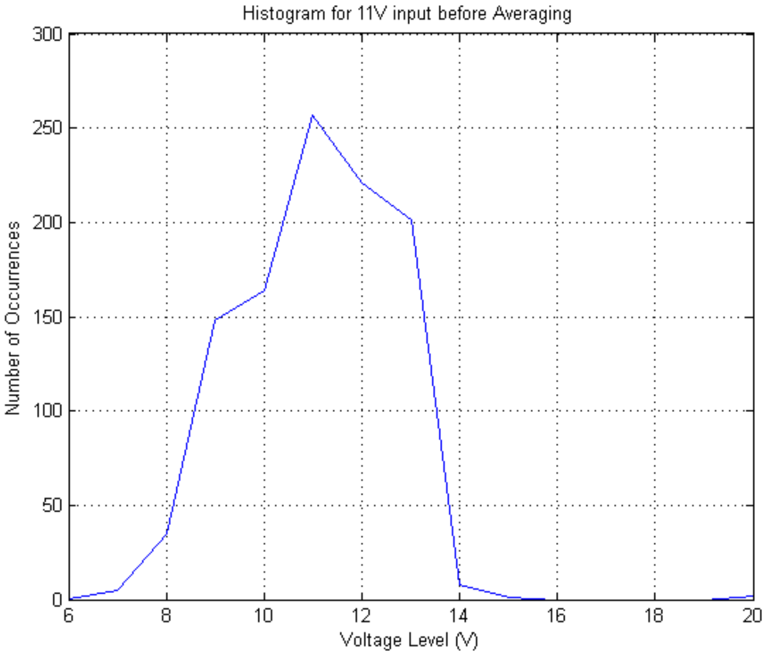


Figure 32 - Histogram for 11V input data before Averaging

Figure 32 shows the histogram for the samples that were sampled before the averaging. Note that the data is shaped like a bell curve, but has a wide standard deviation of 2.53.

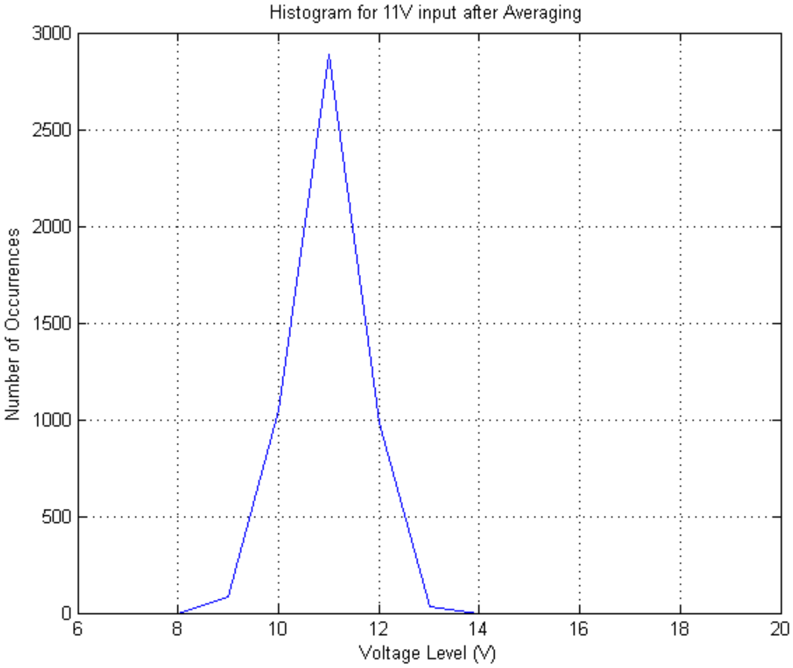


Figure 33 - Histogram for 11V input data after Averaging

Figure 33 then shows the histogram for the samples after using the averaging function. Here the standard deviation went down to 0.58. Using the standard deviations, the numbers of samples being averaged was found to be around 18 samples:

$$\sqrt{N}\sigma_{new} = \sigma_{old}$$
$$N = (\sigma_{old} / \sigma_{new})^2$$
$$N = 18$$

Although the averaging function was helping to reduce the noise, there was still a noticeable amount of noise ranging at around ±2V. It was also noticed that the output being generated by the optocoupler tended to have the shape of a sine wave. Next, each section of the circuit was examined with oscilloscope to see if there were any abnormalities in the outputs. While doing this, it was found that a radiant noise of a 60Hz sine wave was being picked up by the circuit. Furthermore, when a resistor was used between the circuit and the lead of the oscilloscope, this noise disappeared. It was then decided to try and reduce the noise by a filter.

A 10Hz low pass filter was used, as the sampling rate of the DAQ was set to 10Hz, in order to eliminate any noise above 10Hz. The values for the resistor and the capacitor were found by using the cutoff frequency definition:

$$f_{cutoff} = \frac{1}{2\pi RC}$$

The RC constant value was then found:

$$RC = \frac{1}{2\pi f_{cutoff}}$$

$$RC = \frac{1}{2\pi 10}$$

$$RC = 0.0159$$

A capacitor value of 1µF was used, with a resistor value of 15KΩ. After adding the resistor, the noise did disappear, however, the output from the optocoupler was no longer linear when sampling with the DAQ. However, when removing the DAQ and viewing the output using the oscilloscope, the noise was gone. Therefore, it was assumed that the DAQ was causing the non-linearity output in the optocoupler. It was decided to use a buffer, in an attempt to isolate the DAQ from the direct output of the optocoupler, in the hope that this would eliminate the issue. For the buffer, a LM356 op-amp was used, in a unity gain setup.

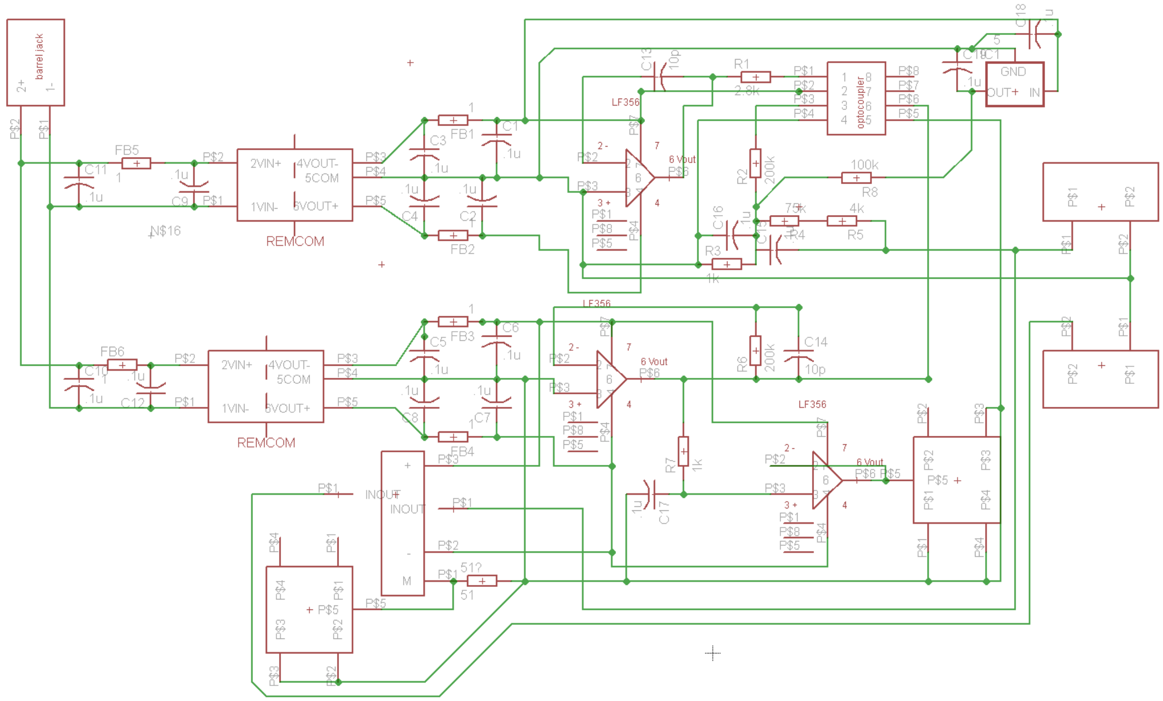


Figure 34 –Final Circuit with Filter and Buffer

Figure 34 shows the final circuit setup with the filter and the buffer. To see if the buffer was eliminating the non-linearity, the circuit was connected again. The non-linearity disappeared, as well as the noise.

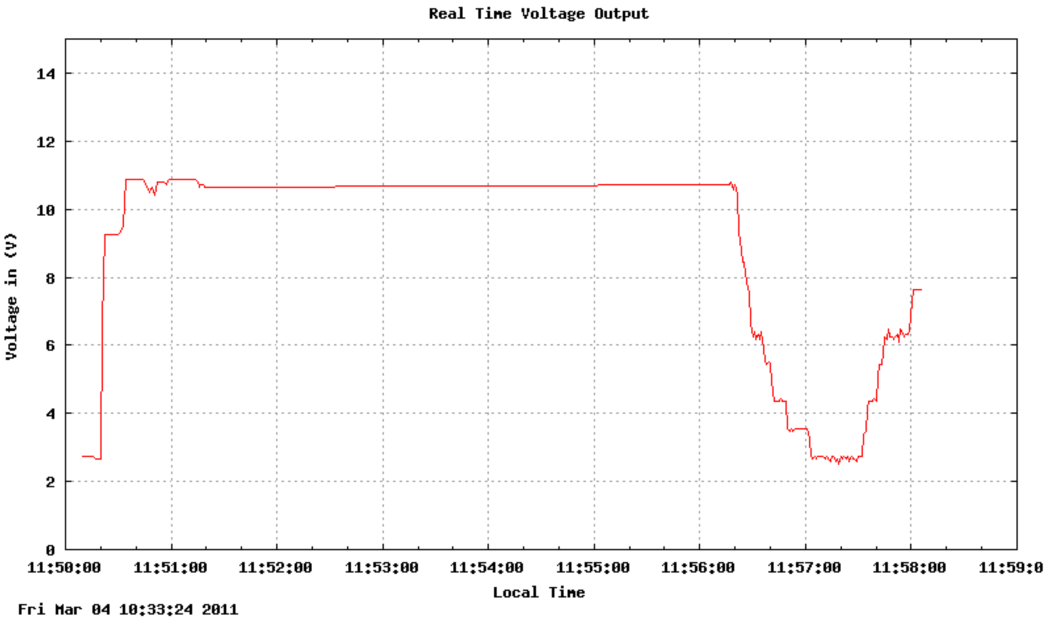


Figure 35 - Noise Reduction using Low-Pass Filter and LabVIEW Averaging

Figure 35 shows the output from the optocoupler using the filter and the buffer. An input of 11V was provided. The graph indicates that there is very minimal noise, meaning that the filter created was sufficient enough to filter out the noise from the circuit. Note that although the input was 11V, the output was a little lower than that due to the calibration error of the system. This was fixed using the calibration through LabVIEW. Finally, error handling was added to the LabVIEW program.

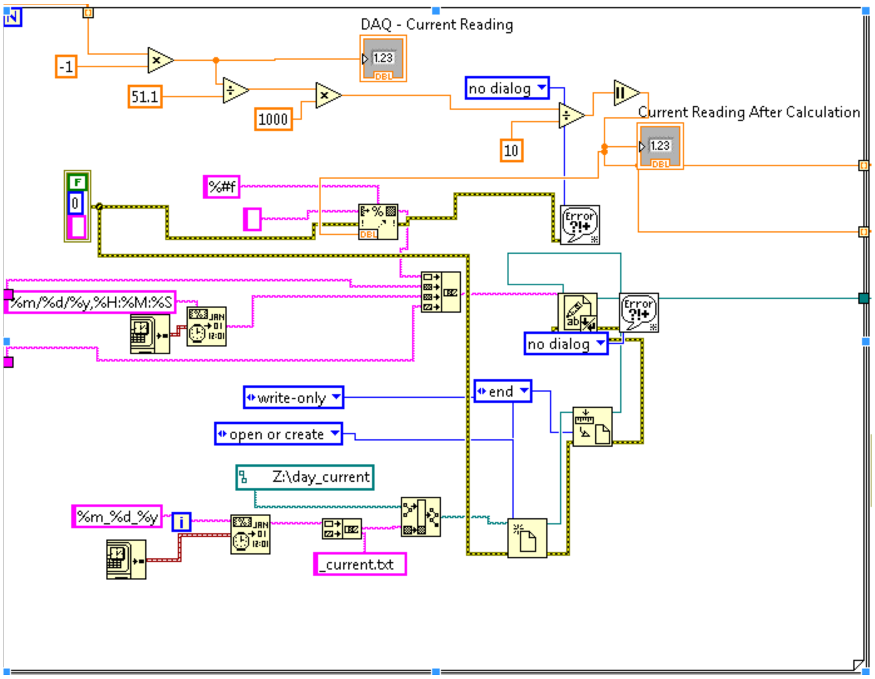


Figure 36 - Current Calculation with Error Handling in LabVIEW

Figure 36 shows the current calculation section of the LabVIEW program with the added functions used for the error handling. Any error would propagate through the program as indicated by the yellow-black dotted lines, and would lead to an error handling functions which were programmed to save the error, but continue to run the program. The errors that are handled in this fashion are errors indicating that one of the measurements wasn’t saved, and other minimal time related errors. These errors do not need immediate attention – but without error handling, LabVIEW must stop the operation of the program and requires external input from the user. To avoid this, error handling was added.

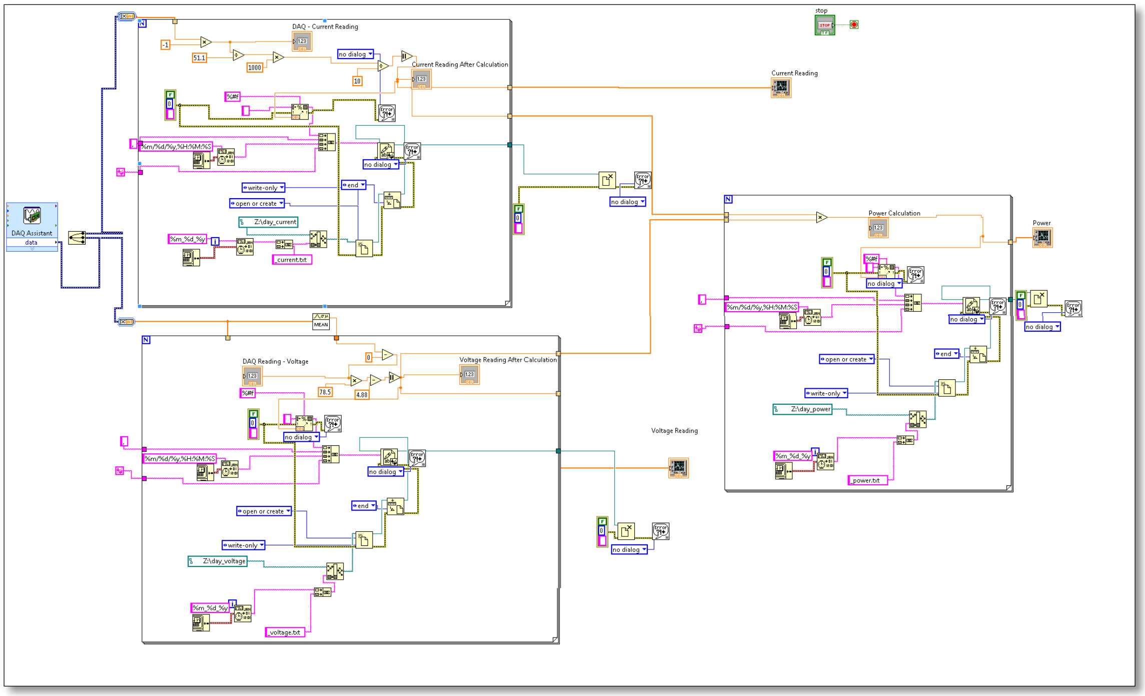


Figure 37 - Entire LabVIEW Program for Measuring and Calculating Input

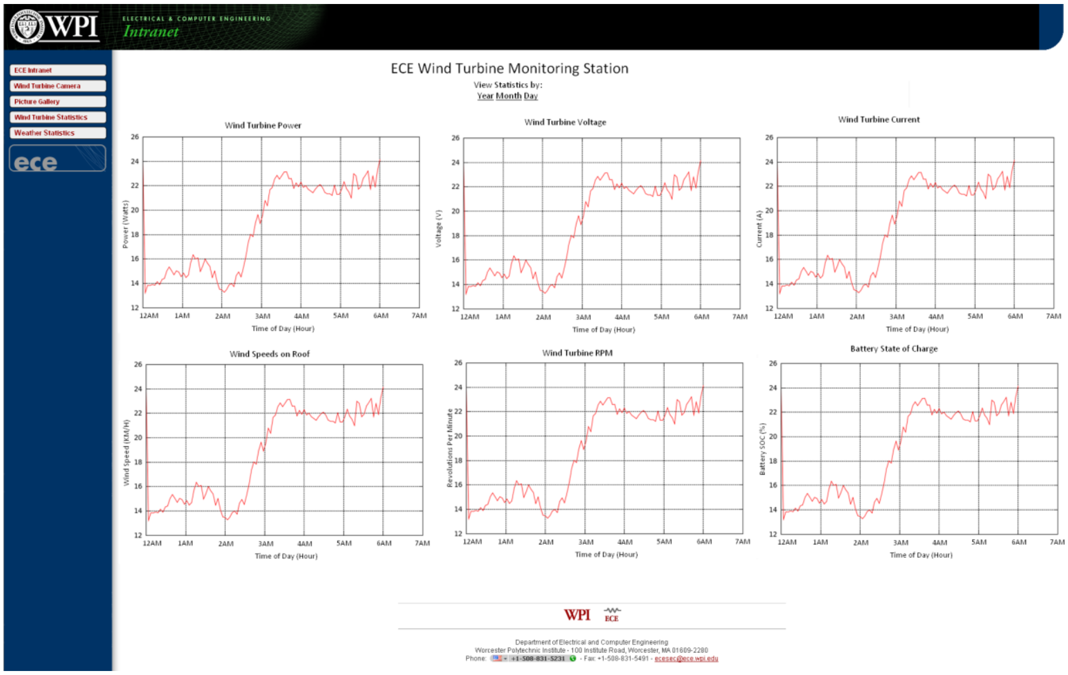
Figure 37 shows the final overview of the LabVIEW program. The program flows from the left to the right. At the outmost left part of the program is the DAQ Assistant, which receives the signal directly from the DAQ. The signal is then split, and the current and voltage are calculated off of it. The top block to the right of the DAQ Assistant is the current calculation and the bottom block is the voltage calculation. Finally, the output signals from the current and voltage measurements are sent into the right most block in Figure 37, where the power is calculated.

4.6 Online Portal

This section discusses the initial concept idea for the online portal, as well as the steps taken to build and finish it.

4.6.1 Initial Concept

The online portal allows visitors to view real-time data regarding the wind turbine online. The main component that was necessary to complete in order to get the portal to function correctly was to plot the images of the data acquired. As the data produced by LabVIEW is in real-time, the plotting needed to happen in real-time or very frequently. The images of the plots would get uploaded directly to the website and the users would view the most recently generated plot.



Department of Electrical and Computer Engineering
Worcester Polytechnic Institute - 100 Institute Road, Worcester, MA 01609-2280
Phone: +1-508-831-5231 - Fax: +1-508-831-5491 - ecesec@ece.wpi.edu

Figure 38 - Initial Mockup for Wind Turbine Website

Figure 38 shows the initial website mockup that was designed. The website mockup included a main page that was named the "ECE Wind Turbine Monitoring Station" where all plots, for the RPM, Power, Voltage, Current and potentially the weather would be available for viewing and comparing. The user had an option to toggle between daily, yearly, monthly and weekly plots, however the website would default to display the daily plots being generated.

Due to the fact that there were plots needed yearly, monthly and weekly on top of the daily ones, there was not only a need to write scripts for plotting, but also to write scripts to concatenate data together. As discussed in section 4.2.2, LabVIEW was creating a new file every day for each type of data.

Before explaining the way the data was concatenated though, it's important to understand how the data was being plotted.

4.6.2 Plotting the Data

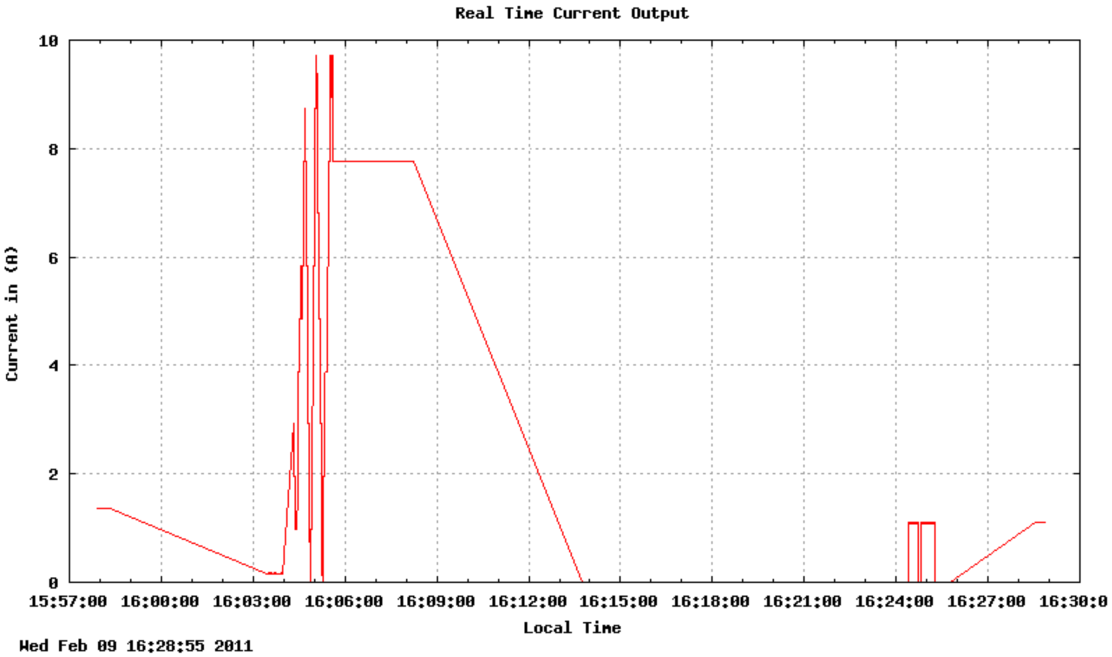


Figure 39 - Plot Produced by Gnuplot

Figure 39 shows a plot that was generated using Gnuplot in unix. Different files were created for each specific type of plot. The dimensions of the plot, as well as the axes and other plot detail were written. Then, the correct file was dissected by Gnuplot to plot the timestamps versus the data points.

Public Html

- **day_current folder**
  - day_current.gnu

    Needs to run every 5 minutes daily
- **day_power folder**
  - day_power.gnu

    Needs to run every 5 minutes daily
- **day_voltage folder**
  - day_voltage.gnu

    Needs to run every 5 minutes daily
- **month_current folder**
  - monthCurrentScript.sh

    If it's the first of the month, it creates a new text file with the month name, and then copies that day's data into it. Otherwise, it just copies/concatenates that day's data into the current file. Needs to run EVERY day 5 minutes before midnight.
- **month_power folder**
  - monthPowerScript.sh

    If it's the first of the month, it creates a new text with the month name, and then copies that day’s data into it. Otherwise, it just copies/concatenates that day’s data into the current file. Needs to run EVERY day 5 minutes before midnight.

- **month_voltage folder**
  - monthVoltageScript.sh

    If it’s the first of the month, it creates a new text file with the month name, and then copies that day’s data into it. Otherwise, it just copies/concatenates that day’s data into the current file. Needs to run EVERY day 5 minutes before midnight.
- **week_current folder**
  - week_current.gnu

    Needs to run daily
  - weekCurrentScript.sh

    If it is Sunday, it creates a new text file with the month name, and then copies that day’s data into it. Otherwise it just copies/concatenates that day’s data into the current file. Needs to run daily.
- **week_power folder**
  - weekPowerScript.sh

    If it is Sunday, it creates a new text file with the month name, and then copies that day’s data into it. Otherwise it just copies/concatenates that day’s data into the current file. Needs to run daily.
- **week_voltage folder**
  - week_voltage.gnu

    needs to run daily
  - weekVoltageScript.sh

    If it is Sunday, it creates a new text file with the month name, and then copies that day’s data into it. Otherwise it just copies/concatenates that day’s data into the current file. Needs to run daily.
- **year_current folder**
  - yearPowerScript.sh

    If it is the first day of the year, it creates a new text file with the year name, and then copies that day’s data into it. Otherwise it just copies/concatenates that day’s data into the current file. Needs to run daily, before midnight.
- **year_power folder**
  - yearPowerScript.sh

    If it is the first day of the year, it creates a new text file with the year name, and then copies that day’s data into it. Otherwise it just copies/concatenates that day’s data into the current file. Needs to run daily, before midnight.
- **year_voltage folder**
  - yearVoltageScript.sh

    If it is the first day of the year, it creates a new text file with the year name, and then copies that day’s data into it. Otherwise it just copies/concatenates that day’s data into the current file. Needs to run daily, before midnight.
- dayGnuplotScript.sh

  Moves all the day Gnuplot scripts (as seen below) to their correct locations once a day (a minute after midnight), and puts the correct date in them. Needs to run daily, a minute after midnight.
- monthGnuplotScript.sh

  Moves all the month Gnuplot scripts to their correct directory locations once a month, and puts the correct date in them. Needs to run every day, a minute after midnight.

- weekGnuplotScript.sh
  Moves al the week Gnuplot scripts to their correct directory locations once a week, and puts the correct date in them. Needs to run every day, a minute after midnight.
- yearGnuplotScript.sh
  Moves all the ear Gnuplot scripts to their correct directory locations once a year, and puts the correct date in them. Needs to run every day, a minute after midnight.
- GnuplotDayCurrent.gnu
  - Used as a template
- GnuplotDayPower.gnu
  - Used as a template
- GnuplotDayVoltage.gnu
  - Used as a template
- GnuplotMonthcurrent.gnu
  - Used as a template
- GnuplotMonthPower.gnu
  - Used as a template
- GnuplotMonthVoltage.gnu
  - Used as a template
- GnuplotWeekCurrent.gnu
  - Used as a template
- GnuplotWeekPower.gnu
  - Used as a template
- GnuplotWeekVoltage.gnu
  - Used as a template
- GnuplotYearCurrent.gnu
  - Used as a template
- GnuplotYearPower.gnu
  - Used as a template
- GnuplotYearVoltage.gnu
  - Used as a template

Table 6- Layout of the Online Portal

Table 6 shows the layout of the online portal – folders are indicated using bold, scripts using underlines, and additional information regarding scripts are also included. The website, its data, scripts and folder were all stored under a public_html folder on a unix server. Under the public_html folder, were different kinds of files and folders. The first group of files was the folders group, as is shown in Table 6. There was a folder for each type of data being deduced from the measurements, which includes the voltage, current and power. Additionally, there was a folder for each time period of data – day, week, month and year. In each folder, the corresponding data was stored. Additionally, each individual folder for the week, month and year data included an additional script. These script were ran once a day, right after midnight, and they would import last days data from the day folders into the week, month and year folders. If the next week, month, or year occurred, these scripts generated new files in which to store the new data.

The Gnuplot script type group shown in Table 6 included all of the individual Gnuplot scripts for each type of data and time frame. When the Gnuplot script ran, the name of the file it had to plot included a date for the days, number of week out of the year, number of month, or year in the title of the file, as well as the type of data – voltage, current and power. An example of this would be 03_03_11_current.txt – this included a days' worth of data for the current measured on 3/3/2011. Another example would be week08current.txt – this included data measured for the 8th week of the year for the current coming out of the turbine. This is how different files were distinguished from each other. Since the names of the files were dynamic, there needed to be a way to change the file names being plotted every day inside of the Gnuplot scripts. For this reason, the general Gnuplot script files were stored in the public_html folder, which did not include dates and file types to be plotted.

The shell script files are a group of files that run every night. Depending on the date, the scripts copy the general Gnuplot scripts from the main public_html into the individual folders. It also includes the name of the file it needs to plot. For example, in the system designed, weeks began on a Sunday. When the script ran, it first checked to see if it was a Sunday. If it was a Sunday, it would copy all the weekly Gnuplot scripts into the week folders. The script would modify the Gnuplot scripts to include the correct file name and type into the Gnuplot script and copy it into the correct folder.

The public_html folder also included some other files. All images that were used by the website were stored in the public_html folder, as well as all of the html files used to create the website, and the javascript files.

Finally, cron jobs were used in the unix environment to control the timing of Gnuplot scripts generated an image that was loaded on the website. Cron is a time based job scheduler, which can run any script in specified schedule. All of the shell scripts were ran once a day after midnight. The Gnuplot scripts that were written into the correct folders were ran every 5 minutes (for the day graphs), and once a day for all other types of graphs (week, day, year) after the data had been copied into the folders.

4.6.3 Final Design of Online Portal

The final design of the website is shown in Figure 40:

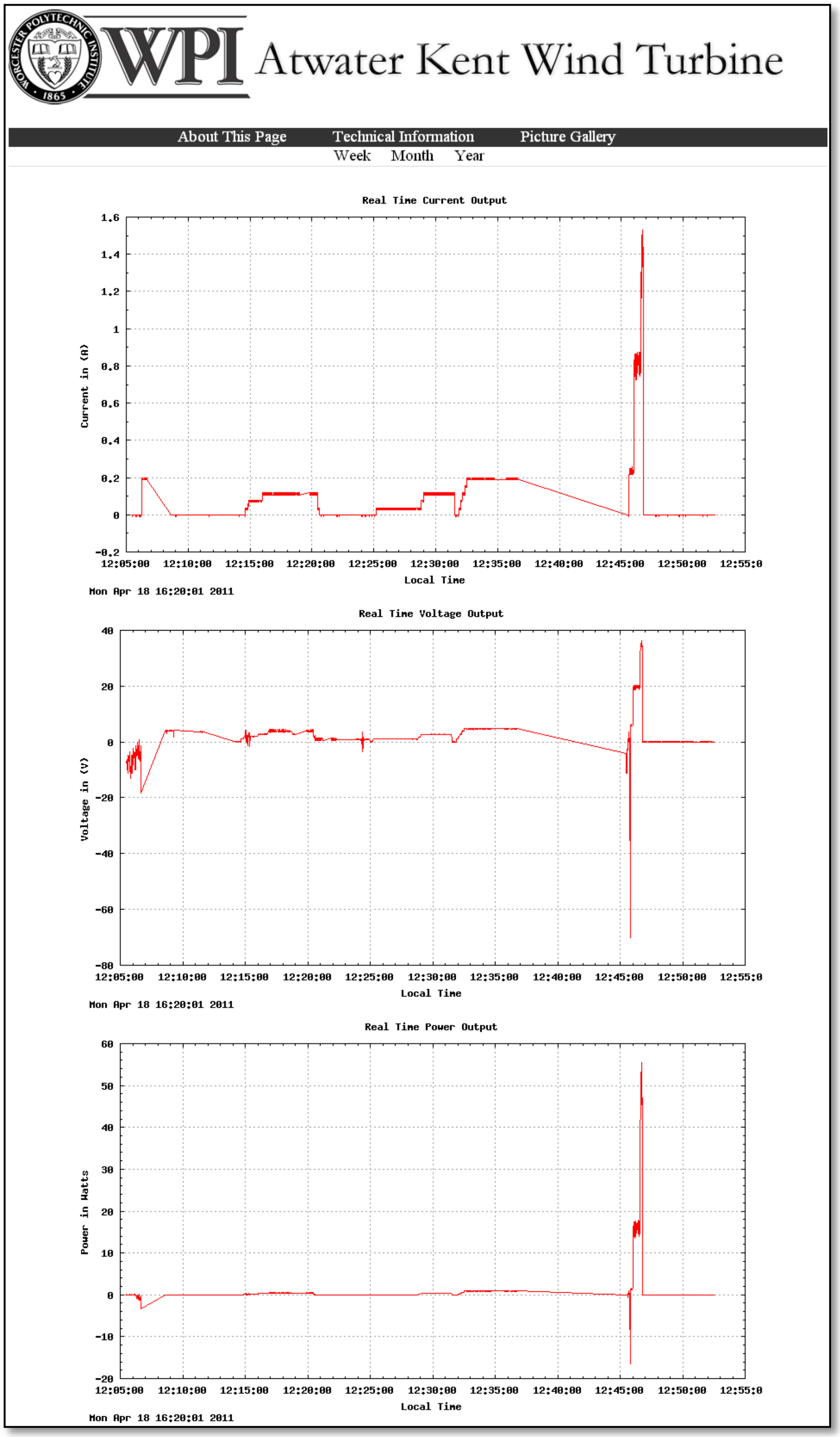


Figure 40 - Final Design of the Online Portal

The website includes the main page which is shown in Figure 40. Here, the visitor is directed to the daily real-time plots of the wind turbine. They can change to view the week, month and year as is shown in the second navigation bar on the website. Additionally, there were three other pages written that can be navigated to – the "About This Page" which included important information about the webpage. The "Technical Information" section that included technical information about the project, and a picture gallery, which included pictures of the circuits and the wind turbine. The picture gallery uses javascript to cycle through the pictures. The portal was designed with HTML and CSS.

4.7 Battery Charging Circuit

This section discusses the design of the converter used to charge a 12V battery off of the wind turbine. After initial prototyping, various issues arose with the design. This necessitated design changes that will be explained in detail. The general design that was chosen was a buck converter design, as the voltages that the circuit will be dealing with will usually be above the target 12V.

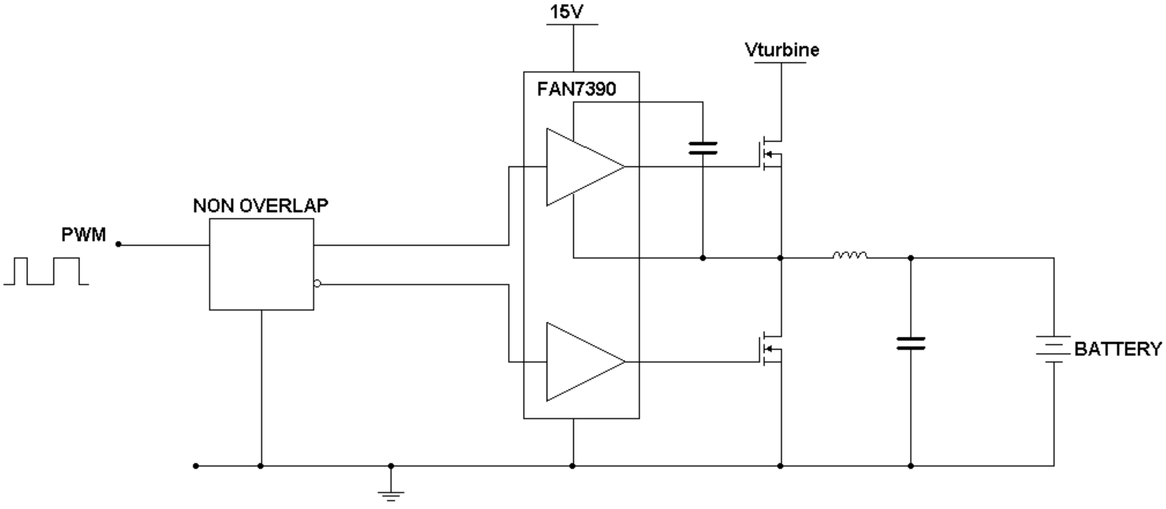


Figure 41 - Simplified Battery Charging Circuit

Figure 41 shows a simplified version of the battery charging circuit. The power comes in from the turbine, as shows as $V_{turbine}$ in the figure. Depending on the voltage the two MOSFETS will be switched with different duty cycles in order to change the output voltage to a different fraction of the input voltage. This is accomplished using a pulse-width modulated (PWM) signal. To assure that the MOSFETS switch properly, a bootstrapping circuit is required to ensure that the gate to source voltage is always positive regardless of the source voltage. This is accomplished using the FAN7390 bootstrapping IC. The non-overlap component insures that the signals to turn on the high and low side MOSFETS are always opposite of each other and that both MOSFETS will not be on at the same time.

4.7.1 Buck Converter Design

A buck converter is a switching power supply that is capable of lowering a DC voltage to a desired level. Once the general problems of the converter design had been solved, a much more detailed analysis of the design was needed in order to build the converter. Specifically, resistor, capacitor and inductor values needed to be calculated, as well as switching frequency and other parameters had to be determined.

An arbitrary decision was made that the switching frequency of the converter should be around 200kHz, and the minimum voltage ripple across the load would be 1% when the turbine outputs 40V. After these parameters were decided, calculations could begin on the duty cycle:

$$D = \frac{V_{out}}{V_{in}} = \frac{12V}{40V} = 0.3$$

Since the voltage and current at the output of the turbine were linearly related, it was possible to calculate the current corresponding to the 40V output that was calculated for:

$$\frac{V_{load}}{V_{max}} = \frac{I_{load}}{I_{max}}$$

$$I_{load} = \frac{V_{load} * I_{max}}{V_{max}} = \frac{40V * 4A}{400V} = 0.4A \ or \ 400mA$$

From that, the ripple current was calculated using the following equation:

$$I_{ripple} = \varepsilon * I_{load} = 0.01 * 400mA = 4mA$$

Finally, the inductance and capacitance of the converter was calculated based on the voltage ripple, current ripple, frequency, and duty cycle:

$$L = \frac{(V_{in} - V_{out}) * \frac{D}{f}}{I_{ripple}} = \frac{(40V - 12V) * \frac{D}{f}}{I_{ripple}} = 0.0105\ H \cong 10\ mH$$

This came out to be around 10mH, which was convenient since 10mH is a standard inductor size.

$$V_{ripple} = \frac{I_{ripple} * \left(\frac{D}{f}\right)}{C}$$

$$\therefore C = \frac{I_{ripple} * \left(\frac{D}{f}\right)}{V_{ripple}} = \frac{4mA * \left(\frac{0.3}{200kHz}\right)}{0.4V} = 15\ nF$$

Once again, the calculations led to a standard component size which simplifies the construction of the converter, since only a single component was required instead of a combination of two or more components.

4.7.2 Buck Converter Simulations

After the values were calculated, a simplified version of the converter was simulated to test whether the converter would operate within specifications. Figure 42 shows the circuit diagram used for the simulations in Multisim along with the pertinent input and output values.

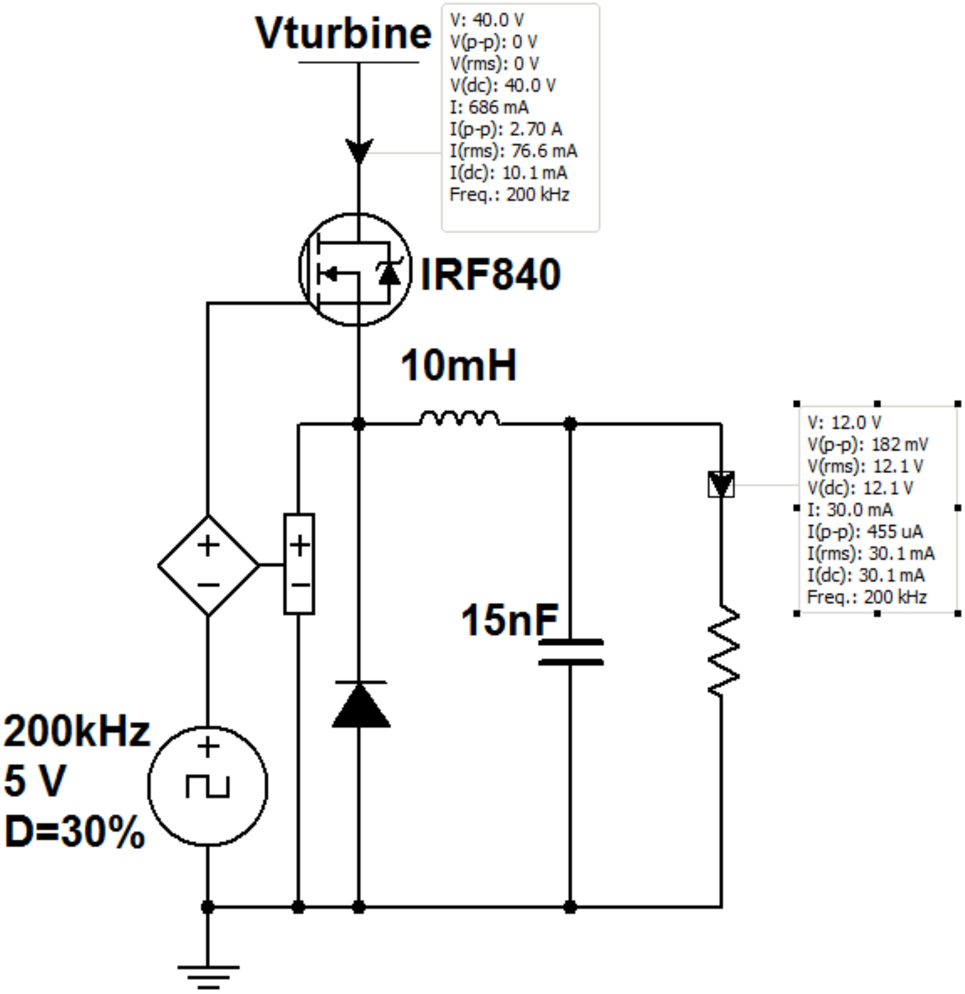


Figure 42 – Circuit Diagram used for Simulation

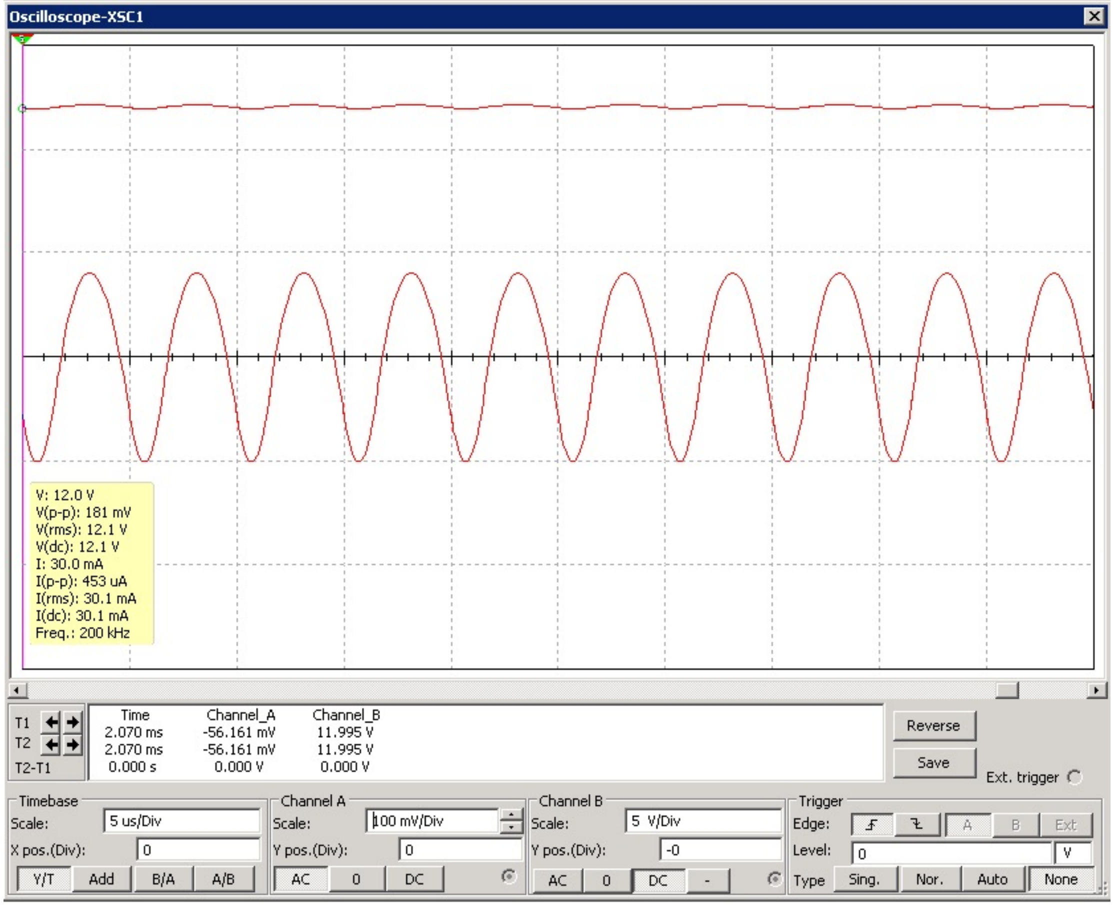
Once these results were obtained, the percent error due to both the voltage and current ripple were calculated:

$$\varepsilon_{voltage} = \frac{V_{ripple}}{V_{load}} * 100\% = \frac{182mV}{12V} * 100\% = 1.52\%$$

$$\varepsilon_{current} = \frac{I_{ripple}}{I_{load}} * 100\% = \frac{455\mu A}{30mA} * 100\% = 1.52\%$$

Both errors were the exact same value, which reaffirmed that they were dependent upon each other, though both errors were higher than expected. This is most likely due to the oversimplification of the converter circuit. But even so, 1.5% ripple is well within the tolerable range of the devices that will be used in the buck converter.

The timing diagrams of the voltage and current ripple were also prepared. In both images, the same signal is shown twice on the oscilloscope: once to show DC offset, and again zoomed in the show the ripple.



V: 12.0 V
V(p-p): 181 mV
V(rms): 12.1 V
V(dc): 12.1 V
I: 30.0 mA
I(p-p): 453 uA
I(rms): 30.1 mA
I(dc): 30.1 mA
Freq.: 200 kHz

```
       Time        Channel_A     Channel_B
T1     2.070 ms    -56.161 mV    11.995 V
T2     2.070 ms    -56.161 mV    11.995 V
T2-T1  0.000 s     0.000 V       0.000 V
```

Timebase
Scale: 5 us/Div
X pos.(Div): 0

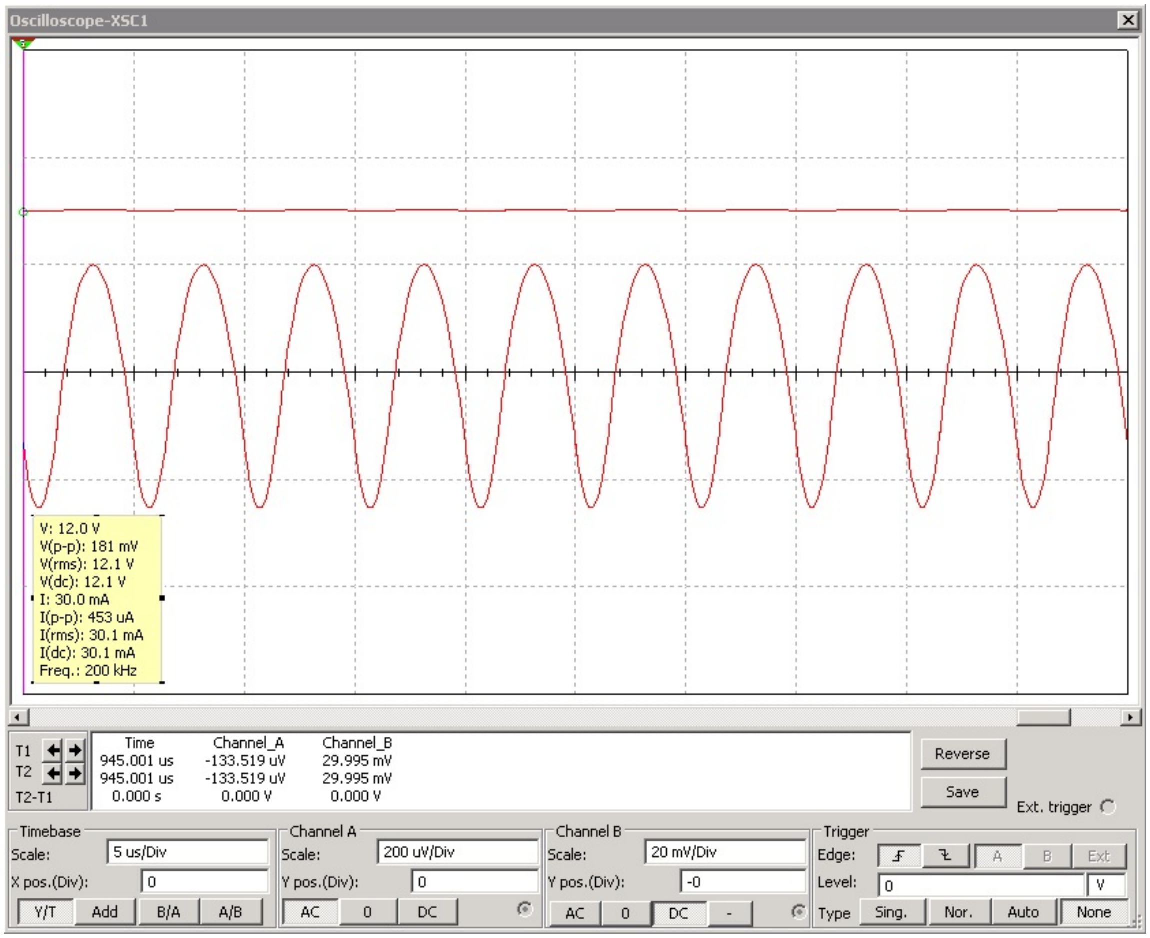
Channel A
Scale: 100 mV/Div
Y pos.(Div): 0

Channel B
Scale: 5 V/Div
Y pos.(Div): -0

Trigger
Level: 0 V

Figure 43 – Voltage Ripple Simulation

In Figure 43, the voltage oscillogram shows a DC bias level of 11.995 V or effectively 12V, indicating that the target was met. When the coupling was then changed to AC and the V/div was increased, the ripple was clearly seen.



V: 12.0 V
V(p-p): 181 mV
V(rms): 12.1 V
V(dc): 12.1 V
I: 30.0 mA
I(p-p): 453 uA
I(rms): 30.1 mA
I(dc): 30.1 mA
Freq.: 200 kHz

```
        Time         Channel_A      Channel_B
T1      945.001 us   -133.519 uV    29.995 mV
T2      945.001 us   -133.519 uV    29.995 mV
T2-T1   0.000 s      0.000 V        0.000 V
```

Timebase Scale: 5 us/Div
X pos.(Div): 0
Channel A Scale: 200 uV/Div
Y pos.(Div): 0
Channel B Scale: 20 mV/Div
Y pos.(Div): -0
Trigger Level: 0 V

Figure 44 – Current Ripple Simulation

In Figure 44, the oscillogram shows the DC bias at 29.995 mA ≈ 30 mA. The sinusoidal ripple of 453 μA is shown below the DC coupled signal. This indicates that the simulations meet the needed specification.

4.7.3 PWM Signal Generation

To generate the desired pulse width modulated signal, an LM555 timer IC and an LM311 comparator IC were used. The LM555 created a triangle waveform that was then compared against a variable DC voltage using the LM311. The LM555 will ideally create a triangle wave with a frequency equal to:

$$f = \frac{1.44}{(R_a + 2R_b)C}$$

For this, 200Ω resistors and 0.01μF capacitors were chosen. This resulted in a frequency of:

$$f = \frac{1.44}{(R_a + 2R_b)C} = \frac{1.44}{(\{200 \pm 5\%\} + 2 * \{200 \pm 5\%\})\{0.01\mu \pm 10\%\}} = 240kHz \pm {\sim}17\%$$

The ideal frequency of 200kHz was overshot because of known non-linearity in the LM555 at high frequencies.

4.7.4 Non-overlap Circuit

To ensure that both MOSFETs were never on at the same time, a break-before-make, or non-overlap circuit was constructed. The danger of having both MOSFETs on at the same time is that it would create a direct path between the turbine voltage and ground. By implementing a non-overlap circuit instead of a simple inverter, the active signal is guaranteed to go low before the inactive signal is pulled high. Figure 45 shows the circuit used for the non-overlap signal generation.

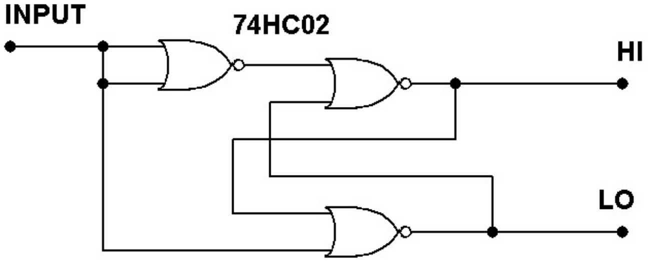


Figure 45 - Break-Before-Make Circuit

4.7.5 Bootstrapping Circuit

The purpose of the bootstrapping circuit is to ensure the proper operation of the high side MOSFET under a varying supply voltage. Since MOSFETs turn on when the gate to source voltage exceeds a certain threshold, it was necessary to ensure that the gate voltage is always above the source voltage when the MOSFET had to be on.

$$V_{gs} \geq V_{th}$$
$$\therefore V_{gate} \geq V_{th} + V_{source}$$

In most applications, this is not a problem, but when using a MOSFET as a high voltage switch, complications arise. Since the MOSFET was simply being used as a switch, ideally, there would be no voltage drop across the MOSFET and the source voltage would be equal to the drain voltage. In reality, there was a resistance associated with the drain to source pathway while the MOSFET was on, but it is usually of negligible resistance. For the IRF 840, the “on” resistance, rds(ON), was equal to 0.850Ω. Since the drain is connected directly to the output of the turbine, and there was little voltage drop across the MOSFET, the source voltage could reach as high as 400V. Since it is impractical to use a 400V voltage source to supply the gate drive signal, bootstrapping circuits were researched and implemented.

A bootstrapping circuit will take the original gate signal and float it on top of a voltage equal to the source voltage. This ensures that V_{gs} is equal to 0V when the MOSFET should be off or 5V (or 15V depending on the application) when the MOSFET should be on, regardless of the supply voltage from the turbine.

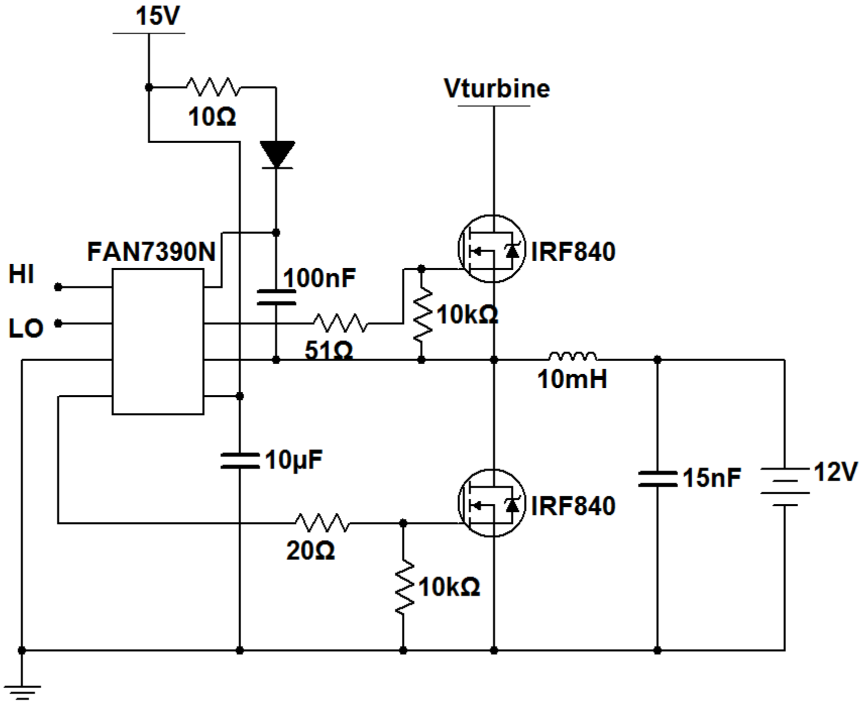


Figure 46 - Bootstrapping Circuit

The FAN7390 by Fairchild Semiconductors was chosen to be the bootstrapping solution because of its capability to handle up to 600VDC along with its ease of use and relatively low price. The FAN7390 takes two input signals, the high and low-side gate input signals, and then shifts them to the desired levels. The circuit diagram of the FAN7390 is shown in Figure 46.

5.0 Results

Once all the hardware and software for the measuring circuit system and the online portal was complete, measuring was done to establish the amount any errors in measurements obtained from the system. Based on the data from these tests, the overall design was evaluated based on the objectives set forth in the beginning of this report. Finally, after completing the battery charging circuit, tests were conducted to ensure the proper operation of all constituent parts as well as the system as a whole.

5.1 Measurement Circuit and Online Portal

After the construction of the circuit, the circuit was tested using a variable output voltage source, which was then rectified using an H bridge to supply direct current. Using this configuration, the variable voltage supply was able to supply up to 100VDC with a maximum of 10A. The circuit was tested by supplying a known, measured voltage and current, and then by measuring the output directly from both the Hall Effect sensor as well at the optocoupler. Additionally, the values calculated by LabVIEW for the input voltage and current were also measured, in order to see if they lie within the given error percentage.

5.1.2 Optocoupler Result Error Estimation

Initially, the measurements were taken for the input voltage. Using the variable voltage supply, inputs ranging from 5 to 100V were used, to try and mimic the range of voltages that may appear from the wind turbine. While the wind turbine can supply up to 400VDC, 100VDC was the highest voltage supply that was available for testing purposes.

Table 7 - Voltage Input and Output Values for LabVIEW

V_{IN}	$V_{LabView}$
0.000	0.11950
5.153	5.21041
10.458	10.47900
15.029	14.79700
20.196	19.99460
25.157	24.72150
30.304	29.99000
34.770	34.54800
40.188	39.90550
45.417	45.18310
50.263	49.90090
54.998	54.97900
60.109	59.77680
65.072	64.37430
70.848	69.97170
75.088	74.44690

80.342	79.32740
85.248	84.36510
90.159	88.92300
95.307	94.04700
100.127	98.75850

Table 7 shows the measurements that were found for the Optocoupler. The input voltage was measured along with the recorded LabVIEW measurement for each input voltage. Next, the recorded LabVIEW voltage was plotted against the input voltage as shown in Figure 47 below.

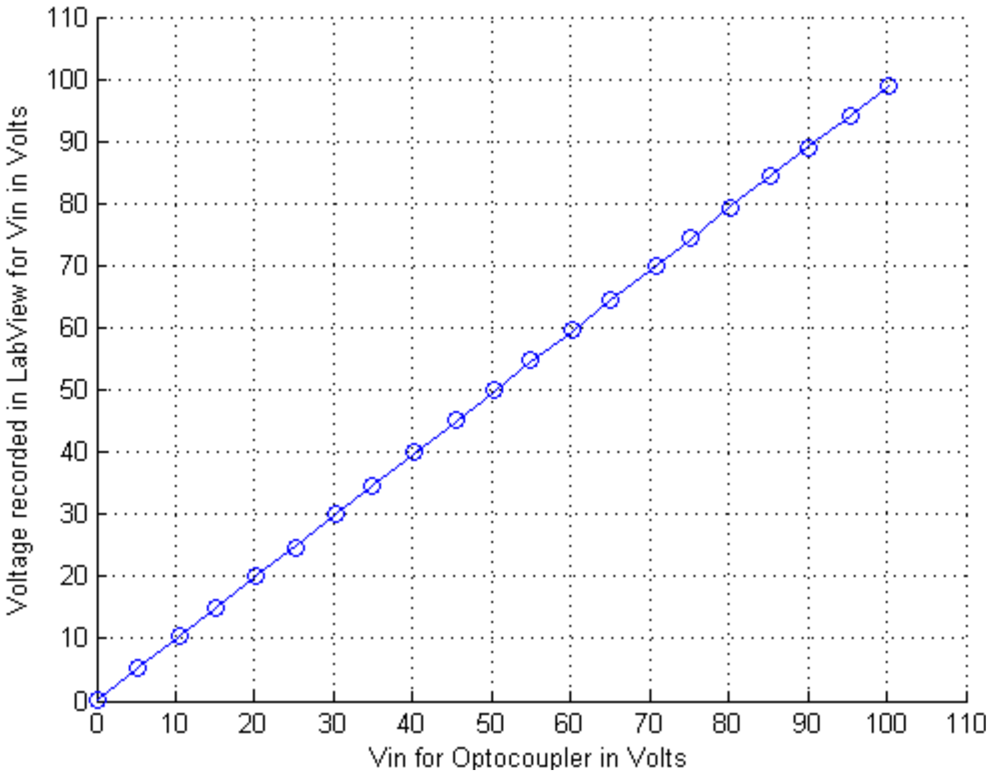


Figure 47 - Input Voltage versus LabVIEW Measurement for the Optocoupler

Figure 47 shows the line of best fit, along with the individual data points for the LabVIEW measurements versus the input voltage. With the line of the best, a model can be deduced for the LabVIEW measurements:

$$V_{LabView} = m_L V_{IN} + b_L.$$

Using this model, $\mathrm{V_{LabVIEW}}$ can be found for any value of $\mathrm{V_{IN}}$. The line of best fit was calculated in matlab using the polyfit function, which returns the value for $\mathrm{m_L}$ and $\mathrm{b_L}$. The values were found to be $\mathrm{m_L}$=0.9867 and $\mathrm{b_L}$=0.1765. Therefore, the model equation was found to be:

$$V_{LabView} = 0.9868V_{IN} + 0.1664.$$

Next, using the model equation, an estimated $\mathrm{V_{IN}}$, noted by $\widehat{\mathrm{V}}_{\mathrm{IN}}$ can be found:

$$\widehat{V}_{IN} = \frac{V_{LabView} - b_L}{m_L} = \frac{V_{LabView} - 0.1664}{0.9868}$$

To continue with the error estimation, values for $\widehat{\mathrm{V}}_{\mathrm{IN}}$ were calculated using the measured value for $\mathrm{V_{LabVIEW}}$ and the model equation found.

Table 8 – Model Estimation $\widehat{V}_{IN}$ value calculation

V_{IN}	$V_{LabView}$	$\widehat{V}_{IN}$
0.00000	0.199500	0.033540
5.15300	5.210410	5.111480
10.4580	10.47900	10.45055
15.0290	14.79700	14.82631
20.1960	19.99460	20.09343
25.1570	24.72150	24.88356
30.3040	29.99000	30.22254
34.7700	34.54800	34.84151
40.1880	39.90550	40.27067
45.4170	45.18310	45.61887
50.2630	49.90090	50.39978
54.9980	54.97900	55.54580
60.1090	59.77680	60.40778
65.0720	64.37430	65.06678
70.8480	69.97170	70.73906
75.0880	74.44690	75.27412
80.3420	79.32740	80.21990
85.2480	84.36510	85.32499
90.1590	88.92300	89.94386
95.3070	94.04700	95.13640
100.127	98.75850	99.91092

Table 8 shows the calculated values found for $\widehat{\mathrm{V}}_{\mathrm{IN}}$. Using these values, the error could be calculated and plotted.

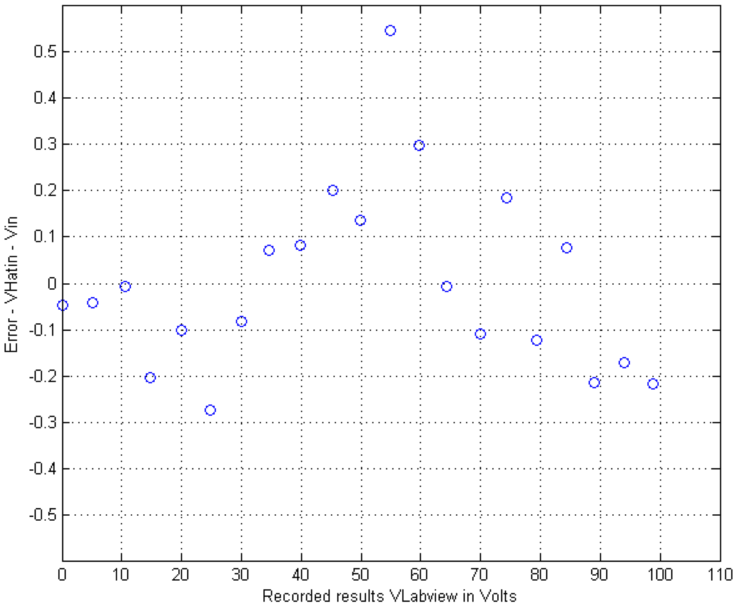


Figure 48 - Error found the Model Equation versus the input Voltage

Figure 48 shows the error found using the model equation versus the input voltage. The error was calculated by subtracting the known $\mathrm{V_{IN}}$ values from the calculated values for $\widehat{\mathrm{V}}_{\mathrm{IN}}$ from Table 8.

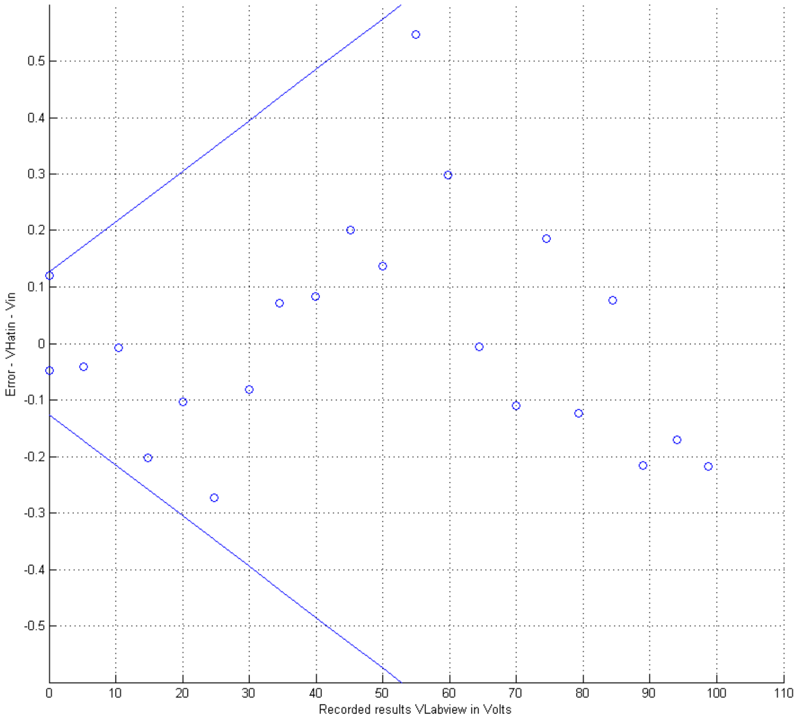


Figure 49 - Voltage Error Bound

Figure 49 shows the error in which the voltage is bound by in LabVIEW. By finding the errors in Figure 48, the outer bound for the outmost data points was calculated. Here, the error is bound by the equation $y = -0.0090x - 0.1250$ and $y = 0.0090x + 0.1250$. The percent error for the voltage sensing was then concluded to be ±0.09%V_{IN} ±12.5mV, which was well within the wanted error of less than 1%.

5.1.2 Hall Effect Sensor Result Error Estimation

When the measurements were taken for the input voltage, the input current measurements were also taken. Using the variable voltage supply, inputs ranging from 5 to 100V were used, and a current of up to 0.38266A was drawn.

Table 9 – Current Input and Output Values for LabVIEW

I_{IN}	$I_{LabView}$
0.000000	0.002578
0.106126	0.114369
0.162960	0.174163
0.189200	0.194094
0.211420	0.214025
0.227570	0.253888
0.241900	0.253888
0.252770	0.273819
0.265720	0.273819
0.276870	0.293750
0.287290	0.313681
0.297140	0.313681
0.307280	0.333612
0.316760	0.333612
0.328240	0.353544
0.336300	0.353544
0.346170	0.373475
0.355630	0.373475
0.365000	0.393406
0.374040	0.413337
0.382660	0.413337

Table 9 shows the measurements that were found for the Hall Effect sensor. The input current was measured along with the corresponding recorded LabVIEW measurement for each data point. Next, the recorded LabVIEW current was plotted against the input current as shown in Figure below.

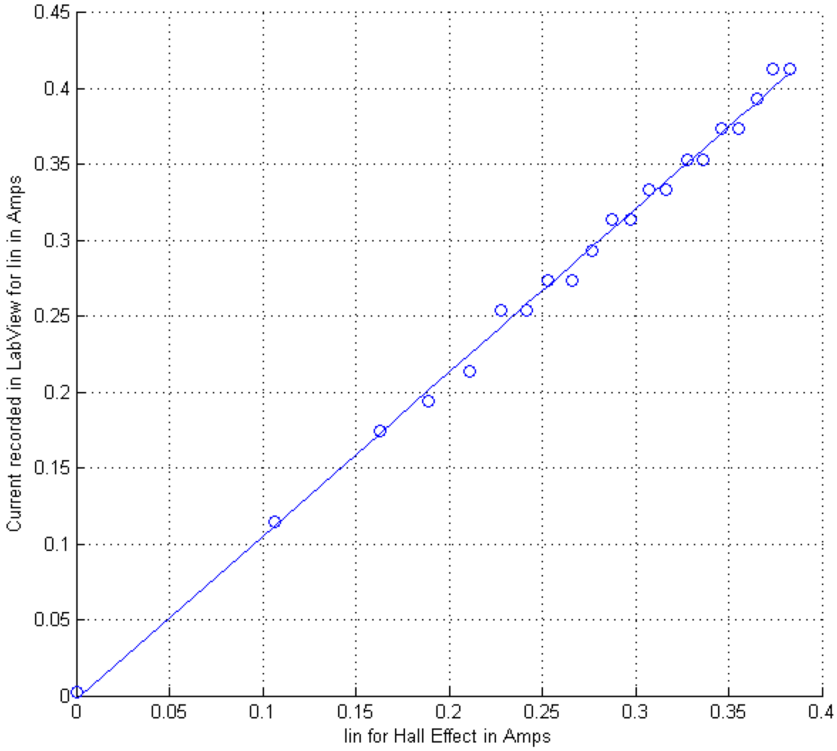


Figure 50 - I_{IN} versus $I_{LabView}$ for Current Measurements

Figure 50 shows the line of best fit, along with the individual data points for the LabVIEW measurements versus the input voltage. With the line of the best, a model can be deduced for the LabVIEW measurements:

$$I_{LabView} = m_L I_{IN} + b_L.$$

Using this model, $I_{LabVIEW}$ can be found for any value of I_{IN}. The line of best fit was calculated in matlab using the polyfit function, which returns the value for m_L and b_L. The values were found to be m_L=1.0927 and b_L=-0.0069. Therefore, the model equation was found to be:

$$I_{LabView} = 1.0787 I_{IN} - 0.0027$$

Next, using the model equation, an estimated I_{IN}, noted by $\hat{I}_{IN}$ can be found:

$$\hat{I}_{IN} = \frac{I_{LabView} - b_L}{m_L} = \frac{V_{LabView} + 0.0027}{1.787}$$

To continue with the error estimation, values for $\hat{I}_{IN}$ were calculated using the measured value for $I_{LabVIEW}$ and the model equation found.

Table 10 - Model Estimation $\hat{I}_{IN}$ value calculation

I_{IN}	$I_{LabView}$	$\hat{I}_{IN}$
0	0.002578	0.0048929
0.106126	0.114369	0.1085279
0.16296	0.174163	0.1639594
0.1892	0.194094	0.1824363
0.21142	0.214025	0.2009131
0.22757	0.253888	0.2378678
0.2419	0.253888	0.2378678
0.25277	0.273819	0.2563447
0.26572	0.273819	0.2563447
0.27687	0.29375	0.2748215
0.28729	0.313681	0.2932984
0.29714	0.313681	0.2932984
0.30728	0.333612	0.3117753
0.31676	0.333612	0.3117753
0.32824	0.353544	0.3302531
0.3363	0.353544	0.3302531
0.34617	0.373475	0.34873
0.35563	0.373475	0.34873
0.365	0.393406	0.3672068
0.37404	0.413337	0.3856837

Table 10 shows the calculated values found for $\hat{I}_{IN}$. Using these values, the error could be calculated and plotted.

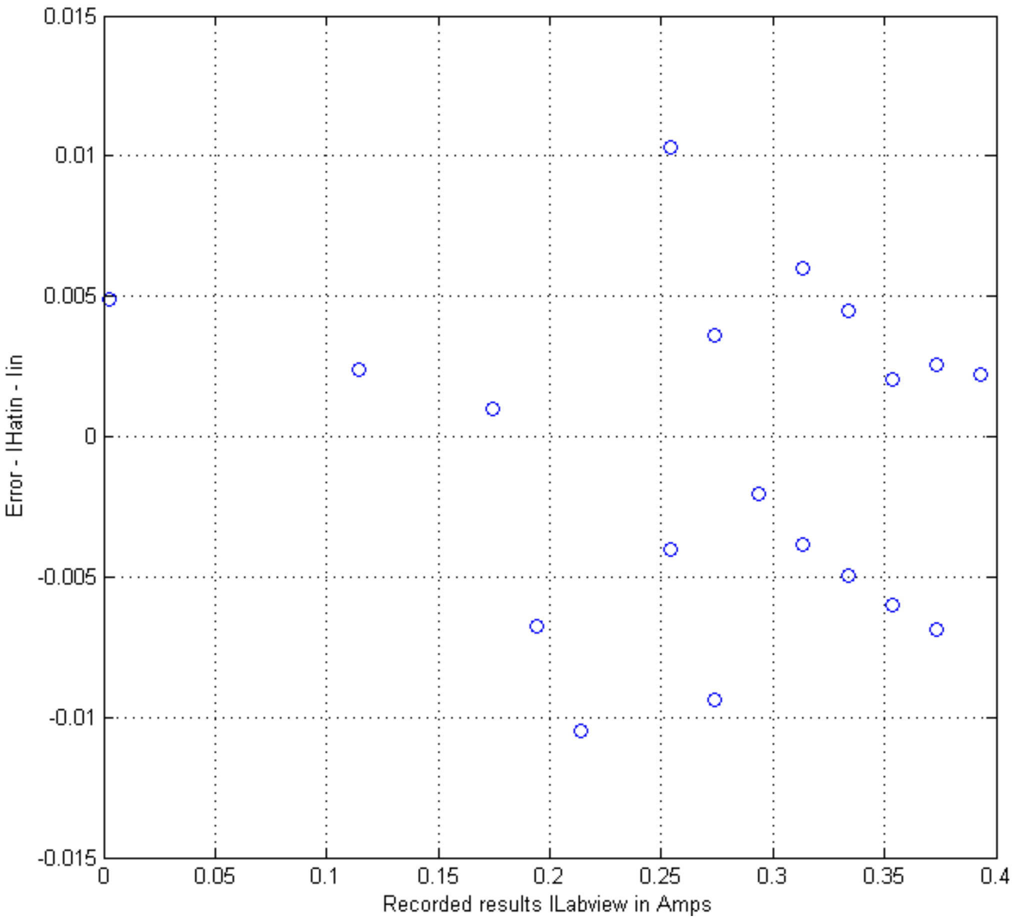


Figure 51 - Error found using the Model Equation versus the Input Current

Figure 51 shows the error found using the model equation versus the input voltage. The error was calculated by subtracting the known I_{IN} values from the calculated values for $\hat{I}_{IN}$ from Table 10.

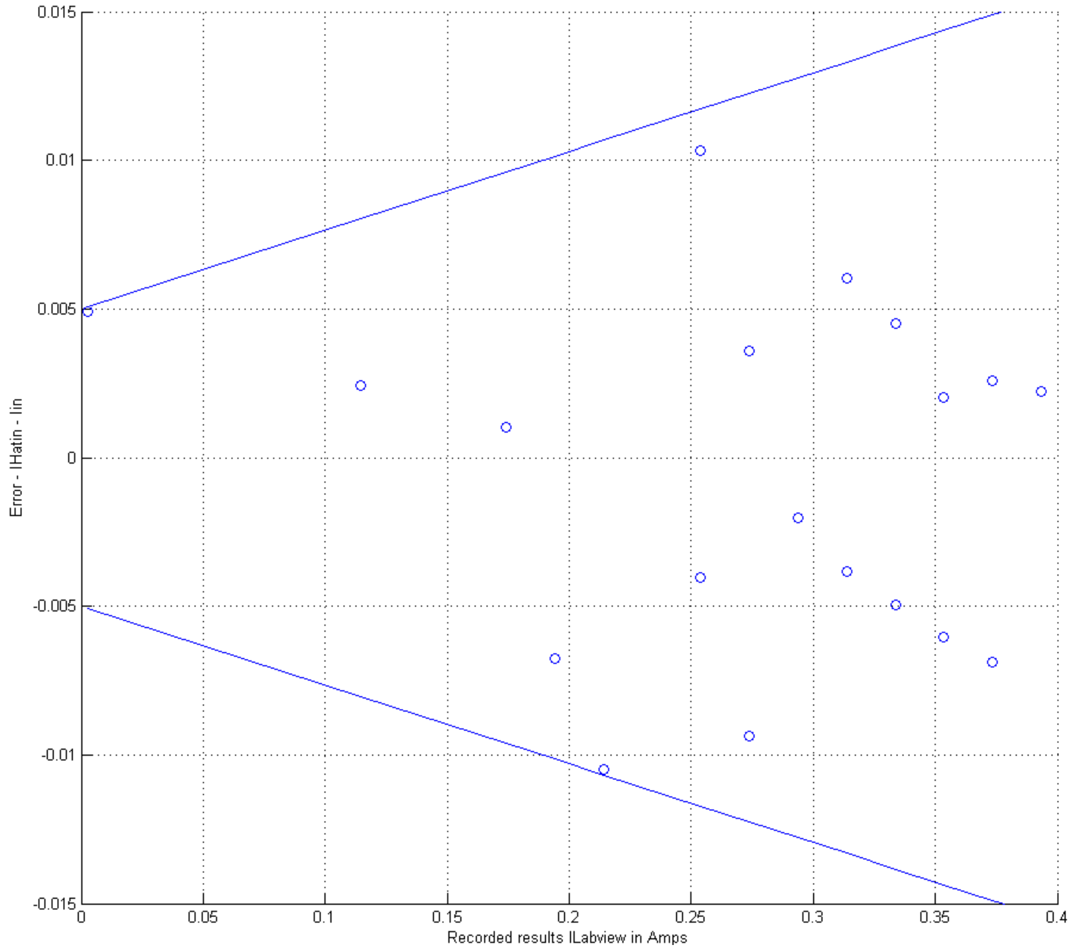


Figure 52 - Current Error Bound

Figure 52 shows the error in which the voltage is bound by in LabVIEW. By finding the errors in Figure 51, the outer bound for the outmost data points was calculated. Here, the error is bound by the equation $y = -0.0265x - 0.005$ and $y = 0.0265x + 0.005$. The percent error for the current sensing was then concluded to be ±0.0265%V_{IN} ±5mV, which was well within the wanted error of less than 1%.

5.2 Finalized Measuring Circuit and PCB Design

The prototype sensor circuit was built using a standard protoboard. For the final design of the project a printed circuit board (PCB) was designed and built. While there is quite a large variety of PCB design software available, the software that was ultimately used was Eagle due to the fact that is was free and met the needs of the project.

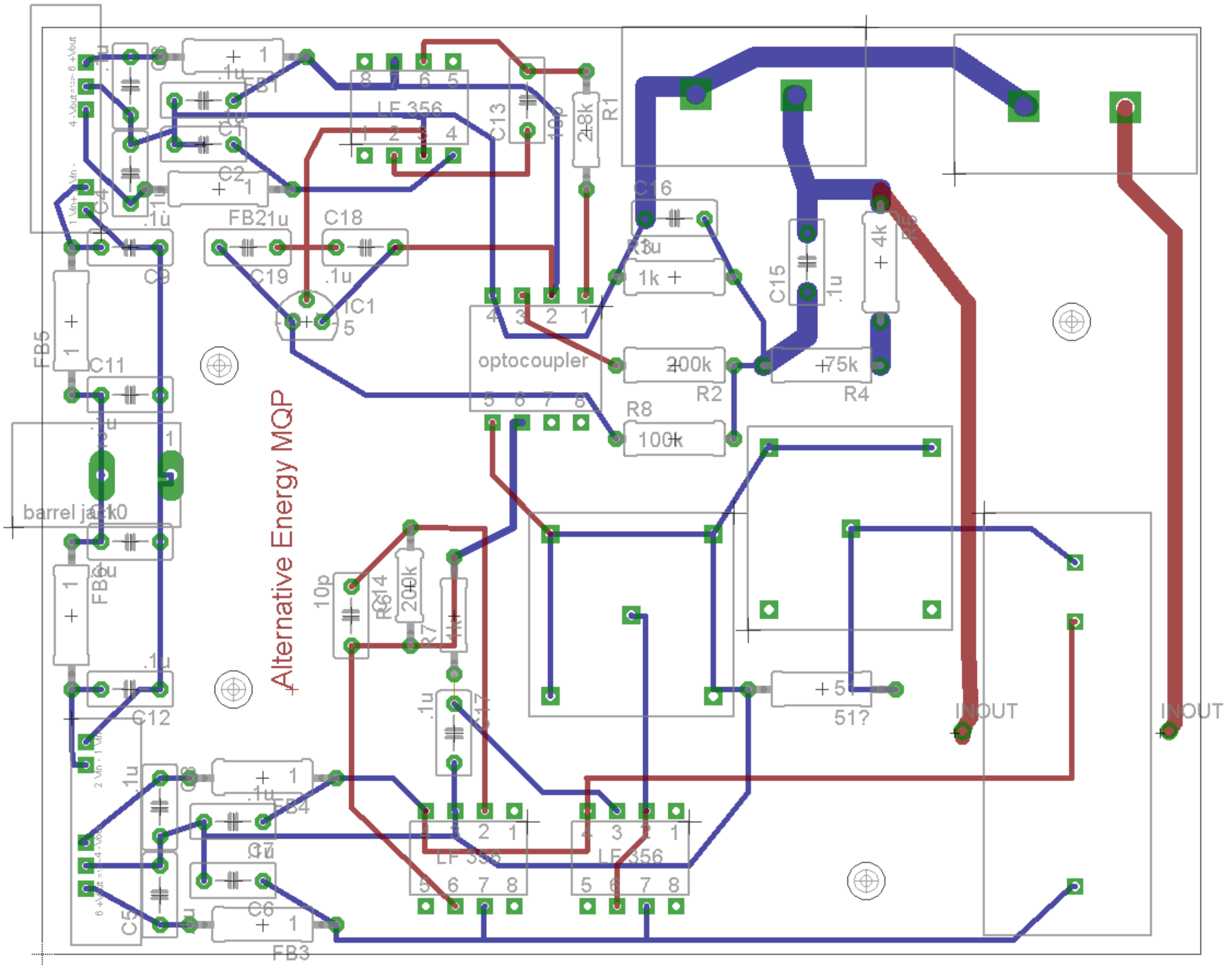


Figure 53 - Final PCB Design for the Measuring Circuit

The PCB, as shown in Figure 53, was designed as a two layer board. The power components of the board were made for the bottom layer, which the ICs and the Hall Effect sensor were designed to lie on the top of the layer. Additionally, room was made so that holes could be drilled as a way to be able to place the board on a place holder.

5.3 Battery Charging Circuit

After the circuit was constructed, each part was tested and compared to the calculated and simulated results. Oscillograms for each stage of the converter will be shown from both the actual circuit, and from Multisim. The oscillograms obtained from the circuit were recorded using a Tektronix TDS 2004B.

5.3.1 PWM Generation

The PWM signal was generated using two subsystems; an LM555 timer generating a triangle wave, and LM311 comparator that compares the steady triangle wave to a variable DC voltage to produce the pulse width modulated square wave.

5.3.2 LM555 Timer

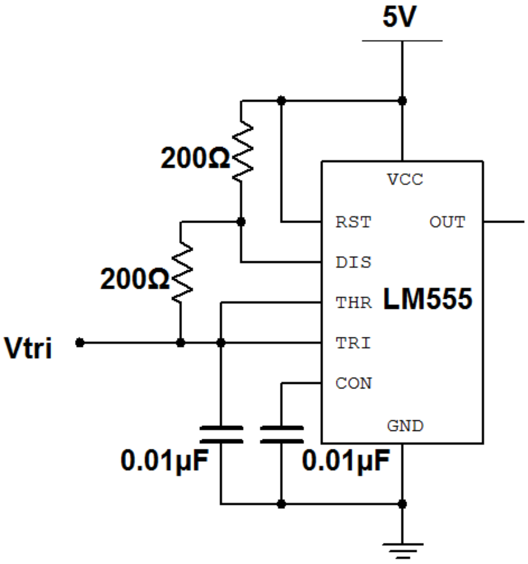


Figure 54 - Triangle Wave Generation Circuit

Figure 54 shows the circuit that was built around the LM555 timer. Based on the components and their associated allowed error, the frequency was calculated and found to be:

$$f = \frac{1.44}{(R_a + 2R_b)C} = \frac{1.44}{(\{200 \pm 5\%\} + 2 * \{200 \pm 5\%\})\{0.01\mu \pm 10\%\}} = 240kHz \pm {\sim}17\%$$

When the triangle wave was measured with an oscilloscope, the frequency was read as 162 kHz as shown in Figure 55.

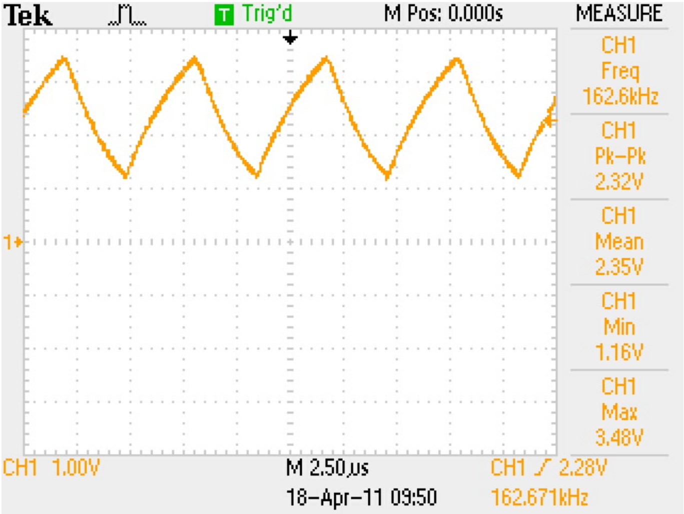


Figure 55 - Triangle Wave produced by LM555 Timer

Although this is outside of the expected error for the frequency, it can be expected since the LM555 loses linearity at higher frequencies. And while the target of 200 kHz was also not reached, the final frequency was within 20% which is an acceptable error. This frequency will carry through the entire circuit, giving the buck converter an operating frequency of 162kHz.

5.3.3 LM311 Comparator

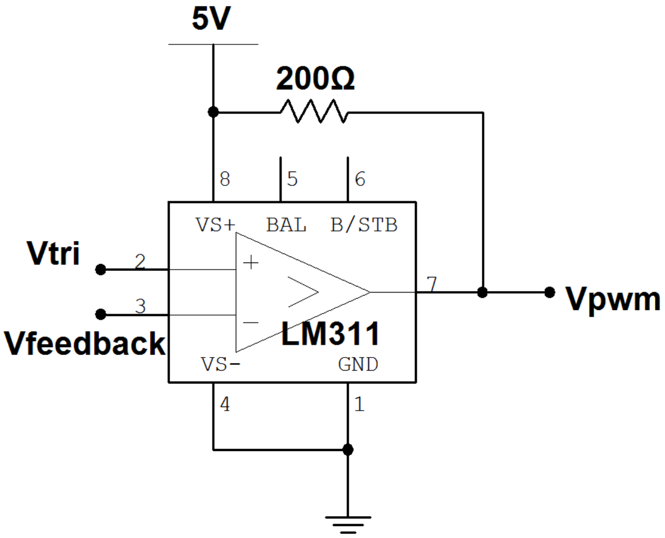
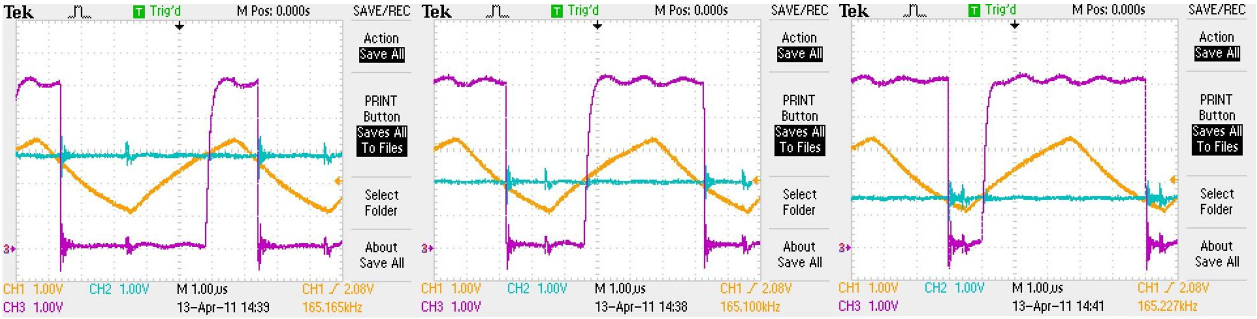


Figure 56 – Square Wave Generation Circuit

Figure 56 shows the LM311 comparator that was used to generate the pulse width modulated square wave signal used by the FAN7390. Below are multiple oscillograms showcasing how the LM311 circuit behaves. Whenever the voltage from V_{tri} crosses the voltage from $V_{feedback}$, the LM311 switches between high and low states. The three oscillograms show $V_{feedback}$ at 3V, 2V, and 1.5V respectively.



CH1 1.00V CH2 1.00V M 1.00µs CH1 ∕ 2.08V
CH3 1.00V 13–Apr–11 14:39 165.165kHz

CH1 1.00V CH2 1.00V M 1.00µs CH1 ∕ 2.08V
CH3 1.00V 13–Apr–11 14:38 165.100kHz

CH1 1.00V CH2 1.00V M 1.00µs CH1 ∕ 2.08V
CH3 1.00V 13–Apr–11 14:41 165.227kHz

Figure 57 - LM311 Oscillograms

The oscillograms indicate that the circuit was behaving as needed.

5.3.4 Non-Overlap Circuit

The non-overlap circuit shown in Figure 58, was used to generate both the high and low side driving signals and ensure that both are not on at the same time.

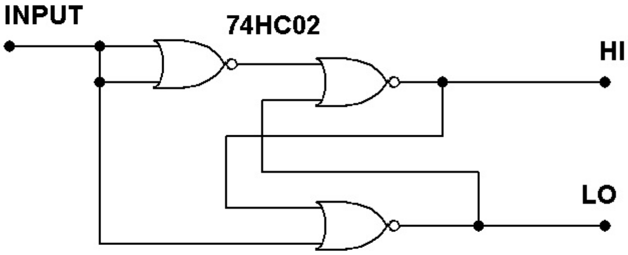
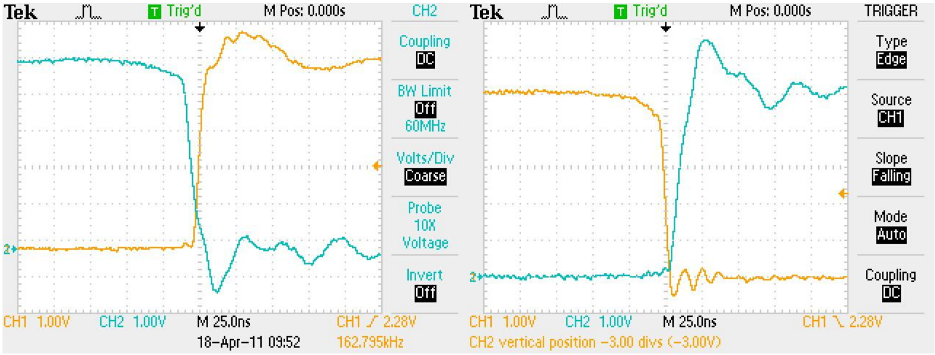


Figure 58 - Non-overlap Signal Generation

In Figure 59 the oscillograms show that the active signal goes low before the inactive signal goes high. This ensures that the two MOSFETs in the buck converter never short the output from the turbine directly to ground.



CH2
Coupling DC
BW Limit Off 60MHz
Volts/Div Coarse
Probe 10X Voltage
Invert Off

CH1 1.00V CH2 1.00V M 25.0ns CH1 ∕ 2.28V
18–Apr–11 09:52 162.795kHz

TRIGGER
Type Edge
Source CH1
Slope Falling
Mode Auto
Coupling DC

CH1 1.00V CH2 1.00V M 25.0ns CH1 ∖ 2.28V
CH2 vertical position −3.00 divs (−3.00V)

Figure 59 - Rising and Falling Edges of the Non-Overlap Circuit

Once again, the results indicate that the operation of the circuit was meeting the specified needs.

5.3.5 FAN7390 Gate Driver

As previously mentioned, a gate driver was needed for the circuit, and the FAN7390 IC was used to accomplish this. The circuit that is shown in Figure 60 shows how the FAN7390 IC interfaces with the rest of the buck converter circuit.

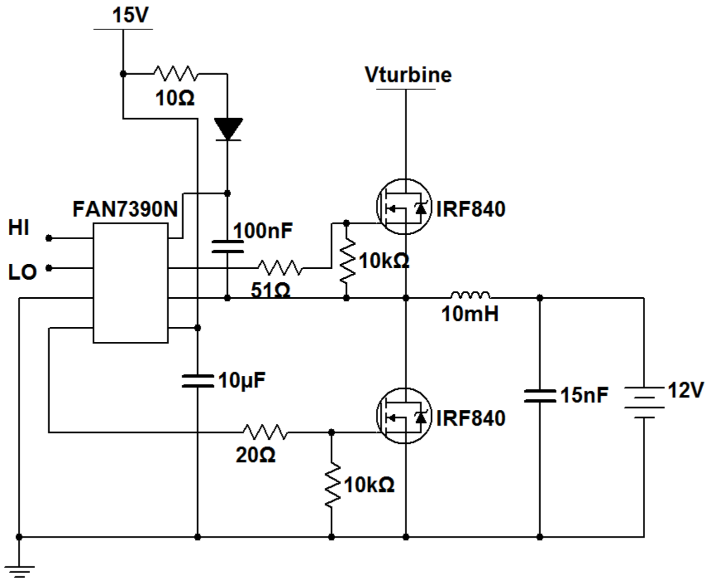


Figure 60 – Interfacing the FAN7390 with the Buck Converter Circuit

The Fairchild FAN7390 not only converts the 5V high and low gate signals to 15V, it also floats the high side signal on top of the bootstrap capacitor voltage.

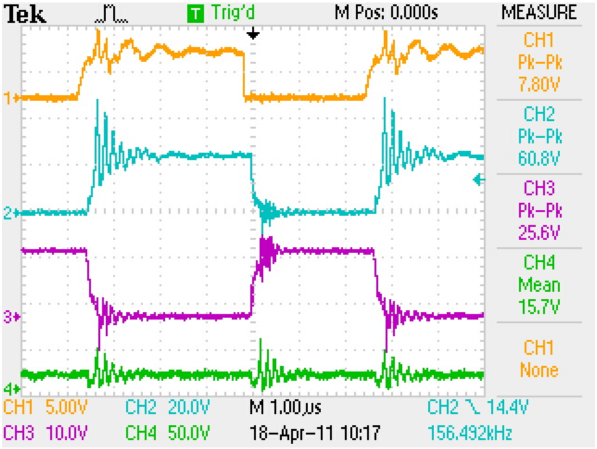


Figure 61 - Bootstrapped Signals

This is shown in Figure 61 where the input to the non-overlap circuit is shown on Channel 1 (in yellow), the high-side MOSFET gate signal on Channel 2 (in blue), the low-side MOSFET gate signal on Channel 3 (in purple), and the turbine voltage on Channel 4 (in green).

5.3.6 Buck Converter

After building the complete circuit the efficiency of the buck converter was determined. This was done by comparing the theoretical output based on the input and duty cycle of the system to the measured output. Figure 62 shows the outputs of buck converter with a duty cycle at around 68%.

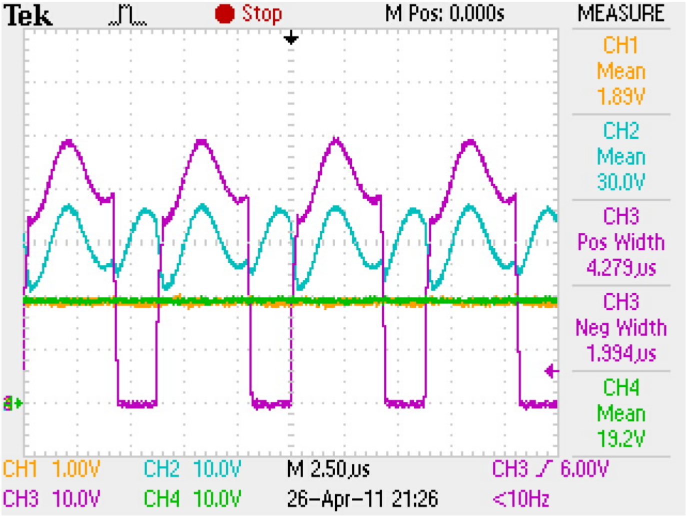


Figure 62 - 66% Duty Cycle

Figure 62 shows the feedback voltage on Channel 1 (in yellow), the turbine voltage on Channel 2 (in blue), the high-side MOSFET gate signal on Channel 3 (in purple) and the output of the buck converter on Channel 4 (in green). Using this signal, the exact duty cycle and the efficiency of the battery charging circuit was then calculated.

The exact duty is calculated to be:

$$D = \frac{t_{on}}{(t_{on} + t_{off})} * 100\% = \frac{4.279\mu s}{(4.279\mu s + 1.994\mu s)} * 100\% = 68.21\%$$

In an ideal system, this duty cycle and turbine voltage would result in an output of:

$$V_{out} = V_{in} * D = 30.0V * 68.21\% = 20.46V$$

From this ideal output, the system efficiency can be calculated:

$$efficiency = \frac{V_{out(actual)}}{V_{out(ideal)}} * 100\% = \frac{19.2}{20.46} * 100\% = 93.82\%$$

Figure 63 shows the system at roughly 50% and 33% duty cycles.

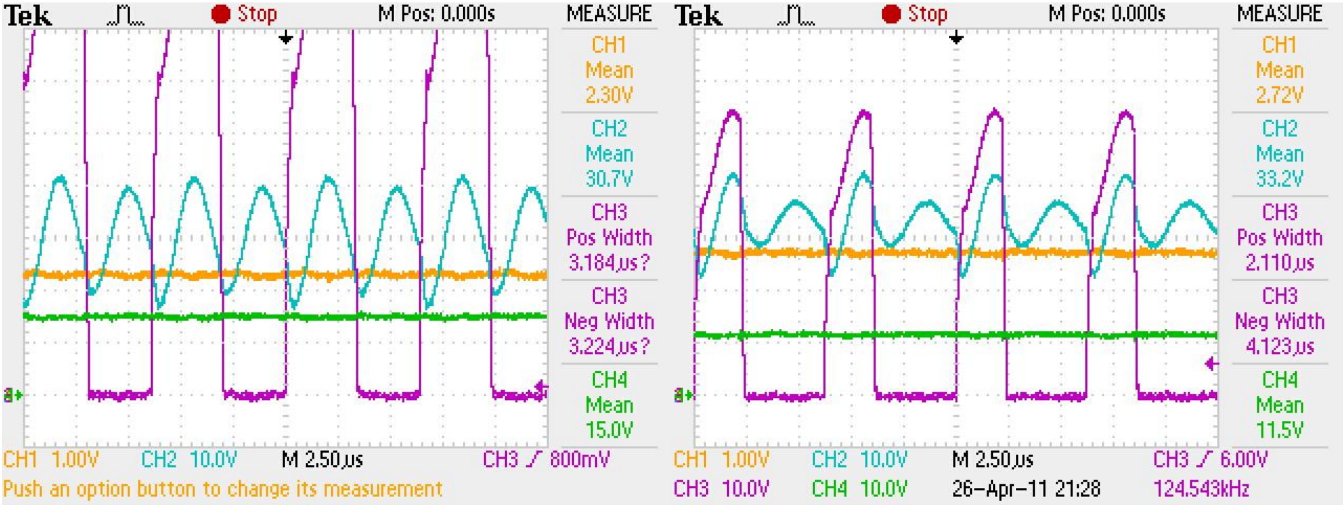


Figure 63 - 50% and 33% Duty Cycles

The same efficiency calculations are repeated for the 50% and 33% duty cycles to not only ensure that the original efficiency is correct, but to also prove that the efficiency is independent of duty cycle. The resultant efficiencies are as such:

$$efficiency_{50\%} = 98.33\%$$

$$efficiency_{33\%} = 97.73\%$$

While the efficiency does seem to vary slightly, all three calculated values are within 4% of each other. The overall system efficiency can be approximated by taking the mathematical average of the three efficiencies that were calculated. The resultant efficiency is:

$$Total\ System\ Efficiency = 96.63\%$$

This result is much greater than any initial expectations, and a very respectable efficiency for a system that is not constructed on a printed circuit board.

6.0 Conclusion

The goals set out in the beginning of the project were achieved. The online portal was configured and operational at the end of the project, and the measuring circuit was built and functional. In addition to designing and building the measuring circuit, a goal was set to achieve measurements through LabVIEW that would be bound by 1% error of the original input. This was another goal that was achieved in this project – the voltage measurements were bound by an error of ±0.09%V_{IN} ±12.5mV and the current measurements were bound by an error of ±0.0265%V_{IN} ±5mV. Finally, the battery charging circuit was tested and its functionality was verified. Additionally, the efficiency of the battery charging circuit was calculated to be 96.63%. Despite being able to achieve the goals set out at the beginning of the project, there are still some other functions that would be helpful to implement if another group would like to continue on with this or a similar project.

One of the ideas that was discussed, but not set as a goal due to the time constraints of the project was creating a switching circuit as shown in Figure 64.

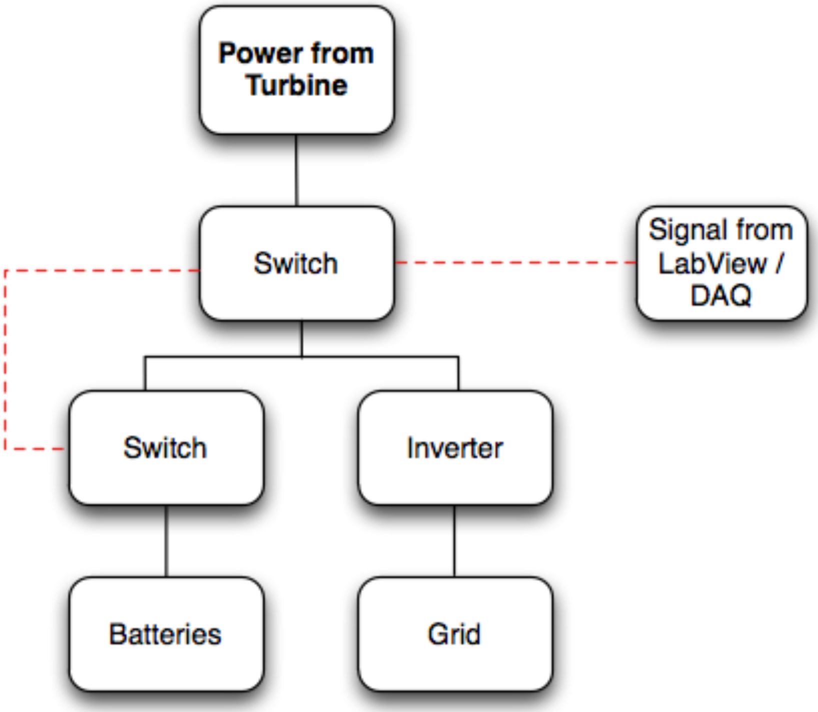


Figure 64 - Switching Circuit Block Diagram

The main goal of the charging circuit is to use the energy produced by the wind turbine to charge the two twelve volt batteries; however there are times when the batteries should not be charged. The two cases when the battery charging is stopped are when the state of charge of the batteries is at full capacity and when wind speeds are high enough to deliver power to the grid. A switch can be controlled by two different signals, which can direct the power to the correct destination. The switch changes from delivering power to the batteries to providing power to the grid. This would include using a calculation of the state of charge of the battery, and sending a signal to the switch from LabVIEW.

Another idea was to include some more information on the online portal. For example, the RPM as well as weather statistics could be integrated into the website. The weather station on the top of Atwater Kent could either be revived, or a new one could be purchased. Finally, the last idea was to have a battery state of charge integrated into the online portal. Visiting users would be able to see how much of the battery was charged and compare it to the power being generated from the wind turbine. These statistics would make for an interesting way to compare the output of the wind turbine under different conditions, and to see how efficient or inefficient it is under certain conditions.

Appendix

Appendix A – Cron Job for Online Portal Functionality

Cron command	Action	Script Affected
*/5 * * * * Gnuplot /www/wind/day_current/day_current.gnu	Runs every 5 minutes, daily	day_current.gnu
*/5 * * * * Gnuplot /www/wind/day_voltage/day_voltage.gnu	Runs every 5 minutes, daily	day_voltage.gnu
*/5 * * * * Gnuplot /www/wind/day_power/day_power.gnu	Runs every 5 minutes, daily	day_power.gnu
1 0 * * * /www/wind/dayGnuplotScript.sh	Runs once a day, a minute after midnight	dayGnuplotScript.sh
1 0 * * * /www/wind/weekGnuplotScript.sh	Runs once a day, a minute after midnight	weekGnuplotScript.sh
1 0 * * * /www/wind/monthGnuplotScript.sh	Runs once a day, a minute after midnight	monthGnuplotScript.sh
1 0 * * * /www/wind/yearGnuplotScript.sh	Runs once a day, a minute after midnight	yearGnuplotScript.sh
*/30 * * * * Gnuplot /www/wind/week_current/week_current.gnu	Runs every 30 minutes daily	week_current.gnu
*/30 * * * * Gnuplot /www/wind/week_voltage/week_voltage.gnu	Runs every 30 minutes daily	week_voltage.gnu
*/30 * * * * Gnuplot /www/wind/week_power/week_power.gnu	Runs every 30 minutes daily	week_power.gnu
5 * * * * Gnuplot /www/wind/month_current/month_current.gnu	Runs once an hour, 5 minutes after the hour, daily	month_current.gnu
5 * * * * Gnuplot /www/wind/month_voltage/month_voltage.gnu	Runs once an hour, 5 minutes after the hour, daily	month_voltage.gnu
5 * * * * Gnuplot /www/wind/month_power/month_power.gnu	Runs once an hour, 5 minutes after the hour, daily	month_power.gnu

5 0 * * * Gnuplot /www/wind/year_current/year_current.gnu	Runs daily, at 5 minutes after midnight	year_current.gnu
5 0 * * * Gnuplot /www/wind/year_voltage/year_voltage.gnu	Runs daily, at 5 minutes after midnight	year_voltage.gnu
5 0 * * * Gnuplot /www/wind/year_power/year_power.gnu	Runs daily, at 5 minutes after midnight	year_power.gnu
5 23 * * * /www/wind/month_current/monthCurrentScripts.sh	Runs daily, 5 minutes before midnight	monthCurrentScript.sh
5 23 * * * /www/wind/month_power/monthPowerScripts.sh	Runs daily, 5 minutes before midnight	monthPowerScript.sh
5 23 * * * /www/wind/month_voltage/monthVoltageScripts.sh	Runs daily, 5 minutes before midnight	monthVoltageScript.sh
5 23 * * * /www/wind/week_power/weekPowerScripts.sh	Runs daily, 5 minutes before midnight	weekPowerScript.sh
5 23 * * * /www/wind/week_voltage/weekVoltageScripts.sh	Runs daily, 5 minutes before midnight	weekVoltageScript.sh
5 23 * * * /www/wind/week_current/weekCurrentScripts.sh	Runs daily, 5 minutes before midnight	weekCurrentScript.sh
5 23 * * * /www/wind/year_power/yearPowerScripts.sh	Runs daily, 5 minutes before midnight	yearPowerScript.sh
5 23 * * * /www/wind/year_current/yearCurrentScripts.sh	Runs daily, 5 minutes before midnight	yearCurrentScript.sh
5 23 * * * /www/wind/year_voltage/yearVoltageScripts.sh	Runs daily, 5 minutes before midnight	yearVoltageScript.sh

Appendix B – List of All Files Included with the Report for the Online Portal

Public_Html

- day_current
  - day_current.gnu
- day_power
  - day_power.gnu
- day_voltage
  - day_voltage.gnu
- month_current
  - monthCurrentScript.sh
- month_power
  - monthPowerScript.sh
- month_voltage
  - monthVoltageScript.sh
- week_current
  - weekCurrentScript.sh
- week_power
  - weekPowerScript.sh
- week_voltage
  - weekvoltageScript.sh
- year_current
  - yearCurrentScript.sh
- year_power
  - yearPowerScript.sh
- year_voltage
  - yearPowerScript.sh
- about.html
- index.html
- month.html
- pic.html
- pic1.html
- tech.html
- turbine.html
- week.html
- year.html
- dayGnuplotScript.sh
- monthGnuplotScript.sh
- weekGnuplotScript.sh
- yearGnuplotScript.sh
- GnuplotDayCurrent.gnu
- GnuplotDayPower.gnu
- GnuplotDayVoltage.gnu
- GnuplotMonthCurrent.gnu
- GnuplotMonthVoltage.gnu
- GnuplotMonthPower.gnu
- GnuplotWeekVoltage.gnu
- GnuplotWeekCurrent.gnu
- GnuplotWeekPower.gnu
- GnuplotYearCurrent.gnu
- GnuplotYearVoltage.gnu
- GnuplotYearPower.gnu

Citation and Bibliography

Wind Industry Statistics. (2009). Retrieved from American Wind Energy Association: http://www.awea.org/faq/wwt_statistics.html

AWEA Small Wind Turbine Global Market Survey 2009. (n.d.). Retrieved from American Wind Energy Association: http://www.awea.org/smallwind/pdf/2010_AWEA_Small_Wind_Turbine_Global_Market_Study.pdf>.

EWEA. (n.d.). *Global Wind Energy Statistics Web.* Retrieved from European Wind Energy Association: http://www.ewea.org/fileadmin/ewea_documents/documents/statistics/gwec/GWEC_-_Table_and_Statistics_2009.pdf

Technologies, A. (2008, November 28). *HCNR2001.* Retrieved February 2, 2011, from AvagoTech: http://www.avagotech.com/pages/en/optocouplers_plastic/plastic_high_linearity_analog_optocoupler/hcnr201

World Wind Energy Report 2009. (n.d.). Retrieved from World Wind Energy Association: http://www.wwindea.org/home/index.php?option=com_content&task=view&id=266&Itemid=43

Wind Industry Statistics. (2009). Retrieved from American Wind Energy Association: http://www.awea.org/faq/wwt_statistics.html

AWEA Small Wind Turbine Global Market Survey 2009. (n.d.). Retrieved from American Wind Energy Association: http://www.awea.org/smallwind/pdf/2010_AWEA_Small_Wind_Turbine_Global_Market_Study.pdf>.

EWEA. (n.d.). *Global Wind Energy Statistics Web.* Retrieved from European Wind Energy Association: http://www.ewea.org/fileadmin/ewea_documents/documents/statistics/gwec/GWEC_-_Table_and_Statistics_2009.pdf

Technologies, A. (2008, November 28). *HCNR2001.* Retrieved February 2, 2011, from AvagoTech: http://www.avagotech.com/pages/en/optocouplers_plastic/plastic_high_linearity_analog_optocoupler/hcnr201

World Wind Energy Report 2009. (n.d.). Retrieved from World Wind Energy Association: http://www.wwindea.org/home/index.php?option=com_content&task=view&id=266&Itemid=43