ECE 445: Senior Design Spring 2017
Final Report

Wi-Fi Enabled Motorized Windows for Automatic Climate Control

Team 34
Trevor Bush (tsbush2)
Alex Casino (casino2)
Ryan Stark (rmstark3)

TA: Kexin Hui

Abstract

Windows are an extremely effective tool for indoor climate control, but they are not utilized regularly due to the uncertainty of rain, variation in temperature throughout the day, and inconvenience of opening and closing windows. The indoor climate can be regulated through air conditioning or heating systems; however, opening windows has many benefits including cheaper, energy efficient climate control that provides fresh air into a home.

We present a solution that allows a user to take advantage of the outdoor weather through automated window control. This system senses indoor and outdoor climate information and adjusts window position to achieve the desired indoor temperature. In this paper, we focus on the design and verification of this system. We will also consider the cost and safety of the product and discuss future work to be done on our design.

Table of Contents

1. Introduction

1.1 Motivation

Opening windows with weather dependent timing is an easy solution to allow natural ventilation to heat or cool a building. Opening windows reduces the total energy usage of a building by reducing the use of HVAC systems. Therefore, natural ventilation is of the utmost importance specifically in hot summers and more moderate seasons [1]. In addition, many people would prefer to use natural climate control rather than air conditioning and heating as it is more environmentally friendly and allows fresh air to flow throughout the building.

However, there are several potential issues and inconveniences in opening and closing windows for temperature control in a building. First, it is cumbersome to continuously open and close windows throughout the day to maintain a comfortable temperature inside buildings. Many individuals are not home during the day to tend to the windows and most are sleeping throughout the night. Second, if a window is accidentally left open during inclement weather, a home could experience water damage near the windows. Thus, most people use HVAC systems rather than windows to control the indoor climate which is why there is a use for an automatic window control system.

1.2 Solution

Our solution is an automatic climate control system that consists of a motorized window that autonomously opens and closes to efficiently regulate the indoor temperature. Each window will have four sensors: rain, temperature, ultrasonic (for position detection), and IR (for motion detection to prevent closing on objects). In addition, each window will have a microcontroller that processes the sensor data and a Wi-Fi module to send data to a hub. The hub is the central processing unit, similar to a thermostat. It contains a temperature sensor to measure the indoor temperature, a microcontroller, and Wi-Fi module. A mobile phone application will be used to enable the automatic climate control, set the desired temperature range, and manually open and close each window. Each window will receive 120V 60Hz AC power from the grid. A power management system will provide the necessary DC voltages for the microcontrollers, sensors, and motor. There are no commercially available products in the market that perform automatic climate control by utilizing Wi-Fi to control window position.

2. Design

Our project is made up of three modules: the window module, the hub module, and the smartphone module. The three separate modules are connected via Wi-Fi. The window

module is the main component in our project since it is what physically opens and closes the window. The hub module acts as a home thermostat, measuring indoor temperature, processing data, and communicating with both the window module and the smartphone. The smartphone is used to set user preferences. The overall block diagram is shown in Figure 1.

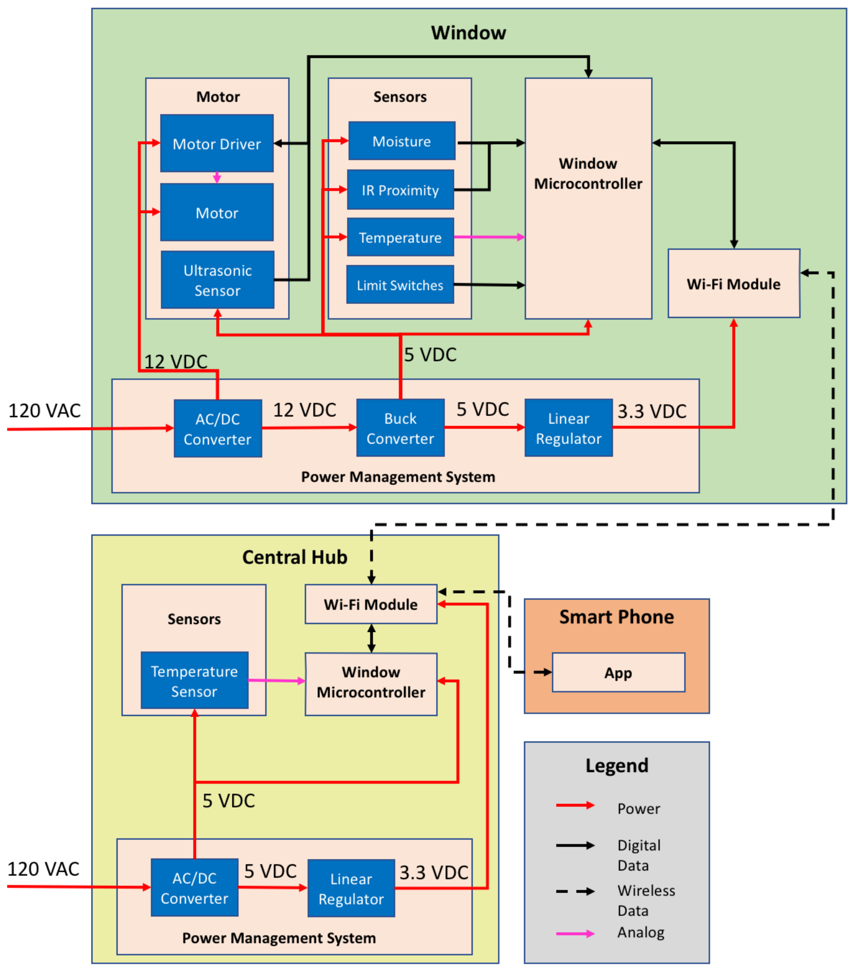

Figure 1: Block Diagram

Physical Design

The window is a single hung window. The inside view is shown in Figure 2 and the outside view is shown in Figure 3. There is a mounted sensor array on the exterior frame of the window that contains the moisture sensor and temperature sensor. These connect to a window control module that is mounted on the inner frame of the window.

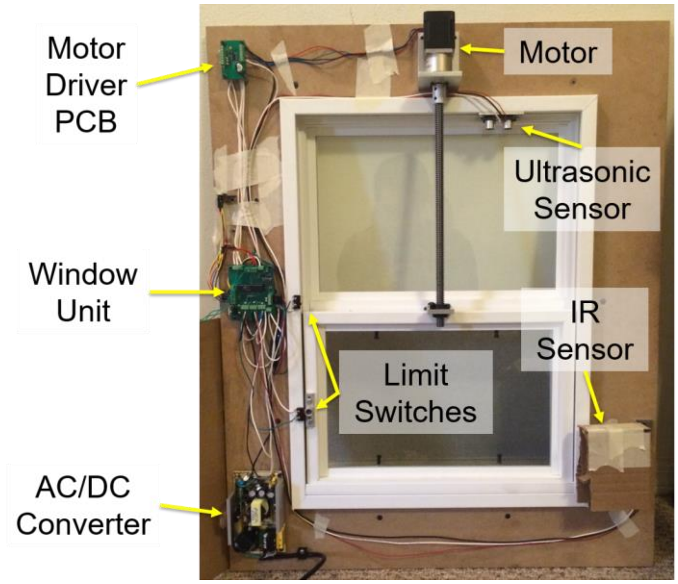

Figure 2: Window – Inside View

The two modules are positioned such that their interconnecting wires can be routed through a drilled hole in the frame. The window control module contains the microcontroller, Wi-Fi module, and power management system. The motor is attached to the top of the threaded rod mechanism, the motor driver is to the left of the motor, and the IR sensor is mounted on the lower right side of the window. The ultrasonic sensor is positioned at the top of the window facing downwards, so that it is pointed at the top of the moveable section of the window. There are limit switches along the left side of the window to sense if the window is fully open or closed. The central hub contains a microcontroller, Wi-Fi module, power management system, and temperature sensor.

2.1 Sensors

2.1.1 Temperature Sensor

The temperature sensor used for both indoor and outdoor temperature measurements outputs a voltage that is linearly correlated to the temperature in degrees Celsius. The voltage was read in by the ADC on the MCU. Once the voltage was obtained, we used simple arithmetic to convert the value to Fahrenheit as shown in Equation 1. A conversion factor of $\frac{5}{1023}$ was necessary because the temperature value obtained from the ADC is a 10-bit number representing an analog voltage between 0 and 5 V.

$$T(°F) = (V_{in} * 1000 * \frac{5}{1023}) \left(\frac{9}{5}\right) + 32 \tag{1}$$

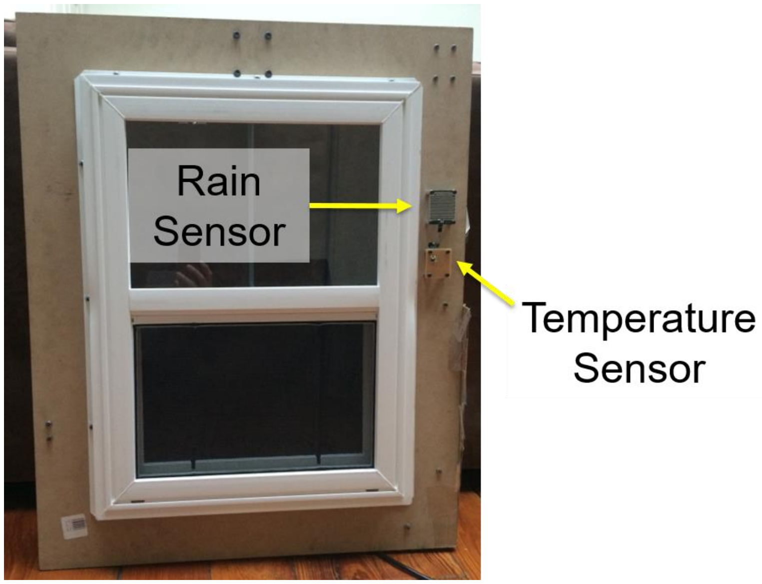

Figure 3: Window – Outside View

2.1.2 Rain Sensor

The Uxcell rain sensor is used to determine whether there is water present near the window by measuring the resistance across a set of thinly separated traces. The sensor has both an analog and digital output. In our project, we used only the digital output to determine whether there was rain present or not. The moisture sensitivity of the digital signal was set appropriately using a potentiometer that was part of the sensor.

2.1.3 IR Sensor

The IR sensor measures the level of IR radiation in its field of view. It calibrates upon startup to ambient IR levels, then it outputs a digital signal that is high if the level of detected IR radiation changes. Since the sensor operates by measuring IR emissions, it can detect the presence of a person or animal moving under the window without incorrectly being triggered based on the movement of inanimate objects. A small enclosure was constructed around the sensor to limit its field of view to only objects directly in front of the window.

2.1.4 Ultrasonic Sensor

The ultrasonic sensor is used to detect the distance that the window has moved. In our original design, the ultrasonic sensor was supposed to measure the absolute position of the window. After initial testing of the ultrasonic sensor, we determined that the ultrasonic sensor was not accurate enough to be used as the primary means for position control. Consequently, we elected to use limit switches to determine when the window is fully open or closed. The ultrasonic sensor was instead used to detect how far the window has moved in a specified amount of time in order to detect a stalled or jammed window. The ultrasonic sensor returns the round-trip travel time of a 40 kHz sound wave to the microcontroller [2]. Then, the microcontroller calculates the distance of the window based on Equations 2 and 3. In order to increase the accuracy of the ultrasonic distance measurements, each measurement is the average of five consecutive measurements. Measurements were taken every 13400 steps of the motor which corresponds to .25 inches of movement. If the delta between measurements was more than 0.1 inches from the expected value, the window is potentially stalled and the window unit is alerted. The details of how the ultrasonic sensor is integrated into the motor control system are contained in Section 2.5.3.

$$v = 331.5 + (.6T) \tag{2}$$

$$d = \frac{v\frac{t}{2}(100)}{(10^6)(2.54)} = \frac{vt}{50800} \tag{3}$$

2.1.5 Limit Switch

We used two limit switches to detect whether the window is fully open or fully closed. We were initially going to use the ultrasonic sensor to measure the absolute position of the window, but during preliminary testing, we realized that it was not feasible to accurately measure the absolute position of the window using the ultrasonic sensor. Instead, we used the ultrasonic sensor to measure the relative position and used two limit switches to detect if the window is fully open or closed. The limit switch is simply a single pole single throw switch. The schematic is shown in Figure 4. The output is high when the switch is open. The output is driven low when the switch is closed. The microcontroller monitors the status of both switches to determine whether the window is fully closed, partially open, or fully open.

2.2 Power Supply

There are four types of power converters that supply power to the system: two AC-DC converters and two DC-DC converters. The window requires 12 V for the motor and motor driver. It also requires 5 V for the sensors and microcontroller. Finally, it requires 3.3 V for the Wi-Fi module. Therefore, the window unit uses a 120 VAC-12 VDC AC-DC converter, a 12 V-5 V DC-DC converter, and a 5 V-3.3 V linear regulator.

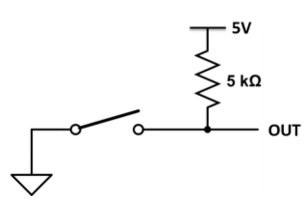

Figure 4: Limit Switch Schematic

The schematic for the window unit's power circuitry is shown in Appendix B Figure 13. The hub unit uses 5 V for the sensors and microcontroller and 3.3 V for the Wi-Fi module. The hub unit uses a 120 VAC-5 VDC AC-DC converter and a 5 V-3.3 V linear regulator, and the schematic for the hub unit's power circuity is shown in Appendix B Figure 14.

2.2.1 Window AC-DC Converter

The primary power converter for the window is a 120 VAC to 12 VDC power converter [3]. The motor and motor driver are the primary loads from the 12 V bus. The 12 V bus also supplies the buck converter on the window unit which provides 5 V to the microcontroller, sensors, and a low dropout regulator which provides power to the Wi-Fi module (see sections 2.2.3 and 2.2.4). The AC-DC converter is a Triad Magnetics ALS75-12 which is a 6 A industrial power supply. The AC input is provided by a 3-prong 6 foot power cable to an outlet. The converter is mounted on the lower left corner of the window.

2.2.2 Hub AC-DC Converter

The primary power converter for the hub is a 120 VAC to 5 VDC power converter. The 5 V bus powers the microcontroller, sensors, and a low dropout regulator that provides power to the Wi-Fi module. The converter is a CUI Inc. SWI10-N which is a simple, inexpensive converter that plugs directly into the wall for its input and has a barrel plug connector for its output.

2.2.3 DC-DC Converter

The DC-DC converter is a TI LM22676 buck regulator [4]. The buck regulator requires an inductor with a value that depends on the desired input voltage, output voltage, and minimum current ripple. For our circuit, the input voltage to the buck converter is 12 V, and the output voltage is 5 V. The recommended maximum current ripple should be less than a third of the minimum load current. We set the value of the minimum load current to be 100 mA based on the expected current draw of the sensors, Wi-Fi module, linear regulator, and microcontroller.

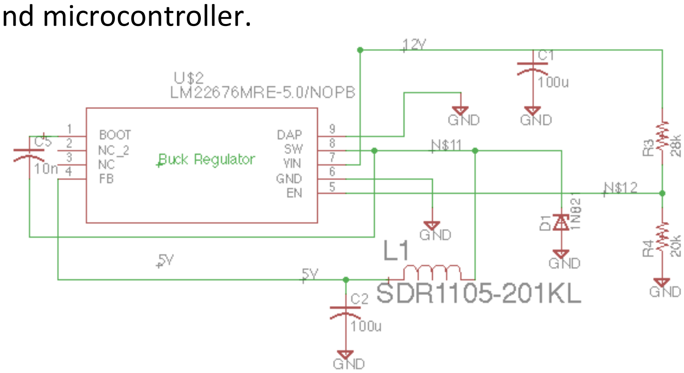

Figure 5: Buck Converter Circuit Schematic

This calculation, shown in Equation 4, led us to use a 200 mH inductor, which ended up being sufficient as shown in Section 3.2. The circuit schematic for the buck regulator is shown in Figure 5.

$$L_1 = \frac{(V_{in} - V_{out})V_{out}}{.3V_{in}FI_{min}} \tag{4}$$

2.2.4 Linear Regulator

The Wi-Fi module requires 3.3 V, so two linear regulators are used to step down 5 V to 3.3 V to power both Wi-Fi modules on the window unit and hub unit. The linear regulator is an LM2937ET which is a simple three pin device: input, output, and ground [5].

2.3 Controller Unit

The controller units take in data from their sensors, process the data, and communicate to each other and to a smartphone through their attached Wi-Fi modules.
Both microcontrollers are ATMega328s that run on their internal 8 MHz clock. The ATMega328 was chosen for its low cost and large number of I/O pins. The internal clock was chosen for its low power consumption.

2.3.1 Microcontroller on the Window

The microcontroller on the window reads in values from its sensors and processes them per our control flow described in detail in Section 2.7. The sensors attached to this microcontroller include one of the temperature sensors, the rain sensor, the IR sensor, the ultrasonic sensor, and both limit switches. This microcontroller receives commands from the hub as to whether or not to open or close. When it receives a command to open or close the window, the microcontroller sends the required signals to the motor driver in order to complete the operation. Additionally, the microcontroller receives communicates with the hub through an attached Wi-Fi module. A diagram showing the communication scheme is shown in Figure 6 and the schematic for the window unit is shown in Appendix B Figure 15 and the PCB layout is shown in Figure 16.

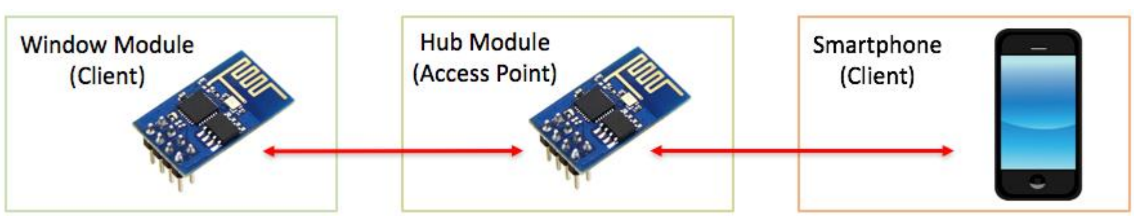

Figure 6: Wi-Fi Communication Diagram

2.3.2 Microcontroller on the Hub

The microcontroller for the hub module receives data from an on-board temperature sensor and from the hub Wi-Fi module. Based on the received data and our control flow it sends a command, along with information on window position and local temperatures, back to the Wi-Fi module. The Wi-Fi module then sends this data to the smartphone or window module as desired. The schematic for the hub unit is shown in Appendix B Figure 17 and the PCB layout is shown in Figure 18.

2.4 Wi-Fi Modules

To connect the three separate modules of our project we used two ESP8266 integrated Wi-Fi modules on ESP-01 boards. We chose the ESP-01 for its size, cost, and integrated antenna. We set up one module as an access point and one as a client using the Arduino IDE. They then communicated with each other using http requests. The access point also sent and received data with the smartphone, again using http requests. WPA security was implemented when setting up the access point as a safety precaution. The data that was sent and received by the Wi-Fi modules was sent to and from the microcontrollers using UART.

2.5 Motor

2.5.1 Stepper Motor

The window system used a Nema 17 stepper motor with an attached planetary gearbox [6]. We chose a gearbox ratio of 27/1. This means that the torque is stepped up by a factor of 27 from the default Nema 17 model at the expense of the speed being reduced by a factor of 27. The torque was 3 Nm and the speed was 7.2 rpm. The highest priority was to have a motor that had enough torque to open and close the window. We desired that the window open and close in less than ten-minutes, therefore, we chose the largest gearbox ratio that would still be able to open the window in this time.

2.5.2 Motor Driver

The motor driver PCB has inputs and outputs to the window unit microcontroller and is responsible for controlling the motor. The PCB is constructed around a Texas Instruments DRV8886 stepper driver [7]. It uses an indexer and is controlled by the step/direction method. In other words, one binary input controls the direction of the motor rotation, and one input is a pulse train where the motor moves one step per pulse. The motor driver has nine inputs: DECAY, TRQ, M1, M0, DIR, STEP, ENABLE, NSLEEP, and RREF. Five inputs were hard-wired to the desired settings. We did not require microstepping since the gearbox already reduced the angle per step, so the microstepping mode was set to ‘OFF’ by connecting inputs M0 and M1 directly to ground. The DECAY pin was connected to ground via a 45 kΩ resistor which set the decay mode to 'Mixed decay: 60% fast'. This mode was chosen because it is possible that there would be large off-time periods where slow decay might not properly regulate the holding torque current. Also, mixed decay gives a good balance between current ripple and current limiting accuracy. The ENABLE pin was set HIGH so that the device is always on. We decided upon 1.5 A for the full scale current so that the motor can draw sufficient current for max torque while remaining under the 1.68 A limit of the motor. Therefore, the value of RREF was chosen by Equation 5.

$$RREF[k\Omega] = \frac{30[kA\Omega] \times TRQ}{I_{FS}[A]} = \frac{30\times1}{1.5} = 20\ [k\Omega] \tag{5}$$

Four inputs were dynamically configured via the microcontroller: TRQ, DIR, STEP, and nSLEEP. TRQ can be used to toggle the full scale current to 75% or 50% of its normal value. We ultimately set TRQ to be 100% of the full-scale current, but this feature could be used to save power after the motor overcomes the initial inertia of the window. STEP and DIR are used to move the motor. The input nSLEEP is used to put the motor driver to sleep and thus saves power when the window is not actively opening or closing. There are five outputs: four motor outputs and nFAULT. nFAULT is sent back to the microcontroller to monitor the status of the motor driver. The motor outputs are sent to the motor, two per phase. The schematic is shown in Appendix B Figure 19 and the PCB layout is shown in Figure 20.

2.5.3 Control

The motor control scheme is shown in Figure 7. The basic idea is that the motor speed is initially ramped up via an acceleration profile. Then, the motor moves in blocks of 50 steps. After 50 steps, the microcontroller checks the outputs of the IR sensor and limit switches. If the IR sensor is tripped, the motor halts until the object moves out of the way. If the limit switches are tripped, the motor halts because the window is fully open. The motor continues this cycle until the motor has been stepped 13400 times which corresponds to a vertical displacement of .25 inches. The ultrasonic sensor verifies that the window has actually moved .25 inches by taking a distance measurement before and after. If the measurements don't match, the motor may have stalled, so the system waits 5 seconds and attempts to restart the motor. If the measurements don't match for the second time, then the motor is assumed to be stalled, so the system halts completely and alerts the user.

2.6 Smartphone Application

The smartphone is the connection between the user and system, so it was designed to have a simple user friendly layout.

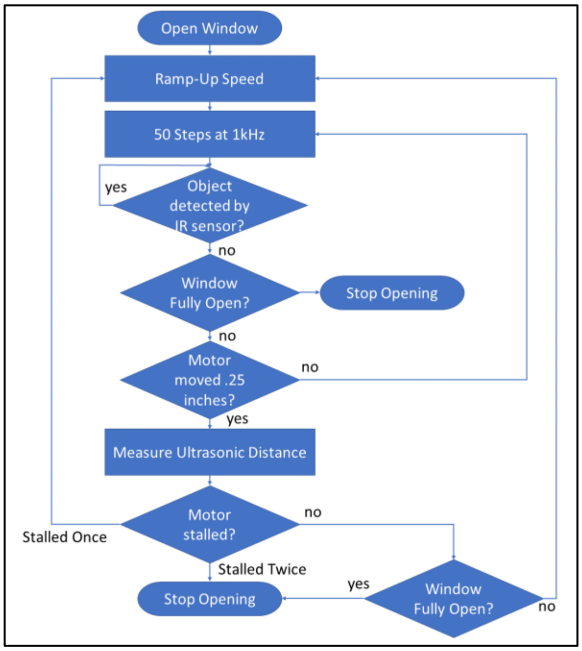

Figure 7: Motor Control Flow

This layout allows the user to specify the temperature range they desire indoors and switch between automatic and manual control of the window. This information is sent to the hub Wi-Fi module, which then sends back updated information on the window position and the presence of rain. This information is then displayed clearly in the interface as shown in Figure 8.

2.7 Control Description and Flowcharts

The overall goal of our control algorithm was to keep the indoor temperature within a range set by the user. Additionally, we do not want the window to oscillate (continuously open and close). A high-level control flow is show in Figure 9.

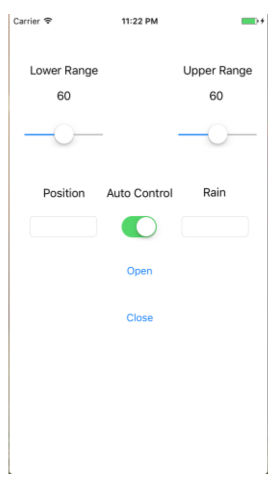

Figure 8: Application Interface

The system initially checks for a manual user control request and for rain presence before attempting to regulate the indoor temperature. Once the system has verified that automatic control is on and rain has not been detected, then the indoor and outdoor temperature are used to determine the desired window position. In situations where the indoor temperature is within the user preferred range, no action is taken. If instead the indoor temperature is outside of that range, then analysis is done to determine if opening or closing the window will improve the indoor temperature.

The window position is determined by first assessing whether the current temperature is above or below the desired temperature. Then the outdoor temperature is compared to the indoor temperature. If the outdoor temperature is in the same direction from the indoor temperature as the desired range then it is concluded that the outside climate is an improvement over the indoor temperature. In this case the window is opened (unless it is already open, in which case it will remain open). Otherwise it is concluded that the outside temperature will worsen the indoor temperature and the window is closed (unless it is already closed, in which case it will remain closed). To prevent oscillation in the event that the indoor and outdoor temperature are similar to each other, another condition was added so that the window position is only changed if the two temperatures are at least 2 degrees apart.

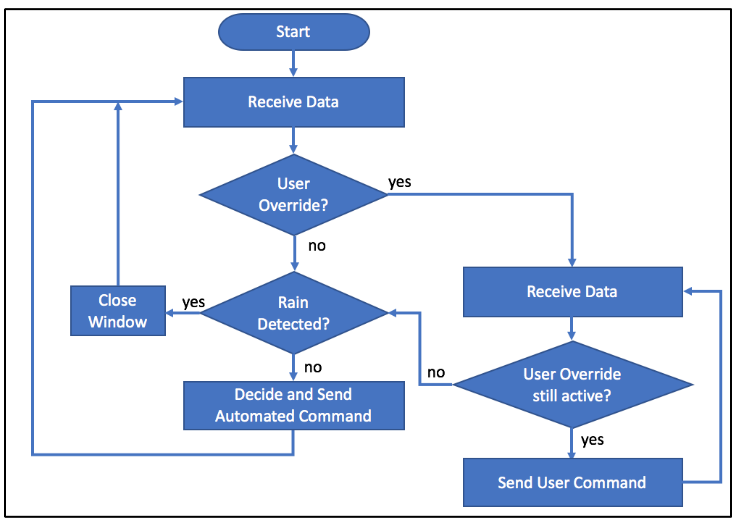

Figure 9: Top Level Flow Chart

3. Design Verification

Initially, we verified each block using a modular approach to ensure that the requirements of each component were met separately. Then, our main verification was performed through full system verification. Essentially, if the entire system functions properly in all cases, then it implies that all individual blocks are functioning as well. In the following sections, we will describe the verification of each module.

3.1 Sensors

3.1.1 Rain Sensor

We verified the operation of the rain sensor by applying drops of water to the board and monitoring the digital output voltage of the sensor with a multimeter. The potentiometer that came with the unit was set to the desired sensitivity during these tests. A set number of 3 droplets were applied to test the threshold of the digital output and the output voltage was verified to be high throughout each trial.

3.1.2 IR Sensor

The IR sensor was allowed to calibrate with nothing in its field of view, then a hand was moved in front of the sensor at set distances and a change in output voltage was verified using an oscilloscope.

3.1.3 Ultrasonic Sensor

First, in order to verify our requirements of the ultrasonic sensor, we performed a simple test. We set the ultrasonic sensor on a table pointed towards a textbook with a ruler to determine the actual distance. The test setup is shown in Figure 10. We moved the textbook in .5 inch increments along a range of 2 inches to 10.5 inches, which is slightly larger than the actual distance range of the window. At each 0.5 inch increment, we took 100 distance measurements and averaged them. The measured distance vs. real distance is shown in Figure 11. The measured distance was so close to the real distance that the mean error vs. real distance, shown in Figure 12, is a more informative plot. The worst-case error was 7.5% which meets our spec of 12%, and the error across the majority of the range is less than 3%.

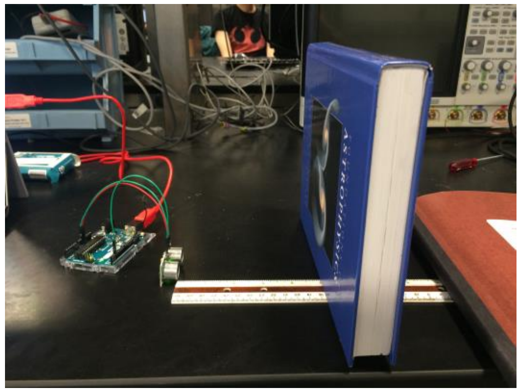

Figure 10: Initial Ultrasonic Test Setup

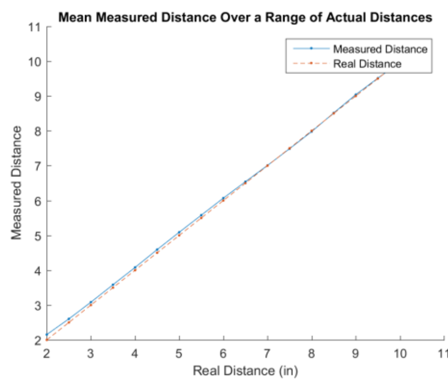

Figure 11: Measured Distance (inches) vs. Real Distance (inches)

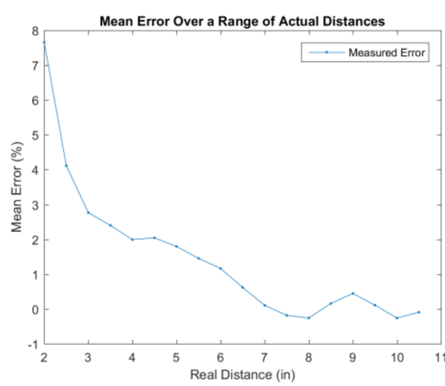

Figure 12: Mean Error (%) vs. Real Distance (inches)

After the machine shop constructed the window, we tested the ultrasonic sensor on the window. We printed off the ultrasonic sensor's data on a serial monitor while the window was moving and compared it to the actual distance using a ruler. A sample is shown in Table 1 as the window was closing. The ultrasonic sensor should have read .25 inches, yet we see a delta measurement as low as .18 inches and as high as .30 inches. Thus, we observe that the error increased compared to the initial tests. After several experiments, we determined that a tolerance range of .15 to .35 inches was necessary due to the inaccuracy of the ultrasonic sensor.

Table 1: Ultrasonic Testing on Window During Closing

Distance (inches)	Delta (inches)
4.89	
5.16	.27
5.34	.18
5.64	.30
5.88	.24

3.1.4 Temperature Sensor

We measured and compared the output of the temperature sensor with the actual temperature reported by a local thermostat. In addition, we heated up the temperature sensor by placing a finger on it and verified that the reported temperature was raised several degrees.

3.1.5 Limit Switches

The limit switches utilize a very simple circuit. The output is HIGH when the switch is open. The output is LOW when the switch is closed. This was initially verified by pressing the switch while monitoring the output on an oscilloscope. Finally, the operation of the limit switches was verified when the motor stopped opening or closing the window when the respective limit switch was tripped.

3.2 Power Converters

The primary requirement for the power converters was that the output voltage was within ±2% of the desired voltage at the expected load current of our system. In order to verify this requirement, we performed load testing using a DC Electronic Load. The results are shown in Table 2 and the output waveforms are shown in Appendix C. Overall, all four power converters passed the load testing. The 12-5 V DC-DC converter was the closest to failing the spec. The output voltage was 4.91 V and the spec required a voltage greater than 4.90 V.

Table 2: Power Supply Load Testing

Converter	I_L (A)	Min Allowed V_out (V)	Max Allowed V_out (V)	Measured V_out (V)
AC-DC 120-5V	1	4.90	5.10	4.96
AC-DC 120-12V	2	11.76	12.24	11.91
DC-DC 12-5V	1	4.90	5.10	4.91
DC-DC 5-3.3V	0.3	3.23	3.37	3.30

3.3 Microcontrollers

3.3.1 Window Microcontroller

Correct operation of the microcontroller was verified by printing status updates serially over the RX and TX pins and the Arduino IDE's serial monitor. We monitored the temperature and rain sensors to ensure they worked correctly by printing off their values as we applied rain to the rain sensor, and heat to the temperature sensor. Additionally, the microcontrollers ability to control the motor was verified as the window was able to successfully open and close repeatedly as desired. The IR sensor was interpreted correctly as the window would stop moving whenever a hand was placed in under the window. Lastly the communication with the Wi-Fi module was verified as described in Section 3.4.

3.3.2 Hub Microcontroller

We verified functionality of the hub microcontroller by adjusting the value of control parameters such as desired temperature, indoor temperature, outdoor temperature, and rain status. These parameters are typically received from the window unit or smartphone. In testing, we sent them to the MCU serially from a computer. The updated values on the hub MCU and the response action were then communicated back to the computer where they were compared with the expected response. In addition, the value read by the MCU from the temperature sensor was compared with the voltage measured on the analog pin from the temperature sensor.

3.4 Wi-Fi Module

To ensure the Wi-Fi modules were working correctly we performed two tests. In the first test, we sent a message from an Arduino to the Wi-Fi module using UART and then sent the message back to the Arduino. We performed this test 100 times and were able to read back the message successfully 100 times. The next test involved sending 100 messages from one Wi-Fi module to the other. To do this, we sent a message from one module to the other. Then, to confirm it was received, the second module would send a message back. This ensured that both modules worked as receivers and transmitters. We also took note of the speed of communication during these tests. Each message took about 0.2 seconds to go from one module to the other and back, being printed to the screen every time. This was well within our goal of 2 seconds per message.

3.5 Smartphone Application

Wireless connectivity between the smartphone and hub Wi-Fi module was done initially by sending a string repeatedly from the phone to the hub. This character was then sent through serial communication from the hub microcontroller to a laptop where it could be verified. Similarly, a string was then sent from the hub Wi-Fi module to the phone and verified from the smartphone. The time from sending a string on a phone to read it on a laptop was under a second, which is within our specified length of time for wireless communication.

We were able to get both the smartphone application, and smartphone communication working; however, we were not able to integrate them. Through full system verification we were able to open and close the window, set the comfortable range, and read status updates from the hub on the smartphone using http requests in a web browser.

3.6 Motor

The first two requirements for the motor were that the motor moves one step when given a step input and that the motor can move in both directions. In order to verify these requirements, we connected the motor inputs to an Arduino and monitored the step input, the fault status output, and the current through one phase of the motor. By stepping the STEP input, we verified that the motor moved one step per input pulse. By changing the DIR input from 0 to 1, we verified that the motor moved in both directions. One such test, where the STEP input is a 1 kHz signal and the DIR input is set to clockwise, is shown in Appendix B Figure 25. The final requirement was that the motor must be able to move the fixture up to the fully-open position. In order to verify the final requirement, we attached the motor to the window and let it run until the limit switch was tripped. We initially encountered an issue where the motor would sometime stall and vibrate in place. We figured out that this was due to starting the motor too abruptly. The stepper motor required an acceleration profile, so rather than immediately applying a 1 kHz signal to the STEP pin upon startup, we first applied pulses at 100 Hz, 200 Hz, 300 Hz, etc. up until 1 kHz. This solved the issue of stalling and we met the requirement of the motor being able to move the fixture up to the fully-open position.

4. Cost

4.1 Labor

Table 3: Labor Cost

Name	Hourly Rate	Total Hours	Total Cost
Trevor Bush	$30	200	$15,000
Alex Casino	$30	200	$15,000
Ryan Stark	$30	200	$15,000
Total		600	$45,000

4.2 Parts

Table 4: Parts Cost

Part	Part Number	Vendor	Unit Cost	Quantity	Total Cost
Temperature Sensor	LM35DMX/NOPB	Digi-Key	$1.97	2	$3.94
Rain Sensor	a13090400ux0745	Amazon	$6.57	1	$6.57
IR Sensor	555-28027-ND	Digi-Key	$12.99	1	$12.99
Wi-Fi Module	WRL-13678	Sparkfun	$6.95	2	$13.90
Microcontroller	ATMEGA328-PU	Microchip Direct	$1.90	2	$3.80
Buck Converter	LM22676MRE-5.0/NOPBCT-ND	Digi-Key	$5.09	1	$5.09
Linear Regulator	LM2937ET-3.3/NOPB	Digi-Key	$1.53	2	$3.06
120VAC-12VDC Converter	ALS75-12	Digi-Key	$24.28	1	$24.28
Power Cord to Cable	Q941-ND	Digi-Key	$4.44	1	$4.44
Fuse	507-1995-1-ND	Digi-Key	$0.35	2	$0.70
120VAC-5VDC Converter	SWI10-5-N-P6-ND	Digi-Key	$6.38	1	$6.38
Power Jack	EJ508A-ND	Digi-Key	$0.83	1	$0.83
Stepper Motor Driver	DRV8886	TI	$5.03	1	$5.03
Stepper Motor	17HS19-1684S-PG27	Stepper Online	$29.65	1	$29.65
Ultrasonic Sensor	28015-ND	Digi-Key	$29.99	1	$29.99
Level Converter	BOB-12009	Sparkfun	$2.95	2	$5.90
Miscellaneous electronic components (R,L,C,etc.)					$20.00
Window	VSH1824	Menards	$39.99	1	$39.99
Total					$216.54

4.3 Total Cost

Table 5: Total Cost

Labor	$45,000
Parts	$216.54
Grand Total	$45,216.54

5. Conclusion

5.1 Accomplishments

By the end of the project, we had a complete system that could open and close a window automatically and determine the optimal window position based on indoor and outdoor temperature, rain measurement and user set preferences. Our project contained functioning safety features and position feedback to stop window movement in cases where the window is obstructed or jammed. There was serial communication between each Wi-Fi module and microcontroller and wireless communication between the Wi-Fi modules. While the smartphone user interface was not integrated with the overall system, all desired data was successfully communicated between the smartphone and Wi-Fi modules.

5.2 Ethics and Safety

We followed the IEEE code of ethics throughout our project. To ensure that we made "decisions consistent with the safety, health, and welfare of the public," we made safety a priority in our project [8]. There were two main safety concerns in our project: physical safety and burglary.

To ensure physical safety, we included two different sensors to prevent the window from closing on a hand, tail of a pet, or some other object that was in the way. An IR sensor was used to detect any motion and immediately stop the window should something move into its path. An ultrasonic sensor was used to measure the window's movement and, should the window stop incorrectly due to a jam, it would stop the motor's movement.

To prevent burglary, we decided to implement our communication with WPA security. This way, only users with the password would be able to alter the window control system preventing potential burglars from causing undesired window movement. Additionally, we allowed the user to turn off the automatic climate control and keep all windows closed.

5.3 Future Work

One of the biggest problems preventing someone from using our window was the speed and operating frequency of the motor. The window takes about seven minutes to open or close and during this time the motor outputs an audible 1 kHz tone. Specifically, we would consider using a brushless DC motor. Additionally, the design of our window is not visually appealing. A more attractive design would be to use a casement window which would eliminate the threaded rod going down the middle of the window. We did not use a casement window this semester due to their high cost.

In the future, we would recommend this product be integrated into an existing Wi-Fi enabled home thermostat. For example, Honeywell has a Wi-Fi thermostat and our solution would be a useful addition to the system, allowing for a more environmentally friendly way to regulate indoor temperature, and providing another degree of freedom for the user [9].

In this scenario, two improvements would be easily implemented. First, a nicer user interface that could present more advanced user options would be available. Second, the system would work off the internet, rather than a local network, allowing the user to set preferences from anywhere in the world.

5.4 Acknowledgements

We would like to specifically thank Glen Hedin and the machine shop for all the work put into our window, as well as our TA Kexin Hui.

References

[1] Hom B. Rijal and Miho Honjo and Ryota Kobayashi and Takashi Nakaya, “Investigation of comfort temperature, adaptive model and the window-opening behaviour in Japanese houses,” Architectural Science Review, Vol. 56, No. 1, pp. 54–69, 2013.

[2] Parallax, “PING))) Ultrasonic Distance Sensor (#28015)” PING datasheet, Revised Feb. 2013.

[3] Triad Magnetics, “ALS75-12” ALS75-12 datasheet, Dec. 2013.

[4] Semtech, “TS30011/12/13 High Efficiency 1A/2A/3A Current-Mode Synchronous Buck DC/DC Converter, 1MHz,” TS30013 datasheet, Revised Aug. 2016.

[5] Texas Instruments, “LM2937 2.5-V and 3.3-V 400-mA and 500-mA Voltage Regulators” LM2937 datasheet, Dec. 2013.

[6] Nema, “Nema 17 Planetary Geared Stepper Motor”, Nema 17 datasheet.

[7] Texas Instruments, “DRV8886 2-A Stepper Motor Driver With Integrated Current Sense”, DRV8886 Datasheet, Jan. 2017.

[8] Ieee.org, "IEEE IEEE Code of Ethics," 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 16-Feb-2017].

[9] "Lyric™ Wi-Fi T5 Thermostat (RCHT8610WF) Honeywell". Yourhome.honeywell.com. N.p., 2017. Web. 4 May 2017.

Appendix ARequirement and Verification Table

Table 6: Requirements and Verification Table

Module	Requirements	Verification	Status (Y or N)
Rain Sensor	1. Correct output 90% of the time when no water is present on the sensor pad. 2. Correct output 90% of the time in very light rain.	1. Read output when sensor pad is dry and ensure correct output. Both initially, and as verifications 2 and 3 are carried out. Record functionality during tests and determine accuracy. 2. a) Add drops of water to pad using an eye dropper. 2. b) Read output voltage using a multimeter. 2. c) Wipe pad clear of water until output indicates no water. 2. d) Repeat 10 times to determine accuracy.	Y
IR Sensor	1. Proper object detection when an object isn't in the way. Accuracy within one foot. 2. Proper object detection when an object is in the way or the window closing. Accurate 95% of the time and not dependent on distance of object to IR sensor. 3. The IR sensor gives its output quick enough.	1. a) Read output voltage of IR sensor using a multimeter with IR sensor attached to window and no objects within one foot of the window. Verify correct output. 2. a) Place hand 4 inches from the IR sensor, directly under where the window would be closed. Verify correct output voltage. 2. b) Place hand 10 inches from the IR sensor, directly under where the window would be closed. Verify correct output voltage. 3. Measure rise time of IR sensor voltage on oscilloscope.	Y
AC-DC Converter (12V)	1. Output voltage is 12V +- 2% 2. Maintain output voltage of 12V +- 2% for 2 amp load.	1. Verify voltage at output pin is in allowed range with Mulitmeter. 2. a) Apply 6 ohm resistive load for 120V AC input. 2. b) Verify voltage at output pin is in allowed range with Multimeter.	Y
AC-DC Converter (5V)	1. Output voltage is 5V +- 2% 2. Maintain output voltage of 5V +- 2% for 1 amp load.	1. Verify voltage at output pin is in allowed range with Mulitmeter. 2. a) Apply 5 ohm resistive load for 120V AC input. 2. b) Verify voltage at output pin is in allowed range with Multimeter.	Y
DC-DC Converter (12-5V Buck)	1. Output voltage is 5V +- 2% 2. Maintain output voltage of 5V +- 2% for 1 amp load.	1. Verify voltage at output pin is in allowed range with Mulitmeter. 2. a) Apply 5 ohm resistive load for 12V DC input 2. b) Verify voltage at output pin is in allowed range with Multimeter.	Y

DC-DC Converter (5-3.3V Linear Regulator)	1. Output voltage is 3.3V +- 2% 2. Maintain output voltage of 3.3V +- 2% for 300 milli-amp load.	1. Verify voltage at output pin is in allowed range with Mulitmeter. 2. a) Apply 5 ohm resistive load for 5V input. 2. b) Verify voltage at output pin is in allowed range with Multimeter.	Y
Window Microcontroller	1. Read correct value from rain sensor. 2. Read correct value from temperature sensor to within 5mV. 3. Reliably communicate with Wi-Fi Module (both receiving and transmitting). 4. Send appropriate control signal to motor. 5. Microcontroller able to step through no rain control system within 1 second.	1. a) Print value read from rain sensor pin for both wet and dry case. 1. b) Measure voltage at rain sensor pin with Multimeter. 1. c) Verify that microcontroller and Multimeter readings match. 2. a) Print value read from temperature sensor pin. 2. b) Measure voltage at temperature sensor pin with Multimeter. 2. c) Verify that microcontroller and Multimeter readings match. 3. a) Send 100 packets to Wi-Fi module. 3. b) Read back and verify 99 packets were received correctly. 4. a) Set signal HIGH and verify pin voltage with a Multimeter. 4. b) Set signal LOW and verify pin voltage with a Multimeter. 5. a) Set input signals (temperature, hub data) manually in the code to ensure only testing microcontroller speed. 5. b) Run control algorithm and time it using timing features of the microcontroller.	Y
Hub Microcontroller	1. Accurately read in signals from temperature sensor. 2. Reliably communicate with Wi-Fi Module (both receiving and transmitting). 3. Compute desired actions based on input signals.	1. Print value read by Microcontroller and compare to voltage measured at input pin. 2. Send and receive 100 packets and ensure 99 were transmit correctly each way. 3. Print computed values for temperature and compare to printed values read from sensor.	Y
Wi-Fi Module	1. Microcontroller can communicate with the Wi-Fi module.	1. a) Send data serially from Arduino to Wi-Fi module and have the data sent back to the Arduino, verify 99% or better accuracy.	Y

	2. Wi-Fi module can send and receive data packets with 99% accuracy (PER of .01). 3. Wi-Fi module receives data within 2 seconds of transmission.	2. a) Send dummy packets from one Wi-Fi to the other, verify that received messages match sent messages. 2. b) Determine correct number of packets received by recording all packets and comparing to what was sent. 2. c) Switch which module is sending and which is receiving. 2. d) Determine accuracy of each module. 3. Send dummy packets from one Wi-Fi to the other, measure time using stopwatch and verify over 20 trials that the desired speed is achieved.	
Smartphone Application	1. Smartphone application has functioning buttons and sliders. 2. User input into the phone is successfully obtained. 3. Data obtained from phone is successfully sent to microcontroller.	1. a) Button for controlling if system is on can slide to left and right, 'On' and 'Off' become bold as they are selected. 1. b) Comfortable range sliders allows user to set lower and upper bounds on their preferred temperature range. 1. c) User can change switches on windows from 'Closed' to 'Open' and proper bolding of window status is done. 1. d) Password pop-up opens when user attempts to change a setting, and closes when password is entered. 2. In app building software, return values of the various settings to the command line to verify that they match. 3. a) Set comfortable range on app and turn on automatic climate control. 3. b) Read out the received comfortable range and system status from the microcontroller. 3. c) Turn off automatic control. 3. d) Change position of window and verify that window changes position properly.	N Y Y
Motor	1. Motor moves one step when given a step input 2. Motor can move in both directions 3. Motor can move fixture up to fully-open position indicated by the upper limit switch on the window frame	1. a) Pulse step pin with 5373 steps (.067° per step) 1. b) Ensure that motor has turned 360°±2° 2. a) Apply logic high to direction input. 2. b) Pulse step pin with a 1kHz PWM signal. The motor should move one direction (up/down). 2. c) Apply logic low to direction input. 2. d) Pulse step pin with PWM signal. The motor should move the opposition direction(down/up) 3. a) Connect stepper motor to threaded shaft with fixture. Do not attach fixture to window.	Y

		3. b) Use microcontroller, motor driver, and stepper motor to move fixture to fully-open position	
Ultrasonic Sensor (Motor Position Sensor)	1. Sensor must provide distance measurements with less than 12% error (worst case spec) between 2 inches and 10.5 inches 2. Sensor must be able to make 20 measurements per second while window is in a position between fully-open and full-closed	1. a) Place sensor facing a side on table in room at room temperature (68-72°F). 1. b) Place textbook (to replicate window pane) centered directly in front of sensor. 1. c) Send chirp and measure distance 100 times at .5 inch increments in 2in-10.5in range. 1. d) Ensure less than 12% average error. 2. a) Place sensor facing a side on table in room at room temperature (68-72°F). 2. b) Place textbook (to replicate window pane) centered directly in front of sensor. 2. c) Send pulses every 50ms to SIG pin and measure echo time on SIG pin. 2. d) Sensor must measure distance to within 12% error on at least 18/20 pulses.	Y

Appendix BCircuit Schematics and Layouts

B.1 Power Systems

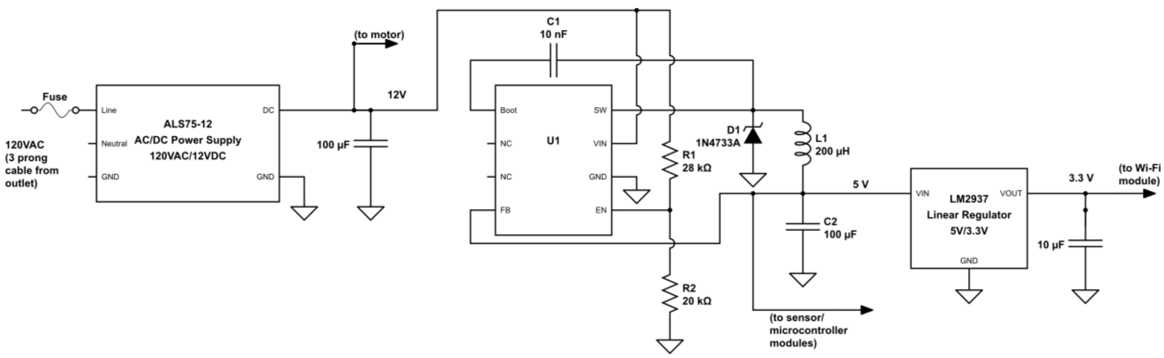

Figure 13: Window Unit Power System

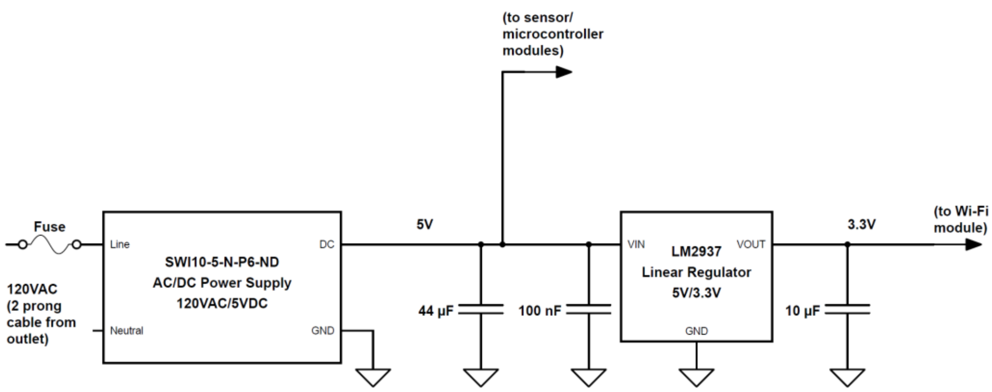

Figure 14: Hub Unit Power System

B.2 Window Unit

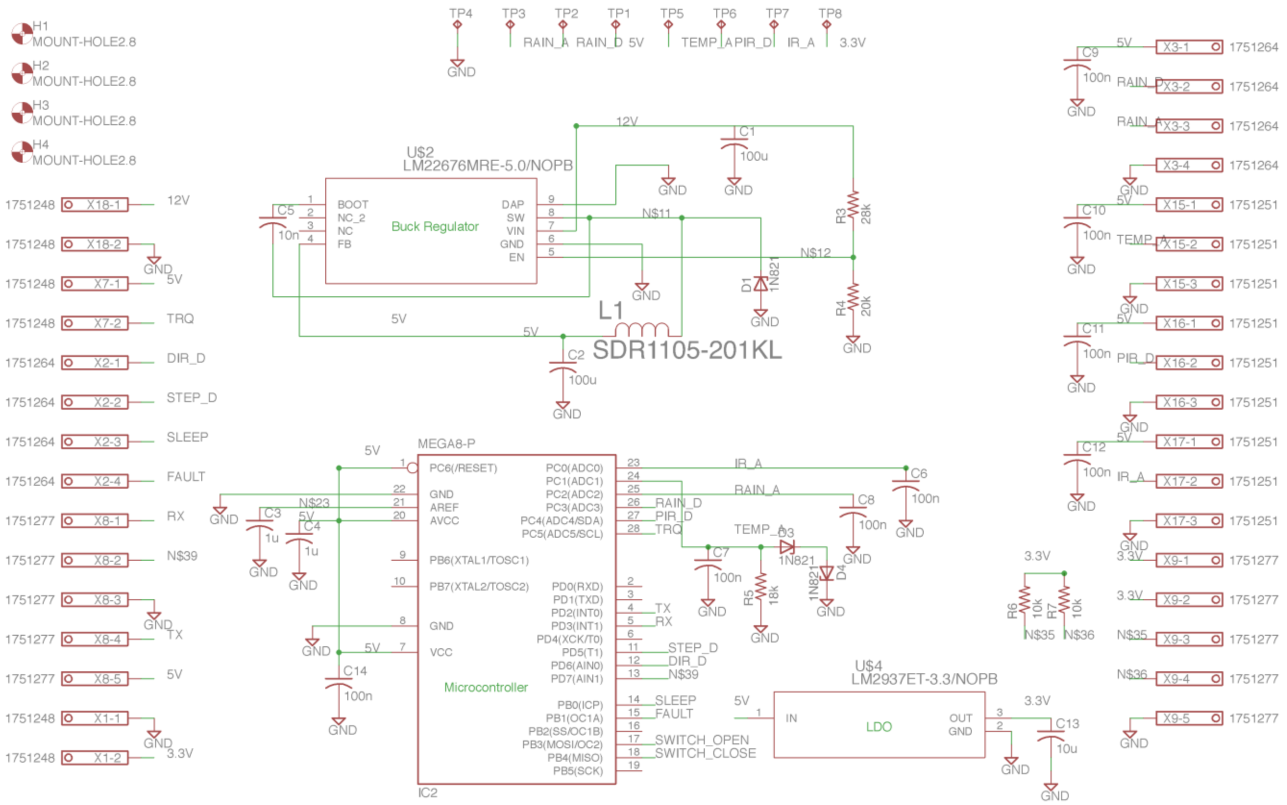

Figure 15: Window Circuit Schematic

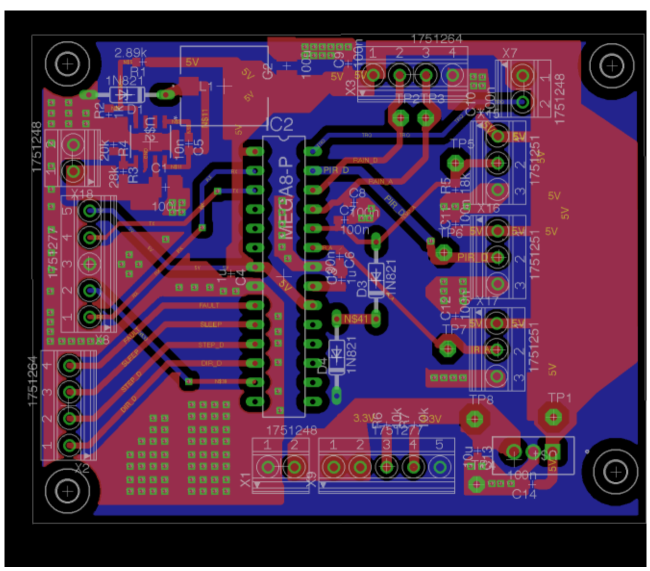

Figure 16: Window Unit PCB Layout

B.3 Hub Unit

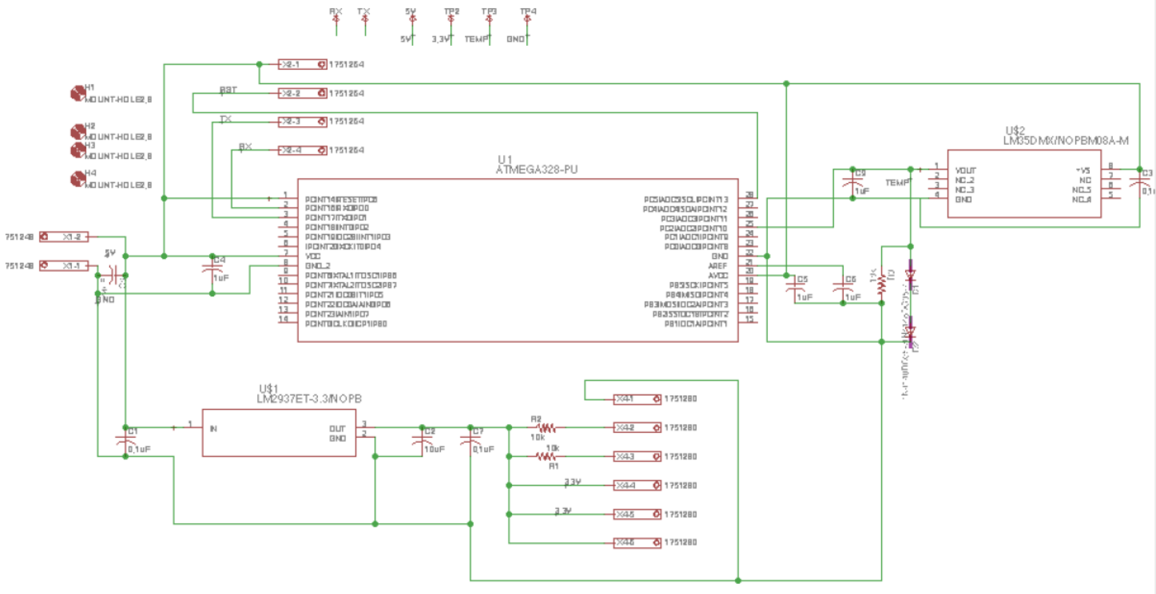

Figure 17: Hub Circuit Schematic

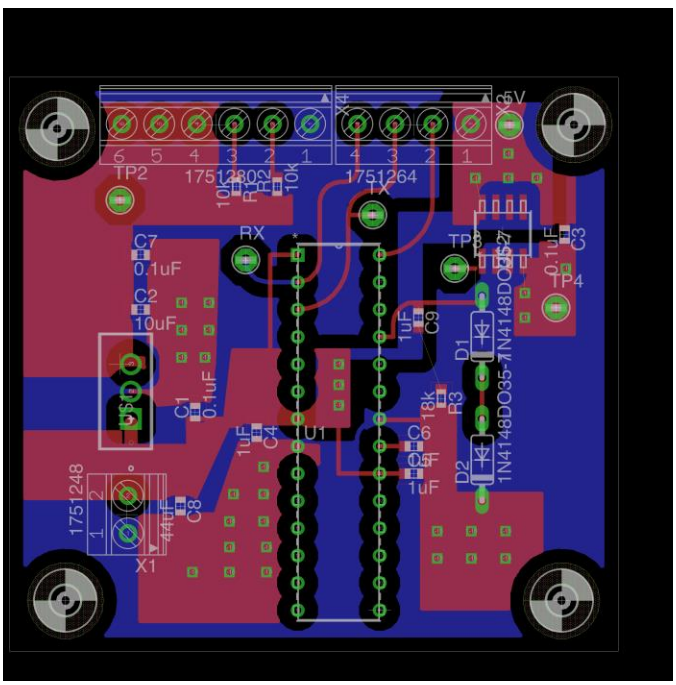

Figure 18: Hub Unit PCB Layout

B.4 Motor Driver

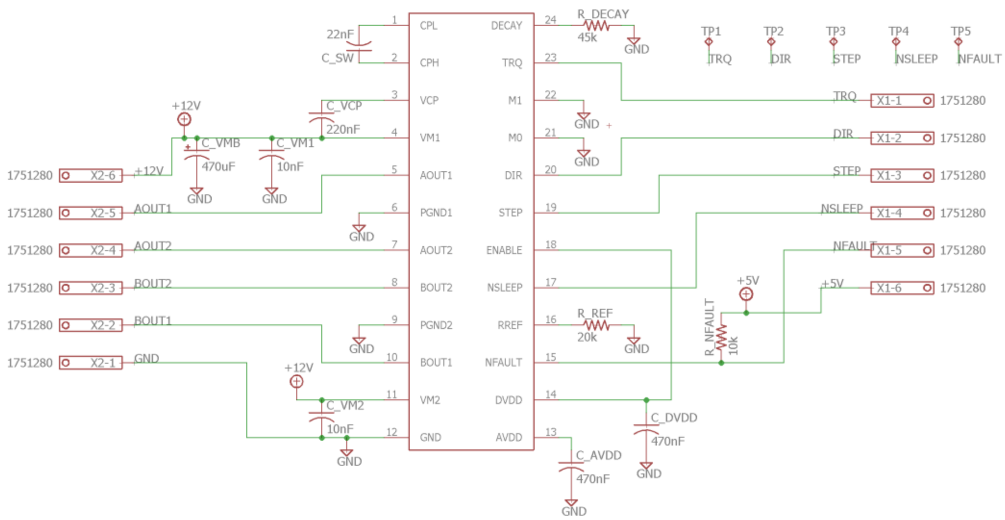

Figure 19: Motor Driver Circuit Schematic

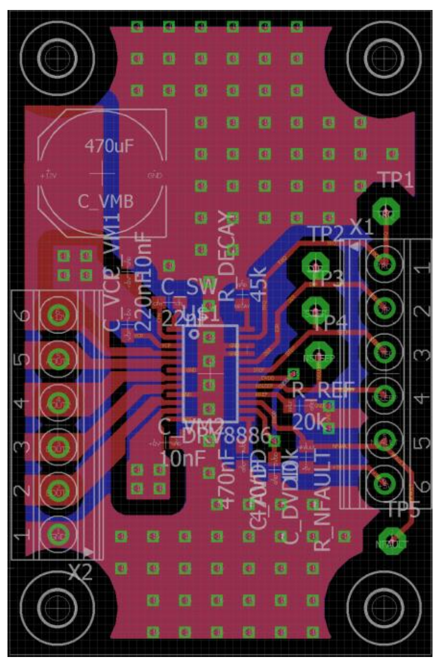

Figure 20: Motor Driver PCB Layout

Appendix COscilloscope Waveforms

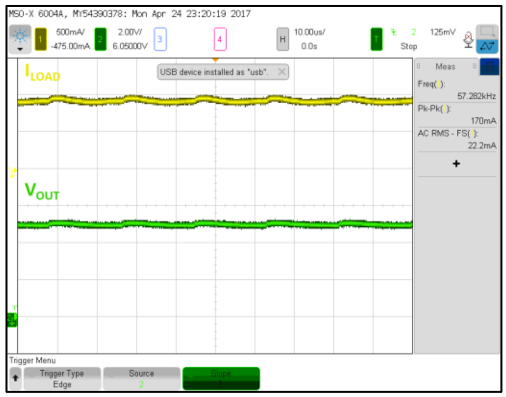

Figure 21: Output Voltage and Current Waveform (5 V AC-DC converter)

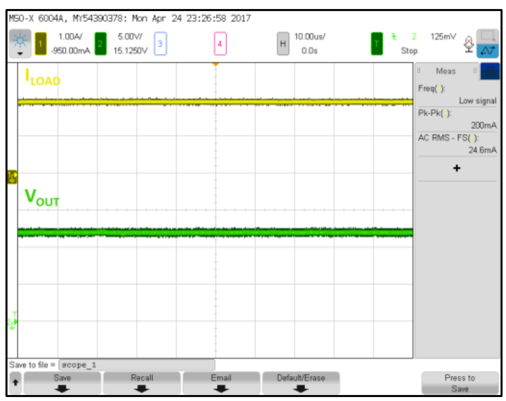

Figure 22: Output Voltage and Current Waveform (12V AC-DC converter)

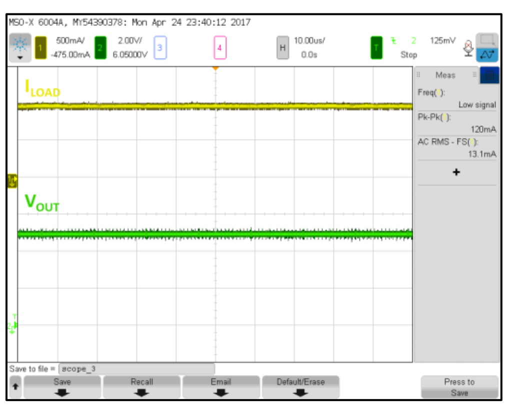

Figure 23: Output Voltage and Current Waveform (12 V AC-DC converter)

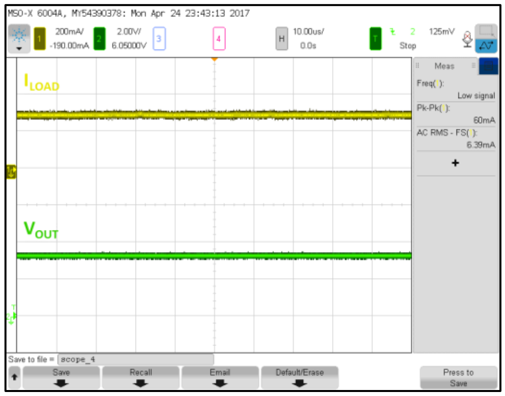

Figure 24: Output Voltage and Current Waveform (5-3.3 V Linear Regulator)

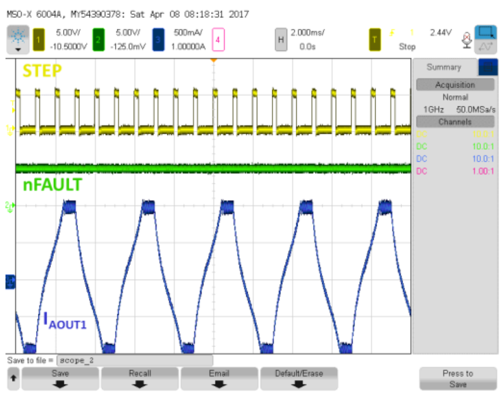

Figure 25: Motor Testing