Weather Adaptive Windows
(Originally FA17 #27 and FA19#22)

Kaiwen Zhao
Canlin Zhang
Yihao Deng

ECE 445 Design Document
Team 53
Spring 2020
TA: Yichi Zhang

Table of Contents

1. Introduction

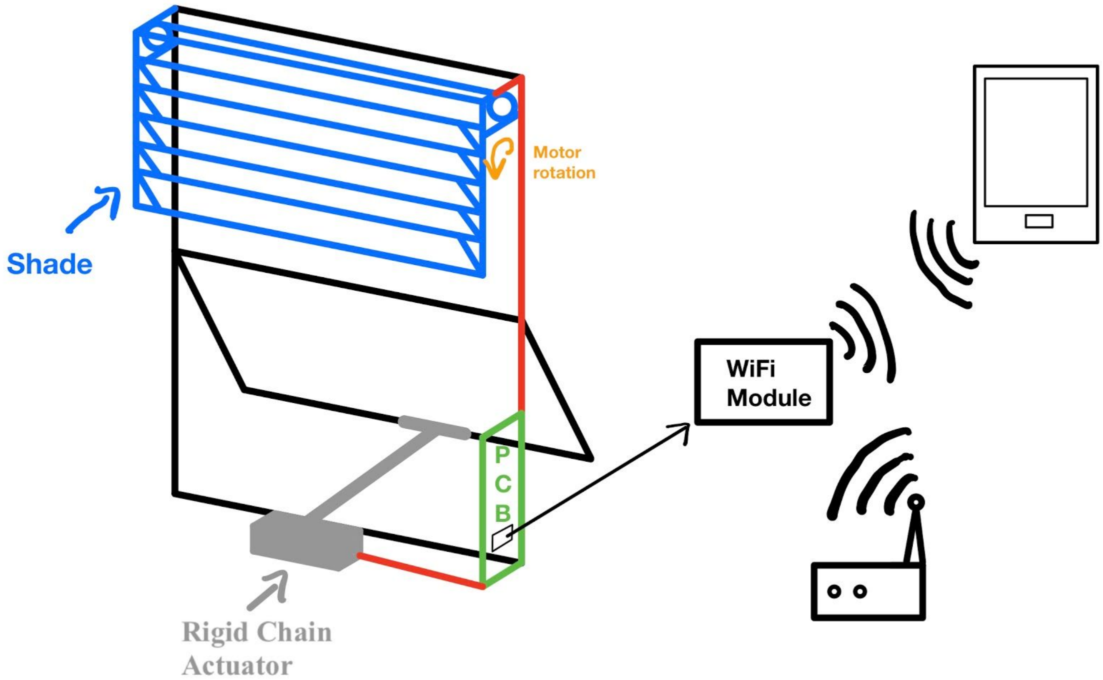
1.1 Objective

Opening and closing windows may seem to be a trivial job in our daily life. However, if people forget to adjust the position of windows in some events like storms, it could lead to serious damage to their properties by flooding. In this case, if the windows can function automatically, it would mitigate the risk of flooding by closing up before storming. Moreover, it would be very convenient and comfortable for customers if the windows can adjust position themselves to regulate the room temperature by allowing a certain amount of breeze to enter the room. This would come in handy when people are sleeping at night. As being integral to the smart-home technology, windows also need to be developed in a way that it will work in a Wi-Fi network. Therefore, we propose a weather-adaptive window that is able to communicate wirelessly and automatically with APIs online by using weather data and with users, and make adjustments itself accordingly.

1.2 Background

We are living in an era that every piece of furniture is starting to be redesigned into IOT devices with some extent of automation and ability to be connected with the IOT network. The idea to integrate windows and shades with some level of automation is clearly not an exotic idea given that many people now have the enthusiasm to make everything automated. There is also a report showing that automation of windows would have a market worth millions of dollars[1]. Nevertheless, there are only a few solutions for automated windows while all the components needed could be found on the market for reasonable cost. Therefore, we would like to design our own solution for an automated window and shade system.

1.3 Visual Aid

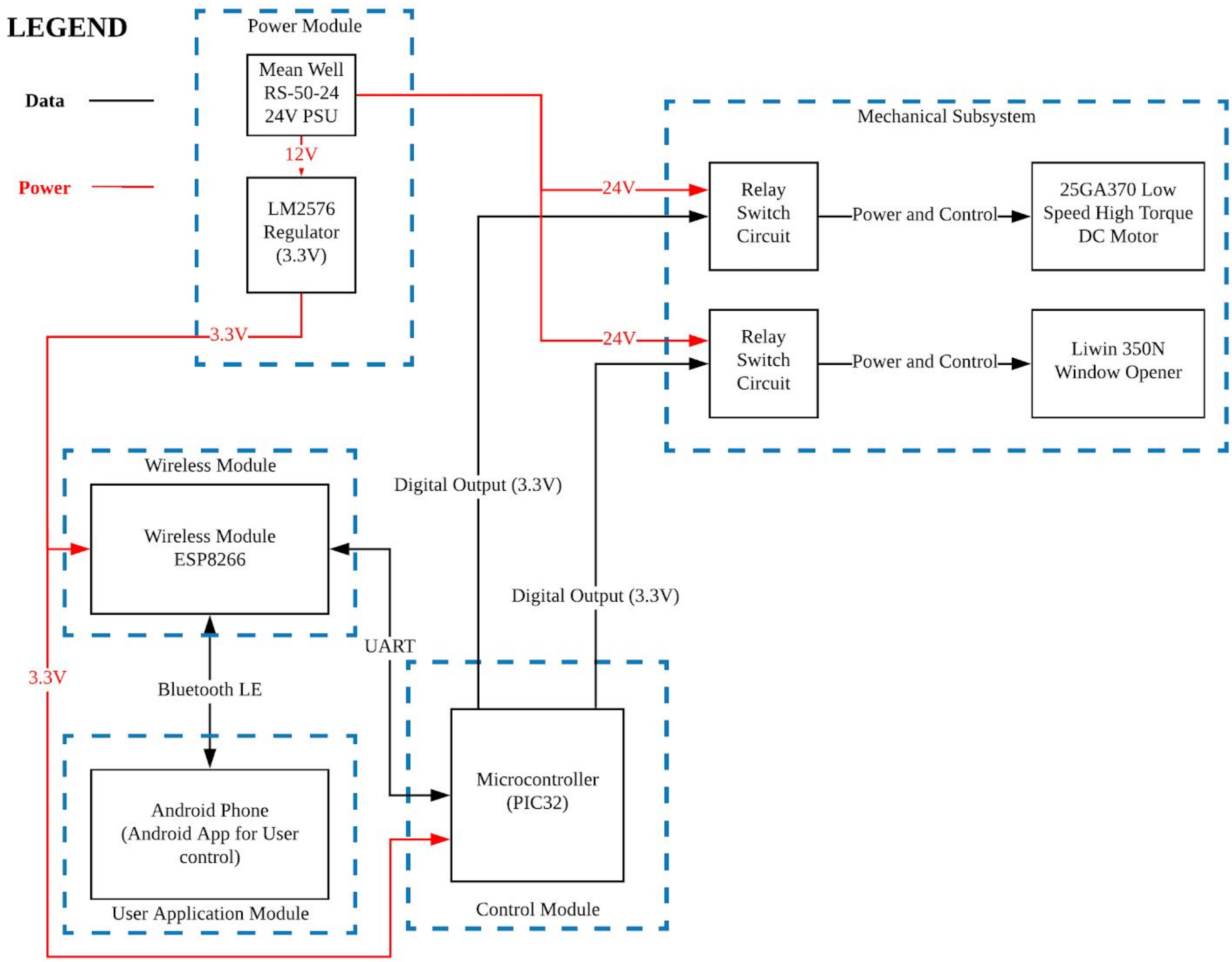


1.4 High-level Requirement List

- The control subsystem can correctly fetch local weather data from the Internet and process the data into according commands.
- The user can view the status of the window and control the angle of the window as well as the angle of shade slats from an Android application.
- The window can open/close to specific angles and the shade slats can adjust to angles according to local weather data from the Internet based on the user's location.

2. Design

2.1 System Overview and Block Diagram



2.2 Control Subsystem

2.2.1 Microcontroller board(MCU)

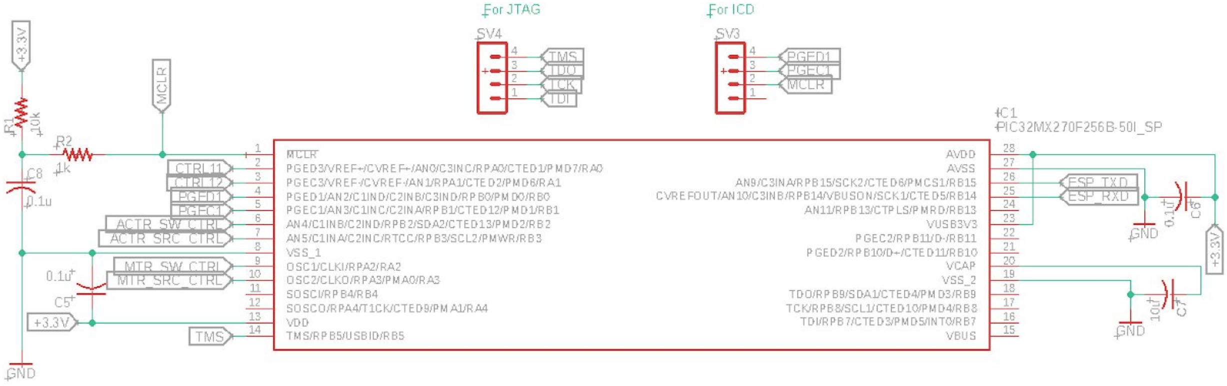


Fig: MCU circuit schematics

The MCU's role is two-folded:

1. The MCU controls the motor for shades and the actuator for opening the window through PWM and digital signals. The MCU will output control signals to the mechanical components either based on direct user commands or weather data fetched from the Internet.
2. The MCU communicates with the wireless module (ESP 8266) to establish connections to the user's android phone or the Internet to fetch weather data.

We used a PIC32 MCU because it is one of the few MCU which comes in DIP packages, which makes soldering and assembly much easier. Also, it has sufficient SRAM and flash memory to work with, and it has enough output pins to control the motor, actuator and wireless module.

Requirements	Verification
1. The MCU must be able to have precise control of the shade motor and window actuator. 2. The MCU must be able to correctly fetch weather data from the Internet. 3. The MCU must be able to establish communication with the user's Android Phone: The user could send data to the MCU and MCU could report data back to the user. 4. The MCU must be able to detect runtime hardware or software errors and close the window and shades in case of emergency or system failure. 5. The MCU must be able send users warning messages through email in case there is extreme weather or system failure through Gmail API.[11]	1. Program the MCU to output different PWM and digital signals to the shade motor and window actuator, then observe the behavior of the shade motor and window actuator to determine whether it could precisely control the motor and actuator. 2. Program the MCU to fetch weather data based on a known location, then compare the fetched weather data to other credible sources. 3. Program the MCU and Android Application to send test data packets between MCU and the user's Android phone 4. Disable the wireless module or do other damages on the system and check whether MCU reacts to the error

	correctly. 5. Program the MCU to use Gmail API to send email to the user’s email address and test whether the mailing functionality is working properly.

2.2.2 Wireless network module

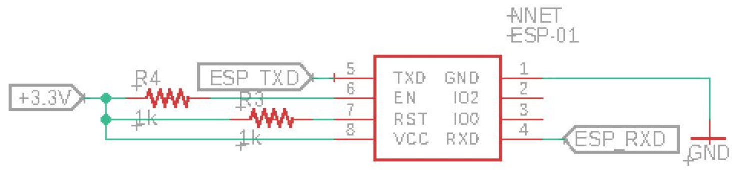


Fig: ESP8266 ESP-01 Wifi module circuit schematics

The role of the wireless network module is to:

1. Connect to the user's phone through wireless network.
2. Connect to the user’s home wireless network to access the Internet for weather data.

We choose ESP8266 because:

1. It is widely available and the API/Interface is well documented.
2. It has relatively low costs and decent performance for our usage scenario. (802.11n) [2]
3. It has low power consumptions. [2]

Requirements	Verification
1. The wireless module must be properly initialized by MCU. 2. The wireless module must be able to connect to a (secure) wireless network (WPA2) and access the Internet. 3. The wireless module must be able to generate its own wireless network for the user’s android phone to communicate with the module.	1. Check status registers after initialization to check whether the wireless module is properly initialized. 2. Control the wireless module to connect to a test network and fetch a certain webpage. Use wireshark to inspect the packets between router and wireless module to check whether a good connection is established. 3. Generate a wireless network by sending commands to the wireless module, then use an Android phone to check whether the phone could connect to the wireless module.

2.3 User Application

2.3.1 Location Information

The user application should obtain the user's current location for the MCU to gather weather data. The source of the location data would be from Google Location Service API, which obtains precise location using both the Internet and GPS module on the user's Android phone.
The format of location would be longitude and latitude data, up to precision of 4 decimal places: Under this precision, the control subsystem could identify up to an individual street.

Requirements	Verifications
1. The user application must be able to obtain the user's precise location using Android's location service API.	1. Verify the accuracy of the location by launching and using the application in a location with known address, then enter the gathered location data into Google Map to verify the correctness of location.

2.3.2 Full Auto control parameters

The user application should allow its users to set certain control parameters for full-auto window/shade control. This would allow the user to adjust the workings of window/shade control with more degrees of freedom.
Some useful control parameters include:

1. Threshold temperatures for the window to open to a certain angle, such as 40 degrees for 75 F.
2. Weather conditions to trigger a window open or close. Such as closing windows during rain or heavy wind.
3. Whether to close the window and shade during extreme weather.

Requirement	Verification
1. The user application must allow the user to adjust and apply the parameters mentioned above.	1. Verify whether user changes in control parameters affect the behavior of the MCU and mechanical subsystem.

2.3.3 Manual control mode

The user application allows the user to have full manual control of the window and shades. That is allowing the user to open the window to any angle (Within the mechanical limits) and pull up the shades to any length. (Within the mechanical limits.)

Requirement	Verification
1. The user application must allow the user to have full manual control to the extent stated above.	1. Generate test user inputs in the user application and see whether user inputs control the mechanical subsystem correctly.

2.3.4 Status report

The user application could request the current status of the system from the MCU. Status messages include current weather, location and any hardware or software errors.

Requirement	Verification
1. The user application must be able to gather the status message stated above from the MCU.	1. Send testing data request packets to the MCU and check whether the correct status message could be fetched.

2.4 Power Subsystem

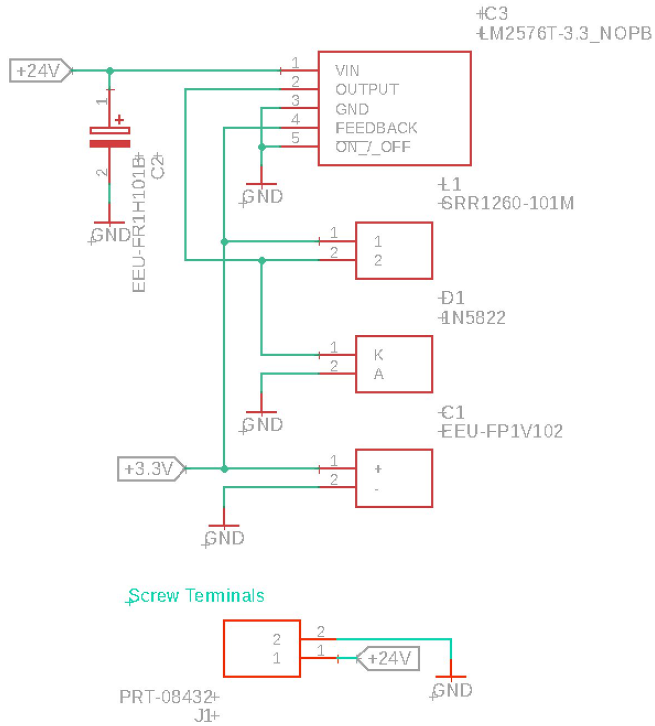


Fig: Power subsystem schematics

2.4.1 AC-DC converter

The AC-DC converter, in this case Mean Well LRS-50-24, should be able to deliver power that is greater than the sum of the rated power consumption of 24V and 3.3V devices in the whole system.

Components	Quantity	Voltage (V)	Power (W)
PIC32 MCU	1	3.3	0.08A * 3.3V = 0.27
Liwin 350N Electric Window Opener	1	24	1.0A * 24V = 24
25GA370 Low Speed High Torque DC Gear Motor	1	24	0.1A * 24V = 2.4
ESP8266 ESP-01 WIFI Module	1	3.3	0.17A * 3.3V = 0.57

Total Power Consumption			27.24

The main components of the whole system would consume 27.24W in total. Multiplying this number by 1.5 would give results of 40.86W. A AC-DC power supply with power rating of 50W, such as Mean Well LRS-50-24, would meet the requirement of power for the whole system.

Requirement	Verification
1. Provide at least 40.86W at 24VDC (+-5%)	1. Connect a TE Connectivity HSC10012RJ 12Ohms wirewound resistor to the power supply and test if the voltage across the resistor remains 24V

2.4.2 DC-DC converter circuit

The DC-DC converter circuit should step down from 24VDC to 3.3VDC to supply PIC32 MCU and ESP-01 WIFI module. The 3.3V output should supply at least the maximum power consumption of MCU and WIFI module which is 0.84W. To ensure the tolerance in the design, the converter should be able to deliver 2W at 3.3VDC.

Requirement	Verification
1. Provide 2W at 3.3VDC(+-3%)	1. Connect a potentiometer to the output of LM3940 and sweep the resistance lower to 5.44ohms and measure if the voltage across the resistive load is at 3.3V(+-3%) consistently.

2.5 Mechanical Subsystem

The mechanical subsystem contains an automated window and an automated blind. Both the rigid chain actuator (working at 24VDC and drawing 1A) for the window and the low speed high torque DC motor (working at 24V and drawing 0.1A) would be driven by 2 channels of L293DNE quadruple half H-bridge IC capable of consistent 1A per channel.

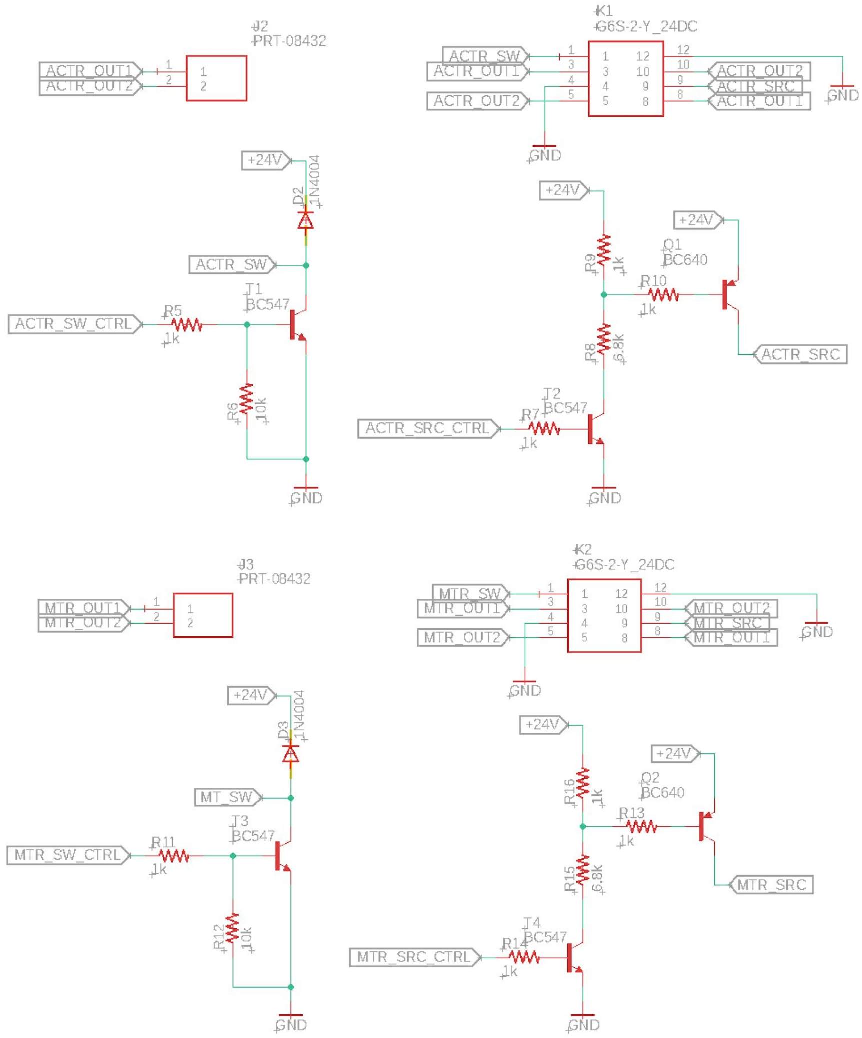


Fig: Actuator and motor driver circuit schematics

2.5.1 Top hinged window with rigid chain actuator

The automated window would be a combination of a top hinged window and a rigid chain actuator. The rigid chain actuator is capable of delivering 350N and works at 24VDC. It would be driven by the circuit combined with a DPDT relay and BJT switch circuits as shown in the schematic above.

As shown in the schematic, the actuator would be connected to ACTR_OUT1 and ACTR_OUT2. The direction of the actuator would be controlled by the DPDT relay. The coil of the DPDT relay is controlled by the NPN BJT switch circuit. When the ACTR_SW_CTRL signal is high, the voltage across the coil would be 24V and when the signal is low, the voltage across would be 0V, resulting in reversed polarity in the connection to the actuator. The NPN-PNP BJT switch circuit controlled by ACTR_SRC_CTRL signal would supply 24V to the relay when ACTR_SRC_CTRL is high, driving the actuator. When this signal is low, the actuator would not move.

Requirement	Verification
1. The window could be pulled close and pushed open using force lower than 250N. 2. The relay and BJT switch circuit would be able to supply 24V (+- 4V for voltage across collector and emitter of PNP transistor) in either polarity to the actuator and could be turned off. 3. The rigid chain actuator can provide at least 250N out of 350N claimed on datasheet	1. Use a hanging scale with a range over 250N to pull the window at the lower edge from both sides and read the number of Newton to see if it is lower than 250N at all angles. 2. Build the circuit as shown in the schematic. Using multimeter, test the polarity of ACTR_OUT1 and ACTR_OUT2 and see if it has a magnitude of 24V(+-2V) when ACTR_SRC_CTRL is high and ACTR_SW_CTRL is low. Pull ACTR_SW_CTRL high, test if the polarity of the two output ports are reversed. Last, Pull ACTR_SRC_CTRL low and expect 0V across the output ports. 3. Find a proper weight laid on ground and measure the friction between the weight and ground. Add weight until the friction is close to 250N. Use the rigid chain actuator to push and pull the weight on the ground and see if it would overcome the friction of 250N and move.

2.5.2 Motorized blind

The motorized blind would be a combination of a blind shade and a low speed high torque DC gear motor installed inside the bar at the top of the blind controlling the angle of the pieces. The brushless DC motor is also controlled by the same circuit as the circuit controlling the actuator.

Requirement	Verification
1. The gear motor is able to control the angle of the blind.	1. Install the gear motor inside the top bar of the blind and connect it to the circuit (tested in window part). Drive the motor and see if the blind turns.

2.6 Tolerance Analysis

The WiFi module is critical to the success of our project since our project relies on gathering and processing online weather data for the automated system to make any adjustment. To provide real-time forecast data to the system, we have to ensure the WiFi module is able to receive and transmit signals within the common 2.4GHz 802.11b/g/n band. Therefore, the location of the WiFi module and its receive sensitivity need to be considered carefully.

Assuming the WiFi signal is propagating in free space without encountering any obstruction (no gain, loses, and fade margin), the maximum/free-space path loss (FSPL) in dB can be calculated using the formula below[13]:

$$FSPL = \text{transmit power(dBm)} - \text{receive sensitivity(dBm)}$$

And the relation between FSPL and the distance from source to receiver in meter is given by the following [9]:

$$d = 10^{((FSPL + 27.55 - 20log10(f))/20)}$$

Where f is the frequency in megahertz, and a constant 27.55 is used, which can be different depending on the unit selections of d and f.

Given that the power of most WiFi access points transmit is 20dBm[8], the estimate distance can be evaluated using the frequency we are interested in (2402-2484MHz) and the lowest receive sensitivity (-95dBm) of our WiFi module from data sheet [10]:

$$FSPL = 115\ dB,\ d = 5492m\ (\text{use } f = 2442MHz)$$

The data rate in this case is minimized down to 1Mbps. If we want to achieve the highest data rate (with receive sensitivity equal to -67dBm), FSPL is then calculated to be 87 dB and the distance is 215m. However, there are many factors, such as wall, humidity or cable connection, that would bring up the attenuation rate of the WiFi signal in the real world. It is impossible to take everything into account, but we have to make a better estimation.
Assuming there are 8 dry walls (each one with a attenuation of 4dB [12]) inside a house that the WiFi signal has to go through and we want to have 11Mbps data rate (receive sensitivity of the WiFi module is now -80 dBm),

$$FSPL = \text{transmit power} - \text{receiver sensitivity} - 8 * 4\ dB = 68dB,\ d = 24.12m$$

This distance is appropriate for our purpose, but the data rate can be lower to increase it since fetching online weather data does not require a high data rate. Moreover, the assumption of 8 walls is excessive for a normal house, but it leaves enough room to account for the other losses.

3. Project Differences

3.1 Overview

The past project implemented sensors for detection of the ambient environment, which may provide incorrect data due to sensor failure and are very costly. Our design would just fetch weather data over APIs online and decide the opening angle of the window based on the weather data. Current weather data has been increasingly accurate, and for telling the current weather in a neighborhood, these weather data will suffice. Also, the design has a fail-safe feature which closes the window and shades during extreme weather or system failure. This will make the design generally safer even without equipping rain sensors or wind sensors to detect hazardous weather.

In addition, the window contains an automated blind limiting sunlight passing the window (which is a different type from the past project). Adding control for the blind adds a new degree of freedom for indoor temperature control and potentially reduces the times of opening and closing the window.

3.2 Analysis

The biggest improvement of our design over the original is using weather data from the Internet instead of using local rain sensors and temperature sensors for detecting weather changes. Currently, the accuracy of weather data is considerably good: A 5-day forecast has an accuracy of 90%. [14] In this case, we are using real-time weather data, which guarantees almost 100% accuracy. In contrast, even professional, agricultural use rain sensors have wide variances in terms of accuracies, from 27% to 97%. [15] Also, using rain sensors means that the rain sensor has to be mounted outside the window. This may lead to damages because of storms or other extreme weather.

Another improvement in the new design is the automated shade would control the amount of sunlight into the room, preventing the room temperature from getting too high in summer and too low in winter. On a clear day, the energy density of solar radiation would be approximately 1000W/m^2 for a surface perpendicular to the Sun's rays at sea level [16]. Assume that the window has an area of 2.5m^2 and the Sun's rays form 60 degrees with the earth. The incoming radiation would be 1000 * 2.5 / 2 = 1250W. Assume there are 4 windows in an apartment, the total incoming radiation would be 5000W, more than the power of an air conditioner. The efficiency of the radiation is higher than an air conditioner because most of the radiation would be trapped inside the room. Using a white plastic shade, more than half of the radiation could be reflected, making a huge difference in the room temperature.

4. Cost and Schedule

4.1 Cost Analysis

The average hourly wage for an Electrical Engineer in the United States is $34/hr [7]. Assuming we need 10 weeks to finish the project from design to product. We have 3 people working on the development of this project, and each of us works about 15 hours per week.
Estimate hours = 3 x 10 x 15 hours = 450 hours
Estimate cost of labor = $34/hour x 450 hours x 2.5 = $38,250

Part	Cost
1x 50kg spring weight scale (Walmart)	$12.43
1x Microcontroller (Mouser; PIC32MX270F256B-I/SP)	$4.03

1x ESP8266 ESP-01 Wifi Module (eBay)	$1.79
1x 24V DC 107 RPM Gear Box Electric Motor (Walmart)	$16.77
1x Mean Well LRS-50-24 Switching Power Supply (Amazon)	$11.49
1x Liwin 350N 24V Electric Window Opener (eBay)	$67.20
2x LM2576-3.3 BT (Digi-key)	$1.74
6x BC640 PNP transistor (Newark)	$0.99
12x BC547 NPN transitor (Arrow)	$4.32
2x G6S-2-Y DC24 DPDT Relay (Digi-key)	$7.64
10x PCB Board (PCBWAY)	$40.00
RCL Components (Mouser)	$15.00
Customized Windows Triple Glaze Double Hung PVC Awning Window (Alibaba)	$70
Pleated Light Filtering Fabric Shade (Amazon)	$19.96
Shipping costs	$30
Total	$303.36

Total cost = $38,250 + $303.36= $38, 553.36

4.2 Schedule

Week	Kaiwen	Canlin	Yihao
1	Test power circuit and measure the 24V and 3.3V to see if they meet the requirement; Order parts	Test the functionality of the MCU board. Initialize wireless module through MCU control.	Research window specifications for the power circuit; Order window

2	Test rigid chain actuator with the window; Install the gear motor inside the blind shade and test using Arduino.	Test more functionality of the wireless module, including status registers and wireless network connection.	Test MCU; Arrange connections in between MCU and the mechanical system
3	Design first version of PCB	Start developing a basic Android application which could connect the user's phone to the MCU through wireless network.	Program the MCU for controlling the mechanical submodule
4	Finish PCB design and order it from manufacturers	Develop and test the functionality of MCU controls for the window actuator and shades motor. Start to further develop the user application.	Test the mechanical submodule using MCU, and identify the possible range of parameters for user input
5	Communicate with group members over software design and combine it with hardware and mechanical design	Develop and test the functionality of weather data gathering and full auto controls.	Test the PCB; Assist teammates on integrating all submodules
6	Solder components onto PCB and test	Test the control system and wireless module as a whole.	Test and mount components onto PCB
7	Debugging	Debugging	Debugging
8	Mock Demo & Debugging	Mock Demo & Debugging	Mock Demo & Debugging
9	Demonstration & Mock Presentation	Demonstration & Mock Presentation	Demonstration & Mock Presentation
10	Presentation & final report	Presentation & final report	Presentation & final report

5. Ethics and Safety

There are few safety concerns in our project. The first one is the power conversion safety issue. The AC-DC converter has an overvoltage CAT III rating which needs to be handled carefully. The power supply with power rating of 50W and 24V falls under the Class 2 of NEC standard, requiring considerations including the location, humidity and being properly grounded[5].

The second concern is the malfunction of the mechanical subsystem. This can happen due to the power shutdown inside the building. However, this event is likely caused by a storm or hurricane, and which will be predicated by the online weather data. So the control system will make decisions ahead of time and close the window to prevent flooding.

The third concern is information safety. For the control subsystem to process weather data, location information is needed. The user enters the location information into the user application, and location information is sent to the control subsystem through wireless network. The location information contains the exact address of the user, traceable to the exact street name, zip code, city and state, and the longitude/latitude information. This is sensitive information which, if fetched by attackers, might be used for malicious purposes. To ensure information safety, the source of information must be secure, and the life-time of sensitive information should be kept as short as possible. The user application will delete the location information right after it is sent to the control subsystem, and the location information will be gathered from Android location services API, which provides adequate security. [4]

Since our design will avoid all the sensors in existing projects, there will be no short circuit caused by exposing any sensor outside the window. With the change of the design and the disclosure of any safety issues stated above, we therefore uphold the IEEE Code of Ethics #1[6].

6. Citations

[1] WhaTech, 'Recent research: Global Window Automation market worth US$ 7970 mn at 6% CAGR forecast by 2025', 2019. [Online]. Available: https://www.whatech.com/market-research/consumer/588419-global-window-automation-market-worth-us-7970-mn-at-6-cagr-forecast-by-2025-supply-demand-market-research. [Accessed: 03-Apr-2020]

[2] SparkFun, *ESP8266 Module (WRL-13678)*, 2019. [Online]. Available: https://cdn.sparkfun.com/datasheets/Wireless/WiFi/ESP8266ModuleV1.pdf [Accessed: 03-Apr-2020]

[3] PIC32MX1XX/2XX 28/36/44-PIN, 32-bit Microcontrollers (up to 256 KB Flash and 64 KB SRAM) with Audio and Graphics Interfaces, USB, and Advanced Analog. Microchip Technology, Inc., 2016. [Online]. Available: https://www.mouser.com/datasheet/2/268/60001168J-967542.pdf [Accessed 03-Apr-2020]

[4] Danny G., *Android Developer's Guide to the Google Location Services API*, 2015. [Online]. Available: https://www.toptal.com/android/android-developers-guide-to-google-location-services-api [Accessed 03-Apr-2020]

[5] Industrial Automation & Controls, 'Technical Note NEC Class 2 DC Power Supplies Revision 2 / May 3, 2002', 2002. [Online]. Available: https://escventura.com/manuals/sola_introNECclass2_rg.pdf. [Accessed: 3-Apr-2020]

[6] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 14- Apr- 2020].

[7] Salary.com, "Hourly wage for Electrical Engineer I," *Salary.com*. [Online]. Available: https://www.salary.com/research/salary/benchmark/electrical-engineer-i-hourly-wages. [Accessed: 14-Apr-2020].

[8] P. Riihikallio, "8 reasons to turn down the transmit power of your Wi-Fi," *Metis.fi*, 13-Apr-2019. [Online]. Available: https://metis.fi/en/2017/10/txpower/. [Accessed: 17-Apr-2020].

[9] "Free-space path loss," *Wikipedia*, 14-Feb-2020. [Online]. Available: https://en.wikipedia.org/wiki/Free-space_path_loss. [Accessed: 18-Apr-2020].

[10] Ai-Thinker Inc, "ESP-01/07/12 Series Modules User's Manual." [Online]. Available: http://wiki.ai-thinker.com/_media/esp8266/esp8266_series_modules_user_manual_v1.1.pdf. [Accessed: 17-Apr-2020].

[11] Gmail API Triggered Flag. [Online] Available https://www.instructables.com/id/Mail-Man/ [Accessed: 17-Apr-2020].

[12]“The Basics of Signal Attenuation,” *CAS Dataloggers*, 17-Jan-2018. [Online]. Available: https://www.dataloggerinc.com/resource-article/basics-signal-attenuation/. [Accessed: 17-Apr-2020].

[13] C. Downey, “Understanding Wireless Range Calculations,” *StackPath*. [Online]. Available: https://www.electronicdesign.com/technologies/communications/article/21796484/understanding-wireless-range-calculations. [Accessed: 17-Apr-2020].

[14] “How Reliable Are Weather Forecasts?”, SciJinks, [Online]. Available https://scijinks.gov/forecast-reliability/ [Accessed: 17-Apr-2020]

[15] Leah M, “Long Term Expanding-Disk Rain Sensor Accuracy”, *ASCE Library*. [Online]. Available: https://ascelibrary.org/doi/10.1061/%28ASCE%29IR.1943-4774.0000381 [Accessed: 17-Apr-2020]

[16] “The Sun's Energy”, *UT Institute of Agriculture*. [Online]. Available: https://ag.tennessee.edu/solar/Pages/What%20Is%20Solar%20Energy/Sun's%20Energy.aspx [Accessed: 17-Apr-2020]