



UNIVERSITY OF CENTRAL FLORIDA

Senior Design II, Summer 2019
Senior Design II Final Documentation

Project Title: **SunShade**

Group number: **13**

Conor Ferring, Electrical Engineering

Alex Rosado, Computer Engineering

Terrell Richardson, Computer Engineering

Miloud Ennemar, Electrical Engineering

Table of Contents

1.0 Executive Summary	6
2.0 Project Description	6
2.1 Project Motivation and Goals	6
2.2 Objectives and Features	7
2.3 Project Requirements and Specifications	8
2.3.1 Hardware Specifications	8
2.3.2 Software Specifications	9
2.4 Additional Features	10
2.5 House of Quality	10
3.0 Research related to Project	11
3.1 Existing Similar Projects and Products	11
3.1.1 Existing Products	11
3.1.2 Existing Projects	12
3.2 Relevant Technologies	18
3.2.1 Communication Comparisons	18
3.2.1.1 WIFI versus Bluetooth	18
3.2.2 Motor Comparisons	21
3.2.2.1 DC motor - Metal DC Geared Motor 6V 15RPM 10kg.cm	22
3.2.2.2 Stepper motor- Entweg Metal Stepper Motor 17HS4401S-PG518	23
3.2.2.3 Servo Motor -JX servo PDI-5521MG Control Angle 360	23
3.2.2.4 Motor with Drive Shaft	24
3.2.3 Power Source Comparisons	24
3.2.3.1 Lead-Acid flooded	24
3.2.3.2 Absorbed Glass Mat	25
3.2.3.3 Lithium Ion	25
3.2.4 Solar Cell Technology	27
3.2.4.1 Monocrystalline Cells	27
3.2.4.2 Polycrystalline Cells	28
3.2.4.3 Flexible Thin-Film Photovoltaic Cells	28
3.2.5 LED	29
3.2.5.1 Dotstar	29
3.2.5.2 NeoPixel	29
3.2.6 Smart Home Integration Packages	30
3.2.7 Environment Sensors	31
3.2.7.1 Light Dependent Resistor	31
3.2.7.2 DS18B20 Temperature Sensor	31

3.2.7.3 DHT22 Temperature Sensor	32
3.2.8 Voice Control Research	33
3.2.8.1 Amazon Echo	33
3.2.8.2 Google Home	34
3.3 Possible Architectures and Related Diagrams	34
3.3.1 Microcontroller Units Architecture	34
3.3.1.1 Von Neumann Architecture	35
3.3.1.2 Harvard Architecture	36
3.3.1.3 ARM Architecture	37
3.3.2 Motors Architecture	38
3.3.3 Power Distribution Architecture	40
3.3.4.1 Solar cell module	41
3.4 Power outputs	44
4.0 Related Standards	44
4.1 Electrical Standards	44
4.1.1 IEEE Standards	44
4.1.2 Other Electrical standards	45
4.2 Communication standards	46
4.3 Safety standards	46
4.4 Battery standards	47
5.0 Realistic Design Constraints	48
5.1 Power Constraints	48
5.2 Economic Constraints	48
5.3 Time Constraints	49
5.4 Weight Constraints	50
5.5 Ethical, Health, and Safety Constraints	51
5.5.1 Ethical Constraints	51
5.5.2 Health and safety Constraints	52
6.0 Hardware Design & Component Testing	52
6.1 PCB	52
6.2 Blinds physicality	54
6.3 Power	55
6.3.1 Battery Source to 12V Voltage Regulation	55
6.3.2 12V to 6V Voltage Regulations	56
6.3.3 6V to 5V Voltage Regulator	57
6.4 MCU	57
6.5 Sensors and LEDs	58
6.5.1 Light Dependent Resistors	58
6.5.1.1 Photodiode Light sensors	60

6.5.1.2 Photovoltaic Cells	60
6.5.2 DHT22 Temperature Sensor	60
6.5.3 NeoPixel LED Strip	62
6.6 Motors	63
6.6.1 Tilt Motor	63
6.6.2 Lift motor	65
6.6.3 Shade Motor	66
6.7 Window Frame Construction	67
6.7.1 Frame Design	68
6.7.2 Frame Construction	69
6.8 Voltage Regulator	70
6.9 Hardware Design Summary	77
6.9.1 Comparison to 2015 Solar Blinds Project	77
6.9.2 Schematic	79
7.0 Software Design	80
7.1 Firmware	80
7.1.1 Arduino Programming	81
7.1.2 I2C Communication	82
7.1.3 UART Communication	83
7.2 Mobile Application	84
7.2.1 React Native	84
7.2.2 iOS/Android Design	85
7.3 Database	87
7.3.1 Amazon Web Services	88
7.3.2 RESTful API	89
7.4 Voice Control	89
7.5 Software Design Summary	91
8.0 Prototype Construction	92
8.1 Parts Acquisition	92
8.2 PCB Layout and Assembly	93
8.3 MCU Programming	95
8.4 iOS/Android App Development	96
8.5 Database Creation	97
9.0 Prototype Testing	98
9.1 Hardware Test Environments	98
9.2 Hardware Specific Testing	99
9.2.1 Stopping Criteria	99
9.2.2 Hardware Test Cases	100
9.2.3 Battery & Controller Tests	103

9.2.4 Motor Tests	103
9.2.5 Voltage Regulation Tests	103
9.2.6 Microprocessor Tests	104
9.2.7 Final Project Testing	104
9.3 Software Test Environment	105
9.4 Software Specific Testing	105
9.4.1 Database	105
9.4.2 Arduino MCU	108
9.4.3 iOS/Android Application	111
10.0 Administrative Content	121
10.1 Milestone Discussion	121
10.2 Budget and Finance Discussion	124
10.2.1 Suppliers	125
10.3 Team Organization and Distribution of Responsibility	125
11.0 Project Summary and Conclusion	127
11.1 Bill of Material	127
11.2 Product User Operations Manual	129
11.2.1 Mobile Application	129
11.2.2 Connecting to the Device	129
11.2.3 User Hardware Installation Guide	129
11.3 Senior Design One Conclusion	130
12.0 Senior Design Two	130
12.1 Hardware	130
12.2 Software	132
12.3 Complications	132
12.4 Conclusion	134
Appendix-A Copyright permissions	135
Appendix-B References	136

1.0 Executive Summary

For Floridians, the sun is a constant facet of everyday life. They call it the sunshine state; we spend all day in its overwhelming heat, and our beaches thrive on its endless light. But what about when we head indoors? By the time the sun is going down, your options are to completely shut it out or face blinding lights streaming into your home every few minutes as it sets. What about for those who have delicate plant life that could face overexposure? Obnoxious television glare after a long day of work? Not to mention the financial toll the Florida sun takes on the monthly electricity bill. Sunlight making its way into your home can push your air conditioning unit to its limit and cost you a pretty penny.

The Sunshade offers the ability to utilize solar power to create an energy saving appliance that also assists in keeping interior temperatures low, as well as provide comfortable natural interior lighting. To do this, the team modified the standard corded blinds found in almost every home, into a voice and app controlled smart system, capable of scheduling, brightness control, and more. Installation is as simple as installing traditional blinds, and application and smart-home voice control linking are a breeze, providing a power-saving, energy efficient source of home comfort.

2.0 Project Description

The following section deals with the general description of what the 2019 Senior Design SunShade project team designed and fabricated over the course of the senior design courses. The section specifies any explicit goals and motivations, as well as objectives, a complete list of features, requirements, specifications and block diagram.

2.1 Project Motivation and Goals

The sunshade team's motivation for the project was to create a smart-home appliance capable of creating a comfortable environment for the user, while being highly cost-efficient. These blinds are lightweight, low cost, and use minimal energy for typical day-to-day functionality. Our primary goal was to design an easy to use a smart-blinds system, capable of raising and lowering standard window-shades, as well as tilt for controlled sun exposure. The project is controlled through a connected smartphone application, Amazon Alexa enabled, or manually, with onboard controls. Because of this goal, our project falls firmly within the realm of technology come to be known as "The Internet of Things", or interconnected home devices for daily use. These devices have grown in popularity exponentially in recent years, and this trend isn't expected to slow down any time soon, and so the chance to dive into this expanding new field of consumer products is an exciting opportunity.

2.2 Objectives and Features

Upon connecting your SunShade to the accompanying smartphone application, users can indicate which direction their window faces, and from there, have a variety of features to utilize. Our objective was to fully realize these features, and create a smart-blinds system capable of, but not limited to:

- App control- this allows the user to adjust the up, down, and tilt blinds positions with ease, having only to lift a finger from the comfort of their homes, or from long distance.
- Automatic brightness control- accounts for direction and exterior brightness to adjust the tilt of your blinds throughout the day in order to only allow a reasonable amount of natural light. This can be useful for keeping a comfortable brightness in the user's home throughout the day. This can be achieved with either a sun-facing sensor, weather information, or temperature sensors.
- Scheduling function- this allows for users to allow for their SunShade to protect their window-dwelling plant life from over-exposure, as well as save the user on monthly electricity costs by limiting the effects of the sun on air conditioning usage. This was done entirely through the application.
- Smart-system enabled- Alexa or Google Home functionalities provide a completely hands-free control for the user. Commands such as "Alexa, shades down.", "Alexa, shades up.", "Alexa, close shades.", "Alexa, open shades." This was done with Skill adding, WIFI chip, and the microcontroller.

Each of these features combine to create a user-friendly and highly versatile product, fitting perfectly within the common smart-home system, while also appealing to energy-efficiency and cost desires.

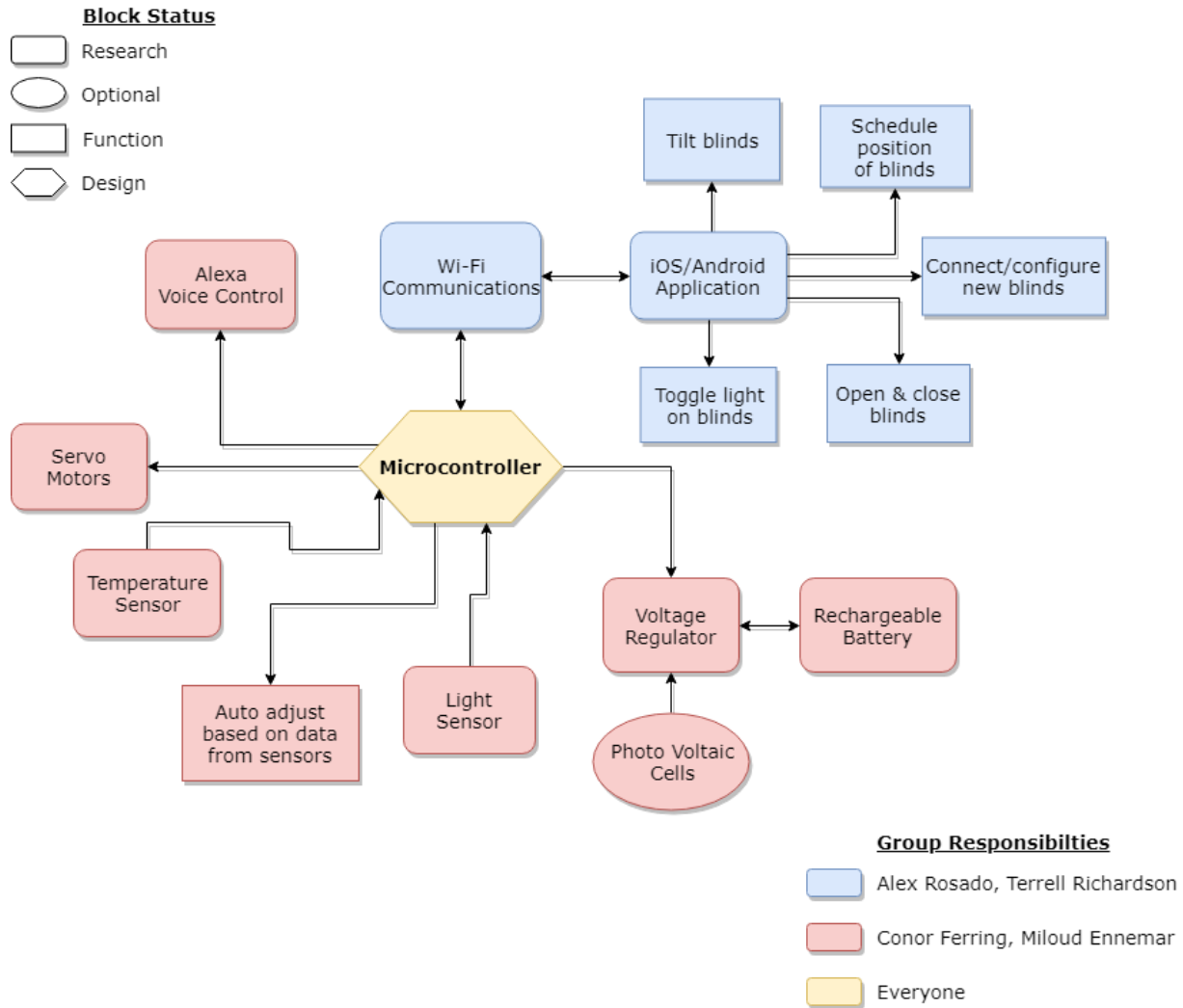


Figure 2.2.1 - Block Diagram

2.3 Project Requirements and Specifications

This section establishes and clearly defines the parameters in which the project needed to meet in order to be considered successful. These requirements and specifications are separated into hardware and software, respectively, and these guidelines were used to explicitly define what would be considered as the finished product.

2.3.1 Hardware Specifications

This is the physical aspect of the project. What is necessary in order for the project to function, and what was originally expected of it as a finished product. These parameters no longer hold in such an exact manner.

Hardware Specifications List

#	Hardware Specification Description
1	Communicate wirelessly to the mobile application
2	Blind panels must tilt for controlled exposure
3	Blinds must ascend and descend on control
4	Manual control buttons must be available
5	Draw energy from either solar panels or charged battery source.
6	Thermometer sensor.
7	Blinds must be compatible with standard windows and frames.
8	Must be aesthetically pleasing.
9	Must be able to charge battery with solar panels
10	Operate motors from the application
11	Wirelessly interact with smarthome devices (Alexa, Google Home)
12	Must be compact and lightweight.

Table 2.3.1.1 - Hardware specifications

2.3.2 Software Specifications

This is the computational and programming aspect of the project. What is necessary in order for the project to function, and what was expected of it as a finished product.

Software Specifications List

#	Software Specification Description
1	Mobile application compatibility for iOS and/or Android
2	Compatibility with smarthome devices (Alexa, Google Home)
3	Must have Scheduling system for days/times
4	Must have ability to adjust based on internal temperature
5	Must have slide-control for tilt of blinds
6	Must have slide-control for ascension and descension of blinds
7	Must send temperature data to application
8	Must adjust based on time of day or brightness
9	Must securely sync with user's mobile device
10	Must be able to control blinds from relative distance.

Table 2.3.2.1 - Software specifications

2.4 Additional Features

Additional features for SunShade include but are not limited to: an expansion to the brightness control functionality, such as a roll-down secondary transparent window shade system, which allows for maximum natural lighting, while cancelling out harsh brightness. We did end up implementing this, and expanded on it further with a Neo-Pixel lighting system along the edge of the blinds system- this is useful because at the end of the day, when the sun is going down and your desired room brightness is no longer achievable through natural light, the integrated Neo-Pixel lighting system compensates for dimming.

Other features kept in mind include a home-security system, which alerts the application if the blinds undergo unexpected high levels of movement, in line with a possible intruder. These ideas have great potential, and should the primary objectives of the SunShade be completed under budget and on-time, these ideas would then shift into development. The security feature was never implemented.

2.5 House of Quality

To help visualize the project goals, we identified the engineering requirements and marketing requirements to show the correlation between the two. The tradeoffs and correlations are shown in the table below.

			Engineering Requirements				
			Efficiency	Installation Time	Power Consumption	Weight	Cost
			+	-	-	-	+
Marketing Requirements	Quality	+	↑				↓
	Functionality	+	↑↑		↑↑	↑	↓
	Control	+	↑↑	↑	↑↑	↑	↓↓
	Cost	-	↓		↑		
	Battery Life	+	↑		↑	↑	↑
Targets for Engineering Requirements			> 90%	< 5 minutes	< 5V	< 3 lbs	<\$400

Table 2.5.1 - House of Quality

Legend:

- + = Positive Polarity (Increasing Requirement)
- - = Negative Polarity (Decreasing Requirement)
- ↑↑ = Strong Positive Correlation
- ↑ = Positive Correlation
- ↓↓ = Strong Negative Correlation
- ↓ = Negative Correlation

3.0 Research related to Project

The following section handles all related research that was performed over the course of the project design phase, to ensure that all design choices came from an educated perspective, as well as to learn about all possible components, compare them with one another, and make critical design decisions.

3.1 Existing Similar Projects and Products

The following section is regarding any existing personal projects or commercial products that may be defined as similar to the SunShade project. Its purpose is to make a comparison between the desired outcome and these existing projects/products, so that the team could learn from their mistakes and successes in order to ensure a high-quality finished project.

3.1.1 Existing Products

When designing a product intended to compete with existing products on the open market, it is crucial to ensure that the designed product is able to meet the level of expectations created by its predecessors. We can use these already-existing appliances in order to succeed where they failed, improve upon existing functions, and learn from possible mistakes. Online user-based reviews are a wealth of information and insight into the opinions of the common consumer, and how they felt about similar projects. In this section we compared the initial design of the SunShade Smart Blinds appliance, to that of commonly found online appliances of a similar functionality. Upon researching the reviews of many existing Smart-blinds products, a number of common problems for consumers begins to emerge. These issues include, but are not limited to:

- Priced too high
- Too difficult to install
- Flimsy feeling
- No external manual controls

With these shortcomings in mind, the proper steps can be taken to avoid the issues of existing smart blinds products. For example, one of the primary constraints of the SunShade project was cost efficiency. Furthermore, instead of a corded attachment which is incompatible with some blinds, or completely non-viable for the common cordless blinds, the SunShade is a fully integrated blinds system, needing only to be inserted into the standard window 'U' bracket and locked into place. The issue of no external manual control has also been directly addressed, as this feature has been stressed as a core functionality of the SunShade project, allowing users to directly adjust their blinds without searching for their phone, or being the app-holding user. Without this feature, only the user synced to the device would be able to change blinds position, which is limiting. With these past mistakes and consumer opinions in mind, it is the desire of the SunShade team to offer a more complete functionality, with the consumer's desires in mind.

3.1.2 Existing Projects

Another valuable source of both inspiration and experience for any project in development, would be existing community-based projects of a similar nature. These projects can be found in a wide range of online communities, such as YouTube, Instructables, Thingiverse, blogs, and forums. Two existing projects are addressed in detail below.

Published on the user-submitted building website 'Instructables.com' [1], a user by the username Biochemtronics created a simple light-activated automatic window-blinds controller, in order to save his potted plants near his windows from both overexposure and underexposure. Similar to the SunShade team, this user felt that existing products ran too high a cost and opted instead to make his own for a low price. Biochemtronics used a common microcontroller, servo motor, and light-dependent resistor, along with various other small parts, to create the motorized window blinds he needed in order to save his potted plants. All necessary parts are disclosed in the table below, totaling up to be around \$20.00.

Part	Manufacturer	Cost
PICAXE -08M Micro Controller	Spark Fun Electronics	\$2.95
Solarbotics GM3 Gear Motor, 224:1 6V	Solarbotics	\$5.50
Light Dependent Resistor	Radio Shack #276-1657 (5 pk)	\$2.99
ULN2003A Darlington Array	BG Micro #ICSULN2003	\$0.59
DPDT 5.0V Relay	BG Micro # REL1106	\$1.29
4 X 1.5V AA Battery Holder	BG Micro #BAT1068	\$0.79
3.5mm Stereo Jack	BG Micro #AUDCA017	\$0.36
Battery Snap (9V style)	BG Micro #BAT1033	\$0.25
LM7805T 5.0V, 1A, Regulator	BG Micro #REG7805T	\$0.40
Small Project Box	BG Micro #ACS1157	\$1.95
Small Proto Board (2 3/8)	BG Micro #ACS1433	\$0.89
(2) SPST Switches	BG Micro #SWT1043	\$0.20
8 Pin Dip Socket	BG Micro #SOC1036	\$0.10
(2) 16 Pin Dip Sockets	BG Micro #SOC1038	\$0.08

Table 3.1.2.1 - Materials for Bio Chemtronics automatic blinds

For Biochemtronic's automatic blinds, the system is run by the PICAXE-08M microcontroller, using an 8-pin DIP. This is a small, and easily programmed controller that also runs on open-source software, perfect for the scale of the project in mind. Unfortunately, it's not quite powerful enough for the SunShade. The light-dependent resistor shown in the table connects directly to the analog-to-digital converter and the PICAXE so that it may effectively monitor the brightness level, and therefore sun activity, outside. The intensity of light is registered as a value, and, if and when this value is exceeded (In this case, between 0-255), the servo motor is activated to turn the blinds open or closed. In order to amplify the PICAXE's low current outputs and switch the high currents necessary for the components like the relay and motor, a device called the Darlington Array was used, which protected the circuit from spikes in voltage. The 5Volt relay listed was used in order to switch between open and closed turns, reversing motor polarity. All of these components interacting with one another in circuit diagram form can be seen in the following figure:



Figure 3.1.2.2 - Outward appearance for Biochemtronic's automatic blinds controller

As far as physical functionality goes, the two projects both manipulate the blind's mechanical system in a very similar fashion, as is it is the most efficient for our purpose. As shown in figure 3.1.2.2, Biochemtronic's design is an add-on to existing blinds systems, which requires the removal of the tilt wand. It can be seen that the motor on the bottom turns the former wand manipulation piece. The primary difference in mechanics is that, while the SunShade used the draw-string pulley system of traditional blinds, Biochemtronic's automatic blinds use a cordless system, where the motor shaft uses a gear to turn the tilt wand, which would typically hang down for a user to spin by hand to tilt the blinds up and down. These systems work in the same fashion where a "tilt rod" is the primary manipulator, the only difference is if it's adjusted with wand or cords. Instead of the gear motor Biochemtronics used, it would be possible for the SunShade team to consider a more expensive option, and instead use a 360-stepper motor in order to allow for greater position accuracy. This would be helpful for brightness control features. The physicality of the blinds system was expanded upon further in blinds design, section 6.2. The PICAXE controller is programmed to read the brightness level provided by the light-dependent resistor and the analog-to-digital converter, which makes the data from the resistor useful digitally, and then defines the parameters for if the current light-level requires the blinds to open or close. This is helpful for the SunShade project because it is a cheap and effective way of measuring external brightness. Expanding on this concept could prove to be very effective in allowing for precision changes and maintaining a consistent interior brightness.

While not quite as versatile as the SunShade, the project provides excellent insight into a home-made smart-blinds appliance made under a budget and with similar desired outcomes. The biggest change between the project discussed in this section, and the Sunshade, is the application. The app opens up a wide variety of feature possibilities, which the team has already expanded into. Biochemtronics achieved a simplistic version

of the desired product with only a microcontroller and motor. It proved to be a very useful source of inspiration for design and troubleshooting throughout the development process.

The second existing project worth noting is a University of Central Florida 2015 Senior Design project as well- titled “Solar Blinds” [2], built by a group of four UCF graduates. This can be considered the grandfather of the SunShade, as it holds many similar qualities as the project at hand, while also maintaining some major differences in design and functionality. The 2015 project ‘Solar Blinds’ is a conventional solar-powered smart-blinds system, with an emphasis on green energy and self-sustainability. The project is different from the SunShade in the regard that instead of using the common “blinds” system, it uses a roll-down shade of sorts, with a flexible solar-panel attached to the window-facing side. The Sunshade has more versatility in brightness control, given the user’s ability to adjust both lift and tilt of the blinds. A similar aspect both projects share is the application connectability, in that the projects can both be controlled remotely with a mobile device application. Listed in the following table are all of the materials that the 2015 group required to construct their prototype.

Part	Manufacturer	Cost
3.3 W 6"x6" Monocrystalline Solar Cells (10)	Sunpower	\$ 46.99
2mm x 0.15mm Tapping Wire (10 m)	MISOL	\$ 6.29
835-P Rosin Flux Pen	MG Chemicals	\$ 8.95
Rosin Core Lead-Free Silver Solder	Trakpower	\$ 11.93
DC-DC LTC1871 Step Up Module	DROK	\$ 10.04
R-14 Rollable Solar Module	Powerfilm	\$ 185.99
UB250 12 V 5 Ah SLA AGM Battery	Universal Power Group	\$ 9.93
GV-5 MPPT Charge Controller	Genasun	\$ 75.00
PCB Assembly	OSH Park	\$ 30.00
Microcontroller CC2640F128RHBR	Texas Instruments	\$ 13.08
Bluetooth Module (on MCU)	Texas Instruments	\$ ---
Wiring	Retailer	\$ 10.00
MintyBoost USB Charger	Adafruit	\$ 19.50
E-paper Display		\$ 21.72
Total:		\$ 449.42

Table 3.1.2.3 - 2015 Senior Design Solar Blinds materials

It can be observed from the table of materials that the prototype was quite expensive, but the biggest load on their budget was the rollable solar panel module. Fortunately, for the SunShade team, such a component wasn’t necessary, as the solar power source is primarily a rigid strip.

The 2015 project works by first taking in power drawn on their rollable panel, and then regulating the current and voltages that are generated to run through the battery. Because of the unpredictability and variation in voltages throughout the day, the group opted to use a pre-built battery charger from solar cells to a lithium-ion battery. They chose to use the Gena-Sun Gv-5 65W 5A solar charge controller with MPPT, which monitors and maintains incoming voltage near-constantly. They used Eagle to create their PCB layout, which the Sunshade team used as well. Each component part is designed

into separate libraries with identification, which is dropped into the full schematic to be routed and cleaned up. Prototyping was done on LTSpice, but it's more likely the Sunshade team used Multisim due to existing familiarity. The microprocessor chosen was the CC2640 due to its Bluetooth capabilities and support, as well as pin-count. This micro also needs a twenty-four-megahertz crystal. All incoming/generated voltage is regulated down from 'twelve to three' volts using a linear voltage regulator, which powers the micro itself, and then a separate 'twelve to five' linear voltage regulator to power the USB charger port, and an e-paper screen display to show battery level and external temperature. The processor is loaded into LTSpice in order to be digitally simulated and tested with the rest of the virtually realized components, which contributes to the Eagle PCB design. As the brains of the operation, designing the microprocessor into the PCB is a critical step requiring delicate work, research, and double-checking.

Next, once the microprocessor does all the heavy lifting, the motor is operated with high voltage from one of the I/O pins, and turns the motor clockwise or counterclockwise, in order to open or close the blinds system by winding up the shade or releasing it. This kind of physical response in the motor would be influenceable from the application, in accordance with the desired capabilities. Another way the 2015 project diverges is that it utilizes the excess energy generated by their solar panel, in order to provide the user with a way to charge any non-apple mobile device, which is useful, given that the blinds are operated by said mobile phones.

Included in the 2015 project is also an antenna circuit, used to communicate with the mobile devices using the application. On the application, users had the ability to first register a device to sync, connect and disconnect to an existing, or registered device, and then manipulate position to open or close the blind, by communicating with the PCB to turn the motor clockwise or counterclockwise. The user also had the option to check the system status, to read data like battery level, external temperature, and charge status. Their user interface was relatively simple and minimalist, where it is our goal to expand upon this in order to allow a sleeker appearance with a wider variety of options for the consumer. Overall, while having some considerable differences in the final look and variation of features, the 2015 project is a wealth of information for the SunShade team, as it functions as a predecessor in regard to both goals, and motivation. It provides an extremely detailed look into how a project of similar design and fashion is created from the ground up, in an extremely educational format for engineers who are just entering the field of major project design and fabrication.

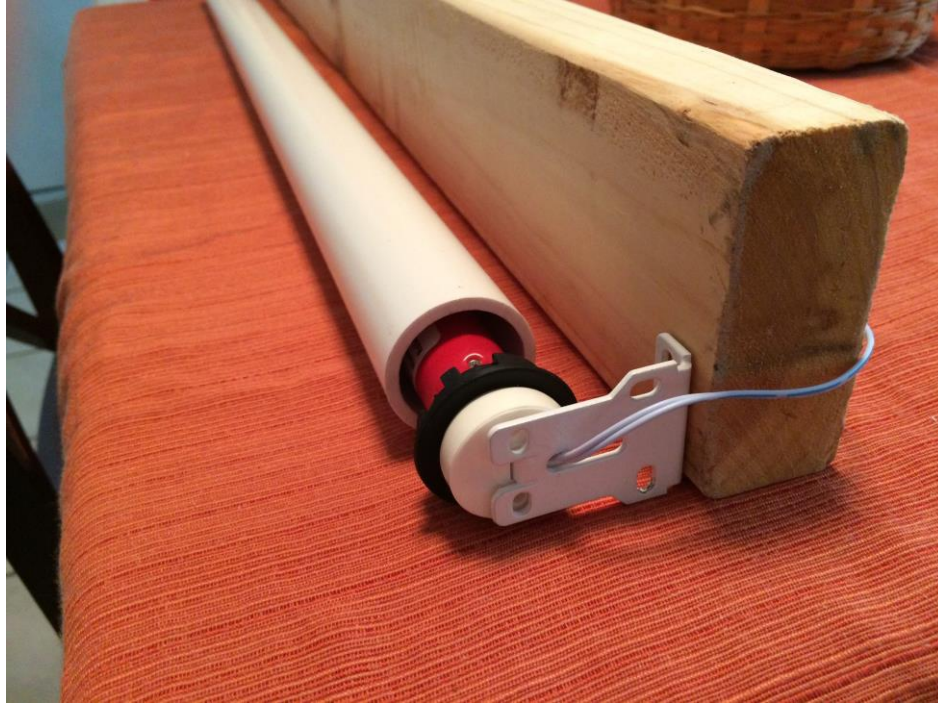


Figure 3.1.2.4 - 2015 'Solar Blinds' project mechanical/motor system.

The above image is useful because it provides insight into an effective design for mounting the motor so that it can turn the tilt-rod clockwise or counterclockwise, similar to the 2015 project, as well as a design inspiration for the possible addition of a diffuser shade.



Figure 3.1.2.5 - 2015 'Solar Blinds' project window-facing side and wooden window frame.

3.2 Relevant Technologies

Relevant technologies are any pre-existing technologies that may become useful over the course of the design and fabrication phases and allows for the SunShade team to make informed decisions based on the research at hand, so that the best possible components for the project can be chosen.

3.2.1 Communication Comparisons

In order for the SunShade smart blinds appliance to perform as intended, it needed internet connectivity in order to effectively communicate between the device and the application. This communication will allow the user to activate and adjust the shade system from the comfort of their mobile device and maintain a proper user-designed scheduling system. Because of this functionality, the project is left with a choice: WIFI-router or blue-tooth? Which was the primary chosen method of connectivity? In this section, both options were explored in-depth, so that an informed decision can be made. Comparisons were made between data sheets, potential modules using the chosen component path, and then compare the specifications of said modules.

3.2.1.1 WIFI versus Bluetooth

While both WIFI and Bluetooth are methods of transmitting and receiving wireless communication over varying distances, their core differences can be found in their design and performance. Each have their own most effective uses, and most people interact with one or both of these systems every single day. While WIFI is primarily used in order to provide access to the internet via a router at high speeds, Bluetooth is mainly used to create a wireless connection between two or more devices over short distances, without any cables or physical connections. This range for Bluetooth is around thirty feet, which is great for internet-of-things devices, such as printers, smart-speakers, mobile phones, and headsets, but has limitations in versatility and functionality. WIFI enables a faster connection over much greater distances, as well as better security, but at the cost of being expensive and more complicated to integrate.

Like Bluetooth, it serves to wirelessly connect, but rather than communicate between two or more devices, it instead connects these devices directly to the internet or ethernet (Like LAN Networks). This proves to be a major benefit for the SunShade project, because the continued direct internet connection provided by WIFI allows for the device to become a part of an Internet-of-things network as well, where so long as the controlling application and the device are both connected to the internet, they may communicate, allowing for relatively limitless range, a massive benefit over Bluetooth. This is not to say that WIFI is superior to Bluetooth, but to say that they each have their own benefits and shortcomings. Below, a figure has been included in order to compare the differences and performance between both WIFI systems and Bluetooth systems. From this, it is desired that a choice can be made for which best suits the SunShade project and its desired uses.

	Bluetooth	WiFi
Cost	Low	High
Year of development	1994	1991
Bandwidth	Low - 800Kbps	High - 11Mbps
Range	5-30 m	32m indoors, 95m outdoors. Antennae can increase range, and connection to a server can amplify range to near unlimited
Specifications authority	Bluetooth SIG	IEEE, WECA
Security	Less secure	More secure
Hardware requirements	Adapters for Bluetooth within all connected devices.	A router, and adapters for Wifi connectivity on within all connected devices within the network.
Primary uses	Mobile devices, Internet-of-things, Mouse & Keyboards, headsets, personal speakers, athletic trackers.	Desktop computers, laptops, printers, servers, Television, Mobile phones.
Power usage	Low	High
Latency	200 ms	150 ms
Bit-Rate	2.1 Mbps	600 Mbps
Ease of use	Simplisitc. Can be utilized to interconnect a handful of devices, and can be easily switched between these connected devices, or locate new connections.	More complex, due to requirement of router setup, software and hardware configurations.

Table 3.2.1.1.1 - Side-by-side comparison for Bluetooth Vs WIFI

From the table, the choice narrows itself down, and the SunShade team comes much closer to a final decision. Breaking down the table, we see that the average cost of

a WIFI module is actually higher than that of a Bluetooth, and as a self-funded university project, cost plays a major role in the process or component selection. Fortunately, upon further research, it can be seen that the price may not be too vast to compensate for, the real issue is the cost of a router, which, even on the low end, run for about sixty to eighty dollars. Both devices require adapters to be built into corresponding devices for communication purposes. Bandwidth is not a major issue because the communication between devices was relatively small, but when it comes to range, the largest issue is brought to attention. The SunShade system and accompanying app requires wireless communication, but the matter of how significant this range must be, is still up for debate. The primary benefit of long-range communication with WIFI is that the user would be capable of adjusting their smart-blinds appliance from anywhere, even from all the way at the office, or another city or state, due to the fact that both ends are connected to the internet. Bluetooth on the other hand, is limited to around thirty feet, which is perfectly fine for usage within the home that the appliance resides, but it begs the question of if this product merits the kind of long-range usage that WIFI offers. With the function of the scheduling system on board, could users frequently need to adjust their blinds from afar without foresight? That is to say, could this alleviate any user problems that scheduling cannot? The element that pushes this debate one way or another is the smart-home voice-controlled features. Appliances like the Amazon Echo, also known as Alexa, and Google Home, have certain abilities that allow for “skill learning” which would allow us to interface the SunShade with these devices, to allow for voice-activated control. This, however, requires WIFI connectivity. Other devices like speakers can sync up with nothing more than Bluetooth, but the Sunshade project requires back and forth communication. If the Sunshade team wishes to implement voice control, WIFI is essential.

The data presented does specify that WIFI is more secure than Bluetooth, which is a point in WIFI’s favor, but security is not necessarily a pressing issue with the Sunshade project, because of the likelihood of blackhat hackers making it their mission to raise users electric bill and kill their houseplants. One interesting aspect also presented in the data comparison above, is that while Bluetooth is typically used for things like keyboards and mice, headsets, and speakers, WIFI, which typically is for desktops/laptops, servers, and printers, may actually be better for the project’s desired purposes, and may actually align better with WIFI than in Bluetooth, despite the fact that the Bluetooth examples are of a similar internet-of-things categorization.

Two aspects that actually work against WIFI’s favor as our selection, is the higher demand of power usage/consumption, which would require a greater power source and a greater strain on solar-powered elements of the project, but also the ease of use being significantly more complex in terms of implementation and user setup. Fortunately, once this user setup is done, including the syncing process and app configuration, this issue is null for the user. It is because of these pros and cons that the SunShade team has chosen WIFI as its primary method of wireless communication between appliance and mobile device. The long-range capabilities, as well as voice control potential are simply too great benefits to ignore.

3.2.1.2 Potential modules using WIFI

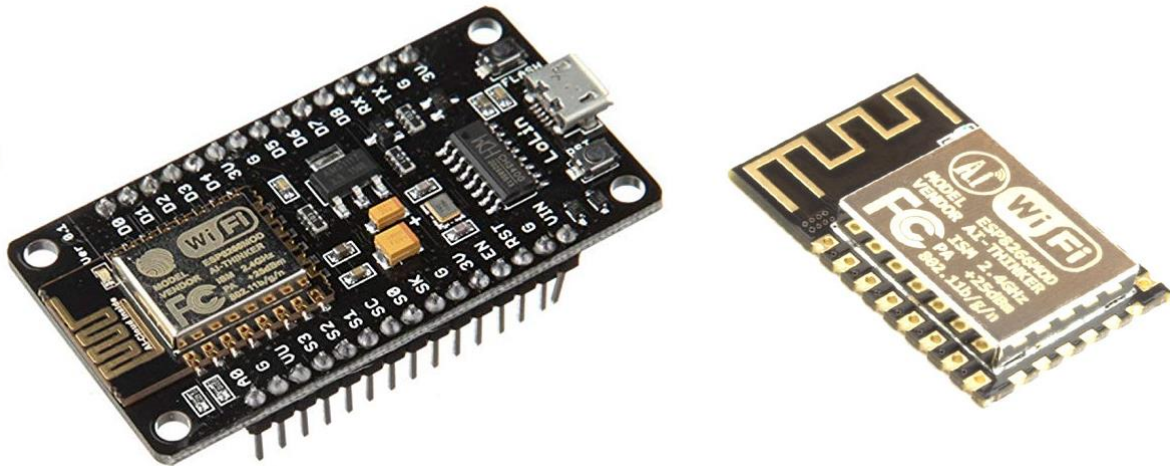


Figure 3.2.1.2.1 - JacobsParts ESP8266 ESP-12E and ESP-12E - WIFI Module

The above image displays the JacobsParts ESP8266 ESP-12E WIFI Microcontroller USB Development Board, which is compatible with Arduino, which the SunShade project team used for their prototyping phase, as well as the actual ESP-12E WIFI module, which can be seen on the Jacobsparts component. The smaller ESP-12E was implemented directly into the project PCB for the final product version. In a description from the manufacturer, they describe the ESP-12E module as a “shielded breakout board that combines the ESP8266 with 4MB of flash and a PCB antenna. This development board integrates the ESP-12E module with a CH340G USB TTL adapter chip, power converter circuitry, buttons, and breadboard compatible pin headers.” Which would be an ideal component in prototyping. It typically costs around \$7. The WIFI module intended for PCB use, is described as being perfect for Internet-of-things related projects and is easy to implement to allow for connection to WIFI networks.

Further diving into the specifications of the WIFI module, we learn that the active power consumption is around 750 mW, which is very high, relative to Bluetooth, and will have to be considered when designing the power supply system. Fortunately, the idle power consumption is so small, it can be considered negligible. In regard to interference, the common WIFI frequency ranges are between 2.5 Ghz and 5 GHz, where the smaller, 2.5 GHz frequency is more proficient at moving through complicated pathways, such as the layout of a home, due to its longer wavelength, compared to the 5 GHz system. Unfortunately, due to this benefit, the 2.5 GHz is fairly common, meaning that it opens itself up to possible interference from other devices or routers operating at the same frequency, within a close proximity. Due to the nature of the appliance and its intended use, this issue is considered to be of little consequence.

3.2.2 Motor Comparisons

Our group has chosen to use horizontal faux wood blinds to convert into a smart blind. In order to allow the user full functionality of the blind’s general mechanical features,

we would need two motors. The blinds need one motor that will attach to the hexagonal bar that controls the tilt, and the other motor to spool the cord that adjusts the blinds height. We are focused on delivering cordless automation for consumer satisfaction. If we are able the budget and other constraints allow, the design will need an additional optional motor for a rolling diffuser shade. The optional shade design will introduce more configuration and efficiency options to the user.

There are a variety of ways other home projects and manufactures have used motors to motorize blind features. The goal is to find the motor that is cost effective and facilitates the features we want to achieve. Motors come in vast varieties that all have specific features. The cost of motors ranges from around \$10 to \$100 for practical motors the design calls for. For one of the two main motors, we will need a motor that provides some stall torque to support blind weight. We want the motors to allow for precise movement that can be tracked by encoders or hall effect sensors.

Through research we found two motor styles that provide the performance and features needed. The stepper motor and servo motor provide the ability for stall torque. With a simple dc motor, we would need to design a pulley system to help with weight support. The optional motor is a motor with drive shaft but can also be achieved with the stepper motor or servo motor.

3.2.2.1 DC motor - Metal DC Geared Motor 6V 15RPM 10kg.cm

The DC motor is the best choice for keeping the budget low, but what the motor lacks will add more complications. Dc motors are not known for torque but luckily, they make geared DC Motors to compensate. This geared DC motor costs around ten dollars, and it is rated to have a torque of ten kilograms per centimeter. The motor will provide the lifting force needed but will lack the ability to encode. This motor design will need to add a frame to mount the motor to itself and the blind mechanism, as well as, it will need to be coupled with some type of encoding device. The device is cylindrical shaped with mounting connections only available at the back end. Designing a mount that will withstand the draw torque without spiral fracturing was a challenge. We need an encoding device to track the zero position of the blinds, and to allow the ability to recalibrate blinds that have been moved manually. This will cost extra for the added sensor and the parts for the sensors to be coupled. The motor itself is driven with analog signal only, so writing code to finagle precise reactions was more complex. The assets to this motor are cost, working voltage, and speed. The motor has rotational speed that will satisfy a customer, and the working voltage is manageable with running off batteries. The motor does not fill in enough assets to be selected to drive either one of our features.

A previous senior design project looked into a Leeson Motors DC motor. This motor is a bit overkill with 12-24 volts and 4200RPM. This is not geared motor, but the motor has plenty of power to provide lift. The size of the motor is cumbersome, bulky, and will detract from aesthetics. The cost of the motor was around 100 dollars which is above a reasonable budget. This motor is not a proper choice for this design.

3.2.2.2 Stepper motor- Entweg Metal Stepper Motor 17HS4401S-PG518

The stepper motor has many positive attributes that can deliver on the feature set we need. The motor can be driven by an analog or digital signal. The motor has the ability to provide stall torque that can support the blinds. The motor does not have an encoder, similar to the dc motor, but the way the motor is driven allows for programmable encoding. Stepper motors are brushless by design so they will suffer less wear. This motor select has a square chassis that will fit perfectly in the blinds top frame to pull the drawstring. This will make mounting a simple task with this design. The motor in question is also geared which will add additional torque. The major drawback is the amount of voltage and current needed. The motor in question need 12-24 volts for working voltage, which the design should be able to operate efficiently on 12 volts. The amount of current draw is a critical setback for our design. The motor needs 1.68 amps per phase, which can easily draw more current during lift. The design calls for up to three motors, logic board with sensors, and LED lights. All these components will need to run on a battery that is capable of delivering all day performance. The battery is charged during the day with solar. With that amount of current draw, heavy use of the device will drain the battery. It may also have the ability to stall the other devices causing unnecessary reboots. The price of the motor is also above the budget with it costing around 45 dollars for one motor. Although we will only need one motor with stall torque, this motor will drive up the budget alone. A smaller stepper can still be used for the tilt which would cost around 10 dollars. This may seem like a decent price, but there is one last issue that couples with stepper motor. Stepper motors need to be coupled with a motor driver which are an additional cost. The motor driver sends the appropriate signals to the stepper motor for precise control of the device, otherwise the device will act like a DC motor. This motor is not an efficient choice for this design.

3.2.2.3 Servo Motor -JX servo PDI-5521MG Control Angle 360

Servo Motors have a lot of features that are pros in the case of this design. Servo motors are geared so they provide torque ratings. There are a variety of servo motors constructed for hobbyist projects that fit the bill for two of our motors. These motors are also affordable and compact. We reviewed previous senior design products and they looked into the pricey industrial grade servos. If you dig a little deeper there are a plethora of lightweight servos with adequate ratings. Most of the servos come coupled with some sort of encoder, which can be a hall effect sensor or potentiometer. This is perfect for our project to offer precise controls without additional costs. Many of the hobbyist servos come rating by degree of rotation. We would need a 180-degree servo costing around 10 dollars, and a 360-degree high torque continuous rotation servo costing around 20 dollars. These prices are well within budget, including the fact that the servos are compact and easy to mount. There is one caveat to the 360-degree servo, and the con is that the servo does not always include an encoder. There are a few that are rated too low for our needs, as well as, a few servos that are complete overkill. Although there is this negative aspect, the servo includes brackets that can easily mount objects to its rotating shaft. This feature will lessen the difficulty of coupling an encoder to that one specific motor. The

motor selected for the 360 degree has a 6-volt peak working voltage, and it is rated for 20 kilograms per centimeter stall torque. This specification is perfect for the needs of a system.

In order to have the ability to share the same power supply with minimal modifications, the 180-degree servo chosen is also rated to 6 volts. It does not need to have high torque so the rating was around 5 kilograms per centimeter. The servo will still be a high-quality metal gear servo, so that the servo will wear less during repetitive use.

3.2.2.4 Motor with Drive Shaft

I included this motor because it was a perfect fit for our optional design. The motor type was selected for utmost convenience. This motor design serves one major purpose and that is reeling in a screen, for example a projector screen. We will make use of this to roll and unroll our diffuser shade. The cost is around 50 dollars for the motor. This will affect our power rail because the motor needs 12 volt supply. The motor runs on with up, down, and off commands. The group we were previously mentioned used this type of motor for their design. The motor seemed to work exceptionally.

3.2.3 Power Source Comparisons

The blinds will need to run off of battery power, and the battery was charged by solar panels. The battery bank should at least last for a full day without a charge on predetermined moderate use case. There are plenty of components that will need energy such as logic, motors, and lights. This will raise considerable concern that the battery needs to have a fairly large capacity. Typically having a large capacity comes at a cost with money and weight. The battery needs to be rechargeable so that will narrow the results down. There is one possible device that will need 12 volts otherwise a 6 volt bank can be used. Batteries sadly have not evolved too much over time which leaves us with a few considerable options. One last factor to pay attention to was the current draw at max load. The concerning component that will draw the most current was the servo motor that lifts the blinds. This motor can draw 3 amps under max load conditions which needs to be accounted for with battery selection. The three main battery types that are practical are: Lead-Acid, Absorbed Glass Mat (AGM), and Lithium Ion. The decision was made on these requirements: weight, cost, longevity, efficiency, capacity, maintenance, and environmental impact. All the selected models are known to be rechargeable and compatible with off-grid solar panel arrays.

3.2.3.1 Lead-Acid flooded

The most commonly used battery on the market. This battery has been around for ages, and it is the battery types we use in automobiles. The lead acid battery is cheap to produce and cost the least of the three. The style of battery has 50% use efficiency, and

the battery will deteriorate rapidly if it discharges under that threshold. The battery requires maintenance to preserve the longevity of the device. The preservation also prevents failure of the battery. This battery model can leak corrosive environmentally unfriendly material. This could be detrimental to a customer who may obtain property damage or bodily harm from such failure. This will also be a hassle to constantly check the battery. These batteries if left unchecked not only can leak, they produce hydrogen over time within their cells. If the hydrogen is not released within time, the battery can potential become harmful and explosive. The battery tends to be generally heavy, as well as, not come in a variety of compact forms. These batteries with proper maintenance can last up to around 10 years. In the case for this design choice, the battery should have the least visible and physical footprint. The lead acid battery may be great for quick, affordable, and hassle-free setup, but for the vision of this project will not be a great fit.

3.2.3.2 Absorbed Glass Mat

AGM batteries are a type of sealed lead acid batteries. These batteries do not require maintenance like its flooded cousin. This battery is typically used in battery backup system appliances such as a computer UPS. This will provide the customer with a greater quality of life. Their longevity is slightly less than the flooded lead-acid. They come with the same 50% efficiency rate but run for less discharge recharge cycles than the flooded. They have a lower failure rate as for leakage than flooded. The batteries tend to be heavier than their flooded cousins, but they do come in more variety of compact forms. This will allow for less of a visual footprint. The cost per watt hour is more. This product is more convenient than the flooded lead-acid, but it still does not manage to fill that right fit.

3.2.3.3 Lithium Ion

Lithium Ion batteries are the most recent technologies of the three. This battery comes in a variety of chemistries, designs, and applications. The lithium battery is used in a large variety of appliances now. The battery is efficient using 80% of its capacity. On top of that fact, the battery has the most outstanding longevity out of the three models. A lithium ion battery can last up to 20+ years with around 4000 cycles depending on manufacturing standards. This makes the battery a premium choice for customer satisfaction. The customer will not have to worry about maintaining or replacing the battery for a hefty amount of time. These features do come at a major cost and straight from the bank. They are the most expensive to manufacture so they are also the most expensive to purchase. The cost of these products are more reasonable now than they have been in recent years. The cost is generally a 5-10 dollars more than the lead-acid counter parts. The two chemistries that contribute the most promise are lithium iron phosphate and lithium manganese oxide. The iron phosphate batteries are used for industrial applications that may need endurance and current. It also has one of the best chemistries for longevity underneath lithium titanite. There are a number of designs that are compact enough to fit in the blinds frame chassis. The battery has reasonable voltage

per cell around 3.7 operating. The manganese oxide is generally used in appliances like power tools. These cells have a decent amount of cycles performing peak around 700 cycles. The operating voltage is around 3.8 volts. This model is already used to run motors with more intensive current draw than the intended use. I believe it still was a reasonable fit. There are many more lithium chemistries on the market. The choice will definitely amount to the cost but the fit is this option for sure. It will allow for comfortable amount of flexibility especially with the amount of shapes, sizes, and styles available.

*Lithium Manganese Oxide: LiMn2O4
cathode, graphite anode*

*Lithium Iron Phosphate: LiFePO4
cathode, graphite anode*

Voltages	3.70V (3.80V) nominal; typical operating range 3.0–4.2V/cell	3.20, 3.30V nominal; typical operating range 2.5–3.65V/cell
Specific energy (capacity)	100–150Wh/kg	90–120Wh/kg
Charge (C-rate)	0.7–1C typical, 3C maximum, charges to 4.20V (most cells)	1C typical, charges to 3.65V; 3h charge time typical
Discharge (C-rate)	1C; 10C possible with some cells, 30C pulse (5s), 2.50V cut-off	1C, 25C on some cells; 40A pulse (2s); 2.50V cut-off (lower than 2V causes damage)
Cycle life	300–700 (related to depth of discharge, temperature)	2000 and higher (related to depth of discharge, temperature)
Thermal runaway	250°C (482°F) typical. High charge promotes thermal runaway	270°C (518°F) Very safe battery even if fully charged
Applications	Power tools, medical devices, electric powertrains	Portable and stationary needing high load currents and endurance

Comments

High power but less capacity; safer than Li-cobalt; commonly mixed with NMC to improve performance.

Very flat voltage discharge curve but low capacity. Most Safe Li-ions. Used for special markets. Elevated self-discharge.

Table 3.2.3.3.1 - Compiled Lithium battery information from Battery University

3.2.4 Solar Cell Technology

In order to eliminate the hassle of wires, solar panels were used for daytime charging of the internal battery. Based off of what is available, we have an option between three different configurations. The market readily has monocrystalline, polycrystalline, and photovoltaic flexible thin-film solar panels. Solar panel setups work similar to battery back setups. If more voltage is needed, then connect each panel in series. To increase the amount of amperage the panels are connected in parallel. This feature can be used in unison to increase voltage and amperes. Since most of the components are low powered, the configuration should supply the same voltage as the battery. The system can probably get away with 0.5 amp current, but for insurance, the configuration will aim for around 1-2 amps. This can allow the capability of running the devices during peak times directly off solar and will allow the battery back to not be cycled too often. After each type of solar cell has been explored, the specification comparison was provided in Table 3.2.4.3.1.

3.2.4.1 Monocrystalline Cells

This cell is the most efficient of the three models. The cell has the greatest efficiency per inch, which allows for smaller panels creating more space. The cell contains higher grade silicone that goes through a special process to increase its performance. This allows the cell to perform in low light situations. The cell is also the longest lasting of the three. This covers the best assets of this cell, but it does have its drawbacks. If the design does not use micro-inverters, the circuit may be broken by shade or debris. This can pose an issue when the panels are mounted within a windowsill which can limit the devices light source. Performance of these cells suffer with high temperatures. This may not be a complete issue since they are mounted indoors, although it can still impede the devices performance. The cells cost is more at a premium than its poly counterpart. Lastly, a lot of silicone is wasted during their construction process which does not affect the consumer much. All in all, this cell still offers best efficiency per square inch, which it can make the difference when the footprint needs to be less intrusive. A customer will not want their window real estate to be massively inhibited by solar panels. If the customer's view was blocked to that point, would a device like that really worth its convenience?

3.2.4.2 Polycrystalline Cells

These cells are the original solar panel designs, and they are the least expensive of the three. For example, an amazon vendor sells ten, six by six, five-volt poly cells for ten dollars. This is an incredible value for a budget. The cell performs worse than its mono counterpart in high heat. It has a lower efficiency rating than the mono, and it needs more real estate. Thankfully, it does not suffer complete circuit break down with shade or debris, but it will need more light to be effective. This cell can pose more problems with use at a windowsill than its mono cousin. The low-cost incentives are not enough to entice its use. The cell itself also tends to have an unfavorable uneven blue appearance. With the fact that it may look unpleasant and take up more viewing space is jarring.

3.2.4.3 Flexible Thin-Film Photovoltaic Cells

This cell style was chosen in combination with monocrystalline cells in a similar project research for design ideas. Their project used a roller shade only so thin-film material was a considerable choice. This project may use an optional roller shade, but the shades roll in this project is to diffuse light not block it. Mounting one of these solar panels will defeat the purpose but it deserves an honorable mention for its market prevalence. Many different materials make up the cells with different varying qualities. The configuration itself has overall the worst efficiency, and they tend to degrade faster than the crystalline panels. The total cost is over budget for the fact that the panel will take a phenomenal amount of real estate. It is valuable material for projects that need flex and durability. In this case, the needs revolve around efficiency.

Solar Cell Comparison Table

	Monocrystalline	Polycrystalline	Amorphous
Typical module efficiency:	15-20%	13-16%	6-8%
Best research cell efficiency:	25.00%	20.40%	13.40%
Area required for 1 kWp:	6-9 m ²	8-9 m ²	13-20 m ²
Typical length of warranty:	25 years	25 years	10-25 years
Lowest price:	0.75 \$/W	0.62 \$/W	0.69 \$/W
Temperature resistance:	Performance drops 10-15% at high temperatures	Less temperature resistant than monocrystalline	Tolerates extreme heat
Additional details:	Oldest cell technology and most widely used	Less silicon waste in the production process	Tend to degrade faster Low availability on the market

Table [3.2.4.3.1] - Table with solar cell comparisons by Energy Informative

3.2.5 LED

Light emitting diodes were used for accent lighting feature and for pairing confirmation. A current trend is the ability to have custom multicolor LED lighting available at home. This project aims to turn a windowsill into a unique light up centerpiece. This can be done by incorporating a programmable lighting strip to the bottom of the blind chassis. Then adding a light diffuser over it to soften the rays. There are two stand out LED pixel styles available to choose from which include Neopixel and Dotstar. These devices allow for user customization in length and lighting effects. This will also allow for the ability to use this light source for pairing confirmation purposes as well.

3.2.5.1 Dotstar

The Dotstar LEDs major claim to fame is its high refresh rate and fast. Unlike its Neopixel counterpart, they can make use of interrupts with no strict timing restrictions. The hardware interfaces with a wider range of devices. The Dotstars positive features do not compliment the overall projects and the cons really weigh against the higher performing tech. The Dotstar cost more than the Neopixel. It also requires an extra data line for operation depicted below. These two problems really hurt the selection choice toward the Dotstar. It also has less form factor options on the market.

3.2.5.2 NeoPixel

The Neopixel LED is a reasonable fit for this design. The product is readily available in many accessible form factors. The design is using an Arduino AT2650 microprocessor, and there is a complete set of open source tools for this product. The cost of Neopixel is highly affordable and lower than Dotstar. These LEDs are far brighter than the Dotstar but that also means that they draw more current. There only needs to be a VCC, ground, and one data line for this device. This was less complex to set up and configure. The pin layout was depicted below for comparison with Dotstar. The brightness of Neopixel was an excellent feature because it can possibly mimic daytime light at night from the windowsill.

The Neopixel after deliberation seems to be the LED of choice. The one factor is the LED strip density per meter. The strips start at 30 per meter and can squeeze up to 144 per meter. In order to get an adequate even lighting from the strip, a 30 LED per meter strip was exceptional. Going up to 144 LEDs per meter was detrimental to the battery life. This will overall affect the customers quality of life for this product. The size of the strip can be modified for both brands by cutting across the designated line between the pins. Both the Dotstar and Neopixel have datasheets with specifications provided from Adafruit's website are presented in Table 3.2.5.2.1. They are also both sold in their varying form factors, but for this project the light strip was the most appropriate.

LED Comparison Table

	DotStar	NeoPixel
Pixel Type:	Epistar 5050	SK6812
Strip Width:	14mm	12.5mm
Pins:	4 (VCC,GND,Data,CI)	3 (VCC, GND, Data)
Operating Voltage:	5V DC	5V DC
Operating Temperature:	-40 °C to 40 °C	-40 °C to 85 °C
LED Brightness (Red,Green,Blue):	(600,500,150) mcd	(700-1000, 1500-2200,700-1000) mcd
Data Transmission Speed:	1 MHz (30MHz max)	800 KHz
Frequency of PWM:	1.2 KHz	1.2 KHz

Table 3.2.5.2.1 - Comparison chart between Dotstar Vs. Neopixel (Specifications provided by Adafruit)

3.2.6 Smart Home Integration Packages

Smart home integration is starting to flood the market. Multiple devices can be managed by one or many source applications. Many stores have an entire section dedicated to these appliances. These appliances included cameras, temperature gauges, motion sensors, etc. There are not many blind applications available and the ones that are only function one feature. The user can choose to have tilt or lift but not both. Majority of the motorized designs are simple smart roller shades which are not too complex. This product is integrating multiple features in one with smart home control. This control includes fine-tuned app control and voice activated features. There are two readily available home integration hubs that provide open source resources to the public. The Alexa echo and the google home dots/minis provide voice control to a plethora of WIFI connected household devices. Both devices have competitive pricing so picking one over cost is ruled out. The choice boils down to the flexibility and ease of each smart home hub. The Alexa device has far more user made customizable packages available than google. For many diy projects presented on the web, Alexa seems to be the device of choice for the majority. Both devices have competitive voice recognition so no clear winner in that department. Both are easily obtainable by virtual anyone. In order to keep the libraries, streamline since the project itself has plenty of taxing overhead, only one device package was selected. It is possible to set up the device to communicate with both hubs, but the scope will grow too large. Both hubs use completely different package

libraries. In this case, just for having more user-friendly resources, Alexa was the hub of choice for integration.

3.2.7 Environment Sensors

Florida is known as the sunshine state, and that statement identifies a fundamental problem that Florida residents must deal with. Heat and humidity is a major factor in individuals daily lives. The project will incorporate the ability to use additional low-cost sensors to help monitor the temperature, humidity, and amount of light. These sensors will also help with conserving energy efficiency within the home. The goal is to have modes that will adjust the blinds in order to block light at certain times of the day when the sun has direct rays entering the home. This will help with maintaining a comfortable temperature within the room and the overall home. The design calls for one sensor for temperature and two for light sensing. One of the light sensors were for outdoor to focus on outdoor brightness and the other for indoor brightness. The comparison of the two light sensors was used to determine the amount of light affecting the room. These sensors in unison will help the customer to have increased control over their home comfort.

3.2.7.1 Light Dependent Resistor

The LDR is the most cost effect and easily obtainable sensor for light detection. There are multiple chemistry make ups for resistors that affect its wavelength detections. The cadmium sulfide chemistry is constructed for sensing visible light by increasing resistance in its presence. The sensors have a lower reaction time than other types, but in this case will not inhibit performance. The LDR is commonly used in many home appliances that rely on light sensing. The sensor is a very dependable choice that is not complicated to operate.

3.2.7.2 DS18B20 Temperature Sensor

This temperature sensor is low cost, simple, and provides snappy performance. Each device contains a unique id and can be connected along the same data line, which is a very useful feature for measuring area coverage. In this case area coverage is not necessary, but multiple devices can be used to account for possible failure or to gather an average. The average of multiple sensors can be used to account for any variance of the device's calibration. This will give the customer a more accurate reading they can use as a reference with other devices in home. The convenient feature of this sensor is that it uses only one data line and is low powered.

3.2.7.3 DHT22 Temperature Sensor

This sensor is slightly more expensive than the other, but it also includes humidity sensor as well. The humidity sensor was an integral feature for a Florida resident. The amount of home humidity affects the AC temperature indoors. Florida is also known to be extremely humid and using a sensor that can monitor temp plus humidity can help. A home that gets over the heat threshold of 70 degrees Fahrenheit for too long can be inviting to black mold. The humidity levels provide their needed moisture. Having a humidity sensor can also help measure a homeowner's dehumidifier performance. This will all the customer more convenience to regulate comfort. The readings of temperature and humidity can be displayed on the application. This will allow the customer to monitor multiple factors that contribute to overall home comfort and efficiency. The chart below will provide technical details between the two temperature sensors. The DHT22 was a more fitting choice for the entire vision of this design. Many of the factors are similar between both devices. In Table 3.2.7.3.T1 a deeper look into either sensors specifications.

Temperature Sensor Comparison Table

Feature	DHT22	DS18B20
Price	2-6\$	1-3\$
Range	Hum 0-100%RH; Temp -40C to +80C	Temp -55C to +125C
Accuracy	Hum +-2%RH(Max +-5%RH); Temp <+-0.5C	+ - 0.5C from -10C to +85C
Resolution / Sensitivity	Hum 0.1%RH; Temp 0.1C	resolution is programmable from 9 to 12 bits
Builtin Alarm	No	Yes
Technology	1 channel Temp and Hum	1-Wire Network
Size	14*18*5.5mm	much smaller !
Waterproof	NO	YES (probe version)

Sensing period	~2s	<1s
Connection	3 wires	3 / 2 wires
DC Voltage	3.3 - 6V	3.0 - 5.5V
Standby consumption	50 uA	750 nA
Measuring consumption	1.5 mA	1.5 mA
Protection case / enclosure	YES	NO
Mounting spot	YES, screw	NO
Interchangeability	YES, fully	NO, unique address

Table 3.2.7.3.1 - Comparison chart for temperature sensors provided by Geekstips

3.2.8 Voice Control Research

The following section deals with all preliminary research performed in the act of learning about the possibilities and requirements regarding voice-controlled devices. This section takes a look at smart-home appliances, such as the Amazon Echo, or the Google Home, and a rudimentary comparison between these two devices, and how they can be used for voice control, as well as existing documentation and resources.

3.2.8.1 Amazon Echo

One of the most utility-based and Internet-of-Things friendly features of the SunShade project is its ability to be controlled by smart home devices, such as the Amazon Echo, also known as 'Alexa', and the Google Home. These devices are voice-activated smart-home appliances capable of linking to user accounts, connect to mobile devices, play music, make calls, set music alarms and timers, ask questions, control smart home devices, and adapt to user-created skills. This feature will be the gateway for the SunShade project team to allow for voice-activated control for the user. The intention is to allow for specific commands, such as:

“Alexa, shades down.”

”Alexa, shades up.”

“Alexa, close shades.”

“Alexa, open shades.”

Etc.

Fortunately, for the common DIY engineer and builder, Amazon has created a user-friendly system for independent builders to control their devices using the Echo product. Alexa Voice Service, typically displayed as ‘AVS’, is Amazon's very own built-in base of services built around its voice-controlled AI home assistant. Basically, the SunShade project team will construct the hardware and software, and AVS provides the automatic speech recognition software required, along with natural language understanding, text-to-speech engines, APIs, hardware kits, software tools and all necessary documentation for such a design.

Amazon also has a particularly useful feature for these designs, called Skill creation, where builders can program specific voice commands, such as those listed above, and select the desired actions, reactions, and so forth for their device. These two features in conjunction with one another will effectively create the exact kind of utility that is desired for the SunShade finished product.

3.2.8.2 Google Home

The Google Home smart-home appliance functions in an almost identical way as the Amazon Echo and Dot products, in that it allows for an Internet-of-Things connection between home devices, so that they may be controlled hands-free via voice recognition. Unfortunately, the community-based software support, as provided by Amazon, is not quite as large for Google products, and so the Echo remains superior in this regard. It is because of this that the SunShade team will be utilizing an Amazon Echo for all voice-controlled features.

3.3 Possible Architectures and Related Diagrams

The following section will cover an overview of architectures and diagrams. Each section will pertain to a system architecture that is involved in this project. Then the compiled information will be used to compare and contrast the choices made. The overview of these architectures will provide a more involved understanding for pcb layout designs.

3.3.1 Microcontroller Units Architecture

In order to manage all the features this project includes; the inclusion of a microcontroller will be necessary to drive them. This section will explore three different

computer architectures, and then compare and contrast each one. The conclusion will be the appropriate architecture for this project's goals.

3.3.1.1 Von Neumann Architecture

The Von Neumann architecture is a popular general computing architecture that is used today. The concept was created in 1945 by the John Von Neuman. This architecture consists of a central processing unit and memory unit. The input and output devices are received externally from other devices. In the central processing unit, there is a control unit and arithmetic/logic unit. The control unit has an instruction register and program counter within. The memory unit stores instructions and data. The architecture uses external mass storage. One drawback of the Von Neumann architecture is that data operations and instruction fetch's share a common bus. That fact concludes that these operations cannot perform at the same time. This fallacy causes a commonly known Von Neumann bottleneck that will limit a systems performance.

This architecture has a simple set up, but it does not allow for modern pipelining. The facts that this structure cannot pipeline, and bottlenecks will impede total performance of modern devices. The project uses many sensors that can easily bottleneck the system on that alone. The only method that helps alleviate some slowdown is to add multiple cache levels. This allows the unit to store and fetch data with more efficiency, as well as, increasing the bus lines to memory. Figure 3.3.1.1.P1 will illustrate a Von Neuman architecture.

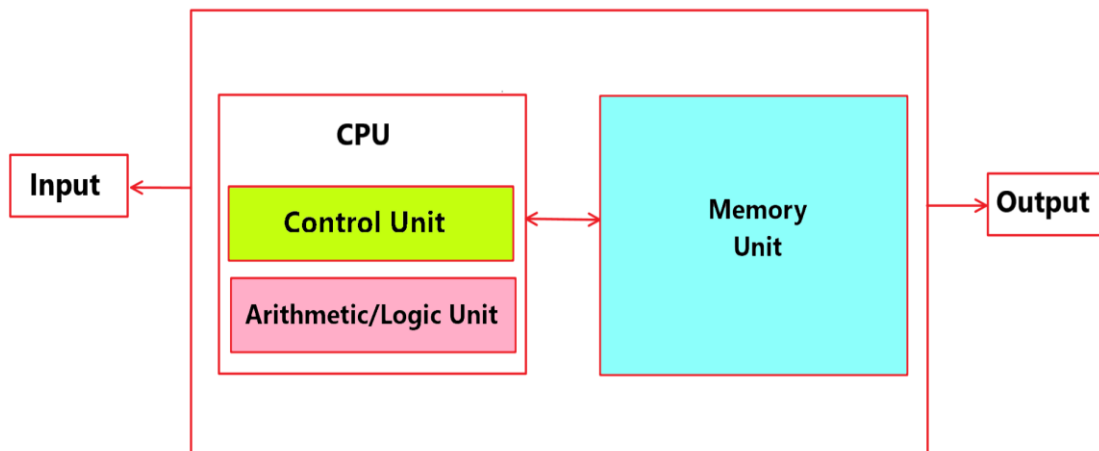


Figure [3.3.1.1.P1]: Von Neumann Architecture

A possible microcontroller choice that uses this architecture is the MSP430g2 from Texas Instruments. This microcontroller specifications will be in Table 3.3.1.T1. The microcontroller is low voltage, has decent flash memory, and 24 GPIO's. The MSP430 uses 16-bit RISC architecture and supports 10-bit analog to digital conversion. The chip supports a variety of communication protocols including UART, IrDA, and SPI. This product has plenty of processing power, memory, and its low voltage. These qualities make it a competitor, but the only drawback might be the GPIO. There are

multiple pins dedicated to a specific task which will limit the number of pins available for all accessories.

3.3.1.2 Harvard Architecture

The Harvard Architecture uses the control unit as its central management device. The instruction memory and data memory are separated unlike the Von Neumann counterpart. This opens up the ability to pipeline instructions which boosts throughput performance in this design. This also allows the device to use parallelism by processing instruction and data memory simultaneously. Figure 3.3.1.2.P2 will provide a diagram of the Harvard architecture. Today most processors use a modified version of the Harvard Architecture. In this architecture they adopt leveled cache and lean less negatively upon the separation of instruction and data within the CPU. There is still multiple buses within this implementation difference. They adopted the ability to consume instruction memory and write them as data memory. Programs can access the memory directly from instructional set instead of wasting a cycle writing it to data to use it. These features increase throughput performance of the unit.

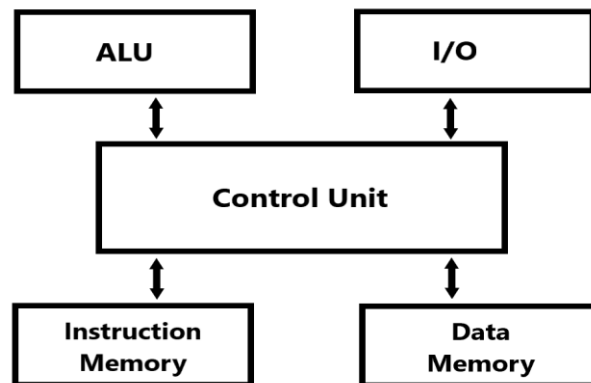


Figure [3.3.1.2.P2]: Harvard Architecture

The original Arduino ATmega328 microprocessor was designed with a Modified Harvard architecture. The new revision uses the ARM architecture which will be discussed in the next sections. This processor is another through hole mount type similar to the Texas Instrument MSP430. It is manufactured by Atmel, which is also referred to as Microchip Technology Inc. Table 3.3.1.T1 will provide specifications for the ATmega328p model that is modified Harvard Architecture based. This processor has similar specs to the MSP430, but it provides a greater clock speed, program memory, and RAM. The chip is commonly coupled with the well renowned Arduino UNO development board. These boards provide ease when prototyping small projects.

Another suitable Modified Harvard architecture in contest, is the ATmega2560 microprocessor by Atmel. This microprocessor is commonly coupled with the Arduino Mega 2560 development board. The clock speed is variable depending on voltage input, and it can be 8 MHz or 16MHz. This allows for the option to run the device at

ultra-low or low power consumption. There is more flash memory, SRAM, and GPIOs than the competing counterparts. The I/O pin count is quite overboard at 54 I/O pins, but this leaves us with much needed flexibility for the number of components. The specifications for the Atmega2560 will be in Table 3.3.1.T1. This microcontroller will be more difficult to mount because it will need to be soldered to the PCB. There are also 100 pins for this microcontroller, so the wiring cannot be mishandled to prevent faults. The major plus to this microcontroller is there are 15 PWM dedicated pins, and many of our components require a dedicated PWM pin.

3.3.1.3 ARM Architecture

ARM has remained the industry's leading chip manufacturer for embedded systems for a long time. They are starting to make their way into the desktop market which has been stronghold by Intel's x86 architecture. ARM chips can be found in a plethora of devices such as microwaves, cell phones, portable gaming consoles, etc.... This shows the variety of different applications their cores can be used for. ARM is low cost, offers performance, reliability, and power. Their architecture is based off an advanced RISC architecture, and this architecture reduces the number of transistors needed for performance. They split their cores into three categories which are the A-Profile, R-Profile, and M-Profile. According to the manufacturer's website, the A-Profile is for rich applications and high-performance operating systems [14]. The R-Profile is for real-time and high-performance applications [14]. The M-Profile is for discrete processing and microcontrollers [14]. All these ARM processors are based on systems on a chip. This means that the core components such as control unit, system clock, memory, interrupt controller, reset controller, and other supporting components are on the same chip. They allow consumers to create custom configurations once the license is purchased. The system on a chip provides the ability to operate parallel programming.

The most ideal profile to look at would be the M-Profile because it fits closely to the needed requirements. The Microchip's SAM3X8E is based off the ARM® Cortex®-M3 processor, and it is integrated with the Arduino Due development board. Like all previous models mentioned, they shared the prototyping standard. Specifications of this microcontroller will be in Table 3.3.1.T1. This microcontroller has the fastest clock speed and incorporates a DC to AC converter. No other controller in the lineup has that capability, which makes this model stand out for utility. Unfortunately, the project has no use for that feature so it would go to waste. It includes a 12-bit DAC that will give increased accuracy during conversation. The chip has low power consumption that will be ideal. The chip has too many GPIO to manage supporting 103. This will make wiring more difficult for designing the PCB because sorting through datasheet will be more taxing.

Microcontroller Specification Comparison Table

Processor:	MSP430g2553	ATmega328p	ATmega2560	SAM3X8E
Mounting Type:	Through Hole	Through Hole	Surface Mount	Surface Mount
Operating Voltage:	1.8V - 3.6V	1.8V - 5.5V	1.8V - 5.5V	1.62V - 3.6V
Clock Speed:	16-MHz	20MHz	16-MHz	84 MHz
Flash Memory:	16KB	32KB	256KB	512KB
SRAM:	512 bytes	2KB	8KB	96KB
Analog to Digital Converter:	10-bit 8 Channel	10-bit 6 Channel	10-bit 16 Channel	12-bit 16 channel
Digital to Analog Converter:	none	none	none	12-bit 2 Channel
GPIO:	24	23	54	103
Architecture:	Von Neumann	Modified Harvard	Modified Harvard	ARM Cortex-M3
Manufacturer:	Texas Instrument	Atmel	Atmel	Microchip

Table [3.3.1.T1]: MSP430G2553 MCU, ATmega328p, ATmega2560, and SAM3X8E Specifications

At this stage there has been no definite choice for microcontroller. The top two front runners are the ARM SAM3X8E and the Atmel ATmega2560 controllers. These chips will provide plenty of I/O and performance power for this project. Unfortunately, the other two microcontrollers will not be a reliable choice because we need 7 PWM capable lines. Those units offer 6 or less PWM capable pins on the unit. Once everything is fleshed out a procured the final decision will be made.

3.3.2 Motors Architecture

The motors are very important in our project, without the motors nothing will happen, and the user will not be able to lift or tilt the blinds. the Motors will be connected to the microcontroller outputs and will give the exact signal for each motor whether the user wants to lift or lower the blinds, or the user wants to tilt the blinds. We are going to

have a DC motor connected to the tilt rod and the other motor will be connected to the cord for the lift/lower mechanism.

Since we are not going to have too many components on the PCB, we may add the ports to connect the microcontroller to the motors to add some efficiency in the design. we are going to use a battery controller to charge the batteries, but they are not going to be on the PCB.

Motors play an absolutely crucial role in any type of electromechanical system. Motors are one of the most commonly used loads in the industry, commercial sector, and households. Motors are the machines which convert electrical energy into mechanical rotation. Using this rotation, a large number of systems and mechanisms can be constructed.

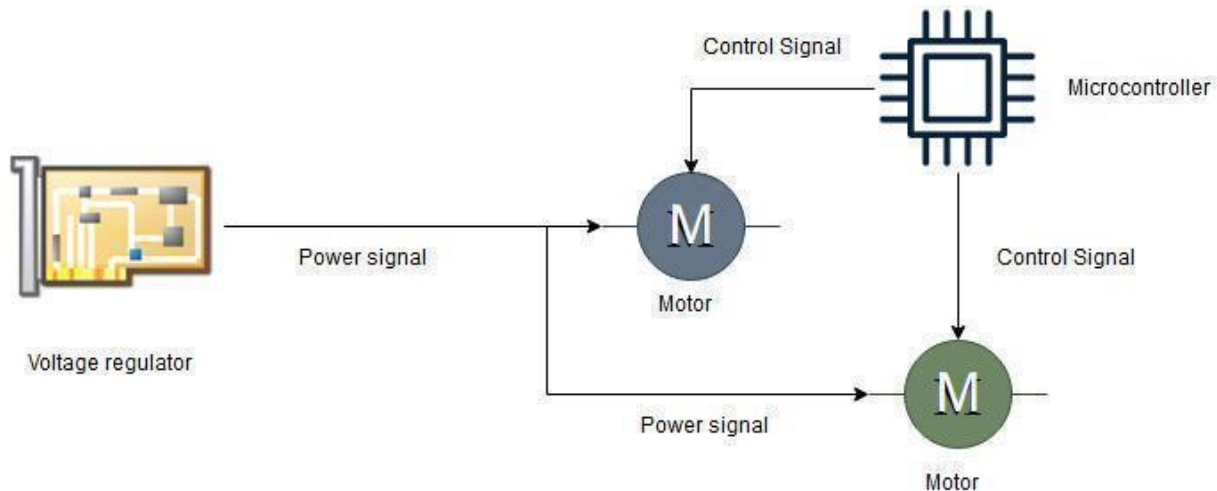
There are basically two types of motors which are AC motors and DC motors. AC motors run on AC power while DC motors require a DC source for their operation. There are a large number of sub-categories of both AC and DC motors such as induction motors, synchronous motors, servo motors, brushless DC motors, and stepper motors. These different types of motors are used in different types of applications.

Since, we are going to use microcontroller based to control the motors in our project, therefore, DC motors suit our needs more appropriately. These motors are simple, smaller, and can be powered by a battery. In addition to that, the speed control of DC motors is way easier than the AC motors. There are special types of DC motors also available which have built in feedback control circuitry and provide very accurate positioning and angular motion. These motors are known as servo motors.

In our project, we are going to use the motors for the movement of blinds. Hence, simple DC motors with microcontroller-based control should be able to meet our requirements. The microcontroller will issue the commands regarding the lifting or lowering of the blinds and the motors will move accordingly to execute those control

commands. We are going to have a DC motor connected to the tilt rod and the other motor will be connected to the cord for the lift/lower mechanism.

Since we are not going to have too many components on the PCB, we may add the ports to connect the microcontroller to the motors to add some efficiency in the design. we are going to use a battery controller to charge the batteries, but they are not going to be on the PCB.



Block Diagram 3.3.3.1 : Distribution of power

3.3.3 Power Distribution Architecture

For the working and operation of any electronic or electrical circuit, power is the most important requirement. Without the electrical power, the circuit or the device won't even be able to power on. Hence, the power distribution system is one of the most crucial components of any engineering system.

The primary source of power in our project is the solar energy. Solar energy is clean, efficient, and reliable. For the conversion of solar energy into electrical energy we are utilizing solar cells. The solar cells are connected directly to a storage battery. The solar power generation is effective only during the day. For the night usage and for cloudy days, we need storage battery for powering up the system. A charge controller device is installed between the solar cells and the battery. The charge controller takes power input from the solar cell and provides it to the battery in an appropriate format. The charge controller protects the battery from overcharging and from over-voltage. The keeps the battery capacity intact and increases its life span.

The electronic components such as motors, ICs and microcontrollers usually require a 5Vdc voltage. Hence, a power supply PCB circuit is present in the system which is used for providing the necessary power to the motors, microcontroller and other peripherals. The motors can also be powered directly from the microcontroller unit; however, we need to provide a provision for powering them up using the solar cell in case of malfunctions or other issues.

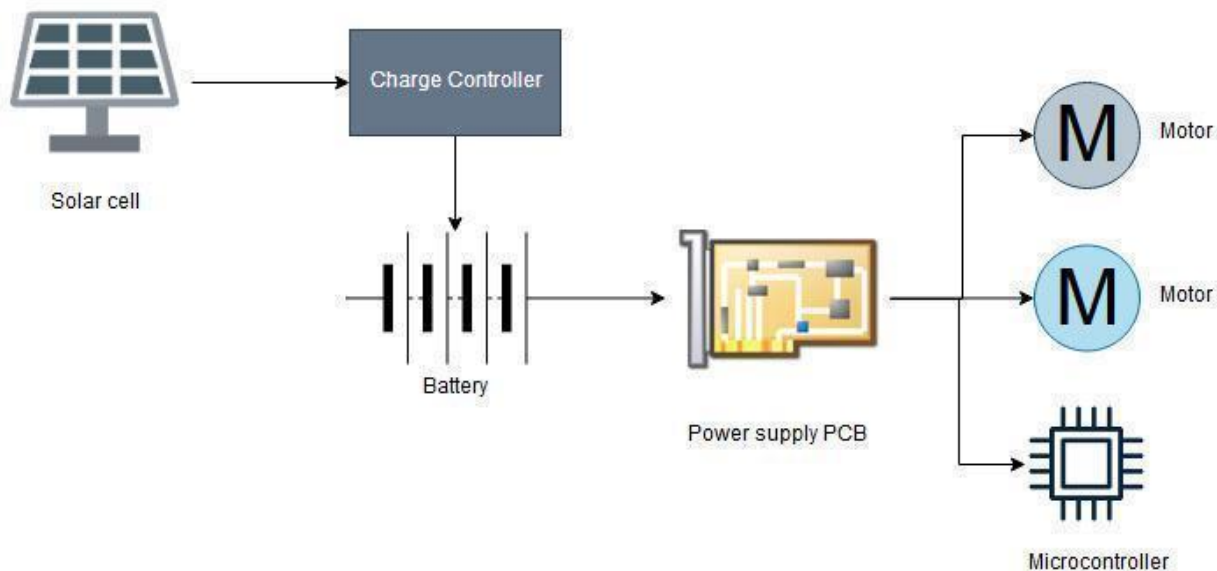
Power distribution is an important step in the building process of our project, and basically it depicts how the power is distributed among the important components of the project. The solar cells are the source of the power and they have a direct connection with the batteries, we are going to have a charge module that is needed to charge the batteries and is connected between the solar cells and batteries, we are going to use the motors for the movement of blinds. Hence, simple DC motors with microcontroller-based control should be able to meet our requirements. The microcontroller will issue

the commands regarding the lifting or lowering of the blinds and the motors will move accordingly to execute those control commands. We are going to have a DC motor connected to the tilt rod and the other motor will be connected to the cord for the lift/lower mechanism.

Since we are not going to have too many components on the PCB, we may add the ports to connect the microcontroller to the motors to add some efficiency in the design. we are going to use a battery controller to charge the batteries, but they are not going to be on the PCB.

Also we are going to have the PCB to power the remaining components which are the motors and the microcontrollers , of course we can always change the design, for example, we can have the microcontrollers to supply the necessary power to the motors, and that is because we are going to connect the motors to the controller unit , the later components can provide power as well , but in any case we should power the motors from the solar cells and or the batteries because we may face a problem of connectivity or malfunctioning of the microcontroller and we then face bigger problems. Solar cells will charge the batteries during the day and therefore we could manage to have the rest of the components and especially the load to draw the minimum voltage and therefore the least power output possible.

The layout of the power distribution is shown below:



3.3.3.1 Power Distribution system

3.3.4.1 Solar cell module

There are many types of solar cells that we can choose from, they are the main components for supplying the necessary power to the motors and the other components. the cells are made of semi conductive materials such as silicon that absorbs light from the sun. Silicon is very shiny in nature so in order to take advantage of the photons and

prevent them from reflecting away from the cells, an anti-reflective coating is applied to the cells, and that make really look for a good product that will not disappoint us later.

One choice is to use Monocrystalline cells which have a symmetrical appearance that indicates the purity of silicon crystals. Monoblastic cells are silicon alloy that have been sliced into strips. we can notice that cells characters are not aligned which gives the monochrome panels their distinct appearance , those type of plates are very expensive because they have a really high efficiency rate that can go up to 22.5%, yet the problem with these type of solar cells is their lifetime, it is noticeable that they have the lowest rate of lifetime. the monocrystalline cells can be connected either in parallel or in series, however it is desired to increase the power supplied to the system as a whole.

Another choice we could use and is a little cheaper than the previous one, is the polycrystalline. The difference between them and the monocrystalline cells is the shape, where the polycrystalline cells have a rectangular shape and they are definitely lower in prices compared with the monocrystalline cells, however, of course, the lower the price is the less quality. the polycrystalline have less efficiency (16.9%), they have an average of 25 years lifetime.



Figure 3.3.4.1.1 - Polycrystalline Solar Cells

Another type of solar cells that seems to be interesting is the Amorphous silicon cells, they type of cells is more effective and the solar cells are composed of silicon and the crystal configuration is cracked with a hydrogen element or other elements deliberately introduced to gain the characteristic properties of electrolytes and silicon Amorphous cells. we can attach these type of cells to our blinds because they are flexible , they are less efficient than the rest of the solar cells that we have discussed, however , I believe that they can do a good job of providing enough charge to the batteries and to the other parts of the smart blinds parts, they also have a reasonable lifetime.

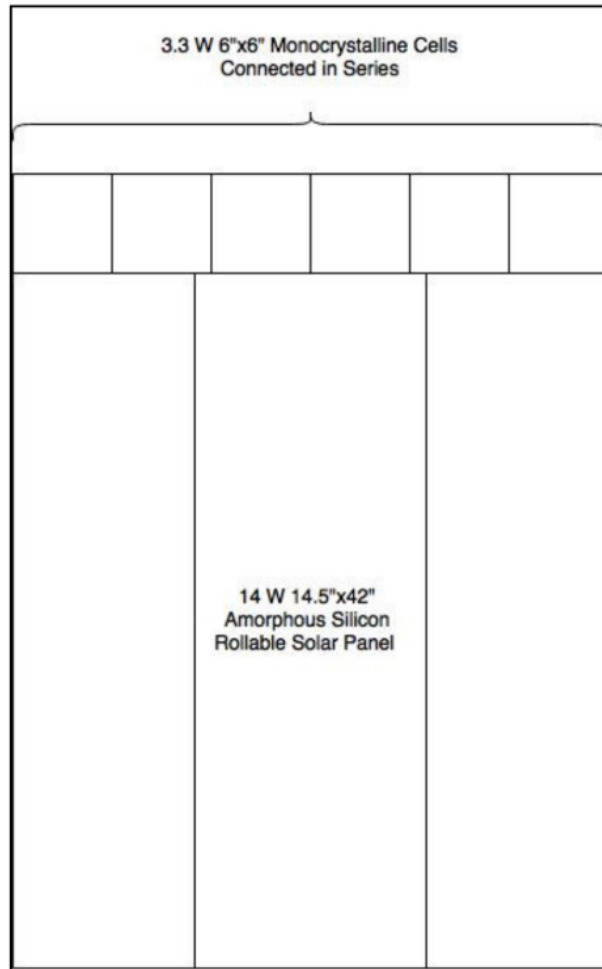


Figure 3.3.4.1.3 - Amorphous Silicon

The Thin Films are also another type of solar cells that could be used in our project , they are cheap and their prices can be found to be \$16.99, they are the least expensive solar module that we can find , after reading some articles I realized that crystalline is much more advanced than thin film technology, beside other factors to be considered, the efficiency is the most important one to have in mind and thin films have a very low efficiency rate that can be 4% to 12%.

We are not going to use the thin films, even though they are much cheaper than the rest but still we have to have equipment that have a decent lifetime and efficient , however the price can be a challenge but regardless we don't want to have a failed project or a project that is disliked by our clients.

After doing some research, we realized that is a good deal if we buy 40 solar cells (3x6 each cell), and each cell provides 1.8 W , the average current is 3.6 A. One thing we need to do is to connect the panels to each other, we probably need to use about 12 cells per panel. The panels need to be hooked in series, we connect the positives of each panel we the negative of the connecting panel, we use a solder to connect a panel with another.

3.4 Power outputs

In the previous sections we have explained the power distribution system of our project in detail. The power system of our project is based on solar power. The solar cells convert sunlight energy into electrical energy. This electrical energy is then stored in the battery by means of a charge controller which limits the current flow and protects the battery. This system describes the input of our power system in detail. However, to complete the discussion on the power system, we should also include the power output of our system. This means that we should also discuss the load characteristics of our system.

The output of our power system is mainly used to power the microcontroller and the associated electronic peripherals. The battery does not provide the required level of DC voltage to power the sensitive microcontroller and electronic peripherals and therefore, we need a DC-DC step down converter or regulator to provide a suitable voltage to our microcontroller. Similar to the microcontroller, the DC motors can also not run directly from the battery level voltage and hence we need another voltage regulator to step down the DC voltage so that it can be utilized by the motors.

We do not have any loads in our system that run on AC power. This is a plus point and reduces the project complexity and cost. In case of AC loads, an inverter needs to be added to the system which is an expensive solution and increases the project cost.

4.0 Related Standards

This will contain standards required by IEEE organization. The following sections will also contain other electrical standards.

4.1 Electrical Standards

Electrical standards are the standards that must be upheld for an electrical system. They include the IEEE standards, National Electrical Code NEC, and OSHA codes.

4.1.1 IEEE Standards

It is crucial to follow standards in our project, because the project can be a product that one day it's sold to customers around the country or may be in other parts of the world. However, following standards and some rules may prevent us from being sued and these standards are a protection from any =thing that can happen against us. the first thing people look into, if an accident happens, is that, did we follow the standards or did we neglected any important criteria that must be included in the design.

the first thing a designer should look at is the IEEE, because of its magnitude and importance.

The IEEE is one of the important organizations that we should follow their standards according to their rules regarding our project. Through the contribution of its

members, the IEEE represent the primary reference for many technical topics, ranging from computer engineering to electrical engineering, as well as standards in a variety of sectors, for example: power, energy and robotics and many others. In our project, we have to focus on a few regulations:

- IEEE829: software test Documentation.
- IEEE 830: Software requirement specifications.
- IEEE488: Standard Digital Interface for Programmable Instruments.

IEEE Code	Standards
IEEE 829	Software test Documentation
IEEE 830	Software requirement specifications.
IEEE488	Standard Digital Interface for Programmable Instruments
IEEE 802.15.2	Bluetooth and WIFI coexistence mechanism
IEEE 830	Software requirement specification.

Table 4.1.1.1 - Important Standards

4.1.2 Other Electrical standards

It is important also to follow the standards of the National Electrical Code NEC, they standardize the rule on how to safely make all the electrical wiring safe and secure. in our project, we will have some electrical wiring, the good news is, we will not have much of it, and it will be safe and secured. we believe the less wiring the safer our project will be, and that is a very important goal in our project, that is safety. we would like to have a reputation and therefore client needs to know that our product is safe and follows the guidelines of every organization out there.

Other Electrical standards to be considered are the OSHA (Occupational safety and health administration). The standards will have a great impact on the safety of our project and the safety of our clients as well. some standards are definitely important to make sure the safety of the electric components, for example the NFPA 70, which regulate any hazardous and dangerous electronic components.

Some of the safety problems is an electric shock when a user touches the equipment, for example to change the batteries or to change a module, therefore the UL50 is introduced and considered when creating the box.

The Bluetooth and WIFI standards are important when selecting which connectivity is appropriate and the RFID standard is important for the security of the project as a whole.

4.2 Communication standards

Communication is a very important aspect in our project because, without it we can't call it a smart blind. We have to make sure that:

- Our project uses WIFI integrated in the microcontroller and therefore we would have to consider the communication standards, Ex. IEEE 802.11b that is very specific to wireless technologies and has a maximum data rate of 11 Mbit/s.
- Communication allows smooth interconnection between parts.
- Not expensive and hard to obtain.

Communication can be a problem in case there is a power outage and user cannot connect to the WIFI or to the Amazon Alexa if we add that feature, we will have to consider and explore all the options and make sure we have emergency options at hand and available for us.

Standards	Description
IEEE802.11b	Regulates the wireless networking speed and specifications up to 11Mbits.
IEEE802.15.1	Regulate the wireless medium access control for the area networks.
IEEE 802.15.4f	Active RFID and sensor applications.

Table 4.2.1 - Communication Standards

4.3 Safety standards

We don't want to see anyone get hurt in our project design nor customers, safety is a primary component of our project. The main goal of this project is to make sure customers are safe as well, beside us. Therefore we noticed that using blinds with cords can cause deaths especially among toddlers playing with the blinds, parents usually like to set the cribs adjacent to the windows, in that case, toddlers like to stand and play around, and once they can grab the cord, that may turn deadly. Our job is to eliminate this problem by introducing a cordless blind that can be controlled with just a push of a button on the app.

Fortunately, our design does not contain any hazardous components nor any dangerous materials. Nothing is expected to occur. We will be using a Metal DC Geared Motor 6V 15RPM that we are familiar with no known problems.

Fires and Explosions may occur or even may be a little as damage to the PCB can happen if we don't take seriously a couple of actions in the process of building the project.

Some statistics indicate that the causes of accidents caused by the use of electricity are for example: overloading, short circuits, using wire that are not suitable for the value of the current going through (current passing through wires may increase the temperature to the point can cause a fire).

Some important ideas of safety in our project:

- Double check and test components connectivity during project
- Ensure the correct use of the equipment and materials and importantly the procedures of safety.
- Follow the proper procedures in order to cut power and check electrical components before maintenance work.
- Handle all electrical appliances as active (should be a permanent caution) because the user may not be aware of the existence of any existent dangerous malfunction in then system.
- Do not use damaged or misuse of electrical equipment.
- Do not connect devices and equipment to the ground

Standards	Explanation
NESC	It is an IEEE code for safety of electric supplies and communication utilities.
UL 50	Protection from incident provided to workers
NFPA 70	Safety in electrical design, installation and inspection

Table 4.3.1 - Standards we should lookout for

4.4 Battery standards

Codes	Standards
IEEE 937	A practice for installation and maintenance of lead acid batteries for some types of sensors, for example the photovoltaic sensors.

IEEE1013	normalizing the size of the lead acid batteries for standalone photovoltaic sensors
IEEE1361	manual talks about charging and selecting, testing and evaluating lead acid batteries
IEEE1526	a practice for testing the stand alone photovoltaic systems

Table 4.4.1 - Battery Standards

5.0 Realistic Design Constraints

The realistic design constraints cover the limits set by real-life obstacles. These obstacles can be attributed by the hardware limitations, economical, or personal limitations. Each team member has their own unique set of personal limitations that will stress the completion of the project. The overall goal is to spell out these constraints, in order to, plan for working around them.

5.1 Power Constraints

The design process of any project involves a number of challenges and constraints. For the successful completion of the project, overcoming these design constraints is of utmost importance. Following are some of the major design constraints which we are faced with for the successful execution of this project.

The main purpose of the blinds is to roll up and down however the user wants to, and therefore this requires having a motor that run efficiently and capable to lift the blinds all the way up if necessary. Blinds weigh less than 12lbs and motor of torque at 6v should be enough. The project needs to meet the requirement of low consumption so it should not impact too much the gains related to heating; therefore, it should not present any danger.

5.2 Economic Constraints

The project we are designing doesn't have a sponsor, therefore we have to buy all the components ourselves. Our design requires components that are available online and easy to find and we should not have any problem acquiring the components. The only problem is the blinds (the core component of the design), it was a little challenging

to find the blinds with cords available. we went to a couple stores (Home Depot, Walmart, Lowe's) and unfortunately, we couldn't find them because the ones they have are without cords. Having the cord in the blinds is necessary. we finally found one seller online offers the blinds for a reasonable price.

Another economic constraints, is that we wanted to add a control module(PD control) in the project, to make sure the blinds operate in an efficient and sophisticated manner, the prices for the controller are expensive and since we don't have the sponsorship, we decided not to continue with the PD controller, and actually if we purchased them, definitely the cost of the blinds will go up and it becomes harder to implement in the design.

Another economic constraint is the Alexa, because it is a little expensive and it is a feature that we would like to add in our project. Economic constraints are definitely an issue when it comes to expand our project and make it look interesting, having a sponsorship would have helped tremendously by adding features that we could not afford to buy. It is important to know that the total cost of our project will not exceed two hundred dollars, including blinds and the frame even though we included a cheap Alexa that price may be the same, because the cost of the electronic parts are very cheap, for example we can buy the Microcontroller for five dollars if we want to, other parts can be found for a few cents and others are found for few pennies. so basically, our main problem is that, adding some interesting features will make us spend more money.

5.3 Time Constraints

Time is a very important aspect in our project, we only had one semester to realize our project before we present it. first, we had to buy all the components , and that is not an easy task because there are so many different types for each component and we had to carefully choose the right one, not only it takes time for a component to be shipped but we had to be prepared in case that component is malfunctioning or damaged. Another important step that definitely consumed time is building the PCB, the schematic does not take much time however the printing phase is definitely time consuming, we were also worried if the PCB is malfunctioning and we have to get it redone by the manufacturer multiple times, we were expecting a couple of weeks for it to be redone. time will affect as greatly if we want to add any feature like the such as a roll-down secondary transparent window shade system and a home security system. Time also will affect our ability to conduct the software development for example we would like to add a temperature sensor in our design.

Having only four months to build the project and have it prototyped, means that we have to work fast and that may lead us to forget things that are necessary in our design, they may be small parts needed that are essential in our design.

These are the disadvantages of having such a limited time to do everything, some of us may have an emergency to leave and the workload would be divided on less people with such a time frame, it can make our progress at risk. we are prepared for the worst-case scenario but still some circumstances may put our project at peril.

Time is valuable and should be well managed, however we will not be able to learn more about other ideas and other methods. we only focus on what is necessary and prioritize our work according to what is important, unfortunately we will not be able to expand our knowledge and learn other ideas. We hope if we have the time to add at least a feature or two, we are going to keep that in the plan, otherwise we have to complete what is important and necessary first before we can even move to other things.

The summer semester is short, not only short but the time runs very fast and writing the paper requires research and it's time consuming. we have to be very careful and we have to organize our time appropriately.

5.4 Weight Constraints

Finding the right blinds is a little bit challenging, because our plan is to use the blinds with the cord attached to it (to lift the blinds and to lower them as necessary), the string in our project is important because we have to connect it with the servo motor and the rod. However, we had to do a lot of research to see who sells the blinds with the cord. As we were searching, we realized that for safety concerns, it's absolutely unsafe to buy the blinds with cords because they are not safe for toddlers. usually parents set the cribs by the window blinds with cords and some toddlers they strangle themselves with the cords.

The less heavy blinds, the quicker response when using the app to lift the blinds and lower them. we don't want to use heavy blinds also because of power usage, which means the heavier the blinds the more power consumption and therefore a frequent change of batteries in case solar panels are not used in the project.

Other blinds we were going to use was the vinyl blinds, but they were expensive, the only good thing about these types is that they are much lighter in weight. In our project we would like to use the regular Faux wood blinds as shown below



Figure 5.4.1 - Faux Wood Blinds

These blinds should be fine for our project, we are going to mount two motors: one for the tilting and it will be attached to the hexagonal bar and the other motor is attached to cord to adjust the height. the main concern that we can face is if the motor will be able to lift the blinds as needed by the user, therefore we think that the motor we picked for this particular task will do the job since the blinds weight is approximately 10 pounds for the 36"x48", and the Metal DC Geared Motor 6V 15RPM 10kg.cm used in this project to lift the blinds should be fine, we are not going to attach anything to the bottom of the blinds so no weight added.

Overall in our project, we are not interacting with anything that is heavy. the heaviest item could be the frame and it may approximately weigh about 15 to 15 pounds. overall, the motors and any other electronic component, will be mounted in the frame so we don't see any problem in regard to the weight.

5.5 Ethical, Health, and Safety Constraints

This section covers the required ethical, health, and safety constraints requested by ABET protocols. The team as discussed many of the constraints on conception of the project.

5.5.1 Ethical Constraints

There are a couple of ethical considerations that we have to take in consideration, for example:

- Impact of the design project is basically how the community will perceive the project, how this design will impact the livelihood of the community, that is the social impact of our design, means that maybe not everyone will need the smart solar blinds but our design will at least have a group in the society that will really appreciate our work, our project aims to have an impact on the society at a large scale and we hope that the community will benefit from it.
- Another aspect of the ethical constraint is to be resilient; it means that not only do we want to have a safe and economic design but mainly the design is flexible and robust.
- Smart blind is operable by any person interested, as long as this person knows how to use the app on the phone. It is very basic and straightforward, the App is simple, the user would only have to create an account, set a username and password, that is a common and easy thing anybody can do. After, the user is all set, can tilt, lift upward or downward the blinds by just using the App.
- The smart blinds is easy for maintenance, first of all there should not be any problems expected using the blinds and the control App, the only thing we can suggest is to avoid having kids play with blinds because they may break the blades.

- Beside all the ethical considerations mentioned above, our group doesn't have any ethical issue in going forward with the smart blind project.

5.5.2 Health and safety Constraints

Our project is a small one compared to some other ones we have seen, we expected to be safe during the process of building the smart blinds. safety is a top priority for everyone, we should take precautions and follow all the rules in order to be safe.

Our project must comply with the codes and standards of the organizations mentioned in above sections. we should look out for the risks and make sure we are safe during the installation of our material. Our users will have a peace of mind if we set the proper rules according to the standards mandated by the IEEE and NEC. We also should test our components and make sure they are safe for use, testing is a critical step toward a safe design. we would like to build a project that aims to gain the reputation of our users and not abolish it.

6.0 Hardware Design & Component Testing

This section will be all about discovering the chosen hardware and testing if all acquired components are operational. The hardware design will also include PCB design introductions and device physicality's. Each component was chosen after team deliberation. The team has discussed the requirements targets necessary for each component.

6.1 PCB

The conventional PCB is usually printed on a thin insulating circuit board card which is often made of fiberglass and usual size can go up to 1.6mm thickness. The printed circuit board is one of the important parts for this project because it has most of the components, we have to make sure it's properly designed and assembled and lastly, it must undergo rigorous testing before full implementation.

Printed Circuit Boards is one of the most important components in the Electrical and Electronic industry. the PCB contains several layers that are combined together using heat and adhesive materials, thus producing one piece and that is the PCB.

Usually there are steps to look for in order to have a successful PCB layout:

- Look for CAD package (in our case we use Eagle Software), and there are many low cost options available online to choose from in designing the PCB, however, let's consider these steps:
 1. Look for community support, by finding out people using the same packages, there are more chances to have libraries that are ready for use.

2. Easy to use: if it looks difficult, it's a good idea to use it from the beginning.
 3. Some free software they limit the use of their package, for example the number of layers, the board size, etc.
- Search for other related work, and we should learn and know what they others have accomplished. This is an important step because it will make us ready for our design and make us aware of any risks and hardship.
 - Continuous training and practice.

There are steps taken in order to have a successful PCB layout ready :

- Schematic
- Determine the required footprints
- Components design and placements
- Routing (determine the location of each element)
- Size of PCB
- Connection of the PCB to the power supply
- Single layered board

Upon further research, it can be found that there are multiple types of PCB; Some have laminated layers, at least two or more, they are rarely used in senior design projects because they are expensive and the interior layer is hard to examine if there are problems of connectivity or so.

It is important to make sure the size of the components is appropriate because the schematic does not give the size of the elements built on the board. Other considerations that must be taken is to make sure the heat dissipated is not overwhelming, otherwise we will have elements that will heats up quickly and may damage the printed circuit board. There is no such a unique design for the PCB, there can be multiple representations of the same schematic, nonetheless, the components layout should be made easy for the manufacturer to build, therefore it is important to have the components to face the same direction. Eagle is the software used in this project, we have used this software in Junior Design class and we became familiar with its content, Eagle has two sections: the schematic and the printing board, the later part is a little challenging because we have to connect the element at a certain way were er can't have the lines making angles beside 90 degree. as discussed previously then PCB has to be small and has less layers to minimize the cost, the smaller the PCB, the greater chance to make it fit in our design.

It's a great idea to take advantage of the simulation content offered in Eagle autodesk, there exist a content called ngspice were can test the schematic, for example, make sure all components have the voltage drops necessary.

6.2 Blinds physicality

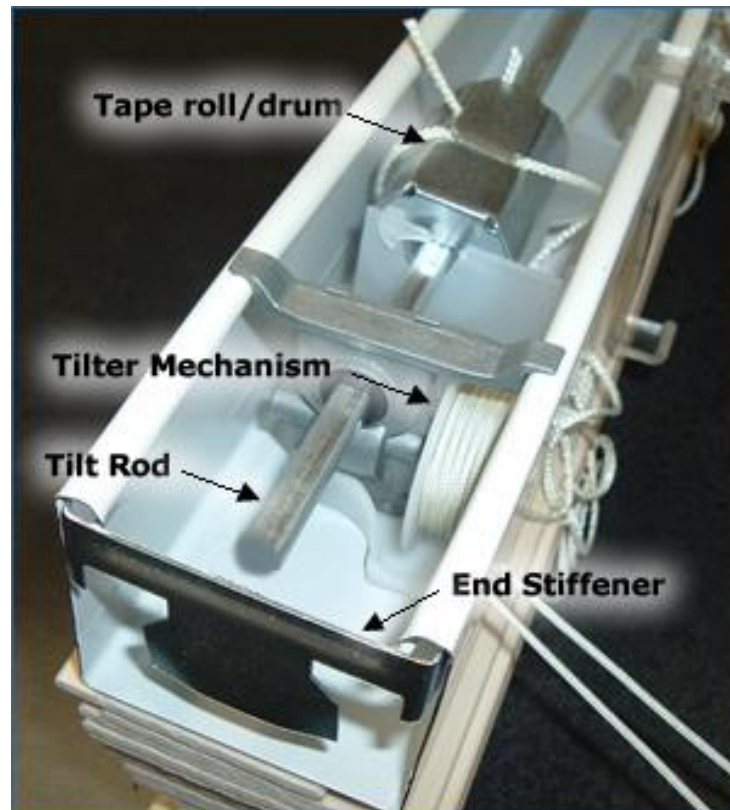


Figure 6.2.1 - Interior view of standard draw-string blinds mechanisms.

Typical window-blinds mechanism function as follows: When the pull-cords are interacted with, or pulled, they rotate the 'tilter mechanism', as shown, which, in turn, rotates a gear to turn the 'tilt rod'. The tilt rod is the spine of the blinds system; When it rotates along the length of the top of the blinds, it pulls and releases the tilt cords in the 'tape roll', or 'drum'. This functionality is what allows the horizontal blinds to tilt open and closed. On the other end of the mechanism, the lift cord is a simple pulley-system, where each tilt cord is accompanied by a drawstring which feeds up into the mechanism for lift.

The primary control for the blinds system will be split into the two mentioned sections: Lift and Tilt. The lift will be performed simplistically, with a servo winding up, and releasing down, the pulley drawstring, replacing the need for a manual pull from the user. The cord will be cut to only allow enough length for full extension, eliminating its outward visibility. This winding servo mechanism will be connected directly to our microcontroller for use via the app, manual, or smart appliance control. The other section of physicality is tilt function. This is done similarly, with the tilt rod being the main point of manipulation. For this, we will integrate another servo, intended for 180-degree use, which will attach to the end of the tilt rod to directly manipulate it clockwise and counterclockwise, which will tilt the horizontal blinds open and closed. Once again, this servo will feed directly into the microcontroller. For precision control, both servos will be equipped with a potentiometer, so that an accurate 'open/close, up/down' value can be established for

varying positions. This will enable a user to make a command such as opening their blinds to only 50% or preventing damage to the system by over-turning. In order to effectively connect the tilt-rod to the motor, a small locking piece will be 3D printed. this piece will be dime-size with a hexagonal fit for the rod.

All physical components including PCB, servos, power supply, etc. will be hidden inside of the covered top plastic bar of the blinds. This will allow for a sleeker, more finished look for our project. If a size adjustment is necessary to properly house the components, a larger, or tailored top cover may be laser-cut from wood or plastic in the UCF engineering Innovation lab. The battery to be charged by solar power will be stored in the housing mentioned below, but the actual solar panels will be mounted to the external, window-facing side of the top housing. The frame designed to mount the SunShade project to, and serve as a stand-in window-frame, will be constructed out of wood, and allow the project the stand free for display and demonstration purposes.

6.3 Power

One of the most challenging tasks in our design is power management. The project contains many sensitive components that have specific requirements. The goal is to have a 12V or 18V battery back powering the device. The battery pack will be charged by solar during the day. A USB interface will be incorporated for flashing the MCU, which will provide an emergency charging port. The design will need to use voltage regulators to step-up or step-down the voltage from the battery. The target voltage from the initial voltage regulator system will be 12V to enable us to run the 12V DC motor. There will be a second voltage regulator at 6V for the servo motors. Then a third voltage regulator that will be used to step-down the voltage to 5V. The second and third regulator will use the 12V regulators power as input. There might be a need for a fourth voltage regulator to power other sensitive components, but the choice to add a fourth will be after the target components are sufficiently adequate. The following sections will describe the distribution components in detail:

6.3.1 Battery Source to 12V Voltage Regulation

The battery source will provide either 12V or 18V in the final design. The battery pack will be made using rechargeable 3.7V lithium flat cell batteries often used in industrial or hobbyist equipment. This battery pack will provide us with the main DC voltage source for all components. The 12V DC motor for the roller shade is the reason the power source target is set to 12V. With any battery pack the fully charged voltage can be higher than specified, and the battery will have a lower voltage as it nears depletion. The goal is to maintain 12V power for the majority of the batteries lifetime. To reach this goal a boost-buck voltage regulator will be needed. Figure 6.3.1.1 will contain a boost-buck schematic. Using a target frequency that will be set and monitored through a pulse width modulation pin on the MCU, the boost-buck converter will output a constant fixed voltage of 12V. The voltage will be boost when it drops below 12V, and the voltage will be stepped down when voltage is above threshold.

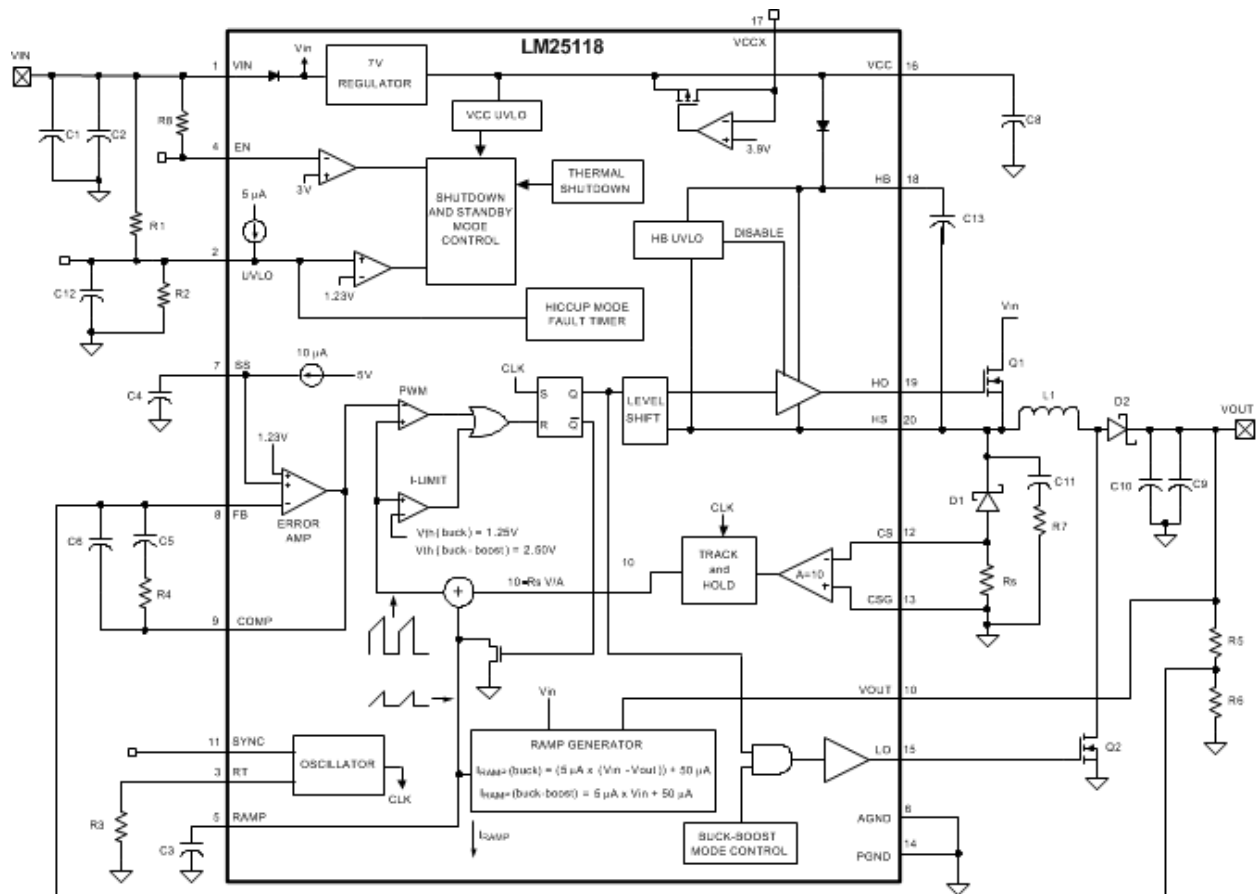


Figure 6.3.1.1 - LM25118 Functional Diagram Schematic (Courtesy of Texas Instruments Inc.)

The Texas Instrument LM25118 3-24V Boost-Buck controller is an accessible reliable component that will suit the project needs. The controller will regulate the voltage to the desired specifications. Texas Instruments provides the full specifications and datasheets on their website. TI offers the ability to design a system to determine the supporting component values.

6.3.2 12V to 6V Voltage Regulations

For the second regulator, the device will input 12V and output 6V. The device of choice will be a buck converter. Buck converters are more efficient than LDOs with energy conservation. The LDOs waste excess voltage as heat. Although the buck voltage regulator is more efficient, it requires more peripheral components than an LDO. With this fact in mind, the LDO is an optional choice in redesign in order to have a more compact PCB. Back to the buck regulator, the model that fits the bill is either the TPS5410D or the LM2576HV. These chips are manufactured by Texas Instruments, and the devices are well stocked at an affordable price. This component will be supplying the 6V to the two servo motors. The servo motors will never be run

simultaneously, so the devices current output will be sufficient for the motors. The datasheet for these parts are available on the Texas instruments website. This will allow us to properly calculate and design all the supporting components in the architecture.

6.3.3 6V to 5V Voltage Regulator

The goal for the third voltage regulator is to drop the voltage from the 6V bucks output to 5V. The goal is to use the same buck converter as the second regulator, since ordering in packaged quantities required by distributor. This is not the final choice for this as for cost we can manage using an LDO for this. The amount of voltage drop is not as significant as the previous, and it would require less peripheral components than the buck. The goal is to aim for the more efficient route, but just like in industry cost can change design goals. This third voltage regulator will be supplying power to the microprocessor, sensors, and LEDs. The one concern for managing these devices is the LEDs. The LEDs may draw an excessive amount of current at full brightness, which may affect the MCU. Further calculations need to be done when all components are acquired to protect against this possible failure. Figure 6.3.3.1 contains a photo of the evaluation model provided by Texas Instruments, which will enable the ability to design for this projects PCB.

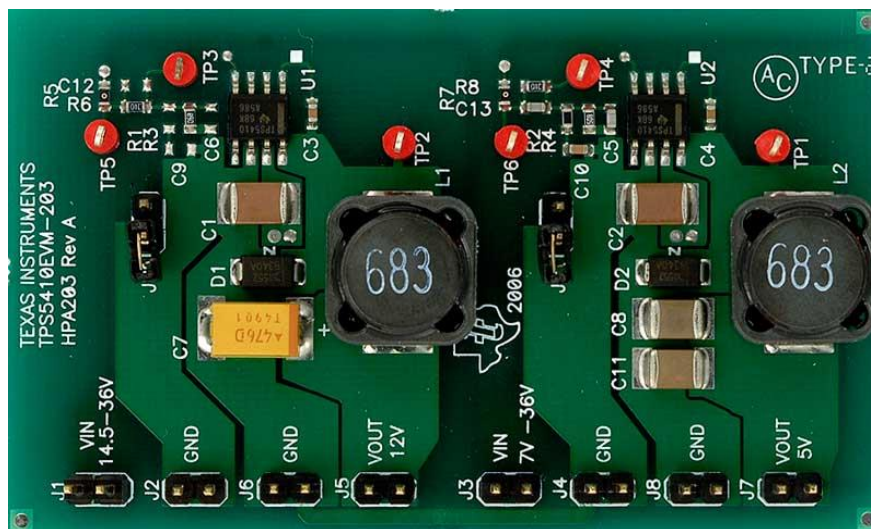


Figure 6.3.3.1 - TI TPS5410 Converter Evaluation Model (Courtesy of Texas Instruments Inc.)

6.4 MCU

This section will be an overview of the microcontroller of choice. The group chose to proceed with the Atmel ATmega2560. The basic specs for the microcontroller are displayed in Table 3.3.1.T1. Prototyping will be done using an Arduino Mega which

contains the same microcontroller. This way all code created during prototyping can be flashed straight to the PCB. The microcontroller has a voltage range from 1.8V to 5.5V, but in order to get max performance, the device needs to be supplied with 4.8V to 5.5V. The clock speed is up to 16MHz which will be plenty for the tasks that it needs to perform. There is 256KB of flash memory, so there is plenty of storage to contain all the necessary commands. The distributor of choice will be Microchip, and the product will be obtained in bulk just in case there is failure. The microcontroller is a surface mount chip. This will create more difficulty if there is any failure on the PCB.

6.5 Sensors and LEDs

This section will introduce the chosen supporting components in the project. These components will include the sensors and the LED strip. Each sensor has been compared and contrasted for proper selection. All devices that were acquired early went through light testing to ensure it is operation and its quality. The tests completed have been performed using an Arduino Mega prototype board.

6.5.1 Light Dependent Resistors

The light dependent Resistors are important in our project, the LDR we are planning to use are called the SUNROM#3190, Table 6.5.1.1 below shows some characteristics:

LDR Characteristics

Parameter	Conditions	Min	Type	Max	Unit
Cell resistance	100LUX	-	400	-	Ohm
	10LUX	-	9	-	K Ohm
Dark resistance	-	-	1	-	M Ohm
Rise time	100LUX	-	2.8	-	ms
	10LUX	-	18	-	ms
Fall time	100LUX	-	48	-	Ms
	10LUX	-	120	-	Ms

Voltage AC/DC peak	-	-	320	V MAX
current	-	-	75	mA MAX
Power dissipation	-	-	100	mW max
Operating temperature	-60	-	+75	Deg.C
Dark capacitance	-	3.5	-	pF

Table 6.5.1.1 - SUMROM#3190 Characteristics

The LDR consist of two cadmium sulphide photoconductive cells, they work based on the principle of photoconductivity. Image conductivity is an optical phenomenon in which the conductivity of matter increases when the light is absorbed from the material. when the photons fall on the device, the electrons in the valence line of the semiconductor material (which is the cadmium in this case), they will be excited in the conduction range. The photons, in the incident, must have more energy than the band gap of the semiconductor material to make the electrons jump from the valence range to the conductivity range. So when the light is lit with enough energy on the device, more and more electrons are being raised to the conductivity of the semiconductor and that would result in a large number of charge carriers. Typical types of the LDR their resistance value decreases with an increase of the light density, during the daylight the resistivity is about 5000Ω and at nighttime the value increases to 20MΩ.

There are a few applications of the LDR:

- Light switch: no need for a manual switch, just let the Light dependent Resistor do the job, so whenever it becomes dark the LDR will turn the light on automatically, whether at home or street lights or even outdoor use, for example the front and back of the house.
- Camera shutter control

There are two types of LDR:

1. negative light resistance, most commonly used, the value of the resistance increases as the intensity of light decreases and vice versa.
2. positive light resistance where the value of the resistance increases as the intensity increases.

LDR is basically a transistor that has a wide base area. when it detects light , a voltage difference is generated and it controls the source current at both ends of the resistance and it acts just like a light switch.

6.5.1.1 Photodiode Light sensors

It works just like a PN diode, but the photodiode works only in reverse bias and has a small reverse bias gap. If the reverse bias voltage is 5V, we have to have a voltage source (a battery) not less of 5V. Therefore, if we place the Photodiode in the forward bias, it will act as a normal diode. Photodiodes are widely used in the electronic industry and in various areas from detectors in CD players to wide-band telephone optical systems. we would use a photodiode sensor in our project if we were going to have Infrared involved, because the infrared consists of the photodiode and an LED emits energy.

Photodiode Characteristics

Parameter	Symbol	Rating	Unit
Reverse Voltage	V_R	32	V
Operating Temperature	T_{opr}	-25 ~ +85	°C
Storage Temperature	T_{stg}	-40 ~ +100	°C
Soldering Temperature	T_{sol}	260	°C
Power Dissipation at (or below) 25°C Free Air Temperature	P_c	150	mW

Table 6.5.1.1.1 - Characteristics of a 5mm PD333 3C/H0/L2 Photodiode

6.5.1.2 Photovoltaic Cells

Photovoltaic Cells, widely used as solar cells and they are used to convert solar light into electricity, this later is known as the Photovoltaic conversion. The conversion of Photovoltaic power is expected to reduce fossil fuel consumption and reduce environmental pollution. Photovoltaic systems are gradually spreading in the lighting industry, communications, water pumping and other applications. They are similar to the photodiode sensors, once they are exposed to sunlight a difference in voltage is created, that will generate an electric current.

Each light sensor we have discussed above has pros and cons, for example the LDR are bidirectional unlike the photodiode sensor which are unidirectional. LDR can be found easily online and they are cheap as well, Photovoltaic cells they basically produce energy, but we are going to use batteries and we can use the photovoltaic cells in case we decide to use solar panels. We have concluded to use the LDR as our light sensor material because they fit the specifications we need for our project, and they are much cheaper than the other types of sensors.

6.5.2 DHT22 Temperature Sensor

The DHT22/AM2302 temperature sensor was selected to be the sensor for the design. The sensor was purchased from Amazon online retailers for 11 dollars and is

manufactured by HiLetgo. This came with two sensors that are mounted to a 3-pin layout. This removes the need to mount our own 10k-ohm resistor between pins two and three. The sensor itself has 4 pins which are VCC, Data, No Connection, and GND. The 3-pin layout is more for convenience than functionality. The sensor requires a 4.7K or 10K resistor to pin 2 which is incorporated in the purchased design. When the product was received, a test was completed to verify quality and performance presented in Figure 6.5.2.2. The test also was achieved to learn how to read results from Arduino chipset. Code provided from Adafruit website was used to test the sensors. The code worked by reading the sensor every moment the sensor took a sample reading itself and displaying it to a serial monitor. The readout provided the humidity, temperature in Celsius plus Fahrenheit, and a temperature high-low range from the readout. The readings were within range of manufacture specifications provided in Table 6.5.2.1 below. Temperature readings were compared to Amazon's Echo temperature readings, as well as, readings taken from in home thermostat. In order to further verify the choice, testing was done using an on-hand DHT11 chip. The readouts of this chip are completely serial. Using header code provided by the kit that the product belonged to, the product was compared to the DHT22. The DHT11 had accurate readings but would fail on majority of the attempts. The DHT22 did not have any failed readouts on the serial monitor. The DHT11 was off by a larger degree but not large enough to make it unusable. At the end of the experiment, the results conclude the DHT22 was a superior selection.

Acquired Manufacturer DHT22 Specifications

Temperature Range:	-40 to 80 degree Celsius
Temperature Accuracy:	+/- 0.5 degrees Celsius
Humidity Range:	0-100% RH
Humidity Accuracy:	-2% to +2%
Power:	Ultra-low power consumption (3.3V or 5V)
Size:	28mm x 12mm x 10mm
Weight:	4g

Table 6.5.2.1 - DHT22 manufacturer specs.

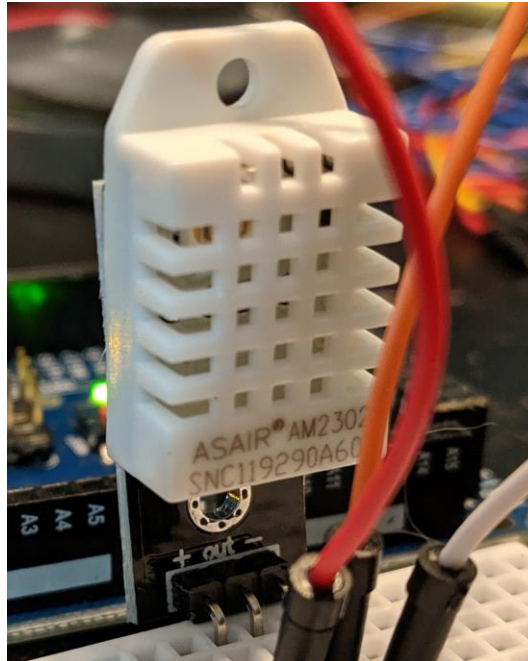


Figure 6.5.2.2 - DHT22 during testing using Arduino Mega 2650

6.5.3 NeoPixel LED Strip

The project is incorporating an LED light strip to utilize a windowsill by transforming it into a lighting centerpiece at night. RGB lighting is a popular market trend in most smart home appliances. The ability to choose custom lighting patterns and brightness levels brings charm to this product. There are many LED brands and choices on the market, but through research narrowed it down to the NeoPixel. The NeoPixel LEDs provide phenomenal brightness and a wide color gamut to select from. In Table 3.2.5.2.1 provides the specification for the NeoPixel strip. The strip purchased is a 30 LED per meter strip that is adjustable by cutting along predetermined lines. The strip is pre-wired with a waterproof diffuser shield. The seller was Amazon online retailers and the distributor was Adafruit. The strip was well within budget at close to ten dollars. To ensure the quality of the product it was tested using an Arduino Mega 2560. The code used to properly test all of the functionality was provided by the distributor Adafruit. The photos in Figure 6.5.3.1 will show the condition of the product and the product during testing. The test showed that the strip had no damaged LEDs and no faults in its wiring. There is one major concern for the strip, the strip is sensitive to overvoltage and humidity. The waterproof shielding will help prevent humidity damage, but the LEDs can be damaged if proper voltage regulation is not acquired. The voltage cannot spike too drastically away from its 5V requirement, which is a product for natural response that can be shielded. Proper regulation will be discussed in the voltage regulation section.



Figure 6.5.3.1 - (Left) Black 30 per meter NeoPixel strip with waterproof sleeve (Right) Strip during testing.

6.6 Motors

Motors are used to drive the blinds features. This section of the document will show the motors selected for each task of the project. The blinds need a motor for lift plus tilt, and the shades need a motor for tilt. Each motor that was selected fits many project goal criteria. Each task will be presented below with the accompanied selected motor. The subtopics will provide a brief specification table for reference. Included will be photos of each product to verify each product, so that recreation of the design components are possible for testing. The photos will also be used to reference against potential low grade copies of the motors.

6.6.1 Tilt Motor

The TIANKONGRC MG-996R is the motor selected to manage the tilt bar for the blinds. In Table 6.6.1.1 will provide the full specification for the MG-996R servo motor. The servo motor was purchased from Amazon online retailers. It cost ~20 dollars for a 4-pack of these servos, so each servo is roughly worth 5 dollars apiece. The motors all came with various attachment peripherals that will help facilitate situational mounting. In order to check the motor quality and performance tests were administered. The results of running a few simple tests using an Arduino mega board with AT2650 processor, the motor ran with ease and precision. Three tests were performed to learn the functionality of this motor. The first test was an Arduino provided sweep test. This test ran the motor from 0 to 180 degrees and back with micro delays. The next test was a knob test provided by Arduino examples. This test uses a potentiometer to adjust the servo position. This

test, presented in Figure 6.6.1.2, is closer to the applicational use this motor will be applied to. The test highlighted obstacles that might become problematic. The 10k potentiometer used was faulty at low ranges. This made the readout jump out of bounds of the servo's degrees of freedom. The false out-of-bound readings caused the servo to thrash. This is something that should be protected against in the final design. The last test used serial commands to adjust the motor position. The design goal is to control tilt with an app slider or Alexa command. The ability to send digital commands to adjust the blinds tilt is a crucial component. The test was fruitful provided a deeper understanding of how to control this device digitally with precision.

MG-996R Servo Motor Manufacturer Specifications

Degree of Rotation:	180 degrees
Dimensions:	40mm x 19mm x 43mm
Weight:	56g
Operating Speed (6V):	0.13sec / 60 degrees no load
Stall Torque (6V):	11 kg-cm
Operating Voltage	4.8 to 7.2V
Gear Type	All metal gears

Table 6.6.1.1 - Specifications for MG-996R servo motor.

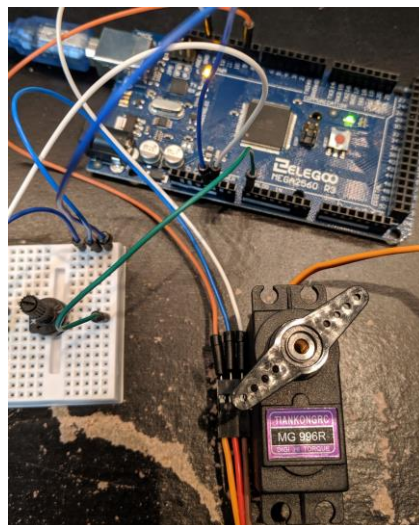


Figure 6.6.1.2 - 180 degree MG-996R Servo Test

6.6.2 Lift motor

This section will pertain to the motors selected and purchased for the project. The motors listed below were chosen for the specifications and inclusion in other similar projects. Both motors are capable of performing tasks that require high torque. Their design allows for ease to include self-created modular components for mounting. These motors provide adequate performance in a compact form factor. This will allow for more space within the blind's chassis for cable management and other components.

The lift motor uses a continuous rotation motor the SRC SM-S4315R. Table 6.6.2.1 below will provide specifications for this servo. This servo was purchased from Amazon online retailers for around 20 dollars for one. A backup servo was also ordered through Amazon retailers for around 20 dollars for one. The bill of materials will state the exact pricing of each product. The backup motor serves two purposes. The first purpose is if the chosen motor fails, and then for the case we need more torque. Table 6.6.2.3 will provide specifications for the JX PDI-5521MG servo. The continuous rotation servos do not come with encoding capabilities so there will need to be a method to compensate. Two methods that will be tested during prototyping will be using timing delays or relying on a rotary encoder. The rotary encoder will need to be geared to the pulley system for the blinds lift cord. This will allow the ability to track the amount of rotations necessary for each rotation. The encoder will provide a faster method for calibration compared to a time delay that may vary in low voltage scenarios. Both motors will be depicted in Figure 6.6.2.2 for visual comparison.

SM-S4315R Servo Motor Manufacturer Specifications

Degree of Rotation:	360 degrees
Dimensions:	39.5mm x 20.5mm x 42mm
Weight:	60g
Operating Speed (6V):	0.14sec / 60 degrees no load
Stall Torque (6V):	15.4 kg-cm
Operating Voltage	4.8 to 6V
Gear Type	All metal gears

Table 6.6.2.1 - Specifications for SM-S4315R servo motor.



Figure 6.6.2.2 - (Left) SM-S4315R Cont. Rot. Servo (Right) JX 5521MG-360 Cont. Rot. Servo.

JX 5521MG-360 Servo Motor Manufacturer Specifications

Degree of Rotation:	360 degrees
Dimensions:	40.5mm x 20.2mm x 38mm
Weight:	55.6g
Operating Speed (6V):	65T/S
Stall Torque (6V):	20.32 kg-cm
Operating Voltage	4.8 to 6V
Gear Type	All metal gears

Table 6.6.2.3 - Specifications for PDI-5521MG servo motor.

6.6.3 Shade Motor

The Design will be using a motorized roller shade to add more lighting variations for customers. The motor used to operate the shades lift will be a 12V dc motor. The motor purchased is a Rollerhouse Electric Roller Blinds Shades Tubular Motor Kit. The motor was purchased from Amazon retailers for 43 dollars. The manufacture specs will be provided in Table 6.6.3.1 with the Figure 6.6.3.2 below it. The blinds are simple and will be controlled through analog means. The functionality goals for the shade will be full up or full down. The diffuser shades purpose it to provide soft light into a room, or to be used to help disperse the LED's light for effects. Since the roller shade is a simple 12V dc motor, the Arduino will need to be calibrated using delays for operated a relay switch. There also needs to be the ability to reverse polarity which will in turn drive the motor in

the opposite direction. Ounce fully mounted testing will be done to set operating delay to learn motor constraints.

12V DC Motor Manufacturer Specifications

Motor Length:	242.5 cm
Operating Speed:	37 rpm
Operating Voltage:	12V
Operating Amps:	0.5A
Max Run Time:	No Limit
Working Temperature Range:	32 °F to 140 °F
Cable Type:	2 x 0.20 mm

Table 6.6.3.1 - 12V DC motor shaft specs.



Figure 6.6.3.2 - Rollerhouse 12V motor for shades.

6.7 Window Frame Construction

This section will cover the frame design and construction for this project. The frame will mimic a windowsill to a degree. The design will be able to accommodate the weight of all components with space for optional additions. The frame will have space to factor in any possible appendage that might protrude from the motors. The frame will need to be disassembled and reassembled for transport to accommodate compact cars. The frame is expected to be low cost, and the total cost shall not exceed 20 dollars. The bottom of the fully extended blinds will meet a border to help with the windowsill elusion. There will be a flat board space above the blinds to allow the PCB to be mounted. These are a set of goals to reach in the final construction.

6.7.1 Frame Design

The blind set purchased dimensions are 24in length by 36in width. These blinds are 1-inch faux wood blinds that have a half in u-mount. The half in u-mount was picked to allow for the roller diffuser shade to be mounted behind. The shade assembly will be covered by the 6 monocrystalline solar panels. The solar panels will be mounted to a flat board that will have a half an inch to a quarter inch border. This design choice was influenced by the solar shades project referenced in section 3.1. The Figure 6.7.1.1 will show a concept of the design. Theoretically the design will be 5 to 5.5 feet tall with 27-inch width. The extra width will be to mount blocks on the side that will be the top of the windowsill. These blocks will help level and mount the flat board above as it will rest on the protrusions. The legs will be 2.5 to 3 feet long to allow for a sturdy center. These will be the main goals in constructing the frame which many similarities can be contrasted from the solar shade project. The key concept ideas taken were the monocrystalline solar panel mounts, frame support, and frame outer skeleton. From a couple trips to the store the cost should be around 15 dollars in wood. The screws should only add a nominal cost to the total design cost.

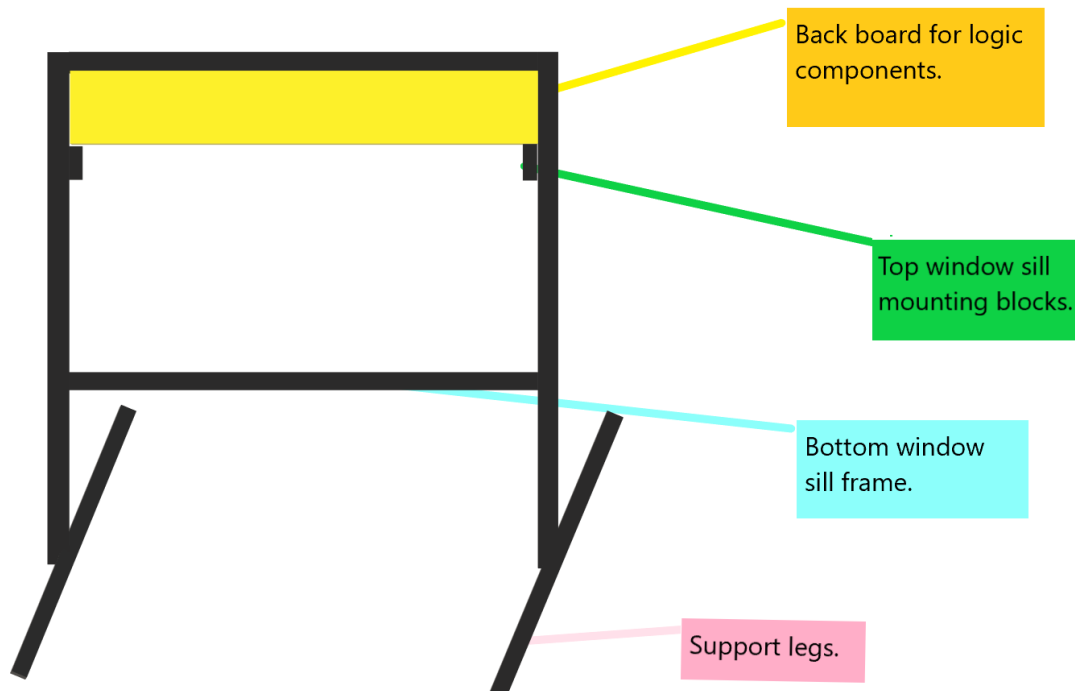


Figure 6.7.2.1 - Concept art of frame design.

6.7.2 Frame Construction

Materials to construct the frame will be purchased from a home improvement warehouse. The wood and screws were bought from Home Depot for a grand total of 19 dollars. This was within the budget constraints set in the design. The wood was cut into lengths of two 27in, two 60 inch, two 6inch, and two 32.5inch beams. The width of each beam is 3.5 inches and the depth of each beam is 1.5 inches. Three total beams were purchased at around 4 dollars apiece, and there are pieces left to spare after the adjustments were made. The wood will be assembled just like the concept design in Figure 6.7.1.1 and will be displayed in Figure 6.7.2.1. The two 27-inch slabs will be used as the very top of the frame and the windowsill simulated bar in between. The two 32.5-inch beams are used as the support legs. The two 6-inch beams will be where the blinds and shades are mounted. These blocks are mounted 10 inches from the top of the frame to allow for mounting space. The 60-inch beams are for the columns that all pieces of wood will be mounted to. The legs will be mounted with bolts to allow for disassembly. The rest of the screws are construction grade wood screws so that the support is sturdy. The two 6-inch slabs will also be supported using wood glue for extra sturdiness in that region. Most of the weight will be pulling from those two slabs. Each beam was sanded to remove any splintering wood fragments.



Figure 6.7.2.1 - Phase one of the frames.

The wood warped in storage, so some alterations need to be made. The legs will be shortened, and the flat panel still needs to be added above. The piece to finish the windowsill appearance is not long enough because the wood warped outward on both sides. The frame is sturdy otherwise the defects affecting it in this stage. The next phase will be the finished frame. The flat board will be measured and cut, yet it will not be placed on the board until after all motors are mounted. The frame for the solar blades will be constructed in the last phase, so there is no obstruction to the shade mechanism. Phase two will fix all the faults, phase three will be full mounting of objects (excluding solar panels), and phase four will be the solar plus PCB panel.

6.8 Voltage Regulator

The power supply circuitry is of extreme importance in the design of any electronic or electrical system. One of the key components in the power supply circuit is the voltage regulator. When an AC voltage is converted into a DC voltage, a ripple still exists in the voltage waveform. A filter capacitor is commonly used to filter out that ripple and smoothen the voltage waveform. However, capacitors are passive elements and do not provide 100% ripple free voltage output. Hence, to achieve an optimal level of voltage regulation, voltage regulator ICs or circuits are used. A voltage regulator ensures a stable

DC voltage without any significant ripple level. Microcontrollers and other integrated circuits are very sensitive to voltage variations and even a small increase in voltage might burn the IC. Hence, the supply of stable DC voltage to electronic components is very important for the safe functioning of an electronic system.

The basic architecture of a voltage regulator comprises of current amplifier, reference voltage and error amplifier. The voltage regulator is basically a feedback control system. The reference voltage is the voltage level that we want to develop at the output. The actual output voltage is sensed and sent back to the input of the amplifier. The error amplifier determines the voltage difference between the actual output voltage and the desired output voltage. According to the magnitude of this difference value, the amplifier adjusts its output and tries to achieve the reference voltage level by making the error difference to be zero.

Voltage regulators have two main types which are linear regulators and switch mode regulators. The linear voltage regulators work on the “buck” principal which means that their output voltage level can only be less than the input voltage level. They cannot provide compensation for lower input voltages. The main advantage of linear voltage regulators is their simplistic design and good EMC characteristics. However, these regulators tend to be bigger in size than the switching regulators. Also, these regulators generate a large amount of heat energy during their operation. This requires the use of bigger heat sinks with these regulators which increase the cost. In case of excessive heat, the regulator might get fried. Due to the absence of higher switching frequencies, these regulators do not cause the issue of electromagnetic interference. However, they are less efficient as compared to the switching regulators.

The linear voltage regulators can further be divided into two categories which are the series regulators and shunt regulators. The series voltage regulators are more efficient as compared to shunt regulators. The reason for this is that in shunt regulators are significant amount of current flows to the ground and causes energy wastage. The series regulators have the voltage adjustment element present in the series of the load. This series element provides easier and more effective regulation of load voltage.

In context of our circuit operation, the linear voltage regulator might not be the suitable solution. Our system is solar powered and needs the circuitry to be present outdoors. The direct exposure of the circuit to the sun causes it to heat up quickly. The linear voltage regulators already generate good amount of heat and the sun exposure might cause them to fail more often. Hence, we must consider some alternative architecture for the voltage regulator in our project.

Since we have established that a linear voltage regulator is unsuitable for our project, we must consider a switching type voltage regulator for our power system. The greatest feature of switching regulator is that the output voltage can be greater than the input voltage. This means that switching regulators have “buck” as well as “boost” capabilities. The switching regulators usually employ power electronic converters with smart control circuitry which ensures higher rate of efficiency. The switching converters adjust the output voltage in incremental steps and due to this reason the amount of generated heat is limited. The switching regulators are more complex than the linear regulators and have more EMC related issues. The high frequency switching operation causes the generation of harmonics and noise which causes the issues of electromagnetic interference.

There are three main types of switching regulators which are pulse width modulation, hysteresis, and buck/boost regulators. The pulse width modulation technique uses digital signal processing technique in which the input signal is compared with a reference signal a constant frequency waveform is generated with varying duty cycle. The duty cycle of the waveform depends on the output power requirement. The greater the output voltage needed, the greater the pulse width or the duty cycle of the waveform. Due to the constant frequency of the pulse waveform, the PWM regulators have a lower EMI issue.

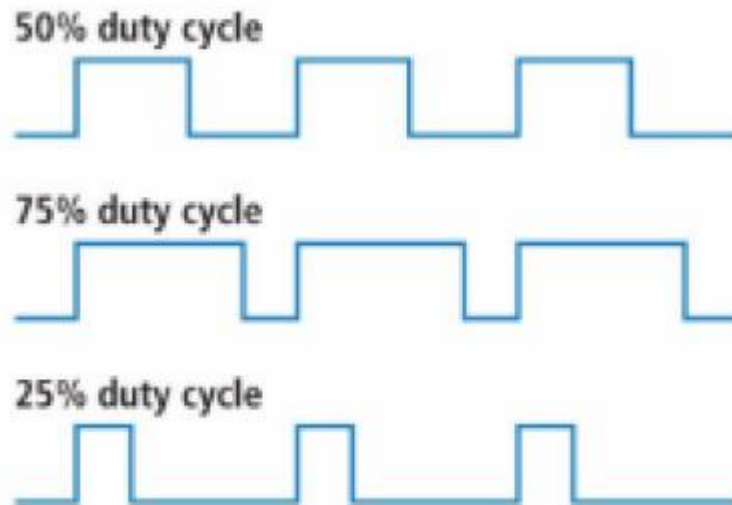


Figure 6.8.1 - Pulse Width Modulation

Hysteresis based voltage regulators are the simplest and most cost-effective type of switching regulators. The design of hysteresis-based regulators only contains a comparator and reference voltage signal. The design does not include any error amplifier. Although the design of this regulator is very simple, the major disadvantage of this design is the lack of accuracy. The simple design and cost effectiveness come at the price of lowered accuracy. The design does not include any clock pulse or duty cycle modulation which renders this design simple but inaccurate. The lack of accuracy of this regulator might cause impediments in our project and might not meet the requirements demanded by the project. Hence, we cannot rely on this type of regulator for our performance critical project.

The third type of switching voltage regulator is the buck boost regulator. The buck converter design is a step-down design which can lower the input voltage level. The boost converter design is a step-up design which is capable of increasing the input voltage level. The combination of these two converters result into a buck boost converter circuit which is capable of stepping up as well as stepping down the input voltage level. Combining the buck boost converter design with pulse width modulation control technique results into a very efficient regulator design.

Voltage Regulator Comparison

Voltage Regulator	Type	Pros	Cons
Shunt	Linear	Simple circuit, Low ripple/emf interference	Draw current to ground; can only buck
Series	Linear	Won't pull full current regardless of load	Can only buck
PWM	Switching	Highest efficiency	More complex circuit
PFM	Switching	High efficiency regardless of load	May cause heavy interference with other electronics
Hysteresis	Switching	Simple and stable circuit	Not synchronized
Hysteresis-based	Switching	Low power and stable	Not synchronized
Buck-Boost	Switching	Wide application	More complex circuit

Table 6.8.1 - Voltage Regulator Technologies

Another voltage regulator is the LP2950Z. There are two types of LP2950Z, 3-Pin SOT 23A which has only one input voltage and one output voltage, and also there is the 3-PIN SOT-89 that has two input voltage and an output voltage, in our project we are going to use the two input voltage. The MCP1702 is an LDO (Low Dropout Regulators), and essentially has a high precision, low voltage controller that requires slightly higher input voltage to work with.

LP2950Z Specifications

Parameter	Symbol	Min	Type	Max	Unit
Input Operating Voltage	V _{in}	2.7	-----	13.2	V
Maximum Output Current	I _{out-mA}	250	_____	_____	mA
Input Quiescent Current	I _Q	_____	2.0	5	µA
Output Short Circuit Current	I _{out_sc}	_____	400	_____	mA

Table 6.8.2 - Characteristics of LP2950Z

The diagram below shows the layout of the voltage regulator pins. There are two types of LP2950Z , 3-Pin SOT 23A which has only one input voltage and one output voltage, and also there is the 3-PIN SOT-89 that has two input voltage and an output voltage , in our project we are going to use the two input voltage.

The MCP1702 is an LDO (Low Dropout Regulators), and essentially has a high precision, low voltage controller that requires slightly higher input voltage to work with, the table below shows some of the characteristics of the LDO Voltage Regulator:

MCP1702 Specifications

Parameter	Symbol	Min	Type	Max	Unit
Input operating voltage	V _{in}	2.7	-----	13.2	V
maximum output Current	I _{out-mA}	250	_____	_____	mA
Input Quiescent Current	I _Q	_____	2.0	5	μA
Output Short Circuit Current	I _{out_sc}	_____	400	_____	mA

Table 6.8.3 - Some characteristics of MCP1702

We need two voltage regulators in our circuit, one for the microcontroller and one for the servo motors. For the servo motors we need a voltage regulator which converts 12Vdc to 5Vdc. Whereas for microcontroller we need a voltage regulator which converts 5Vdc to 2Vdc. The first voltage regulator that we have selected for our power distribution system is the LM2576 from Texas Instruments. LM2576 is a series of monolithic integrated circuit type voltage regulators. These are step down voltage regulators and do not have the boost capability. These regulators have a high current rating of 3A with excellent line and load regulation. These regulators are available in 3.3V, 5V, 12V, and 15V variants. An adjustable output version is also available.

These voltage regulators have a simple construction and require very less number of external components. They also include a fixed frequency oscillator and fault protection circuitry. These are high efficiency voltage regulators and have excellent heat transfer capabilities. Due to this reason, in many cases, these regulators might not need a heat sink.

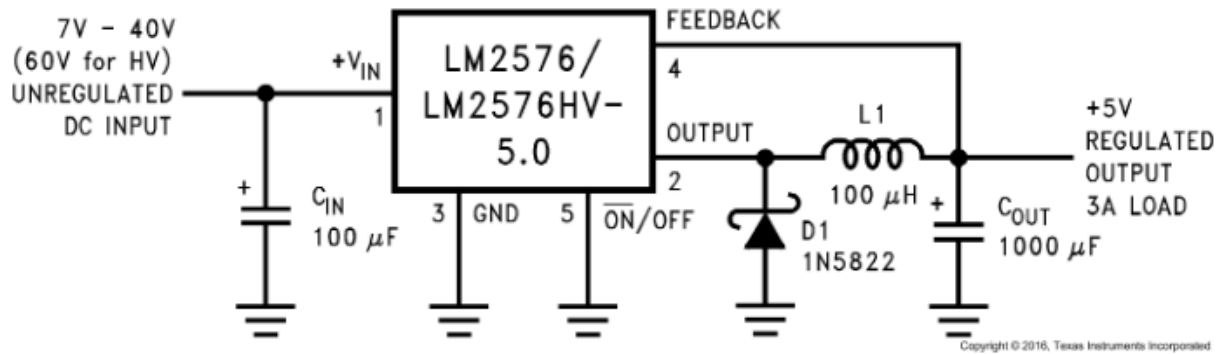
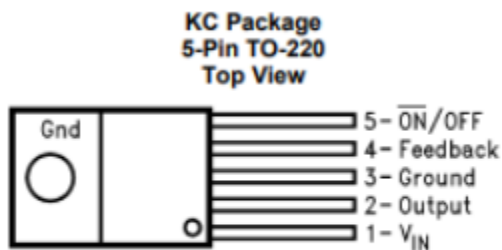


Figure 6.8.2: LM2576 Used in our project

The pin configuration, dimensions and different mechanical packages of the voltage regulator are shown in the following image:



A detailed description of the pins present on the package and their respective functions is given in the following table;

Pin Functions

PIN		I/O ⁽¹⁾	DESCRIPTION
NO.	NAME		
1	V _{IN}	I	Supply input pin to collector pin of high-side transistor. Connect to power supply and input bypass capacitors C _{IN} . Path from V _{IN} pin to high frequency bypass C _{IN} and GND must be as short as possible.
2	OUTPUT	O	Emitter pin of the power transistor. This is a switching node. Attached this pin to an inductor and the cathode of the external diode.
3	GROUND	—	Ground pin. Path to C _{IN} must be as short as possible.
4	FEEDBACK	I	Feedback sense input pin. Connect to the midpoint of feedback divider to set V _{OUT} for ADJ version or connect this pin directly to the output capacitor for a fixed output version.
5	$\overline{\text{ON}}/\text{OFF}$	I	Enable input to the voltage regulator. High = OFF and low = ON. Connect to GND to enable the voltage regulator. Do not leave this pin float.
—	TAB	—	Connected to GND. Attached to heatsink for thermal relief for TO-220 package or put a copper plane connected to this pin as a thermal relief for DDPACK package.

(1) I = INPUT, O = OUTPUT

Table 6.8.4: Pin Functions of the LM2576

The power ratings, current ratings, and voltage ratings of any electronic device are of utmost importance. Failure to comply with these ratings can very easily result into the damage of the device as well as the failure of the system. This can cause undesirable situations as well as security and safety risks in safety critical systems. Hence, it is important to keep the power ratings of a device in consideration while designing the electronic circuit. The voltage, current, and power ratings of the voltage regulator are presented in the table below:

Pin Config Table for LM2576

		MIN	MAX	UNIT
Maximum supply voltage	LM2576		45	V
	LM2576HV		63	
$\overline{\text{ON}}/\text{OFF}$ pin input voltage		$-0.3\text{V} \leq V \leq +V_{\text{IN}}$		V
Output voltage to ground	(Steady-state)	-1		V
Power dissipation		Internally Limited		
Maximum junction temperature, T _J			150	°C
Storage temperature, T _{stg}		-65	150	°C

Figure 6.8.3 : LM2576 Pin configuration

6.9 Hardware Design Summary

The following section deals with the exact specifications of the SunShade design, including an analysis of similar project schematics, and the finished SunShade schematic, as well as a justification of design. In this section, each component of the 2015 project will be analyzed, so that educated choices can be made in the new design. Once this comparison is made, the current schematic will be realized, and the process and specifications will be recorded.

6.9.1 Comparison to 2015 Solar Blinds Project

The 2015 Senior Design solar blinds project has already been discussed at length in previous section but finds new usefulness when designing the SunShade schematic. Tearing down both projects, while very different in functionality and appearance, we can find that the only major difference between the two is an addition of extra motors, WIFI usage over Bluetooth, major mobile application changes, no e-paper display, Manual buttons, LEDs, and voice control connectivity. This may seem like quite a lot, but within the schematic, the two are quite similar. Below is the complete 2015 schematic, as represented in Eagle electrical CAD software. This design will be used in order to complete the SunShade design, and ideally, improve upon the older design, learning from its shortcomings and improve efficiency, as well as streamline the appearance of the finished schematic. Figure 6.9.1.1 Displays the 2015 schematic for comparison.

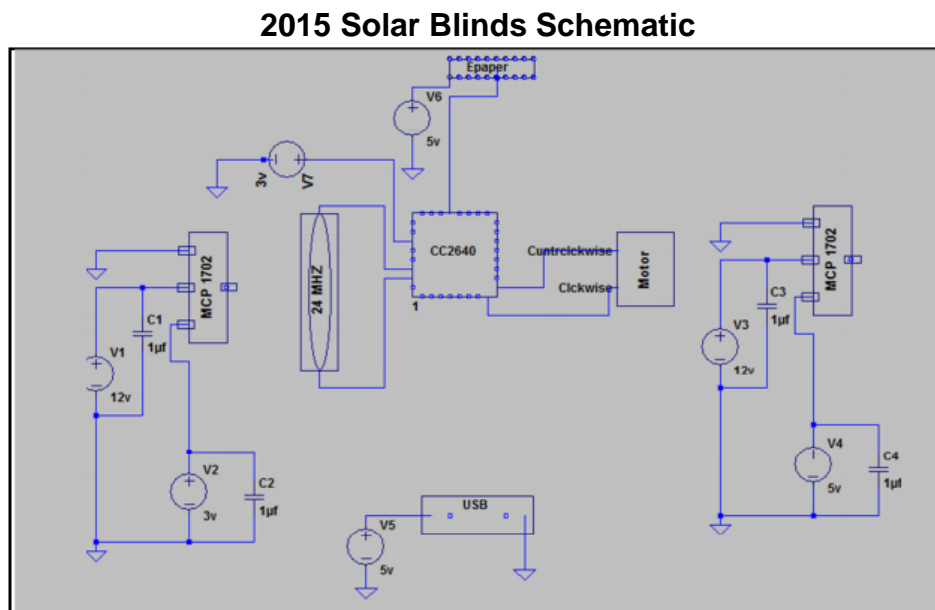


Figure 6.9.1.1 - Complete 2015 project represented in Eagle

Above, we can see the CC2640 Microprocessor, and its various pins, denoted as a square in the center of the schematic. This is interesting, as it provides insight into how these components are represented within digital designs. For software such as EasyEDA, users can submit new components for implementation, so that almost any purchased microcontroller can be perfectly added to a schematic. It is likely that for final schematic design, EasyEDA will be the primary medium of creation. Connected directly into the microprocessor is the 24MHz crystal component used for timing, and the motor. The motor has two pin connections, for clockwise, and counterclockwise movement control, in order to allow for up and down motions. The SunShade will have additional motors for tilt, and the possibility of a 360-degree turned diffuser shade, all of which will require clockwise and counterclockwise movement in order to extend and retract or tilt up or tilt down. The project may need a rectifier between the motors and the microprocessor to protect the processor from unwanted voltages from the motor.

Another element connected directly to a data line of the microprocessor is the e-paper. Similar to the processor, it is a 24-pin layout, which the 2015 project uses to display battery life, temperature, etc. Since the SunShade team will not have a display, as all information will be conveyed through the app, this component and its connections can be removed. The 2015 Project also has a phone-charging feature, using a USB port, which the SunShade team will also not require for their design, since the power constraints put forth by using WIFI, which draws greater power, as well as having a much smaller/cheaper solar power source, the team does not wish to negatively influence performance of the final product by adding an extra, and significant, draw on available energy, when there is no practical need to charge a phone from the blinds appliance.

Moving on to the crystal- this element is in fact something the SunShade team will still require, as it provides computational reference for timing by vibrating the crystal at specific frequencies. It will be connected by two inputs into the MCU. Capacitors may need to be implemented in order to allow the crystal to oscillate properly, but the microcontroller may actually have internal capacitors, which may eliminate a need for them. This component can be easily drawn up in Eagle or easyEDA for schematic design. The next element is the voltage regulator, which is imperative for effective, safe, and efficient operation of the PCB. Voltage regulators are designed to take in the input voltage of a system, and automatically maintain a consistent voltage level, using a feed-forward design or negative feedback, to lower the output voltage to a consistent and stable level acceptable to given components. The 2015 team used a MCP1702 Linear Voltage Regulator, which is represented by a rectangle on both the leftmost and the rightmost sides of the schematic, with three pins on one side and one pin on the other. The first is the ground, followed by the voltage input, followed by the voltage output. The design required two regulators, as can be seen on the schematic above, one of which is a 12v - 3v regulator, while the other is a 12v-5v regulator. The 12-3 is used to sufficiently power the microcontroller safely, while the 12-5 is used to power the USB and e-paper LCD display. It is currently under consideration whether or not the Sunshade team will use 12 or 6v for their power source, because of the fact that almost all elements and components can be powered by at most 6 volts, but the motor under consideration for this lifting feature, to pull the blinds up, will require more power than that of the tilting motor. If a proper motor can be selected to operate at 6v, it will be selected to streamline the design.

Due to the similarities between the 2015 SolarBlinds Senior design project and the 2019 SunShade senior design project, the analysis above will serve as a critical lesson in schematic design and development, as the documentation provided by the previous team will allow the Sunshade team to proceed with an unparalleled, in-depth look into what it takes to complete such a project, as well as any major mistakes, errors, or oversights that may have occurred for the previous team, so that it may be avoided presently. When reaching out personally to a member of the original 2015 team, the University of Central Florida Engineering alumni spoke of one such example of errors to avoid, and the Alumni stressed the importance of the team member in charge of application development being fully informed with regards to the motor specifications at hand. An error that the 2015 team encountered was a direct result of a shortsighted error made by their hand, sending too strong a voltage back into certain components, which resulted in their printed circuit board igniting, and needing to be reordered and soldered soon before the final deadline. This extreme case of design accident is a perfect example of the importance of all team members being fully aware of all moving parts involved with the electrical system, as it is extremely delicate. Fortunately for the 2015 team, they were able to successfully complete the project with it being fully operational at the secondary presentation opportunity.

Moving forward, the Sunshade team will begin their design of the project schematic, as well as a testing phase, so that it may be implemented into first a prototype, and then a full printed circuit board.

6.9.2 Schematic

The schematic will serve not only as a comprehensive diagram of functionality of the SunShade project, but also as a precursor to the prototype, as well as the PCB design. For this stage of the schematic, a simple wiring diagram will suffice, as a way to explain what connections must be made and where, while also visually representing all components involved in the project, and which elements they must communicate with. This section requires a solid understanding of pin structure in order to effectively meet the requirements of each component in order for their integration to be successful.

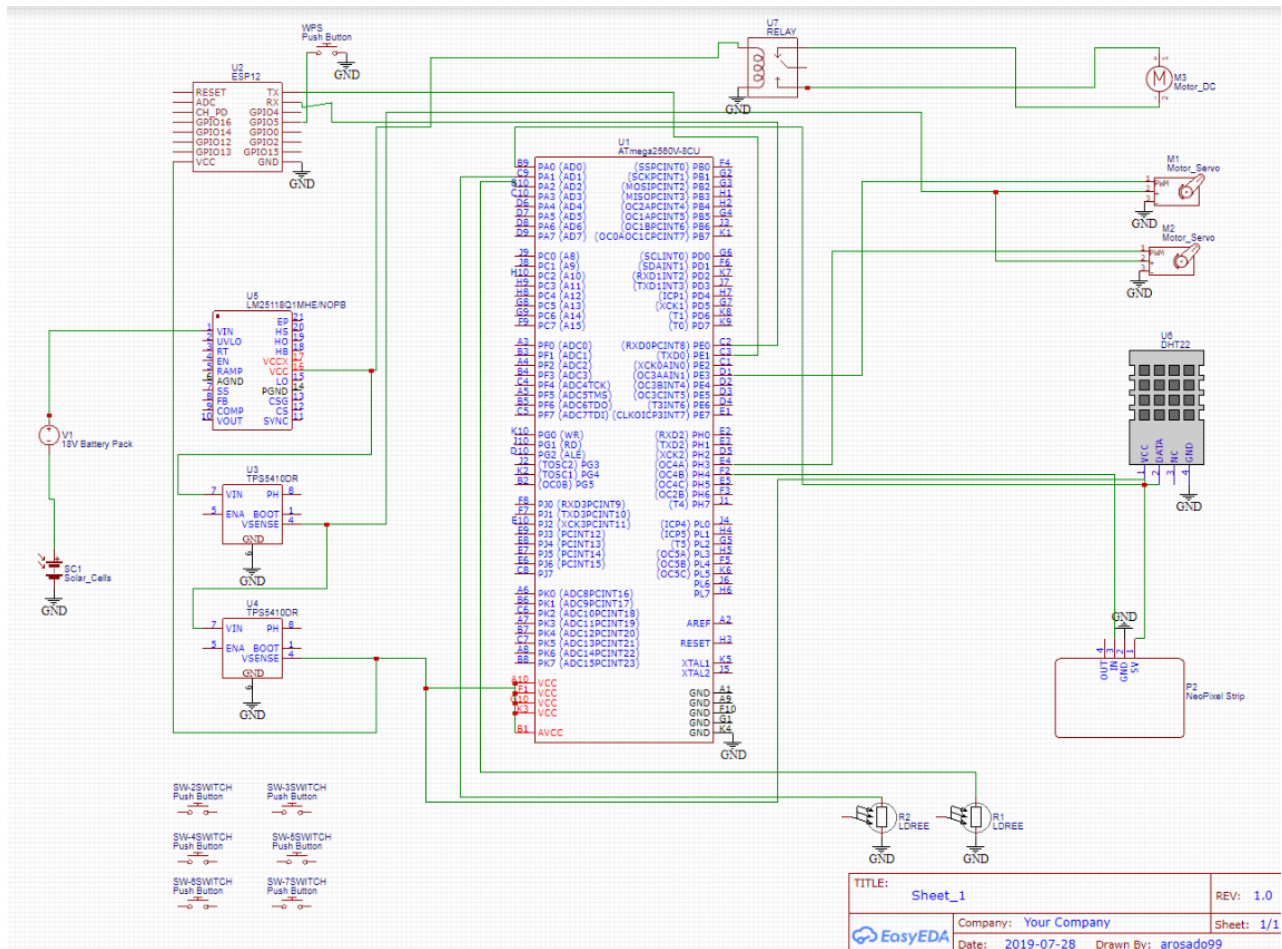


Figure 6.9.1.2 - Mockup SunShade electrical schematic

7.0 Software Design

The software design section will cover the core components for programming. This will include components necessary for the Arduino microcontroller and the application.

7.1 Firmware

Firmware is defined as the software of a specific hardware used for many different applications. The firmware we will be using on the microcontroller will allow for the communication of the temperature and light sensors, servo motors, lights, Bluetooth, and WIFI modules. The microcontroller will receive data from these modules/peripherals and control them as well. Microcontrollers use an assembly language-based software to develop the logic, some will allow use of intermediate level languages such as Verilog, or microcontrollers such as the Raspberry Pi will accept high level languages i.e. Java, C, and C++. The Arduino software utilizes a compiler based on C and C++. The Arduino

language is specific to the brand and many of the libraries that are written to support the hardware are written using C++. There are certain abstractions that are added to the language which make it simple to use for enthusiasts, however, more experienced programmers can utilize the object-oriented methodology to compile software.

An object-oriented software design will allow classes of object to be created to encapsulate the variables and methods. By approaching the software design in this manner, the program flow begins to step away from normally implemented procedural execution and moves toward the object-oriented approach in which different objects interact with one another. By encapsulating each one in this way, we begin to see interaction between the device classes rather than the loop method needing to manage the interactions with continuous checking statements or comparisons. Each device class will be responsible for not only managing itself, but also understanding when to interact with the other device classes.

7.1.1 Arduino Programming

The operational software for the Arduino is divided into five main components: libraries, global variables, setup, loop, and additional functions. The software running on the Arduino allows for an initialization of all of the modules and continuous processing of the inputs and outputs of the connected sensors and peripherals.

To begin, the Arduino microcontroller must import the correct libraries that are necessary to simplify the future operations. The libraries define pins and allow for easier manipulation of data and registers. The global variables are defined before the program executions for the variables that will be frequently used or changed. In the case of smart blinds, these global variables will be the WIFI SSID/password, the database server, and servo motor positions.

The setup component is where pins are defined as inputs or outputs, serial communication is initialized, and different functions that require single execution during startup for the rest of the code to run continuously. When the board is turned on, the setup function will run first and complete these initial operations. If the microcontroller loses power or rebooted, the software will once again run the setup function a single time. Upon completion, the microcontroller will start to execute the loop function of the code.

Within the loop component, the main operations of the microcontroller are executed. The loop function runs continuously while the device is powered on until it is reset or loses power. The purpose of the loop function is to read the inputs of the sensors and create outputs based on the data. The microcontroller will continuously run the loop function so to perform other operations the software will call additional functions based on current conditions of the sensors and servo motor.

The additional functions will include a process to communicate through WIFI or Bluetooth, controlling the servo motors and lights, and converting the analog readings from the sensors into a bounded range. These functions will be called during the loop if needed.

7.1.2 I2C Communication

The Arduino microcontroller has many methods of communication via digital, analog, and PWM pins. Due to the many sensors connected to the microcontroller, we will be taking advantage of the microcontroller I2C protocol to communicate between the sensors and peripherals.

The I2C communication protocol is a popular technology due to its nature of acknowledgments between a master and slave devices. The protocol relies on just two wires for communicating with multiple devices. Two signal wires, SCL and SDA, are used for communication. The two signal wires are considered to be open drain since pull up resistors are used to pull the signal high when the devices are not communicating, and the devices will pull the signal low during communication. During design, a master device must be selected as the device which initiates all communication between itself and the slave devices.

With I2C communication, every device is designated as a Master or a Slave. In most implementations, a system will consist of a single master device which drives the SCL clock line and initiates all communication with slave devices. Slave devices act as responders to the requests being made by the master device. Each slave is assigned a specific address so that the master is capable of selecting the device it wished to communicate with. In this case the Arduino microcontroller will be the master while the sensors and peripherals will be the slaves. Since only a single master device exists within the system, conflicts between devices cannot exist due to the nature that slave devices are unable to initiate transfers of data.

When a master device requests data from a slave device, a specific pattern of bit sequences must be adhered to. The data can be divided into 8-bit sequences of request and response between the master and slave. To begin, the master must transmit a start sequence along the SCL and SDA communication lines. The start sequence is defined by holding the SCL 13 clock line high while the SDA data line is allowed to change from high to pulled low. As will be addressed in the future, this is significantly different than the data transmission stage where the SDA line must remain constant during a high clock cycle.

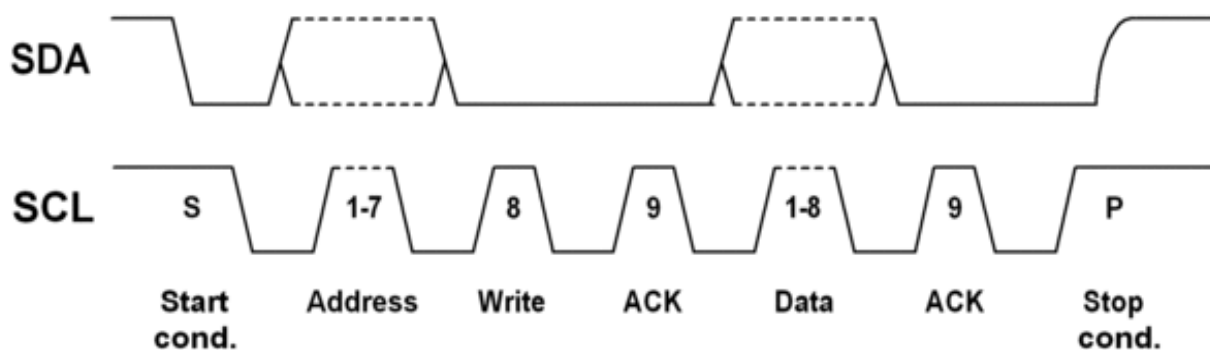


Figure 7.1.2.1 - I2C Bus Protocol

Following the start sequence, the master device will transmit a 7-bit sequence containing the address of the slave device in which it will be requesting communication with followed by a read/write bit. Due to the 7-bit addressing sequence, there are 128 possible addresses, but I2C reserves certain addresses for specific functions, decreasing the number of possible devices to 112 as previously mentioned. The 7-bit address is sent with the most significant byte first, ending with the least significant byte. The final byte of the first 8-bit sequence is the read/write bit, indicating the operation the master wishes to conduct on the slave.

Following the first 8-bit sequence addressing the device the master wished to communicate with, the master device will relinquish control of the SDA data line for use by the slave device. The slave device sends an acknowledgement to the master by pulling the SDA line low, as a way of stating that it has received the message and is prepared for the next stage. In the case that the acknowledgement bit is not sent, the master device must decide how to proceed with the operation.

The master device will continue to pulse the SCL clock line at regular intervals while data is transmitted over the SDA until a stop bit is transmitted. The stop bit is the opposite operation of the start bit, where SDA must be allowed to return to high from being pulled low by the device during a point in time where the SCL clock line is high.

7.1.3 UART Communication

For the initial setup of the Smart Blinds, we will have a connected WIFI module to access the internet. This module will be connected via UART communication to ensure a reliable exchange of data between the database and Smart Blinds.

UART Serial Communication, is a peer-to-peer asynchronous protocol in which two devices communicate using two signal lines. The universal asynchronous receive/transmit 14 protocol is an intermediate serial stage used to connect the parallel interfaces of two devices. This is completed by shifting out a bit sequence from a register and transmitting the data bit by bit across the signal lines, where the receiving device will shift in the data bit by bit at the same rate as transmission. In short, the parallel data is converted to serial data by the transmitter, and serial data is converted to parallel data by the receiver. Unlike I2C, a device can act as both a receiver and a transmitter in the same system and is not assigned a set role.

Key concepts in understanding the UART communication method lies in first recognizing that the protocol is asynchronous. Asynchronous communication is defined as the transmission of data intermittently between two devices rather than a continuous data stream with an external clock governing the transfer. In this design, the devices are permitted to complete internal processor operations and query the UART buffer when the data is desired, rather than yielding the current operation to service the communication. The following discussion explains in detail the concepts used in UART communication and the differences between this communication protocol and I2C.

The two lines used for communication are the TX transmit and the RX receive data lines. Unlike I2C, a clock line is not required. Within each device's UART exists a clock generator to keep the proper timing baud rate for reads and writes to the signal lines.

Typical baud rates are 9600 bits per second but are capable of reaching up to 115200 bits per second if such speeds are required for the application.

UART data is transmitted in the form of packets between the transmitter and receiver. Similar to the I2C method, a 1-bit start signal is used to signify the beginning of transmission by the transmitting device pulling the TX line low during a clock cycle. Following the start bit, the data bits are then transferred, typically 8 bits followed by a parity bit. Following the parity bit, a stop bit sequence of at least two clock cycles is used to signify the end of transmission.

Parity bits are used by the receiver to check for errors during transmission that may have been incorrectly changed along the path from the transmitter to receiver. UART utilizes a parity bit based on the count of bits that were set to high. The transmitting device is aware of the count of high bits in the data frame it is transmitting and sets the parity bit to 0 if there is an even number of 1 bit and sets the parity bit to 1 if there is an odd number of 1 bit. The receiver compares the parity bit to the information it received and if there is a mismatch between the data, it can conclude that a bit must have been corrupted during transmission. The system will then handle the error, possibly by requesting a retransmission.

7.2 Mobile Application

This is an overview of the required elements for the software design. The following will explain the tools used for creating the application, and it will cover the target devices. The application gives the user the most control of the Sunshade unit. This requires us to broaden the scope to a multiplatform approach.

7.2.1 React Native

For our mobile application we choose to develop using React Native to create a cross platform application. The goal of the project is to create smart blinds that are compatible with most windows and an application available for most smartphones. Most mobile applications are developed using object-oriented language, but due to time constraints and the desire to learn a new programming language we decided to use a cross platform language to make the application available for both iOS and Android. We believe with the time remaining to create to do this project there is not enough time to develop an application in two different programming languages optimized for their respective platform.

React Native is based on Facebook's JavaScript library, React, for creating UI on web browsers. React Native is so popular because even though it is written for cross platform, the application feels native due to the JavaScript library used. React Native applications just like its web version React is written using a mixture of JavaScript and JSX. For the native feel, the React Native rendering APIs use Objective-C for iOS or Java for Android.

7.2.2 iOS/Android Design

In order to give the best user experience, automated devices should be controlled from anywhere in the world. The Smart Blinds mobile application gives the user that experience by giving the user control of their blinds with a touch of a button. The mobile application will provide user customization and the ability to monitor the room the blinds are in.

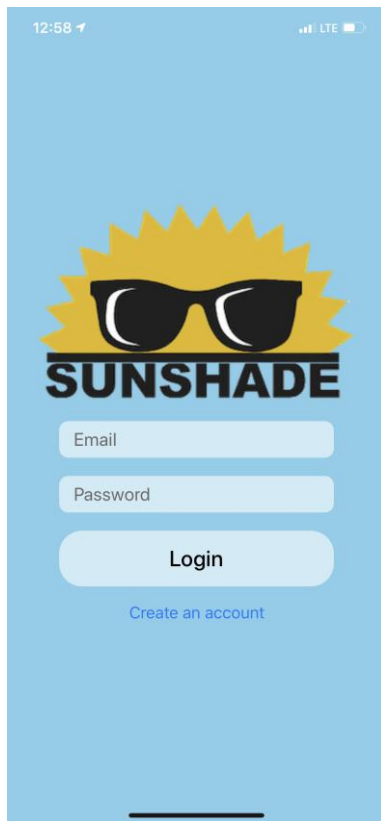


Figure 7.2.2.1 - Launch Screen

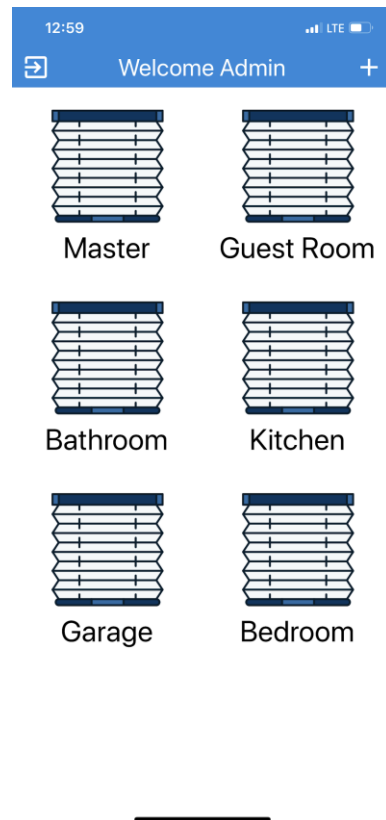


Figure 7.2.2.2 - Dashboard Screen

On launch screen of the application, shown in Figure 7.2.2.1, it requires the user to sign into their account using their email address and desired password. Smart Blinds requires an account to store scheduling information, accessibility away from home, values from the sensors, and controlling multiple blinds. The account registration screen, shown in Figure 7.2.2.4, will require the user to enter their email address, password, and zip code. The information button at the top will show the production team.

The dashboard upon signing in, shown in Figure 7.2.2.2, is a homepage where you see all the blinds connected to your account. To add a new Smart Blinds the user will click the plus icon and the screen will show a pop up. The pop up will prompt the user to enter the blind ID associated with the Smart Blinds. If the blinds are not connected to another user's account, the blinds will connect to the user and he/she will be able to monitor and control them from the dashboard. The device list provides the user will a list of all available Smart Blinds connected to their account that they can control. Clicking any

of the blinds will send them to the blind screen showing the readings from the sensors, positioning, and the scheduling of the blinds.

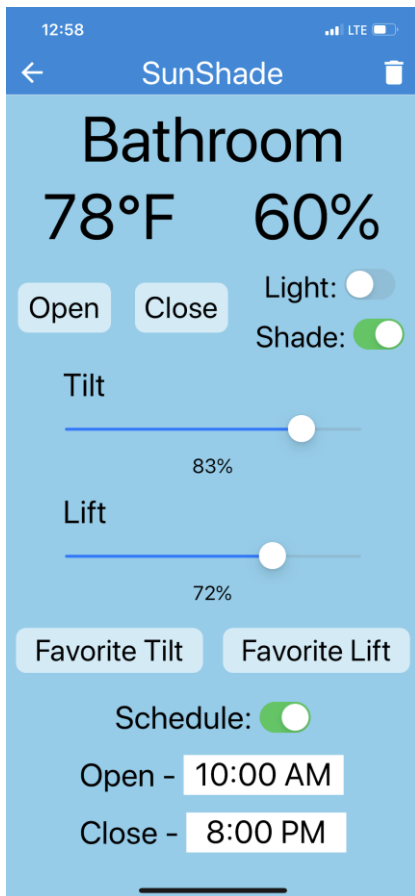


Figure 7.2.2.3 - Blind Screen

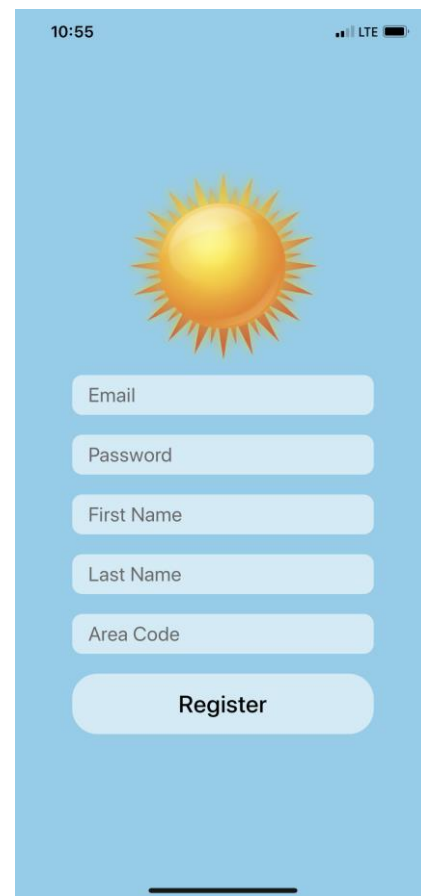


Figure 7.2.2.4 - Register Screen

The blind screen, shown in Figure 7.2.2.3, displays the temperature, light, and current status of the blinds. When the open button is pressed the blinds will open the user's favorite position. The close button will shut the blinds to the position where no light is coming in. The light and shade can be toggled through the switches. The two sliders allow the user to adjust the tilt and how raised the blinds are. The user can set their favorite position by adjusting the sliders to the position and pressing the favorite button. The scheduling feature of the blinds is done by the user entering a time for the blinds to open and close. The feature can be toggled and will be unavailable until turned on again. When the real-world time matches the user inputted open time, the blinds will open to the users favorite position. When the real-world time matches the user inputted closing time, the blinds will shut all the way. The trash bin icon on the top right allows the user to delete the blind from their account.

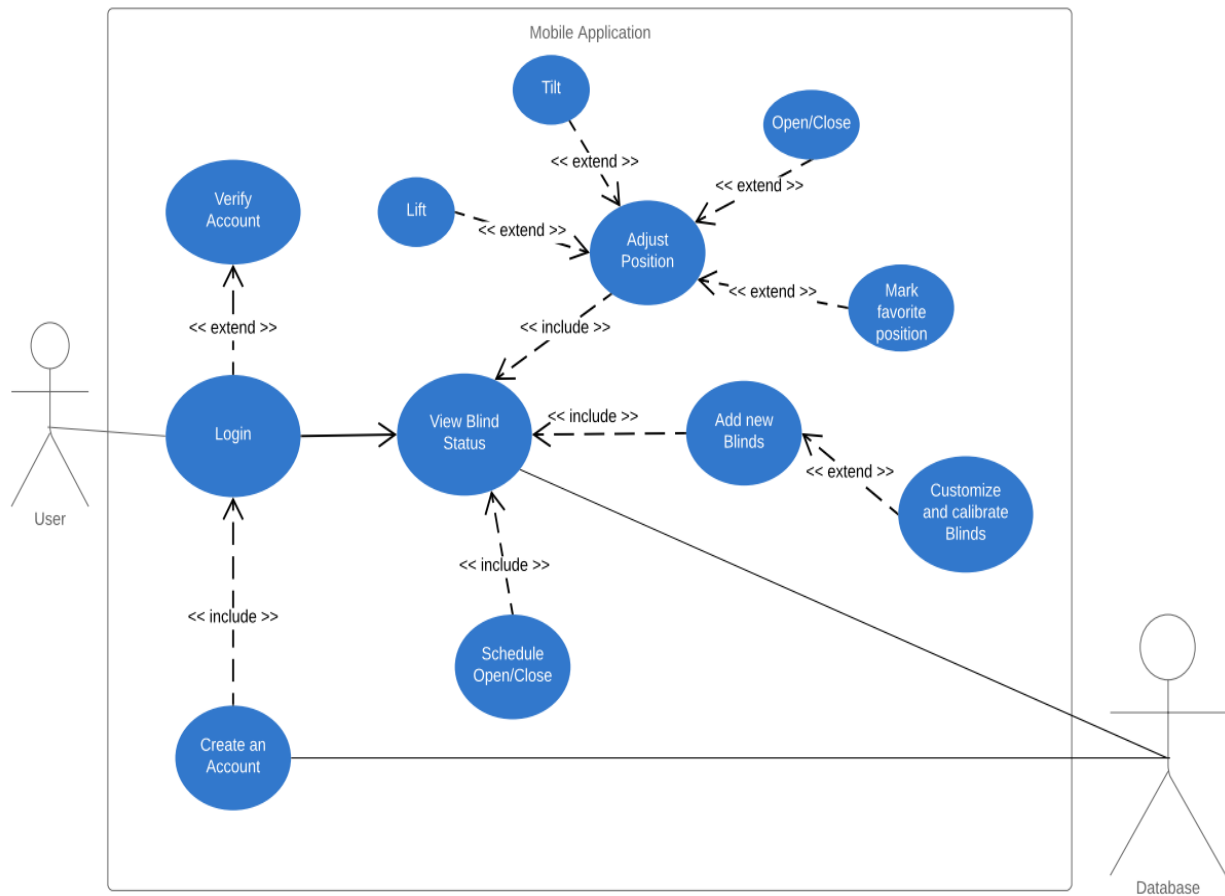


Figure 7.2.2.5 - Use Case Diagram

The use case diagram, shown in Figure 7.2.2.5, displays how the user will interact with the mobile application. The user must log in to access the features of the application and control the blinds. If the user does not have an account, they must create an account to be added to the database and log in. Without this account anyone could control the blinds via Bluetooth or WIFI if they obtain the IP address of the blinds or are close enough. After login the user can control the blinds, add new blinds, and schedule the positions of the blinds through the day. Any changes made on the application to the blinds are sent to the database and sent to the blinds to adjust.

7.3 Database

One of the main features of Smart Blinds is the monitoring of the conditions of the room they are in and sending this data to the user. With the data from the sensors the user can adjust the blinds remotely to improve conditions. To accomplish this a database is

required to record this data from the sensors, send commands, and store user information and settings.

An online database is a system used to store, manipulate, and update various forms of data. While databases can be implemented using different technologies, they typically differ based on how they store data. The most known organizational model used today is the Relational Database model. The Relational Model stores data in a collection of tables filled with entries. Like objects in Object Oriented Programming, each entry is an abstract representation of an object, event, or person relevant to the content of the database.

A key component of a relational database is the relationships between tables. A relation is created by referencing a column (attribute) in one table with a designated column in a different table. Relationships between tables are categorized by the type of association exists between related records in each table. For example, a single record in the table may be associated with multiple records in another table. This is a one-to-many relationship. There are also one-to-one, many-to-one, and many-to-many relationships.

A less common database model is the document-oriented database (also known as a document store). In a document store, data and information describing the data (attributes) reside in key-value pairs within an entity known as a document. Each of these documents are in turn stored in the database. This allows for information to be grouped logically and retrieved based on common attributes. In recent years, the document-oriented model has gained popularity, but is still less commonly used than the relational model.

7.3.1 Amazon Web Services

To create a relational database to store database we will be using Amazon Web Services to store data and create an endpoint to process requests from the mobile application and Smart Blinds. Amazon Web Services offers everything needed to create seamless connection between the user and the Smart Blinds without the need for a local server. To create this connection, we will using Amazon Relational Database Service (RDS) and Amazon Lambda.

Amazon Relational Database Service allows developers to set up and operate a relational database through the cloud. Amazon RDS is so popular because it is compatible with the different database engines including PostgreSQL, MySQL, MariaDB, etc. RDS automatically backs up database instances and implements replication of the database to ensure high availability. We will be using a MySQL database with a one to many architectures to create a relationship between the user account and the one or many Smart Blinds they have. The database will contain two types of entries, the User and the Blinds. The user entry will contain their email, password, area code, and unique personal ID. The blinds entry will contain a unique blind ID, the nickname of the blinds, temperature and humidity from the sensor, current tilt and lift position, whether scheduling is on, time for blinds to open, time for blinds to close, the user favorite lift and tilt positions, and the unique user ID that can control the blind.

Amazon Lambda is a cloud service that reads the actions of an application and responds to these events using functions. The functions act based on the information received and can do anything from resizing an image to upload data to a database. These functions run automatically and require no administration. Amazon Lambda creates a serverless experience by handling all the data between applications and databases. We will be using Amazon Lambda to respond to the requests from the Smart Blinds and the mobile application. The mobile application and Smart Blinds will be making requests to either retrieve data or update data in the database. Amazon Lambda will serve as a middleman between the two endpoints to avoid creating a hosted server. Amazon Lambda functions can be written in JavaScript, Python, Java, C#, Go, and Ruby.

7.3.2 RESTful API

A RESTful web architecture, comprised of web services, uses URL endpoints to determine the actions a server will take when a route is called. When a web service is called, the server will execute any functions and/or serve up any web pages that are defined for that endpoint. RESTful API breakdowns requests into smaller modules which provides developers with more flexibility. RESTful API takes advantage of HTTP methodologies by using GET to retrieve a resource, PUT to change or update a resource, POST to create a new resource, and DELETE to remove a resource. REST is well known in cloud applications because the requests are stateless and require no prior knowledge from the connection. In the case of Smart Blinds, we will be using RESTful API to send JSON objects between clients and the database.

The transmission of data for any web-based application will require a standard method of moving data between endpoints. To accomplish this, document-oriented databases use JavaScript Object Notation (JSON) for this purpose, and as a means of storing and interacting with data. This is a text-based data interchange format 50 derived from the ECMAScript programming language that defines a set of rules for the representation of structured data. Its internet media type is “application/json”.

Today JSON has become one of the most popular data formats for web applications. Its primary usage is to store or move objects of data within applications and between browsers and servers. It is language-independent, meaning it can be used across a wide variety of programming languages.

The JSON format has two underlying data structures that are universally recognized across modern programming languages. The first is a group of key-value pairs, which can be represented in a variety of forms across different languages such as a struct in C/C++, objects in Object-Oriented languages like Java and Ruby, or as a dictionary, hash map or associative array.

7.4 Voice Control

The project goal is to integrate smart home voice commands using an Amazon Alexa. The logic board will have WIFI integration, so the communication will be done through the home router. In order for the communication to detect the Arduino, open source software will be used to register the device as compatible. Many do it yourself projects use Alexa for its many resources and ease to set up. The app that comes with Alexa handles all the custom voice commands by creating routines. The routines can control a grouping of devices or just the sole device. This can be used to make efficiency routines. For example, there can be a routine for energy saving that will keep blinds down and closed. Another option can be a full brightness setting that will open all blinds. There are many variations that can be added to the project. Table 7.4.T1 will include the all target voice commands for the project. If the design allows for extra time for configuration, then there will be added voice packages to the design. All voice commands listed will be simple commands for ease of use and familiarity. The app included with the project will allow user full precise access to the functionality of each device within its metrics. The voice commands are custom to the customer through the Alexa itself so they can be altered at any time.

List of Target Alexa Commands

Device	Command	Action
Diffuser Shade(12v DC motor)	“Alexa (location), shades up”	Shade fully retracted
“ “	“Alexa, (location) shades down”	Shade fully down
Blinds Lift (Cont. rot. servo)	“Alexa, (location) blinds up”	Blinds fully up
“ “	“Alexa, (location) blinds 50%”	Blinds halfway down
“ “	“Alexa, (location) blinds down”	Blinds fully down
Blinds Tilt (180 degree servo)	“Alexa, (location) tilt open”	Tilt at to open full lighting
“ “	“Alexa, (location) tilt close”	Tilt down to block light
LED control	“Alexa, (location) blinds LED on”	Turns on last used lighting mode
“ “	“Alexa, (location) blinds LED off”	Turns off LEDs
Temp and Humidity sensor (DHT22)	“Alexa, (location) blinds comfort readout”	Tells the temperature and humidity of room through speaker

Table 7.4.1 - Voice Command Table

7.5 Software Design Summary

The software involved with Smart Blinds are designed to make the remote use of the blinds seamless and efficient. The mobile application is designed to give an easy and simple user experience, but at the same time looks appealing to the user. The mobile application will be cross platform to be available to the majority of mobile users. Amazon Lambda will serve as a middleman between the database, mobile user, and Smart Blinds. It will handle all the requests sent to the endpoint and will create an appropriate response based on the type of request and its data. The response created from Amazon Lambda will be data from within the database. The database will contain tables for each user and each Smart Blinds with a one to many relationships. The tables contain information about the user and the blinds connected their account. The blind tables contain all the data collected from the sensors on the Smart Blinds and its current status. To access these tables the Smart blinds, send requests to receive data and to send data collected from its sensors.

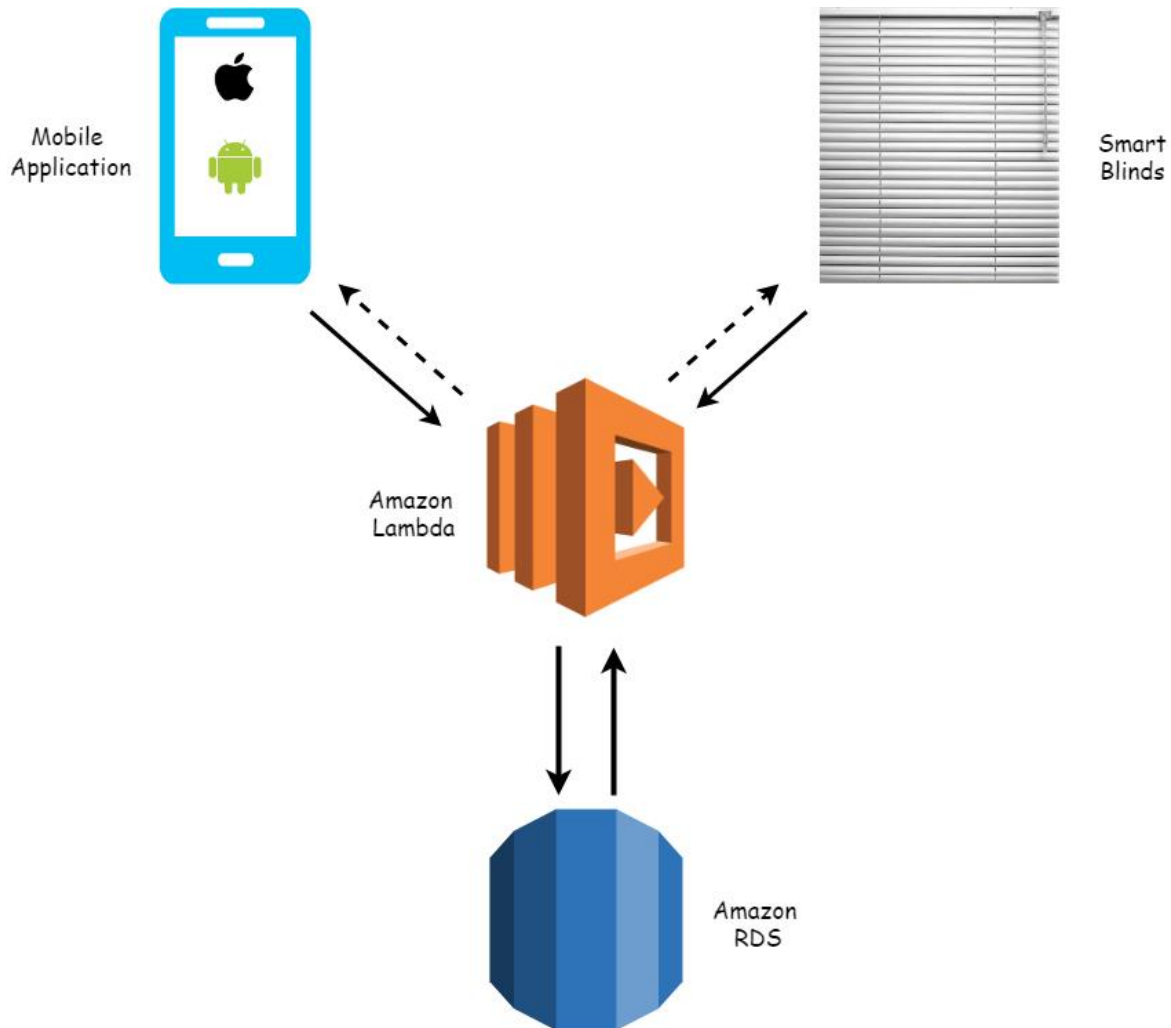


Figure 7.5.1 - Software Design

8.0 Prototype Construction

Prototype construction is the preliminary design of the project. The following sections will break down the prototyping plans into digestible sections. The Sunshade team has already tested a few components.

8.1 Parts Acquisition

There are a variety of websites online that offer good prices and reviews about parts, we were trying our best to have cheap cost but good performance, we know that is hard to happen however we are going to navigate online , there are websites like: Adafruit, Mouser , eBay and Amazon. the table below shows where we are planning to buy some of the parts and their cost, it's an initial process because afterwards we are going to have our bill of material.

Mock Bill of Materials List

Parts	Quantity	Unit Cost	Total Cost	Vendor	Acquired
Blinds	1		\$33.88	Amazon	Yes
Servo Motor	1	\$11.55	\$11.55	Banggood	No
DC Motor	1	\$5.99	\$5.99	Amazon	No
MCU	1	\$6.95	\$6.95	Jacobs parts	Yes
LDR	20	\$4.99	\$4.99	Aliexpress	No
Solar panel	1	\$16.99	\$16.99	Amazon	No
Temperature sensor	5	\$13.47	\$13.47	ebay	No
Frame	1	\$5	\$5	Home depot	Yes
LED	10	\$5.99	\$5.99	ebay	No
Battery	2	\$2.95	\$2.95	adafruit	No
SPST Switches	2	\$0.20	\$0.20	BG Micro	No

Battery Holder	4	\$0.79	\$0.79	BG Micro	No
Pin Dip Sockets	2	\$0.08	\$0.08	BG Micro	NO

Table 8.1.1 - Parts Acquisition

8.2 PCB Layout and Assembly

The PCB layout is straightforward, we used the Eagle software to come up with our PCB layout, we are familiar with the Eagle software since some of us had to take the Junior design class which was very helpful in this matter. we are going to design using libraries and we will give them names, each specific symbol will have a name attached to it. The second phase, the components are traced and connected, the good thing about the Eagle software is that whenever there is a mistake, the software can notify you.

The table below shows the parts we used on the PCB, the list is subject to change in case we want to add anything else or remove any unnecessary parts. these parts have symbols and they are on the libraries:

PCB Component List

Component	Symbol	Quantity	value
24Mhz Crystal	Y1	1	32Mhz
Battery Controller	b1	1	N/A
Capacitor	c1	1	1µf
Capacitor	c2	1	1µf
Capacitor	c3	1	1µf
Capacitor	c4	1	1µf
MCP1702	R1	1	12V to 3V
MCP1702	R2	1	12V to 5V
USB Jack	J1	1	N/A
CC2640	M1	1	N/A

Motor1	N1	1	N/A
Motor2	N2	1	N/A

Table 8.2.1 - Parts list for the PCB

As of right now we haven't decided which vendor we are going to be committed with, there are a lot of vendors but some of them don't give an estimate on how much it will cost per one PCB, and the other vendors that give the prices up front some of them have longer processing time, it may go up to two or three weeks in some cases. Since we did not have too much time, and we chose to go with the vendor JLC PCB to create our PCB faster and has a reliable work production. here are some of the other vendors that we considered as well:

PcbWay is a manufacturer that gives a 20% discount if use the shipping with HDL, they have a very reasonable time and delivery, and what I found out is if I register you may get up to 80% discount on accessories.

OSH Park, no expensive at all, very cheap means reliability and performance is low. but with the cheap price, I think we expect how their work would be , their delivery time is about 2 weeks.

Olimex PCB Prototype Company is another vendor that is in Bulgaria, their have good reviews on the quality and service, but the prices are much higher than the others I found.

4PCB prototype company, you actually get a 50% off on your first order and they have a good reputation among engineers, they are a full service manufacturer and proudly made in the USA. Some of the problems we may encounter during the step of building the PCB are:

- Internal Noise: can be created between components, parts with high frequency affects other parts with low frequency. Not having filters can also make a tremendous difference.
- External Noise: Interference Noise could be coming from outside the PCB, through may be some of the inputs or by electromagnetic induction, for example, if an electric motor works beside the design, which we will have the stepper motor and the DC motor in our case, usually if there is a high voltage involved we may have breakdowns in the digital parts of the PCF (MCU for example).

Vender Characteristics

Vendors	Prices per layer	Delivery Time
4PCB	50% after first registration, quote online	1 week, pay extra \$15 to have shipped overnight

Olimex	Inexpensive,	3-5 working days, solder mask and component print
OSHPARK	\$5 per square inch, per set of three	1 Week approximately
PCBWAY	Approximately about \$30	3-5 days

Table 8.2.2 - Different Vendors characteristics

8.3 MCU Programming

The final coding plan of the Arduino MCU will involve the integration of all the components and peripherals on the PCB. The Arduino MCU will be continuously checking its sensors and sending requests through its WIFI module. These requests will either send information to the database or receive data to update the servo motors.

Upon powering up the Arduino MCU will include the necessary libraries and define the pins for the temperature/humidity sensor, servo motors, and light. These libraries contain predefined functions to use the servo motors, WIFI, real time clock, and I2C communication. After the variables that will be used consistently are defined with the appropriate variable type. The sensors will require a temperature and humidity float variable to store the readings. The servo motors require integer variables because they are bounded between 0 and 180. The real time clock has its own variable type from the library included. To use the sensors the I2C module is turned on using the typical 9600 baud rate. The last part of setup is using the WIFI module to connect to the internet. The Arduino MCU will not begin the loop function until it is connected to the internet.

In the loop function, the Arduino MCU begins by sending a GET request to the database to receive the stored position of the servo motors. If the stored position is different from the current position already stored on the Arduino MCU then that means the user using the mobile application requested a change. If a change has been made then the servo motors will update their positions, if not the loop will continue. Next in the loop the Arduino MCU will read its sensors and compare them to their last readings. If the current readings and the new readings are different a POST request will be sent containing the new readings to update the database. From the database the readings will be sent to the mobile application when they request the information. If the new readings are the same as the last readings, the loop will continue. Last, if the user has a schedule on the Arduino MCU checks the real time clock and compares to the schedule. If the time is in between the scheduled time the loop waits 5 seconds and starts over. If the time is outside of the schedule, the Arduino MCU enters low power mode until the time is in between the schedule.

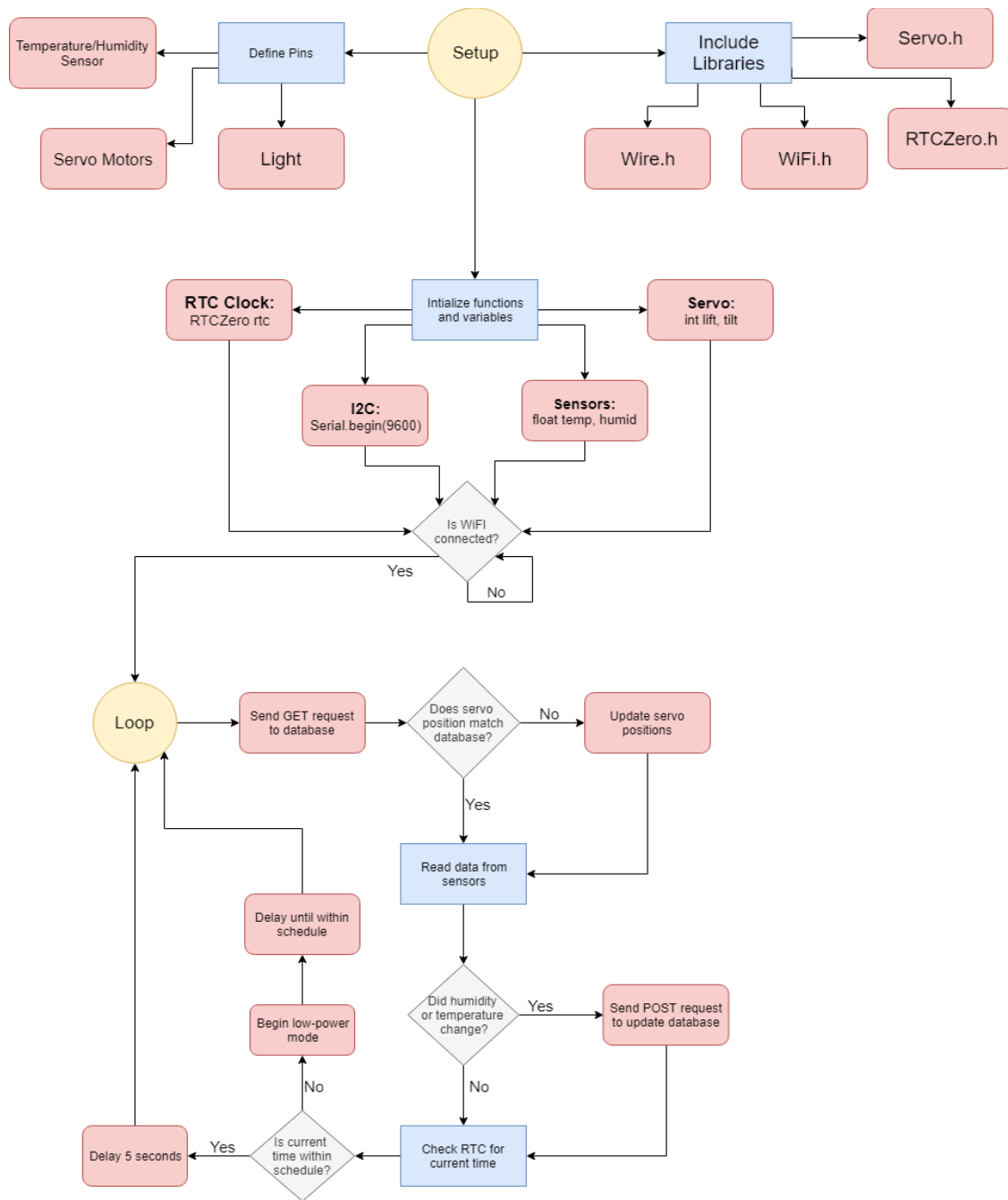


Figure 8.3.1 - Arduino MCU Flow

8.4 iOS/Android App Development

The mobile application is the main tool for controlling and monitoring the Smart Blinds. The application must be easy to use and seamless to ensure there is no confusion or error while using the application that can potentially break the blinds. To ensure these specifications the final coding plan for the mobile application will be done in three phases.

The first phase will involve the integration of the user interface of the mobile application by creating a seamless user experience. The user experience is critical to the mobile application. All the different user screens will be created and the features that belong to each page. The user interface will be created as designed as proposed with a few additional features. Using React Native the creation of the user interface will look the same on both Android and iOS devices. React Native helps save time by being able to develop cross platform instead of in two different languages.

The second phase will involve working on the backend of the mobile application. After the user interface is working without any error, the backend API will be created to communicate to the database. This involves login authentication on the startup screen and pulling data from the database to fill the dashboard with information on the blinds. Using RESTful API and React Native, the mobile application will be sending requests to the database endpoint and receive a response based on the request. These response and requests will be in the form of JSON filled with data on the blind or the user. The user should be able to click any button and receive the correct response from the database.

The third phase will involve testing and polishing the user interface of the mobile application. Once the mobile application works smoothly, we will begin looking for minor bugs and reworking the design to be more appealing to the user. This step is done last because we would like to have a working mobile application without any major bugs before improving the user experience. The minor bugs we will be looking for should not break the mobile application or cause any major error, but instead should be spelling errors, wrong numbers, etc. This phase will continue until the mobile application is error free and meets the specifications of the design. If any other major issue arises in any other section of construction this phase will be halted until the issue is fixed.

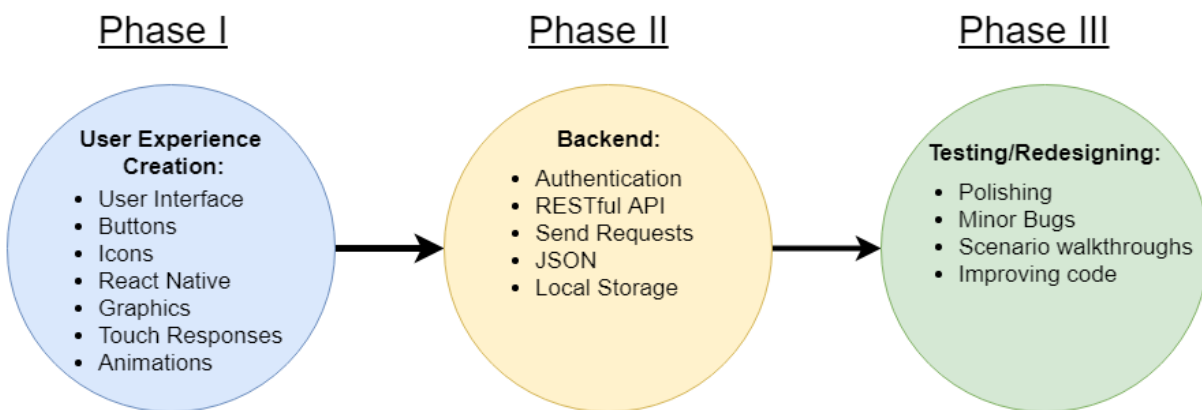


Figure 8.4.1 - Mobile Application Development Flow

8.5 Database Creation

The database will be created using Amazon Relational Database Services to create a MySQL database. The database will have two different tables, the user and the blinds. The user tables will have a one to many relationships with the blind's tables. The blind tables will contain the users ID that can control and monitor the statuses and positions of the blinds. The database will be accessed through Amazon Lambda, a serverless cloud service. Amazon Lambda will handle all the requests sent from the mobile application and the Smart Blinds. Every request will be handled appropriately, and the respective device will receive the correct data pertaining to the type of request they sent.

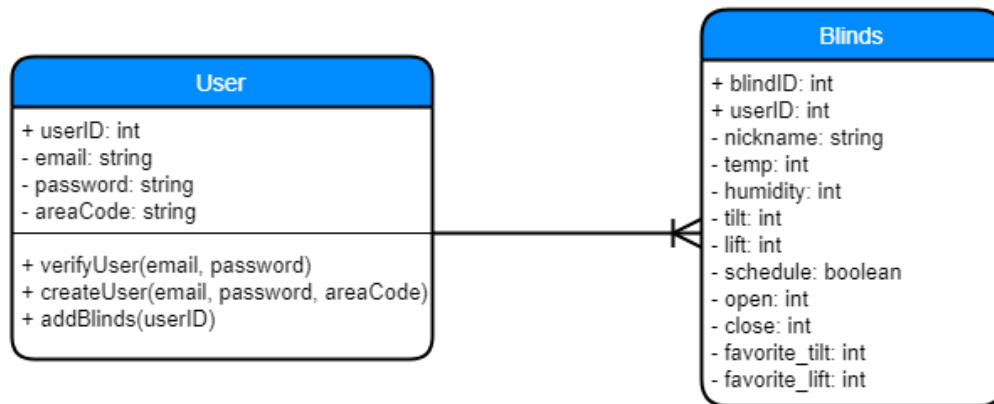


Figure 8.5.1 - Database ERD Diagram

9.0 Prototype Testing

Upon completion of the Sunshade project, the most important element of the creation procedure will become imperative: The testing phase. This will be done to absolutely ensure that all aspects and elements of the finished Sunshade prototype and finished product are in working order, and ready for presentation with no major issues. If even one part or element were to fail, the project could not operate, and could even lead to a domino effect that could cause other systems to fail as well, possibly leading to redesigns or part replacements, which is a significant setback. Because of this, the testing phase must be precise, and yield the best possible results. This will include a hardware and software test environment, as well as hardware and software specific testing.

9.1 Hardware Test Environments

For the environment in which the SunShade project will reside, a model window frame was constructed from wood, and the prototype was mounted, as discussed earlier. This is to allow for mobile demonstrations of functionality, and an accurate test of the motors pulling the weight of the blinds, to remove any inconsistencies in performance. Once the proper materials arrive, the SunShade project team hopes to perform environment specific testing for different levels of external brightness from the sun, and how it affects the brightness-control feature, ideally, closing the blinds with the tilt feature if brightness levels are too high, and opening if too low, until a certain time. Different times of the day, and therefore the cycle of the sun, will affect the smart-blinds system differently, and therefore, requires testing to ensure performance.

Another environmental testing procedure the team hopes to complete, is a full-scale analysis of solar-powered charge potential given the varying types of glass, as the solar-panels will be mounted to the inside of the window, facing outwards towards the sun. The team can test with no glass panel, standard panel, tinted, or even double-paned. From there, a multimeter can be used to measure the voltage and current being supplied to the battery from the solar panels under varying conditions. The only foreseeable margin of error for this testing is that light quality is susceptible to over shade, cloud cover, weather, and time of day in relativity to these variables. One possible solution to this issue is to utilize a high-intensity illuminator in order to simulate sunlight beaming onto the panel with different types of glass-related environments. It is worth exploring if these tests have already been performed by official labs, and their data may be of use for our testing phase with regard to window effect.

9.2 Hardware Specific Testing

Hardware specific testing is terminology used to describe the testing phase to be completed on the performance of the finished device and its functionalities. Its desired outcome is to ensure all hardware systems are running at their optimal performance. Various test situations will be examined, as well as their desired results, environments, actual results, and any possible solutions to errors, as well as re-testing once these have been completed. This section is critical because it provides the SunShade team a chance to test their project in a wide variety of situations in order to determine if the finished appliance performs as it is desired to.

9.2.1 Stopping Criteria

During the testing phase at hand, it is imperative that all appropriate testing procedures be performed continuously, and without stopping, unless a significant error arises that may hinder or halt further testing procedures or may result in incomplete data or skewed results. This criteria for the event of halting testing could be any number of things, from software problems, malfunction, cosmetic, or functional. If the criteria is met, the first step is to consult with all appropriate developers for the specific issue to ensure the complexity of the necessary fix or alteration, and if it is a simple fix, it must be completed at once so that continuous testing may proceed as planned.

After all documented test cases, as listed in section 9.2.2, are completed and meet the proper standards of passing each test, the final project will be considered deliverable, meaning that it functions without any bugs, malfunctions, or general issues to the best of the SunShade project team’s knowledge, effectively completing the project for presentation.

9.2.2 Hardware Test Cases

In the following section, the test cases for the SunShade smart-blinds appliance development will be presented with their individual objectives, descriptions, conditions, and desired results, allowing for an easy to follow format for each possible testing procedure. They will be presented as shown:

Example Hardware Test Case

Test Objective: What are the goals for this specific test case?
Test Description: What is the procedure for testing this case?
Conditions: Any and all conditions relevant to the test case must be defined.
Desired Results: What <i>should</i> happen as seen by the hypothetical user?

Table 9.2.1 - Example hardware Test Case

Table 9.2.2 through 9.2. are set in place to follow the format of table 9.2.1, to accurately describe each of the various test cases which are to be explored by the Sunshade team, likely in October of 2019, once primary hardware development is completed, in order to ensure complete hardware functionality.

Hardware Test Case 1 - Power on

Test Objective: SunShade powers on
Test Description: Ensure that the solar energy collected can sufficiently turn on the device.
Conditions: Standard, see section 9.1: Hardware test environment
Desired Results: SunShade turns on, illuminating LED and powering board.

Table 9.2.2 - Power on Test Case

Hardware Test Case 2 - Tilting Blinds Up and Down

Test Objective: Ensure that the motor is capable of tilting blinds open and closed.
--

Test Description: Use the mobile application to command the blinds to tilt open and closed.

Use the manual control buttons to command the blinds to tilt open and closed.

Conditions: Standard, see section 9.1: Hardware test environment

Desired Results: The tilt motor should be able to turn tilt rod with ease, on command, allowing for the limiting or increasing of sunlight.

Table 9.2.3 - Tilting Blinds Up and Down Test Case

Hardware Test Case 3 - Manual Control

Test Objective: Ensure that the manual control buttons function.

Test Description: Use the manual control buttons to command the blinds to tilt open and closed, as well as lift up and down.

Conditions: Standard, see section 9.1: Hardware test environment

Desired Results: All motors should respond accordingly to manual activation.

Table 9.2.4 - Manual Control Test Case

Hardware Test Case 4 - Overtilt

Test Objective: Ensure motor does not over-turn the tilt.

Test Description: Use the mobile application to command the blinds to tilt open and closed.

Use the manual control buttons to command the blinds to tilt open and closed.

Conditions: Standard, see section 9.1: Hardware test environment

Desired Results: The tilt motor should not turn the tilt-rod too far clockwise or counterclockwise, as this would result in physical damage to the system.

Table 9.2.5 - Overtilt Test Case

Hardware Test Case 5 - Retracting Shades

Test Objective: Ensure that the motor is capable of lifting shades.

Test Description: Use the mobile application to command the shades to lift up, only after setting blinds to fully open tilt.

Use the manual control buttons to command the shades to lift up, only after setting blinds to fully open tilt.

Conditions: Standard, see section 9.1: Hardware test environment

Desired Results: The lift motor should be able to retract the shades with ease, while not damaging the system.

Table 9.2.6 - Retracting Shades Test Case

Hardware Test Case 6 - Lowering Shades

Test Objective: Ensure that the motor is capable of lowering shades down.

Test Description: Use the mobile application to command the shades to lower.
Use the manual control buttons to command the shades to lower.

Conditions: Standard, see section 9.1: Hardware test environment

Desired Results: The lift motor should be able to lower the shades with ease, while not damaging the system.

Table 9.2.7 - Lowering Shades Test Case

Hardware Test Case 7 - Brightness Control

Test Objective: Ensure that the blinds react accordingly to brightness level.

Test Description: Shine a bright light at the appliance.

Conditions: Standard, see section 9.1: Hardware test environment

Desired Results: The blinds should tilt closed at the response of brightness level increasing

Table 9.2.8 - Brightness Control Test Case

Hardware Test Case 7 - Brightness Control Pt.2

Test Objective: Ensure that the blinds react accordingly to lower brightness level.

Test Description: Shine light away from appliance, but not off.

Conditions: Standard, see section 9.1: Hardware test environment

Desired Results: The blinds should tilt open at the response of brightness level decreasing, and then close at a certain level of darkness.

Table 9.2.9 - Brightness Control Pt.2 Test Case

Hardware Test Case 8 - Retracting Diffuser

Test Objective: Ensure that the motor is capable of rolling up diffuser shade

Test Description: Use the mobile application to command the diffuser to roll up.
Use the manual control buttons to command the diffuser to roll up.

Conditions: Standard, see section 9.1: Hardware test environment
Desired Results: The lift motor should be able to retract the diffuser with ease, while not damaging the system.

Table 9.2.10 - Retracting Diffuser Test Case

Hardware Test Case 6 - Lowering Diffuser

Test Objective: Ensure that the motor is capable of rolling down diffuser shade
Test Description: Use the mobile application to command the diffuser to come down. Use the manual control buttons to command the diffuser to come down.
Conditions: Standard, see section 9.1: Hardware test environment
Desired Results: The lift motor should be able to lower the diffuser with ease, while not damaging the system.

Table 9.2.11 - Lowering Diffuser Test Case

9.2.3 Battery & Controller Tests

In order to determine if the solar panel and its power system is effectively charging the battery with the controller, it will be necessary to leave the panel in the sun to charge for an extended period of time, perhaps a day, while it is connected to the battery controller, so that it may charge the battery to its full charge. This can be done with a standard voltmeter to check that power is in fact flowing into the battery at the command of the battery controller.

9.2.4 Motor Tests

The primary step for testing motor functionality is the test the motor by itself, directly. This will be done by applying voltage directly to the DC motor, and will spin either clockwise or counterclockwise, and then reverse when the voltage is reversed. This will continue until power is disconnected. Once we verify that the motor functions both clockwise and counterclockwise, we have ensured the functionality of the motor, and may proceed.

9.2.5 Voltage Regulation Tests

The voltage regulator's operation is of critical importance to the system, and so in order to test it, once the motor, battery, and controller are tested and confirmed to be in operating condition, the SunShade team is able to go throughout the schematic and test for proper voltages at each and every point, making sure that the voltage in the design

meets the same voltages in the physical version. Checking this will let the SunShade team know that the correct voltages are being applied to each node and component.

9.2.6 Microprocessor Tests

Testing for the microprocessor is the most extensive phase of the SunShade project testing and is fairly complex. First and foremost, pin must be insured to be going in and out of the proper terminals, and functioning correctly, exactly as desired. This can be performed by commanding the processor to assign a high or low to a specific pin, which can then be measured using a standard multimeter, at the pin in question, confirming that communication is occurring between the pins. Improper communication can have results such as incorrect responses or even damage to the PCB itself, and so must be avoided.

Once proper pin configuration is ensured, the motor pins may be tested, by applying highs and lows on the targeted pins. This will determine if assigning a high to the proper pin will enable the motor to rotate clockwise or counterclockwise, from there, it would be necessary to check the second motor, which would tilt the blinds open or closed, or lift them up or down. There will also be a limit set to each of the motors, telling them when to stop, using an encoder. If the tilt function turns too far, it could damage the blinds, or completely change the desired effect when lifting is performed, causing further damage. The retraction and lowering features must be tested for the same goal, to ensure that the microprocessor tells the motor when to stop.

The Light Dependent Resistor, or, the LDR, will also have a parameter to work within, and so, by the use of a simple flashlight, testing for this component and making sure that it communicates with the microprocessor will be straight forward. Shining the light directly at the LDR to exceed its bounds, and then shining away, will demonstrate a self-correcting system.

Finally, the LED system may be tested, as it also communicates with the microprocessor. The SunShade team must verify that it is able to turn the LED strip on and off without error. Should the LEDs function with no problems, then the microprocessor will have been effectively and completely tested, and ready to be a part of a deliverable final project.

9.2.7 Final Project Testing

The final phase of the Sunshade project testing is only done once all other individual systems and components have been properly tested. The SunShade smart-blinds appliance will be affixed to the prepared wooden window stand/housing, and placed in front of an open sun-facing window, or in a garage. If the size of the SunShade project is compatible with a typical window U-bracket, then it will be affixed to a standard window in a team member's home. The blinds will sit in this position for a full day, or a sun cycle, and the movements will be recorded. From this, the team can also learn about what kind of power is required to function throughout a single day, and if the project is in fact energy efficient, and to what degree. This will take into account both brightness-correction and schedule usage. At the end of the day, the remainder of the

charge will be measured from the battery. The team will then allow the LED brightness control system to activate, draining even more of the leftover power, so that an accurate assessment can be made about maximum and standard energy usage.

This kind of testing may be performed over multiple days in order to get a better understanding of what can be expected of the SunShade smart-blinds appliance, and to see if any improvements are to be made. This will also allow for analysis of how weather conditions might influence the effectiveness of the product, or if the lack of required power output on darker days will balance out the lesser charge gained through said days, and so on.

9.3 Software Test Environment

Software testing will be conducted during and after hardware testing to verify the architecture is stable and able to interact with the other hardware. Most software tests will be done on the computer using Arduino IDE and Visual Studio Code before being transferred to their respective hardware. The software to be tested will be used on the mobile application, the Arduino MCU, and the database.

9.4 Software Specific Testing

The following software specific testing will ensure the database, Smart Blinds, and mobile application perform their expected outputs. Completing these tests prove that the software can handle different scenarios without error or complications. The basic tests will be completed first to ensure basic functionality and after more complex tests will be done to test uncommon but possible scenarios.

Software Test Case Example

Test Objective:	What are the goals for this specific test case?
Test Materials:	Any materials used in this test case
Procedure:	Any steps taken relevant to the test case
Desired Results:	What <i>should</i> happen as seen by the hypothetical user?

Table 9.4.1 - Software Test Case Example

9.4.1 Database

The primary purpose of database testing is to ensure data is stored from the blinds and the user and is accessible from both the Arduino MCU and the mobile application. The database is essentially the median between the user and the Smart Blinds therefore

it must be able to be accessed easily and communicate to different devices. The database will be tested for accurate fields and ability to handle different types of requests.

Software Test Case 1 - User tables

Test Objective:	Ensure user table contains correct fields
Test Materials:	<ol style="list-style-type: none"> 1. MySQL Database 2. Visual Studio Code
Procedure:	<ol style="list-style-type: none"> 1. Connect to database 2. Make a request to show user table 3. Disconnect from database
Desired Results:	The database will return: userID - int email - varchar password - varchar areaCode - varchar

Table 9.4.1.1 - User tables Test Case

Software Test Case 2 - Blind tables

Test Objective:	Ensure blinds table contains correct fields
Test Materials:	<ol style="list-style-type: none"> 1. MySQL Database 2. Visual Studio Code
Procedure:	<ol style="list-style-type: none"> 1. Connect to database 2. Make a request to show blinds table 3. Disconnect from database
Desired Results:	The database will return: blindID - int userID - int nickname - varchar temp - int humidity - int tilt - int lift - int schedule - int open - varchar close - varchar

Table 9.4.1.2 - Blinds tables Test Case

Software Test Case 3 - API GET Requests

Test Objective:	Ensure database can handle GET requests
Test Materials:	<ol style="list-style-type: none"> 1. MySQL Database 2. Visual Studio Code
Procedure:	<ol style="list-style-type: none"> 1. Send a GET request to API endpoint
Desired Results:	<p>The database will return a JSON with:</p> <pre> "blindID" : "1" "userID" : "1" "date" : "Current date" "nickname" : "Test Blinds" "temp" : "50" "humidity" : "50" "tilt" : "50" "lift" : "50" "schedule" : "1" "open" : "09:00" "close" : "20:00" </pre>

Table 9.4.1.3 - API GET Test Case

Software Test Case 4 - API POST Requests

Test Objective:	Ensure database can handle POST requests
Test Materials:	<ol style="list-style-type: none"> 1. MySQL Database 2. Visual Studio Code
Procedure:	<ol style="list-style-type: none"> 1. Create a JSON with user information <pre> "email" : "test@testing.ucf" "password" : "test123" "areaCode" : "12345" </pre> 2. Send POST request to database 3. Send GET request to database
Desired Results:	<p>The database will return a JSON with:</p> <pre> "userID" : "1" "email" : "test@testing.ucf" "password" : "test123" "areaCode" : "12345" "date" : "Current date" </pre>

Table 9.4.1.4 - API POST Test Case

9.4.2 Arduino MCU

The primary purpose of the testing is to confirm the Arduino MCU is capable of controlling the servo motors and light, reading the sensors, and communicating with the database. The cases tested are the basic functions of the Arduino MCU that the everyday user will be using in most scenarios.

Software Test Case 5 - Wireless Network

Test Objective:	Ensure Arduino can connect to a wireless network
Test Materials:	<ol style="list-style-type: none">1. Arduino MCU2. Arduino IDE
Procedure:	<ol style="list-style-type: none">1. Load WIFI.h and Serial.h library in Arduino IDE2. Create constant variables with WIFI SSID and password3. Attempt to connect the network using SSID and password4. Print IP and MAC address to console via serial
Desired Results:	Arduino MCU should connect to wireless network and print its assigned IP and MAC address.

Table 9.4.2.1 - Wireless Network Test Case

Software Test Case 6 - WIFI Protected Setup

Test Objective:	Ensure Arduino can connect to a wireless network via the WPS button
Test Materials:	<ol style="list-style-type: none">1. Arduino MCU2. Arduino IDE3. Router
Procedure:	<ol style="list-style-type: none">1. Load WIFI.h and Serial.h library in Arduino IDE2. Create variables with WIFI SSID and password3. Press WPS button on Arduino4. Press WPS button on router5. Print IP and MAC address to console via serial
Desired Results:	Arduino MCU should connect to wireless network and print its assigned IP and MAC address.

Table 9.4.2.2 - WIFI Protected Setup Test Case

Software Test Case 7 - Servo Control

Test Objective:	Ensure Arduino accurately controls servo motor
Test Materials:	<ol style="list-style-type: none"> 1. Arduino MCU 2. Arduino IDE 3. PCB 4. Servo Motor
Procedure:	<ol style="list-style-type: none"> 1. Connect to servo motor to PWM pin on MCU via PCB 2. Load Servo.h and Serial.h library in Arduino IDE 3. Create variable for servo and assign pin 4. Read servo position and map between 0 and 180 5. Write position 0 to servo, delay until move is finished, and print position 6. Write position 90 to servo, delay until move is finished, and print position 7. Write position 180, delay until move is finished, to servo and print position
Desired Results:	Servo should move to 0 position and print 0, then move to 90 position and print 90, and last move to 180 position and print 180.

Table 9.4.2.3 - Servo Control Test Case

Software Test Case 8 - Light Control

Test Objective:	Ensure Arduino is able to control LED lights
Test Materials:	<ol style="list-style-type: none"> 1. Arduino MCU 2. Arduino IDE 3. LED lights
Procedure:	<ol style="list-style-type: none"> 1. Connect light to digital pin on MCU via PCB 2. Assign light pin and change to output 3. Assign pin to HIGH and delay 2 seconds 4. Assign pin to LOW
Desired Results:	The light should turn on for two seconds and then turn off

Table 9.4.2.4 - Light Control Test Case

Software Test Case 9 - Reading Sensors

Test Objective:	Ensure Arduino is able to communicate with sensors and receives accurate readings
------------------------	---

Test Materials:	<ol style="list-style-type: none"> 1. Arduino MCU 2. Arduino IDE 3. PCB 4. Temperature/Humidity Sensor
Procedure:	<ol style="list-style-type: none"> 1. Connect sensor to analog pin on MCU via PCB 2. Load Serial.h library in Arduino IDE 3. Create variable for temperature and humidity 4. Read temperature/humidity sensor 5. Convert readings into appropriate temperature and humidity values 6. Print values to console via serial 7. Check actual temperature and humidity to see if sensors are accurate
Desired Results:	Sensors should read values close to actual temperature and humidity values without error.

Table 9.4.2.5 - Reading Sensors Test Case

Software Test Case 10 - Arduino GET Requests

Test Objective:	Ensure Arduino is able to send GET requests
Test Materials:	<ol style="list-style-type: none"> 1. Arduino MCU 2. Arduino IDE 3. MySQL Database
Procedure:	<ol style="list-style-type: none"> 1. Load WIFI.h and Serial.h library in Arduino IDE 2. Connect to wireless network 3. Make a GET request for the table with the blinds ID to the API endpoint 4. Wait for response from database 5. Print JSON received
Desired Results:	<p>The database will return a JSON with:</p> <pre> "blindID" : "1" "userID" : "1" "date" : "Current date" "nickname" : "Test Blinds" "temp" : "50" "humidity" : "50" "tilt" : "50" "lift" : "50" "schedule" : "1" "open" : "09:00" "close" : "20:00" </pre>

Table 9.4.2.6 - Arduino GET Test Case

Software Test Case 11 - Arduino POST Requests

Test Objective:	Ensure Arduino is able to send POST requests
Test Materials:	<ol style="list-style-type: none"> 1. Arduino MCU 2. Arduino IDE 3. MySQL Database
Procedure:	<ol style="list-style-type: none"> 1. Load WIFI.h and Serial.h library in Arduino IDE 2. Connect to wireless network 3. Create JSON with blind information “blindID” : “1” “userID” : “1” “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “50” “tilt” : “50” “lift” : “50” 4. Make a POST request for the table with the blinds ID to the API endpoint to update the temperature value 5. Check database for updated information
Desired Results:	The database should contain a table with: “blindID” : “1” “userID” : “1” “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “50” “tilt” : “50” “lift” : “50”

Table 9.4.2.7 - Arduino POST Test Case

9.4.3 iOS/Android Application

The primary purpose of the mobile application is to control and monitor the Smart Blinds wirelessly. The mobile application will be controlled by the user so it must be seamless and easy to use by the average user. Every test done involves the average scenario that most users will encounter.

Software Test Case 12 - iOS/Android Application GET Requests

Test Objective:	Ensure iOS/Android Application is able to send GET requests
Test Materials:	<ol style="list-style-type: none"> 1. iOS/Android Application 2. MySQL Database
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application on either platform 2. Enter test blind ID and send GET request for its information 3. Display blind information
Desired Results:	<p>The mobile application should display a JSON with:</p> <pre> "blindID" : "1" "userID" : "1" "date" : "Current date" "nickname" : "Test Blinds" "temp" : "50" "humidity" : "50" "tilt" : "50" "lift" : "50" "schedule" : "1" "open" : "09:00" "close" : "20:00" "favorite_tilt" : "50" "favorite_lift" : "50" </pre>

Table 9.4.3.1 - Mobile GET Test Case

Software Test Case 13 - iOS/Android Application POST Requests

Test Objective:	Ensure iOS/Android Application is able to send POST requests
Test Materials:	<ol style="list-style-type: none"> 1. iOS/Android Application 2. MySQL Database
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application on either platform 2. Enter following information and send POST request to change blinds information <pre> "blindID" : "1" "userID" : "1" "date" : "Current date" "nickname" : "Test Blinds" "temp" : "75" "humidity" : "75" "tilt" : "75" "lift" : "75" "schedule" : "1" </pre>

	<pre> “open” : “09:00” “close” : “20:00” “favorite_tilt” : “50” “favorite_lift” : “50” 3. Check blind tables on database for inputted information </pre>
Desired Results:	<p>The mobile application should display a JSON with:</p> <pre> “blindID” : “1” “userID” : “1” “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “75” “tilt” : “75” “lift” : “75” “schedule” : “1” “open” : “09:00” “close” : “20:00” “favorite_tilt” : “50” “favorite_lift” : “50” </pre>

Table 9.4.3.2 - Mobile POST Test Case

Software Test Case 14 - Creating a user account

Test Objective:	Ensure mobile application allows users to create an account
Test Materials:	<ol style="list-style-type: none"> iOS/Android Application MySQL Database
Procedure:	<ol style="list-style-type: none"> Open mobile application Attempt to create an account using the following user information: Email: test@testing.ucf.edu Password: test123 Area Code: 12345 Check user tables on database for inputted information
Desired Results:	<p>The database should contain a table with:</p> <pre> “userID” : “1” “email” : “test@testing.ucf.edu” “password” : “test123” “areaCode” : “12345” </pre>

Table 9.4.3.3 - User Creation Test Case

Software Test Case 15 - Invalid account creations

Test Objective:	Ensure mobile application does not allow for accounts with the same email
Test Materials:	1. iOS/Android Application
Procedure:	<ol style="list-style-type: none"> 1. Check user table for test account information “userID” : “1” “email” : “test@testing.ucf.edu” “password” : “test123” “areaCode” : “12345” 2. Open mobile application on either platform 3. Attempt to create an account using the same test account information: Email: test@testing.ucf.edu Password: 123test Area Code: 54321
Desired Results:	The mobile application should display an error screen telling the user the email is already used for an account.

Table 9.4.3.4 - Invalid User Creation Test Case

Software Test Case 16 - User log in

Test Objective:	Ensure mobile application allows users to log in to their account
Test Materials:	1. iOS/Android Application
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application 2. Attempt to log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123
Desired Results:	The mobile application should log in the user and change the screen to the dashboard.

Table 9.4.3.5 - User Login Test Case

Software Test Case 17 - User log out

Test Objective:	Ensure mobile application allows users to log out of their account
------------------------	--

Test Materials:	1. iOS/Android Application
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application 2. Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 3. Click the menu button 4. Click log out
Desired Results:	The mobile application should log out the user and return them to the login display

Table 9.4.3.6 - User Logout Test Case

Software Test Case 18 - Add new blinds

Test Objective:	Ensure mobile application allows user to add new blinds to their account
Test Materials:	1. iOS/Android Application
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application 2. Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 3. From dashboard click add new blinds 4. Enter provided blind device ID
Desired Results:	The mobile application should prompt the user is connected to the blinds and return the user to the dashboard of the connected blinds. Within the database, the user ID should be added to the field in the blind table entry.

Table 9.4.3.7 - Add Blinds Test Case

Software Test Case 19 - Dashboard display

Test Objective:	Ensure mobile application displays the user blinds information on the dashboard after login
Test Materials:	1. iOS/Android Application
Procedure:	1. Open mobile application

	<ol style="list-style-type: none"> 2. Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 3. View dashboard
Desired Results:	<p>The mobile application should display the test blinds information:</p> <p>Temperature - 50 Humidity - 50 Current Position - 50 Schedule - On Open - 9:00am Close - 8:00pm</p>

Table 9.4.3.8 - Dashboard Test Case

Software Test Case 20 - Adjust Tilt

Test Objective:	Ensure mobile application can change the tilt of the blind using the slider
Test Materials:	<ol style="list-style-type: none"> 1. iOS/Android Application 2. Database
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application 2. Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 3. View dashboard 4. Adjust tilt slider to 20 percent 5. Check database
Desired Results:	<p>The database should contain a table with:</p> <p>“blindID” : “1” “userID” : “1” “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “50” “tilt” : “20” “lift” : “50”</p>

Table 9.4.3.9 - Tilt Test Case

Software Test Case 21 - Adjust Lift

Test Objective:	Ensure mobile application can change the lift of the blind using the slider
Test Materials:	<ol style="list-style-type: none"> 1. iOS/Android Application 2. Database
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application 2. Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 3. View dashboard 4. Adjust lift slider to 80 percent 5. Check database
Desired Results:	<p>The database should contain a table with:</p> <p>“blindID” : “1” “userID” : “1” “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “50” “tilt” : “20” “lift” : “80”</p>

Table 9.4.3.10 - Lift Test Case

Software Test Case 22 - Favorite Positions

Test Objective:	Ensure mobile application set favorite positions for the blinds
Test Materials:	<ol style="list-style-type: none"> 1. iOS/Android Application 2. Database
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application 2. Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 3. View dashboard 4. Set lift to 75 and tilt to 25 5. Click favorite button
Desired Results:	<p>The mobile application should mark the position with stars and also in the database there should contain a table with</p> <p>“blindID” : “1” “userID” : “1”</p>

	<pre> “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “50” “tilt” : “20” “lift” : “50” “favorite_tilt” : “25” “favorite_lift” : “75” </pre>
--	---

Table 9.4.3.11 - Favorite Positions Test Case

Software Test Case 23 - Open/Close

Test Objective:	Ensure mobile application can open and close blinds using buttons
Test Materials:	<ol style="list-style-type: none"> iOS/Android Application Database
Procedure:	<ol style="list-style-type: none"> Open mobile application Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 View dashboard Click close Check database Click open Check database
Desired Results:	<p>When close is clicked database should contain a table with:</p> <pre> “blindID” : “1” “userID” : “1” “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “50” “tilt” : “0” “lift” : “0” “schedule” : “1” “open” : “12:00” “close” : “21:00” “favorite_tilt” : “25” “favorite_lift” : “75” </pre> <p>When open is clicked database should contain a table with:</p> <pre> “blindID” : “1” “userID” : “1” </pre>

	<pre> “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “50” “tilt” : “25” “lift” : “75” “schedule” : “1” “open” : “12:00” “close” : “21:00” “favorite_tilt” : “25” “favorite_lift” : “75” </pre>
--	---

Table 9.4.3.12 - Open/Close Test Case

Software Test Case 24 - Scheduling

Test Objective:	Ensure mobile application can create a schedule to the blinds
Test Materials:	<ol style="list-style-type: none"> iOS/Android Application Database
Procedure:	<ol style="list-style-type: none"> Open mobile application Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 View dashboard Turn on schedule Set open time to 12:00pm Set close time to 9:00pm
Desired Results:	<p>The database should contain a table with:</p> <pre> “blindID” : “1” “userID” : “1” “date” : “Current date” “nickname” : “Test Blinds” “temp” : “75” “humidity” : “50” “tilt” : “20” “lift” : “80” “schedule” : “1” “open” : “12:00” “close” : “21:00” </pre>

Table 9.4.3.13 - Scheduling Test Case

Software Test Case 25 - Delete blinds

Test Objective:	Ensure mobile application allows user to delete blinds from their account
Test Materials:	1. iOS/Android Application
Procedure:	<ol style="list-style-type: none"> 1. Open mobile application 2. Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 3. View dashboard 4. Click the menu button 5. Click delete blinds
Desired Results:	The mobile application should prompt the user the blinds have been deleted from their account and they should not be able to control them. Within the database, the blind table should remove the users ID from the field.

Table 9.4.3.14 - Delete Blinds Test Case

Software Test Case 26 - Blind list display

Test Objective:	Ensure mobile application displays all the blinds connected to users account
Test Materials:	<ol style="list-style-type: none"> 1. iOS/Android Application 2. Database
Procedure:	<ol style="list-style-type: none"> 1. Create test blinds in database connected to users ID 2. Open mobile application 3. Log in to test account using the following user information: Email: test@testing.ucf.edu Password: test123 4. View dashboard 5. Click the list icon
Desired Results:	The mobile application should display the two blinds connected to the users account

Table 9.4.3.15 - Blinds List Test Case

10.0 Administrative Content

The administrative content section, 10.0, is to properly document the process by which the SunShade senior design team completed their preliminary research and labors to contribute towards the project, as well as a milestone discussion, budget and finance discussion, Suppliers, future scalability, team organization and distribution of all team-related responsibilities. This is a step back from the operational and technical examinations of other sections, and examines the project as a team objective, as well as the plan for achieving goals and how successful it has been thus far.

10.1 Milestone Discussion

The milestone discussion is the section in which the intended timeline of events was expected to transpire, compared to when they did, as well as any reasons for delays, or preparations for future milestones.

	Objectives	Deadlines	Time Frames
Senior Design 1			
Documentation	Project Idea	5/28/2019	5/19 - 5/28
	Initial Project	6/2/2019	5/30 - 6/1
	Table of Contents	7/1/2019	6/12 - 6/29
	Draft of document	7/7/2019	6/12 - 7/2
	Final document	8/1/2019	7/8 - 7/30
Formalization	Standards, specs, and cost	6/1/2019	5/28 - 5/30
	Choose servos	6/7/2019	6/2 - 6/7
	Choose controller	6/7/2019	6/2 - 6/7
	Choose communication	6/7/2019	6/2 - 6/7
Prototyping	Power supply	6/18/2019	6/9 - 6/17
	Sensor and servo Interface	6/20/2019	6/9 - 6/19
	Communications	6/21/2019	6/9 - 6/20
Design	PCB	7/4/2019	6/17 - 7/3
	Connections	7/4/2019	7/1 - 7/7
	App	7/4/2019	7/1 - 7/7
	Casing	7/8/2019	6/17 - 7/4
Senior Design 2			
Presentation	CDR Presentation	TBA	TBA
	Conference Paper	TBA	TBA
	Midterm Demo	TBA	TBA
	Final Presentation and Demo	TBA	TBA
	Exit Interview	TBA	TBA
Fabrication	Order and Assemble	TBA	TBA
	Specifications Review	TBA	TBA
	Testing phase	TBA	TBA

Table 10.1.1 - Initial Semester 1 Milestones

The Milestone chart in Table 10.1.1 was deliberated upon and created at the start of the semester and has been adhered to as well as possible without major changes. It shows that the documentation period, including the completed paper, is to be worked on from 5/19 to 7/30, and to be due on 8/1. With the exception of the paper being actually due on 8/2, this is consistent, and allows time for final documentation binding and printing. Fortunately, creating an acceptable timeline for Senior Design 1 was simple, as it had a consistent number of deadlines spread throughout, in order to ensure an effective drafting process. All of these deadlines were met. Unfortunately, not all of the research and component decision deadlines were met, as the SunShade team is still deliberating over minor power changes, and some research was completed a few days after the desired time. This is considered to be typical for most projects of this nature, and the overall deliverability of the SunShade smart-blinds appliance is still, at the time of this documentation, progressing on time.

It is the primary goal of the SunShade team to have a completed, and fully functioning and deliverable project, in regard to both software and hardware, by the third week of the month of November. This also includes any and all necessary additional documentation, and a completed testing phase, so that there is enough time for members to prepare a professional presentation.

Many of the final deadline dates to occur throughout Senior Design 2 are still unknown, but it is imperative that the project team maintain consistent and realistic goals regardless. Because of this, working schedules have been made to serve as monthly milestones, as shown in tables 10.1.2 - 10.1.4.

September 2019 Milestones

September 2019		
Category	Goal	Completion
App	Basic app menu system working	09/01
General	All hardware acquired	09/01
App	Syncing system functional	9/4
Hardware	Motors installed to blinds	9/4
Hardware	Solar panels installed	9/4
App	App Menu optimized	9/11
Hardware	PCB design Finished & Ordered	9/11
Hardware	LDR and temp sensors installed	9/11
Hardware	Manual Buttons installed	9/11
Embedded	Battery charging operational	9/18
Embedded	Manual buttons Functional	9/18
Hardware	Battery charge system COMPLETE	9/18
App	Send/recieve data from MCU operational	9/25
Embedded	Wifi connection to phone operational	9/25
Hardware	Componenets wired	9/25

Table 10.1.2 - Target milestones for September

September will be the fullest month of the fabrication process, due to the slowness of August, due to a break in university schedule. It is also important to get started with force, so that procrastination is less of a possibility, and counterbalances the fact that the first month of the semester is typically much slower than the rest, and so the SunShade team hopes to take advantage of this, to properly utilize their time, and allow for a complete and effective testing phase later on.

October 2019		
Category	Goal	Completion
App	Motor functions controllable	10/2
General	Alexa link created	10/2
General	Alexa skills created	10/5
App	Scheduling system operational	10/12
Hardware	PCB system installed	10/15
Hardware	blinds fully mounted to test frame	10/15
App	Support for multiple blinds sync	10/23
Embedded	App displays Temperature and brightness	10/23
App	Motor functions screen optimized w/ slider	10/26
General	Alexa commands tested and fully oper.	10/26

Table 10.1.3 - Target milestones for October

November 2019		
Category	Goal	Completion
Hardware	Battery charge efficiency tests	10/6
Hardware	Hardware motor tests	10/6
App	Optimized schedule system	10/13
Software	Software system testing	11/13
App	Complete optimizations	11/15
General	Create Demo/presentation material	11/18
General	Final all-systems testing	11/22
General	Corrections phase, if needed, complete	11/30
General	All documentation complete	11/30

Table 10.1.4 - Target milestones for November

10.2 Budget and Finance Discussion

Since the SunShade team is comprised of students, and the project has no sponsors, the budget for the smart-blinds project was kept to a minimum. This is discussed further in section 5.2: Economic constraints. It was decided that the project should cost no more than \$400 for total development costs and should be relatively easy to stay within this parameter. The cost of materials would be divided evenly between the four members. Most of the components were purchased online, and a document was formed in order to maintain an accurate bill of materials, as well as personal expenditures, so that all members could pay or be repaid accurately and fairly.

10.2.1 Suppliers

Suppliers for all components used in the SunShade senior design project are included in Table 10.2.1.1.

Suppliers	
Arduino	Large distributor of electronic components and circuit boards
Amazon	an American multinational technology company that focuses on e-commerce, cloud computing, digital streaming, and artificial intelligence. It is considered one of the Big Four technology companies along with Google, Apple, and Facebook.
JacobsParts	Large distributor of electronic components
Home Depot	Retailer for lumber and construction equipment
Adafruit	Adafruit Industries is an open-source hardware company that designs, manufactures and sells a number of electronics products, electronics components, tools and accessories.
BG Micro	Distributor of electronics, electronic components, electronic parts, electronic kits, hobby electronics, surplus electronics

Table 10.2.1.1 - List of suppliers

10.3 Team Organization and Distribution of Responsibility

From the beginning, the SunShade team was very aware of the critical role that an effective plan of completion was for a successful design and fabrication of a full-scale project, as well as accompanying documentation. The team was distributed into two halves, based on field of study - electrical engineering, and computer engineering. Table 10.3.1 displays a breakdown of members by their discipline.

Team Members	
Team Member	Field of study
Conor Ferring	Electrical Engineering
Alex Rosado	Computer Engineering
Terrell Richardson	Computer Engineering
Miloud Ennemar	Electrical Engineering

Table 10.3.1 - Team Members and Majors

Each of the two halves held their own unique and independent responsibilities, as well as general responsibilities open to all members and with expected contributions. A breakdown of the responsibilities of both disciplines, as well as general responsibilities, can be seen in table 10.3.2. This table is under development and subject to change as specifics unfold and new responsibilities become apparent.

Distribution of Responsibilities by Discipline	
Discipline	Responsibilities
Electrical Engineering	Component comparison
	Component selection
	Power system design
	Schematic design
	PCB design
	PCB soldering
	Hardware Testing
Computer Engineering	App Development
	Embedded systems
	Blinds motor programming
	Brightness control programming
	Architecture research
	Software Testing
	Schedule programming
General	Frame construction
	Frame mounting
	Servo mounting
	Schematic analysis
	General testing

Table 10.3.2 - Distribution of Responsibilities by Discipline

In addition to the responsibilities listed in table 10.3.2, all team members were also expected to adhere to the complete senior design milestones chart, shown in table 10.1.1. It is also the responsibility of every member to adhere to the senior design 2 September, October, and November milestones, completing the above responsibilities according to said timelines.

In order to accurately gauge the amount of labor put forth by each individual team member, it was assigned that each member of the senior design team write a total of at least thirty pages, for four members totaling to one-hundred and twenty. Once the SunShade team deliberated over what was necessary to be included in the documentation, and the table of contents was devised as per the needs of the smart-blinds appliance, responsibilities pertaining to the final documentation were ready for distribution. For this, the team chose to color-code the table of contents, with each color

representing a member of the team. If a member desired to write a section, they highlighted in their color, and left an X next to that section to indicate it had not yet been completed, and was therefore, up for negotiation. The color-coding system can be seen in figure 10.3.3, where a snapshot of the system’s progress tracking is displayed, taken at a random point in the development of the documentation. The method proved to be an effective way to keep track of team goals and the overall progress of completed work, as well as prevent sections from being forgotten, under-counted, or over-counted.

Example snapshot of progress tracker

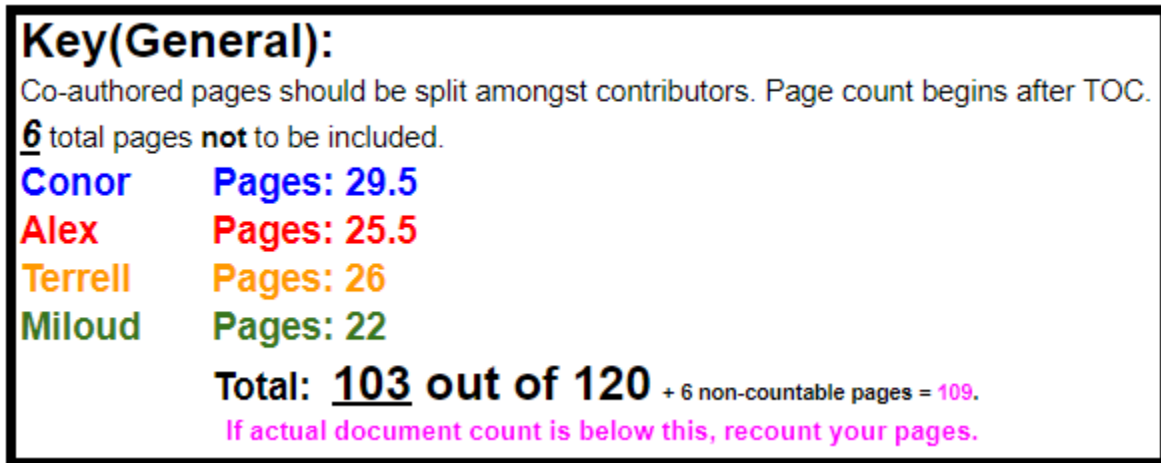


Figure 10.3.3 - Color Coded Progress tracker

11.0 Project Summary and Conclusion

The project summary and conclusion section of the 2019 SunShade senior design documentation is where the SunShade team discusses the overarching elements of the project construction, from an independent perspective. This section is where the bill of materials is to be presented and discussed, as well as the product user operation manual, and complete conclusion.

11.1 Bill of Material

Table below shows the details of the pricing, we have been able to purchase the materials. the table has been updated the table as soon as we purchase any components, or we find other cheaper vendors. we may also add parts to our bill of material, if we have enough time and money, we would be delighted to add features and therefore other components.

Our Bill of Material, as of right now is final, it’s been updated so we purchase other components, and also from the Eagle software we printed out our Bill of Material of the schematic for the PCB board.

Bill of Materials

Parts	Quantity	Unit Cost	Total Cost	Vendor	Acquired
Arduino Mkr1010 prototype board	1	\$33.45	\$33.45	Arduino	Yes
Blinds	1	\$33.88	\$33.88	Amazon	Yes
Diffuser Shades	1	\$~12	\$12	WalMart	Yes
Servo Motor 180 Degree	1	\$19.99	\$19.99	Amazon	Yes
Servo Cont. Rot. JX	1	\$22.79	\$22.79	Amazon	Yes
Servo Cont. Rot. SM	1	\$20.99	\$20.99	Amazon	Yes
DC Motor with Electric Roller	1	\$42.88	\$42.88	Amazon	Yes
MCU Atmel 328 Mega	1	\$6.95	\$6.95	Jacobs parts	Yes
LDR	2	\$4.99	\$0	On hand	Yes
Solar panel	6	\$16.99	\$101.94	Amazon	Yes
Temperature sensor DHT22	2	\$10.99	\$10.99	Amazon	Yes
Frame	1	\$24	\$24	Home depot	Yes
LED NeoPixel Strip	10	\$10.99	\$10.99	Amazon	Yes
SPST Switches	2	0.20	0.20	BG Micro	Yes

Pin Dip Sockets	2	\$0.08	\$0.08	BG Micro	Yes
Amazon Dot	1	\$22	\$0	Loaned	Yes
Total Cost:	-----	-----	\$392.17	-----	-----

Table 11.1.1 - Bill of materials for the SunShade project

11.2 Product User Operations Manual

The user operations manual is a documentation of the proper installation and usage of the finished and refined product, designed for use by the consumer. It contains all of the essential information regarding proper product usage, to ensure optimum effect and efficiency of the SunShade device.

11.2.1 Mobile Application

The SunShade control application will be found at the Google Play store for mobile applications and is to be installed. Ensure that the mobile device has a strong internet data connection, and find the SunShade Control App. The green 'INSTALL' button must be pressed, prompting a user password for their mobile device. On the mobile home screen, find the icon for the SunShade device, and select it to open it. Once the application is installed and opened, users are required to make an account, so that they can organize all of their smart-blinds appliances in their home. Once an account is made, the dashboard will appear, displaying temperature, schedule, and blinds positions. Users will need to set up their new device by pairing to their blinds and beginning Smart Blinds Setup, which will require an ID. From there, users may select which room their blinds will be in, for the purpose of user organization. Users may then select which blinds they wish to adjust and will be presented with a slider to manipulate the tilt of the blinds up or down, lift the blinds up or down, or drop the diffuser shade up or down.

11.2.2 Connecting to the Device

To connect to the device to Wi-Fi, there will be a WPS protocol button. Once the device is connected to WiFi it will be discoverable by the app from the same router. Once the app discovers and pairs the device, then communications can be sent over the internet to the device at any time. The Alexa module will communicate either directly to the device using the smart home set up provided by Amazon.

11.2.3 User Hardware Installation Guide

In order to successfully install the Smart Shade device in your home, users must first find their selected window frame and ensure that the window size matches the device. This can be done with a tape measure. Then, ensure that the previous blinds, if any, have

been removed completely, except for the side, wall-mounted U-brackets. This can be done by lifting the lever panel on each side of the blinds, designated as the U-bracket, and then sliding out the blinds structure. Once clear, users may replace the previous blinds with their SunShade smart-blinds appliance, and lower the U-bracket levers, locking the blinds into place. For the sake of compatibility for the user, the SunShade installs the same way any standard home blinds are installed, and so, any standard installation procedure can be followed, if the provided on here is insufficient. Manual control buttons can be found on the outer shell of the device, for ease of use for the user that does not require the mobile device on hand.

The Solar panels are to be placed on the window itself, facing outwards, towards the sun.

11.3 Senior Design One Conclusion

The project's requirements and specifications are met, by acquiring the components and tools in hands to make sure the sunshade project is going to be successful. The SunShade team worked diligently and efficiently as a unit with no problems and issues. Proper communication channels have been established between members in order to successfully complete the project on time and under budget, so that it may be presented at the end of the fall 2019 semester.

The next goal for the SunShade team is to closely follow the September, October, and November milestone tables in order to maintain steady progress to complete the sunshade without any issues. The specifications and requirements that have been outlined in this document will be strictly followed and have been considered for every element of the project design phase. It is the goal of the senior design team to create a highly energy-efficient smart home appliance with relatively high ease of use for potential consumers, while allowing for a comfortable living environment for the user, in terms of both temperature and brightness.

12.0 Senior Design Two

In Senior Design too many changes were made to the initial hardware design. Some of these changes were made to avoid complications. There were very few changes made to the software design. Lastly, this section will cover all issues that complicated the project before concluded.

12.1 Hardware

Hardware had four major changes that happened with the PCB alone. The PCB was changed in order to simplify design and require less components to drive down cost. In figure 12.1.1 shows our original target design to refer to changes.

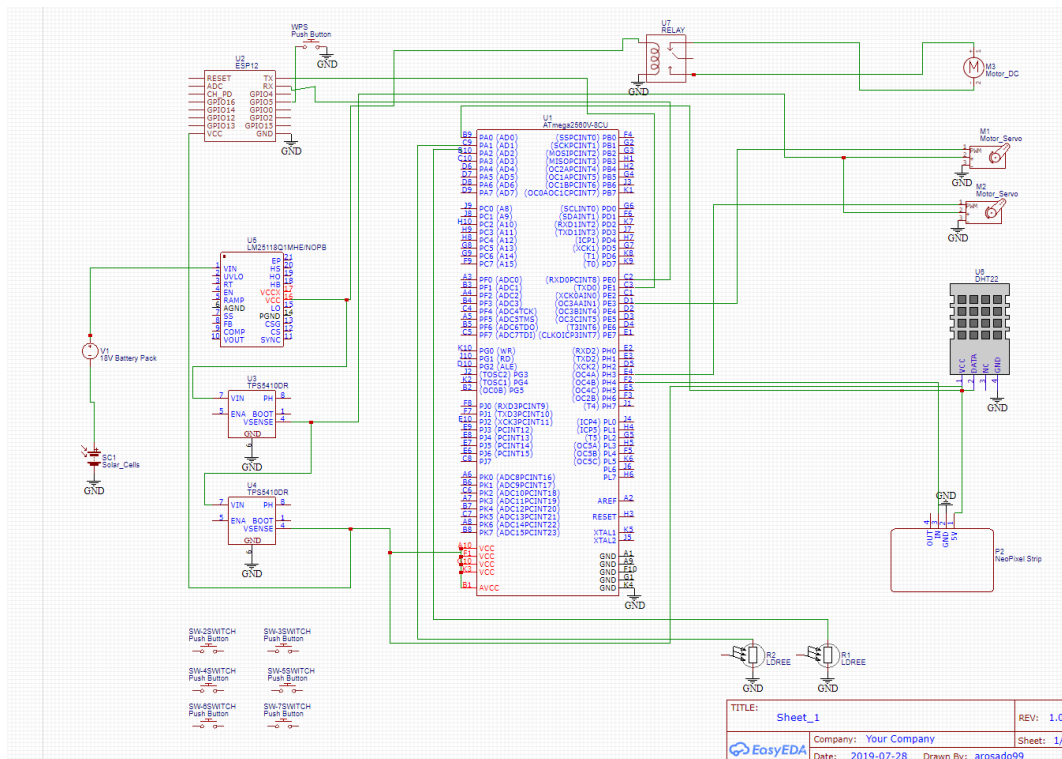


Figure 12.1.1 Mock Schematic

The first change made was swapping out the Atmega2560 with the Atmega328p. This was done to reduce space and complexity on PCB. The memory space on the Atmega328p was sufficient for all needs. The change did not affect the power supply because both chips have 5V operating voltages. The change also allowed us to continue using the same code.

Next we removed the Buck-Boost from the design when we decided to use a 12V source. The 12V source allowed for ease when integrating the solar panel charge controller. This also allowed us to have a larger mAh rating for our battery source. The Buck-Boost had a significant footprint with all the supporting components. The changes were also affected the 3.3V buck, which was changed to an LDO. The 3.3V LDO was then sourced by the 5V buck instead of the 12V source to increase efficiency. This greatly reduced the footprint from the power supply.

A motor driver was added instead of using a makeup h-bridge using transistors. The motor driver was used to increase our control over the 12V DC motor. This change reduced the footprint of the design and introduced speed control to the motor. The speed control was never integrated due to other complications.

Manual control buttons were removed from the PCB design simply to reduce scope and footprint. The team felt that the app control feature alone would be sufficient for a cleaner look. Products we found usually included only a remote for control so, it felt unnecessary to include manual buttons on frame.

The figure 12.1.2 will show the final schematic for comparison of the changes made from the original design goals. In all the PCB ended up supplying all sufficient voltages across the board. There were some last-minute complications that will be later discussed.

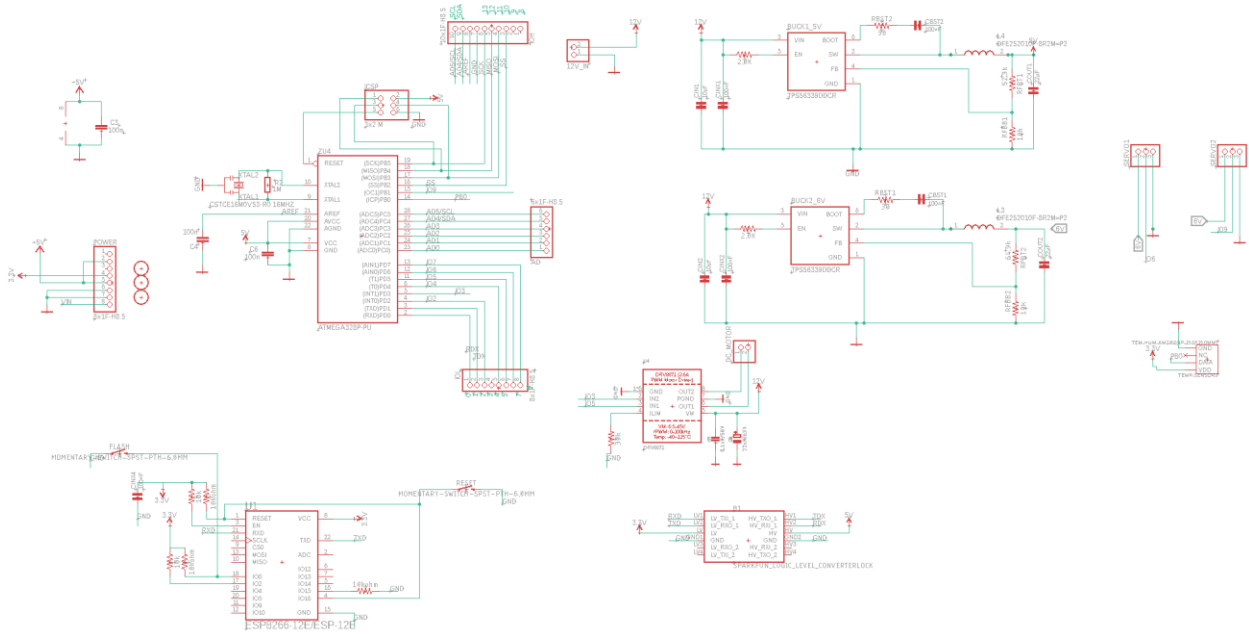


Figure 12.1.2 Final PCB Schematic

12.2 Software

The only change made to the software was the logic to move the blinds positions, control the schedule, lights, and shade was done on the ESP8266 instead of being done on the Atmega328p. The Atmega328p was responsible for waiting for any changes being sent from the ESP8266 and determining any changes to the temperature, humidity, or light in the room. If there was a change in any of the conditions the Atmega328p would send it to the ESP8266 and it would send the change to the database.

12.3 Complications

Unfortunately, there were a few complications that dictated the way features work. During breadboarding all core features were available. The app and Alexa were updating the database and sending commands to the Wi-Fi. The board ran each motor with ease. The complications came with PCB timing and design. Our first PCB had the wrong footprints for many major components, so this did not allow us to test on the first PCB. The second PCB was closer to a success, but during testing the software team discovered that the buck's components were getting hot. One of the resistors were wrong so both bucks were outputting 6V. The team fixed the resistor issue to test code on the PCB. Then the esp12e fried during testing. This was a result from not including a level converter to drop the 5V transmitted signal from the Atmega328p. The Final PCB design which will be displayed in figure 12.3.1 fixed the majority of the issues.

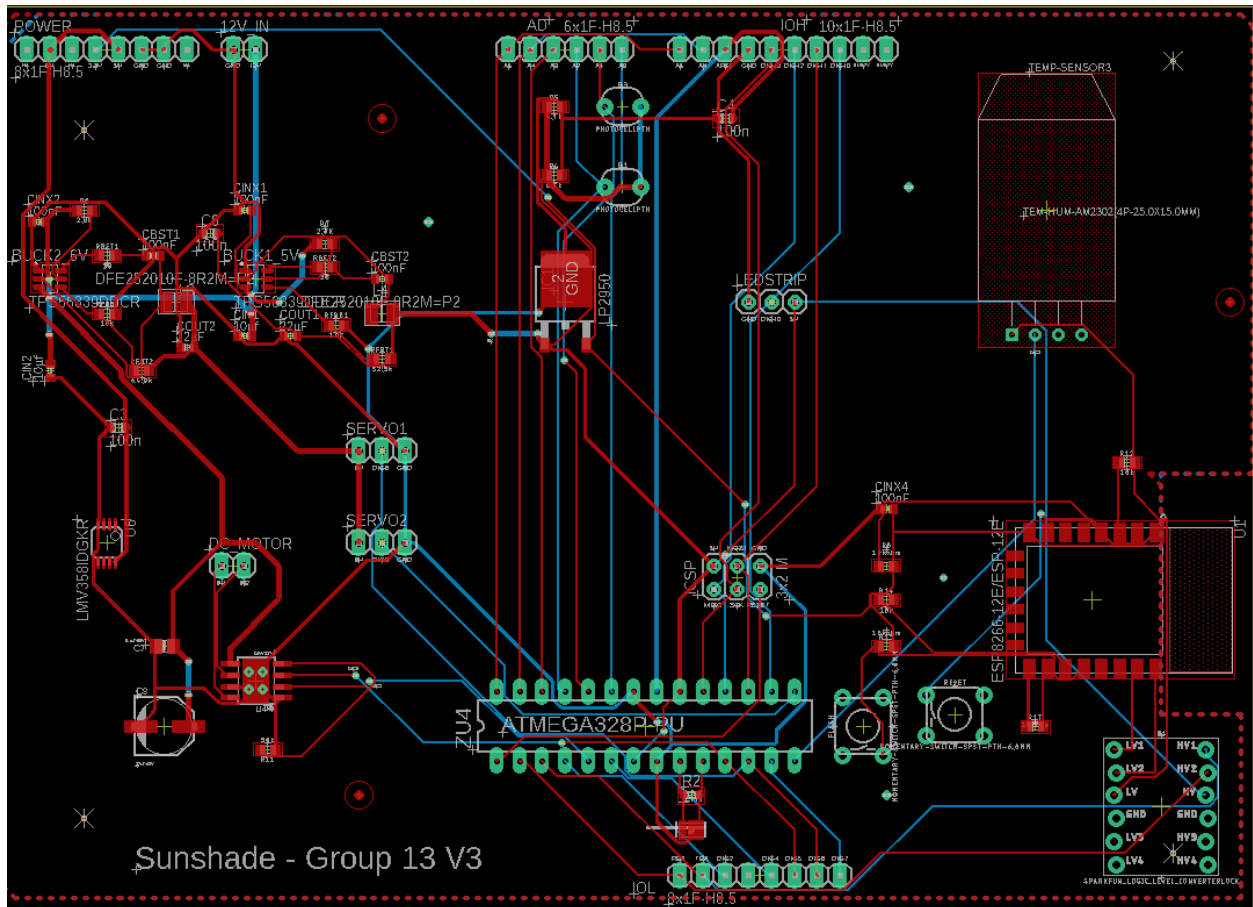


Figure 12.3.1 Final PCB Board File

One of the major issues drastically changed our final demonstration and product. The resistor for the 6V buck resistor fried the night before our demo. This was during a final check to make sure all functions still worked properly. This disabled the ability to run the lift motor on demo day. On top of that the esp12e would not flash and then got stuck in a boot loop. Time restrictions did not give us the ability to recover from the problems presented to us late. Luckily the team documented functionality with a video we were able to present to the judges. To combat the Wi-Fi and 6V complications a demo code was uploaded to display the rest of the PCB functionality without remote access. The rest of the PCB worked flawlessly, and if more time was permitted a fully functional product would have been completely possible.

12.4 Conclusion

In conclusion, the project is regarded as a success. The milestones set forth by the team were adhered to within relative accuracy, as well as budget constraints being reasonable, despite the cost of PCB fabrication, shipping, soldering, and component costs. The SunShade team was introduced to new concepts, procedures, problems and engineering elements they had not yet dealt with. Fortunately, each member was able to learn new skills and methods in order to complete a functioning final product.

The finished SunShade project was presented before a panel of University of Central Florida professors and faculty on November 21st, 2019. At this panel, the team presented their research, methodology, and experience working on the project SunShade, before finally presenting the product itself for demonstration. With only minor performance hiccups, which were explained by the team, the project displayed nearly full functionality, and all elements were performed within the constraints set forth by the team. The members of the esteemed panel deemed the finished project to be a success, bringing senior design to a close.

Appendix-A Copyright permissions

BatteryU <BatteryU@cadex.com>

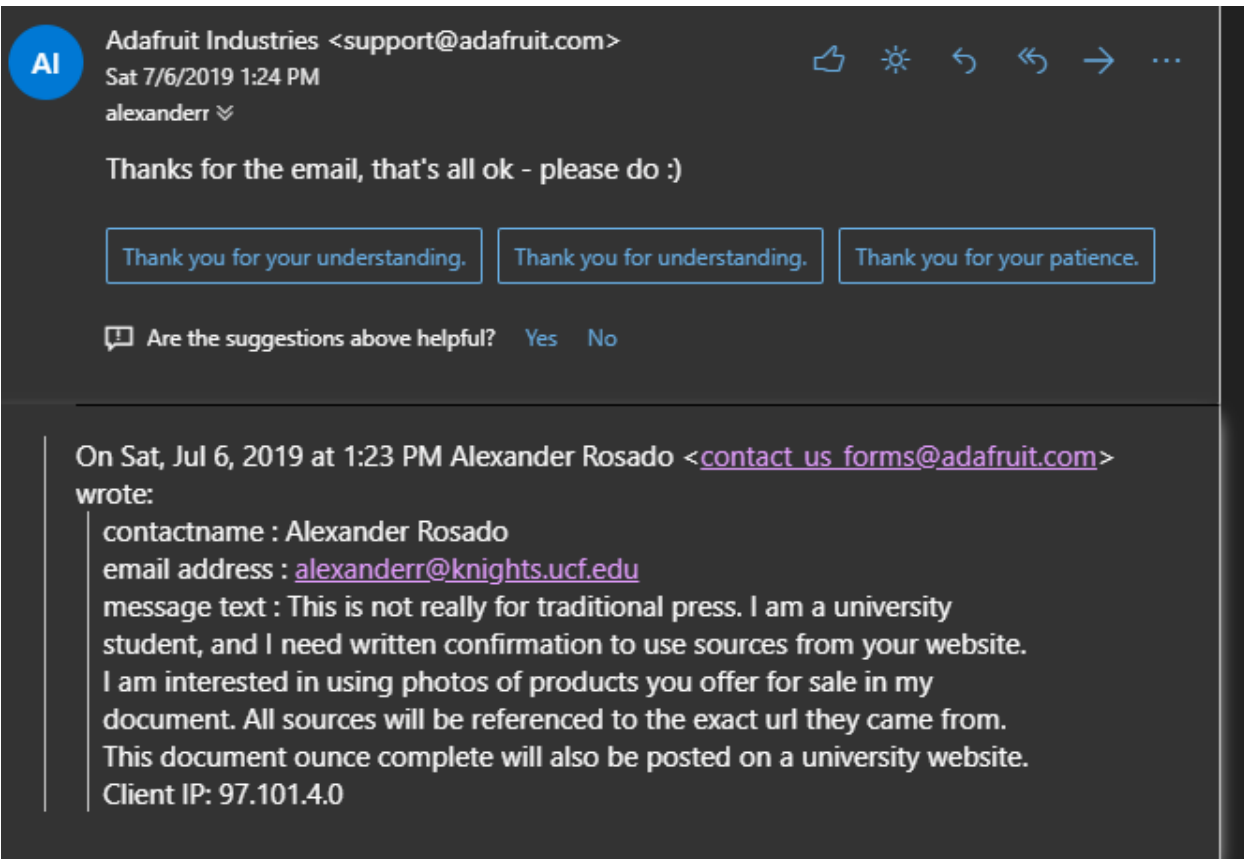
to me ▾

Hi Alexander,

Yes, you may use the material as requested. Please cite source where appropriate.

Regards,

John Bradshaw - Marketing Communications Manager
Cadex Electronics Inc. | www.cadex.com
Vancouver | Minneapolis | Frankfurt
Tel: +1 604 231-7777 x319 | Toll Free: 1-800 565-5228



The screenshot shows an email interface with a dark background. At the top, a blue circular profile icon with the letters 'AI' is next to the text 'Adafruit Industries <support@adafruit.com>'. Below this, it says 'Sat 7/6/2019 1:24 PM' and 'alexanderr ▾'. To the right of the header are icons for thumbs up, a sun, a refresh arrow, a left-pointing arrow, a right-pointing arrow, and a three-dot menu. The main body of the email contains the text 'Thanks for the email, that's all ok - please do :)'. Below this text are three rectangular buttons with rounded corners and blue borders, each containing the text 'Thank you for your understanding.', 'Thank you for understanding.', and 'Thank you for your patience.' respectively. At the bottom of the email body is a feedback prompt: 'Are the suggestions above helpful?' followed by 'Yes' and 'No' options. Below the email body is a white-bordered box containing a quoted message: 'On Sat, Jul 6, 2019 at 1:23 PM Alexander Rosado <contact_us_forms@adafruit.com> wrote:'. The quoted message text is: 'contactname : Alexander Rosado', 'email address : alexanderr@knights.ucf.edu', 'message text : This is not really for traditional press. I am a university student, and I need written confirmation to use sources from your website. I am interested in using photos of products you offer for sale in my document. All sources will be referenced to the exact url they came from. This document ounce complete will also be posted on a university website. Client IP: 97.101.4.0'.

- AR** Alexander Rosado 11m ago
I am a UCF student completing a senior design document. I am required to have verified proof to use pictures or information from any source. I am interested in using diagrams that are presented in voltage buck converter pdf. May we have permission to use the desired information in our document? All information used for this platform will be referenced.
- FL** Farley Liboon 10m ago
Thank you for contacting TI Customer Support. My name is Farley.
Hi Alexander, Let me look into your query and I will get back to you as soon as I have the information.
- AR** Alexander Rosado 9m ago
Thank you
- FL** Farley Liboon 5m ago
Please stay on the line I am still checking
- AR** Alexander Rosado 5m ago
No problem
- FL** Farley Liboon just now
thank you for your patience,
you may cite the requested document for your project we only ask that you include "Courtesy of Texas Instruments Inc." under the image.
Is there anything else that I can help you with?
- AR** Alexander Rosado just now
Thank you for all your help, this is great.



Artis Coleman

JUN 27, 2019, 12:14 PM

Hey! Could I ask you about your UCF senior design project from a few years ago?



What about it?

You can now call each other and see information like Active Status and when you've read messages.

I'm a current senior design student working on a very similar project for app-controlled smart blinds and in the research portion of my paper I'm discussing existing projects. I was wondering if I could have your permission to discuss your project and use any photos/info you may have included in your original paper? I was told I had to get permission for this kind of stuff

JUN 28, 2019, 12:02 PM



Yea you can 😊 I don't even have a copy myself.

Appendix-B References

[1] Instructables.com, 'Automatic Window Blinds Controller (PICAXE)', 2015. [Online]. Available: <http://www.instructables.com/id/Build-A-Motorized-Window-BlindsController-For-Les/>. [Accessed: 25 - June- 2019].

[2] University of Central Florida Senior design 'Solar Blinds', 2015 [Online]. Available: <http://www.eecs.ucf.edu/seniordesign/su2015fa2015/g11/> [Accessed 20 - June - 2019]

- [3] http://sensorembded.com/product_extra_files/ldr.pdf
- [4] <https://www.kitronik.co.uk/blog/how-an-ldr-light-dependent-resistor-works/>
- [5] Burgess, P. (2014, July 24). Adafruit DotStar LEDs. Retrieved July 3, 2019, from <https://learn.adafruit.com/adafruit-dotstar-leds/overview>
- [6] BU-205: Types of Lithium-ion. (n.d.). Retrieved July 7, 2019, from https://batteryuniversity.com/learn/article/types_of_lithium_ion
- [7] Light Sensor including Photocell and LDR Sensor. (2018, February 15). Retrieved July 4, 2019, from https://www.electronics-tutorials.ws/io/io_4.html
- [8] Solar Cell Comparison Chart – Mono-, Polycrystalline and Thin Film. (2013, June 3). Retrieved July 2, 2019, from <https://energyinformative.org/solar-cell-comparison-chart-mono-polycrystalline-thin-film/>
- [9] Schoder, T. (2018, January). Monocrystalline Cells vs. Polycrystalline Cells: What's the Difference? Retrieved July 7, 2019, from <https://www.civicsolar.com/article/monocrystalline-cells-vs-polycrystalline-cells-whats-difference>
- [10] Types of thin film solar panels. (2019, February 28). Retrieved July 7, 2019, from <https://www.energysage.com/solar/101/about-solar-panels/thin-film-solar-panels-amorphous-cadmium-telluride-and-cigs/>
- [11] Adafruit Industries. (n.d.). Feedback 360 Degree - High Speed Continuous Rotation Servo. Retrieved July 7, 2019, from <https://www.adafruit.com/product/3614>
- [12] Temperature sensor comparison – DHT22 vs DS18B20: Arduino tutorial. (2019, April 22). Retrieved July 6, 2019, from <https://www.geekstips.com/temperature-sensor-dht22-ds18b20-arduino-tutorial/>
- [13] Adafruit Industries. (n.d.). Stepper motor - NEMA-17 size - 200 steps/rev, 12V 350mA. Retrieved July 5, 2019, from <https://www.adafruit.com/product/324>
- [14] Arm Ltd. (n.d.). CPU Architecture. Retrieved July 1, 2019, from <https://www.arm.com/why-arm/architecture/cpu>
- [14] Arm Ltd. (n.d.). CPU Architecture. Retrieved July 1, 2019, from <https://www.arm.com/why-arm/architecture/cpu>
- [15] Singh, S., & Pandit, D. (2015). Emerging Trends of Embedded System for Industrial Applications. International Journal of Advances in Engineering, 1(9), 661-666. doi:<https://pdfs.semanticscholar.org/6b97/daf611cf54f58074eb0a308107d42cb8ef1a.pdf>