"SunShade" Smart Blinds - Senior Design II, Fall 2019

Conor Ferring, Alex Rosado, Terrell Richardson, and Miloud Ennemar

College of Engineering & Computer Science, University of Central Florida, Orlando, Florida 32816-2450

***Abstract* - The SunShade is an internet-of-things home appliance designed for daily use, in order to create a comfortable and cost-efficient home environment. SunShade is a cordless, solar-powered window blinds system, controlled either with the manual control pad, Alexa personal voice assistant, or the mobile application. It is capable of tilting shades, lifting or lowering blinds, or lifting or lowering a diffuser shade for enhanced environmental control, as well as numerous other technical features. The primary feature of the SunShade project is its green-energy oriented solar power. The project is capable of charging and powering itself without adding to the cost of monthly electric bills, while lowering strain on home air-conditioning units, while also adding to personal home utility and comforts.**

INTRODUCTION

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 SunShade team has 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 will be just as simple as installing traditional blinds, and application and smart-home voice control linking will be a breeze, providing a power-saving, energy efficient source of home comfort.

Features:

App control - 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.

Scheduling - Allows for users' SunShade to protect their window-dwelling plantlife from over-exposure, as well as save the user on monthly electricity costs by limiting the effects of the sun on air conditioning usage.

Voice controlled - Amazon Echo, aka Alexa 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.”

DESIGN PHYSICALITIES

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 integrated another servo, intended for 180-degree use, which are attached to the end of the tilt rod to directly manipulate it clockwise and counterclockwise, which will tilt the horizontal blinds open and closed. For precision control, both servos are equipped with a potentiometer, so that an accurate ‘open/close, up/down’ value can be established for varying positions. This enables 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 was 3D printed.

In addition to the lift and tilt capabilities mentioned above, the SunShade project is also fitted with both an automatic roll-down diffuser shade, and a neopixel LED strip across the top frame of the device in order to allow for a greater degree of ambient light control. The roller-motor was fitted with a PVC-pipe and a rolling shade, and controlling the direction of roll, and therefore, the up or down of the shade, is a simple matter of reversing the voltage flow to the motor.

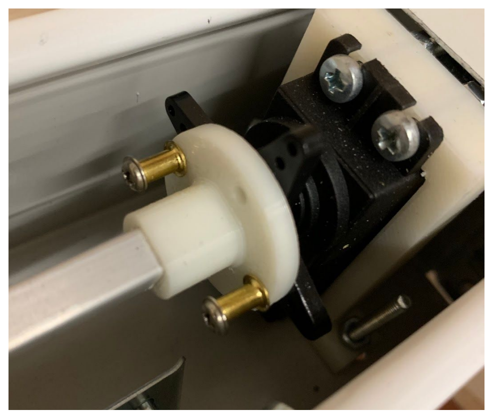

Figure 1: 3D printed tilt-rod servo adapter & housing

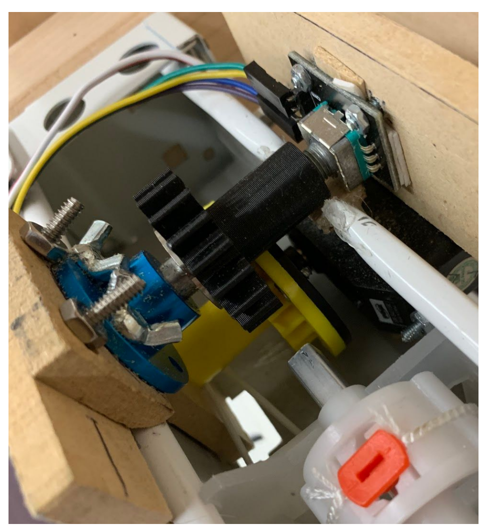

Figure 2: 3D printed gearbox for lift mechanism

TECHNICAL DESIGN

The technical system of the SunShade appliance begins with the ATmega 328 Uno Microcontroller unit. The 328 takes in data received from the ESP12 wifi module, which communicates with the smart-phone application, as well as amazon Echo. With the command signal received from the user, to the wifi module, and to the 328, the microcontroller unit then utilizes this information to control which physical response is output. For example, should the user command on the application that they desire the shades to tilt to fifty percent, then the designated value for that position will be communicated to the 328 in order to turn the 6-volt servo to the half-way position. This will be done for all three servos, as well as the neopixel LED strip. The three servos control the tilt, shade lift mechanism, and the diffuser shade mechanism. All of these are considered to be the physical outputs.

Figure 3: Light Dependant Resistor

Since the SunShade is capable of automation, it also requires certain inputs other than direct user control. These inputs include data from the light-dependent resistor photocells, which assign levels of brightness to specific values for use; as well as the DHT temperature and humidity sensor, which does the same for its corresponding external readings. These values are then set within user-designated parameters in order to automate the system, as well as feed data directly back to the user via the smart-phone application. For example, if the user has set that they only want a certain level of brightness or temperature throughout their home, then that brightness or temperature is given a data value. If the actual data value exceeds or is below the range of values decided by the user, then the blinds will self-alter in order to accommodate and reach the desired temperature or brightness, to the best of its ability. If the brightness level is below the desired level, the neopixel LED strip can be activated in order to shine on the lowered diffuser shade in order to provide ambient lighting.

Figure 4: Temperature and humidity sensor

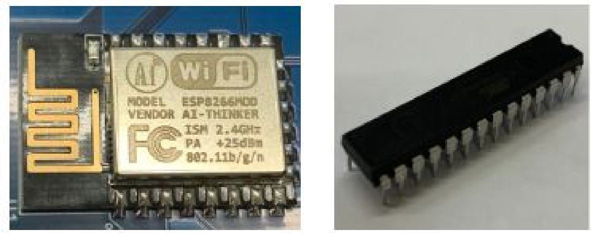

Figure 5: the ESP-12E Wifi module & the ATmega328p microprocessor

POWER

The power system of the SunShade project is separated into two major areas: Power sources, and power output.

A: Power Sources

A set of ten 3.3W, 5”x5” monocrystalline photovoltaic solar cells, are aligned across the back of the Sunshade project, across what would be the externally-facing side. These are arranged and soldered in series. Each cell has an average power of 3.3W, with a max voltage of 0.58V, and current of 5.93A. Using a charge controller, the ten cells charge a 12v lithium-ion battery. The charge controller serves to protect the battery from overcharging.

Figure 6: 5”x5” monocrystalline photovoltaic solar cell

B: Power output

The power system of the SunShade design must meet certain requirements in order to effectively supply power to all of its many components. The complete system is displayed in figure 7:

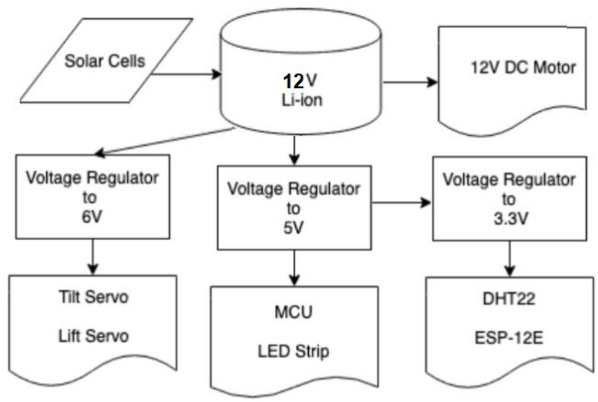

Figure 7: SunShade power system flowchart

From the flow chart it can be seen that once the 12v lithium-ion battery pack is charged, our 12v DC motor for the diffuser shade is directly powered, but three voltage regulators step down the power to the appropriate voltages. One regulator brings the voltage from 12v to 6v in order to power the 'tilt and lift' servos, another brings it down from 12V to 5V in order to power both the 328p MCU and the neopixel LED strip, and then from the 5v source, a third regulator brings the voltage down from 5v to 3.3v, which is used to power the DHT22 temperature/humidity sensor, and the ESP-12 Wifi module. With regards to the 12V DC motor, changing the direction of the shade to up or down is a simple matter of reversing the voltage flow, which is done by the MCU.

SOFTWARE DESIGN

The software design consists of two major areas: Mobile application and backend.

A: Backend

The backend of the software was created and hosted using Amazon Web Services. The database is a MySQL created using AWS Relational Database Service containing two different tables used to store the users and blinds. The user table contains the user's information such as their name, email, and area code. Each user has a unique personal key that identifies them and gives them to multiple blinds. The blind table contains a unique personal key and the blind's information received from the sensors including the tilt/lift positions, temperature, humidity. The blind table also contains information inputted from the user such as the scheduling time and favorite positions. The user and blind table has an one-to-many relationship which means one user can be connected to many blinds but a blind can only be controlled by one user. To achieve this the user's personal key is stored in the blind tables that belong to them.

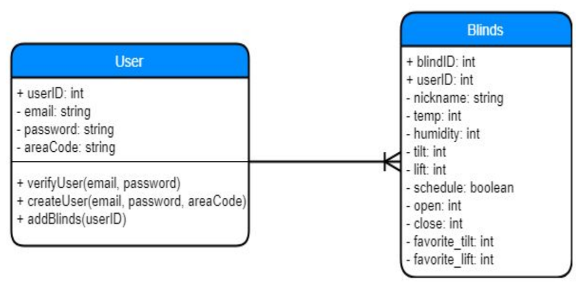

Figure 8: SunShade class diagram

The database is accessed using AWS Lambda, a serverless computing platform, through functions written in NodeJS. Lambda is set up to act as a server for the RESTful API communication between devices. The functions processes and transforms the information received and sends an appropriate response via a JSON. Lambda accesses the database based on the type of request and the path then creates a response with the requested data and response code. To activate Lambda, API endpoints are set up via AWS API Gateway. API Gateway allows the blinds and mobile application to send requests to one endpoint but with different paths. These paths are set up to handle POST, GET, PUT, or DELETE requests to manipulate or retrieve data from the database. API Gateway is accessible from both the blinds and mobile application through the same endpoint but the two devices use different paths.

B. Mobile Application

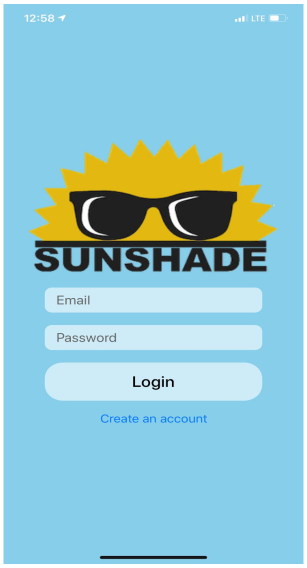

Figure 9: Mobile app login screen

The mobile application is used to monitor the environmental conditions the blinds are located in and control the position of the blind. The mobile application was created using React Native, a Javascript framework for cross platform mobile development. React Native allowed for support on iOS and Android devices without the need to develop the same application in two different native languages. To use the application the user must register an account within the database and log in. After login the user is shown the dashboard of the application which serves as the home screen. During the login process the app sends a GET request to the API Gateway to retrieve the user information needed for the dashboard. The dashboard contains a list of the blinds connected to their account along with the nickname of the blinds, a button to add a new blind, and a button to log out. When the user selects a blind it will take them to the detailed view of the blind. The blind detailed view gives the user the ability to change its positions via sliders or buttons, change the nickname, mark favorite positions, and add a schedule for the blinds to follow. From the blind screen the user has the option to remove the blind from their account if needed. Upon updating any of the values of the blinds, the app sends a POST request to the API gateway to update the database with the new values. If the request fails the values will return to the old values and the app will prompt the user that the update failed.

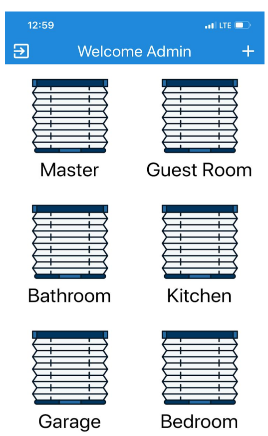

Figure 10: Mobile app dashboard screen

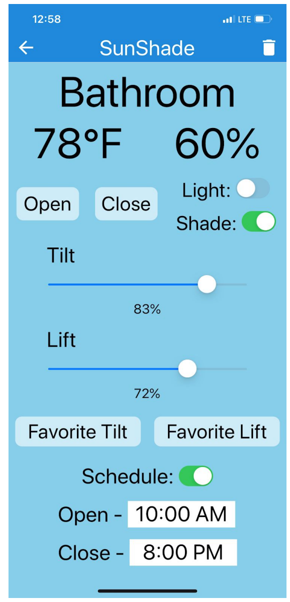

Figure 11: Mobile app blind screen

PRINTED CIRCUIT BOARD

We tried to make our PCB with the minimum cost, we used less components as possible and we made sure we go more than two layers.We based our circuit design on the flowchart described above. The power source was straightforward, we used just two pin header, since we are using just the batteries as an input voltage, we realized by using the power jack we will have more unnecessary footprints, for example, the power jack schematic we would have to use a capacitor that has a significant footprint size and also we would have to use a Diode. For that reason, we eliminated the comparator that is connected to the power jack , after doing some research we found out that comparator is made to compare if the input voltage is through a DC jack or batteries. 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 we made 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 is necessary in our design to use switching voltage regulators, after long consideration we realized that it's best to the TPS56339DDRC for its great efficiency and simple diagram, also this switching regulator can provide the desired output voltage regardless of what is the input voltage. The TPS56339DDRC can take an input voltage from 4.5V to 24V which is within the range needed, and outputs a wide range of voltages , we designed two schematics since we needed one to output 6V as shown in the figure below and another to output a 5V. We thought about using the Linear voltage regulator but after doing research we discovered that they will heat up quick because since they are linear they would only take an input voltage that is closer to the output or desired voltage, for example, we are using the LP2950 linear voltage regulator that provides 3.3V to the temperature sensor and to the ESP12E wifi chip, this particular linear voltage regulator's input voltage is 5V , and we expect that is not going to heat at all. The schematic of the switching voltage regulator was taken partly from the TI webench design, although the webench gives excellent specifications , for example, it shows the efficiencies of different designs and also different footprints, however, it was not a good idea to copy the schematic to the Eagle software because by copying, it wouldn't copy the libraries for all the components and also we had to manually connect all the components because eagle gave us errors.

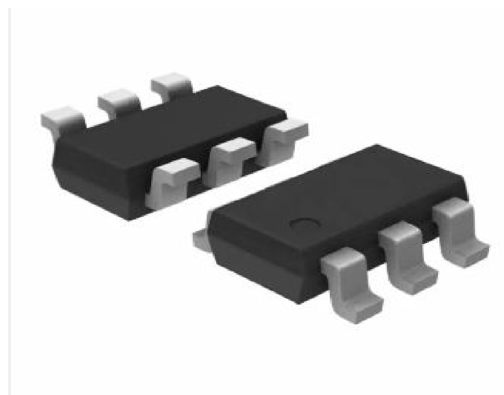

Figure 12: TPS56339 DDRC

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

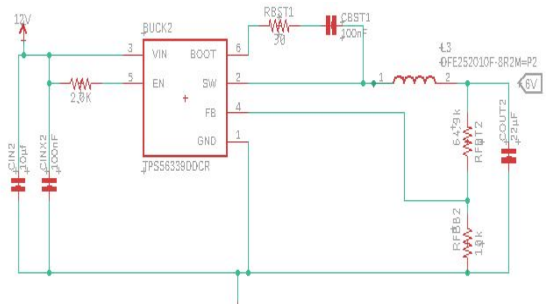

Figure 13: Power Distribution System

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. Also we have the PCB through the 12V batteries 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

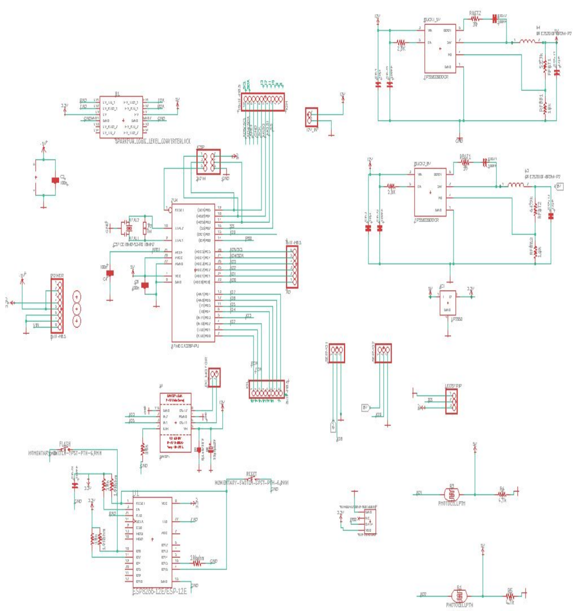

Figure 14: SunShade Schematic

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.

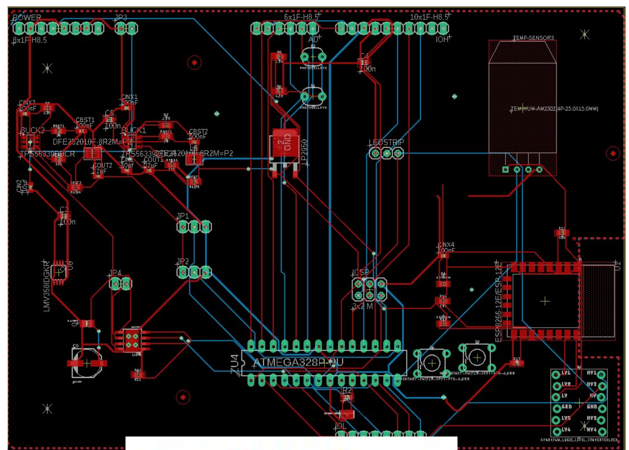

Figure 15: Board Layout

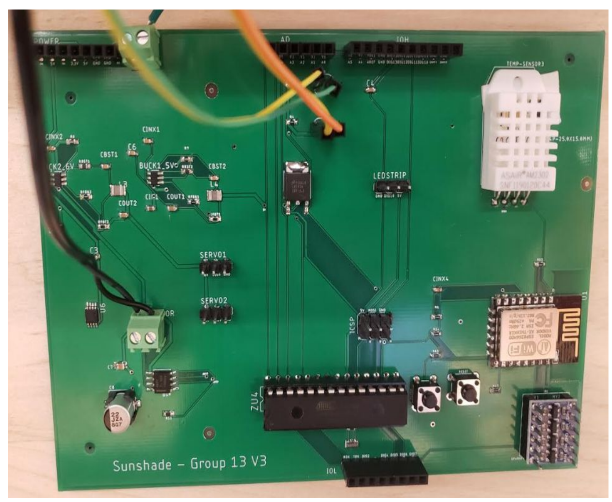

Figure 16: Final PCB

BIOGRAPHY

Conor Ferring is an Electrical Engineering student with a minor in Intelligent Robotic Systems at the University of Central Florida. He will Graduate on December 14th, 2019, with plans to begin his career in the field of engineering. His interests include robotics, electrical system design, power distribution, and power systems.

Alexander Rosado is a Computer Engineering student at the University of Central Florida. He will graduate on December 14th, 2019 with plans to pursue a career in robotics within the field of medicine. His interests include robotics, fabrication, artificial intelligence, and embedded programming.

Terrell Richardson is a Computer Engineering student at the University of Central Florida. He will graduate on December 14th, 2019 with plans to pursue a career in software engineering. His interests include embedded programming, software development, and artificial intelligence.

Miloud Ennemar is an Electrical Engineering student at the University of Central Florida. He will graduate on December 14th, 2019 with plans to pursue a Master Degree in control systems, and control design.