Jetsons Living – Smart-Home Blackout Shades

Stephen Fryer, Brian Mora, Priyank Patel, John Rossi

Dept. of Electrical and Computer Engineering University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The objective of this project is to design and test an internet connected blackout shades system prototype for Jetsons Living, a local home automation company. The solution is expected to be wireless and interface with Amazon's Alexa Smart Assistant, as well as a web application, with the expectation of being integrated into future Jetsons Living Infrastructure. The primary focus of this paper is on the hardware and software design of a Smart-Home device.**

***Index Terms* — Battery Management Systems, Home automation, Internet of Things, Microcontrollers, Solar Panels, Wireless Communication**

I. Introduction

The Motivation for this project is to enhance the living conditions of a typical homeowner who wishes to live more comfortably and connected. These days, many homeowners typically already own smart home devices such as voice assistants (Amazon Echo and Google Home), smart lighting (Phillips Hue), smart switches (GE Z-Wave Wireless switches), video surveillance, smart thermostats (Nest), video door bells (Ring), etc. Our goal is to add an integrated smart shades solution to their home. As smart home devices become more popular and advanced, users will be looking for an increasing number of smart-home devices. Thus, creating an initial product for Jetsons Living's suite of devices is in the best interest of the company.

Smart home devices are meant to be simple to operate, and useful. These two concepts are the basis for the entire design of the system. To make the smart shades system intuitive, various simple input modes are incorporated including voice commands through Amazon Echo, a web application, and a remote control. The utility of the smart shades system is enhanced through hardware and software features that will be explored further in future sections.

The expectations for this project are, simply put: that an effective, operational prototype of the smart-home blackout shades system be produced. In greater detail, the project expectations include researching solar power generation, wireless networking, power management, and web application design. Further including the design and testing of the entire system.

II. Product Overview

The main priority for our project is that the device must be internet connected, and be able to power itself. That being said, there are still many other opportunities to add peripherals to our system. This can increase the overall utilization of the product and add rigor to our project. Our product will include many different features that each help show what is possible with the communication of electronic devices over the internet.

A. Features

Table I. Features

Power	120VAC **OR** Lithium-Ion Battery [w/ Solar & 5VDC Charging]
Size	24x30 in Window
Inputs	Web Application, Amazon Echo, PIR sensor, IR Remote
Outputs	Motor, LED Strip

The smart shade system has a great focus on utility, and as such has various useful features for shades control.

As mentioned previously, the smart shade system responds to voice controls. These voice controls can activate all core features such as changing the LED color and brightness, raising and lowering the shades.

Additionally, the web application provides users control over these features as well as others. Features unique to the web application are the ability to set times of day to automatically lower and raise the shades; ability to enter user settings, such as email; a quick-start guide; display current battery level; ability to choose the response to motion. A nightlight motion setting, which activates the LEDs at a low brightness and turns them off 20 seconds after motion is detected. An alarm motion setting, which upon detecting motion sends the user an email and flashes the LED strip blue and red until the alarm is disarmed.

This feature set allows for a robust user experience that enhances this smart shade system far beyond traditional window shades.

B. Market Requirements

During our initial brainstorming sessions we started by coming up with some market requirements that could help the product be more successful. Aside from all the features mentioned before, it would be ideal if some, or all, of the following requirements could be met:

1. Simple User Interface: System should be easy to use.
2. Wireless Connectivity: The system should be able to be communicated with wirelessly, preferably via Wi-Fi.
3. Battery Life: System should have an extensive battery life, preferably indefinite.
4. Easy to Install: First-time install should be fast and simple.
5. Noise: System must not produce any loud and unnecessary noise.
6. Cost: production cost of the system should be minimized for the sake of commercialism.

III. Problem description

The objective of this senior design project is to research and design a prototype smart-home blackout shades system. After researching, and discussing with Jetsons Living, a list of requirements was decided upon for the smart shades system prototype.

A. Requirements

The requirements put forth by Jetsons Living for an effective system are listed below:

- The system will be responsive, and as such, should react to user input in at least 1 second.
- Can charge itself effectively solely through solar power by supplying, at peak, 6 watts of power to the system.
- Equipped with a motion sensor with a range of at least 3 meters.
- Sufficiently low power, such that peak power consumption is less than 10 watts.
- A physical remote control with a range of at least 15 meters.
- Manufacturing cost of at most $300.
- Feature at least 3 Amazon Echo voice skills.
- Move the shades at a minimum speed of 2 in/sec.

IV. Internet of things background

Since the main method of control over our device will be Wi-Fi, this classifies it as an Internet of Things device. The Internet of Things (IoT) is the concept of small embedded electronic devices all sending and receiving data over the internet. [1] Smart home devices are especially popular IoT devices since the internet is one of the main mediums used in communicating to these devices, Other communication platforms include Bluetooth [2], Zigbee [3], Z-Wave [4], but Wi-Fi proved to be more convenient to use and offers an experience that users are already familiar with. Since the functions of our window shade system are relatively simple, it was decided to use an all-in-one System on Chip (SoC) microcontroller unit. The SoC solution allows us to directly communicate with the microcontroller using HTTP commands. The specifications of importance being good Wi-Fi reliability, inexpensive, and sufficient ports to support all the system peripherals.

A. Final Selection

The final SoC solution selected was the ESP-12S from Ai-Thinker. [5,6] The chip's specifications fit the needs of this project at a competitive price point.

B. Specifications

Some specifications of note are as follows:

- Supports IEEE 802.11 b/g/n
- 1 MB Flash Memory
- Integrated low power 32-bit CPU
- Standby power consumption of < 1 mW
- 14 GPIO Pins
- $6.95 per chip

V. Design

Our design approach was initially to come up with a complete, ready to market product that is easy to install and fully self-contained. Below in *figure 1* is an initial sketch of our design.

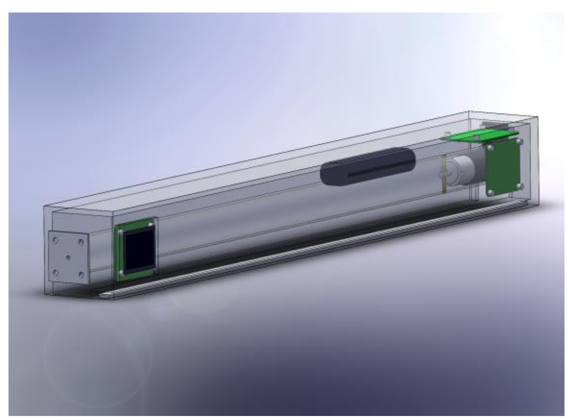

Fig. 1. Initial System Design

This sketch includes; the shade shaft that attaches to the motor, the power PCB, control PCB, PIR PCB, and the batteries. The Power PCB handles the AC to DC conversion and then outputs a steady DC voltage. It also handles the battery charging and load sharing circuitry for the solar panels, and 5VDC charger. The power board can supply 5 Volts at up to 3 Amps to the control board to power the rest of the system. These boards will be discussed further in this section.

A. Power Board

Pictured below, in *figure 2*, is the power board schematic. As stated above, the power board handles the AC-DC conversion, power regulation, and charging of the battery.

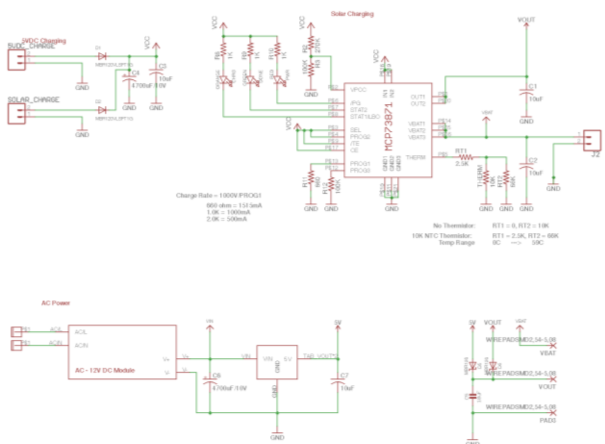

Fig. 2. Power Board Schematic

The battery charging circuit is designed around the MCP73871 [7], which is a smart load-sharing battery charger IC. The chip is designed to allow the input to power the system while charging the battery if enough power is present. This is ideal to allow for time periods where the solar panels are in direct sunlight and able to handle the system power while supplying power to the batteries. The AC-DC conversion is handled with a 30W 12V Mean Well power module, and is further converted down with a 5 Volt output LDO from Texas Instruments (LM1085ISX-5.0) [8]. The layout of our Power PCB is shown in *figure 3* below.

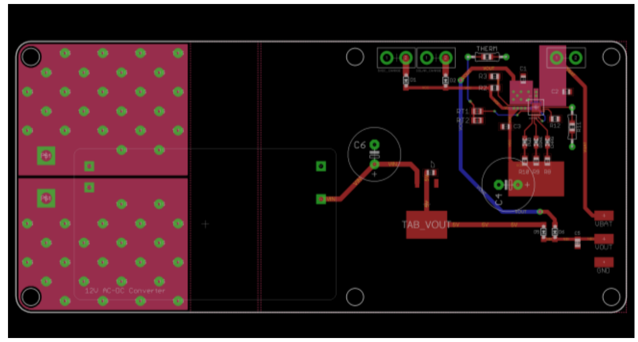

Fig. 3. Power Board Schematic

B. Control Board

Pictured below is the control board schematic. The control board is in charge of connecting the system to the internet and connecting all the peripherals to our SoC microcontroller. In *figure 4* below, the inputs are shown on the left-hand side, while the outputs are shown on the right.

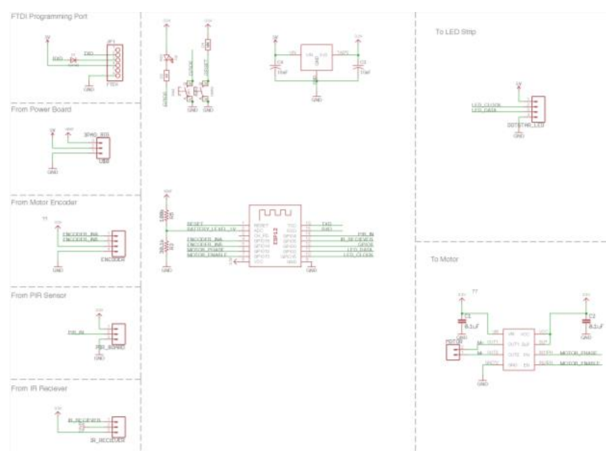

Fig. 4. Control Board Schematic

These inputs include, our programming port, the motor encoder inputs, the PIR input, the IR remote input the input from the power board. The outputs include, the LED strip, and the bridge motor driver circuit which uses the TI DRV8838 to control our motor. The input from the power board then goes through a 3.3 volt LDO voltage regulator (TPS75833KC) to power the MCU, motor, PIR sensor, IR reciever. Below, in *figure 5,* is the layout of our control PCB. Notice that the antenna is placed in a way to prevent interference from the ground plane or other peripherals.

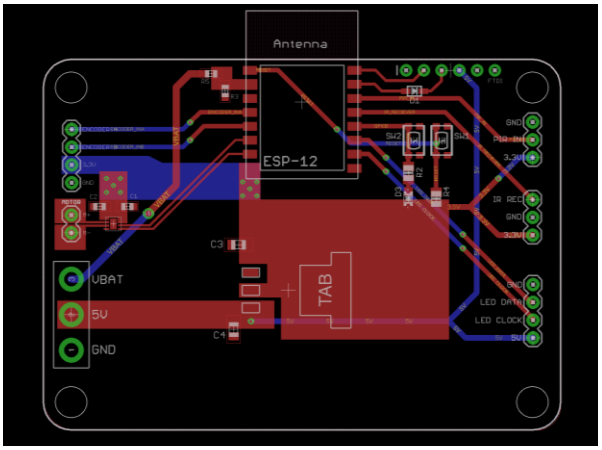

Fig. 5. Control Board PCB Layout

C. *Passive Infrared Sensor Board*

Next is the Passive Infrared (PIR) Sensor board. This board is to be mounted on the front panel of the product behind a specially picked lens allowing for maximum range of the sensor. The main purpose of this sensor is to be integrated with the sponsor companies existing alarm system infrastructure. Below, in *figure 6,* is the PIR board schematic.

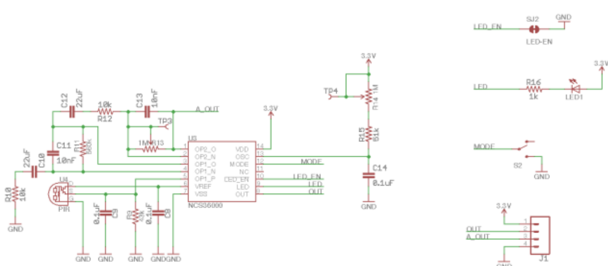

Fig. 6. Passive Infrared Sensor Schematic

The PIR board runs primarily off the NCS36000 [9], which is a chip from ON Semiconductor that amplifies the signal of the sensor and compares the signal with its reference voltage to output a digital high/low signal based on motion being detected. The two potentiometers in the circuit are to fine tune the timing of the digital output and the range/sensitivity of the sensor. *Figure 7* is a picture of the board with the PIR sensor on the top layer, and the rest of the supporting circuitry on the bottom layer of the board.

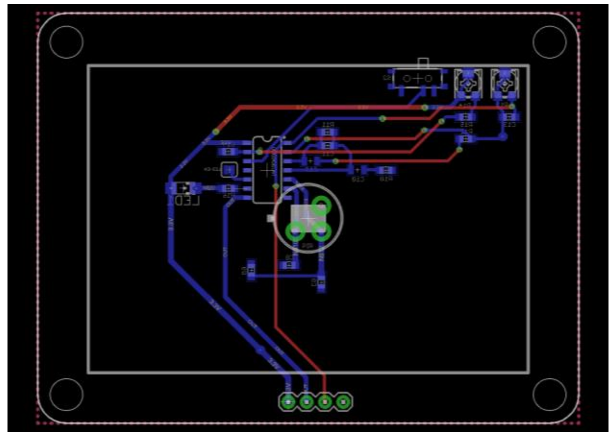

Fig. 7. Passive Infrared Sensor PCB Layout

D. *IR Remote*

Since Wi-Fi connectivity can sometimes be faulty, we designed an IR remote in order to control the shades without using the internet. Below you will see the schematic of the IR remote.

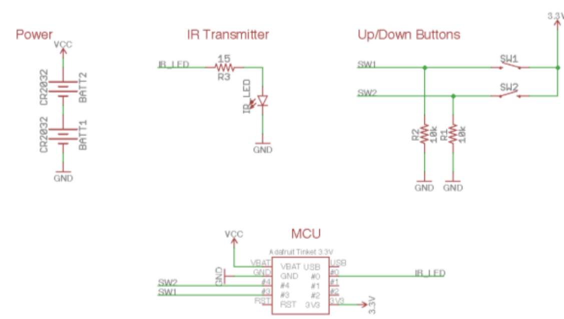

Fig. 8. IR Remote Schematic

The IR Remote is intended to be very simple, just two buttons, an MCU, and two batteries in series. Below is the IR remote board.

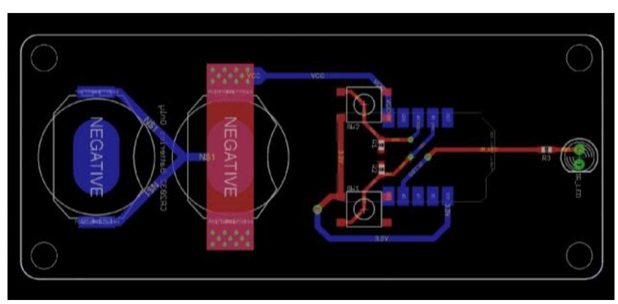

Fig. 9. IR Remote PCB Layout

E. Finished Prototype

Pictured below, in *figures 10 & 11*, is our finished prototype. The prototype is built around a 24x30 inch window. Since this is the first built prototype, a stylistic housing has not been developed. Our intention with this prototype is to develop the electronic hardware and software, and to provide proof of concept for our design. From here our sponsor company can develop a fully-packaged and install-friendly product for them to use in their smart home system.

Fig. 10. Finished Prototype (with Face Plate)

Fig. 11. Finished Prototype (without Face Plate)

In figure 11 above, you can see the roller shade that is attached to our motor on the right. Looking further you can see the battery in the middle of the frame and the two boards mounted on the top of the window frame. The power board is mounted on the far left and the control board on the far right. On the top, is the faceplate with the LEDs, PIR board/Fresnel lens, and IR receiver all mounted to it.

VI. Energy usage

A very important feature for our sponsor was that the system could be powered and controlled wirelessly. This is why our system can be powered using a battery as well as solar and DC charging. Normally, the system will be able to run on battery for multiple days. This battery life is extended by adding solar panels to charge the battery. Ideally, with enough sunlight, the system should be able to provide power for itself indefinitely. For areas where there is not enough sunlight to provide the necessary energy, the user would eventually need to plug the 5 volt DC charging jack into an adapter and charge the shades battery when it becomes low. In this section, we will present some rough calculations proving the sustainability of the product.

On the outside of our window is where our five, 6 volt, 2W solar cells are housed. These cells are wired in parallel to achieve a tested 1.43 amps in direct sunlight at around 7 volts. Since there will be some loss from the solar panel circuitry, we will consider the output 6 volts. In indirect sunlight the current decreases to around 650mA.

To calculate the total power usage of the system we conducted a current measurement across the 5V input to the control board. This was tested to be 110 mA while the system is on standby/waiting for instruction, an average 425mA while the motor is turning up/down, and a varying 340mA – 700mA while the LEDs are running. Since the LEDs draw a relatively large amount of current, we decided to disable the LEDs while the user is on battery power.

To calculate the daily energy usage, we assumed that the user will draw the shades by 100% 4 times daily, taking 8 seconds each time, and that the solar cells will see a total of 4 hours of sunlight each day, 1 of which will be in the direct sunlight. The following equation was used to calculate the total energy consumed by the system.

$$I_{measured} * V_{measured} * t_{hours} \quad (1)$$

$$(0.11 \text{ A}) * (5 \text{ V}) * (23.99 \text{ hr}) + (0.425 \text{ A}) * (5 \text{ V}) * (0.01 \text{ hr}) = \mathbf{13.216 \text{ Wh}} \quad (2)$$

The same following equation was used to calculate the energy being inputted into the system.

$$I_{measured} * V_{measured} * t_{hours} \quad (1)$$

$$(1.43\ A) * (5\ V) * (1\ hr) + (0.670\ A) * (5\ V) * (2\ hr) = \mathbf{13.85\ Wh} \quad (3)$$

Therefore, based on the comparison of the input to the system (3), and the output of the system (2), it can be seen that the system can keep itself fully charged with a minimum of 3 hours of sunlight per day. Something worth pointing out is that our 10,400mAh, 3.7V battery will last approximately 8 years without being recharged at the calculated energy usage. With these considerations, this power solution should be more than sufficient for the system.

VII. Peripherals

Within the Smart shades system there are several peripherals that add to the system's utility. Some of these peripherals satisfy project requirements, while others allow for useful features that the everyday user would appreciate.

To fulfill the physical remote-control requirement, the smart shades system is equipped with an infrared receiver and a physical remote control with an infrared LED. This physical remote control comprises a microcontroller, in this case an Adafruit Trinket, that can read button inputs and output a pulse code to an IR led. Using this IR pulse-code, separate "Up" and "Down" commands can be sent from the remote control and received by the shade system via IR.

After researching, the best solution for a motion detection scheme was determined to be a passive infrared (PIR) detector. This solution is low power, as it effectively consumes no power when not detecting motion. When the PIR detector is outfitted with a Fresnel lens, the detection range is increased dramatically, and to a point that satisfies the 3-meter project requirement.

A smart shades system also comprises a programmable RGB LED strip by Adafruit. This LED strip allows for users to choose amongst a variety of colors and brightness settings, and when used in conjunction with the motion sensor allows for features such as the alarm setting and nightlight setting previously discussed. Additionally, the LED strip facilitates simple user feedback, such as an initialization sequence that allows the user to visually see when the system is fully configured and ready for operation.

The motor that moves the blinds is outfitted with two encoders. These encoders allow the relative position of the shade to be tracked to an accuracy of a quarter rotation of the shaft. This solution is used to precisely move the shades to desired percentage increments of the total window length.

VIII. Software

A. Web Application

Our web application is developed using Model View Controller (MVC) architecture and .NET framework. This allows for the development of user interfaces in separate, but connected, parts. This MVC architecture allows for efficient code reuse. The model portion of MVC is our service layer. That is, the service layer configures, validates, and sends the web request to the microcontroller using REST client. The view portion of MVC represents the user-interface and is developed with a variety of libraries, such as bootstrap, jQuery, and JavaScript. Bootstrap allows the web application to accommodate a visually appealing experience to a multitude of users across a range of devices. jQuery allows us to easily use AJAX requests to send user data from the view to the controller in the MVC. The controller portion of MVC is used to validate requests before they are sent. MVC, as a system, first takes user-input from view, then sends a request to controller, then sends the request to model, which finally sends the request to the microcontroller through the REST client.

In our service layer, an HTTP client provides a base class for sending/requesting the HTTP requests from a microcontroller. We are using this HTTP client to enable our system to handle multiple concurrent requests, which allows us to queue requests while they are still being executed by the shades system.

The various tools mentioned above make the web application possible. The graphical design of the web application can be summarized into four main modules: remote utility, LED lighting, battery status, and automation and settings. The remote utility module consists of an "Up" and "Down" button that moves the shade in 25% increments. The LED lighting module consists of buttons for every color the LED is capable of, as well a slider to adjust the LED brightness, and a power button to turn the LEDs "On" and "Off". The battery status module consists of a button that displays the current battery level. The automation and settings module consists of buttons to enable and disable the various motion detector settings, as well as fields to input times to update "Wake up" and "Bedtime" alarm clock times.

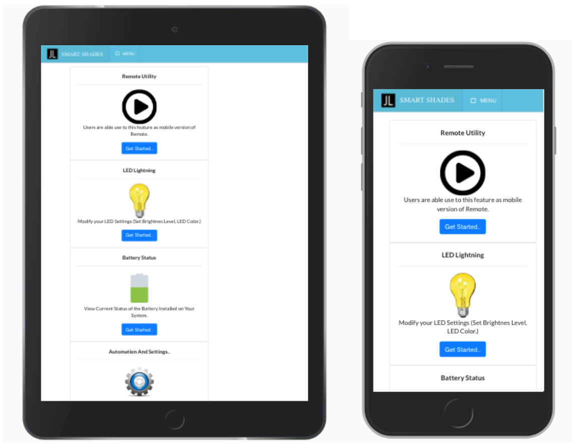

Fig. 12. Web Application Dashboard

B. Amazon Echo

Our product will be fully integrated with voice control using Amazon Echo. To make this possible we needed to first develop the Alexa skill set that will be used to verbally control the device. These skills are the phrases that the user speaks to Alexa, to send the correct command. To set up our own Alexa skills, we log in to the Alexa developer console and create a new skill. The developer console will walk you through the steps that are necessary to complete and build your skill set. Essentially, each different action that Alexa can perform is labeled as an 'intent.' The phrase that our skill model follows is "Alexa, tell my shades to [intent]." A few examples would be; "Alexa, tell my shades to roll down." Or "Alexa, tell my shades to turn blue." After the skill kit is developed and built, the .js file is hosted on Amazon Web Services Lambda cloud computing platform. The .js file can be directly edited from AWS Lambda. The skills that we developed are primarily to; move the shades up and down by 25% intervals, change the color of the shades LEDs, turn the LEDs on and off, and change the brightness of the LEDs.

C. Firebase

The Google Cloud Platform offers a robust, flexible, reliable, and scalable platform for hosting websites. Firebase hosting enables fast and secure static hosting for web apps. Firebase offers features like real-time database, test lab, cloud storage, authentication, cloud functions and remote configuration. Specifically, for the smart shades system Firebase is used to store the current local IP address of the microcontroller. The local IP address of the microcontroller varies depending on the network it is connected to. The web application and Amazon Alexa both require the local IP address of the microcontroller to send commands. Firebase is then used to store the current local IP address of the microcontroller. As such, Firebase allows the web application and Amazon Alexa to pull the current IP address dynamically to ensure that the current IP address is the correct local IP address of the microcontroller. Additionally, Firebase is also used to store the current battery level. This is done so that the current battery level can be pulled and displayed by the web application.

D. Firmware

The hardware programming of the smart shade system was a sizable portion of the software development. The hardware programming was implemented using Arduino. Arduino was chosen primarily because it offered an intuitive programming environment alongside a wealth of relevant documentation. The ease-of-use of Arduino allowed for very rapid testing and development, which in the condensed time frame of senior design, was of great benefit. Firmware was written for all hardware actions that the smart shades system takes. This allows for simple and extendable interactions with the various forms of user input, as every input calls the same appropriate firmware function for each individual action. Thus, parity between all user interactions is inherent in the design of the system.

A critical element of this firmware design was the careful calibration of each function, such that the software functions implement the desired physical action with great precision. A great deal of effort was spent on calibrating the precise movement of the window shades, as real-world resistance on the motor shaft resulted in the need for a scaling method.

IX. Conclusion

Senior Design has provided an exceptional opportunity to experience engineering in a manner that is much closer to a real-world case than the classroom. It also provided experience in technical paper writing in the context of a larger scale project, choosing and sourcing components, and experience working with a group on a prolonged project. All of which are useful skills for a working engineer.

Applying the technical knowledge gained from classes to a real system proved to be a wholly new experience. Proving that theoretical knowledge provides the foundations for systems but troubleshooting and problem solving eventually yield working solutions.

The smart shade system works fully to the specified requirements, and as such is a testament to this group's teamwork, arduous work, and ingenuity.

ACKNOWLEDGEMENT

Our group would like to acknowledge the individuals who, along with out group, helped make this project a success. Christian Jaen (Jetsons Living - CTO) provided the web hosting account for us as well as helped with the Alexa skill development. Ramin Ragbir (UCF ME Student) helped us develop the CAD model of our shades system which we later turned into our finished prototype. Burick Graham (Jetsons Living – Founder/CEO) provided the funding for the project which will be turned over to him and his team for further development after our completion of the senior design course. Dr. Richie for his guidance throughout the entire course of this project.

REFERENCES

[1] Margaret Rouse, Ivy Wigmore, “Internet of Things (IoT) ,” July 2016. [Online]. Available: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.

[2] Wikipedia, "Bluetooth," [Online]. Available: https://en.wikipedia.org/wiki/Bluetooth.

[3] E. Betters and C. Hall, "What is Zigbee and why is it important for your smart home?," [Online]. Available: http://www.pocket-lint.com/news/129857-what-is- zigbee-and-why-is-it-important-for-your-smart-home.

[4] Wikipedia, "Z-Wave," [Online]. Available: : https://en.wikipedia.org/wiki/Z-Wave . [Accessed 15 November 2017].

[5] Elecrow, “ESP-12s User Manual” July, 2016. [Online]. Available: https://www.elecrow.com/download/ESP-12S_User_Manual.pdf

[6] Adafruit, “ESP8266 Datasheet” [Online]. Available: https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf

[7] MicroChip, “MCP73871 – Stand-Alone System Load Sharing and Li-Ion/Li-Polymer Battery Charge Management Controller” Octover 25, 2017 [Online]. Available: http://www.microchip.com/wwwproducts/en/en536670

[8] Texas Instruments, “LM1085 3-A Low Dropout Positive Regulators” January, 2015 [Online]. Available: http://www.ti.com/lit/ds/symlink/lm1085.pdf

[9] ONSemiconductor, “NCS36000 – Passive Infrared (PIR) Detector Controller” December, 2015 [Online]. Available: http://www.onsemi.com/pub/Collateral/NCS36000-D.PDF

John Connor Rossi is a UCF student who intends to graduate with his BS in Electrical Engineering in May 2018. He is currently an intern at Hoverfly Technologies. He intends to further his career and knowledge in the area of consumer electronics.

Stephen Fryer, a senior computer engineering student at the University of Central Florida. His areas of interests include embedded programming, computer hardware, software, and digital circuit design. He plans on pursuing a career in computer engineering that is closely related to his interests after graduation.

Brian Mora, an electrical engineering student at the University of Central Florida who intends to graduate with his BSEE in May 2018. While at UCF, he did research in Computer Vision under the Research Experience for Undergraduates (REU).

Priyank Patel, a senior computer engineering student at the University of Central Florida. He is currently an intern at HealthGrid. He intends to further his career and knowledge in web application development.