

Electrical Engineering Department
California Polytechnic State University

Senior Project Report

Wireless Window Blinds

06/02/2017

Cooper Laone
Zachary Malig
Niraj Morar
Jake Phillips

Advisor:
Professor Richard Murray



TABLE OF CONTENTS

<i>Section</i>	<i>Page</i>
Abstract.....	iv
Chapter 1. Introduction.....	1
Chapter 2. Customer Needs, Requirements, and Specifications.....	2
Chapter 3. Functional Decomposition.....	5
Chapter 4. Project Planning.....	9
References and Citations.....	12
<i>Appendices</i>	
A. Senior Project Analysis.....	16
B. EE 461 Addendum.....	21
C. EE 462 Addendum.....	31
D. Senior Project Expo Pictures.....	39

LIST OF TABLES AND FIGURES

<i>Tables</i>	Page
TABLE I: Project Requirements and Specifications.....	2
TABLE II: Project Deliverables.....	4
TABLE III: Window Blinds Functionality.....	5
TABLE IV: Smartphone Functionality.....	6
TABLE V: ESP8266 Functionality.....	7
TABLE VI: Arduino Functionality.....	7
TABLE VII: DC Motor Functionality.....	8
TABLE VIII: Voltage Regulator Functionality.....	8
TABLE IX: Team Member Responsibilities.....	10
TABLE X: Project Cost Estimates.....	11
TABLE XI: ROB-12281 Micro Gearmotor Current Measurements.....	27
 <i>Figures</i>	
FIGURE I: Customer Gains.....	1
FIGURE II: Level 0 Wireless Blinds Controller.....	5
FIGURE III: Level 1 Wireless Blinds Controller.....	6
FIGURE IV: Wireless Window Blinds Gantt Chart.....	9
FIGURE V: Wireless Window Blinds Gantt Chart (Winter).....	22
FIGURE VI: User Interface “Test Application”	23
FIGURE VII: User Interface “Final Application”	24
FIGURE VIII: Initial Microcontroller to WiFi Module Circuit.....	25
FIGURE IX: ESP8266 Development and Testing Circuit.....	26
FIGURE X: Current ESP8266 Circuit.....	26
FIGURE XI: ESP8266 to L293D Motor Driver.....	26
FIGURE XII: System Implementation V1.....	28

FIGURE XIII: System Implementation V2.....	29
FIGURE XIV: Wireless Window Blinds Gantt Chart(Spring).....	32
FIGURE XV: User Interface.....	33
FIGURE XVI: Motor Controller Schematic.....	34
FIGURE XVII: Motor Controller Circuit.....	35
FIGURE XVIII: Tilt Motor, and Removed Worm Gear.....	36
FIGURE XIX: 3D Printed Spindle Design and Specifications.....	36
FIGURE XX: Motor Setup Version 3.....	37
FIGURE XXI: Version 4 (Final Version)	37
FIGURE XXII: Final Complete System.....	38

Abstract

The Internet of Things asserts that there exists a global need for all objects to connect easily and accessibly. Today, smartphones, tablets, laptops and smart watches makeup just a few of the interconnected devices by the greater population. As a result a growing need for wireless connection between personal devices like phones, and computers to everyday appliances exists. This idea extends directly to households, businesses and buildings, where a growing need for smart home or smart-business appliances has taken root. Many companies like Nest, ADT and Apple make products for automated home security and in home media control. The Wireless Window Blinds project aims to provide smart home technology to window blinds.

The Wireless Window Blinds gives users wireless control over pre-existing wireless blinds. Users, through a mobile device app, have direct control over basic window blind functionality: opening, closing, raising and lowering window blinds. Additionally, users have flexible control over any number of blinds across any number of rooms. The app allows users to easily open and close all their connected blinds remotely. This project strives to create a self-contained system that gives users wireless, automated control of a network of window blinds.

Chapter 1: Introduction

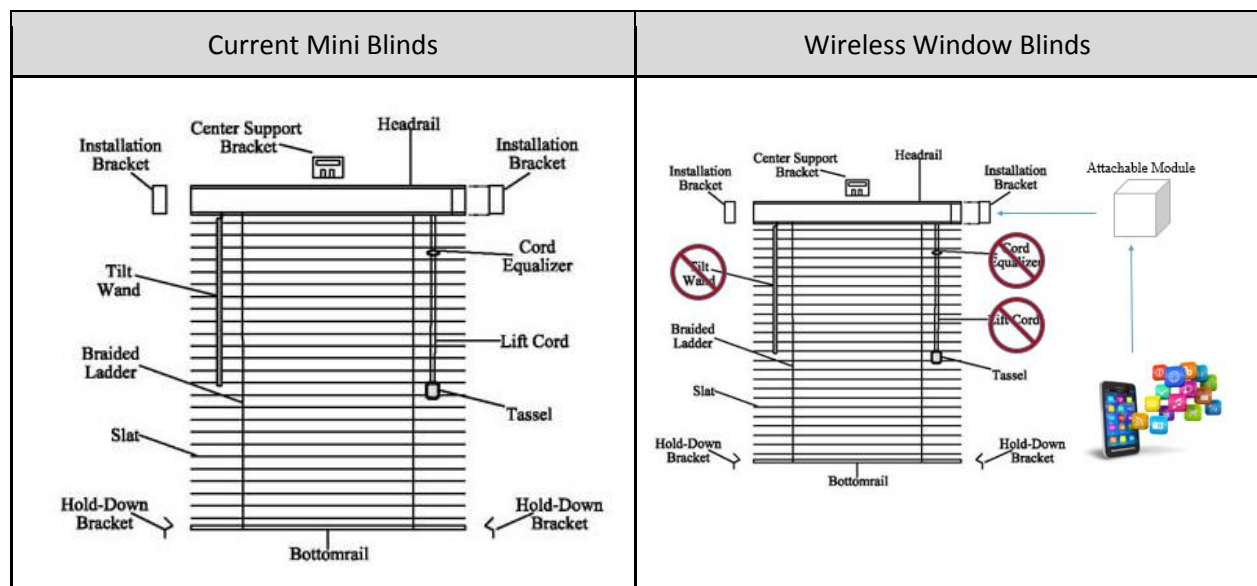
According to Pew Research Center, 64% of adults in the United States owned a personal smartphone in 2015[4]. Due to the tremendous growth of the smartphone and tablet industry, wireless devices have become essential to daily life. Wireless devices provide a widespread platform to fulfil the IoT goal to make all objects connected and easily accessible. Today, many products and services exist to support this need, and have laid a foundation for IoT household connected devices [9, 10, 12]. Many household specific IoT devices, including the Smart Lights with Wi-Fi control, prove reliable resource for developing the software and hardware for IoT devices and applications [8]. This has resulted in a growing industry for smartphone controlled devices and appliances.

The growing need for wirelessly controlled appliances directly impacts homes and businesses. Growing companies like Nest and ADT provide households and office-spaces with smartphone connected thermostats, security cameras and systems [9, 10]. Markets and Market's global forecast predicts the smart-home industry worth 121.73 Billion USD by 2022 [5]. This project aims to break into the smart-home industry with a wirelessly controlled window blinds device.

The Wireless Window Blinds (Wireless Window Blinds) project implements a Wi-Fi controlled solution for automated control of household mini blinds. The project comes in two discrete parts: an attachable module and a smartphone app. The attachable module provides motorized control over generic mini blinds. The smartphone app implements an easy to use interface for control over the mini blinds.

Compared to competitors, our product offers the same reliability and functionality of motorized window blind sets but at a less expensive price. Additionally, our product does not require the customer to replace their current set of blinds, while many competitors require their customers to purchase a new set of window blinds [16]. Additionally, competitor's systems limit their user by requiring a remote to control the window blinds, while the Wireless Window Blinds allows customers to use their smart phone to control the window blinds [16]. Figure I below depicts what customers gain from this product compared to a non-motorized set of window blinds. The Wireless Window Blinds eliminates the need for a tilt wand, cord equalizer, and a lift cord, as well as providing remote control through a smartphone app.

FIGURE I
Customer Gains



Chapter 2: Customer Needs, Requirements, and Specifications

Customer Needs Assessment

We acquired the customer needs through market research of current window blind systems as well as questioning home owners and company owners about what they would want out of a wireless window blind controller. Homeowners with smartphones and Wi-Fi represent the primary portion of the wireless window blinds target customers, so we weighed their response heavier than the company owners. From talking to homeowners, they need an aesthetically pleasing and easy to install system with a good battery life. The smart phone app must provide users with an easy-to-use interface for raising, lowering, and tilting blinds. Additionally, the complete system must scale to support any number of windows the customer may have. Currently, market motorized blinds cost between \$250 to \$300, but many owners who want to upgrade their house search for the cheapest solution.

Requirements and Specifications

Customer needs dictated the Marketing Requirements shown in Table I. Marketing requirements include a sleek, easy to install, and cheap system. Fulfilling these requirements makes our product appealing to customers and competitive in the current market. Additionally, maintenance and upkeep costs should remain low for users, which motivates the need to seldom replace the system's batteries. The Engineering Specifications derived from the Marketing Requirements and external research as summarized in Table I below.

TABLE I
Wireless Window Blinds Requirements and Specifications

Marketing Requirements	Engineering Specifications	Justification
1/8	Controller starts responding to app input controls in less than 1 second	The customer needs a set of blinds that operate in a quick manner.
2	At least 3 month battery life for average user of raising and lower blinds two times a day	Not only do batteries represent an impact on the environment, they also add an avoidable task for upkeep of the blind system.
2	Runs on 9 volts and draws less than 100mA of current	The consumer needs common, easily accessible, cheap batteries found in any store. 9 Volt batteries make up a large portion of available consumer batteries.
2	Complete window blinds system runs continuously for 1 hour.	The consumers need a durable, power efficient system that requires little to no maintenance or upkeep.
3/5/6	An average homeowner's installation time should not exceed 30 minutes.	Market blinds require a short installation time. This system must perform on par to remain competitive.

3	Product must follow the IEEE NESC safety standard.	Devices released to the public must practice safe installation, operation, and maintenance [17].
4	Production cost should not exceed \$75.	Motorized blinds cost between \$250 to \$300. This system must perform on par to remain competitive.
5	Size smaller than 2 in tall x 4 in wide x 2 in deep	Customer needs dictate a demand for a hidden system to not represent an eyesore.
6/7	Calibration tests allow the system to run on many horizontal mini blinds systems.	Customers own a wide variety of Window Blind types. To account for this, our system must fit and work on a diverse group of blinds.
1/8	Communicates with smartphone app remotely using Peer to Peer Wi-Fi Data transfer.	Customers with in-home Wi-Fi need an easy-to-use, convenient interface to control their window blinds system [9, 13].
Marketing Requirements <ol style="list-style-type: none"> 1. Responds to remote raise, lower, and tilt commands quickly 2. Battery powered 3. Easy and safe to install 4. Inexpensive 5. Looks sleek/hidden 6. Attaches to multiple window blind sets 7. Durable 8. Allows user to control window blinds remotely 		

Table II lists deliverables as well as delivery dates for product flow. These dictate the outline for the product's design and production. These follow the Cal Poly EE 460/461/462 requirements to complete the Senior Project three part course.

TABLE II
Wireless Window Blinds Deliverables

Delivery Date	Deliverable Description
Feb 16	Design Review
March 2	Beta Prototype done
March 13	EE 461 demo
March 13	EE 461 report
June 2nd	EE 462 demo
End of May	ABET Sr. Project Analysis
June 2nd	Sr. Project Expo Poster
June 7th	EE 462 Report

Chapter 3: Functional Decomposition

Block Diagram

The level zero block diagram appears below in Figure 1. It outlines the overall inputs and outputs of the system. The inputs to the system include user input from the smartphone app and the connection to WiFi between the app and controller. The output of the system includes a pulse width that changes the orientation of the window blinds based on the user input. Additionally, the smartphone app connects to multiple controllers allowing network control of any connected window blinds in the home. The implementation of the network software and structure takes consideration of the patent on connection of multiple IoT devices [6, 7, 10].

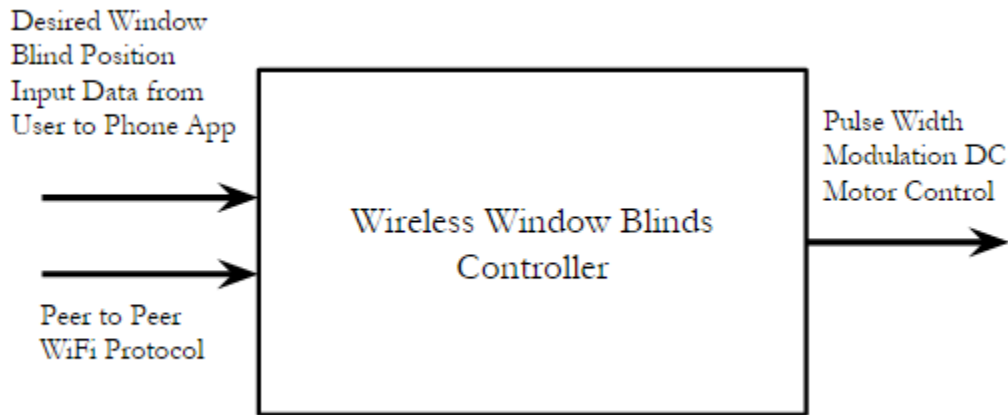


FIGURE II
Level 0 Wireless Window Blinds

Table III describes the Level 0 block diagram. The parameters of the inputs and outputs of the Wireless Window Blinds become further characterized. Additionally, Table III describes the Level 0 block diagram's functionality.

TABLE III
Wireless Window Blinds Functionality

Module	Wireless Window Blinds
Inputs	Desired window blind position input data from user to phone app: User works with app by choosing requested blind position Peer to Peer Wi-Fi Protocol: Wi-Fi connection to wireless controller and phone
Outputs	Pulse Width Modulation DC Motor Control: 3.3 V pulse length defined motor signal that sets the blinds to the user desired position.
Functionality	Adjusts blinds to user's app input specifications. Input specifications travel from phone app to physical controller via Wi-Fi. When no user request signal occurs, no motor control happens.

The Level 1 Block Diagram shown below in Figure III describes each module of the Wireless Window Blind Controller in more detail. Each module is further dissected in the respective Table below listing its inputs, outputs, and functionality. Table V below gives a functional description of the Wi-Fi module, ESP8266, including data exchange [12, 13]. Table VI below describes the functionality of the Arduino microcontroller, with reference to Javed's, *Building Arduino Projects for the Internet of Things* [11]. Table VII provides a functional description of the DC motors used to motorize the window blinds system with reference to *Motorizable Tilt Shade System and Method* [14].

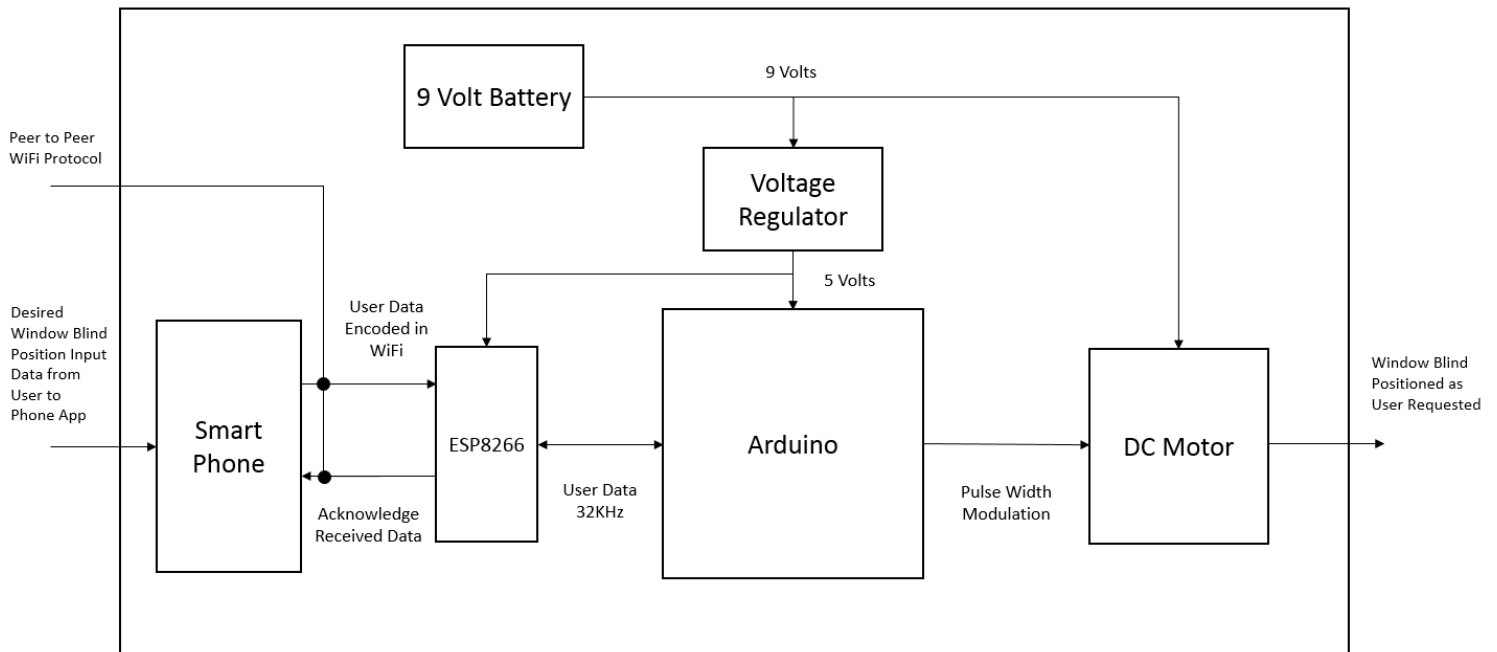


FIGURE III
Level 1 Wireless Window Blinds

TABLE IV
Smart Phone Functionality

Module	Smart Phone
Inputs	Desired window blind position input data from user to phone app: User works with app by choosing requested blind position Acknowledge Received Data: Acknowledge that the controller received the data, and the request does not need to resend
Outputs	User Data Encoded in Wi-Fi: Using Wi-Fi, sends a packet containing the user's desired window blind position
Functionality	Wirelessly communicates from the user to the controller

TABLE V
ESP8266 Functionality

Module	ESP8266
Inputs	<p>User Data Encoded in Wi-Fi: Decodes User data from Wi-Fi</p> <p>User Data at 32KHz: Arduino can send data to the ESP8266 to be sent to the Smart Phone notifying user of action complete, or if any errors occurred.</p>
Outputs	User Data at 32KHz: Passes the signal containing the desired window blind position from Wi-Fi to the Arduino to be processed.
Functionality	Deciphers Wi-Fi encoded data into an Arduino friendly data form.

TABLE VI
Arduino Functionality

Module	Arduino
Inputs	<p>5 Volts: Provides 5 Volts with capabilities of supplying 150 mA of current.</p> <p>User Data at 32KHz: Data containing the desired window blind position for the Arduino to decode and convert to DC Motor Control</p>
Outputs	Pulse Width Modulation: 3.3 Volt Pulse Width Modulation that controls a DC motor to move the Window Blinds into the Desired Position
Functionality	Is the Central Processing Unit of the Wireless Window Blind Controller that processes the requests and turns it into motor control.

TABLE VII
DC Motor Functionality

Module	DC Motor
Inputs	<p>Pulse Width Modulation: Moves the motor a certain amount of degrees based on the duty cycle of input wave.</p> <p>9 Volts: Powered by 9 volts that can supply up to 150 mA.</p>
Outputs	Window Blind Positioned as User Requested: Physically moves the blinds a certain amount based on the PWM input.
Functionality	Physically moves the window blinds into the position that the user requested.

TABLE VIII
Voltage Regulator Functionality

Module	Voltage Regulator
Inputs	9 Volts: takes in 9 volts from a battery.
Outputs	5 Volts: Steady 5 volts that cannot be loaded easily
Functionality	The Wireless Window Blinds's modules require different amounts of voltage to power them. The Voltage Regulator supplies these different voltages.

Chapter 4: Project Planning

Gantt Chart

The Gantt chart below shows a design approach to the Wireless Window Blinds. This chart gives detailed tasks and milestones from EE 460 through EE 462 dictating the flow of the project. It includes multiple project iterations as well as the EE 461 and 462 Report and Demo deadlines.

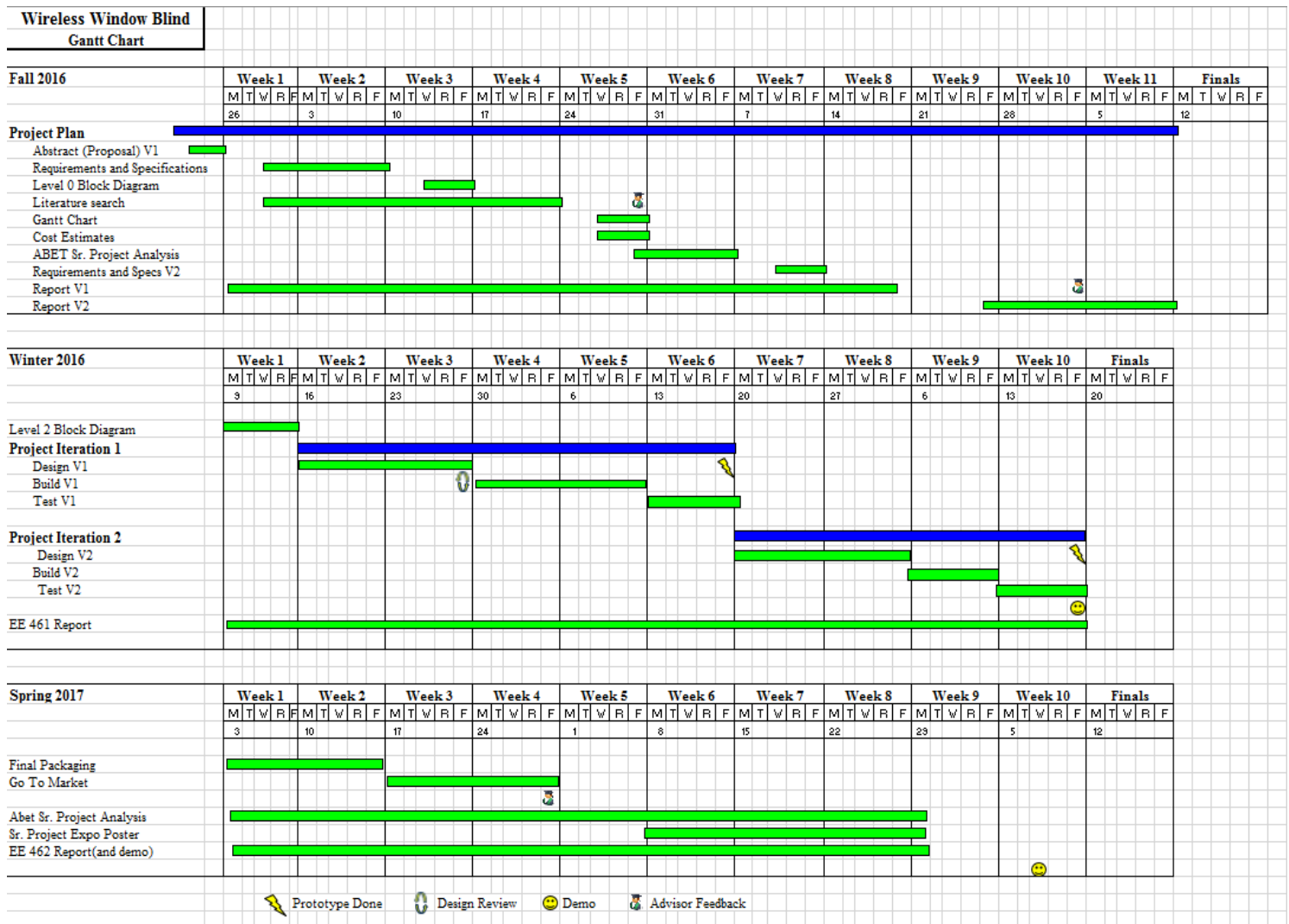


FIGURE IV
Wireless Window Blinds Gantt Chart

Individual Responsibilities

During the project's Design, Build, and Test phases each member has a unique responsibility to certain tasks. While our work must interconnect, each member's role fits their knowledge and skill set, shown below in Table IX. Note that all responsibilities not listed in the table below become split evenly among team members. For example, all team members helped write the EE 461/462 Report.

TABLE IX
Team Member Responsibilities

Role	Responsibility	Team Member
Manager	Checks that each member's work meets given deadlines, all work meets product specifications, and design specifications interconnect with other member's work.	Cooper Laone
Sales / Outreach	Looks for potential customers of the product along its lifeline.	Zachary Malig
Wi-Fi Firmware Development	Writes Arduino code for the Wireless Window Blinds to connect to Wi-Fi and communicate with the customer's Smart Phone App.	Cooper Laone
Motor Firmware Development	Writes Arduino code for the Wireless Window Blinds to control the motors based on the customer's request.	Zachary Malig
Smart Phone App	Writes an iPhone and Android version of an app to properly connect to the Wireless Window Blind Controller and send window blind position requests.	Niraj Morar
Motor Setup	Configures a set of Arduino controlled motors to connect to a set of mini-blinds that can raise, lower, and tilt the blinds.	Jake Phillips
Case Holding Product	Create an aesthetically pleasing case to hold and mount the product on a set of mini-blinds.	Jake Phillips

Cost Estimates

Table IX shown below lists the items, quantity, and costs associated. This project requires at least two sets of window blinds to create a network of blinds all controllable from a single smart phone. Since each member of the group already owns a smart phone, this item added no additional cost. The estimated Labor Costs arose from Ford and Coulston's Cost Model^[3] averaging the worst case (12 hours per week), best case (3 hours per week), and most likely case (8 hours per week). This led to 7.8 hours per week with four members working together over three quarters. 30 weeks multiplied by 4 people and 7.8 hours per week led to 900 Labor hours.

TABLE X
Wireless Window Blinds Cost Estimates

Item	Quantity	Cost	Total
Arduino	2	\$20.00	\$40
Window blind set	2	\$20.00	\$40
Wi-Fi Card	2	\$5.00	\$10
Miscellaneous Circuit Components	x	x	\$20
DC Motors	4	\$5.00	\$20
Work Hours	900	\$25.00	\$22,500
			\$22,630

REFERENCES AND CITATIONS

Literature Search

1. A. Javed, *Building Arduino Projects for the Internet of Things*, Lake Zurich, IL: Apress, 2016.

I chose this book because it describes the fundamentals of creating an IoT project using the Arduino. This directly relates to how the creating of the Wireless Window Blinds, so the advice and suggestions in this book should prove valuable over the course of the project. Additionally, the book delves into Android App development with IoT devices, which represents another major key of the Wireless Window Blinds.

This book has authority because it contains experiments with real world tested examples and examples proven to work on Arduino devices. Additionally, Adeel Javed has written a multitude of articles for companies like IBM on how IoT devices work and what the future of IoT looks like.

2. Nicola Bui, Angelo P. Castellani, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi, "Internet of Things for Smart Cities", *IEEE Internet of Things Journal*, vol. 1, pp. 22-32, 2014, ISSN 2327-4662.

I chose this source because it describes the urban Internet of Things and the idea of a Smart City, which both correlate to the Wireless Window Blinds. Additionally, this article goes into technical detail on an IoT's services, data packets, and connect-ability, which our project utilizes for sending and receiving data from the app to the controller.

This source has the authority to provide reliable information because IEEE published it in the highly peer reviewed and accredited journal *IEEE Internet of Things Journal*. If the authors made any mistakes or untrue statements, the peer reviewers would have caught these and brought it to the authors' attentions.

3. Espressif Systems, "Espressif Smart Connectivity Platform: ESP8266," ESP8266 datasheet, Oct. 12, 2013.

I chose this datasheet because many low power Wi-Fi using devices recommend the ESP8266, as well as our advisor recommended using it. While this does not guarantee our usage of the ESP8266, it contains valuable knowledge of how to connect to an Arduino via Wi-Fi.

This source has the authority to provide reliable information because the publisher, Espressif, has a good reputation for creating products and datasheets. Additionally, Source [6] directly references this product. Had any major errors occurred in the creation of the datasheet, the company would hold responsibility and we could contact them as a customer.

4. R. Y. Kim, O. Zeira, "Setup of multiple IoT network devices," US 20160044032 A1, Feb 11. 2016.

I chose this patent because it describes various techniques to set up a network of IoT devices. This relates heavily to our project, because many customers have multiple window blinds that need to be controlled on the same network. This project can implement a system similar to the one Kim and Zeira describe while attempting to improve on it.

This patent has authority to provide reliable information because the Applicant for the patent, Belkin International, Inc., symbolizes a successful electronics company that specializes in networking. Their products include routers and network switches, which represent the fundamentals of large scale networking.

5. Parul Kalra, Deepti Mehrotra, and Navjot Walla, "An IoT by information retrieval approach: Smart lights controlled using WiFi", *2016 6th International Conference - Cloud System and Big Data Engineering (Conference)*, Jan. 2016.

I chose this source because it describes an implementation of an IoT product very similar to the Wireless Window Blinds. They implement Smart Lights with Wifi, giving valuable insight on what hardware and software implementation works well.

This source has the authority to provide reliable information because they completed a project using methods similar to how we hope to implement the Wireless Window Blind Controller. While this article has not been recognized for any awards, it has hands on practical application knowledge that our project can use.

6. W. J. Mullet, Y. Rodriguez, D. W. Brunk, R. S. Hand, "Motorizable tilt shade system and method," US 20120111509 A1, May. 10, 2012.

I chose this patent because it recognizes a system similar to the mechanical portion of our Wireless Window Blinds. Our mechanical pulley system can learn and improve on the design dictated by this patent, as well as include our wireless implementation.

This patent has authority to provide reliable information because the inventor successfully implements a motorized blind system, which falls under the scope of this project. Our project must make a motorized blind system, and then develop a controller for it, making their patent a building block for our project.

7. K. Sood, S. Yu and Y. Xiang, "Software-Defined Wireless Networking Opportunities and Challenges for Internet-of-Things: A Review," in *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 453-463, Aug. 2016.

I chose this article because it discusses software-defined networking with IoT devices in mind. This project aspires to create a wireless network of IoT devices, and the advice from Sood, Yu, and Xiang helps our project integrate a wireless network.

Similar to Source [2], this article has authority to provide reliable information because of its highly peer reviewed nature due to its publication in the renowned journal *IEEE Internet of Things Journal*. Additionally, multiple other papers have cited this article for its research.

8. Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen and S. Shieh, "IoT Security: Ongoing Challenges and Research Opportunities," *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, Matsue, 2014, pp. 230-234.

I chose this IEEE Conference Article because it discusses the impact of IoT devices every day and home applications. Security represents a major concern for this project and all IoT devices, which this article sheds light on.

This article has authority to provide reliable information because its authors have written papers published in IEEE journals on the same IoT subject. Additionally, many of the references have over 200 citations, proving the validity of their research material.

9. L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, 2010, pp. 2787-2805.
I chose this article because it discusses the many ways IoT devices can integrate into society. Different technologies give different advantages, and each has its own special niche. It describes the different visions of an IoT world, and what its development can mean for society.
This article has incredible authority to provide reliable information because other papers and articles have cited it over 4400 times according to Google Scholar.
10. A. M. Fadell, M. L. Rogers, Y. Matsuioka, D. Sloo, M. Veron, S. Honjo, "Devices, Methods, and Associated Information Processing for Security in a Smart-Sensored Home," U.S. 2014 0266669A1, Sep. 18, 2014.
I chose this patent because it represents a successful application from one of the biggest smart home companies: Nest. It discusses ways for devices to process information in a smart home with security in mind. Similar to why I chose Source [8], security remains a huge factor in IoT devices and our project aspires to give IoT device users as much privacy as possible.
This patent has authority to provide reliable information because the applicant, Nest, represents one of the biggest market holders for smart home devices. Their company's multitude of Smart Home products speak for their authority in the subject of IoT devices.

Citations

- [1] R. Ford and C. Coulston, *Design for Electrical and Computer Engineers*, McGraw-Hill, 2007, p. 37
- [2] *IEEE Std 1233, 1998 Edition*, p. 4 (10/36), DOI: 10.1109/IEEESTD.1998.88826
- [3] R. Ford and C. Coulston, *Design for Electrical and Computer Engineers*, McGraw-Hill, 2007, p. 205
- [4] Aaron Smith, *U.S. Smartphone Use in 2015*, Pew Research Center, 2015
- [5] *Smart Home Market worth 121.73 Billion USD by 2022*, Markets and Markets, 2016.
- [6] R. Y. Kim, O. Zeira, "Setup of multiple IoT network devices," US 20160044032 A1, Feb 11, 2016.
- [7] K. Sood, S. Yu and Y. Xiang, "Software-Defined Wireless Networking Opportunities and Challenges for Internet-of-Things: A Review," in *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 453-463, Aug. 2016.
- [8] Parul Kalra, Deepti Mehrotra, and Navjot Walla, "An IoT by information retrieval approach: Smart lights controlled using WiFi", *2016 6th International Conference - Cloud System and Big Data Engineering (Conference)*, Jan. 2016.
- [9] L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, 2010, pp. 2787-2805.
- [10] A. M. Fadell, M. L. Rogers, Y. Matsuioka, D. Sloo, M. Veron, S. Honjo, "Devices, Methods, and Associated Information Processing for Security in a Smart-Sensored Home," U.S. 2014 0266669A1, Sep. 18, 2014.
- [11] A. Javed, *Building Arduino Projects for the Internet of Things*, Lake Zurich, IL: Apress, 2016.
- [12] Nicola Bui, Angelo P. Castellani, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi, "Internet of Things for Smart Cities", *IEEE Internet of Things Journal*, vol. 1, pp. 22-32, 2014, ISSN 2327-4662.
- [13] Espressif Systems, "Espressif Smart Connectivity Platform: ESP8266," ESP8266 datasheet, Oct. 12, 2013.
- [14] W. J. Mullet, Y. Rodriguez, D. W. Brunk, R. S. Hand, "Motorizable tilt shade system and method," US 20120111509 A1, May 10, 2012.
- [15] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen and S. Shieh, "IoT Security: Ongoing Challenges and Research Opportunities," *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, Matsue, 2014, pp. 230- 234.
- [16] *Motorized Blinds and Shades – Easy Open and Close*, Blinds.com, 2016.
- [17] *2017 National Electrical Safety Code*, IEEE Standards, August 2016.

- [18] O, Thomas. "What Options Are There For Programming MCU's In Production?", *Electronics.stackexchange.com*. Web. 15 November, 2016.
- [19] R. Nidumolu, C.K. Prahalad, M.R. Rangaswami, "Why Sustainability Is Now the Key Driver of Innovation," Harvard Business Review, September 2009, pp57-64.

Additional References

- [20] [ESP8266-01 Datasheet](#)
- [21] [L293DNE Datasheet](#)
- [22] [ROB-12281 Datasheet](#)
- [23] [Arduino Uno Datasheet](#)
- [24] [CP2102 Datasheet](#)
- [25] [ESP Wiki](#)
- [26] [Android Development Help](#)
- [27] [ESP8266 Development Help](#)
- [28] [Arduino Development Software](#)

APPENDIX A – ANALYSIS OF SENIOR PROJECT DESIGN

Wireless Window Blinds

Cooper Laone:

Zachary Malig:

Jake Phillips:

Niraj Morar:

Rich Murray:

1. Summary of Functional Requirements

The Wireless Window Blinds project creates a window blind system that can be controlled via a smartphone app. It fits normal blinds and implements an individually operable window blind network. The app allows users to easily open and close their blinds remotely. The goal of this project is to create a self-contained system that gives users wireless, automated control of a network of window blinds.

2. Primary Constraints

Price represents the main constraint for the Wireless Window Blinds. To stay competitive, the product must cost less than \$75. Wireless communication restrains this product by requiring Wi-Fi communication from a smart phone to the controller [11]. This challenges the battery life, as Wi-Fi requires more power than other communication methods such as Bluetooth. While many microcontrollers could work for this project, our decision to use the Arduino impacts the price of the physical product, but decreases the labor hours required to assemble the device. Additionally, choosing to only have our system fit on horizontal 'mini-blinds' represents a major limiting factor for the system and our customer base.

3. Economic

The Wireless Window Blinds negatively effects the Natural Capital of the earth. To reduce this impact, we chose to use as little packaging as possible and only enough materials to satisfy the customer's needs. Similarly, if we mass produced this product for homeowners, it would provide and need Human Capital via jobs to assembling companies, shipping companies, and companies that produce parts used by the Wireless Window Blinds. The Financial Capital would arise through homeowners interested in IoT connected households purchasing our product. Real Capital impact occurs through similar means that Human Capital came about. The more companies involved in the production, advertising, and marketing of the Wireless Window Blinds, the more Real Capital would take place.

Cost benefits accrue after the release of the product and after making \$22,630. If each module sells for \$75, and \$25 goes towards profits, selling 900 units breaks us even and any additional sales are profit. Partnering with a home-renovation/IoT company such as ADT represents a long term goal for a major avenue for economic benefit from our product. These profits split between shareholders. After the project ends, the Wireless Window Blinds marketing and advertising campaign begins.

Additional equipment cost for testing does not affect the overall price of the product significantly, as the majority of costs come from engineering labor. Funding comes from the engineers and adviser based on the product's test equipment needs.

4. If manufactured on a commercial basis:

Researchers from MarketsandMarkets.com expect the Smart Home market to exceed a \$100 billion industry by 2020 [5]. From this, the Smart Blind market holds approximately \$2 billion [5]. Each device costs \$75, giving \$35 of profit for each device sold and \$40 into manufacturing costs. We expect to sell 1000 units in the first year, so profits from year one estimate to \$35,000. Additional yearly profits depend on previous year's results, but should steadily increase until the release of a new version.

Installation time for the customer after purchase should not exceed 30 minutes on average. The only recurring cost for the customer is replacing the battery every few months.

5. Environmental

This product affects the environment through using its limited resource pool to create our system. Batteries represent one larger, less recyclable piece of our product. We recommend the customer use a rechargeable battery, so the Wireless Window Blinds produces less waste. Additionally, to reduce this impact, we would build and package our product with the least amount of materials as possible. Once the product's assembly is complete, the product itself represents no more significant resource and ecosystem impacts. However, transportation for the product from the assembler to the customer uses fossil fuels and emits pollution. Pollution impacts have proven to affect many species via global warming.

6. Manufacturability

The wireless Window Blinds requires an assembly line for manufacturing. It requires each component shipped to the assembler's location, each component soldered on a printed circuit board, the microcontroller's code downloaded, and the completed circuitry placed into its enclosure [18]. Then the product must power up and demo successfully before placing the total product into a secure, shippable container.

7. Sustainability

The customer's responsibility to maintain the completed system solely includes replacing the battery. To reduce the resources impacted, we recommend the customer use a rechargeable battery. Some upgrades to the system include implementing multiple types of wireless communication and using a different microcontroller to reduce costs and increase speed. However, these upgrades require a large amount of labor for the design and implementation. Keeping these future upgrades in mind while building the current system makes it easier to shift from our current system to a possible future system. Nidumolu, Prahalad, and Rangaswami argue that sustainable development represents a huge selling point for companies today [19]. Starting a design with sustainability in mind can help to minimize the effects of our product in the long run. One step in the right direction they mention is having a plan to do business with eco-friendly business partners and choosing sustainable parts from the conception of the product.

8. Ethical

Analyzing this product using the IEEE Code of Ethics shows that this product follows their guidelines. This product minimally effects safety, health, and welfare of the public as discussed in section 9 below. If any factors arose that did endanger the public, we would promptly disclose them. Additionally, the Wireless Window Blinds attempts to improve the understanding of technology, and accept criticism as to improve the product. This product's design and manufacturing does not break any of the IEEE Code of Ethics rules. Additionally, we intend to attempt to mitigate any misuse of this product through using proper cyber security techniques.

Analyzing this product using the Golden Rule ethical framework can deduce whether this product's ethical implication positively or negatively affects society. If we put ourselves in the place of the customer and bought this product, we would want it to do its job, not break quickly, and not overcharge us. This product does not attempt to trick customers into buying it. It meets the customer needs and adheres to the product specifications. Therefore, this product positively affects society using the Golden Rule ethical framework. Additionally, as customers of this product, we would want each individual involved in the production of the product to have a fair share. We would want the manufacturing companies to not take advantage of its employees, and all parts legally obtained. Through research and carefully picking our partners, the Wireless Window Blinds attempts to meet each of these needs.

9. Health and Safety

The Wireless Window Blinds' installation instructions, use instructions, and maintenance guides all follow IEEE's NESC [17]. The Wireless Window Blinds moves relatively large sized window blinds, which could represent a safety concern to small objects such as fingers or small pets near the blinds while they rise and fall. Should an installation malfunction occur, the blinds could drop from open to closed quickly, representing a safety hazard for people directly below the blinds. Installation while DC motors are powered could cause harm to user or damage window blinds system. Setup and maintenance should take certain consideration to information on motorized window blinds [14]. To reduce these safety hazards, the packaging provides proper warning to alert users. Users should follow proper installation instructions and maintenance of the battery as specified. To reduce health issues, the manufacturer should not use lead based solder.

10. Social and Political

One political issue includes mass production of devices which use cheap labor. Some countries, like China, provide labor doing menial tasks at much cheaper rates than the USA. Choosing between livable wages and profits should not represent a hard decision. Therefore, the Wireless Window Blinds should not exploit underpaid workers.

Additionally, professionals in IoT fields review many IoT devices poorly because they represent a security issue. Hacking an IoT device impacts customer safety and privacy, therefore, we hold the security of this device paramount [15].

This project impacts the Internet of Things market as well as Smart Home companies. Our product enters the market and challenges current IoT window blind companies, negatively impacting their sales and, therefore, employees. This creates inequities for our competitor's employees. However, the customers (homeowners) benefit from a product that meets their requirements for less money. People who worked on the design aspect of this project benefit equitable to the amount of work put in. Further stakeholders benefit equally regarding the resources they provided for the product's development.

Some more obscure stakeholders include all the companies that work with us during the manufacturing and shipping of the product. Manufacturers enjoy our business, and to transport the completed product from manufacturing to customers we enlist a shipping company. Because the manufacturers and shipping companies have our business, they can hire more employees, meaning more people can move to the cities they work in. More people moving to a city yields more people dining, shopping, relaxing, and overall money flow. This shows that our product's stakeholders involves entire cities that we interact with.

11. Development

Throughout EE 460, 461 and 462 we learned how to design and implement a complete project from the ground up. We created a working remotely controlled window blinds module that raised, lowered and

tilted a window blinds system. We learned how to connect wireless devices, and send/receive commands between the ESP Wi-Fi module and a smartphone. Additionally, we improved our skills in making smart phone app using Android Studio. We also developed firmware geared towards a low power system utilizing Wi-Fi as its means of communication. We built a mechanical system to raise, lower and tilt the blinds that attached externally to the window blinds. The biggest learning experience from this project included: architecting an entire product from the ground up, working with Wi-Fi communications, designing a mechanical system and working as a team of engineers.

Sources used for developing the Wireless Window Blinds appear in the Literature Search section. From the patents found in the literature search, we learned many different possible implementations of mechanical window blinds and wireless controllers. Additionally, the literature search shows standards and requirements for marketable products in homes. Finally, the datasheets found in the literature search allow us to correctly interact with the individual components bought to make this product.

12. Future Steps

Looking forward this project could be expanded and improved on in a variety of ways. Taking our base module and application there is a lot of room for growth for future projects. First, the Android application could be converted to iOS. If this product were to be sold in stores, having the application in iOS as well as Android would satisfy a larger user base and make the overall product more appealing. Within the application, one could expand the current app to allow groupings of blinds. This feature would allow users to control multiple blinds at once and manage groupings of blinds by room or location. Many home owners have multiple sets of blinds in one room and having the ability to manage many blinds at a time would save the user time and effort. Another feature that would save time and effort for the user would be to add presets, like raising or lowering the blinds to a specific preset position or tilt. For example, this feature would allow a user to automatically tilt their east facing windows by selecting one preset. Additionally, the application could support more tutorials and settings to answer any potential questions the user might have. Also, adding a feature that allows users to set timers to have their blinds raised or lowered on specific days at specific times would give the customer the ability to have their blinds controlled automatically.

This project could also be taken on by a mechanical team. One major improvement for the mechanical system would include creating a compact 3D printed gear system that takes less space than the current module. A smaller module would be able to provide the same gear ratio while looking sleeker and less obstructive for the customer. Additionally, experimenting with different motors to make the raising and lowering of the system faster and less taxing on the batteries would be a huge improvement for users. Our current motors were chosen due to their small size and compact nature, but through testing we have found drawbacks in both current draw and speed. Experimenting with additional motors would provide useful insight into creating the best system. Another future idea includes making the module attachable and self-contained, so that users can easily attach the entire motorized module to any existing blinds unit. Some additional safety features could be added that would protect the blinds from attempting to move when being interfered with. For example, if an object were to prevent the blinds from raising, a slip-clutch could detect this and proper precautions could then be enacted.

Lastly, some goals for the firmware and electrical hardware include implementing a calibration protocol so that users can command the blinds to move to a specified position. A calibration protocol was created, but was never implemented due to a limitation on time and resources. This would allow the user to choose a window blind position in terms of percentage raised/lowered instead of manually pressing the raise or lower button. The firmware could also be developed onto a PCB to make the design smaller and more compact. This would likely be a last step to be performed after designing a housing for the entire system. Another goal includes researching and implementing sleep modes for the ESP to conserve power. While the Smart Phone application is closed, no commands will be sent to the ESP8266, therefore it does not need to use power listening for communication. This would expand the battery life of the system and reduce

reoccurring costs for the user. Some other IOT related future goals include a security protocol increase for remote accessing and control. Currently the system uses Telnet to communicate from smartphone to ESP8266. This protocol is easy to use but also easy to eavesdrop on. By applying some sort of encryption between the sender and receiver the product would better secure the customer's data such as their Wi-Fi password.

APPENDIX B – EE 461 ADDENDUM

Summary Statement

During EE 461 the Wireless Window Blinds evolved from a set of specifications, requirements, and block diagrams into a functioning project that moves window blinds wirelessly. Through the quarter we divided the project into three different development teams: Mechanical, Software, and Firmware.

The Mechanical Team built a system focused on raising, lowering and tilting the mini blinds. Its main goals were to finalize motor selection and to build a working mounted prototype. Currently we assembled a mounted prototype that accomplishes raising and lowering the blinds. The most challenging aspect included creating a mechanical system that raised and lowered the blinds without getting stuck or tangled. Additionally, it was challenging mounting the system in a way that did not interfere with the movement of the blinds and provided the least amount of resistance while raising the blinds. Our future goals are to implement the tilting mechanism for the window blinds as well as make the system smaller and less noticeable for the customer.

The Software Team developed a smartphone application for the user to wirelessly control the window blinds. The main objectives for the application were to create a basic UI and establish wifi connection protocols with the ESP. Currently a barebone Android application has been developed and our goal for the future is to expand our current app as well as creating an iOS app.

The Firmware Team developed a code base for the ESP8266 WiFi module that tethers the smartphone application to the mechanical system. Currently the module has the ability to connect to WiFi, communicate with the app, and deliver commands to control the motors. The most rewarding and challenging part of the development was moving past the initial learning curve of working with a new, unknown module. Our future goals are to include a calibration phase for setup and implement a more secure communication protocol.

Gantt Chart

The Gantt Chart shown below describes the process our team took in order to achieve our current progress. The first week was used to establish the teams and expectations to be done by one another. After the first week, we split up into three separate teams: mechanical, software, and firmware. Throughout this quarter a large amount of work was accomplished on the first iteration of our project. The mechanical team built a system to raise and lower weights, but was unable to complete development. The software team was able to create a bare-bone application which has a basic UI along with communication with the ESP8266 via Wi-Fi and is currently developing Peer-2-Peer. The firmware team was able to develop firmware for the motor driver and begin communication with the application through Wi-Fi, but still needs to be further developed. Major milestones were achieved during Week 6 when the software and firmware teams had shown communication with one another over Wi-Fi, and Week 7 when the mechanical team was able to lift and lower weights with their first system. Future goals are to finish our initial prototype at the start of Spring Quarter 2017 and then move onto a second iteration up until the Senior Project Expo in May.

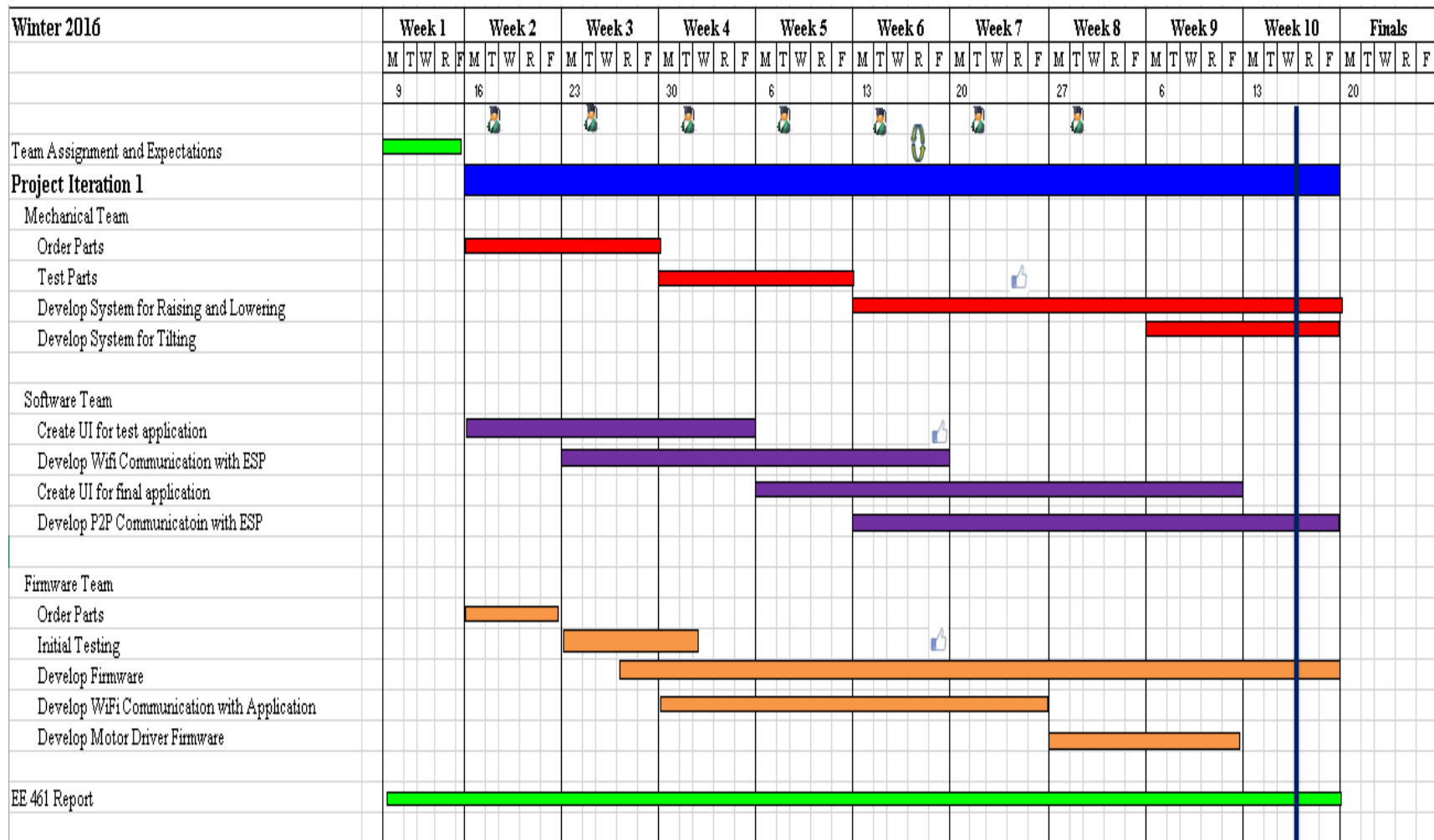


FIGURE V
Wireless Window Blinds Gantt Chart (Winter)

Development

Software Team:

At the beginning of the quarter, software development for the Android application began with a combination of features for a sleek, simple and effective app for users. The basic concept was to provide users with an easy-to-use controller for their motorized mini blinds units. The application would allow users to automatically connect a new mini-blind module to their home network. The connected mini-blinds could be accessed individually through the app's remote interface and users would be able to raise, lower and tilt the mini blinds through an intuitive interface. Some additional features included being able to create groupings of blinds for mass adjustment and saving preset modes for rooms (like closing the blinds for movies).

Development began with the initial creation of a “test application” in order to test wireless connectivity and command sending protocols. The initial application was created to lay a foundation in Android Studio, and to research the methods and commands needed to perform WiFi connection and communication with the ESP module. For the first iteration of the application, extensive research and information was taken from the Android Developers^[7] website and several key libraries and packages were used to create the UI and WiFi connection protocol.

Figure VI below shows the initial interface for the test app. The initial layout uses Android Connective Components, which allows the designer to specify the layout without constraints. However, this layout format does not work well for differing screen sizes. The application included two menus. The first is an initial controller menu with simple buttons for raising, lowering and tilt adjustment. In the implementation, these buttons do not have any functionality. The second menu was initialized to connect the app to the ESP and send commands. This “connect” menu was successfully used to test the WiFi protocol for connecting to the ESP and sending commands. The key library for WiFi connection was the Android Socket Library which includes all the methods for creating a new socket connection given an IP and Port number. Initially, we hard-coded the IP and Port number of the ESP after it was connected to the local network, and tested sending commands using the Socket in Android.

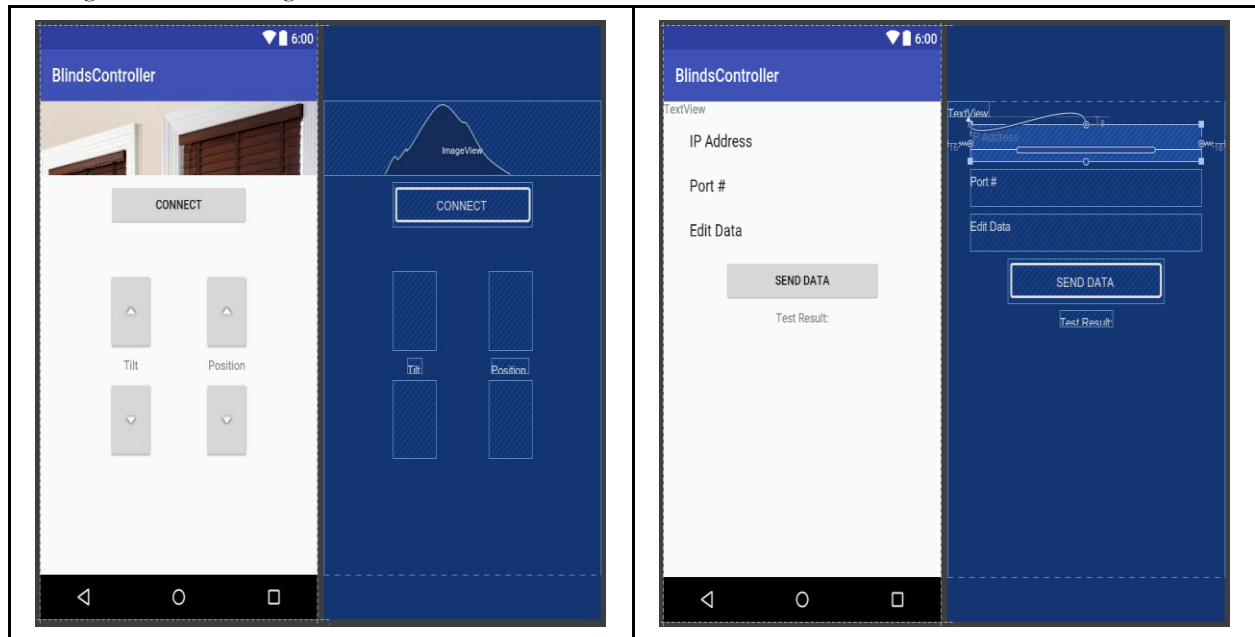


FIGURE VI
User Interface “Test Application”

The current application includes a complete menu system and has room for expanding the application to include a wider feature set. This second iteration was created in order to provide a more polished UI and to research and complete the WiFi Peer-to-Peer protocol with the ESP. Figure VII below shows the UI for the application. Rather than use a connection constraints layout, the new UI was implemented using relative layouts and linear layouts to allow the UI to be more flexible to different screen sizes and resolutions. The layouts also allow for expansion of background images and icons to be implemented in the future. Figure VII shows three different screens. The first is a main menu that can be expanded to include preset functions and grouping screens. The second is a controller with a blinds selection scrollbar and buttons for raising, lowering and tilting the blinds. The third is a connection menu for tethering a new ESP module to the local network.

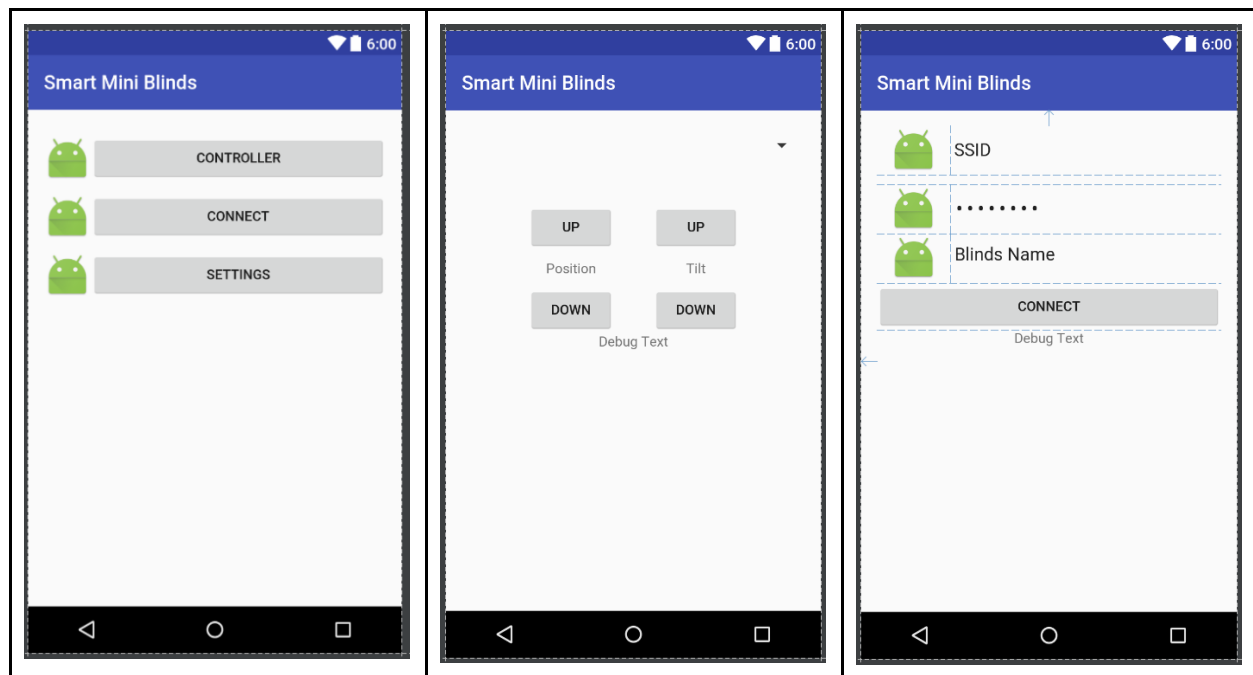


FIGURE VII
User Interface "Final Application"

The bulk of the new development was in the connection menu. The new connection screen uses a SSID, Password and desired blind nickname. The activity connects the ESP module to the specified local network and saves an internal blinds object with the given nickname. The WiFi P2P protocol uses the Android built in Wifi Configuration packages as well as the Android Socket package. Wifi connection uses the ESP as an active hot-spot, and connects to the module with a predetermined SSID and password. After connection, the application sends setup commands following the telnet communication protocol and verifies commands that were received. The setup commands include sending the local network SSID and password, setting the IP and PORT, and sending a reset command. The IP, PORT and user provided nickname are compressed using Gson, into a JSON file and saved to Android SharedPreferences file. Overall, we created a finalized UI with an expandable menu system. We moved from using a global class to use SharedPreferences in order to save each new blind unit's information (Nickname, IP, Port, and position data). We learned about WiFi and Telnet protocol as well as how to implement the Android built in Socket method to initialize a bufferedstream reader/writer with the ESP module. Finally, we are using the WiFi

configuration package to initialize the ESP module with the local network. Looking forward, we want to polish the UI and implement more intuitive widgets for user control. Additionally, we plan to port our Android development over to iOS.

Firmware Team:

The Firmware Team strove to develop a communication system to tether the smartphone application to the motor system. To begin, we chose a WiFi module and a microcontroller that best suited our specifications. With only two quarters to develop, we chose the Arduino Uno as our microcontroller. It offers a friendly developing environment^[9] that would not require extensive development time and has many resources and libraries available for use. Our WiFi module was the ESP8266 due to its low power and inexpensive nature. Our first development platform for initial testing and programming is shown below in Figure VIII.

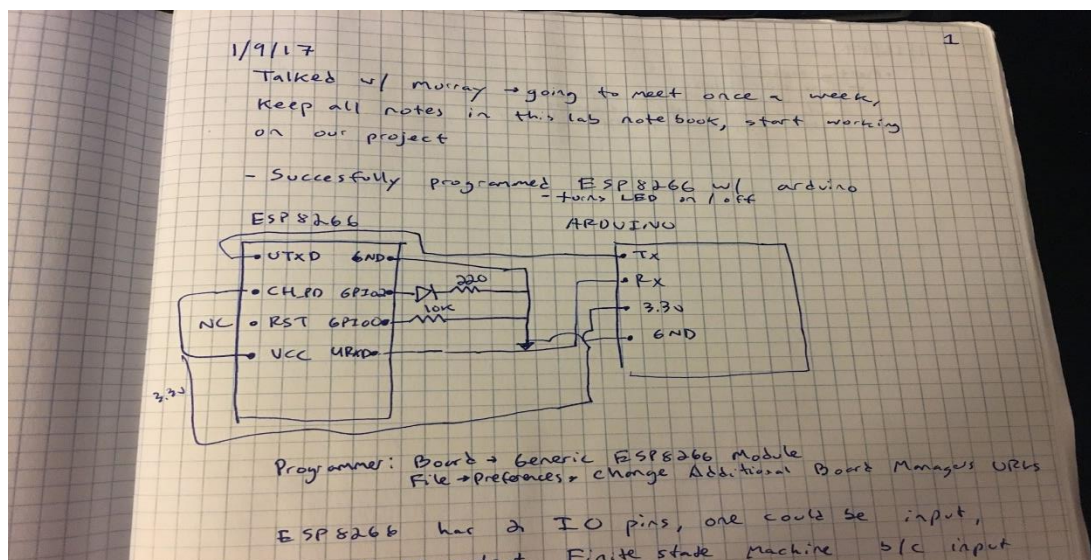


FIGURE VIII
Initial Microcontroller to WiFi Module Circuit

Initial testing goal included getting familiar with the ESP8266 via discovering its capabilities, limitations, and power requirements. The WiFi module is sold with pre-programmed firmware that takes in commands serially. It can act as either a server to be connected to or as a client to connect to other devices. Some of the tests performed included connecting to WiFi and receiving confirmation that it was online, telnetting to a device and receiving information, and gathering information online from the internet.

After more research, we learned that the ESP8266 can act as both a WiFi module and a microcontroller, eliminating the need for the Arduino Uno in our design. This major development reduced the amount of current needed to power our circuit and reduced the overall cost of our project significantly. While this decision increased complexity of the processes running on the ESP, it reduced the need for a communication protocol between the Arduino and the ESP.

One of the most challenging parts of developing with the ESP8266 was programming to it. Instead of just flashing new, factory made firmware onto it we needed it to communicate with the app, decipher a set of instructions, and deliver commands to the motor controller. With the help of online resources like Forward Computing and Control^[8], we were able to find a methodology, circuit, and platform that allowed us to input our own code into the device. The final circuit that we used for development and testing, shown

below in Figure IX, included three output pins that went directly to the motor controller. The schematic for a finished product using this setup would look like Figure X.

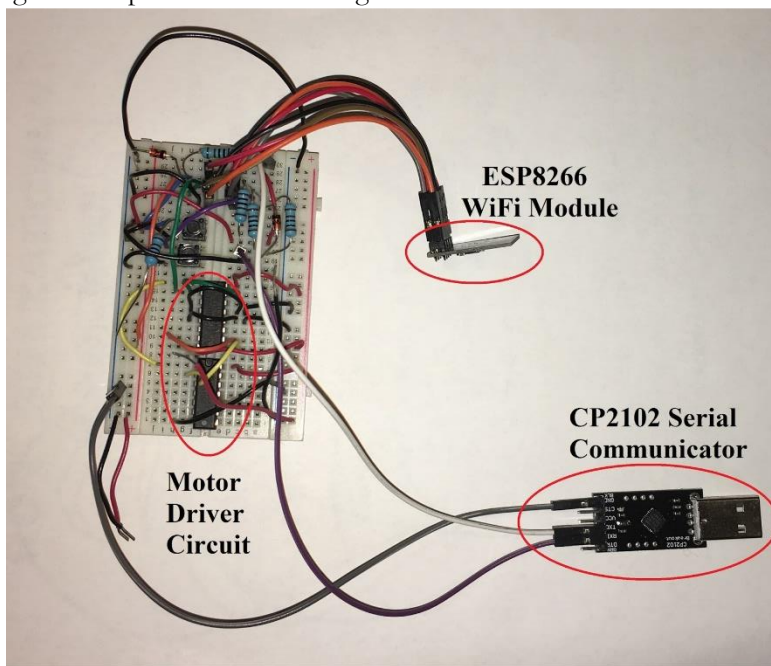
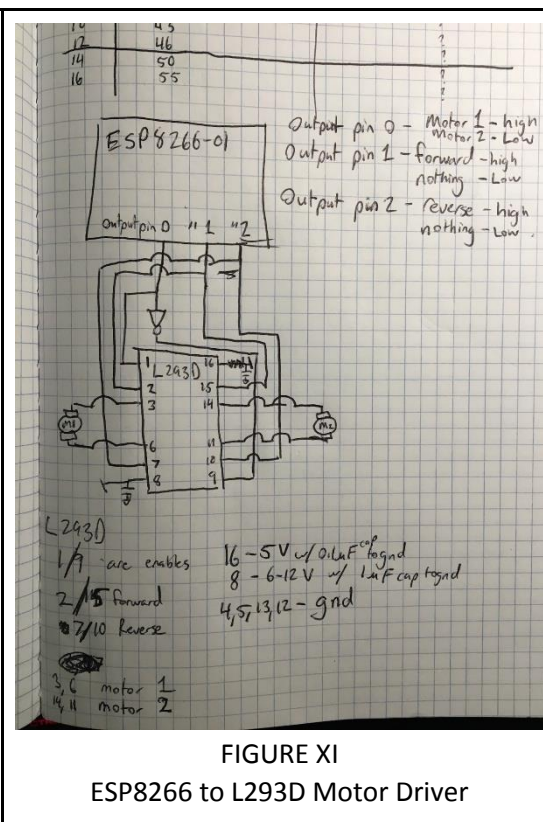
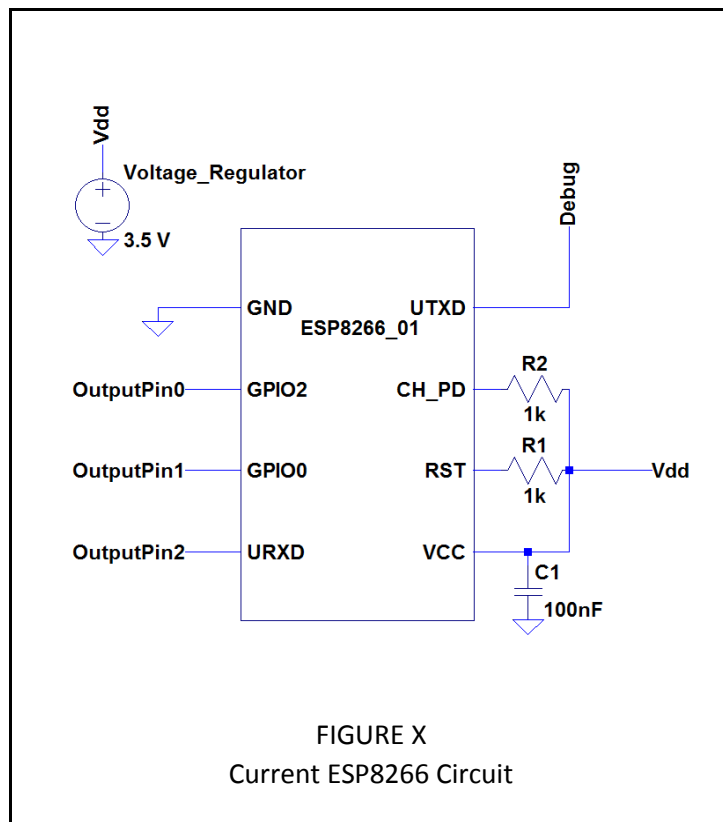


FIGURE IX
ESP8266 Development and Testing Circuit



The outputs of the ESP8266 feeds directly to a motor driver that can move two motors forward and backwards based on the command input. The circuit for the current setup is shown above in Figure XI. If more output pins are required in future development we may use a Programmable Logic Array to communicate with our three output pins from the ESP8266. Additionally, future development includes adding a single battery source to power both the motors and the ESP. This will require a battery that can supply around 9 volts DC and above 400 mA of current for a short duration only a couple of times a day. Currently more development time is being invested into a low power mode that can be turned on and off when the smartphone app is open and closed. Current draw for the ESP8266, when in full power, ranges between 60 mA to 150 mA. Our goal is to reduce this to 15 mA when the ESP is not communicating with the smartphone app by utilizing the ESP's built in sleep mode. Additionally, communication with the smartphone app is being fine tuned to improve communication and reduce errors. Our goal regarding future development for wireless communication is to implement a security measure that secures communication for our IoT device.

Mechanical Team:

The mechanical team goals were to select and characterize motors and build a working prototype to raise, lower and tilt the blinds. Starting week one, motors were researched and selected. The ROB-12281 Micro Gearmotor has a small package size, 130 RPM, a 6-12 V rating and 210:1 gear ratio. Once received, the motor specifications were tested by hand and current measurements were taken as shown in Table XI below.

TABLE XI
ROB-12281 Micro Gearmotor Current Measurements

Voltage (V)	1	2	3	4	5	6	8	10	12	14	16
No Load Current (mA)	29	34	34	34	36	39	41	43	46	50	55
Stalled Load Current (mA)	54	108	162	216	270	324	400	475	563	802	1439

The first iteration of the mechanical system was to use a pulley system. By moving the cords through a roller system, the rollers would raise and lower the blinds. Initially, we thought there would be enough friction between the roller and the cords to allow movement. We believed the 210:1 gearing would be sufficient so no additional gearing was applied to the system. The preliminary designs are shown in Figure XII below. After completion and testing, it was obvious that the current design did not have enough traction or torque to move the blinds.

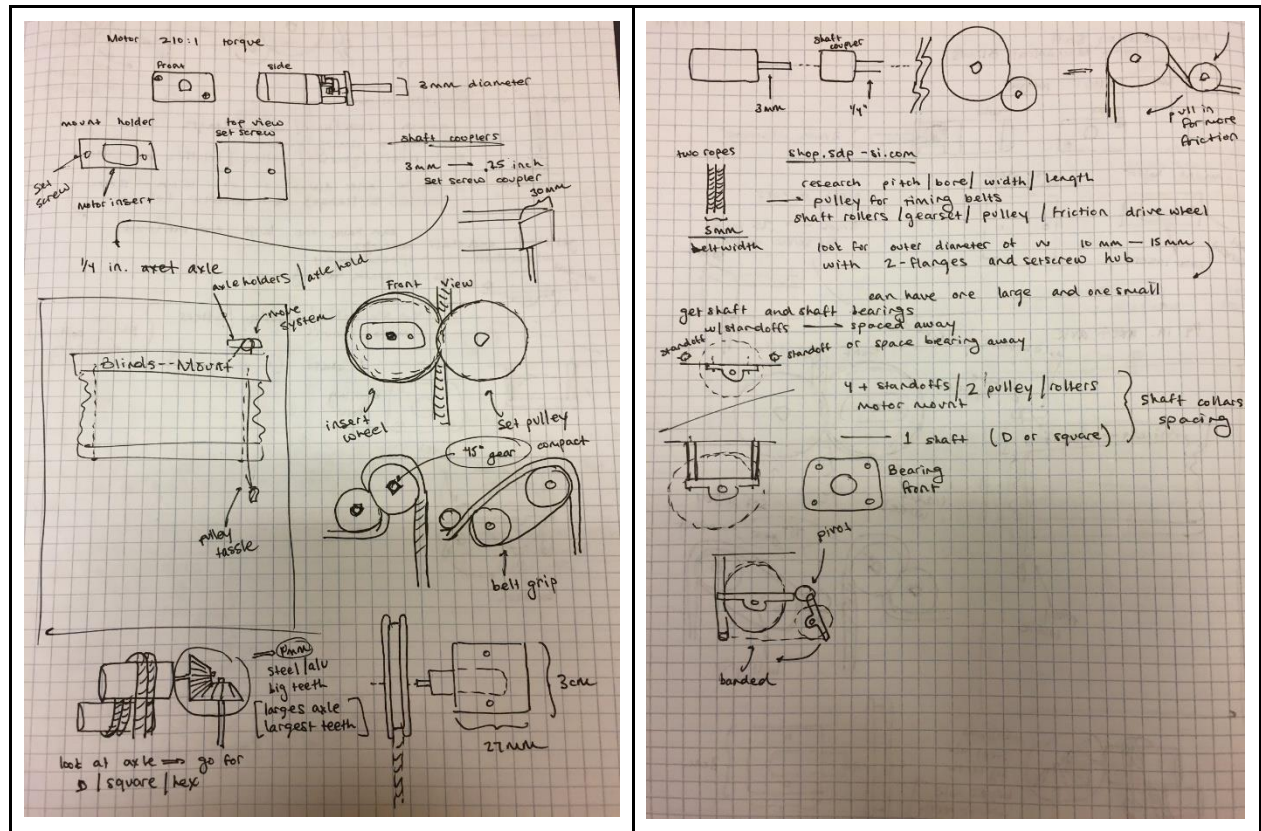


FIGURE XII
System Implementation V1

The goal of the second iteration of the mechanical system was to create a mechanism that would raise and lower the blinds by gathering the cords on a spindle. VEX robotics parts were used to implement the new design because they are flexible and allow for easy rebuilding for future iterations. Figure XIII below shows the new implementation. The new system uses an additional 5:1 gearing to provide more torque. Additionally, a 45-degree gear was used to place the motor in a more optimized position. From testing, we found that the second approach worked much better than the first iteration. We also found that the module was more effective when mounted on the side of the blinds unit, to “pull down” on the cords. The new system worked within the desired voltage range and worked best at 10V. While the system worked better than the first iteration, the spindle (cord gathering mechanism) entangled easily. Furthermore, the current mounting position interfered with the blinds.

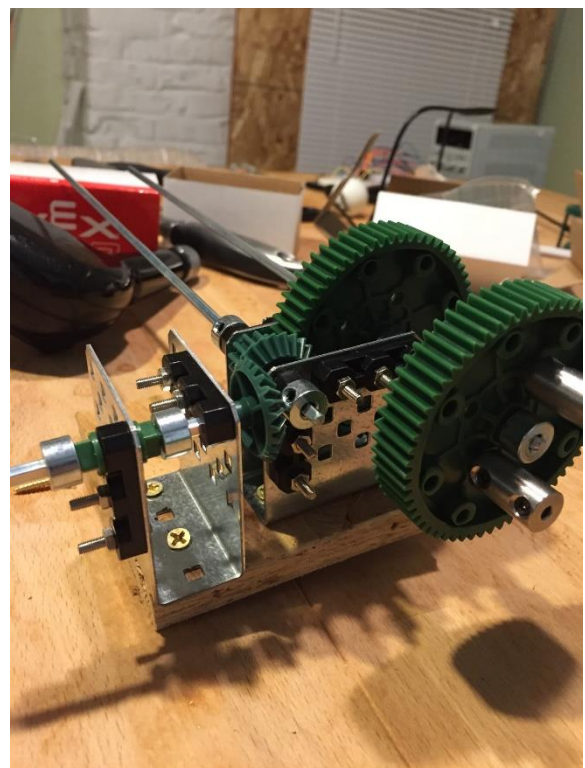
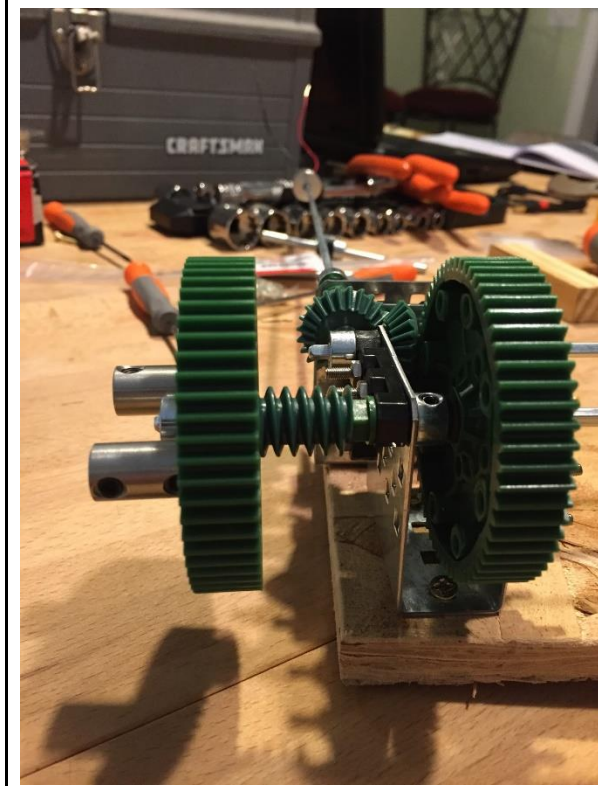
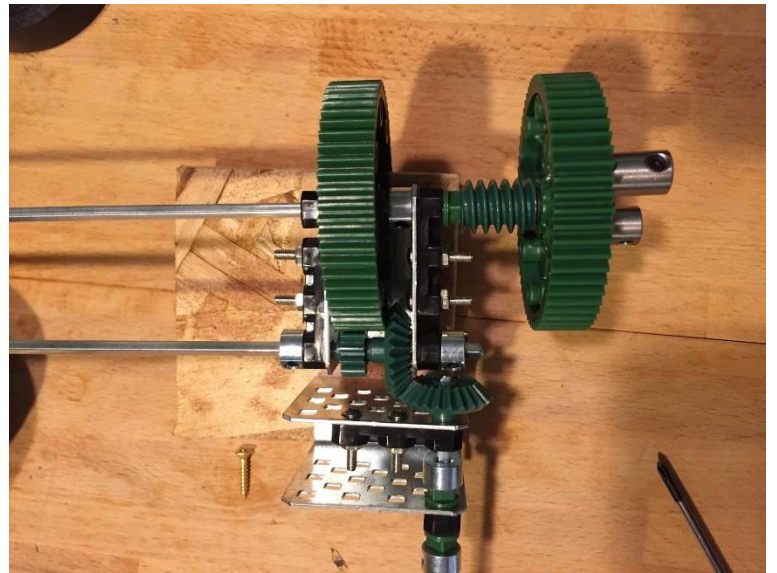
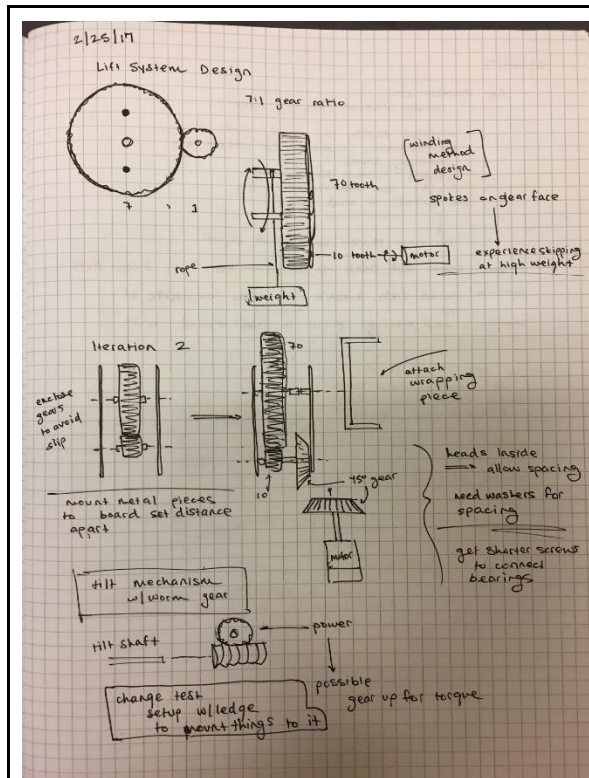


FIGURE XIII
System Implementation V2

Looking forward, for future implementations we plan to implement a similar design using a worm-gear. We believe the worm-gear will provide significant back-torque and will strike a balance between speed and torque. We plan to implement and test a tilt control system using a second motor. Additionally, we want to design a new spindle for winding the cords that does not tangle and a new mounting platform to improve performance. Some additional goals are to compress the entire system into a smaller package, and to mount the system to a pre-existing blinds unit in a sleek package.

APPENDIX C – EE 462 ADDENDUM

Summary Statement

Across the scope of EE 462 the Wireless Window Blinds developed into a cohesive, working system. The three development teams of Mechanical, Software, and Firmware worked together to create a final prototype that successfully raises, lowers, and tilts a set of window blinds remotely.

The Mechanical Team improved upon the previously built raising and lowering system by making it more compact, power efficient, and mountable. A spindle for winding the strings within the blinds was 3D printed which allowed us to minimize the risk of the strings tangling during operation. Additionally, another motor was inserted into the mechanical system that tilts the blinds upon command of the controller. The most challenging task the mechanical team overcame was finding an easy mounting system that could securely fasten the system with the spindle remaining in the correct location. Some future goals to optimize the system include making the system smaller and less noticeable for the customer.

The Software Team furthered the smartphone application for the user to wirelessly control the window blinds. Additional user interface features were added to make the application more appealing to the customer and easier to use. One future objective, if this project was to be continued, includes porting the application from Android to iOS.

The Firmware Team finalized a circuit for receiving commands from the smartphone and delivering the proper response to the motor driver. As the Software Team furthered its app, much time was spent debugging issues within communication between the ESP and the smartphone. Moreover, we added support to properly handle when multiple commands are sent from the smartphone. We created a calibration methodology but were unable to implement it into the system within our time frame. Future goals would be to implement a more secure communication protocol.

Gantt Chart

The Gantt Chart shown below describes the process our team took in order to achieve our final progress. Through the course of EE 462, we completed both an alpha and beta prototype. The mechanical team fully developed a system for both raising, lower, and tilting. After the completion of both of these systems, the team amalgamated the two and then was able to improve the enclosure of the entire system. The software team completed development of peer-to-peer connection with the ESP module and finished prototyping the android application. The firmware team finished developing the communication system with the app over Wi-Fi and constructed a protoboard of our final circuit. Major milestones were achieved Week 3 when the firmware and software teams were able to communicate through the app over Wi-Fi, Week 6 when our alpha prototype was complete, and Week 8 when our beta prototype was finalized.

Development

Software Team:

At the beginning of EE 462, the Software development took the progress from EE 461 and built on the already existing knowledge to create a finalized remote controller app. Some key aspects from the EE 461 that were used throughout EE 462 included: the wireless communication protocol for exchanging data with the ESP, the Telnet commands used to raise, lower, and tilt the blinds, and the basic menu structure. After using and testing the EE 461 application (version 1), we didn't like the system flow and removed the "home" menu activity from the application. As a result, the application was reworked so that the "home" menu displayed the controller buttons, with buttons to add new blinds, and adjust settings. Overall, the new application is cleaner and easier to use and understand.

As shown in Figure XV below, the user interface of the finalized application includes a "home" controller screen, an add blinds menu screen, and a settings menu. First, the home screen contains all the controller buttons and the menu buttons. When using the application, the user would first press the "select blind button" to select which blind they want to control. After the center icon turns green (signalling that the blind has been paired to), the user can control the raising, lowering and tilting of the blinds using the controller buttons. The raising, lowering and tilting commands are sent using a Telnet protocol to the ESP, which then decodes the command and activates the correct motors.

Within the "add blinds" menu, the user can register a new blinds unit with the application. If the user wants to register a new blind in the app, they first must connect the ESP to the local network. This process has several steps. First, the user turns on the ESP control unit circuit. After some time, the ESP wireless hotspot will appear in the Wi-Fi menu of the user's phone. The user then connects to the Wireless Blinds AP in the Wi-Fi menu. After connection, a pop-up window will appear and the user is prompted to enter their home network's name, password, the desired IP Address and port number. The ESP will reboot and connect itself to the local network. Finally, the user should reconnect to the local network. After the ESP module is connected to the local network, the user can register the new blinds unit with the application, connect to it, and control it. Finally, the settings menu includes the option to delete all registered blinds within the application, and some additional Easter Eggs for users.

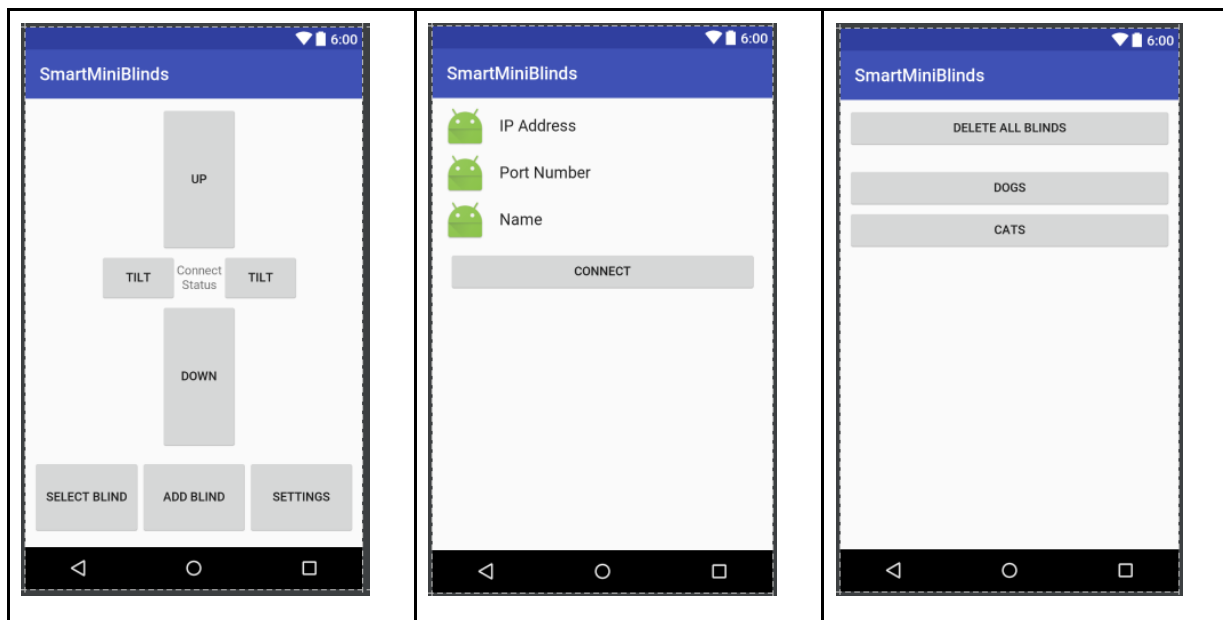


FIGURE XV
User Interface

Overall, we were able to finalize a working application that communicates with the ESP. The application successfully controls the raising, lowering and tilting of the window blinds module system. The bulk of the new development in EE 462 was finalizing an easy-to-use UI and creating a working protocol for communicating with the ESP. Looking forward, some areas to expand and improve include: allowing users to connect the ESP to the local network using the application, adding window blinds grouping and group control protocols, and creating automated blind adjustment menu so that users can set timers or schedules for the raising, lowering and tilting of their home blinds.

Firmware Team:

To begin EE 462, the Firmware Team finalized and tested a circuit to properly receive and deliver commands to the motors while being powered completely from 9 volt batteries. The first main addition from the circuit in 461 included adding two voltage regulators to deliver 3.7 and 5 volts to different components in the circuits. Then, noise reducing capacitors were added throughout the circuit to cut down on unwanted noise. The final circuit is shown below in Figure XVI.

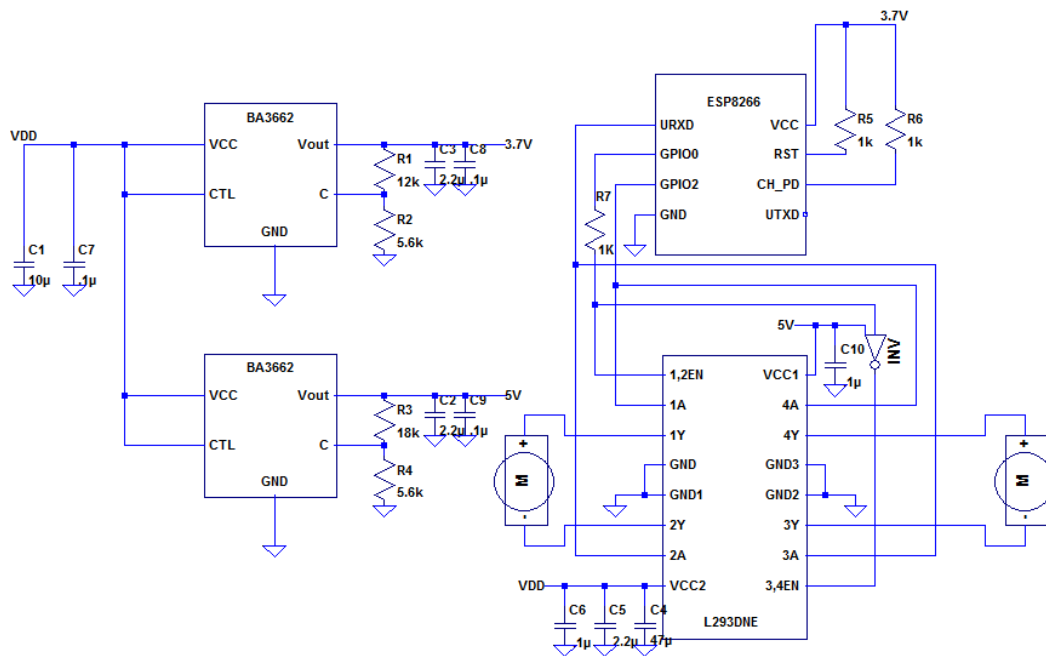


FIGURE XVI
Motor Controller Schematic

We then adapted this circuit onto a protoboard, shown below in Figure XVII. This provided additional stability and compactness when testing our system as well as looks cleaner than a breadboard.

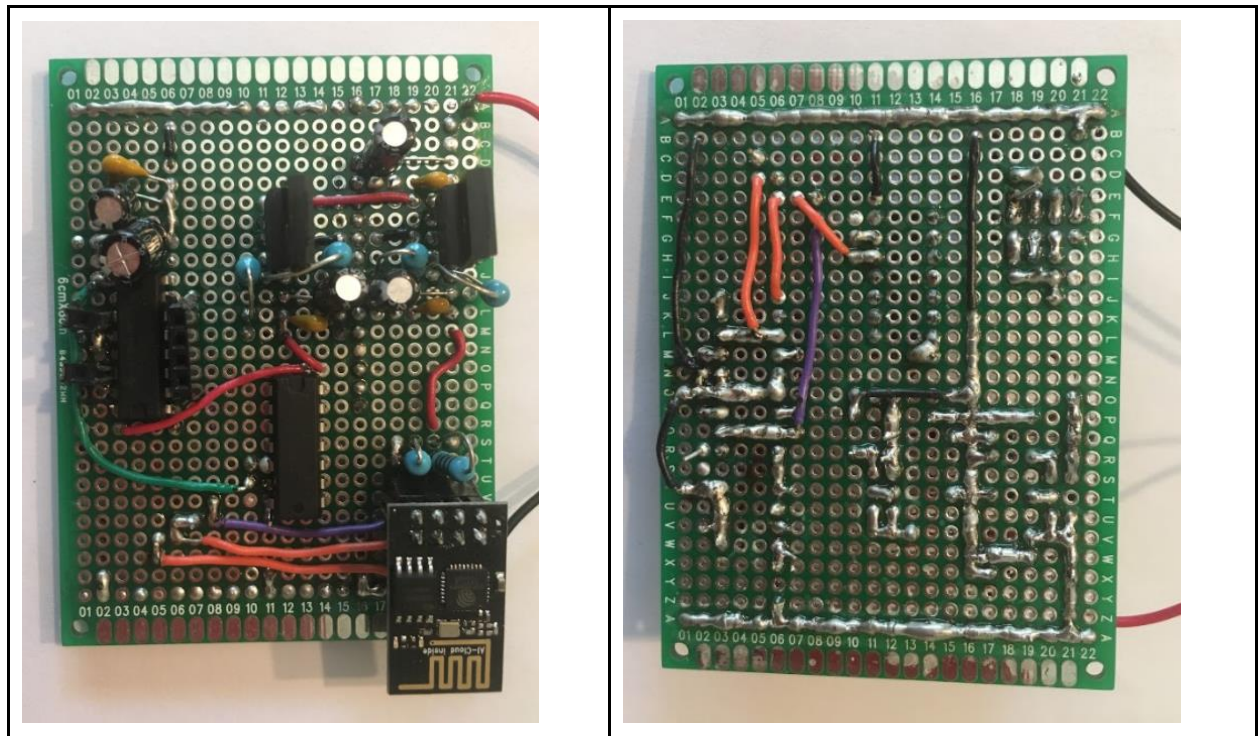


FIGURE XVII
Motor Controller Circuit

The Firmware Team also worked in tandem with the Software Team to provide the best user experience for the customer. Some of the improvements made to the codebase for the ESP8266 include creating a queue to store commands, polling to see if commands have timed out, and saving the Wi-Fi name and password when the board is reset. These changes made our system more robust against losing power and overloading the ESP with commands. Overall, the changes made in EE 462 increased usability for the customer and progressed to a more compact, stable system.

Mechanical Team:

Building off the mechanical system from EE 461, we revised the window blinds module into a final working version. Throughout EE 462, we tested several configurations, gear-ratios and mounting systems. Our main goals were to create a working product while trying to reduce space, increase speed, and make it efficient. During our testing, we tried a variety of spindles mechanisms, and had our spindle 3D printed. The final version has the full capability to tilt, raise and lower the window blinds using a 9V battery.

First, in EE 462 we finalized the tilt mechanism. Originally, we tried to tilt the blinds by attaching our Micro Gearmotor directly to the tilting shaft in the window blinds. When we initially tested the tilting, the motor would not tilt the blinds. After more investigation, we found that this was due to the worm gear/rod system that allows normal users to tilt the blinds. The back torque of the worm gear was inhibiting the motor from tilting the shaft. After removing the worm gear, the motor worked great. Figure XVIII below shows the tilt motor, and the removed worm gear mechanism.

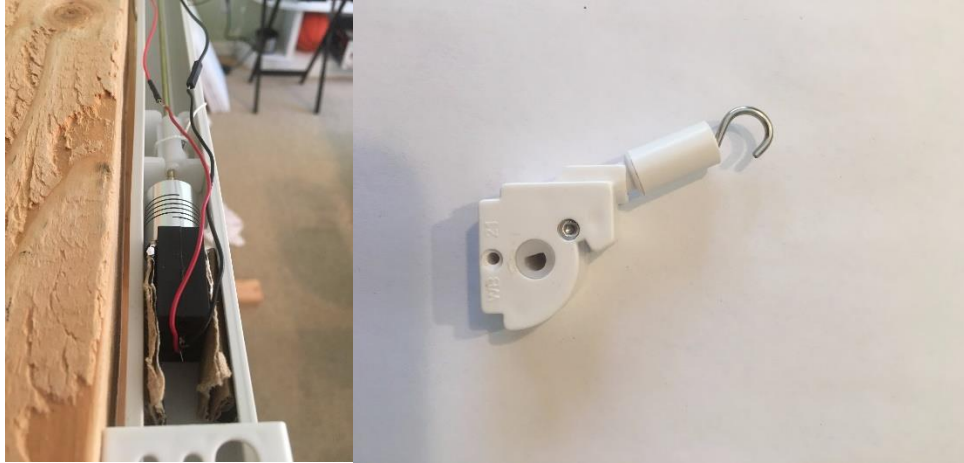


FIGURE XVIII
Tilt Motor, and Removed Worm Gear

Next, we found that raising and lowering the blinds was much more difficult. Like in 461, our module was developed using VEX robotics parts. Version 3 was developed using a worm gear to add a higher torque for raising and lowering the blinds, and to add additional back torque to avoid slippage. After testing we found that the worm gear, while providing enough torque, slowed down the system too much. The worm gear system was beneficial because it was small and compact and provided easy surfaces to mount to the blinds. However, due to the slowness, it was unusable. Next, we tried using the same worm gear system with additional gearing for speed. We attached a 5:1 gearing to increase the speed, in hopes of speeding up the raising and lowering. This version was also too slow. While it provided sufficient torque, the system was too slow and took a full 10 minutes to raise or lower the blinds. This final version 3 module is shown in Figure XX below.

Additionally, a new 3D spindle was constructed for the system with a through hole for attaching the blinds string. The spindle was printed with the help of our advisor Richard Murray and the specifications and design are shown in Figure XIX below.

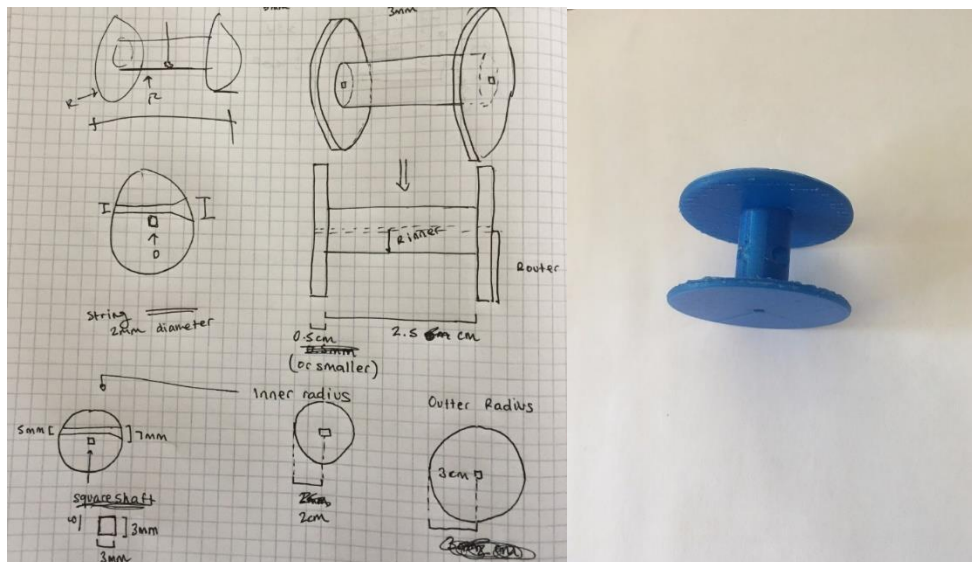


FIGURE XIX
3D Printed Spindle Design and Specifications

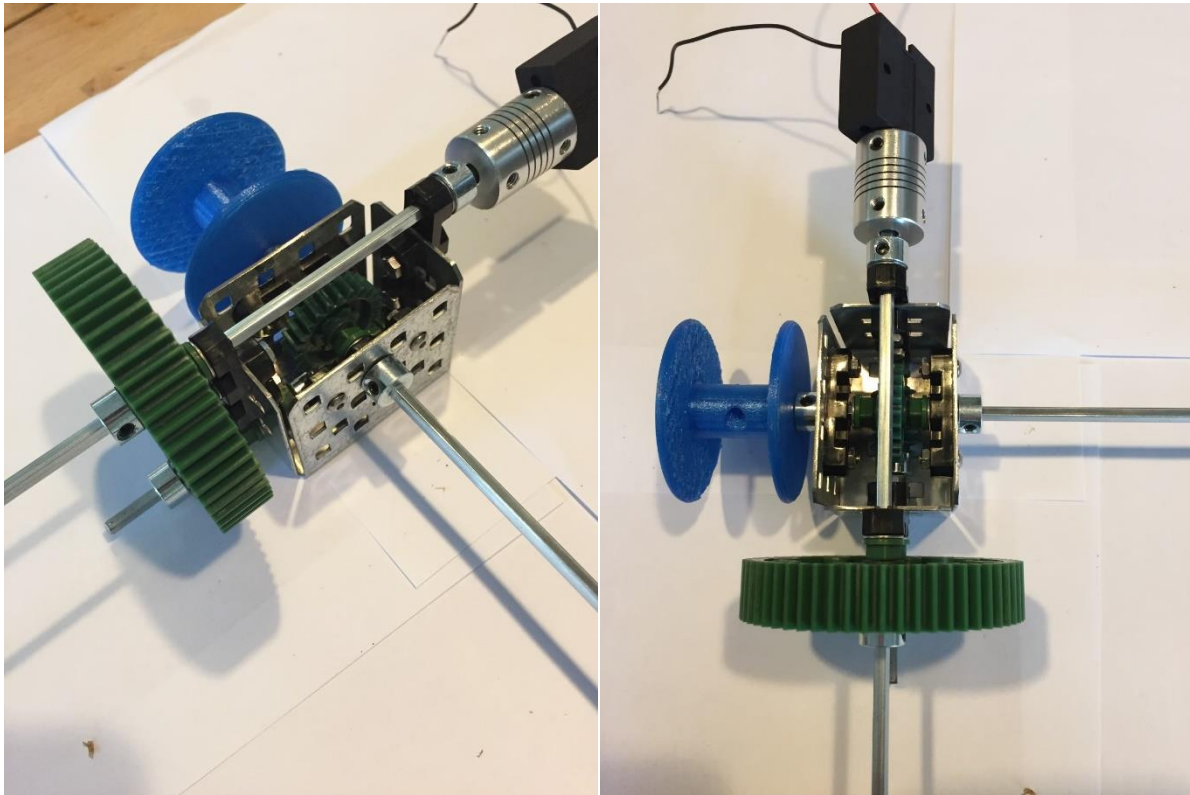


FIGURE XX
Motor Setup Version 3

Next we tested and built version 4 using VEX robotics parts, which is our final version. The working module uses an additional 9:1 gearing for torque to provide more force for raising and lowering. The final result is slow, but usable. In addition to the new gearing, the final system uses a 3D printed spindle which helps gathering the strings more uniform and causes less entanglement. The module was easily mounted to the window blinds and successfully holds the motor in place. Additionally, the system is small, compact and takes 2.5 minutes to raise and 2 minutes to lower. Figure XXI below shows the final version mounted to the mini blinds. Figure XXII below shows the final completed blinds system.

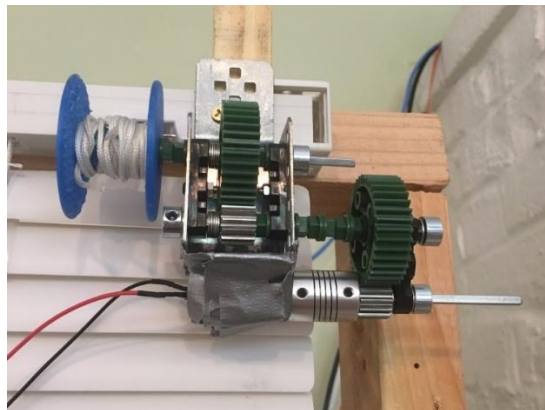


FIGURE XXI
Version 4 (Final Version)

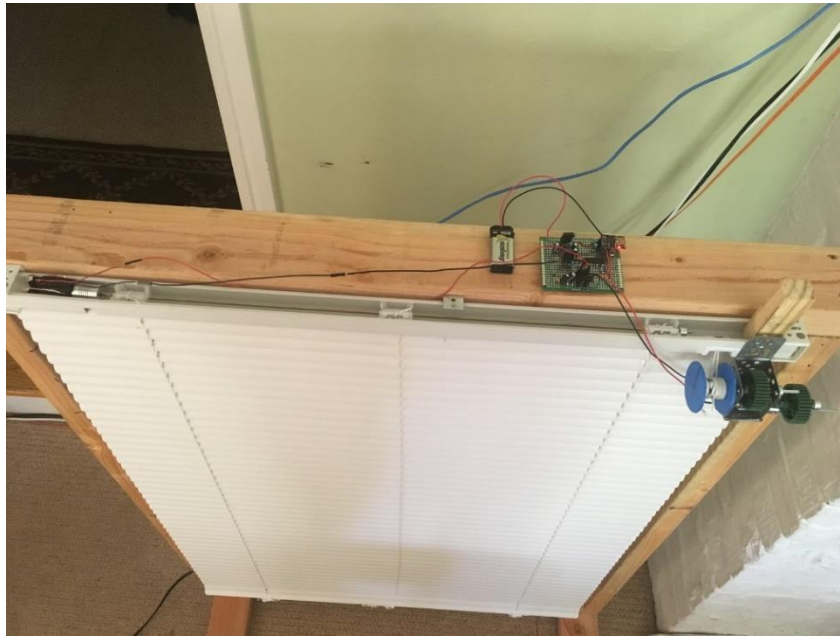


FIGURE XXII
Final Complete System

Future Project Improvements:

Looking forward this project could be expanded and improved on in a variety of ways. Taking our base module and application there is a lot of room for growth for future projects. First, the Android application could be converted to iOS. If this product were to be sold in stores, having the application in iOS would satisfy a larger user base and make the overall product more appealing. Within the application, one could expand the current app to allow groupings of blinds. This feature would allow users to control multiple blinds at once and manage groupings of blinds by room or location. Another feature might be to add presets, like raising or lowering the blinds to a specific preset position or tilt. Finally, the application could support more tutorials, settings, and might want a feature that allows users to set timers to have their blinds raised or lowered on specific days at specific times.


This project could also be taken on by a mechanical team. Some major improvements for the mechanical system might include: creating a compact 3D printed gear system that takes less space than the current module. Additionally, experimenting with different motors to make the raising and lowering of the system would be a huge improvement for users. Some other future ideas might be to make the module attachable, and self-contained, so that users can easily attach the entire motorized module to any existing blinds unit.

Lastly, some goals for the firmware and electrical hardware include implementing a calibration protocol so that users can command the blinds to move to a specified position. The firmware could also be developed onto a PCB, to make the design smaller and more compact. Another goal includes researching and implementing sleep modes for the ESP to conserve power. Some other future goals include a protocol security increase for remote accessing and control.

APPENDIX D – Senior Project Expo Pictures


Purpose/Background

Wireless Window Blinds is an IoT inspired project. Our goal was to build a product that would fit into the Smart Home industry. This project allows users to wirelessly control a motorized module that attaches to pre-existing window blinds. Using a smartphone app, user's can wirelessly raise, lower and adjust the tilt of their window blinds.



Market Research

We focused on homeowners as our primary customer. The Smart Home industry is rapidly growing and most homeowners are looking to upgrade their homes to include smart devices. Our goal was to build an affordable product that allowed homeowners to wirelessly control their window blinds using a smartphone as a controller.



Specifications & Goals

Marketing Requirements	Engineering Specifications	Justification
1/5	Controller responds to app input controls in under 1 second	The customer needs a set of blinds that operate in a quick manner.
2	Runs on 9 volts and draws less than 100mA of current	The consumer needs common, easily accessible, cheap batteries found in any store.
2	Complete window blinds system runs continuously for 1 hour.	The consumers need a durable, power efficient system that requires little to no maintenance or upkeep.
3	Product must follow the IEEE NESC safety standard.	Devices released to the public must practice safe installation, operation, and maintenance.
4	Production cost should not exceed \$75.	Motorized blinds cost between \$250 to \$300. This system must perform on par to remain competitive.
5	Small attachable module that doesn't interfere with existing blinds.	Customer needs dictate a demand for a hidden system to not represent an eyesore.
1/6	Communicates with smartphone app remotely using Peer to Peer Wi-Fi Data transfer.	Customers with in-home Wi-Fi need an easy-to-use, convenient interface to control their window blinds system.

Marketing Requirements

1. Responds to remote raise, lower, and tilt commands quickly
2. Battery powered
3. Easy and safe to install
4. Inexpensive
5. Looks sleek/hidden.
6. Allows user to control window blinds remotely


Wireless Window Blinds

Zachary Malig, Cooper Laone, Jake Phillips, Niraj Morar
Advisor: Richard Murray

Design Overview

Software


- Android Studio
- Intuitive UI
- Telnet Wi-Fi Communication
- Multiple iterations



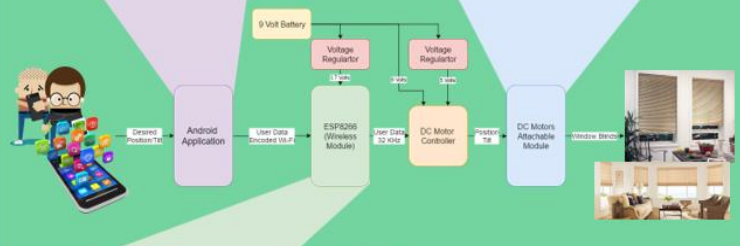
The software team developed an Android Application. The application acts as the user interface for controlling the raising, lowering and tilting of the window blinds module. The mechanical user cannot build or assemble to perform. Telnet communication with the ESP8266 module. The application has gone through several iterations of testing. Throughout the year, the application has gone through several iterations of testing. The final application allows users to connect a new blind to their system, control the raising, lowering and tilting of the blind, and manage the saved window blinds.

Mechanical

- Raises and Lowers Blinds
- Tilt Blinds
- 9 Volt Battery
- Attachable Module
- Multiple Iterations

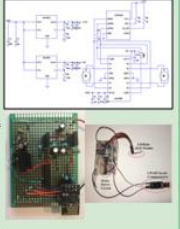


The mechanical team developed a mechanical system to raise, lower and tilt the window blinds system. The goal of the mechanical system was to be an accessory that could be attached to a pre-existing window blind. The system uses an additional gear train to get the appropriate torque for raising and lowering the window blinds. Throughout the year, the system was through 4 major iterations. Several configurations and gear ratios were tested. In the final iteration, the system used 1:1 gearing and a 9:1 torque gearing, and a 3D printed spindle to raise and lower the blinds.



Firmware and Hardware

- ESP8266 Wi-Fi Module
- Motor Driver Control
- Voltage Regulators
- Telnet Wi-Fi Communication



The Firmware Team developed a system to connect the smartphone application to the motor system. The ESP8266 was the best solution due to its low power and inexpensive nature. The ESP8266 speaks directly to a motor driver that controls the tilt and raising/lowering systems. The system is powered by 9 volt batteries that directly supply the motor and is regulated to 5 and 3.3 volts for the device chips and the ESP8266 respectively. The final application uses a set of defined commands to remotely control, display, and measure data from the Android Application to the motor.

Results

- Remotely Controllable by Smart Phone
- Responds to app input in under 1 second
- Powered off 9 volts
- Max current requirement of 300 mA
- System cost of \$74.90
- Medium sized mechanical module that requires some changes to existing blinds
- 150 seconds to fully raise blinds
- 100 seconds to fully lower blinds
- Communicates using Telnet Wi-Fi protocol


Conclusion

During the course of senior project the Wireless Window Blinds evolved from a set of specifications, requirements, and block diagrams into a functioning project that moves window blinds remotely. Through the quarters the project was divided into three different development teams: Mechanical, Software, and Firmware. Each team faced different obstacles and challenges throughout the year, but in the end the three teams successfully combined their development to produce a final prototype. Overall, this project taught us professionalism, collaboration, communication, and accountability.

Unit Cost

Total: \$74.90

Expenditures



Future Work

This project could be expanded in several ways. First, the mechanical system could be 3D printed as an attachable module that is faster, and more space efficient. The system circuit could be made into a PCB to reduce space. The firmware could be expanded to include deep-sleep modes for power saving and encryption for additional cyber security. Finally, the software could be ported to iOS and could include more features. We hope this project does not end here, but rather other students in future senior projects build off of what we have developed.

Photos from expo here