ENSC 305W/440W Grading Rubric for Design Specification

Criteria	Details	Marks
Introduction/Background	Introduces basic purpose of the project.	**/05%**
Content	Document explains the design specifications with proper justification for the design approach chosen. Includes descriptions of the physics (or chemistry, biology, geology, meteorology, etc.) underlying the choices.	**/20%**
Technical Correctness	Ideas presented represent valid design specifications that will be met. Specifications are presented using tables, graphs, and figures where possible (rather than over-reliance upon text). Equations and graphs are used to back up/illustrate the science.	**/20%**
Process Details	Specification distinguishes between design details for present project version and later stages of project (i.e., proof-of-concept, prototype, and production versions). Numbering of design specs matches up with numbering for functional specs.	**/15%**
Test Plan	Provides a functional test plan for the present project version. (Note that project success will be measured against this test plan.)	**/10%**
Conclusion/References	Summarizes functionality. Includes references for information from other sources.	**/05%**
Presentation/Organization	Document looks like a professional specification. Ideas follow in a logical manner.	**/05%**
Format Issues	Includes letter of transmittal, title page, abstract, table of contents, list of figures and tables, glossary, and references. Pages are numbered, figures and tables are introduced, headings are numbered, etc. References and citations are properly formatted.	**/10%**
Correctness/Style	Correct spelling, grammar, and punctuation. Style is clear concise, and coherent. Uses passive voice judiciously.	**/10%**
Comments		

Simon Fraser University
8888 University Dr.
Burnaby, BC Canada

September 23, 2013

Dr. Lakhsman One
School of Engineering Science
Simon Fraser University
Burnaby, BC, V5A 1S6

Re: ENSC 305/440 Design Specification for Smart Blinds System

Dear Dr. One,

Enclosed is a design specification from BikoTech describing the features and attributes of the Smart Blinds System. We are designing an improvement to conventional home and business window blinds by integrating a unique control system that offers users enhanced functionality and automation capabilities.

Our design specification outlines how we will achieve each prototype deliverable in our Functional Specification document for our Smart Blinds System. This document will serve as a detailed guide to our team members that they can refer to when implementing each system module. This document will also be used as a design and verification guideline to ensure all requirements are met in the final prototype model.

The BikoTech team consists of Jorden Bryer, Chaman Toor, Willy Wong, and Clark Zhao. As engineering students we believe we have the drive and the skills to make BikoTech a competitive and engaging company. If you have any questions, please contact me by email at jordenb@sfu.ca.

Sincerely,

Jorden Bryer

Jorden Bryer
President and CEO
BikoTech Automated Systems

Enclosure: ***Design Specification for Smart Blinds System***

Design Specification for

Smart Blinds System

Project Team
Jorden Bryer – Chief Executive Officer
Willy Wong – Chief Financial Officer
Chaman Toor – Chief Technology Officer
Clark Zhao – Chief Operating Officer

Contact Person
Jorden Bryer
jordenb@sfu.ca

Submitted To
Dr. Lakhsman One
School of Engineering Science
Simon Fraser University

BikoTech Automated Systems

Executive Summary

The design specification for our Smart Blinds System describes in detail the design and development of our prototype model. In this document we aim to outline all the necessary facts pertaining to the implementation of our [Rn-I] and [Rn-II] level functions as outlined in our functional specification.

In this document we describe a fully functioning automated blinds solution that is designed to blend in with conventional window blind systems. Its parts and control systems will be integrated into the structure of the blinds themselves and will utilize the housing for the blind's gears for connections to our motors. These motors will integrate with an Arduino microcontroller that acts as the brain of our control system, receiving ambient light data as well as user inputs to achieve a fully functioning system.

The prototype of our product aims to achieve the following key features:

- Motorized blind lifting and tilting mechanisms to control light flowing into a room
- Ambient light sensing, allowing our microcontroller to obtain real time data on light conditions
- Power independence through battery power and a compact solar charging array positioned on the upper part of the blinds behind their housing and hidden from inside view
- Interactive software interface that allows users to program when or under what light conditions they want their blind system to open or close
- Aesthetically pleasing design with cords, motors, and other electrical components hidden from view

The BikoTech team is motivated to integrate these features successfully into our Smart Blinds System. We also realise that creating a safe device with robust software is a top priority. With this in mind we will follow the applicable ISO (International Standards Organization), CSA (Canadian Standards Association), and WCMA (Window Covering Manufacturers Association) standards to help make these requirements a reality.

Table of Contents

List of Figures and Tables

Glossary

ISO	(International Standards Organization)
CSA	(Canadian Standards Association)
WCMA	(Window Covering Manufacturers Association)
Actuators	A electronic device that produces motion, typically a motor
Arduino Mega	A microcontroller for controlling and communicating with various open-source hardware, it can be programmed with the standard coding language

1. Introduction

The Smart Blinds System is designed to be an automated blind control system. It will allow users to specify their own expectations for natural light entering their room, and to program specific times to open and close the blinds. It serves as an alternative to traditional blinds systems that are used daily but offer no other useful features. The detailed design requirements for the Smart Blinds System are contained in in the following sections.

1.1 Scope

This document outlines the designs of our solutions to achieve the requirements listed in our functional specification at [Rn-I] and [Rn-II] priorities. Our proof of concept model will differ from our production model in many cost and design aspects so the detailed low level specifications of our production model are outside the scope of this document.

1.2 Intended Audience

Our design specification is targeted towards all members of the BikoTech team. Each member will use this document as a guideline when implementing each feature of our Smart Blinds System. This document will also be relied on during all stages in the testing procedure to ensure the outcomes of our prototype adhere to our original specifications. Throughout the implementation of our design our team members will use this document to assess the progress of our design and to eventually gauge the success of our project.

2. System Specification

The BikoTech Smart Blinds will serve as a fully functional blind automation system. All components of the Smart Blinds System are programmable through a simple LCD and pushbutton interface. Through this interface the user can specify times of the day they would like to alter the position of their blinds, or to set specific amounts of light they would like to enter their room at all times. Through the use of a two motor control system, the Smart Blinds System can balance light levels throughout the day, and maintain the preferred lighting conditions for the system's user in all conditions. The alarm functionality also offers a unique ability to open or close your blinds on a set timer, so that waking up to the morning sun or closing them shut after a sunset can all be done automatically and without hassle.

3. Overall System Design

This section will provide a high level overview of the entire design. It will explain all design aspects of the Smart Blinds System, such as mechanical and electrical components in the following subsections. Design details corresponding to specific parts, as well as software design are listed in their respective sections.

3.1 Mechanical Design

The overall mechanical design of the Smart Blinds System can be seen in the figure below, it shows the blind inside its wooden frame and housing. The housing is made transparent for illustration purposes. The control panel can also be seen on the top right corner of the frame, which will be the user's main access to the blind controls.

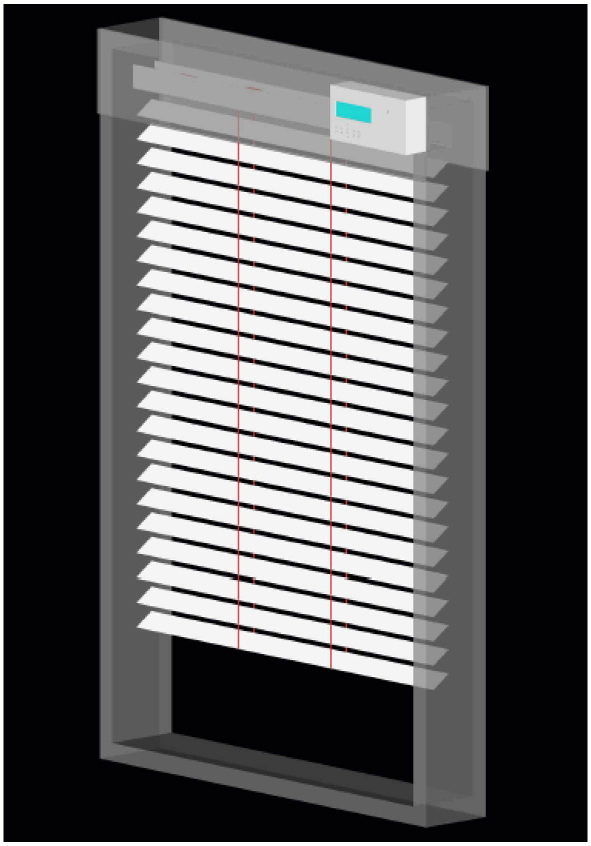

Figure 1: Smart Blinds System in Frame and Housing

The initial decision in the design was to choose the right kind of blinds that will satisfy our functional requirement R23-I, which states that the entire blinds shall not exceed 12kg. This requirement is important, as it will also be the limiting factor regarding the torque and speed of the lift mechanism motor. The following blinds were considered:

- Shade-O-Matic 2in Faux Wood Blinds. Considered for the quality and the larger 2-inch housing. Later rejected due to heavy weight and price [1].
- Design view 3 1/2in Vertical Blinds. Considered for its reasonable price and lightweight design. Eventually rejected due to its complex internal mechanisms [2].
- Perfect Home 1in Aluminum Blinds. Chosen as our winning candidate. It's lightweight and compact, with simple internal components that can be linked with actuators with minimal alterations [3].

The blinds were picked was initially 48 inches long but were shortened to 30 inches for easier transport. After the modification it has the following physical requirements on the frame and actuators.

Tilt Torque	20 oz.in
Lift Torque	40 oz.in
Tilt Angle	180 degrees
Weight	5 pounds
Dimensions	18”x1.5”x30”

Table 1: Blinds Basic Requirements

The torque requirements were estimated from weight tests with actuators with known torque limits. From the requirements listed above, we designed the frame out of solid wood planks that are ¾”x3¼”, and made the frame 18”x30” supported by steel corner braces. This provided a stable platform to mount our blinds and electronics, fulfilling our mechanical requirement R25-I. The housing in which all system hardware must fit will be 3½” tall and attached upper frame to conceal the retracted blinds and the other components. Next we picked the appropriate geared motor and servo that provided the required torque, speed, and range of motion. All of which are documented is their corresponding sections.

3.2 High-level System Design

This section provides a visual overview of the high-level system design. It shows each of the modules interfacing with the brain of our system, an Arduino microcontroller. User inputs include setting the clock, open/close commands, setting open/close time, and setting the interior light thresholds. System outputs include displaying current time, menu item and selections, and currently engaged actuators

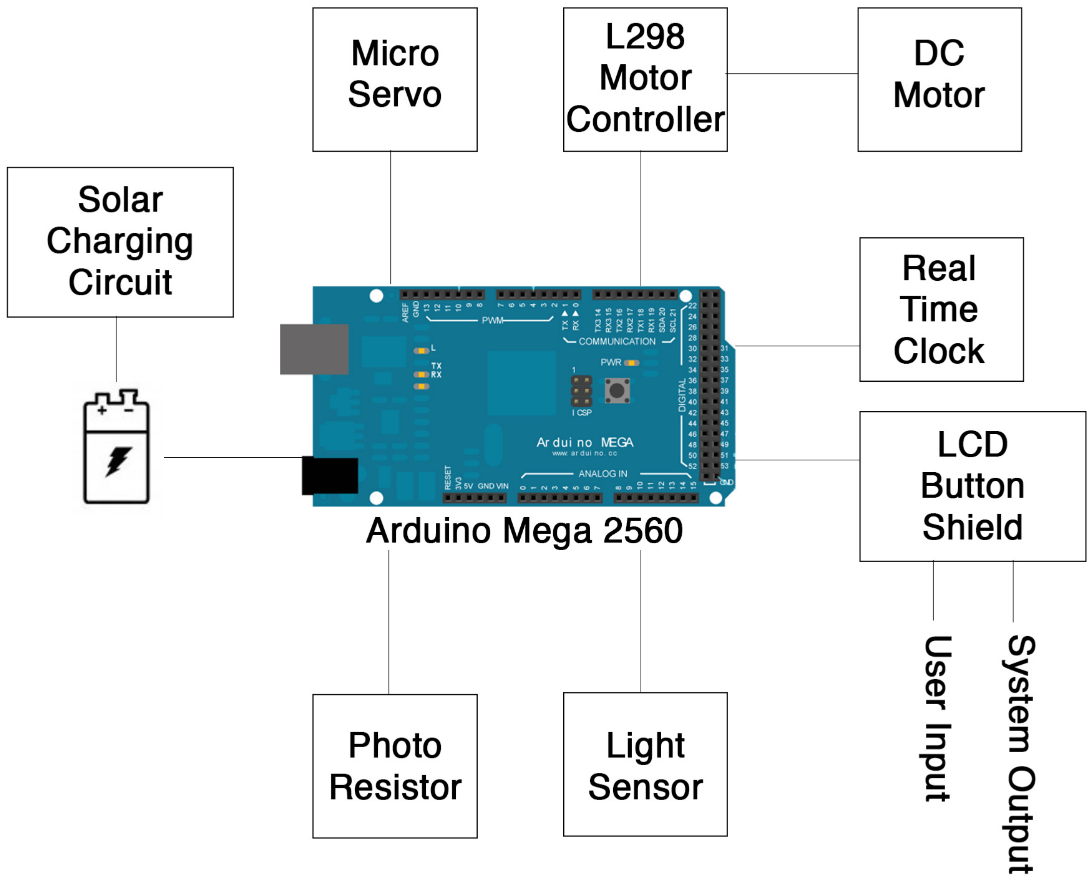

Figure 2: High Level System Diagram

Below is also a flowchart of the high-level operations of our Smart Blinds System. It shows the basic control stages of our system and how it reacts when receiving inputs.

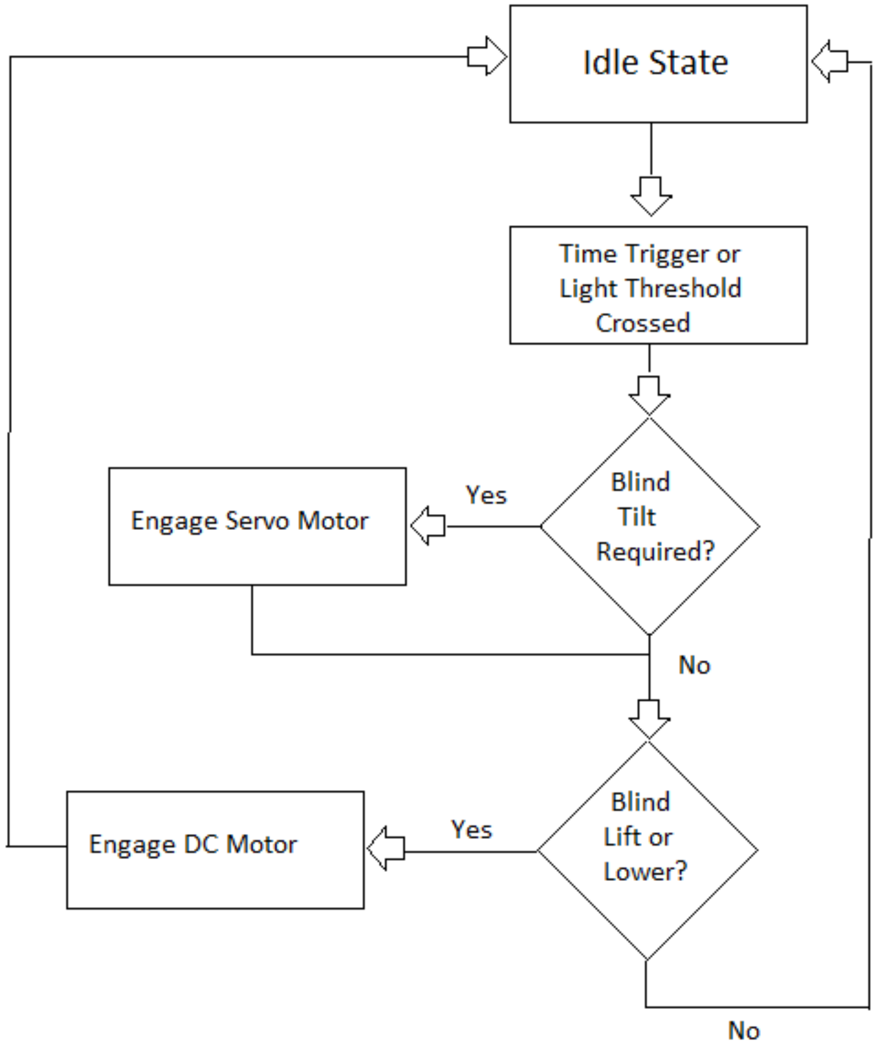

Figure 3: System Flowchart

3.3 Sensors

Two sensors are implemented in our Smart Blinds System. The first is a photo-resistor and the second is an I2C ambient light sensor. The placement of these sensors can be seen in the diagram below.

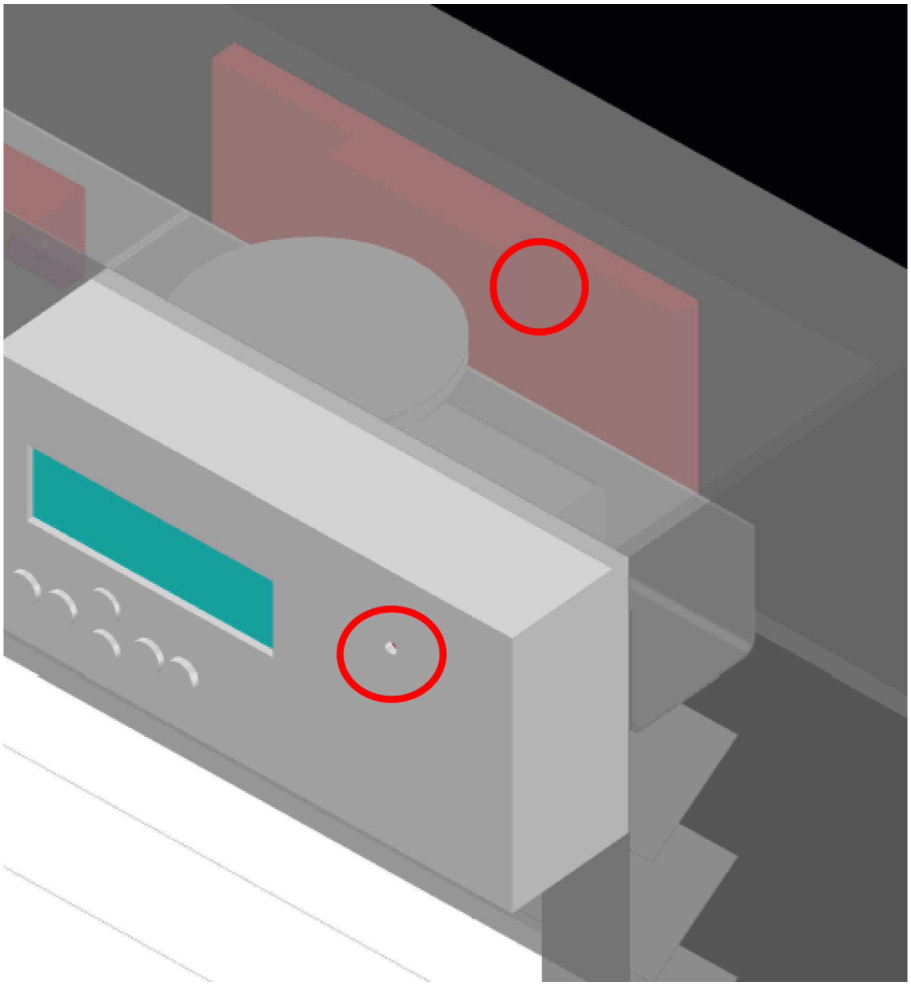

Figure 4: Sensor Placement Shown in Red

We decided to point the photo-resistor towards the exterior to get a rough reading on the lighting condition outside so it can send information to the Arduino on whether it is day or night. The more accurate ambient light sensor is pointed towards the interior within the control panel, to give an accurate lux reading which the user can set to regulate the amount of light entering their room.

3.4 Actuators

Two actuators are implemented in our system; a DC geared motor, and a micro servo. The placement of these actuators within our system can be seen in the following diagram.

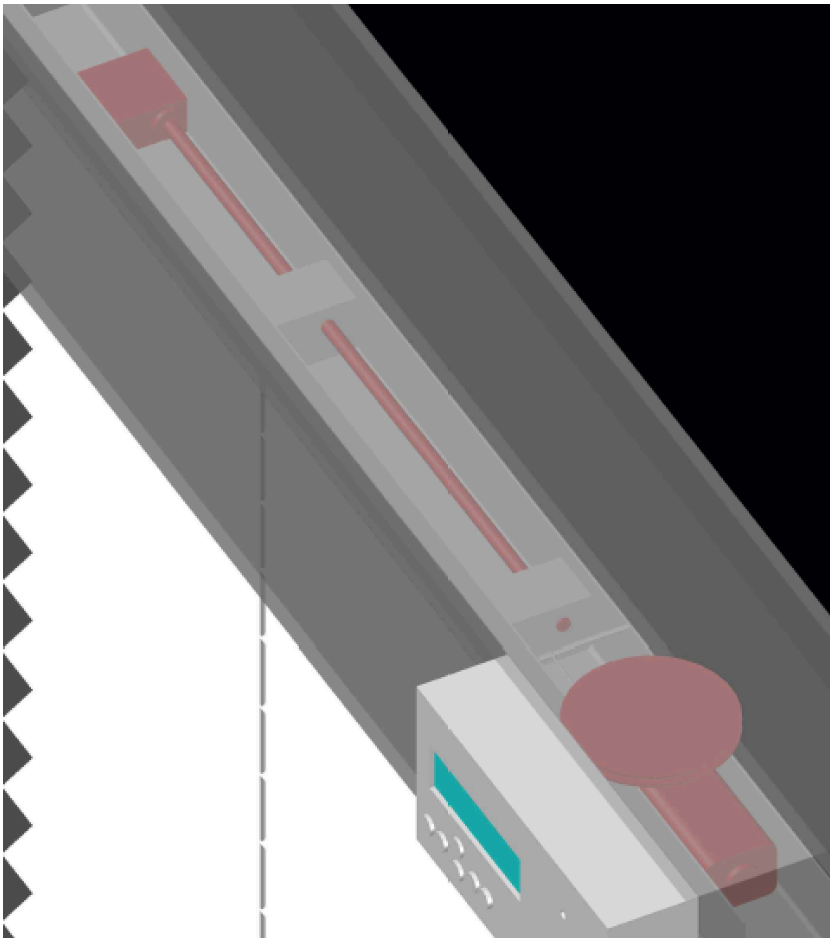

Figure 5: Actuators shown in Red

Both actuators fit securely within the aluminum housing of the blinds. The servo is linked to the original tilt mechanism, and the DC geared motor is connected to a bobbin that winds the original lift chord of the blinds. A rod and roller is used to prop up the chord so it does not wrap outside of the bobbin.

3.5 Electrical System

The Arduino Mega micro controller will control the entire Smart Blinds System; it is responsible for reading all the sensors and running the actuators. The circuitry will be soldered onto a prototype board, and will be secured to the internal of the blinds housing. The pin assignment for the Arduino Mega is listed in the table below.

Modules	Pin Assignment
RTC	20,21,5V,GND
Light Sensor	20,21,3V3,GND
L298 Motor Controller	2,12,9,5V,GND
Servo Motor	8,5V,GND
Photo-resistor	3,5V,GND
LCD Button Shield	A0,D4-D9, 5V,RST, GND
Battery and Charging Circuit	Power

Table 2: Pin Assignment

All the modules will be attached to the prototype board, with the exception of the light sensor; since it is reading the interior light conditions we fixed the sensor inside the LCD control panel. Connecting our prototype board to the Arduino with an Ethernet cable will minimize wires.

4. User interface

The user interface of our system is designed to allow the user adequate control over the Smart Blinds System. By interfacing with the LCD and pushbutton Arduino shield, our microcontroller gives its user numerous control options that dictate the Smart Blinds System's use.

From the user's perspective, the blinds could also function as a normal blind. However, it is our primary design goal to create an interface such that a guest with no prior knowledge or awareness of this product would be able to use it additional features by following simple instructions. A user should never have to make additional consultations with a users' manual for any aspect of operation, or our intuitiveness target will not have been met. We utilize the 6 push buttons available on the LCD button shield for user input. The stacked LCD button shield and Arduino Mega will be placed into a box case designed to allow the user to view the LCD screen and access the pushbuttons.

Below is our menu tree diagram that details the different menu options accessible through the 16X2 LCD screen:

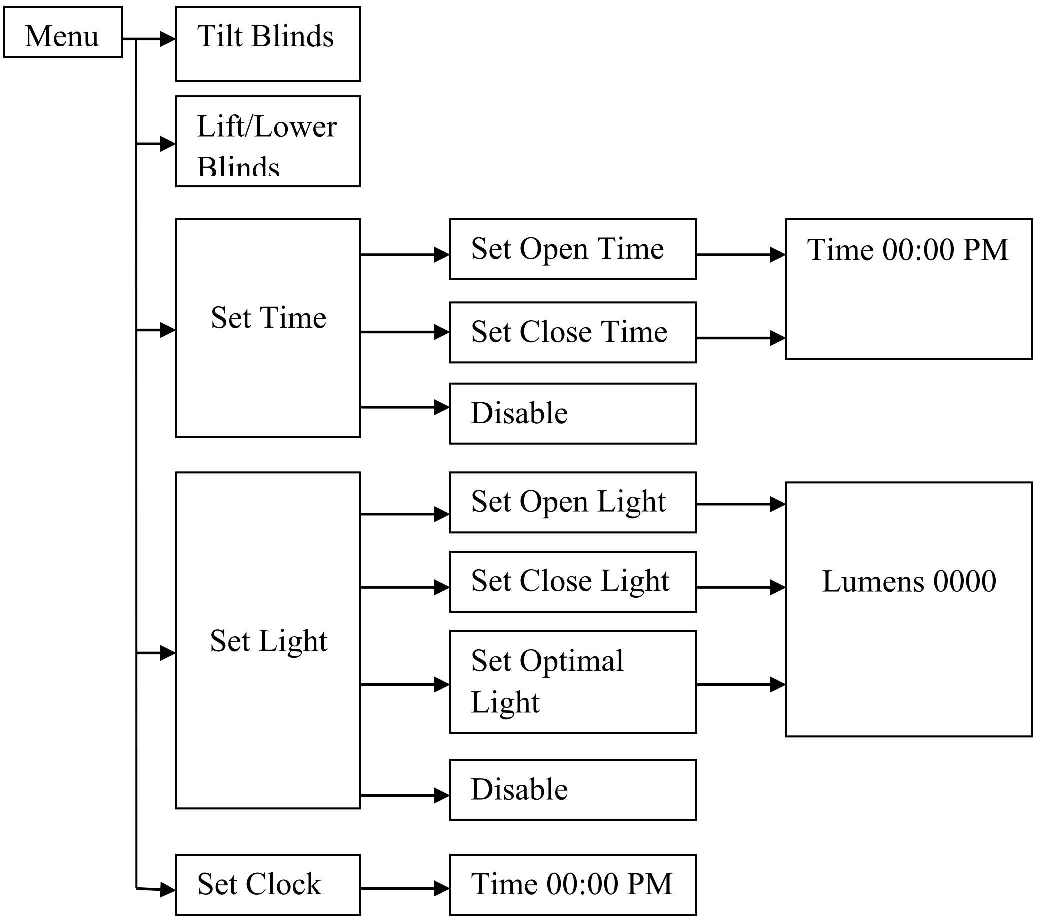

Figure 6: Menu Tree

4.1 LCD Button shield

The LCD Button Shield consists of a 16x2, black character, green backlight LCD with a keypad consisting of 5 keys — select, up, right, down and left. The shield requires averagely 5-volt input for standard operation and current limit ranges from 39.6 mA to 65.0 mA.

Item	Min	Typical	Max	Unit
Voltage	4.3	5.0	5.5	V
Current	39.6	53.3	65.0	mA
Dimension	81.1 ×55.8×22.7			mm
Net Weight	55.0			g

Table 3: Specification of the LCD Shield [9]

The LCD Button shield could be connected directly to the Arduino Mega Board by overlapping the correct pins as below.

Arduino Pin	A0	D4	D5	D6	D7	D8	D9
LCD Shield	Buttons	Bit 4	Bit 5	Bit 6	Bit 7	RS	Enable

Table 4: Pin connection between Arduino and Shield. [9]

5. Blind Tilt Mechanism

5.1 Servo Motor

The other component involved in the system is the Blind Tilt Mechanism, which consists of two parts, the first being a servomotor (figure below). We decided to choose the servomotor as it provides us with the ability to carefully and precisely control the angle of the tilt on the blind flaps. The servo at 2.6 kg provides enough torque to tilt the blinds from fully closed to completely open and any other angle in between. The servo motor shaft is connected to the second part of this mechanism, the tilt rod, and the motor itself is powered by pin 8 on the Arduino Mega.

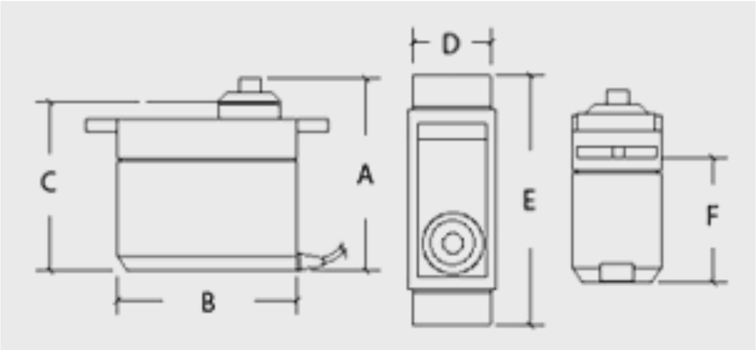

Figure 7: Servo Dimensions

Weight (g)	16
Torque (on.in)	36
Speed (Sec/60deg)	0.11
A(mm)	34
B(mm)	29
C(mm)	29
D(mm)	11
E(mm)	40
F(mm)	20

Table 5: Servo Specifications

5.2 Tilt Rod

The second part of the blind tilt mechanism is the tilt rod. The tilt rod has connects to the strings supporting the blinds themselves and allowing them to be tilted by their user. The three strings which are shown in the figure below rotate with the tilt rod, allowing the blinds to be opened or closed.

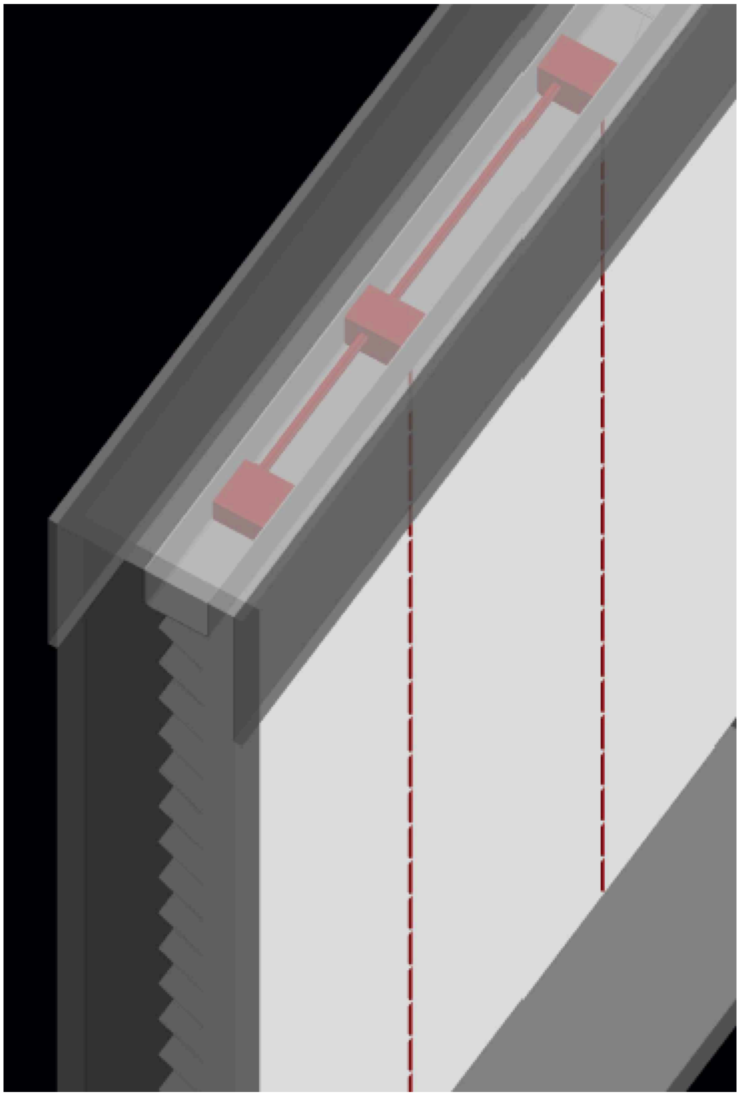

Figure 8: Tilt Rods and String Shown in Red

6. Blind Lift Mechanism

6.1 DC Motor

The main lifting mechanism consists of the DC motor attached to the string spool. The L298 DC motor controller that is connected to the Arduino Mega powers the DC motor. We decided to choose the DC motor because we needed a motor that could pull the weight of the blinds at a constant rate in addition we preferred something that could provide us with simple operation. The other solution that we could have opted for was the 5V stepper motor; however the stepper motor that could fit inside our blinds enclosure did not provide us with an acceptable amount of torque. For the blind lift mechanism we wanted to control how the blinds were lifted and how they were lowered. To lift the blinds we simply turn the motor on via the DC motor controller and the blinds start to get lifted and reveal the whole window. To lower the blinds we just reverse the polarity on the contacts of the DC motor again via the DC motor controller, this causes the blinds to be lowered.

6.2 String spool

In order to lift the blinds up we need to pull the blinds up by the string they are connected to. Therefore when we are lifting the blinds up there is excess string that has to be taken care of. To address this we use a 2 inch spool (**Figure below**) onto which we wind the string. The spool itself is attached to the shaft of the DC motor, as the motor turns the spool spins wrapping the string around accordingly and in turn lifting the blinds. The spool is angled in order to compensate for the tension the blinds are asserting on the string. If the spool was not angled then as the spool spins the string will fail to wind properly onto the spool thus putting unnecessary stress on the DC motor.

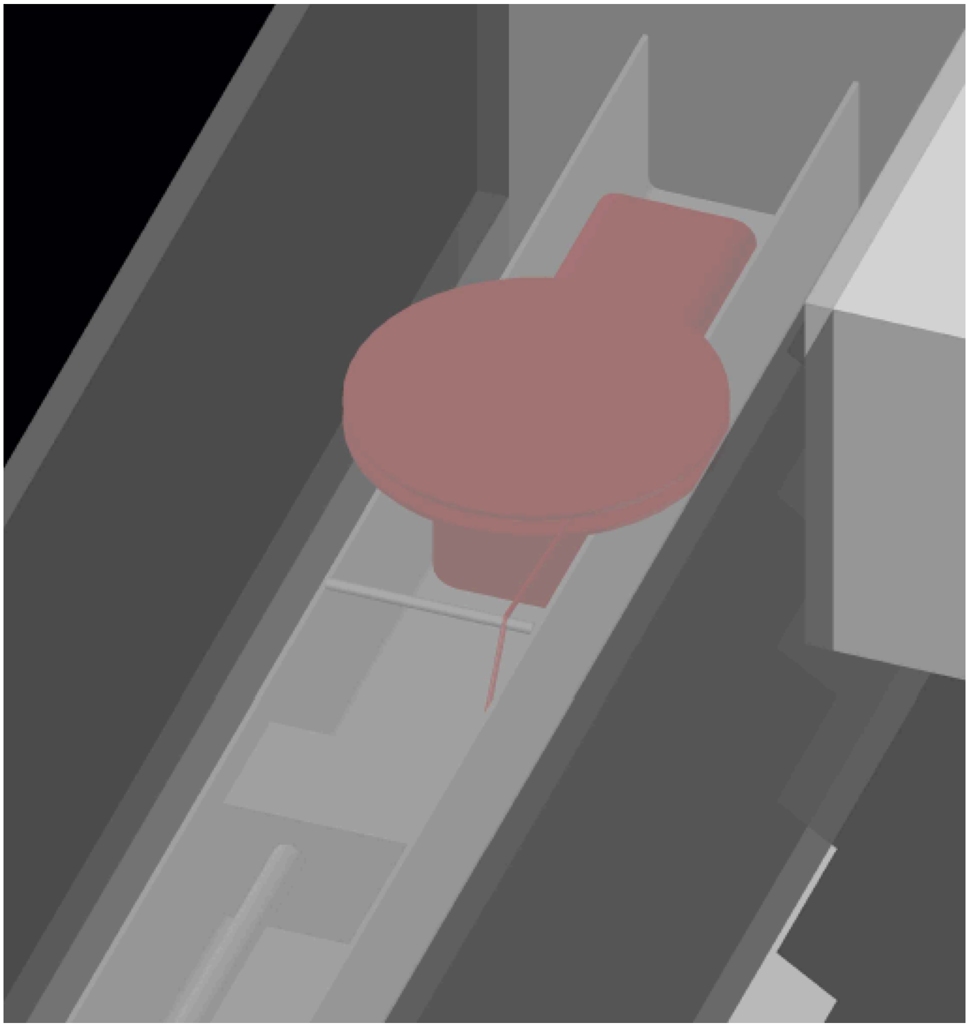

Figure 9: DC Motor With Spool Shown in Red

6.3 L298 DC Motor Controller

The L298 DC motor controller is another part of the blind lift mechanism. The motor controller provides us with the ability to turn the motor on and off whenever need be. This functionality is provided via power and an enable input that allow for the motor to be turned on and off. The L298 motor controller also allows us to easily reverse the direction of the motor by reversing the polarity of the contacts on the DC motor. This motor controller can power two different motors however we only needed to power one DC motor. Therefore in reference to the figure below we are only using pins 1 – 9.

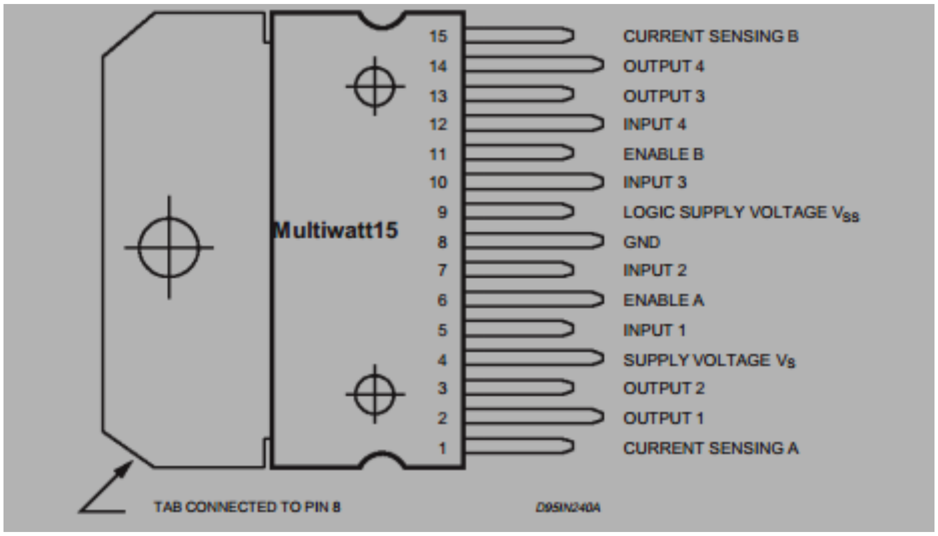

Figure 10: L298 Motor Controller

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100μs) –Repetitive (80% on –20% off; ton = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (Tcase = 75°C)	25	W
T_{op}	Operating Temperature	–25 to 130	°C
T_{stg} , T_j	Storage and Junction Temperature	–40 to 150	°C

Table 6: L298 Requirements

7. Electrical Hardware and Control System

7.1 Arduino Mega 2560 Microcontroller

Arduino Mega Microcontroller acts as the brain of our control system, receiving ambient light data as well as user inputs to achieve a fully functioning system. With the help of L298 motor controller circuit, it allows full control over our motors speed and direction. The Grove RTC is added to enable real time information. Arduino LCD and button shield collect the user inputs.

A couple of other options were taken into consideration when choosing the brain of the project, such as Arduino Uno microcontroller and Raspberry pi. Compared to Arduino Uno, the Arduino Mega 2560 comes in a bigger size both physically and more importantly in terms of available Flash memory and RAM. The Mega has 256 KB of flash memory and 8 KB of RAM whereas the Uno only offers 32 KB of flash memory and 2 KB RAM. Moreover, the Mega 2560 offers additional I/O pins and hardware serial ports. Being able to run two I2C devices at once was also a very desirable feature, which became necessary as we used an I2C clock and digital light sensor. Even though a smaller physical size is favoured for this project, we decided functionality was most important as physical space in our system could be adjusted with a larger blind enclosure. Raspberry Pi could have fulfilled the same requirements for this project but was overlooked due to the team’s familiarity with the Arduino IDE.

Input Voltage	7 – 12 Volts
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
Flash Memory	128 KB (4 KB used by boot loader)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Table 7: Arduino Mega 2560 Specification

7.2 Grove Real Time Clock

The real time clock module is based on an I2C protocol supported by its DS1307 chip that connects to a Lithium cell battery for consistent power. The module provides detailed clock and calendar information in the format of seconds, minutes, hours and date, month, and year respectively. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The user could choose the clock format in either 24-hour or 12-hour mode in with the AM/PM indicator.

The following is the key specification for the RTC SEN12671P module. The small size of the unit could fit perfectly within the range of the Arduino Mega board and it is connected to SCL (serial clock) and SDA (serial data) pins on the Arduino board. On the Arduino Mega 2560 these are pins 20 and 21 respectively.

Items	Min
PCB Size	2.0cm*4.0cm
Interface	2.0mm pitch pin header
IO Structure	SCL,SDA,VCC,GND
ROHS	YES

Table 8: Grove RTC Key Specification [4]

Table 2 introduces the electronic characteristics of the module. The unit takes a 5 V power input. However, a 3-Volt CR1225 lithium cell in the battery-holder is used in order to gain robust performance and to maintain clock readings as the Arduino is powered down. In many cases the RTC'S crystal may not oscillate if it is only connected to primary power.

Items		Min	Norm	Max	Unit
VCC		4.5	5.0	5.5	V
Logic High Level Input		2.2	-	VCC+0.3	V
Logic Low Level Input		-0.3	-	+0.8	V
V_{BAT} (Battery Voltage)		2.0	3.0	3.5	V
V_{BAT} Current	(OSC ON),SQW/OUT OFF	-	300	500	nA
	(OSC ON), SQW/OUT ON (32kHz)	-	480	800	nA
	Data-Retention Current (Oscillator Off)	-	10	100	nA

Table 9: Grove RTC Electronic Characteristics [5]

7.3 BH1750FVI Ambient Light Sensor

BH1750 FVI is a digital Ambient Light Sensor IC using the I2C serial bus interface. This IC is makes it possible to detect wide range of light readings at high resolution (1 to 65535 lux). The following table details the operating conditions for the light sensor:

Parameter	Symbol	Min.	Typ.	Max.	Units
Vcc Voltage	Vcc	2.4	3.0	3.6	V
I^2C Reference Voltage	VDVI	1.65	-	Vcc	V

Table 10: Light Sensor Operating Conditions [10]

The light sensor is connected to pin 20 and 21 on the Arduino Mega 2560 board, sharing the same I2C bus with the Grove real time clock.

7.4 Cadmium Sulphur Photocell

The photocell is about 5 mm in diameter and its resistance ranges from 10 KΩ to 200 KΩ depending on the amount of light it's exposed to. It is able to detect light in the 400 nm to 700 nm rage and uses less than 1 mA of current during operation. In respect to our design the CDS photocell is installed on the outside facing side of our blinds to obtain a rough reading of outside lighting conditions.

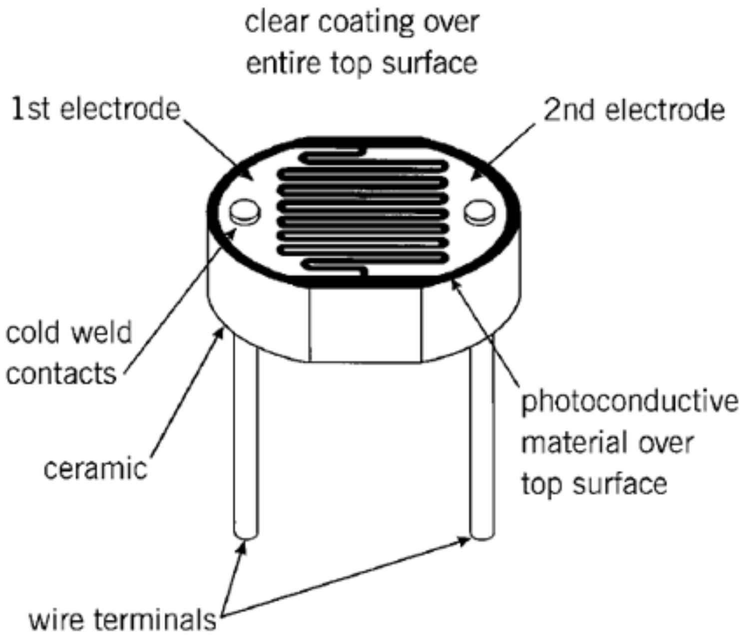

Figure 11: CDS Photocell [7]

ABSOLUTE MAXIMUM RATING (TA)= 23°C UNLESS OTHERWISE NOTED

SYMBOL	PARAMETER	MIN	MAX	UNITS
V_{pk}	Applied Voltage		150	V
$P_{d\ \Delta po/\Delta t}$	Continuous Power Dissipation		100	mW/°C
T_O	Operating and Storage Temperature	-30	+75	°C
T_S	Soldering Temperature*		+260	°C

Table 11: CDS Photocell Specification [8]

8. System Test Plan

The following test plan has been designed to implement specific system testing for quality assurance. The testing steps and expected results must be accounted for during the system testing stages. We must not only guarantee that there are no faults in our design, but also to ensure that it continues to perform as desired over a long period of time. Our test plan will therefore be divided into 3 main steps: First testing the systems individual components before they are added to the design, then testing the combined modules and how they interact together, and finally testing the entire prototype system.

Smart Blinds System Testing Sheet - BikoTech Automated Systems - Company Internal Use Only Software Version ____________________ Product Version ____________________ Page 1 of _____ QA Tester _____________________________ Date of Testing _____________________________		
Subsystem	**Test Procedure**	**Required Results**
Component Tests		
Arduino 2560 Mega Microcontroller	Upload the “Blink Example” code onto the microcontroller’s memory and connect a standard LED to its digital output pins.	Software should run on the microcontroller. Each digital output pin should be able to supply power
	Using the sample example code, connect an LED to the board’s analog pin. Utilize a multimeter to detect voltage provided by the analog pins. Sweep the input voltage and note the mutlimeter values.	Analog pins should be able to supply between 0 and 5 volts.
	Connect an ammeter to the boards 5 volt output and observe the amount of current that can be drawn from the Arduino.	We expect that at least 500 mA of current should be available from this output pin.
L298 Motor Controler	Assemble the motor controller circuit and use a DC power supply with input voltage of 9 V. Connect an ammeter in series with the output of the controller. Cycle the power supply voltage up to 30 volts and observe output power.	The motor controller should be able to supply up to 2 amps of current when operating at maximum input voltage.
	Still connected to the same circuit, connect a multimeter to observe changes in voltage across the circuits sensing resistor.	As the input voltage to the motor controller is varied the voltage drop across this resistor should change proportionately.
Bidirectional DC Motor	Utilize a 9 volt battery connection to power the motor. Then switch the pin contacts and observe direction.	The motor should operate in both directions at a similar speed.
	Wrap a string around the motors axis of rotation and engage the motor with a 9 volt battery. Tie this string to the pull string of our blinds.	The motor must be able to lift our aluminum blinds without more than a 50% reduction in rotational speed at this input voltage.
Servo Motor	Connect to the Arduino 5 V power supply, ground, and an analog pin. Upload the “Servo Sweep” example code and observe results.	The servo motor must be able to rotate between 0 and 180 degrees, always stopping at an identical place in its rotation.
	Connect a string to the corner of one of the servo’s spokes. Tie this string to the blind pull string and allow it to sweep in both directions.	The servo must be able to lift or lower the blinds only slightly as the turning mechanism for the blinds require much less torque than the lifting mechanism.
I2C Digital Light Sensor BH1750	Load the “Light Test” file onto the Arduino. Connect the sensor to the required pins. Move the light sensor between different environments to test the accuracy of their readings. If readings are off by a significant degree (+-10%) establish new benchmarks for each different lighting environment.	Ensure that the light readings are displayed on the serial consol. Typical lux benchmarks are 0.001 – 0.02 for night conditions, 50 – 500 for cloudy conditions, and 100 – 1000 for sunny conditions.
Real Time Clock	Connect the clock to the I2C pins of the Arduino microcontroller. Using the Wire.h library create a simple programing and read back script to allow the setting of the RTC and observation of its performance. Attempt to reprogram the time on the RTC during operation.	The real time clock should be able to be programmed through the Arduino microcontroller, and its time should increment as accurately as a standard clock. It must be able to display these values and have its

		time reset at any moment.
LCD and Button Shield	Load the “LCD Test” script onto the Arduino. This script lights each of the pixels of the LCD screen to test for faults. And it uses a performance test to judge the speed at which the screen updates.	Each pixel of the 16x2 LCD panel must update quickly and accurately. There should be no dead pixels.
	With the LCD panel connected to the Arduino, utilize the “Bounce Test” script which ties button inputs to screen outputs. Press each button in close succession and attempt to generate a delayed or multiple outputs on the LCD screen.	There needs to be only one change to the LCD screen per button press. If debouncing occurs we may avoid this by adding delays for each button press.
Solar Array and Battery	Connect the solar panels output connector to a mutlimeter and observe the voltage and current provided. Utilize different light sources and gauge their effectiveness. Attempt to create a maximum voltage environment with any external light source.	Our solar panel must provide at least 9 volts and no less than 100 mA in order for it to be used effectively with our rechargeable battery.
Aluminum Blinds	Perform basic usage tests on the blinds. Inside their upper housing there must be sufficiently low friction on the pulley system such that they can be lifted and lowered with ease.	The blinds should not kink or bend during normal usage. They must be resistant to wear and be able to endure constant lifting and lowering.
Integrated Module Tests		
Kinematics	After the DC motor has been installed inside the upper housing of the blinds system, utilize the Arduino microcontroller to toggle its speed and direction. Obtain a time measurement of how long it takes to fully lift and fully lower the blinds.	The motor must be able to lift and lower the blinds in less than 10 seconds. Excess vibration must not resonate through the device, and strain on the motor must not be obvious.
	Once the servo motor is installed inside the blinds housing, run its sweep script and observe how the blinds tilt. Orientation of the blinds tilting mechanism may need to be altered to better align with the servo.	The servo motor must be able to tilt the blinds fully closed to fully open in less than 5 seconds. High enough resolution must be possible such that specific angles of open and closing can be programed.
	Connect the DC motor controller and DC motor to the blind system. Allow the current sense output to be input into an Arduino analog-input pin. Load the “Current Sense” script and attempt to lift the blinds beyond their maximum lift height.	The script should read the current sense voltage coming from the motor controller and be able to stop the motor from damaging the blinds.
Communication	Connect both the real time clock and the digital light sensor the Arduino Mega’s SDA and SLC pins. Use a 1.5 K Ohm pull-up resistor between the 5 V supply and each of the power inputs. Attempt to read back from both of these devices operating at the same time. Perform the same tests as done in the individual component tests to ensure they are operating as expected.	According to the I2C serial protocol both of these devices can operate together on the same bus. By using their component tests it will be guaranteed that they operate as effectively together as they would on a separate bus.
	Program a basic menu onto the microcontroller. It must allow the user to shift between menus and make selections using the push buttons.	The push buttons and LCD need to accurately interface with the user. Displaying correct menu settings and allowing the manual programing of new settings via setting preprogramed variables. There must

		not be a delay of more than half a second between button presses and LCD menu updates.
Power	Connect all components to the microcontroller and power it with a 9V battery. Allow the real time clock to run and light sensor to continuously obtain data. Toggle between motor functions with the LCD screen connected.	The Arduino Mega 2560 must be able to supply adequate power such that all of these devices can operate in parallel. During motor operation the LCD screen should not dim whatsoever. The lifting and tilting speed of the blinds must not be negatively affected in comparison to when the motors were used independently.
	With all devices connected and motors toggling between operations, allow the system to operate connected to the solar panel and charging circuit. Ensure that maximum light is supplied for the solar panel.	The devices should not be able to drain the battery enough to allow the system to lose power while the solar panel is supplying 9 volts to the charging circuit.
System and User Acceptance Tests		
Typical Usage Cases	Allow a user unfamiliar with the product to attempt to set a time to open or close the blinds.	Any user should be able to become familiar and understand the devices interface within 10 minutes of usage.
	Instruct a user to set the amount of light they want to enter their room.	The descriptions of the light amounts in the menu must be adequate such that the user can make a decision in line with their expectations.
Corner Use Cases	Instruct the user to attempt to break the devices control routine by programming a time to close or open the device, and then manually sending an open or close signal to the microcontroller.	The software must be able to manage multiple and sometimes conflicting requests. The variables that dictate whether the blinds should be open or closed next must not be miss-assigned during this stress test.
	Allow the user to set a standard light amount they want to enter their room, and then immediately override this by sending an open or close signal manually to the microcontroller.	The software must be able to recognize this conflicting request and allow the user to open or close the blinds as they did manually and leave them in that position.

9. Conclusion

This document has specified all of the design requirements for BikoTech's Smart Blinds System with the goal of our product functioning in a consistent, safe, and in an efficient manner. Also included is the complete test plan that will be executed during our projects quality assurance phase. With this document BikoTech will be able to implement all required functions of our prototype model. The first stage is planned to be complete by December 3, 2013.

If we meet enough requirements outlined in this document we will design a second stage in which we will investigate further development of this product. Further development may include enhancements to the user interface of the design, and changes to the control system itself to ensure a more marketable and affordable product.

10. References

[1] http://www.homedepot.ca/product/2-inch-faux-wood-blind-white-72-inch-x-48-inch/925064

[2] http://www.homedepot.ca/product/3-1-2-inch-vertical-blinds-white-78-inch-x-84-inch/925066

[3] http://www.homedepot.ca/product/18-in-x-48-in-white-1-aluminum-blind/969181

[4] http://www.seeedstudio.com/wiki/Grove_-_RTC#Key_Specification

[5] http://www.seeedstudio.com/wiki/Grove_-_RTC#Electronic_Characterstics

[6] http://www.instructables.com/id/BH1750-Digital-Light-Sensor/

[7] http://learn.adafruit.com/photocells

[8] http://learn.adafruit.com/system/assets/assets/000/010/127/original/PDV-P8001.pdf

[9] http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Dev/Arduino/Shields/16x2_LCD_Shield_Spec_Rev_1.1_20130318.pdf

[10] http://www.rohm.com/web/global/search/parametric/-/search/Digital%2016bit%20Serial%20Output%20Type%20Ambient%20Light%20Sensor%20ICs