

# Senior Design 1

*Solar-Powered Flower Sculpture*



**COLLEGE OF ENGINEERING  
AND COMPUTER SCIENCE**

*Department of Electrical Engineering and Computer Science*

*University of Central Florida*

*Dr. Lei Wei*

*Draft Senior Design Project Document*

## **Group 21**

Nickolas DeVito	Computer Engineer	<a href="mailto:ndevito@knights.ucf.edu">ndevito@knights.ucf.edu</a>
Kibwe Williamson	Computer Engineer	<a href="mailto:kwilliamson12@knights.ucf.edu">kwilliamson12@knights.ucf.edu</a>
Kelechi Ukachi-Lois	Computer Engineer	<a href="mailto:kukachil@knights.ucf.edu">kukachil@knights.ucf.edu</a>
Mahaley Vann	Electrical Engineer	<a href="mailto:mahaley.vann@knights.ucf.edu">mahaley.vann@knights.ucf.edu</a>

# Table of Contents

- 1.0 Executive Summary..... 1
- 2.0 Project Description..... 2
  - 2.1 Motivation..... 2
    - 2.1.1 Existing Products..... 2
    - 2.1.2 Similar Projects/Proposals..... 3
      - 2.1.2a Smart Umbrella..... 3
      - 2.1.2b Dancing Water Display..... 3
  - 2.2 Project Goals..... 4
  - 2.3 Design Summary..... 4
    - 2.3.1 Sculpture Display..... 4
      - 2.3.1a Presentation..... 4
      - 2.3.1b Base..... 5
      - 2.3.1c Petals..... 5
    - 2.3.2 Hardware..... 5
      - 2.3.2a LEDs..... 5
      - 2.3.2b Photovoltaic Cells..... 5
      - 2.3.2c Power Control..... 6
      - 2.3.2d Wireless Module..... 6
      - 2.3.2e Microcontroller..... 7
      - 2.3.2f Texas Instruments TLC5940..... 8
    - 2.3.3 Software..... 8
      - 2.3.3a Application..... 8
      - 2.3.3b User Control Interface..... 9
    - 2.3.4 High Level Architecture..... 10
  - 2.4 Component Requirements..... 12
    - 2.4.1 Solar Statue..... 12
    - 2.4.2 LEDs..... 13
    - 2.4.3 Power Supply..... 14
      - 2.4.3a Solar Power Design..... 14
      - 2.4.3b PCB Design..... 15
      - 2.4.3c Battery..... 15
    - 2.4.4 Wireless Module..... 15
    - 2.4.5 Microcontroller..... 17
    - 2.4.6 Application..... 17
      - 2.4.6a Control Interface..... 18
    - 2.4.7 Texas Instruments TLC5940..... 18
- 3.0 Standards and Design Constraints..... 20
  - 3.1 Economic..... 20
  - 3.2 Time..... 20
  - 3.3 Environmental..... 21
  - 3.4 Social and Political..... 21
  - 3.5 Ethical..... 21
  - 3.6 Health and Safety..... 22
  - 3.7 Manufacturability and Sustainability..... 24
    - 3.8 Android Standards..... 24

3.9 Bluetooth Standards.....	26
4.0 Component Research.....	38
4.1 Solar Statue.....	38
4.1.1 LEDs.....	42
4.1.2 Power Supply.....	45
4.1.2a Photovoltaic Cells.....	49
4.1.2b PCB Design.....	50
4.1.2c Battery.....	51
4.1.3 Wireless Module.....	53
4.1.4 Microcontroller.....	57
4.2 Software.....	62
4.2.1 Existing Similar Software Projects.....	62
4.2.1a Project 1.....	62
4.2.1b Project 2.....	62
4.2.2 User Control Interface.....	63
4.2.2a Language Considerations.....	63
4.2.2b Platform Considerations.....	64
4.2.2c Communication Considerations.....	64
4.2.3 Hardware Programming Languages.....	64
4.2.3a Assembly Language.....	65
4.2.3b C Language.....	65
4.2.3b C++ Language.....	66
5.0 Design.....	68
5.1 Design Method.....	68
5.2 Hardware.....	68
5.2.1 Power Supply.....	69
5.2.1a PCB Design.....	69
5.2.1b Battery.....	71
5.2.2 Microcontroller Interface.....	72
5.2.2a Interfacing with LEDS.....	73
5.2.2b Pulse Width Modulation.....	76
5.2.3 Photovoltaic Cells.....	76
5.2.4 LED Circuitry.....	76
5.3 Software.....	77
5.3.1 Application.....	77
5.3.1a Platform.....	77
5.3.1b Android Interface.....	78
5.3.2 Microcontroller.....	85
5.3.2a Arduino IDE.....	86
5.3.2b Bluetooth.....	87
6.0 Testing.....	94
6.1 Safety Procedures.....	94
6.2 Breadboard Testing.....	94
6.2.1 Testing Conditions.....	94
6.2.2 Measuring Approach.....	95
6.2.3 Implementation.....	95

6.3 Power Supply Testing.....	105
6.3.1 Testing Conditions.....	106
6.3.2 Measuring Approach.....	106
6.3.3 Implementation.....	107
6.4 LED Testing.....	112
6.4.1 Testing Conditions.....	112
6.4.2 Measuring Approach.....	112
6.4.3 Implementation.....	113
6.5 Wireless Module Testing.....	116
6.5.1 Testing Conditions.....	116
6.5.2 Measuring Approach.....	117
6.6 Microcontroller Software Simulations.....	118
6.6.1 Testing Conditions.....	118
6.6.2 Measuring Approach.....	118
6.7 Microcontroller and External Circuit PCB Interface.....	119
6.7.1 Testing Conditions.....	119
6.7.2 Measuring Approach.....	120
6.8 Application Interactions Testing.....	120
6.8.1 Testing Conditions.....	118
6.8.2 Measuring Approach.....	119
6.9 Final Testing Phase.....	120
6.9.1 Testing Conditions.....	121
6.9.2 Measuring Approach.....	122
7. Administration.....	123
7.1 Budgetary/Financial Data.....	123
7.2 Project Timeline.....	125
7.2.1 Work Distribution.....	127
7.3 Learned Materials and Skills.....	128
8. Conclusion.....	129
8.1 Acknowledgments.....	130
9. Appendices	
9.1 Appendix A: References	
9.2 Appendix B: Data Sheets	

## List of Figures

Figure 1: Wireless Module (made by Kelechi Ukachi-Lois).....	7
Figure 2: High Level Design (By Kibwe Williamson).....	10
Figure 3: Component Flow Diagram (By Kibwe Williamson).....	10
Figure 4: Bluetooth Standards [24].....	28
Figure 5: Examples of Bluetooth master/slave piconet topologies .....	31
Figure 6: Example of Bluetooth unique 48-bit address .....	32
Figure 7: Example of Bluetooth Device Names .....	33
Figure 8: <i>Example of RS-232 vs Serial Port Profile (SPP)</i> .....	35
Figure 9: <i>Example of HID interface, from RN-42-HID User's Guide</i> .....	36
Figure 10: <i>Example of Advanced audio distribution profile (A2DP)</i> .....	37

Figure 11: Example of A/V Remote Control Profile (AVRCP).....	37
Figure 12: Red Oak.....	40
Figure 13: White Oak.....	41
Figure 14: First Circuit Prototype.....	46
Figure 15: MAX712 Schematic.....	47
Figure 16: LM7805 Schematic for Circuit.....	48
Figure 17: MAX639 Inner/Outer Schematic.....	49
Figure 18: Printed Circuit Board Schematic.....	51
Figure 19: PCB Schematic using EAGLE.....	70
Figure 20: PCB Design.....	71
Figure 21: 9 V NiMH.....	72
Figure 22: Voltage-Current Graph of LED (By Kibwe Williamson).....	74
Figure 23: Adding Resistor before LED to Increase Current .....	75
Figure 24: How the LEDs are hooked up to the Arduino.....	75
Figure 25: Voltage-Current Characteristics.....	75
Figure 26: Photovoltaic Cells in Series.....	76
Figure 27: LED Strip Connections.....	77
Figure 28: Application Utility Menu.....	79
Figure 29: Application Home Screen Design.....	81
Figure 30: Application Clock Mode Screen Design.....	83
Figure 31: Application Lamp Mode Screen Design.....	84
Figure 32: Application Party Mode Screen Design.....	85
Figure 33: Arduino Connection.....	87
Figure 34: Android Symbol.....	88
Figure 35: How Bluetooth Reacts with Arduino and Android Application.....	89
Figure 36: Prototype Android Application Interface.....	90
Figure 37: Bluetooth Pairing Screen from Application.....	91
Figure 38: MAX712 Chip.....	95
Figure 39: MAX712 Circuit - Side View.....	96
Figure 40: MAX712 Circuit - Top View.....	96
Figure 41: MAX639 Chip.....	97
Figure 42: MAX639 Circuit - Side View.....	98
Figure 43: MAX639 Circuit - Top View.....	99
Figure 44: ATmega328 Microchip with Arduino Nano Breakout Board.....	99
Figure 45: TLC5940 Chip.....	100
Figure 46: Arduino Nano - Side View.....	101
Figure 47: Arduino Nano and TLC5940 without LEDs.....	101
Figure 48: Arduino Nano and TLC5940 with LED Wiring.....	102
Figure 49: Combined System Circuit.....	103
Figure 50: Combined System - Side View.....	103
Figure 51: Combined System - Top View.....	104
Figure 52: System with Photovoltaic Cells.....	105
Figure 53: MAX712 Testing Schematic.....	107
Figure 54: MAX712 Tested Values.....	107
Figure 55: MAX639 Circuit to be tested.....	109
Figure 56: MAX639 Tested Values.....	110

Figure 57: Arduino Nano and TLC5940 Test Circuit.....	111
Figure 58: Arduino Nano and TLC5940 Tested Values.....	112
Figure 59: NeoPixel Digital RGBW LED Strips.....	114
Figure 60: LED Strips Connection.....	115
Figure 61: Software Logic Flowchart.....	127
Figure 62: Hardware Block Diagram.....	128

## List of Tables

Table 1: LED Comparisons (By Kelechi Ukachi-Lois).....	14
Table 2: Wireless Data Comparison (By Kelechi Ukachi-Lois).....	16
Table 3: Comparison of LED Lights (By Mahaley Vann).....	45
Table 4: Bluetooth Version Comparison (By Kelechi Ukachi-Lois).....	54
Table 5: Bluetooth Module Comparison (By Kelechi Ukachi-Lois).....	56
Table 6: Bluetooth Module Comparison [45].....	57
Table 7: Microcontroller Specifications (By Kibwe Williamson).....	58
Table 8: Arduino Uno's Specifications (By Kibwe Williamson).....	59
Table 9: Arduino Mega's Specifications (By Kibwe Williamson).....	60
Table 10: MSP430's Specifications (By Kibwe Williamson).....	61
Table 11: Estimated Budget for Sculpture.....	123
Table 12: Estimated Budget for Testing.....	124

# 1.0 Executive Summary

This paper will clearly and sequentially outline each aspect of this Solar Powered Flower Sculpture project. The purpose of this documentation includes but is not limited to, the explaining of, technical and administrative content along with other relevant information. Topics such as, objectives, goals, specifications, research, design, and many more can be expected to be covered within the scope of this paper.

An objective of this project is to draw attention and highlight the fact that structures can and should be built to harvest electricity without destroying the planet. Although this project is on a smaller scale, it shows that there are ways to keep the Earth beautiful while generating electricity. The goal of this Solar Powered Flower Sculpture project is to bring renewable energy in an artistic design. By creating this Solar Powered Flower Sculpture people can enjoy access to the time of day without confining themselves inside or staring at an ordinary analog or digital clock. The Solar Powered Flower Sculpture will accomplish this via an LED interface to tell time. This Solar Powered Flower Sculpture will also act as an LED lamp, for daytime or nighttime use, with color and brightness settings. Another mode will cycle through colored lights on the sculpture, providing a fun display for party or social events. This will be possible with Bluetooth connection through an app that is user friendly. A compatible mobile app can change the settings on these three modes, so that the sculpture can be used separately.

In order to accomplish the Solar Powered Flower Sculpture a thorough understanding of both the software and hardware is necessary. Research and design are major factors of this project. From researching individual components to understanding how each of the components co-exists and interact with each other, there was much research to done.

The hardware involves many different blocks that all need to work harmoniously as one system. The hardware includes, photovoltaic cells, a charge controller, rechargeable batteries, a voltage regulator, a microcontroller, a Bluetooth module and the LEDs. The software also has many different components working together. The software includes, the Arduino programming which controls the microcontroller, the Bluetooth software which acts as a pipeline between the application and Arduino, and the android application which will allow the user to communicate with the Solar Powered Flower Sculpture. With the hardware serving as the base and the software operating on top this Solar Powered Flower Sculpture, it will be able to produce the desired outcome of a user being able to interact with an android graphical user interface to control different functionality of the Solar Powered Flower Sculpture.

## **2.0 Project Description**

This project involves the creation of a Solar Powered Flower Sculpture that is lightweight and portable. The flower's 12 petals can be illuminated by multicolored LEDs based on the input received by an Android platform mobile app. The phone app allows the Solar Powered Flower Sculpture's LED display to set and be used as a two-hand clock (where one or two petals are illuminated at once corresponding to the time), as a lamp (where all twelve petals are illuminated, and LED color/brightness can be adjusted), or to show a light cycled pattern for festive events (where the user can select different patterns). The flower head is connected to a base which holds the photovoltaic cells on its outside, and houses a programmable Arduino microcontroller with a battery power supply. Bluetooth technology is used to connect the instructions of the Android app to the Solar Powered Flower Sculpture's microcontroller and LEDs. The whole of the system is designed to be energy efficient and environmentally friendly, reflecting its flower presentation.

### **2.1 Motivation**

Our group wanted to create a solar-powered device that had many possible uses depending on the type of person using it. This project idea was something that interested all of our members, and was something we could connect with on a personal level. Implementing an LED display, as well as mobile app compatibility, further cemented our project goal for developing a Solar Powered Flower Sculpture with multiple, situational uses. Lovers of the outdoors would connect to its flower design, and its portability would make it a great companion piece for camping trips, providing a source of time and light. People could still use the Solar Powered Flower Sculpture indoors as a desk lamp, or exotic clock. The Solar Powered Flower Sculpture's ability to cycle through different colored displays means it could be a valuable addition to parties and other social gatherings, or as a holiday decoration. Our group was interested in using our combined skills to create an energy-efficient device that could also interact with mobile phones. We hope that whoever uses the Solar Powered Flower Sculpture will find something about it that appeals uniquely to them, and that it will bring people happiness.

#### **2.1.1 Existing Products**

For this next section, the group did extensive research into past senior design projects to grasp ideas as to what will be expected. It also allowed the group to know how to expand and make the sculpture better.



## **2.1.2 Similar Projects/Proposals**

Researching similar projects has its merits and is an important part of research. Carefully researching and observing similar projects can give insight towards how different systems can be designed, but more importantly, can highlight flaws in current methods. Taking a better look can allow a team to invent new strategies which will avoid inefficiencies and allow for a new and fresh design. Having an open mind will allow the development team to notice small details and possibly allow for innovation in some cases.

### **2.1.2a Smart Umbrella**

For the fall 2015 graduating class, a group of students decided to create a smart umbrella for patio tables. This certain smart umbrella consisted of photovoltaic cells, LEDs, light dependent resistors, an on board battery, motors, and a maximum power point tracker. During the daytime the smart umbrella's electrical system was able to observe the direction of the sunlight and rotate the smart umbrella to continue to shade the people below. This rotation also allowed the photovoltaic cells on top of the smart umbrella to continuously receive direct sunlight and in turn continuously charge the on board battery. To add a flare to their design, the group also installed LEDs to the underside of the smart umbrella to provide a light source for the people using it.

This project is similar to the Solar Powered Flower Sculpture seeing as though the position of the petals being almost completely flat to the sky allows maximum sunlight to be acquired for solar power. Also, the addition of LEDs to the bottom of the smart umbrella is similar to the addition of LEDs to the petals of the Solar Powered Flower Sculpture since they were added to benefit the customer with either light or for spectacle.

### **2.1.2b Dancing Water Display**

In the graduating class of spring 2016, a group of students created a dancing water display which utilized water speakers and their own version of a spectrum analyzer. This dancing water display consisted of LEDs, water pumps, and a user friendly smartphone application. To realize their spectrum analyzer the group used a Fast Fourier Transform which sorted through an array of frequencies and signals and delivered inputs to the water pumps. As the frequencies increased and decreased, the water pumps adjusted their power accordingly. To add color to the water display, LEDs were installed at each water pump and would follow the audio inputs that were transmitted to the water pumps. To control everything mentioned above, the group also created a smartphone application that could access the dancing water display via Bluetooth.

Compared to the project in section 2.1.2.1, this project served as the main inspiration for the Solar Powered Flower Sculpture. The connection between the dancing water display and the Solar Powered Flower Sculpture is that both groups wanted to create something customers would think would make the world a more beautiful place. To do this, both groups created a user friendly smartphone application that connected to the project via Bluetooth. Also, adding LEDs that would rotate colors depending on the signals or the patterns that were selected from the smartphone application, makes the project customizable for the consumers.

## **2.2 Project Goals**

The main goal of this project is to create a working Solar Powered Flower Sculpture that can power its LED functionality using solar energy as the only outside source. Other goals are for the LED display to work as desired under different conditions, without causing the Solar Powered Flower Sculpture's battery supply to run out; and the mobile app should connect to the Solar Powered Flower Sculpture's board and LEDs effectively, with all functions acting as intended. The group aims to create something that each member can be proud of, and that can bring happiness to those using it. During the creation of the Solar Powered Flower Sculpture, each member was able to bring their talents and knowledge to help benefit the project, and the group has learned valuable skills from the experience. Lastly, the group aims to meet all project deadlines on time, and complete the Solar Powered Flower Sculpture in a way that satisfies the set budget.

## **2.3 Design Summary**

The following subsections describes the requirements and specifications that have been placed on this project. These subsections cover both hardware and software.

### **2.3.1 Sculpture Display**

For this section, the physical description of the Solar Powered Flower Sculpture will be discussed.

#### **2.3.1a Presentation**

The group as a whole decided that the presentation of the Solar Powered Flower Sculpture would be to bring beauty back into the world without harming the environment around it. To achieve this, the group hopes that the Solar Powered Flower Sculpture will look appealing sitting outside in a customer's front or back yard, and to do this it will need to look as if it almost belonged in nature.

## **2.3.1b Base**

The base of the Solar Powered Flower Sculpture will be constructed out of wood. The top and bottom will be matching at 26.4 inches by 26.4 inches. All four of the side panels will be 26.4 inches by 4 inches. All of the wooden panels will be treated with weatherproofing wood stain since this flower sculpture is intended to be used outside to capture the solar rays from the sun. To also allow for easy portability, there will be handles installed on the side panels. And then for easy access to retrieve the rechargeable battery, a battery pack will also be installed into one of the side panels.

## **2.3.1c Petals**

To compensate for the length of the NeoPixel digital LED strips, each petal will be about 17 centimeters in length. In design aspects, the group is wanting to have each flower petal to be 3D printed at the Texas Instruments Innovation Lab. Once all twelve of the petals have been 3D printed, the group will adhere the NeoPixel digital LED strips to the petals running latitudinal from tip to tip.

## **2.3.2 Hardware**

For this section, the group will discuss the specific hardware used in each section of the Solar Powered Flower Sculpture.

### **2.3.2a LEDs**

The LEDs chosen were NeoPixel digital red, green, blue, and white LED strips. These certain LEDs are contained within a white flex PCB case holding 60 LEDs per meter. This certain LED strip contains white LEDs with a yellow phosphor on one half and then red, green, and blue LEDs on the other half of the strip. The difference between the NeoPixel and regular LEDs is that the NeoPixel are 5050-sized LEDs that contain an embedded microcontroller within the actual LED itself. This then allows each individual LED down the strip be controlled and programmed separately. To run this certain LED strip, a constant five volts DC needs to be delivered from the Arduino Nano microcontroller throughout the whole LED strip.

### **2.3.2b Photovoltaic Cells**

For this Solar Powered Flower Sculpture, the group decided to series stack multiple small high powered photovoltaic cells to achieve a large enough input to charge the rechargeable Nickel–Metal Hydride battery that supplies power to the Arduino Nano microcontroller. By doing this, the group was able to optimize the size the base of the Solar Powered Flower Sculpture since the photovoltaic cells would be small enough to be placed at the outskirts of the petals along the edges

of the wooden base. These certain photovoltaic cells happen to be made from copper indium diselenide which has a dark glossy appearance. With this, the photovoltaic cells will blend into the base since the base will be painted a darker color, such as black.

### **2.3.2c Power Control**

To power the Solar Powered Flower Sculpture, a battery source will be in parallel with the photovoltaic cells and the Arduino Nano microcontroller. This is accomplished by the photovoltaic cells being connected to a charge controller which will then provide a constant charging source for the battery. The battery will then be connected to a switching regulator that will provide a constant voltage source to the Arduino Uno microcontroller without overloading the microcontroller system. With this constant voltage being provided from the switching regulator, the Arduino Uno microcontroller is then capable of constantly providing enough voltage to power the NeoPixel LED strips and operate the program that is uploaded onto the Arduino Uno microcontroller. To successfully have the Solar Powered Flower Sculpture operate, the battery was decided to be of the rechargeable selection. To correctly power the Solar Powered Flower Sculpture, a Nickel-Metal Hydride 9 volt rechargeable battery was selected.

### **2.3.2d Wireless Module**

The wireless module is the pipeline between the Android application and the Arduino. The wireless module holds a vital role in enabling communication between both the Android application and the Arduino. Because, it is what enables the Android application and the Arduino to communicate. Without the wireless module, the Solar Powered Flower Sculpture would not be able to communicate with the Android application which is a required feature.

For this Solar Powered Flower Sculpture, the wireless module must be a duplex. Meaning that both parties will be able to receive and transmit messages. To achieve this, a Bluetooth module will be used. The Arduino will be equipped with a Bluetooth module that has both a transceiver and receiver which will enable the Arduino to transmit and receive messages via Bluetooth communication. The Android application is intended to be run on a smartphone, which have Bluetooth built-in. Meaning that the Android application will be able to transmit and receive messages via Bluetooth communication.

The wireless Bluetooth module works physically with the Arduino microcontroller. The Bluetooth wireless module is to be physically connected to the Arduino microcontroller. The wireless Bluetooth module has four pins to be physically connected to the Arduino microcontroller. The pins that will be connected to the Arduino microcontroller are the transmitter pin, the receiver pin, the V<sub>CC</sub> pin, and the ground pin.

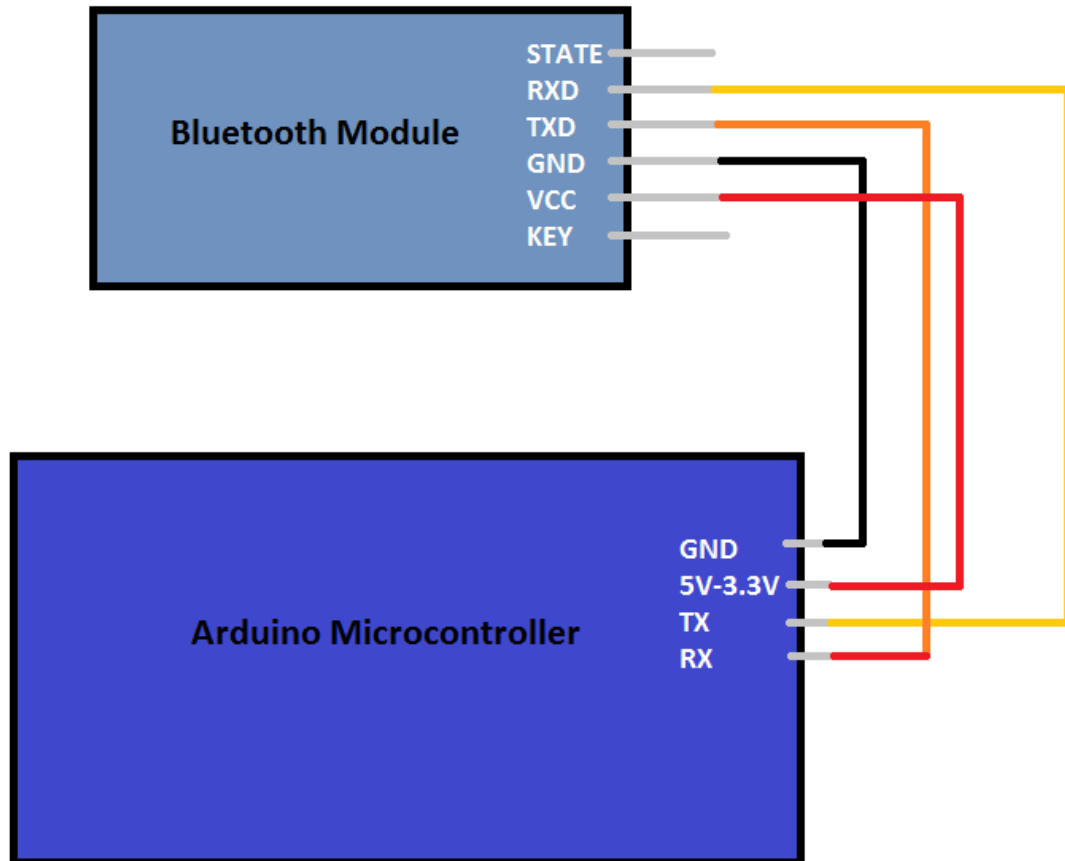


Figure 1: Wireless Module (made by Kelechi Ukachi-Lois)

## 2.3.2e Microcontroller

The microcontroller will need to be able to take a receiver signal from the Bluetooth shield and send a transponder signal to the Bluetooth. The microcontroller will need to provide a grounding pin and a 3.3V pin for the Bluetooth shield. Either directly or through a pin extension module the microcontroller will need to be able to send out twelve PWM signals to the LED strips. The LEDs can be grounded through the Arduino's ground but will get their voltage directly from the battery to prevent them from crashing the microcontroller due to voltage drop. The microcontroller will get its power directly from the battery. The microcontroller will have 5V and 3V pins limiting what can be directly hooked up to the controller.

## **2.3.2f Texas Instruments TLC5940**

To keep the size of the solar statue down we went with the smaller microcontroller the Arduino Nano. This gave us the small size we wanted but a major trade up was that it didn't have enough PWM outputs for our project. This required that we include a LED driver to expand the number of PWM outputs available for the design. To expand the number of PWM outputs available for our project we used the TLC5940 from Texas Instrument. This gave us 16 PWM channels to work with, without having to go with a bigger microcontroller. The TLC5940 is daisy chainable allowing us to make more elaborate displays by just adding another unit. The duty cycle is 12 bit and the limit control is 6 bit.

## **2.3.3 Software**

The software to control the microcontroller will be developed on the open-source Arduino IDE 1.6.12. The software for the microcontroller will control the LED display and the clock for the solar statue. The software will be able to accept commands from the user via the mobile app.

## **2.3.3a Application**

The Arduino microcontroller will receive interpretable data from user input on an Android device mobile application, via Bluetooth connectivity. This app will be created using Android Studio, and with its official library. The app will serve four main purposes for the Solar Powered Flower Sculpture, which are identified as different "modes" on the home screen of the application's menu interface. The first purpose is to power the sculpture on and off from its current state. Although the sculpture is designed and intended to be used over long periods of time, there may be situations where the user wishes to turn the sculpture off. At any time, the sculpture may be turned back on again, including when it is being powered on for the first time. The second purpose is to operate the Solar Powered Flower Sculpture as a clock. This includes setting or changing the time that will be displayed by the LED clock interface. The LED display will move appropriately as the time changes, and as other modes are selected to be used by the application, the time entered by the user will remain up to date. Only two petals, representing the hour and minute hands of a traditional clock face, will be illuminated by their respective LEDs depending on the existing time. The petal representing the hour hand will have only some of its LEDs lit up (to give the appearance of shorter length), which the minute hand petal will have all of its LEDs lit; in the case where these hand overlap, all LEDs will be lit. Also, the app will serve to alter the brightness, and change the color of light being given off by the LEDs. The third purpose of the application is to operate the sculpture as a lamp, in which all LEDs across all petals of the flower sculpture will be illuminated. As with the prior mode, brightness and color of the LEDs can be changed with the app. The fourth purpose of the application is dubbed as a "party mode," in which different LEDs on the solar sculpture will illuminate in a patterned fashion, similar to a laser light display or electronic disco ball. By using the application, users can select from a series of

created display patterns the sculpture can present, as well as adjust the brightness being given off.

### **2.3.3b User Control Interface**

The mobile application will be presented with a control interface designed especially for simplicity and user-friendly capabilities. The home screen of the app shows the four previously described modes that can alter the Solar Powered Flower Sculpture, appearing as clickable icons. In the order displayed, these four modes are titled “CLOCK”, “LAMP”, “PARTY”, and “POWER”. By selecting POWER, a pop-up menu will appear where the user can select to turn the sculpture in an ON or OFF state. By selecting any of the other three modes, the app will open up an entirely new screen, with different clickable icons that, upon being selected, will open pop-up menus. At any time, users can return from these separate screen to the home screen by selecting the back icon, which appears on these screens as an arrow. The CLOCK screen shows three different icon options to set time, change colors, and adjust brightness. By clicking on the time icon, a pop-up menu allows the user to enter the desired HOUR, MINUTE, and AM/PM specification. The color icon opens a pop-up menu with a list of available LED colors that can be chosen to change the HOUR and MINUTE hand respectively (including the same color for both). The brightness icon pops up the ability for users to specify the PERCENTAGE they want the LEDs to give off, where 100% is supposedly full brightness and 0% is supposedly no brightness whatsoever. The LAMP screen shows two different icon options for changing the total color of the light being given off, and adjusting the brightness of the LEDs. These icons open pop-up menus similar to the color and brightness options under the CLOCK screen, only with there being one selection for the color icon as opposed to two. The PARTY screen shows two different icon options representing the light cycle patterns to be selected, and the brightness being given off. These icons also open similar pop-up menus, where for the pattern option the user can select one light pattern to be displayed from a list of several.

## 2.3.4 High Level Design

### High-Level Design

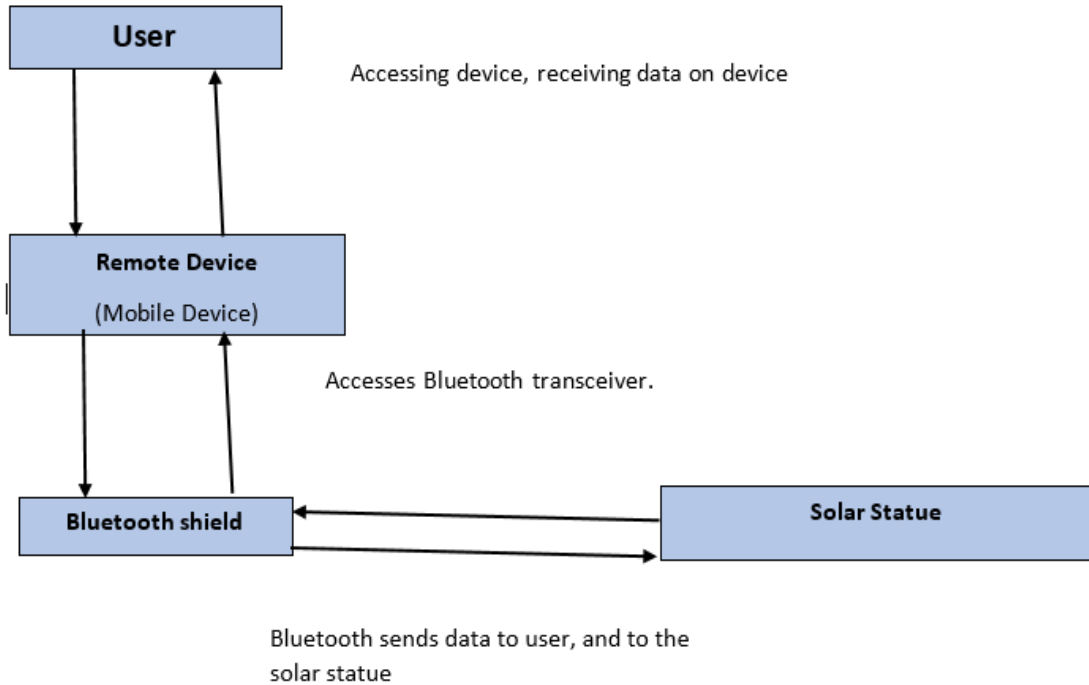


Figure 2: High Level Design (By Kibwe Williamson)

### Component Flow Diagram

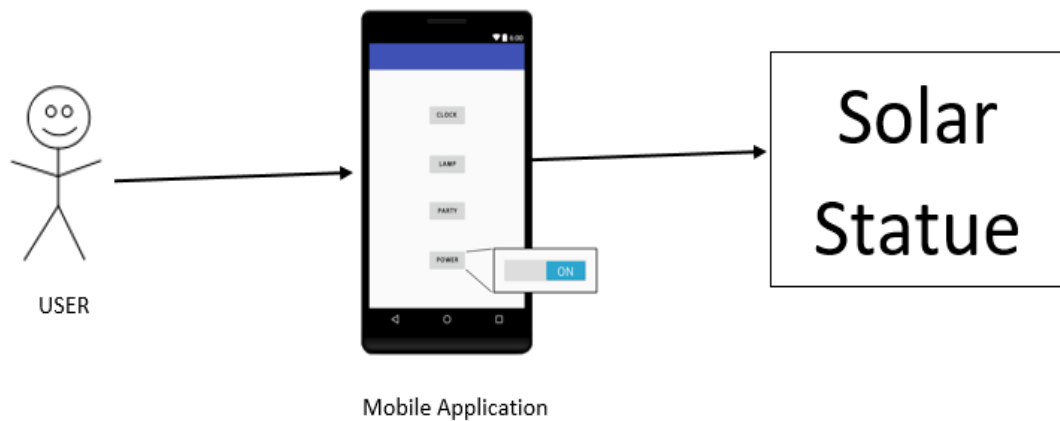


Figure 3: Component Flow Diagram (By Kibwe Williamson)



Bluetooth architecture was utilized for this design. Are design implements the master radio and slave radio symmetric architectural style as each device may operate as a slave and also as a master? The Users remote device will function as the master to initiate the Bluetooth communication link. Once the Bluetooth link is established then the slave may request a master/slave switch so that the slave is now the master.

We can also claim that our design implements the two-tier architectural style as there is direct communication taking place between the slave and master or client and server. For example if the User wants to check the current mode of the solar statue the information will be passed directly from the solar statue to the user's remote device.

### **User-Remote Device:**

The user uses an app on a mobile device to communicate with the solar statue. The remote device will display the information to the user.

### **Remote Device – Bluetooth:**

The remote device utilizes a Bluetooth transceiver to communicate with the solar statue.

### **Bluetooth Shield:**

The Bluetooth Shield makes the master/slave link between the user's remote device and the solar statue.

### **Design Issues:**

Reusability: How long will the system be usable before it becomes outdated, obsolete or is replaced by a better system?

Maintenance: What level of technical expertise is required to operate and maintenance the device.

Testing: What step will be taking to make sure that the device and all of its components are working correctly? What tests can we do to verify that all of the functions are working properly? For example, performance and function testing. How will issues that arise once the system is up and running be handled.

Portability: Will the system be mobile, if so how mobile will it be? How will the system perform in different types of locations?

Security: How secure will access to the device be?

### **Design Trade Offs:**

Deciding to use the smaller Arduino Nano instead of a larger board has easily identified advantages and disadvantages. The Arduino Mega has enough PWM pin so that we wouldn't need any additional components to connect all of our LED strips to the microcontroller. Using the Arduino Nano required us to use an additional component, the TLC5940 LED Driver, to interface all of our LED strips with the Arduino Nano. The advantage is that the smaller microcontroller with the TLC5940 LED Driver combined is significantly smaller than the Arduino Mega, which was important for keeping the size of the final product down for portability.

### **Technical Difficulties and Risks:**

By utilizing the TLC5940 LED Driver for additional PWM output instead of a microcontroller with enough PWM outputs for our project we increased the technical difficulty of the project. Addressing each individual pin through the TLC5940 LED Driver isn't straightforward as it is with addressing a pin directly on a microcontroller. This is also not a technique that any of the group members has done before, so it added an additional learning curve for the hardware programmer.

The above described difficulties are risks that the group had to assimilate before the project could be successfully completed. The security risk of another device other than the intended user's getting control of the device is beyond the scope of this project.

## **2.4 Component Requirements**

The next subsections will detail how each of the Solar Powered Flower Sculpture project's components will meet their own set requirements. These requirements coincide with the overall project goals, and serve as guidance for what our group wishes to achieve for each component. In addition, the subsections provide a more thorough, specific explanation of what each component will be comprised of, and what their main purpose serves to be.

### **2.4.1 Solar Statue**

The wooden paneled infrastructure serving as the Solar Powered Flower Sculpture's base has a few different requirements. The base needs to be sturdy enough to hold all components of the sculpture, and be protective of all sensitive parts used in building it. It also needs to be lightweight. These requirements will be attributed to the type of wood used to build the structure, as well as how efficiently it is constructed. The base must also hold all LED strip-infused petals and solar panels (oriented appropriately), and its interior must include the Arduino microcontroller (with Bluetooth connection component), rechargeable battery, and

appropriate wires. Details of the hardware components of the Solar Powered Flower Sculpture, and their project requirements, are explained in the following subsections.

## 2.4.2 LEDs

Visually the LEDs are the most important components of the project. The user will be able to get information displayed on their mobile device, but the different display mode featuring the brilliance of the LED display is the primary focus for anyone looking at the solar statue.

LEDs are light emitting diodes and play a major factor in the functionality of this light driven Solar Powered Flower Sculpture. LEDs come in many shapes and sizes and from LED to LED they vary quite more than most would realize. Obviously, LEDs can vary in shape and size but there is much more variation than that. LEDs offer a variety of different features such as, brightness, color range, drivers, intensity, and some are even waterproof. Hence, there are a multitude of different measurements associated with LED lights. Also, different LEDs are designed for different purposes so with that in mind it makes sense that there are a wide range of LEDs on the market. So when selecting LEDs it is vital to clearly define what they will be used for. This will allow you to narrow such a broad market and focus in on what is the most ideal for the project at hand.

Based on our purposes our Solar Powered Flower Sculpture's design needs, programmable LEDs were required. However, this became a problem when the need of how many LEDs were needed, versus, how many pins were available on the board was considered. This led to a search for LED strips that could be grouped and programmed via one pin per LED strip. This ability became very advantageous to the Solar Powered Flower Sculpture's design needs. Which allowed the Solar Powered Flower Sculpture's design to overcome some of the physical constraints of the project.

Apart from the physical constraints of the Arduino board, the factors that contributed to the design and selection of the LEDs were the intended features of the Solar Powered Flower Sculpture's design. For this Solar Powered Flower Sculpture, the LEDs are designed so that they will tell time, act as a lamp, have a party mode feature that will cycle through colored lights on the sculpture. In order to accommodate these features efficiently, it was included in the design that the LEDs selected would have an RGBW color scheme. This enables the Solar Powered Flower Sculpture to have a single LED that can scroll through multiple colors. This offers a single LED to complete the task as opposed to a needing four for the same job, thus reducing waste.

Table 1: LED Comparisons (By Kelechi Ukachi-Lois)

Type	Working temp(°C)	Power	Color Range	Working Voltage	Water Proof ability
NeoPixel Ring - 24 x 5050 RGBW SK6812	40°C To +85°C	0.2w/LED	RGBW	5VDC	Non-Waterproof
SMD 5050	-20°C to 50°C	0.2w/LED	RGB	12V DC	Non-Waterproof
SMD 3528	-20°C To +50°C	0.13w/LED	RGB	DC 12V	Waterproof
NeoPixel Digital RGBW SK6812	-40°C To +85°C	0.25w/LED	RGBW	5VDC	Weatherproof of casing
Adafruit NeoPixel Digital RGB WS2812B	-25°C To +80°C	0.32w/LED	RGB	5VDC	Weatherproof of casing
5630 SMD	-20°C To +50°C	0.5w/LED	W	12VDC	Non-waterproof

## 2.4.3 Power Supply

The power supply is the heart of the solar statue, serving as the life force of the project. The power supply will be made up of two components: a rechargeable battery source, and solar cells. Electricity from the solar cells will be used to charge the onboard battery. Both sources will provide a supply of voltage that will power the system's many functions over long periods of time.

### 2.4.3a Solar Power Design

The featured power source for the statue are the solar cells. The solar cells will be responsible for powering the statue and charging the battery for use when solar power is not available. Solar power is expected to be the primary source of power when the statue is in use outside where power outlets are not available. This is a key component to the portability feature of the solar statue. The solar cell will be

responsible for converting energy from a light source and convert it to a charge for the battery.

### **2.4.3b PCB Design**

For this Solar Powered Flower Sculpture's printed circuit board, the group decided to put all of the components together on one printed circuit board. By doing this the group will be able to optimize maximum space within the base of the Solar Powered Flower Sculpture. However, there are some downsides to doing so. This includes that the copper wires connecting all of the components on the printed circuit board have the ability to overlap and cause misconnections. Another negative, is the fact if one side of the board fries or overheats, there is a huge chance that the whole board will fry and become inoperable. With these negatives, the group will have to make sure to run multiple breadboard tests to determine if the hand drawn schematic will in fact be effective.

### **2.4.3c Battery**

The batteries will serve as the sole source of power for the Solar Powered Flower Sculpture when solar power is not available (such as when photovoltaic cells have been temporarily depleted and it is overcast or nighttime). The batteries shall get their charge from the photovoltaic cells whenever conditions allow the photovoltaic panels to collect sun rays. The batteries shall have be able to go through multiple charge and depletion cycles before losing functionality. The batteries should be able to store enough power for the entire Solar Powered Flower Sculpture to be in continued use, for an extended amount of time, and without any other needed power source.

### **2.4.4 Wireless Module**

In order for the Android application to communicate with the Solar Powered Flower Sculpture, wireless communication must be possible. There are many different ways of achieving this with the current technology of this era. Standard forms of wireless communications available today include but are not limited to, Wi-Fi 802.11 b/g/n, radio 433MHz, Bluetooth 802.15.1/SIG/PAN, and cellular 3G. Each of these types of wireless modules have their own unique protocol, benefits and tradeoffs. When selecting a wireless module a range of various factors should be considered. The most important thing to consider when selecting a wireless module is the project needs. Do you require online connectivity? How large of a connection range is needed? What are your security concerns? What is your budget? Are there size limitations? It is very likely all these factors and more are necessary when deciding on a wireless module.

Wi-Fi offers the most connectivity of the previously mentioned which is no surprise seeing as how it is used to connect people all over the world via the World Wide

Web. Using a Wi-Fi module for a project would allow for remote access from any Wi-Fi accessible connection. Meaning that it could be accessed from next door or even the other side of the world. If a project requires remote access from maximal distances Wi-Fi may be a plausible module.

Cellular 3G also had great connectivity. The availability of 3G is dependent on cell towers. Depending on where the project is located and other needs, a 3G module may be the best bet. In areas where Wi-Fi is not available, cell towers may be able to provide internet connection. However, keep in mind this form of communication is much slower and if possible, Wi-Fi may be a better option.

Radio 433MHz offers great range for a small scale connectivity. Its price is also low cost. In some cases these radio modules can offer up to 200 meters of connectability. The standard Bluetooth module pales in comparison and only offers up to 10 meters of connectability. This means that a radio module 2000% more effectively than Bluetooth modules when looking solely from a distance perspective.

Bluetooth also has its benefits. Especially, its convenience with android application and microcontroller integration. Android phones are equipped with Bluetooth making it much easier to communicate via Bluetooth module versus a radio module. Bluetooth is also much better known to the general public as opposed to radio modules. Therefore, Bluetooth modules are much more user friendly. At the user level Bluetooth is much easier, simpler, and convenient which makes it more ideal when keeping general users in mind.

Table 2: Wireless Data Comparison (By Kelechi Ukachi-Lois)

	<b>WIFI 802.11g</b>	<b>Cellular 3G</b>	<b>Radio</b>	<b>Bluetooth v4.0</b>
<b>Operating Frequencies</b>	2.4 GHz	850MHz GSM	433 MHZ	2.4 GHz
<b>Data Rate</b>	54 Mbps	9.6 kbps	4800 bps	2.1 Mbps
<b>Maximum Range</b>	Global	Global	200 meters	10 meters
<b>Cost*</b>	\$30	\$80	\$7	\$4

\*Cost may vary

For this Solar Powered Flower Sculpture a Bluetooth wireless module was selected. A few of the major factors that went into account when selecting this module were, project requirements, cost, ease of integration, along with experience, time, ability and many others. The wireless module selected works

well with the selected microcontroller and android application. This Bluetooth module allows the Android application to transmit data. Once the data is received the Bluetooth module then relays the information to the micro controller. This is what allows the Android application to communicate with the microcontroller. An analogy would be to think of the wireless Bluetooth module as a courier that takes and delivers message from the Android application and microcontroller which effectively allows the two to communicate.

## **2.4.5 Microcontroller**

The Microcontroller shall have at least 20 pins or be compatible with a module that expands the number of available pins to 20. The pins on the boards shall provide PWM output. The board shall have at least 12kB of flash memory and at least 1 kB of RAM. For programming the board must have high level programming language capabilities. All of the components for the solar statue will be controlled by the microcontroller. Being the brain of the project, picking the right microcontroller is essential to the success of the project. Being able to output a standard voltage between the ranges of 3V and 5V is essential as most modules and components require a voltage within this range to operate. Not having to shift voltage up and down will help to keep the design simpler, with less potential points of error. Having both digital and analog pins will allow for the microcontroller to take input from multiple sources and send the output to the various components. Having extra pins available will allow for the addition of additional features as the project progresses through development.

## **2.4.6 Application**

The mobile application is a necessary component of the project. It easily solves the issue of how users would be able to modify key parts of the sculpture's LED display, and the wide array of options that can be performed with the app exemplifies both the different situations the sculpture can be used in and the variety of purposes it can serve. Having a compatible app creates a level of interaction with the user that parallels with one of our group's main requirements for the project: creating a personal experience unique to the user.

Having an application that controls the Solar Powered Flower Sculpture requires an up-to-date Android phone; the phone should be running a version more recent or equal to Android 5.0 - "Lollipop" (which is the oldest supported Android version as of November 2014), though backwards compatibility should be available through devices running Gingerbread. If the Android phone is on in which the app can be successfully downloaded to, then it is also required that it offers Bluetooth capabilities for communication with the Arduino microcontroller. The Android Studio code used to create this application should be concisely written, and with strong performance in mind to ensure as minimal delays as possible between user-submitted app requests and the sculpture display taking appropriate effect.

## 2.4.6a Control Interface

As mentioned in sections 2.4.3.1 and 2.4.3.2, the project's smart phone application will serve the functional purpose of offering four different modes for the sculpture's LED display to take effect from. The first "clock mode" sets the time that will be displayed by two LED petals representing an hour and minute hand, and also can change the color and brightness of these LED lights. The second "lamp mode" illuminates all of the sculpture's LEDs and grants the user the ability to adjust their color and brightness. The third "party mode" lets the user select from a list of created light patterns that the sculpture's display will cycle through (as well as having the user adjust the brightness). The fourth "power mode" toggles the system on or off.

The main requirement of the application's control interface is to have it be completely user-friendly. The goal is to create as little confusion as possible for whenever the user wishes to interact with the solar sculpture. Our group believes a simple design for the app interface reflects the clear-minded design of the flower sculpture itself, and helps users to have a more positive experience with it. This is why the app only ever will bring the user to one of four different screens, one of which is the home screen. The other three screens, which open when the clock, lamp, and party modes are selected, all include a back icon that returns to the home screen. All other interactive functions are enabled with the appearance of pop-up menus when other icons are selected. This choice was clear, as it gives the user the ability to still see other untouched options after an icon has been selected, and it allows the user to easily exit out of the option by selecting anywhere outside of the pop-up menu. Never will the user have to type in numbers for what time they wish to enter, or the name of the color they want the LED to reflect. Rather, all options will be in list form for the user to select from, so that a case is avoided where the user wishes to enter an option not supported by the app. This also gives the opportunity to observe all available options for respective mode functionalities. The app interface is presented as a comforting, risk-free environment, any change made to the system can easily be kept or undone depending on the desires of the user.

## 2.4.7 Texas Instruments TLC5940 LED Driver

The Arduino Nano / Atmega328 microchip only has six pulse width modulation outputs. At a bare minimum we needed at least twelve pulse width modulation input / output pins to get our Solar Powered Flower Sculpture's NeoPixel Digital LED strips to work. Our solution was to look for a LED driver. The LED driver was one of the more straight forward components we needed to include in our design. Put simply, we just needed something that would expand the amount of pulse width modulation input / output pins so we could connect all twelve individual LED strips to the circuit. After doing some research, the group decides to go with Texas Instruments TLC5940 LED driver, which provided the group with the simplest



solution to this problem. It was compatible with the Arduino Nano and did what we needed it to do without adding any unwanted complexity to the overall design of the project. The LED driver shall have 16 pulse width modulation input / output channels with a twelve bit duty cycle. It shall be daisy chainable and have a current limit control of six bits.

## **3.0 Standards and Design Constraints**

The world we live in is finite and imperfect. Therefore, regardless of what is designed on this earth, there will be limitations and matters that are beyond our control. These limitations and matters which are beyond our control are mostly responsible for something that is very prevalent in engineering, tradeoffs. Tradeoffs are a major part in any engineering project and must be taken into consideration in the design process. What needs to be kept in mind is that adding to a desirable feature can in turn, negatively affect another desirable feature. This means a choice must be made and there will be limitations that force the developer to place constraints on a system in order to keep balance between one desirable feature and another. However, constraints do not only come from physical limitations, constraints appear from multiple sources and for multiple different reasons. Constraints may be produced because of economic reasons, time, environmental reasons, social and political reasons, ethical reasons, health and safety reasons, manufacturability and sustainability, among other things.

For the Solar Powered Flower Sculpture, a great deal of planning went into its design. The Solar Powered Flower Sculpture has many systems working together. Whenever there are multiple hardware components working together, tradeoffs and constraints appear. This Solar Powered Flower Sculpture has physical components, electrical features and physical display output necessities. With all of these working parts many different constraints and tradeoffs arise.

### **3.1 Economic**

Economic constraints are restrictions placed on the design due to our budget. Since this is not a sponsored project the budget constraints played a significant factor in our design. A limit of \$400.00 was set for the total prototyping budget of the project. This will limit the amount of features that were considered for the project as the amount of resources for features were severely limited. This also limited the size of the statue that will be developed, as a smaller statue requires less raw material to construct. This limit was chosen after the group cataloged the all the parts believed to be required to complete the project and their costs were summed up.

### **3.2 Time**

Time constraints are constraints due to design schedule, development schedule, production schedule and delivery schedule. Design schedule is a major factor in this project as most of senior design one is spent in project planning. All project planning had to be done by the end of the semester. The rest of the time constraints will be implemented in senior design two. Development schedule is made up of design detailing and compliance testing. Until the parts and material are ordered and shipped design detailing and compliance tests are severely

limited. The production schedule for this project won't include packing and transport as the prototype will be demonstrated on property. Only manufacture and assembly time will be a part of the production schedule. Delivery schedule will not be considered, since the prototype will be demonstrated on property.

### **3.3 Environmental**

Environmental constraints are the effect that the statute or its components may have on the environment. The electrical components specifically have six substances that fall under the Restrictions of Hazardous Substances (RoHS). These substances are Polybrominated Diphenyl Ethers < 1000 ppm, Lead < 1000 ppm, Polybrominated Biphenyls < 1000 ppm, Mercury < 100 ppm, Hexavalent chromium < 100 ppm, and Cadmium < 100 ppm.

### **3.4 Social and Political**

Social constraints are the constraints that the society place on the products they buy. The customers in our target consumer will expect a product look and work a certain way. The main social constraint we considered was portability. We wanted a device that consumers could take with them on outdoor excursions. The look of the final project was also affected by social constraints. Consumers not only purchase products for their function but also for the aesthetic qualities as well. The flower appearance was a result of this constraint.

Political constraints affect the design due to political reasons. The product will not discriminate in any way. It will not be used to push any political agendas. In summary there are no political constraints that will restrict this product.

### **3.5 Ethical**

Ethical constraints are the constraints that result from ethical concerns. This product will perform its functions as advertised and will have a reasonable expected operational lifetime. The material and parts used in construction will not be the just the cheapest available but will be the parts that are necessary to meet the requirements at a reasonable cost. The product will be built in such a manner to withstand normal use without excessive wear and tear leading to a premature breakdown of the product. The product will be safe for users to operate and come into contact with. There won't be any harmful emissions or any material that can affect the user from contact with the product during normal operation of the product. The product doesn't require the user to enter any personal information, so misuse of that isn't a concern.

## 3.6 Health and Safety

When designing any engineering project health and safety must be taken into account. Especially when a project involves components or features that can potentially become hazardous to people and or the environment. Moving parts, heat, choking hazards, electrical components, heavy-weight, toxic substances, toxic gas due to fire, are just a small list of potential hazards that can lead to injury, sickness, loss of life, and even environmental catastrophe. The health and safety of people and the environment are dear and must be protected as much as humanly possible. Failure to comply with rules and regulations can not only lead to tragedy, but can also lead to criminal negligence and other criminal charges. Hence, it is just as imperative to keep health and safety in mind for yourself, as it is for others and the environment.

For the Solar Powered Flower Sculpture, there are many health and safety factors to be kept in mind. Of those, the key factors to keep in mind are electrical factors, and heavyweight. If some health hazard is to occur, it will most likely come from one to these to attributes. Although this Solar Powered Flower Sculpture is designed to be within three cubic feet and less than 40 pounds, heavy weight can still be a valid health and safety concern. Because this sculpture is designed to be small enough to sit on a table its weight can be a major concern in the event the sculpture falls off of the table. In order to combat this safety concern the Solar Powered Flower Sculpture will be equipped with rubber legs. Similar to those of designed for furniture. These rubber legs will increase friction and generate enough force to keep the sculpture from sliding in most cases. This safety feature will prevent many falls and potentially avert possible damage and injury due to falling.

Another major safety concern is due to the presence of electricity. A term that most if not all people are familiar with is electrocution. Fortunately, this generally only happens at high voltage but is still possible at low voltage because it is the current which is deadly. In addition, short circuits pose the greatest risk electrocution but can also cause fire. Even without a short circuit, overheating is also a possible danger. Whenever electrical circuits are involved, caution must be taken in order to prevent dangers. Although electrocution is an unlikely threat in this Solar Powered Flower Sculpture, electrical burns are still possible at low voltages and electricity is always something that is to be respected.

In order to combat some of the dangers of electricity, the design of this Solar Powered Flower Sculpture has taken preventative measures. The first and most obvious precaution is waterproofing. It is well known knowledge even among the general public that water can cause short circuits. Considering that the Solar Powered Flower Sculpture is designed to work indoors as well as outdoors, it is likely that at some point it will be exposed to rain. This is why a waterproof wood stain will be used to cover the exterior of the wooden paneling that makes up the base of the solar sculpture. In addition, the placement of the Arduino board and

wires within the base will be especially considered with protection from water leakage in mind. Any possible openings will be tightly sealed, including the locations at which the 3D-printed petals meet with the structure's base, and where the solar panels are attached. The structure is not intended or prepared for extreme water-damaging situations, such as complete submergence, but for resistance from rainfall (or possibly minor spills) the structure should prove functionally efficient.

Another electrical liability comes from battery charging. The Solar Powered Flower Sculpture uses photovoltaic cells to charge a rechargeable battery. However, if the photovoltaic cells are connected directly to the battery problems can occur. There are three main cases in which problems are most likely to arise. The first is when the photovoltaic cells have no sunlight. This situation will cause the battery to power the photovoltaic cells which is the opposite of the desired effect and will drain the battery.

Although inconvenient, this does not pose much of a health and safety hazard with the Solar Powered Flower Sculpture. On the contrary, the next two problems can likely pose anywhere from minor to serious injuries and in the rarest of cases even death.

Of these two hazards, one problem is overcharging. Due to the chemical properties of batteries, excessive heat is a major problem. Overcharging is a serious problem because the batteries can potentially overheat. The excess of heat can affect the chemicals reactions taking place in the battery. Once this occurs, the resulting effects can ultimately lead to an electrical fire. Keep in mind electrical fires are especially dangerous because if water is used to extinguish the fire, short circuits will occur. The dangers of short circuits have been previously stated above.

The second of these two hazards is charging too rapidly. Today, we have fast charging smartphone devices however, these are different. The fast chargers of today have special circuitry which allows for fast charging without the overheating issues. But when photovoltaic cells are connected directly to the rechargeable battery problems arise. If the rechargeable battery is connected directly to then it is possible that the battery will be flooded with more current and voltage than it can handle. This will lead to the rechargeable battery overheating. Similar to the case of overcharging, the excess of heat can affect the chemicals reactions taking place in the battery. Once this occurs, the resulting effects can ultimately lead to an electrical fire. Keep in mind electrical fires are especially dangerous because if water is used to extinguish the fire, short circuits will occur. The dangers of short circuits have been previously stated above. Therefore, connecting the battery directly to the photovoltaic cells is a bad idea and potentially hazardous to the health any affected.

In order to prevent the safety concerns of overcharging and overly rapid charging, the circuits of the Solar Powered Flower Sculpture been designed to handle these

potential charging issues. The Solar Powered Flower Sculpture does not connect the photovoltaic cells directly to the rechargeable batteries. Instead the Solar Powered Flower Sculpture uses a charge controller in order to control the charging to prevent the three issues stated previously. The charge controller works by controlling whether or not charge will flow based on state of the batteries.

### **3.7 Manufacturability and Sustainability**

Manufacturability constraints are restrictions that may affect how the product is manufactured. The solar statue is intended to be operated outdoors, therefore it will potentially be exposed to the elements such as humidity, high heat, cold, and salt mist. Standard 9223:2012 on the NSSN webpage may affect the manufacturability of this product. This standard establishes a system for classifying the corrosive nature of atmospheric environments. It defines corrosive nature categories, gives dose-response functions for normative estimates of the corrosive nature category, and makes it possible to make an informative estimation of the corrosive nature category of a product and its components. A proper coating will have to be used on the product to make it resistant to the elements. One of the requirements considered for the LEDs that will be used in the product was water resistance, so the manufacturability constraint is not applicable to them. Properly protecting the any external wiring will have to be considered for compliance with this constraint.

Sustainability constraints are restriction that affect the overall operational lifetime of the product. There won't be a manufacturer's warranty on this product but the product should perform without any issues for at least a year. Because of the quality of the components and materials used in the construction of the product, the group believes that an operations lifetime of several years is reasonable for this product. Battery and LED failure due to a large amount of on and off cycling resulting in the need for replacement of these parts are the only anticipated repairs that may need to be made over the product's lifetime. Battery and LED failure due to excessive on and off cycling is a limitation of current technology and beyond the capability of this group to alter.

### **3.8 Android Standards**

Our project requires using an Android-based mobile application for implementing the interactivity of the solar sculpture and its LED display. Developing with Android means following several Android standards in order to make use of our created software. These standards relate primarily to the design of the user display for the app, but also to its overall performance and functionality.

The purpose of the Android interface standards are mainly to benefit the user. Having a more presentable, understandable working app helps the user to adapt to using the app at a faster and more enjoyable rate. This includes styling the app to have a standard color scheme, with primary and secondary colors that can be

used to differentiate components, such as different menus or option icons. Differentiating between components of the app also is expressed through the design of certain icons. The Android standards state that no icon used in the app should be used for conveying multiple purposes. The arrow icon used in our app as a back button should only send the user back to the previous (i.e. home) menu screen, and not also be used to send the user to different screens entirely. As our app involves several menus and option icons, it is important to present these as appearing non-cluttered whenever they pop up. Android standards require the use of whitespace whenever interactive regions are presented. Creating too many of these regions, especially without appropriate whitespace, creates a confusing, unprofessional looking layout. One way the standards dictate pop-up menus to appear presentably is by having the new regions appear as cards. These cards would break an edge that was established by the menu as it was used before the user clicked on an icon. Standards define similar guidelines for toolbars, and list dimensions to be used for any titles or text that would be featured on them. Conversely, some standards, such as those detailing app notifications, would not apply to our project's app, as we are not planning on implementing notifications in the design.

How our application performs is also dependent on Android standards. The app should be designed and tested to ensure high performance, but even in cases where performance is not optimal, the app should still inform the user of such. As an example, if an application function takes longer than some period of time to load (a few seconds), then the app should present a message or icon that lets the user know loading is taking place; perhaps even with some form of loading bar depending on how long the task is. High quality text, images, audio, and transitions (if any) should also be present, according to the standards. This would mean no blur or visible "jaggies" in any image used, or unnatural distortion in audio that plays when an icon is selected. For text, this implies words should be written in a non-distracting font, centered when need to be, and should not be cut off anywhere. Obviously, there should be no spelling errors either.

Lastly, some Android standards are in place to describe the functionality of our app's interface. Our application is intended only to be used with our solar sculpture, and is not reliant on any other application installed on the user's phone (although it is dependent on the phone's Bluetooth capabilities). Regardless, standards require our application functionality to anticipate different types of Android phones that may be making use of it. This means the app should correctly orient itself if the phone is physically tilted sideways or upside down. Also, due in part to the simplistic, menu design of our project's app, progress should be resumed depending on several possible actions the user may take. In other words, if the user goes to a certain menu screen, or modifies a component on an app mode (changing the time, selecting an LED color, altering brightness, etc.), these conditions should remain the same if the user were to switch to using a different application on their phone, or put their phone into sleep mode by turning their phone off (not relating to a system reboot). The Android standards for functionality dictate that the latest changes made to the application, including which screen the

user was last left on, should be saved for most actions the user would take on their mobile phone.

### **3.9 Bluetooth Standards**

The Solar Powered Flower Sculpture requires use of a method of wireless communication. The wireless form of communication selected is Bluetooth. Wireless communication is necessary to link the Android-based mobile application to the microcontroller of the Solar Powered Flower Sculpture. Bluetooth will be connected using the built-in Bluetooth of the smartphone along with a Bluetooth module that will be directly connected to the microcontroller. Developing the Solar Powered Flower Sculpture with Bluetooth wireless connections means that it is inevitable that several Bluetooth standards will become applicable to the Solar Powered Flower Sculpture.

These standards relate primarily to the use of the Bluetooth devices to perform with the Android application via smartphone and microcontroller via Bluetooth module. Also, Bluetooth standards are to be considered with the overall performance and functionality of the parts involved and the Solar Powered Flower Sculpture as a whole as it pertains to the wireless Bluetooth communication.

Bluetooth uses specific communication protocol to handle its wireless communication. This Bluetooth protocol is standardized and has specific specifications that are required for a communication protocol to be considered Bluetooth. In a broad sense, Bluetooth simply sends and receives data through a 2.4 gigahertz wireless link. Bluetooth devices use standardized protocols to communicate and have a general process that they use to achieve the protocol standards. Bluetooth has core system protocols, these protocols are “radio (RF) protocol, link control (LC) protocol, link manager (LM) protocol and logical link control and adaptation protocol (L2CAP), all of which are fully defined in the Bluetooth specification.” - (<https://www.bluetooth.com/specifications/bluetooth-core-specification>).

Of the Bluetooth protocol, the radio protocol, the link control protocol, and the link manager protocol are the lowest of the system layers. These three layers play a special role and are generally grouped together. When these three system layers are linked together, they form a subsystem which is known as the Bluetooth controller. This implementation is used frequently. This implementation uses an optional standard interface. The interface is known as, Host to Controller Interface (HCI). This interface enables two-way communication with the remaining Bluetooth system which is known as the Bluetooth host.

Although Bluetooth has core system architecture, the configurations still have a degree of variance. As discussed later on in the text, Bluetooth has many different versions. In these different versions new features were implemented which led to different implementations of architecture. Two of the most dominating



implementations of the core system architecture specification come from Bluetooth version 2.0/2.1 and Bluetooth version 4.0/4.1/4.2. Bluetooth version 2.0/2.1 uses Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR). Bluetooth version 4.0/4.1/4.2 uses Bluetooth with low energy (LE). Each of these versions have different use cases and implementations. Also these different implementations use different chipsets to meet hardware requirements. Adding on, there are also Dual-mode chipsets available that are able to use both implementations.

These different variations of Bluetooth versions have different configurations for their primary controllers. Bluetooth version 2.0/2.1 uses Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) and the Bluetooth controller includes, radio, baseband, Link Manager and optionally HCI. Bluetooth version 4.0/4.1/4.2 uses Bluetooth with low energy (LE) and the Bluetooth controller includes, LE controller including the LE PHY, Link Layer and optionally HCI. In Dual-mode chipset, the Bluetooth controller Combined BR/EDR controller and LE controller, with one Bluetooth device address shared by the combined controller.

Despite the different Bluetooth versions and core system architecture, Bluetooth systems allow for interoperability between systems, which is enabled by the Bluetooth specification. The interoperability between the different systems is achieved by defining the protocol messages that are exchanged between equivalent layers. Interoperability between independent Bluetooth subsystems are also supported. This is made possible by defining the common interface between Bluetooth controllers and Bluetooth hosts.

The different compatibility between the Bluetooth has core system architecture is a significant ability of the Bluetooth architecture. This compatibility allows for many different versions of Bluetooth such as, Bluetooth version 2.0/2.1 Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR), and Bluetooth version 4.0/4.1/4.2 Bluetooth with low energy (LE) to successfully communicate with each other. It also supports the communication of independent Bluetooth subsystems. These features allow Bluetooth to be extremely versatile and communicate regardless of different features, different primary controller configurations, and different core system architecture. This makes it possible to communicate with a wide range of devices with different intended to purposes to still communicate via Bluetooth protocol.

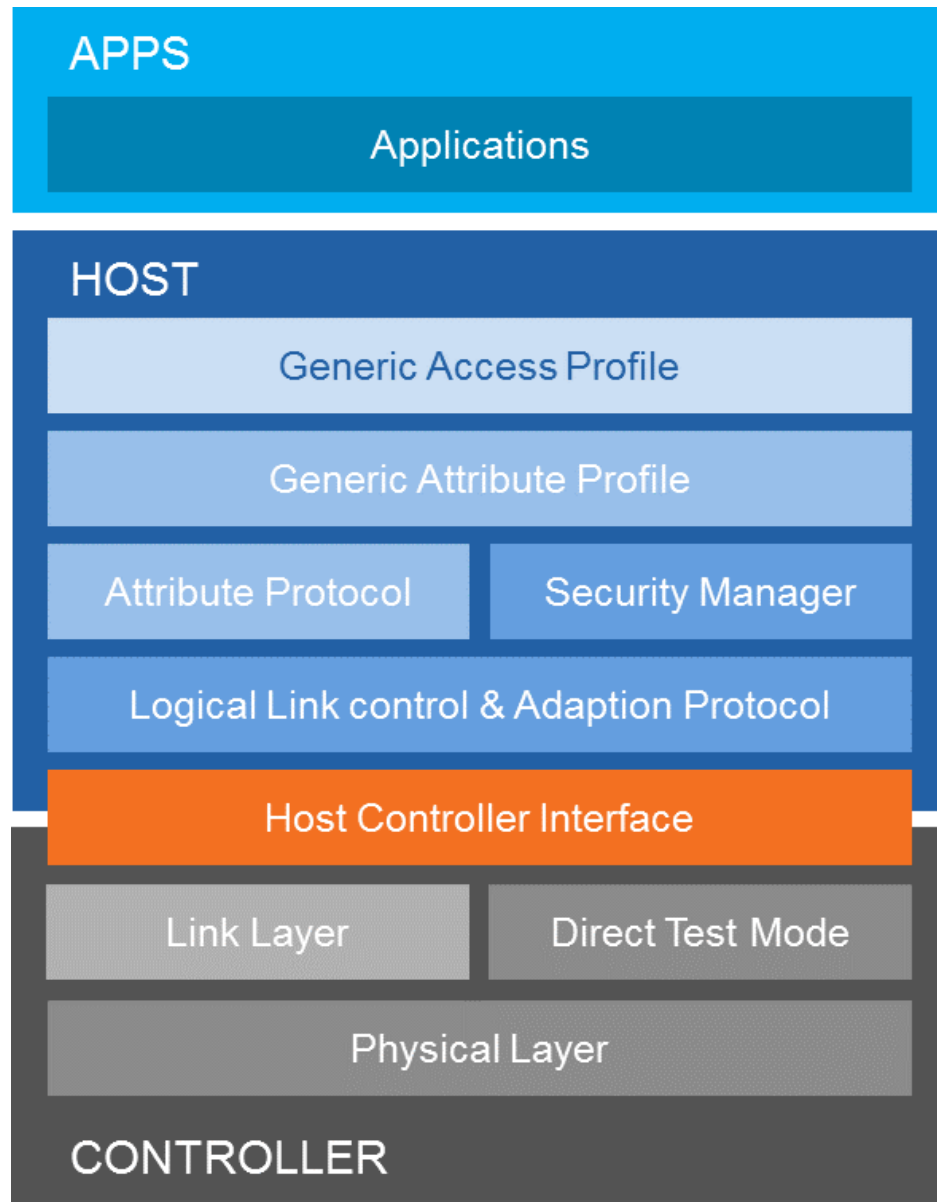


Figure 4: Bluetooth Standards [24]

Image from: <https://learn.sparkfun.com/tutorials/bluetooth-basics>

Above each a diagram visually displaying the layers and sublayers of the Bluetooth communication stack protocol. Each layer in the stack has its own specific responsibilities. Each part the stack layer has a specific job to accomplish and has its own specific ability which is necessary to complete its task. The Bluetooth stack protocol includes these layers:

- Physical (PHY) layer
- Link Layer
- Direct Test Mode
- Host to Controller Interface (HCI)
- Logical Link Control and Adaptation Protocol (L2CAP) Layer
- Attribute Protocol (ATT)
- Security Manager
- Generic Attribute Profile (GATT)
- Generic Access Profile (GAP)

The Physical (PHY) layer is as it sounds, the Physical (PHY) layer involves the physical basic networking hardware. The Physical (PHY) layer is the fundamental layer to the logical data structures and facilitates the logical higher level functions in the network. The Physical (PHY) layer is the means for transmitting the raw bits, these bit streams are usually grouped into code words or symbols and converted to a physical signal that is transmitted over a hardware transmission medium. Bluetooth uses 2.4 GHz radio with Bluetooth communication channels for controlling the transmission and receiving of bits in the Physical (PHY) layer. Adding details, Bluetooth version 2.0/2.1 Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) provides more channels with narrower bandwidth in the Physical (PHY) layer. Bluetooth version 4.0/4.1/4.2 Bluetooth with low energy (LE) uses fewer channels but broader bandwidth.

In Bluetooth stack protocol, the Link Layer and Direct Test Mode are on the same level because, they both have responsibility to work in-between both, the Physical (PHY) layer and the Host to Controller Interface (HCI). This layer supports the functional and procedural means of transferring data. The purpose of this layer is handle both the detection and possibly the correction of errors that may occur in the Physical (PHY) layer. The link layer in the Bluetooth protocol stack is responsible for and defines packet structure and channels, is responsible for the discovery and connection procedure, and, is responsible for and defines how to send and receive data.

The Host to Controller Interface (HCI) is an optional standard in the Bluetooth stack protocol. This standard is useful but not necessarily an imperative standard. However, the Host to Controller Interface (HCI) is extremely useful and as a result the Host to Controller Interface (HCI) is commonly used even though the Host to Controller Interface (HCI) is optional and not mandatory. Host to Controller Interface (HCI) is when the bottom three layers are grouped together to create Optional standard interface known as the Bluetooth controller subsystem.

The next layer is the Logical Link Control and Adaptation Protocol (L2CAP) Layer. The Logical Link Control and Adaptation Protocol (L2CAP) Layer is a packet-based protocol that transmits to the Host to Controller Interface (HCI). However, the Host to Controller Interface (HCI) is an optional standard interface which means it is not always implemented. In the instances where the Host to Controller Interface (HCI), the optional standard interface has not been implemented, the Logical Link Control

and Adaptation Protocol (L2CAP) is able to transmit packets directly to the Link Manager in a hostless system. The Logical Link Control and Adaptation Protocol (L2CAP) Layer also supports higher-level protocol multiplexing, packet segmentation and reassembly, and the passing on of quality of service information to the higher layers.

The following layer has two distinct features and functionality which are the Attribute Protocol (ATT) and the Security Manager. This layer is the layer which handles communication between networks which explains why the Security Manager is part of this layer. Once a connection is established, the Attribute Protocol (ATT) is what defines the client/server protocol for data exchange. The Security Manager defines the protocol and behavior that manages pairing integrity, authentication and encryption between Bluetooth devices. The Security Manager also goes a step further and provides access to a toolbox of security functions. The toolbox of security functions provided by the Security Manager are used by other components to support the level of security needed by the wide range of applications. The Attribute Protocol (ATT) uses Generic Attribute Profile (GATT) to group attributes into meaningful services. The Attribute Protocol (ATT) is mainly used in Bluetooth version 4.0/4.1/4.2 Bluetooth with low energy (LE) implementations and occasionally used in Bluetooth version 2.0/2.1 Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) implementations.

The next layer is the Generic Attribute Profile (GATT). The Generic Attribute Profile (GATT) works in conjunction with the Attribute Protocol (ATT). The Generic Attribute Profile (GATT) groups together attributes in a specific way based on its functionality. The Generic Attribute Profile (GATT) groups together attributes by grouping services that encapsulate the behavior of part of a device and then describes a use case, roles and general behaviors. The Generic Attribute Profile (GATT) can be thought of as an organizer. The Generic Attribute Profile (GATT) “defines procedures and formats of services and their characteristics, including discovering, reading, writing, notifying and indicating characteristics, as well as configuring the broadcast of characteristics.” - <https://www.bluetooth.com/specifications/bluetooth-core-specification>. The Generic Attribute Profile (GATT) is used only in Bluetooth version 4.0/4.1/4.2 Bluetooth with low energy (LE) implementations.

The last layer in the Bluetooth Host of the Bluetooth communication stack protocol is the Generic Access Profile (GAP). Generic Access Profile (GAP) works in conjunction with the Generic Attribute Profile (GATT) in Bluetooth version 4.0/4.1/4.2 Bluetooth with low energy (LE) implementations in order to “define the procedures and roles related to the discovery of Bluetooth devices and sharing information, and link management aspects of connecting to Bluetooth devices.” - <https://www.bluetooth.com/specifications/bluetooth-core-specification>. The Generic Access Profile (GAP) is the layer that interacts with the application layer from the Bluetooth Host of the Bluetooth communication stack protocol. The Bluetooth communication stack protocol ends at the application layer

As stated previously, Bluetooth protocol operates in the unlicensed ISM frequency band at 2.4GHz. Through the Bluetooth communication stack protocol, Bluetooth devices are able to connect and communicate. Once the Bluetooth devices are connected they are considered to be on a Bluetooth network. These Bluetooth networks are referred to as piconets. Bluetooth networks use the master/slave model. It is this master/slave model that is used to regulate when and where devices can transmit data. In Bluetooth's master/slave model, a single master Bluetooth device is able to be connected to up to seven other slave devices. This means the maximum number of devices a single Bluetooth network can support is eight devices in total. Also, Bluetooth network slave devices can only be connected to a single master. Below is a visual representation of how the Bluetooth devices are connected on a Bluetooth network.

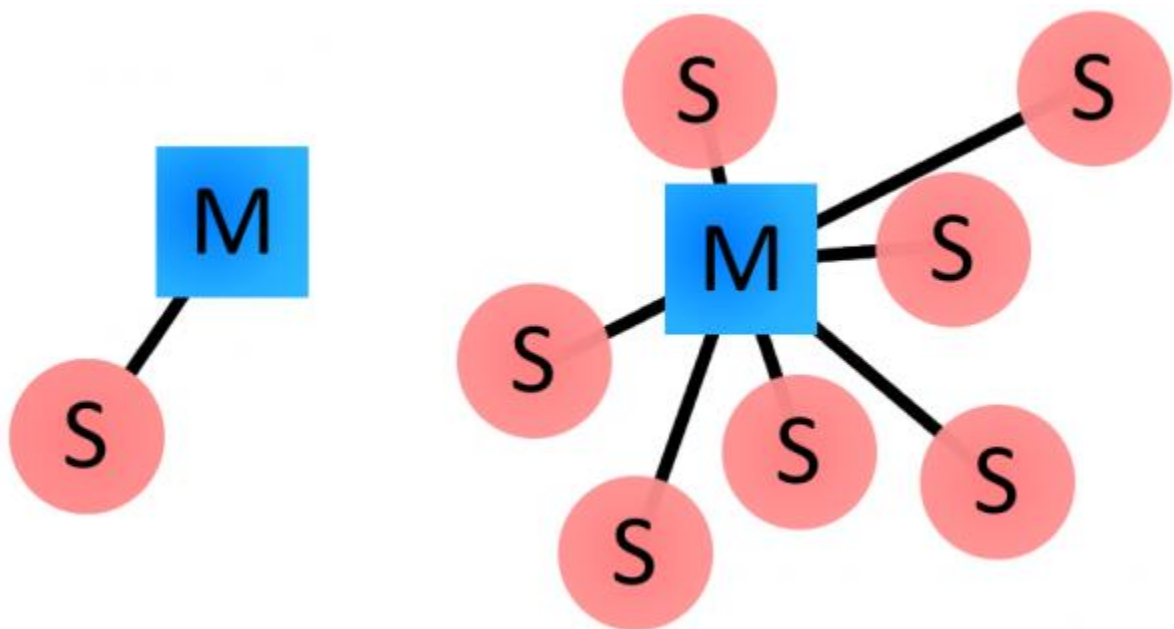


Figure 5: *Examples of Bluetooth master/slave piconet topologies.*

Image from: <https://learn.sparkfun.com/tutorials/bluetooth-basics>

On the Bluetooth network, masters and slaves have different levels of control when it comes to transmitting and receiving data. As expected the master controls the flow of information that is being transmitted. Slaves that are on the Bluetooth network can communicate with the master but cannot communicate with other slaves. Slaves on the Bluetooth network are only allowed to communicate by transmitting and receiving data with the master only and no other slaves. The master on the Bluetooth network can communicate with all of the slaves on the Bluetooth network which means the master can communicate with all the other devices on the network. Also, the master is allowed to request data from all the slaves on the network. Due to the restrictions placed on the slaves and the fact that slaves can only communicate with the master means that all communication

on the Bluetooth network goes through the master device on the Bluetooth network.

Bluetooth standards have specific protocol for Bluetooth addresses and names. Every single Bluetooth device has a unique 48-bit address. These address are typically abbreviated with “BD\_ADDR”. The Bluetooth device’s unique 48-bit address is typically represented in the form of a 12-digit hexadecimal value. The unique 48-bit address is split into two logical sections. The upper 24-bit and more important half of the unique 48-bit address is used as the organization unique identifier (OUI). The organization unique identifier (OUI) is used to identify the manufacturer. The lower 24-bits are the more unique part of the address and is used to identify the specific Bluetooth device. Together these two logical sections are used to create the unique 48-bit address for the specific Bluetooth device.

On most Bluetooth devices, this address should be visible for the most part. In the image below the unique 48-bit address is represented as a 12-digit hexadecimal value. Next to “MAC NO.” the device is displaying its unique 48-bit address. The “000666” represents the organization unique identifier (OUI). The “422152” represents the unique Bluetooth address for this manufacturer.



Figure 6: *Example of Bluetooth unique 48-bit address.*

Image from: <https://learn.sparkfun.com/tutorials/bluetooth-basics>

Bluetooth device unique 48-bit Bluetooth addresses are not very user friendly. However, Bluetooth also has the ability to give user friendly names in place of the unique 48-bit address. The user friendly are generally used in place of the unique 48-bit address to help identify the device. The Bluetooth device name protocols are much less stringent than the unique 48-bit address. Bluetooth device names can be selected with a much higher degree of freedom. Two different Bluetooth devices can share the same Bluetooth device name which means that unlike the unique 48-bit Bluetooth addresses, Bluetooth device names do not have to be unique. However, because the Bluetooth device names are not unique,

sometimes the unique digits of the address might be included to differentiate the Bluetooth devices. Also, Bluetooth device names can be up to 248 bytes long.

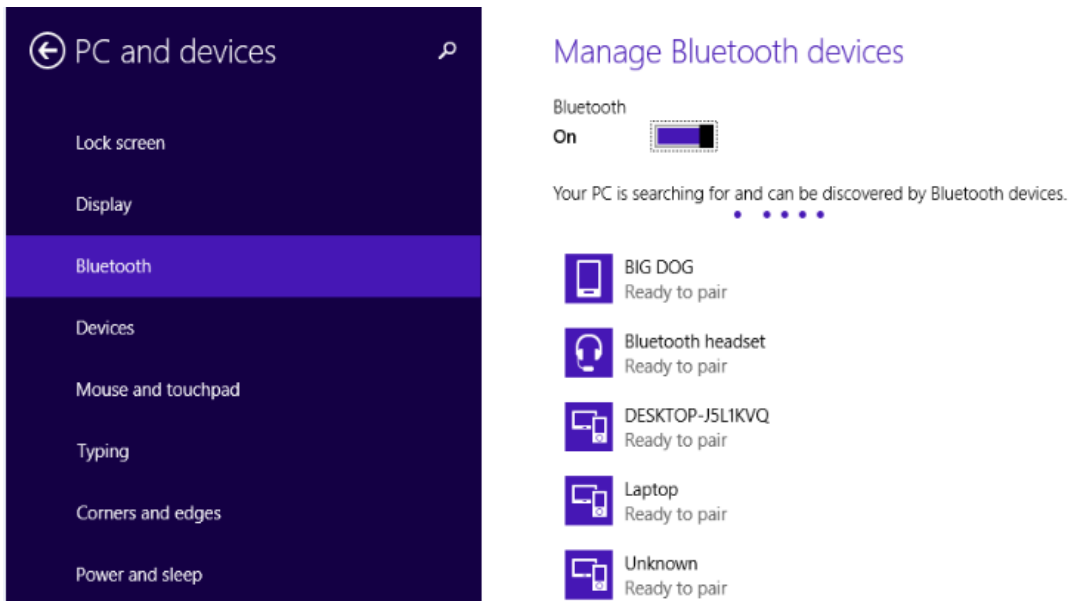


Figure 7: Example of Bluetooth Device Names

Another piece of Bluetooth standards are the Bluetooth profiles. The Bluetooth profiles are the protocols which build upon the more basic Bluetooth standards. Bluetooth profiles are higher on the pyramid of protocols since they are in addition and build upon the Bluetooth standards. Bluetooth profiles are used to reduce ambiguity and more clearly define the type of data a Bluetooth module is transmitting. The standards lower than Bluetooth profiles on the pyramid of protocols tend to define how the Bluetooth technology works, whereas, Bluetooth profiles tend to define how the Bluetooth technology is used in perspective with the of Bluetooth technology as a whole.

A Bluetooth profile is used to support the need of the intended use of the Bluetooth device. Bluetooth technology is used in many different ways and for many different purposes. Meaning that Bluetooth technology will be used for different applications. Bluetooth profiles are designed to accommodate the multitude of Bluetooth applications in order to optimize efficiency. Bluetooth can be used for a hands free headset, which uses the Bluetooth headset profile (HSP). Bluetooth technology can also be used for a Nintendo Wii controller and would use the human interface device (HID) profile. Although Bluetooth profiles are designed cater to the needs of the application, there is a drawback. The drawback of Bluetooth profiles is that they create issues for incompatibility which is known to cause trouble. Bluetooth profiles can make it impossible for devices to be compatible if both Bluetooth devices do not support the same Bluetooth profile.

Bluetooth profiles are a major part of the Bluetooth standard since if two Bluetooth devices do not support the same profiles then the two Bluetooth devices will be viewed as not compatible. Bluetooth profiles are numerous and there are more than thirty different Bluetooth profiles today. Bluetooth profiles each have their own specialized types of applications that the Bluetooth profiles are designed to run. Some of the Bluetooth profiles are more common than others. In no particular order here are a list of a few common Bluetooth profiles of today.

- Serial Port Profile (SPP)
- Human interface Device (HID)
- Hands Free Profile (HFP)
- Headset Profile (HSP)
- Advanced Audio Distribution Profile (A2DP)
- A/V Remote Control Profile (AVRCP)

The Serial Port Profile (SPP) is a profile designed for a serial communication and may replace interface such as, Recommended Standard 232 (RS-232) or universal asynchronous receiver/transmitter (UART). The Serial Port Profile (SPP) Bluetooth profile is optimal for sending bursts of data between two Bluetooth devices. The Serial Port Profile (SPP) is one of the more fundamental, barebones type of Bluetooth profile since sending data is such an integral part of Bluetooth device communication. The Serial Port Profile (SPP) enables each of the connected devices to communicate as if wires were connected. Of course, The Serial Port Profile (SPP) is wireless and grants the ability to transmit and receive data from greater distances without the use of bulky and messy hardware known as wires.



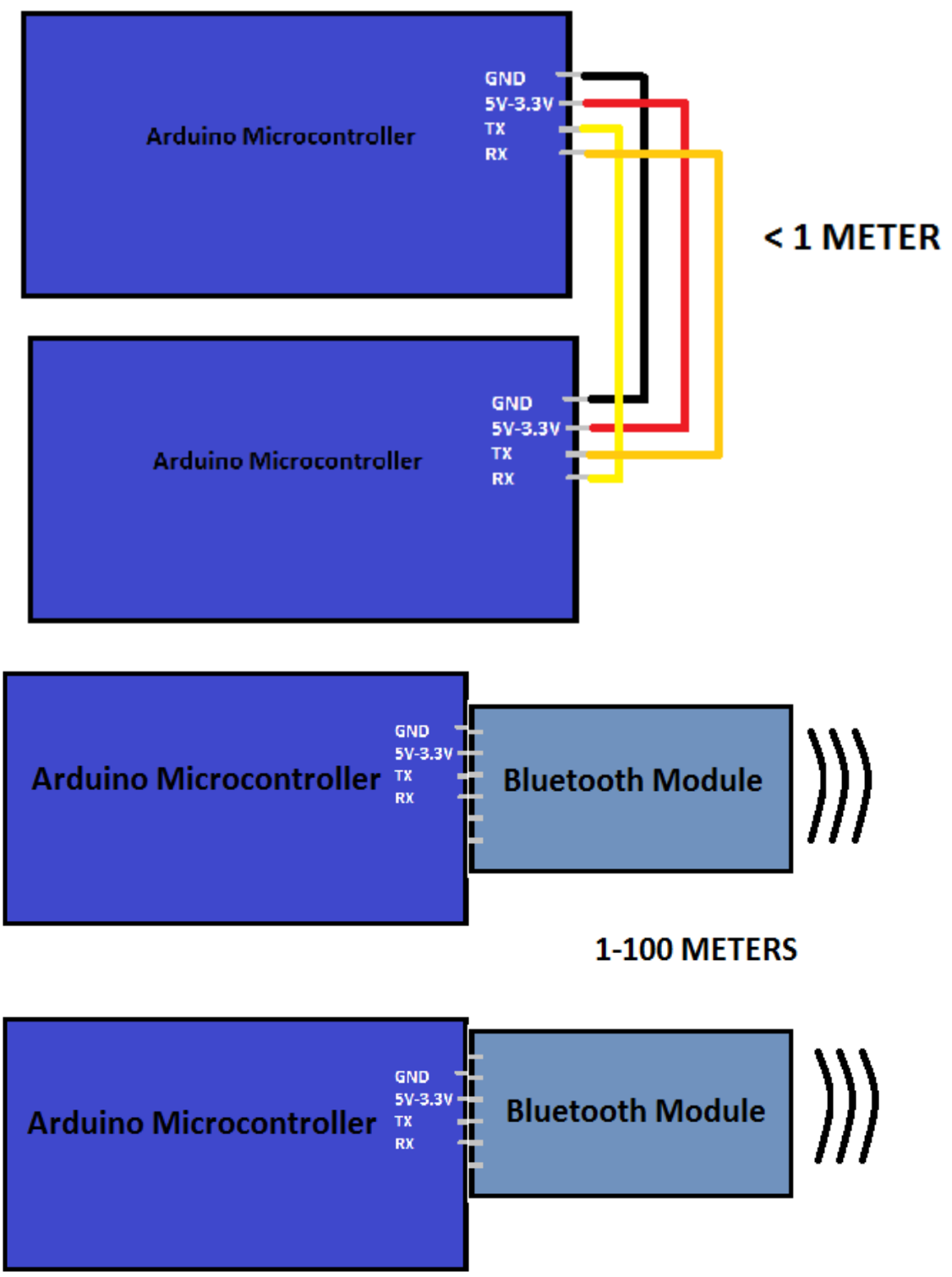


Figure 8: Example of RS-232 vs Serial Port Profile (SPP)

The Human Interface Device (HID) is a profile designed for Bluetooth-enabled user-input devices. The Human Interface Device (HID) profile generally is used with devices such as mice, keyboards, and joysticks. Many modern video game controllers, like WiiMotes and PlayStation controllers use The Human Interface Device (HID) profile. The Human Interface Device (HID) profile is generally used to replace USB cables.

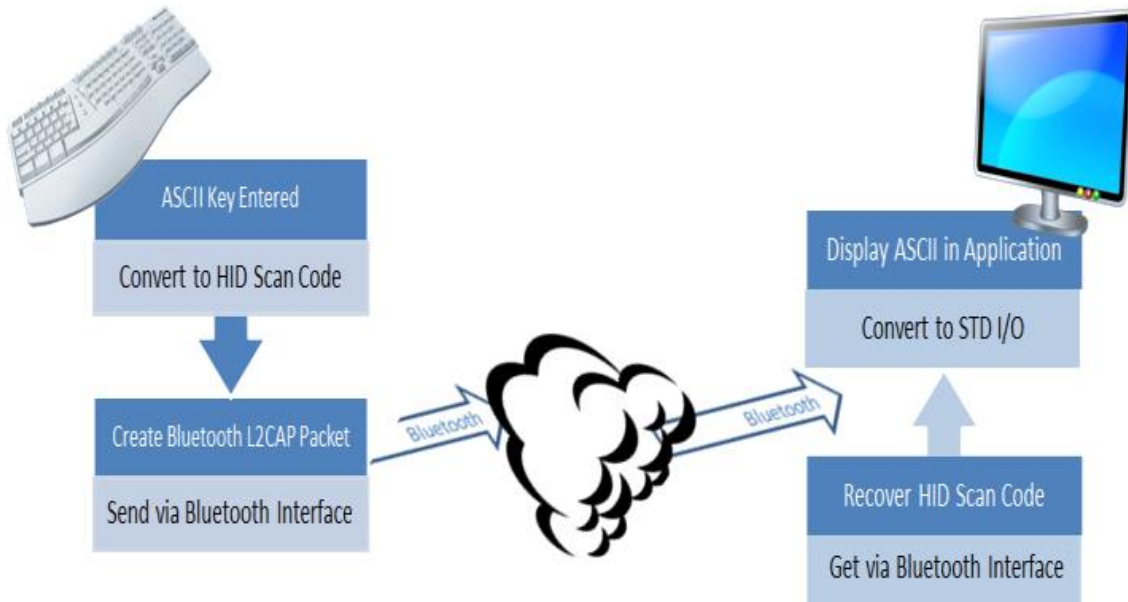


Figure 9: Example of HID interface, from RN-42-HID User's Guide

The Hands Free Profile (HFP) and the Headset Profile (HSP) are quite similar. Both the Hands Free Profile (HFP) and the Headset Profile (HSP) are designed to communicate hands free. The Headset Profile (HSP) is generally used in Bluetooth headsets and earpieces for hands free communication. The Hands Free Profile (HFP) is generally used in cars and offers the ability to use basic phone interactions. These interactions usually include the abilities of accepting calls, rejecting calls, hanging up, dialing, and other basic abilities.

Advanced Audio Distribution Profile (A2DP) and A/V Remote Control Profile (AVRCP) are also quite similar to each other. Both Advanced Audio Distribution Profile (A2DP) and A/V Remote Control Profile (AVRCP) are used transmit audio. Advanced Audio Distribution Profile (A2DP) works using simplex transmission. This means that the audio can only travel in one direction with a fixed transceiver and a fixed receiver so to speak.

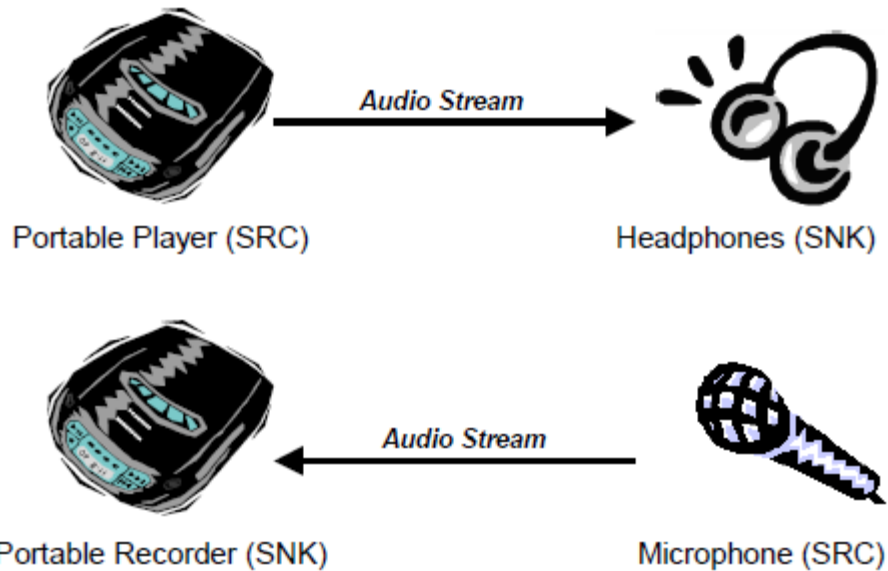


Figure 10: Example of Advanced audio distribution profile (A2DP)

Image from (<https://learn.sparkfun.com/tutorials/bluetooth-basics>)

A/V Remote Control Profile (AVRCP) is also used to send audio. However, A/V Remote Control Profile (AVRCP) is not limited to sending audio in a single direction and can send audio in both directions. This can be used with the headset that has a microphone. A/V Remote Control Profile (AVRCP) is more versatile since it allows for multi-directional audio transmission. Advanced audio distribution profile (A2DP) is limited to one direction but may have better quality transmission.

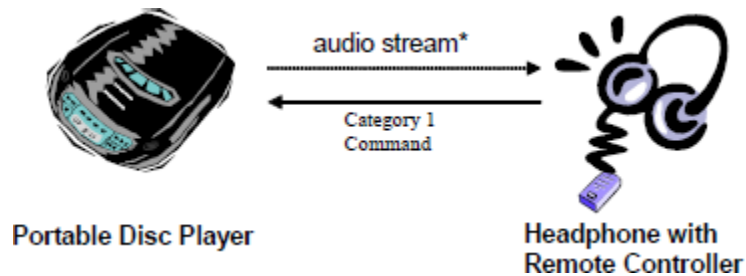


Figure 11: Example of A/V Remote Control Profile (AVRCP)

Image from (<https://learn.sparkfun.com/tutorials/bluetooth-basics>)

## 4.0 Component Research

This section will cover the group's research into each and every aspect of the Solar Powered Flower Sculpture. For each component of the sculpture project, steps taken to determine the procedures and actions that best suited our group's project will be retraced.

### 4.1 Solar Statue

This was one of the biggest challenges that the group faced...deciding what to build the Solar Powered Flower Sculpture's physical statue out of. The main components of the Solar Powered Flower Sculpture's structure is the base and then the flower petals.

#### **Base Idea 1**

To begin, the group figured that the base could be custom ordered online. This customization would include the Solar Powered Flower Sculpture's dimension specifications, a battery pack holder either on the bottom or on a side panel, a hole drilled in the center so that the wires easily could be ran from the NeoPixel RGBW LED strips to the printed circuit board, where they will be soldered, and then handles on the side for easy gripping. Now with this idea, the next step was to determine exactly which material to have the Solar Powered Flower Sculpture's base be made out of. This material would have to comply with being in the outside elements. This would include being waterproof, not melting or becoming malleable in extreme heat, not cracking in extreme cold, and be able to withstand the weight of the Solar Powered Flower Sculpture flower petals. The group then chose to consider polyvinyl chloride, melamine, or Bakelite.

Polyvinyl chloride just so happens to be both water resistant and fire resistant [16]. Due to the chemical structure of polyvinyl chloride, most people now a days, would know polyvinyl chloride as being the material of shower curtains, raincoats, and even the water pipes for plumbing in houses are made of. Due to its usefulness, there is an abundance of the material in different sizes and thicknesses. This polymer was a top competitor into the group choosing the base material.

Melamine is an organic-based, nitrogen-rich compound used to manufacture cooking utensils, plates, plastic products, and more. Melamine resin is durable, fire and heat resistant and virtually unbreakable, making melamine products more desirable than other plastic housewares [18]. With further research it was determined that this melamine resin is considered virtually indestructible because it is in the family of thermosetting plastics. However, it was also found that even though melamine resin is fire retardant, if heated to a certain temperature there is a possibility of releasing toxic chemicals that could make children or animals sick in case of ingestion. So the group decided against using this plastic.

Bakelite is considered the first trademarked synthetic resin. It is used in the making of circuit boards and heat proof insulated parts in the electronics industry [19]. Due to its insulating abilities, Bakelite is also considered the first commercially produced synthetic material. Through more research, it was found that Bakelite is made from phenol-formaldehyde resins, and as such are heat resistance and waterproof. Despite these positives, it was determined that Bakelite could become brittle over time and the group decided to not use this plastic material for the Solar Powered Flower Sculptures base.

From analyzing each and every plastic listed above, the group decided to potentially make the base of the Solar Powered Flower Sculpture out of the polyvinyl chloride. This material seems to be the most reliable when tested against the elements.

### **Base Idea 2**

The next idea for the base of the Solar Powered Flower Sculpture would be to make it out of wood. This idea came about since a wooden base would allow a rustic feel for the customer, and blend with the intricacies of the electrical components. This rustic feel also allows for the base to blend in with nature in the customer's own designed back yard.

With wood being porous, the group needed to find a way in which it could protect the electrical components within the base. To do this, it was determined that a sealant of sorts would need to be applied at the crevices where the wood meets. Also, to protect the actual wood, a paint or sealant would be needed. Once everything would be put together and sealed to prevent water damage, the group would be able to test whether or not the sealant was also fire/heat resistant. If all of these requirements are met, then handles would also need to be installed along the sides of the base. These handles would allow for ease of lifting the Solar Powered Flower Sculpture as a whole. This would also meet the group's health and safety specifications and specifications requirements for portability.

Another addition to the base would be to add a battery compartment on the side. With building the base out of wood, the group would have an easier time being able to cut out a section. This section would then allow a battery pack to be harnessed inside for whenever the customer would need to replace the rechargeable battery. The cover for the section would then be able to latch in some way as to protect the battery compartment from the elements.

After doing some research, the group narrowed the wood selection types down to teak and ipe since they have the highest resistance to rot over time and then oak.

Teak is considered one of the most desirable woods in the world [22]. This type of wood is mostly used in the construction of ship building, furniture, or exterior construction which would work perfectly for the Solar Powered Flower Sculpture's base. This style of wood would also work well the group member's idea of having

a wood that looks rustic since teak has the ability to darken in color with age. And even with its high rot resistance, teak is also easy to work with. Despite all of these great qualities, teak happens to be quite expensive due to its high quality and ability to be highly resistant to wood rot and water damage.

Ipe is a wood that is commonly used while laying flooring or decking, tool handles, and exterior lumber [23]. The color would vary depending on the variety of ipe the group would decide to choose since ipe can be reddish brown, yellowish olive brown or a darker almost blackish brown that could also look striped, alternating between darker brown and black. Ipe is also rated as very durable and having excellent insect resistance which would make for a great base for the Solar Powered Flower Sculpture due to being placed outside for an extended period of time. However, it was determined that ipe is not easily worked with. Due to ipe being extremely hard and dense and having a high cutting resistance, the group decided to not go with this style of wood.

The group's last option was to construct the Solar Powered Flower Sculpture out of oak since most commercial manufacturers choose this type of wood for its durability and fungal resistance [25]. Oak is mostly used in the construction of hardwood floors, tables, and other kitchen decorations and furniture pieces. Due to its popularity in consumer markets, oak wood is readily available for purchase. With that being said, there are two different types of oak wood to choose from, white or red.



Figure 12: Red Oak



Figure 13: White Oak

After doing some research, about the difference between the red oak and the white oak, it was determined that when the end grains of each were analyzed under a microscope that the pores looked completely different. In the red oak, the pores were open and porous, which means it would not be well suited for any contact with water without some kind of super sealant. The white oak pores on the other hand, when analyzed under the microscope, were plugged with tyloses, which are interior growths that form inside a stressed plant to keep any more harm from damaging the plant. These tyloses plugs help make white oak suitable for water resistance and give it increased resistance to rot and decay. Based on this discovery, the group decided to use white oak wood as the base of the Solar Powered Flower Sculpture with a wood sealant to further protect the white oak wood from rotting.

In an effort to now protect the white oak base of the Solar Powered Flower Sculpture, the group did further research into how to waterproof the white oak wood. It was determined that wood could be waterproofed using different types of oils, a water resistant sealant, or by using a proper wood stain.

The main oils that are used to treat wood are linseed, walnut, and Tung. There is a complication with using linseed oil seeing as though it contains poisonous metal drying agents. The only way to avoid these metal drying agents is by using linseed oil that is completely raw since any kind of cooking is what activates the metal drying agents. Tung oil when found in its raw form is quite expensive. To avoid this additional increase in costs, Tung oil can also be found as an additive in certain cleaning products. The most convenient of all the oil types would have to be walnut oil. This is because walnut oil is so abundant that it is sold at the grocery store in the aisle next to olive oil. The only downside is that since it is an oil, it has a lingering residue which would cause health and safety harm to anyone who had a nut allergy.

When it comes to finding the correct sealant for the white oak base, the group focused solely on a water based wood sealant. Also, with the white oak wood already having such an earthy tone to it, the groups decided against the thought of using a tinted water based wood sealant. Further research also led the group to search after a marine wood sealant that is used on outdoor wooden furniture and watercrafts or boats. This marine wood sealant would help the white oak wood be protected against humidity, the sun's harmful ultraviolet rays, and then of course water resistant. The group would be able to apply the sealant from either a spray aerosol sealant or by painting the sealant onto the Solar Powered Flower Sculpture's base by hand.

From researching into wood stains, it was determined that an oil based stain would work best. Stains are just what the name says, but a stain that seals the pores of the wood the stain is protecting. It was also determined that the lighter the stain the less likely it will maintain in the outside weather. To make a wood stain work for the Solar Powered Flower Sculpture's base, a dark exterior stain would have to be chosen. With this, the group chose not to go with this seal option since the group wanted to keep the white oak woods natural color preserved.

Now knowing all the different types of ways to make the white oak wood water resistant, the group decided to use a marine wood water based sealant. This water based wood sealant would allow the white oak wood to withstand the elements while maintaining the beauty of the natural white oak wood.

### **Flower Petals**

With the flower petals being the main focus of the Solar Powered Flower Sculpture, the group really had to put some thought into what material would be used for the flower petals. The original idea was to construct the Solar Powered Flower Sculpture completely out of white oak wood. This idea was then decided against because all of the white oak wood would weigh down the Solar Powered Flower Sculpture, making it not fit within the group's requirement specifications. It was then discussed about the possibility of three-dimensionally printing the flower petals. This would then allow the group to pick not only the shape, size, and dimensions but also the exact material of the flower petals. The group would be able to three- dimensionally print the flower petals on campus at the Texas Instruments SMART Lab. This would also allow for the NeoPixel RGBW Digital LED strips to be easily adhered to a plastic, polymer material.

## **4.1.1 LEDs**

Currently there were are a few options lighting systems, the incandescent light bulb, the fluorescent lamp and the light emitting diode. The project required the light system to run off of the battery in the unit, be controlled by the microcontroller, have a high tolerance for rapid on and off cycles, and be used in displays of multiple colors.



The first source of light that we looked at was the incandescent light bulb. This is the oldest of the lighting technologies that we looked at and had the simplest of designs. Light is produced by applying a current to a tungsten filament enclosed by either a vacuum or an inert gas. The tungsten glows when the current heats it up, producing the light. The problems we would have with this light source is the lack of endurance, and size limitations. As the tungsten is heated up it will evaporate into the bulb. Over time the tungsten will erode, until it's too thin to support the current being passed through it. The thinning of the tungsten will cause the light bulb to fail and need replacing to often to make this a viable choice. This light source is also very inefficient, which makes it a poor source of a solar powered device. Most of the current generated will be used to heat the filament, causing a lot of lost energy due to heat loss. Only about ten percent of the power generated will be converted to light energy. The heat generated principle of the light source's mechanics also make starting fires an additional safety concern for the device. The size requirement of the filament also limits how small we can make the bulb. This is not an issue for large light displays, but our display will be too small to accommodate this type of light bulb. In addition to these issues the incandescent light bulb is being phased out, shorting the potential lifespan of the project.

The second light source that was considered was the fluorescent light bulb. These were actually designed to replace the incandescent light bulb. Instead of relying on heat to produce light, these rely on a reaction between excited mercury molecules and phosphor. Using electricity to excite the mercury in the tube, causes it to give off ultraviolet light. The ultraviolet light then goes on to collide with the phosphor coating the inside of the light bulb. The collision between the ultraviolet light and the phosphor causes the phosphor to glow, producing the light. This makes the fluorescent light bulb a lot more energy efficient than the incandescent light bulb, which is a major concern for our design. The problems we found with this light source was that there was a delay between the time the light was activated and the time that it actually produced light. Similarly there was a delay between the time when the light was deactivated and the glow of the light to finally fade. This made the light source unfit for rapid on and off cycling that will be used in our light display. The size limitation was also an issue for this light source. Creating a display large enough to accommodate a sufficient amount of fluorescent light bulbs was prohibitively large and expensive for this project.

The third and final type of light source that we considered is the LED. The Light Emitting diode is still more efficient yet than the fluorescent light bulb. This light source uses a very different technology than the other two previously considered light sources. Just as its name suggest, this light source is a light emitting diode. Like other diodes the LED has a positive and negative region within a solid state body. Energy is released when a current flows across the boundary. This energy is the light that is emitted from the diode. To get the appearance of white light three different colors are emitted in close proximity. These colors are blue, red, and green. This lighting source meets all of the requirements for our project. Because the light source is an electrical component it is easy to control as other diodes.

Being made from solid state parts make it more durable than the other light sources considered. This also gives the LED a longer life of operation than the other two. Which translate into fewer light bulb changes for the user. They are also the right size for the type of display we want to make on a scale that fits our budget. The other light sources would've cost us the portability feature of our project. The lack of components that contain toxic compounds like mercury also meets our environmental constraints. LEDs also don't have any issues with rapid on and off cycles, which is essential for our light display. The way we intend to use the lights in our display could case the microcontroller to crash due to a voltage drop, but that can be overcome by powering the LEDs directly from the battery. LEDs also have the advantage that one LED can emit different types of color. This prevents us from having to use a different light for each color we want to incorporate in our design. Because LEDs aligned so well with our requirements, this is the type of light source we choose for our project.

The first LEDs we looked at were the LTROP 2 Reels 12V 32.8ft Waterproof Flexible RGB LED Strip Light Kit, Color Changing SMD5050 300 LEDs, LED Strip Kit & Mini 44-key IR Controller + 12V 5A Power Supply, Adhesive Light Strips. We hadn't finalized our display design yet, but we know we wanted to incorporate multiple colors in the display. These LEDs were waterproof which would allow us to use them outside without worrying about inclement weather ruining them. They come with self-adhesive on the back which would simplify installation of the LEDs. Every three LEDs on a strip can be cut off of the main strip without damaging the functionality of the lights.

The second LEDs we looked at were the SUPERNIGHT (TM) 16.4FT 5M SMD 5050 Waterproof 300LEDs RGB Color Changing Flexible LED Strip Light. This one was cheaper than the LT0025 and still had many of the features of the LT0025. They are also waterproof, come with self-adhesive on the back. Instead of using different LEDs for different colors the entire strip was capable of changing color. The LC-128 also had the bonus of being a tri-chip setup. They have three LEDs in one housing for added brightness.

The last LEDs that we looked at were the Adafruit NeoPixel Digital RGBW LED strip features 4 LEDs in them (red, green, blue and white) which allow for excellent lighting effects. These were the most expensive LEDs that we looked at but they also had the most of the features we wanted in our design. These were also waterproof and had the extra feature of one integrated RGB LEDs per segment that could be controlled individually. This would allow us to get create the biggest variety of displays with the fewest amount of LEDs... These LEDs require specific timing because they use 800 KHz protocol. Table 3.1.1a is the data of the lights being compared.

Table 3: Comparison of LED Lights (By Mahaley Vann)

Model Number	LTROP LT0025	Super night(™) LC-128	Adafruit NeoPixel Sk6812RGB W
Price	\$0.09 per LED	\$0.05 per LED	0.45 per LED
Minimum Operating Voltage (volts)	12V	12V	5V
Operating Amperage (milliamps)	5000mA	5000mA	80mA
Operating Temperature range (degree Celsius)	-20 to 60	-20 to 60	-20 to 60
Power Consumption (mill watts)	105	64	180
Color Temperature (kelvin)	4000 to 4500	4000 to 4500	4000 to 4500
(Lumens)	360	360	360

Table 3.1.1a

## 4.1.2 Power Supply

The power supply as a whole is what will run the Solar Powered Flower Sculpture. The group wants to power the system using photovoltaic cells, a rechargeable battery, and a microcontroller unit. By placing these in parallel, the photovoltaic cells will absorb the solar rays, which will then be converted into DC electric current. This current will then be split between the rechargeable battery and the microcontroller unit. So that the microcontroller unit does not receive excessive voltage, which would cause damage to the interior unit, a voltage divider network is added between the photovoltaic cells and the microcontroller unit. Also, so as to not completely destroy the photovoltaic cells, due to the reverse current sent out

by the rechargeable battery, a diode will be placed right after the photovoltaic cells. The idea to set up the power system in this way was found from a do it yourself website called Simple Solar Circuits and the circuit looks as follows:

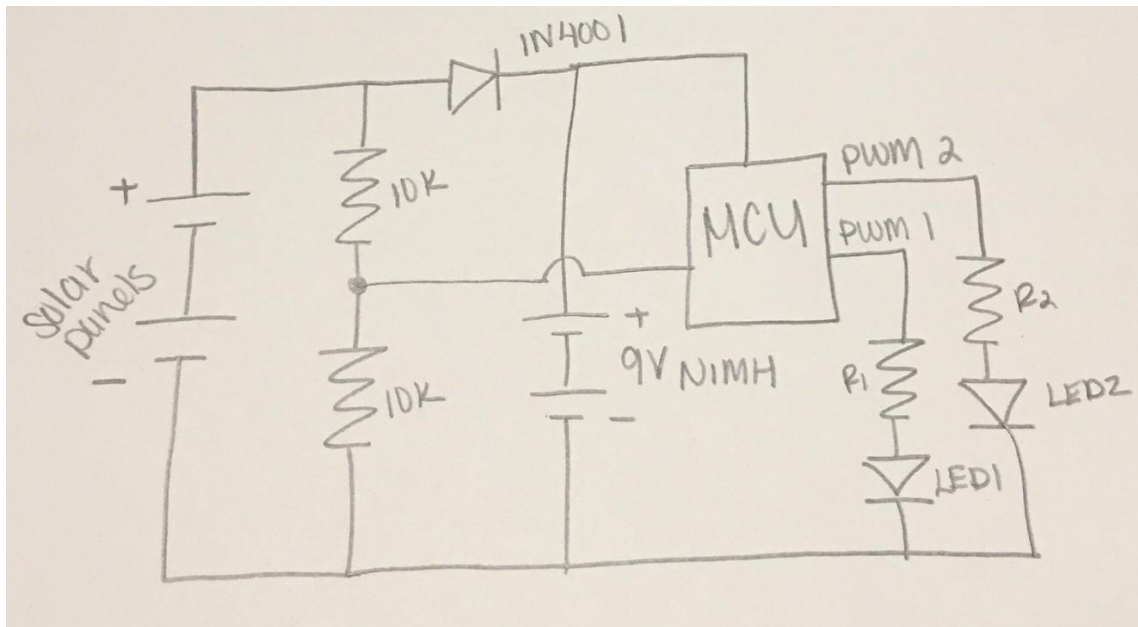


Figure 14: First Circuit Prototype

Despite finding this circuit from a website, it was determined that the rechargeable battery would fail sooner than it was supposed to due to not having any regulations. Further research on a more efficient circuit was required.

After doing more research on how to preserve the life cycle of a rechargeable battery, the group found a better way to connect the photovoltaic cells, rechargeable battery, and microcontroller unit. This would include putting multiple components in parallel starting with the photovoltaic cells, a charge controller, the rechargeable battery, a voltage regulator, and then the microcontroller unit. With this more stable circuit, the group could then research individual parts for the two new components.

### **Charge controller**

To make the group's lives easier when it came time to build the printed circuit board, extensive research into the Eagle library was done to determine which charge controller would benefit the Solar Powered Flower Sculpture the most. When scrolling through the Eagle library, the group came across a specific charge controller that was named nickel cadmium/nickel metal hydride battery fast charge controller. This charge controller is created by a company name Maxim Integrated and is component number MAX712. This seemed to be the perfect charge controller since the group chose to use nickel metal hydride as the battery for the Solar Powered Flower Sculpture.

Now that all of the pinout locations are known, the group read further into the MAX712 datasheet to see if a schematic was already constructed to show how to connect a power source to the charge controller and then be connected to the rechargeable nickel metal hydride battery. The schematic below was then found:

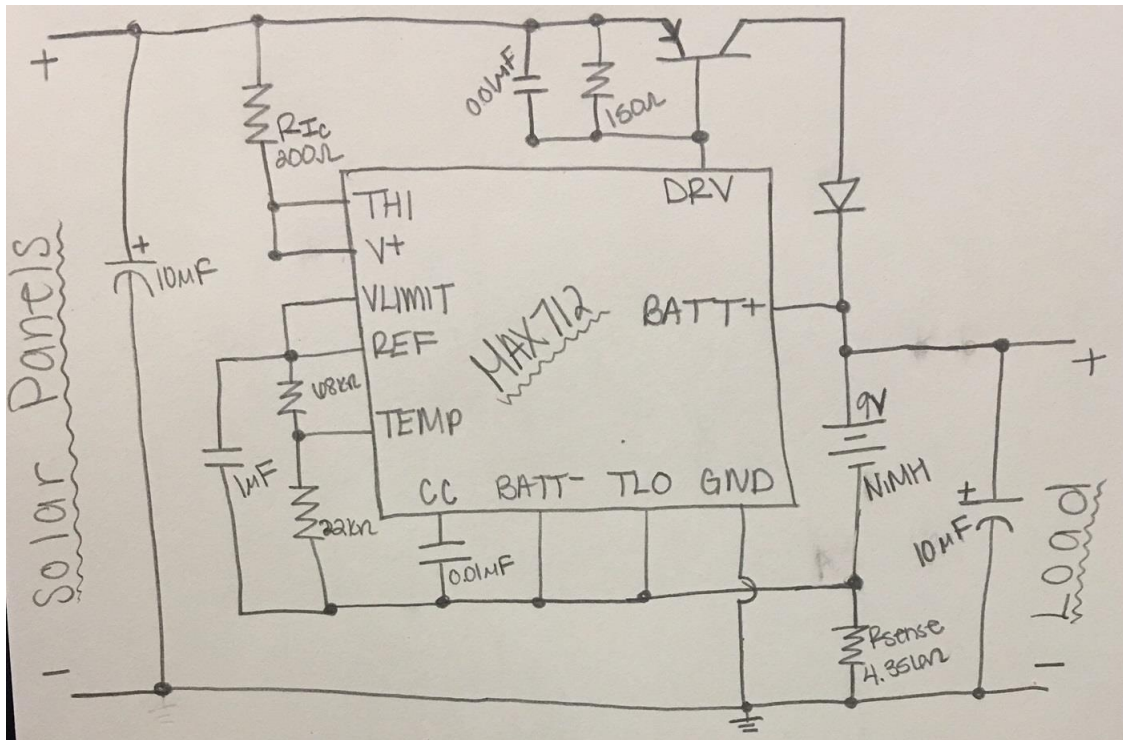


Figure 15: MAX712 Schematic

Even though the figure above shows the input being a wall cube, the group will put the photovoltaic cells as the input for the charge controller. The nickel metal hydride battery will be placed where the figure says BATTERY, and then the voltage regulator will be placed where the figure says LOAD.

### Voltage Regulator

Since the group is using a nickel metal hydride battery as the power source to the microcontroller, a DC to DC step down voltage regulator is needed to drop down the DC voltage of the nickel metal hydride battery to an acceptable voltage that the Arduino Nano can use without frying the internal chip. The Arduino Nano needs an input voltage of 1.5 volts to 5 volts maximum, and since the nickel metal hydride battery chosen is rated 9 volts a 5 volt DC to DC step down voltage regulator is needed.

With previous knowledge from electronics classes a linear voltage regulator, LM7805, was chosen. This linear voltage regulator outputs a steady and constant 5 volts and 1 amp.

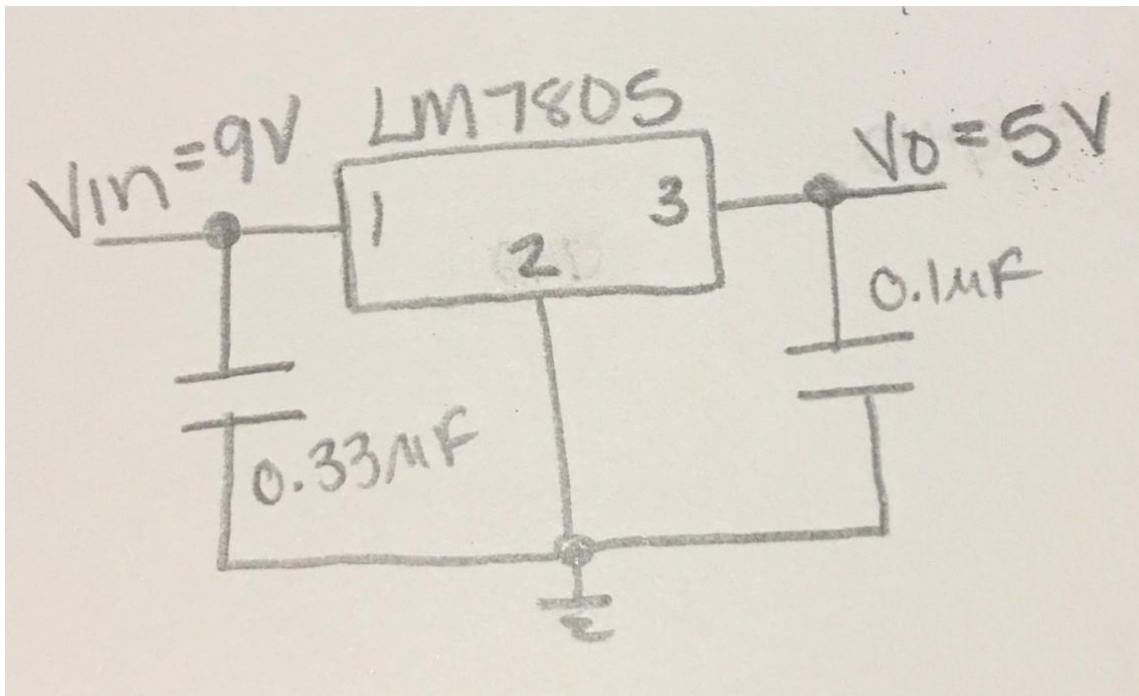


Figure 16: LM7805 Schematic for Circuit

From this schematic, the group is able to connect the MAX 712 charge controller output to the input shown above. This circuit is able to step down the DC voltage to where the voltage at the output will be 5 volts, of which then the output can be connected to the power supply of the Arduino Nano microcontroller. However, after testing, the group decided that this LM7805 voltage regulator would be too inefficient and too much power would be lost.

To correct this, a high efficiency step down DC to DC converter was chosen. This converter is technically a buck switching regulator. Yet again looking through the Eagle library, the group found a 5 volt 225 mA switching regulator, the MAX639. This switching regulator takes in an input voltage of 9 volts but can vary in the range of 4 volts to 11.5 volts and output a constant 5 volts. This switching regulator is perfect since the Solar Powered Flower Sculpture is operating with a 9 volt nickel metal hydride battery.

With knowing this pinout configuration, the group would be able to also find a schematic in the datasheet to connect to the MAX712 charge controller and the Arduino Nano microcontroller. In the datasheet, the group found this schematic

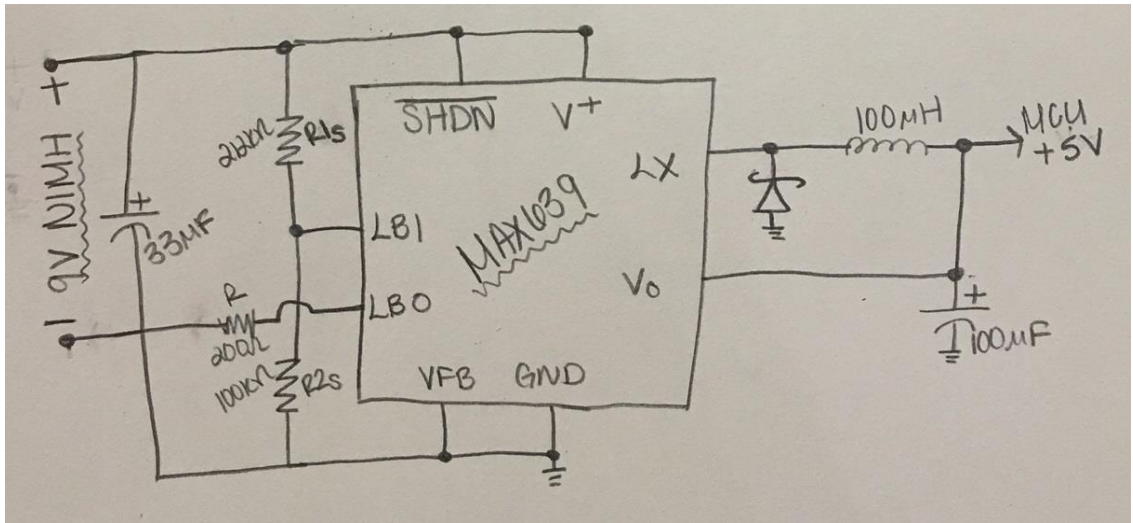


Figure 17: MAX639 Inner/Outer Schematic

From the schematic found, the group is also able to learn the inner workings of the MAX639. With this knowledge, the group can connect the input of the MAX639 switching regulator to the output of the MAX712 charge controller. Next, the output of the MAX639 switching regulator is connected to the power supply pin on the Arduino Nano microcontroller. With this, the power supply system of the Solar Powered Flower Sculpture is complete.

## 4.1.2a Photovoltaic Cells

Before deciding which size of photovoltaic cells to use for the Solar Powered Flower Sculpture, the group decided to research exactly what a photovoltaic cell is and how it works. From NASA's science beta website, they said that photovoltaic cells are based on the photoelectric effect. The photoelectric effect states that there are certain materials that have the ability to absorb light particles, known as photons, and convert that energy absorbed into electricity. This is relevant to photovoltaic cells since they are composed of semiconductor wafers that are able to produce an electric field within the two opposing electric plates of the photovoltaic cells. This induced electric field is then able to create a steady stream of current throughout the photovoltaic cell. With this steady stream of current, any electrically powered device could be attached and then operate off of the solar power provided.

With the research into exactly how photovoltaic cells operate, the group was then able to look for the specific characteristics needed for the Solar Powered Flower Sculpture. These characteristics include physical size, number of photovoltaic cells needed to operate, and the power rating/wattage. Since a specification requirement for the Solar Powered Flower Sculpture is for it to be portable, the photovoltaic cells needed to be able to fit within the three cubic feet requirement set by the group. This at first seemed to complicate the research process since

most photovoltaic cells on the market are for residential and commercial use of which does not meet the group's specification requirement. After doing some searching on the internet a website selling individual photovoltaic cells for at home, do it yourself projects was discovered. All of the photovoltaic cells offered were less than five inches in length and width. To fit the group's requirement and to not cause complete disarray on the base, the photovoltaic cells were decided to be less than three inches by three inches. From here, it was then determined the number of photovoltaic cells needed. Through research and talking with people in the solar industry, the group discovered the inefficiency of harvesting and utilizing solar energy. Due to the ability of certain photovoltaic cells being blocked from collecting solar rays, the group wanted to increase the opportunity to harvest solar power by increasing the number of photovoltaic cells that would be connected in series. The group also need to determine the power rating/wattage need to operate the Solar Powered Flower Sculpture. The photovoltaic cells will be connected in parallel with a charge controller of the group's choice. The specific charge controller used needs a voltage at least 1.5 volts larger than the battery being charged. With these requirements the group was able to determine which photovoltaic cell to use. The photovoltaic cell chosen is a monolithic copper indium diselenide photovoltaic cell. It is a 60 mm by 60 mm square that produces 4.5 volts and 90 mA. These details match the group's size requirement, but to match the number and power rating/wattage more than one photovoltaic cell is needed. The group decided to series stack six of these photovoltaic cells around the base of the Solar Powered Flower Sculpture. Six of these photovoltaic cells will generate 27 volts which is more than 1.5 volts larger than the battery size the charge controller is charging. Also, by series stacking six of these photovoltaic cells, if by chance, a portion of the photovoltaic cells are shaded, there will still be enough power harvested by the unshaded photovoltaic cells to power the charge controller.

## **4.1.2b PCB Design**

For this Solar Powered Flower Sculpture, a printed circuit board is required to control, power and operate the whole design. To do such, multiple schematic to printed circuit board programs were researched. After looking around and then also getting feedback from graduated students and local professional engineers in the field, the group decided to use EAGLE 7.7.0 standard to construct the printed circuit board. This was decided since the schematic editor is similar to Multisim, which was used extensively throughout the group's higher degree education. EAGLE also has an interface which generates and switches from schematic editor to printed circuit board. The initial printed circuit board for the Solar Powered Flower Sculpture will be based off of the image below:



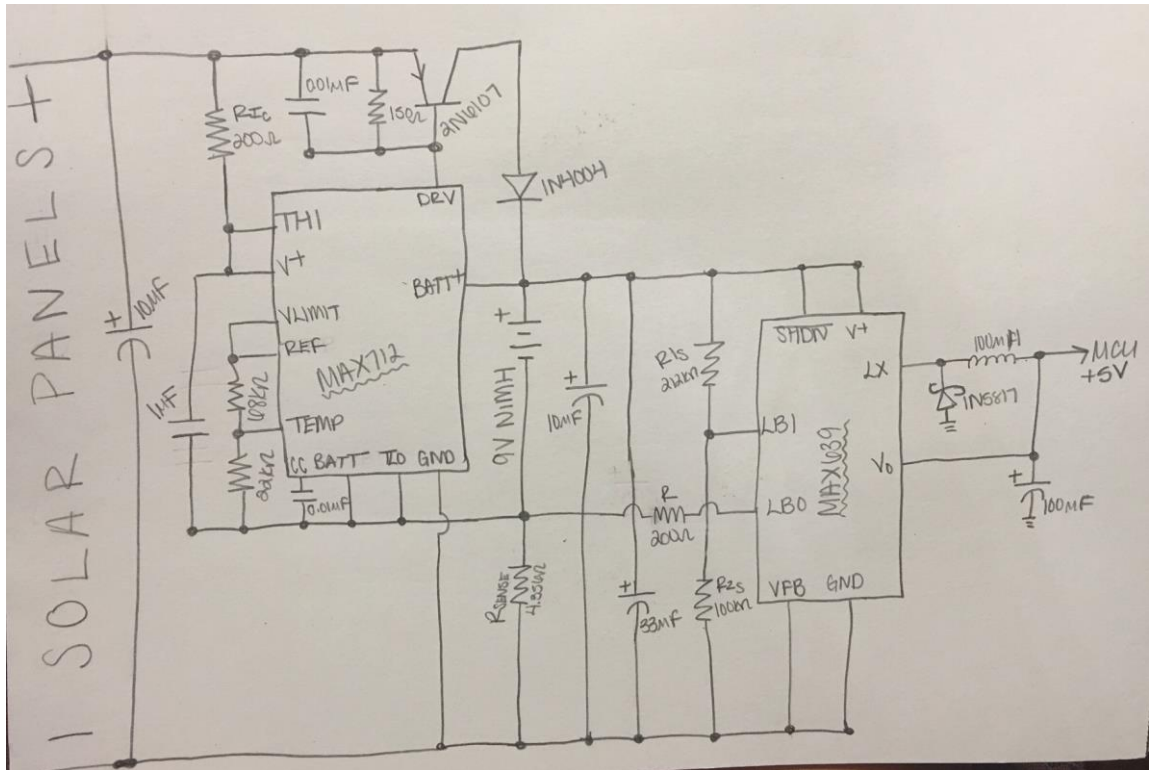


Figure 18: Printed Circuit Board Schematic

## 4.1.2c Battery

In today's day in age, batteries are the main source of power for most portable electronics that are sent out to the everyday consumer. Even if a certain product has the capability of being plugged into an outlet and running off of AC power, almost every product produced has some sort of battery within itself that can hold some charge to power the system. So, even though the Solar Powered Flower Sculpture is sustained by the natural power of solar energy, the efficiency of solar power cannot be trusted on its own without the support of some kind of battery. After extensive research, it was determined that there are five different kinds of batteries available on the market today. These include: Lead Acid, Lithium Ion, Lithium Polymer, Nickel Cadmium, and Nickel Metal Hydride.

### Lead Acid Battery

The first style of battery to be researched is actually the oldest rechargeable battery to still be used today in existence. This style of battery is what is used to maintain power in larger electronics including, but not limited to, cars, motorized scooters, motorized wheelchairs, golf carts, or any larger electronic device that requires a battery to run its operating system. Compared to some of the other batteries offered on the market, the lead acid battery does not have the best recovery rate when completely discharged. This means that when the battery is

either drained completely of all its stored energy or not completely charged to its saturated capacity, that a detrimental impact on the life cycle of the battery is affected. With this problem and the fact that the lead acid batteries are substantially larger in size, the group decided against using this style of battery.

### **Lithium Ion**

This certain style of battery is created with one of the lightest metals known to humankind. With this, the lithium ion battery also has the largest energy density for its weight and the greatest electrochemical potential. In comparison, to the other style of batteries, the lithium ion battery's density is double that of the nickel cadmium batteries. Based on the density of the lithium ion battery's energy, companies are able to manufacture single cell battery packs in this style. These packs are usually seen nowadays inside of mobile phones or other electronic devices that allow for clip in single cell battery packs. With this, each lithium cell battery pack is capable of producing 3.6 volts in comparison to the standard 1.2 volts of other single cell batteries. However, despite having a higher voltage, there are no rechargeable lithium ion batteries available on the market today due to the fragile nature of lithium metals. With this the group decided to go with another style of battery.

### **Nickel Cadmium**

This certain style of battery is composed of nickel hydroxide at the cathode and cadmium at the anode. In comparison to the lead acid battery, the nickel cadmium battery is double in energy density. The voltage of a single cell is a standard 1.2 volts. The nickel cadmium is used in the consumer market in low power products such as back up lighting, electric grooming equipment, power tools, and battery powered children's toys. With this, this style of battery is great for any product that would typically cause a battery to overheat since the internal resistance is so low that almost no heat is produce. Due to the construction of the nickel cadmium battery, the recharge rate is only a matter of a couple of hours. This makes it a great choice of battery since if it were to discharge completely, it has the capability of being recharged quicker than most other battery styles. However, the group decided to not use this style of battery due to its susceptibility to memory. This means that if the battery does not discharge or recharge fully at least once a month to a couple of months, it "remembers" that voltage level and will not meet the original capacity requirement.

### **Nickel Metal Hydride**

The nickel metal hydride battery is an advancement to the nickel cadmium battery. With this being said the chemical composition is almost identical. The main difference is between the anodes, of which the nickel metal hydride has a hydrogen absorbing anode compared to a cadmium anode. Also, in comparison, the nickel metal hydride is superior to the nickel cadmium since there is no memory effect. This means that the nickel metal hydride is capable of being drained partially of its

capacity and then recharged without needing to be periodically fully discharged. This being said, nickel metal hydride batteries are typically used in the market with products that require large rates of capacity and are used on a frequent basis such as digital cameras, portable music devices, and electronics that are run off of solar power. With this research, the group decided that it would be best to use this style of battery to run the Solar Powered Flower Sculpture. Due to the AA nickel metal hydride batteries only having a standard voltage of about 1.25 volts, the group decided to go with the 9 volt alternative.

### **4.1.3 Wireless Module**

A Bluetooth wireless module was selected for this Solar Powered Flower Sculpture. The Solar Powered Flower Sculpture has its specific project requirements. Ease of integration, along with experience, time, ability and many others factors went into the decision of using a Bluetooth module. Nevertheless, there was much researching and exploration required to find out what the Bluetooth module offered along with its limitations.

First, what is Bluetooth and how does it work? Bluetooth is a wireless communication standard. It allows devices to connect wirelessly over a distance. Bluetooth uses radio waves to accomplish the wireless communication. Bluetooth devices have a tiny computer chip which have Bluetooth software and a Bluetooth radio which make it easy to connect. Before Bluetooth devices can connect they must pair. Once to Bluetooth devices are connected the resulting networked formed is known as a piconet. The piconet can range from two to eight device. Once a network is established one device takes on the role of master and all other become slaves.

Second, what can the Bluetooth module accomplish? The Bluetooth module can easily connect the intended parties together, which are the Android application and Arduino board. Wirelessly through radio waves the Android application will be able to communicate and effectively have control over the Solar Powered Solar Sculpture through the wireless module. Using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz, the wireless module will work from a range of approximately 10 meters (about 30 feet). Also, since it is Bluetooth technology, two to eight devices can potentially be connected on the network.

Third, how can the Bluetooth module be integrated? The Bluetooth module is easily integration with the Arduino board. Very little is required in order to be able to integrate the Bluetooth module with the Solar Powered Solar Sculpture. To integrate the Bluetooth module, it simply needs to be connected with the Arduino board. The Bluetooth module needs four pins to be connected which are, the ground pin, the VCC, the Rx receiver, and the TX transmitter. Once all four pins are connected correctly the Bluetooth module has been integrated.

There are a wide range of Bluetooth module devices designed for microcontrollers. These Bluetooth modules can support different features and have different capabilities based upon which version of Bluetooth they support. Each version of Bluetooth can perform the features that that version of Bluetooth was designed for. At this point in time certain versions are used more than others and some versions have benefits compared to other versions.

Table 4: Bluetooth Version Comparison (By Kelechi Ukachi-Lois)

Version	Data Rate	**Range	EDR*	High Speed	Low Energy
<b>Bluetooth Version 1.x</b>	Up to 721 kbps	Approximately 33 feet	N/A	N/A	N/A
<b>Bluetooth Version 2.x</b>	Up to 2.1Mbps	Approximately 100 feet	✓	N/A	N/A
<b>Bluetooth Version 3.x</b>	Up to 24 Mbps	Approximately up to 300 feet	✓	✓	N/A
<b>Bluetooth Version 4.x</b>	Up to 25 Mbps	Approximately up to 300 feet	✓	✓	✓
<b>Bluetooth Version 5.x</b>	Up to 50 Mbps	Estimated Approximately up to 1,200 feet	✓	✓	✓

\*EDR - Enhanced Data Rate

\*\*Range - the maximum range is determined only by the output power of the device.

The earliest versions of Bluetooth available are Bluetooth version 1.x. (Bluetooth version 1.x refers to all versions of Bluetooth starting with 1. Meaning 1.0, 1.1 etc. this notation will be used when referring to the all the different patches within that version) these versions of Bluetooth have been available for the greatest amount of time. However, Bluetooth version 1.x offers a limited amount of features. The features that Bluetooth version 1.x offers includes only the barebones connection abilities. Bluetooth version 1.x is not of much use today in the year 2016. At this point Bluetooth version 1.x is practically extinct and probably should not be purchased unless it is to be used as a collector's item.

The earliest version of Bluetooth that is still worth using is version 2.x. Bluetooth version 2.x is still somewhat relevant in today's market. Obviously, Bluetooth version 2.x was a step up from Bluetooth version 1.x. Bluetooth version 2.x offers improved capability and also some new features compared to Bluetooth version 1.x. Bluetooth version 1.x had limitations when it came to pairing and had compatibility issues which made it difficult for the products to be interoperable. Bluetooth version 2.x fixed a lot of these compatibility issues which made pairing a great deal easier. Bluetooth version 2.x also introduced an Enhanced Data Rate (EDR). This Enhanced Data Rate provided many benefits when compared to the previous Bluetooth version 1.x. This Enhanced Data Rate provided three times faster transmission speed and in some cases the Enhanced Data Rate offered a speed up of up to 10 times (2.1 Mbit/s) in some cases. Enhanced Data Rate not only increased speed but also reduced complexity of multiple simultaneous connections through to additional bandwidth. When compared to Bluetooth version 1.x, Bluetooth version 2.x also had lower power consumption through a reduced duty cycle.

Bluetooth version 3.x is one of the more recent versions of Bluetooth when compared with Bluetooth version 1.x and Bluetooth version 2.x. Bluetooth version 3.x was released in the year 2008 with the High Speed (HS) feature released in the year 2009. Bluetooth version 3.x's major addition was the High Speed option. As the name suggests Bluetooth version 3.x HS was designed to increase the data transfer to a higher speed with equal or better reliability. Bluetooth version 3.x is able to reach optimum speeds of up to 24 Mbps. This increase in speed comes from the Introduction of support for an alternate lower layer. All the applications that were available with Bluetooth radio earlier can be run over an alternate radio. For instance, 802.11, hence, the increase in speed is technically due to transferring data through Wi-Fi once the Bluetooth connection is achieved.

Bluetooth version 4.x is a very recent Bluetooth version. Bluetooth version 4.0 was released in the year 2011 and Bluetooth version 4.2 was released in the year 2014. Bluetooth version 4.x's most notable additional feature is the Low Energy (LE) feature. The Low Energy feature can add many benefits to users, one of the most obvious and concrete benefits is that this Bluetooth version is less taxing on the power supply. Low Energy uses significantly less energy but it comes with a tradeoff. The Low Energy feature severely limits the data rate. The data rate with Low Energy drops to approximately 100 kilobits per second. Compared with the 2.1 megabits per second the Enhanced Data Rate offers, Low Energy can only deliver close to one twentieth of the speed. Some of these Low Energy Bluetooth devices offer dual mode while others only offer single mode. Dual mode allows the Bluetooth device to switch between classic Bluetooth with the Enhanced Data Rate and Low Energy Bluetooth, however, dual mode will not experience the full benefits of reduced energy consumption. Single mode however, is more ideal for devices that focus on low power consumption.

Bluetooth version 5 is yet to be released. However, its release date is planned for early 2017. Bluetooth version 5 supports the same features as Bluetooth Version

4.x along with improved capability. Bluetooth Version is planned to significantly boost the basic features of Bluetooth. Bluetooth Version 5 has an expected range of 1200 feet which is four times greater than that of Bluetooth Version 4.x. Also the data rate of Bluetooth Version 5 is expected to double and boasts a new data rate of 50 megabits per second. The capabilities of Bluetooth Version 5 are expected to be much improved from its predecessors.

Table 5: Bluetooth Module Comparison (By Kelechi Ukachi-Lois)

Bluetooth Module	HM-10	HC - 05	WRL-12577	HC - 06	HM-15
Bluetooth Version	Bluetooth v4.0	Bluetooth V2.0+EDR	Bluetooth v2.0+EDR	Bluetooth v2.0+EDR	Bluetooth v4.0
Input Power Supply	2-3.7V	3.3 to 5 V	3-3.6V	3.1V~4.2V	5V
Connection type	Pins	Pins	Pins	Pins	USB
Basic Rate	✓	✓	✓	✓	✓
EDR*	✓	✓	✓	✓	✓
High Speed	✓	N/A	N/A	N/A	✓
Low Energy	✓	N/A	N/A	N/A	✓

\*EDR - Enhanced Data Rate

An important feature of the different Bluetooth versions is the range in which they are able to operate. Since the purpose of using Bluetooth with the Solar Powered Flower Sculpture is to connect wirelessly, range is a vital attribute of the Bluetooth capabilities. As previously stated different versions of Bluetooth allow for different lengths of communication. Starting from Bluetooth Version 1.x, the range of connectivity is increased. However, there is something to keep in mind. The range displayed for each of the different Bluetooth versions is simply the maximum supported range and in some cases the range of connectivity will be less. This is because range is determined by output power and there is no guarantee the Bluetooth module will be outputting using full power based on the system it is in.

Table 6: Bluetooth Module Comparison [45]

Class Number	Max Output Power (dBm)	Max Output Power (mW)	Max Output Range
Class 1	20 dBm	100 mW	100 m
Class 2	4 dBm	2.5 mW	10m
Class 3	0 dBm	1 mW	10cm

## 4.1.4 Microcontroller

In the emerging Internet of Things world nearly every device has some sort of microcontroller in it. With so many microcontrollers to choose from we started narrowed down our selection by asking how much processing power do we need, how many lines of communication do we need, do we want one with a FPGA drive for circuitry or can we find one that already does what we need it to do? Because of our experience level with microcontrollers we wanted to use processor with abundant documentation and library available. To simplify the software development process for the LED displays, it was also desired to get a board with PWM outputs. Since we are not a sponsored group cost was also a major factor that had to be considered. The microcontroller will take commands from the user and manage the functions of the solar statue. The microcontroller will have to be able to keep track of the time and managing LED displays of varying complexity. Throughout development the process the software for the processor will go through many iterations, so it needs to be able to handle many in field updates. The capability for the FPGA to update firmware either directly loading it or having the on-board storage device reprogrammed was also desired. Table 3.1.5a# list the desired specifications for the processor. These are the specifications we used to choose which microcontroller to use.

Table 7: Microcontroller Specifications (By Kibwe Williamson)

PWM outputs
USB connectivity
In field program updates
3v3 desired
C or C++ compiler
Low Power
Ample RAM
Ample pins connection
Table 3.1.5a: Desired Specification

**Arduino Uno: ATmega328P**

The Arduino Uno is Arduino’s flagship microcontroller. Six of its input/output pins can be used as PWM outputs, it has a USB connection, and 16 MHz quartz crystal oscillator. It has 32 kb of flash memory of which 0.5 kb is used by the bootloader, 2 kb of SRAM and 1 kb of EEPROM. Most of our requirements are met by the Arduino Uno. Six PMW outputs will not be enough so we will have to use an expansion module with this microcontroller. The online support for the Uno is plentiful. The Arduino Uno’s specifications are listed below in Table 3.1.5b



Table 8: Arduino Uno's Specifications (By Kibwe Williamson)

Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage(limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 kb used by bootloader
SRAM	2 KB
EEPROM	1 kb
Clock Speed	16 MHz
Programming language	C/C++
PC connection	USB
Table 3.1.5b: Arduino Uno Features and Specifications	

**Arduino Mega: ATmega1280**

The Arduino Mega is the Arduino Uno's big brother. Fourteen of its input/output pins can be used as PWM outputs, it also has a USB connection, and 16 MHz quartz crystal oscillator. It has 128 kb of flash memory of which 4 kb is used by the bootloader, 8 kb of SRAM and 4 kb of EEPROM. Most of our requirements are met by the Arduino Mega. Fourteen PMW outputs should be enough for our project so, we wouldn't need an expansion module with this microcontroller, but the Mega cost significantly more than the Uno. The Arduino Mega is almost perfect for our project but, the extra features we won't be making use of might not be worth the extra cost. The online support for the Mega is plentiful. The Arduino Mega's specifications are listed below in Table 3.1.5c

Table 9: Arduino Mega's Specifications (By Kibwe Williamson)

Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage(limit)	6-20V
Digital I/O Pins	54(of which 14 provide PWM output)
PWM Digital I/O Pins	14
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 kb used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Programming language	C/C++
PC connection	USB
Table 3.1.5c: Arduino Uno Features and Specifications	

**Texas Instruments MSP430:MSP430FR2532**

The TI MSP430 is one of TI's most popular MSP microcontroller. FRAM-based microcontrollers feature CapTIvate touch technology which is the highest resolution capacitive-touch solution on the market. The MSP430 features a low voltage operation range of 1.8 to 3.6 volts with a standby. This was the lowest operating voltage that we looked at, which is ideal for a solar powered system. The online support for the MSP430 is not as plentiful as the Arduino products. The TI MSP430's specifications are listed below in Table 3.1.5d

Table 10: MSP430's Specifications (By Kibwe Williamson)

Operating Voltage	1.8 – 3.6V
Standby Power	1.5 microAmps
Clock	16 MHz
Non-volatile Memory	8.5 KB
RAM	9.5 KB
Low Power	126 microAmps
CapTIvate touch technology	64 capacitive touch buttons
Input/output pins	16
Table 3.1.5d: TI MSP430 Features and Specifications	

### **Arduino Nano: Atmega328**

The nano is the smallest microcontroller that is available by Arduino. A fraction the size of its bigger brothers the nano is a complete board based on the ATmega328. It maintains the same functionality of the Arduino Duemilanove, but is arranged in a much smaller package. From the Duemilanove it only lacks the DC power jack and requires a Mini-B USB cable in place of the stand size USB cable. The advantage to this microcontroller is that it maintains all the functions we require for an Arduino in such a smaller package. This was important to us for the portability feature we wish to incorporate into our product.

Choosing a microcontroller was extremely difficult. We had to balance keeping a small size with the loss of processing power and memory, along with fewer features and outputs. To solve this issue we did some simple “napkin” design layouts and compared the potential complexity of each design. The designs with the bigger boards were fairly simple, pretty much everything was there on the board that we needed to be there and it was plug and play. The smaller boards required some finest and additional components sometimes. With the loss of processing power and memory space, the restraints this put on the possible programming languages that could be used to program them had to be considered also.

In the end we settled on the Arduino Nano, the smallest of the microcontrollers considered. The Arduino Nano was chosen because it meets all of requirements for the project with the exception of output pins, which could be satisfied with the use of a pin extension module. The hobbyist community that has developed

around the Arduino brand ensures that there will be support in the form of online documentation, tutorials, and forums in addition to manufacturer's documentation.

## **4.2 Software**

Research was taken for the development of the project's application (software) components, with considerations toward what group members were most familiar with and what best suited the project.

### **4.2.1 Existing Similar Software Projects**

The decision to use Android Studio to develop the Solar Powered Flower Sculpture project's application was beneficial for our group, as several members were familiar with the platform. However, researching existing software projects were helpful in providing reference points to how similar methods could be taken to develop applications. In no such cases were pieces of code ever plagiarized; merely, looking at existing projects gave inspiration and guidance for how to go about developing our own app. Two particular software projects were found online that showed enough similarities to our solar sculpture application to prove useful.

#### **4.2.1a Project 1**

A project was found on Instructables that involved the creation of an Android app that used Bluetooth connectivity to toggle an LED light. This project was very similar to ours in the sense that it relied on Bluetooth to transfer data to a microcontroller, and that the application was created with Android Studio. However, this project's goals are on a much more basic, smaller scale than that of the Solar Powered Flower Sculpture. Still, this project proved useful to developing our app. It provides clear explanation on how to successfully utilize a Bluetooth connection between our Arduino microcontroller and Android phone app. Most usefully, these project instructions are specified in detailed code, as opposed to just relying on Android Studio's widget displays. The interface used also shows a similar pop-up menu style designed for simplicity's sake (though these are default options for Android Studio development). Still, this project has far different end goals than our group's solar structure app. Only a single LED connected to a breadboard is manipulated, as opposed to several multi-colored LEDs, all with different purposes depending on the mode selected by the user. This project is much more useful for obtaining an idea of how to feasibly program Bluetooth connectivity for a microcontroller instruction, in parallel to a user-friendly mobile app.

#### **4.2.1b Project 2**

A second project also showed similarities to our project, though proved not as helpful for our project's software development as the first project used for

reference. PfordApp is an independently run platform that can be used for Android applications specifically on Arduino microprocessors (although support is offered for other types of microprocessors). Numerous project guides, complete with respective coding examples and pford libraries, show similarities to our group's software goals. This includes a project for altering certain components connected to an Arduino board's pins (such as multiple LEDs). These examples only prove useful in a sense, however, as pfordApp applications are not created with Android Studio, and can often be created completely using an interactive mobile display (without programming). For specification, code can be written to improve these projects, but the fact that it is not written within Android Studio means they are not especially relevant to the Solar Powered Flower Sculpture application. PfordApp projects prove useful in the sense that they have a wide array of Arduino-based functions with observable code, but in a different platform than the one our group is using.

## **4.2.2 User Control Interface**

Before group members ever programmed the project's Android application, its interface was designed completely on paper. This included the basic layout of having four accessible screens (one of which serves as a home screen), and a series of pop-up menus displaying more options when certain icons are selected. Having a clear plan of what the interface would look and operate like meant that there was a strong level of group agreement and understanding of what the app would be coded to look like. It also meant that user-friendly mobile app designs could be researched, and ideas could be implemented into the app's intended layout.

At one point, when our project only used an LED interface to operate as a clock, it was considered to have the user change the time by pressing different buttons connected straight to the microcontroller. The digital clock results would appear on a screen alongside these buttons. This idea was forgotten once the project was expanded to include LEDs acting under a lamp or disco mode, as well as functioning with a mobile app. When these decisions were agreed on, it became clear that there were many options for which platform the application could be written for, in what language, and how the app would connect with the sculpture's Arduino microcontroller.

### **4.2.2a Language Considerations**

Android was chosen somewhat early by the group as the platform of choice for developing the application. Once this was decided on, languages were considered to be determined of which would be the most appropriate one for writing the software with. One of our group members had experience developing an app with ActionScript, but it was not for Android devices, and no other group members knew the language. Other languages such as C++ and Python were considered, but group members similarly did not have much experience writing with these

languages. Ultimately it was decided to use Java for developing the mobile application. Not only did several members of the group already have experience writing with Java, but there was also little worry that a device used would be incompatible with the app, as Java is considered the official language of Android.

## **4.2.2b Platform Considerations**

Briefly, it was assumed that our solar sculpture could be compatible with both Android and iOS devices. As the application evolved to serve more purposes for the Solar Powered Flower Sculpture's LED display, the group decided to limit the application's functionality to Android devices. For one thing, as Java was already decided as the language to write the app in, it was noted that it was much more compatible for Android Studio than writing iOS-powered programs in Java. Also, some group members already had experience with Android Studio, while none of us had developed a fully functional app for iOS. Not as many helpful guides for Bluetooth connectivity between a mobile app and Arduino board were found written on iOS platforms, while significantly more were found that could be referenced and were based around Android Studio devices. Lastly, not all group members had devices that supported iOS apps, while all of us had Android devices that could potentially be used to test the app's functionality. Choosing the Android Studio platform was clear for our group project's goals.

## **4.2.2c Communication Considerations**

Communication between the mobile app and Arduino microcontroller was assumed to be through Bluetooth early on during project considerations; however, other methods were discussed. At very early stages, it was considered to have some functions be operated by the user pressing buttons on the microcontroller itself (appearing through the exterior of the structure's base), and other functions be based on the mobile app. When the project's functionality supported more than just a clock interface, all user input was decided to be sent to the Arduino board from the mobile app. Simple Wi-Fi was ruled out as an option because it was thought that maintaining the signal would be too needlessly complicated (let alone coding the connectivity). Cellular 3G and Radio 433MHz were similarly decided against. Bluetooth proved to be the best option as an Arduino attachment was directly compatible with the protocol. Also, all Android devices owned by group members supported Bluetooth connection.

## **4.2.3 Hardware Programming Language Considerations**

There are a few choices available for programming hardware. The languages that were considered were the lower lever Assembly language and the higher lever C and C++ languages. The programming language will be chosen based on the

programmer's knowledge of the languages and the approach that will be taking to develop the software.

### **4.2.3a Assembly Language**

Assembly language is also known as machine language. This is because it has a straight connection to machine code. There aren't any extra features like loops used for generating assembly code. So, the processor only has to execute what is necessary to perform the given task. The simplicity of the code also means that it takes up less memory than the higher languages. So, if memory is limited then Assembly code will be the programming language of choice. The debugging process can also be easier with shorter programs. This is often the case due to entire functions fitting on one screen, instead of having to trace a functions logic while scrolling through a screen. Assemble codes have a faster execution time for these same reasons. This makes Assembly the better choice when dealing with time critical applications. If the application is not time critical, then a higher language can be considered. For some programmer's assembly is harder to work with because the language doesn't readily translate into written language like higher level programming languages. Function name are more abstract, so can be less intuitive for a programmer to work with. Another feature of Assembly language that makes it more difficult for some programmers to work with, is that it requires an understanding of the microcontrollers hardware functions. Without an understanding of the hardware level of the specific chip you are working with Assembly language isn't a viable option. Knowledge of the chips hardware gives the programmer the option of using special feature specific for a chip that higher level languages don't give you access to. Programs written in Assembly language are work as you go programs. The chip will execute anything you tell it to do. This allows for warning and acknowledgments to be given as you program so you know whether exceptions should be considered. Debugging basically works the same as it does in any other language.

Use of this programming language was strongly debated. None of the team members were particularly strong with this programming languages. One of the primary features of the solar statue was its ability to keep time, which made this language a strong candidate despite the reluctance to use it from the team members. Selection of this language would depend on the ability of the software developer to get a reliable clocking working with a higher level language. This outcome isn't likely given the wealth of the libraries available for the higher level languages, but was still a concern of the software developer the entire time spent trying to get a reliable clock working with a higher level programming language.

### **4.2.3b C Language**

C is considered a high level language. It has a few advantages as an embedded systems language. It's usually the first programming language a programmer learns so it's usually a very familiar language. It's simple, easy to use and

understand. There are a lot of good compilers available for this language. Because it is a foundation language there is a wealth of knowledge and resources online for programming with it. Code written in see doesn't have to be tailored for a specific processor. It's relatively easy to manage I/O access for embedded projects.

C is higher than Assembly but lower than Object Orientated programming. This makes C more of a middle language with some of the benefits of lower level languages like direct hardware control. It might take more coding to duplicate some of the functions of higher level languages, but this also give it more flexibility than the higher level languages. It even supports low level bit-wise data manipulation.

The advantage that makes it more desirable than Assembly for most programmers is its reliability and scalability across platforms. Since C uses more natural language it is easier to understand, debug and maintain. C code can also be converted into assembly. So, if Assembly code is needed ultimately C gives the option of written code in a language that is easier to read and write in while allowing for efficient conversion to Assembly code.

The software developer had a great deal of confidence in his ability to get the device working using this language. Most of his programming experience is in this language so here is where his programming background was the strongest. Being a computer engineer major and going through the computer science one course, he felt confident that some of his assignments would prove to be more challenging than the algorithms he would have to write for this project. The hard part would be adjusting for embedded programming, instead of simply writing algorithms without any memory or processing constraints. This was the favorite of the software developer going into the research of which language would best fit this project.

### **4.2.3c C++ Language**

The final programming language being considered is the object orientated version of the C programming language. This is the highest level programming language of the three, and the last type of language that software developers learn how to use in school. Although this language has a lot of powerful features being the last language that a student has to demonstrate proficiency in makes it less desirable than the C language. Being an Object Oriented language C++ features will be helpful when implementing monitors. The major concern with this language is the performance variances across compilers for the language. This variance is a result of different standards being used by different vendors and open source compilers. The libraries for C++ are very large and complex, this is a double edged sword. Having a large and complex library may make the programming process easier, but it also increases the size the program will take up in memory. There is an ongoing debate in the C++ community about which features should be used and what to avoid in embedded programming.



Being the “Object Orientate version” of the C language, it is very familiar to C programmers. This reduces the learning curve for C programmers. Just about anything that can be done in the C language can also be done in the C++ language. If desired C code can be migrated to C++.

Most of the features of C++ don't add any cost at runtime. This is due to these features being front-end issues. Front-end issues don't have an effect on code generation. When no arguments are specified by the programmer the compiler will insert defaults for the programmer. Overloading is a feature of Object Orienting programming allows for functions to have the same name but different arguments. The arguments used during the function call determines which function will be called and executed.

Another major difference is that C++ doesn't use pointers, C++ uses references instead. References are safer than pointers because a reference can't be null. You also cannot uninitialized a reference or change it to point to something else.

Another feature of Object Orientated programming is the use of classes, methods and objects. The classes and their member functions are usually poorly understood and executed. This is a difficult concept to teach, so programmers usually take some time before a programmer gets proficient with it. Methods are pretty much the same as functions, except they can be used to generate objects. In Object Orientated programming there are not structures. Instead classes are used with public members. A member of a class is just a function that has a pointer to an object of its class. The object of the class is an implicit parameter.

In Object Orientated programming there is a lot of control over access to different portions of the code. Different parts of the code can be declared as public, private or protected. These compiler restrictions prevent misuse of data from other portions of code and interfaces.

This programming language held a great deal of interest for the software developer because he had recently started to work more with Object Orientated programming. His tabling with video game design on the side had gotten him to a point where he felt fairly confidence that he could complete the software in an Object Orientated language is a reasonable amount of time and in an efficient manner.

## 5.0 Design

These sections detail all of the components of the Solar Powered Flower Sculpture in great detail, subject to the design phase of the project. This includes both the hardware and software components in comprehensive fashion.

### 5.1 Design Method

We decided to use the Agile process for issue and problem solving during the research and development of this project. The process of laying out the design first and then assigning task for completion seemed to be our best first for this project. This allowed us all to monitor any changes made by any of the group members and to made record keeping easy. We utilized the simplest process of the Agile Scrum cycle, doing our test first, listing any errors that were found, correcting the errors, and redoing the test.

During each cycle we designated task to be completed and tried to have them done by the next Scrum Sprint. This allowed us to work on the multiple subsystems of the project simultaneously, while consistently working towards the overall concept of the project. Focusing on the small parts while keeping the picture in mind worked well for us. This allowed us to maintain small baselines of progress each week.

### 5.2 Hardware

To begin designing the hardware for the Solar Powered Flower Sculpture, the group first had to find a way into which efficiently connect the photovoltaic cells, the nine voltage nickel metal hydride rechargeable battery, and the Arduino Nano microcontroller. To start powering the Solar Powered Flower Sculpture, the group needed to choose photovoltaic cells that would supply enough power. This was accomplished by series stacking six photovoltaic cells together. Since solar power is not efficient on its own to pull in a constant voltage, a device in which will stabilize the voltage and current is needed. This device is called a charge controller. Most solar powered devices are created with connections to a charge controller since it will take in a varying voltage and supply a constant, unwavering voltage. This voltage can then be used to charge the nine voltage nickel metal hydride rechargeable battery once it drops low enough in voltage to trip the charge. The next step to think about is that the Arduino Nano microcontroller needs a constant five volts to operate, but no more than six volts. To supply this voltage from the nine voltage nickel metal hydride rechargeable battery a step down switching regulator is needed. With this switching regulator, the Arduino Nano microcontroller and the Atmega328 microchip will safely be delivered five volts. From here, since the Atmega328 microchip only has six pulse width modulation

output pins and the Solar Powered Flower Sculpture needs twelve to control the NeoPixel RGBW LED strips, a LED driver called TLC5490 will be implemented.

## **5.2.1 Power Supply**

For this Solar Powered Solar Sculpture project, there are multiple power supplies. Firstly, there is the power drawn from the photovoltaic cells. These cells were arranged in a way that allows voltages from each photovoltaic cell would combine in series. These photovoltaic cells combined in series then supply the power to the MAX712 charge controller. This MAX712 charge controller is able to regulate the photovoltaic cells power into a constant supply rate. This rate then sends power to charge the nine voltage nickel metal hydride rechargeable battery. This battery then becomes the second power supply source for the Solar Powered Flower Sculpture. With this rechargeable battery having a constant charging source, it is able to then supply power to the MAX639 switching voltage regulator. This switching voltage regulator then maintains a constant five voltage that will be able to supply the ATmega328 microchip. This constant five volts is then transmitted to the TLC5940 and the twelve individual LED strips. In the end, the unregulated voltage that the photovoltaic cells absorbs is transmitted to powering the twelve individual LED strips.

### **5.2.1a PCB Design**

After spending time learning how to manipulate designs in EAGLE, the group was able to create an initial printed circuit board schematic.

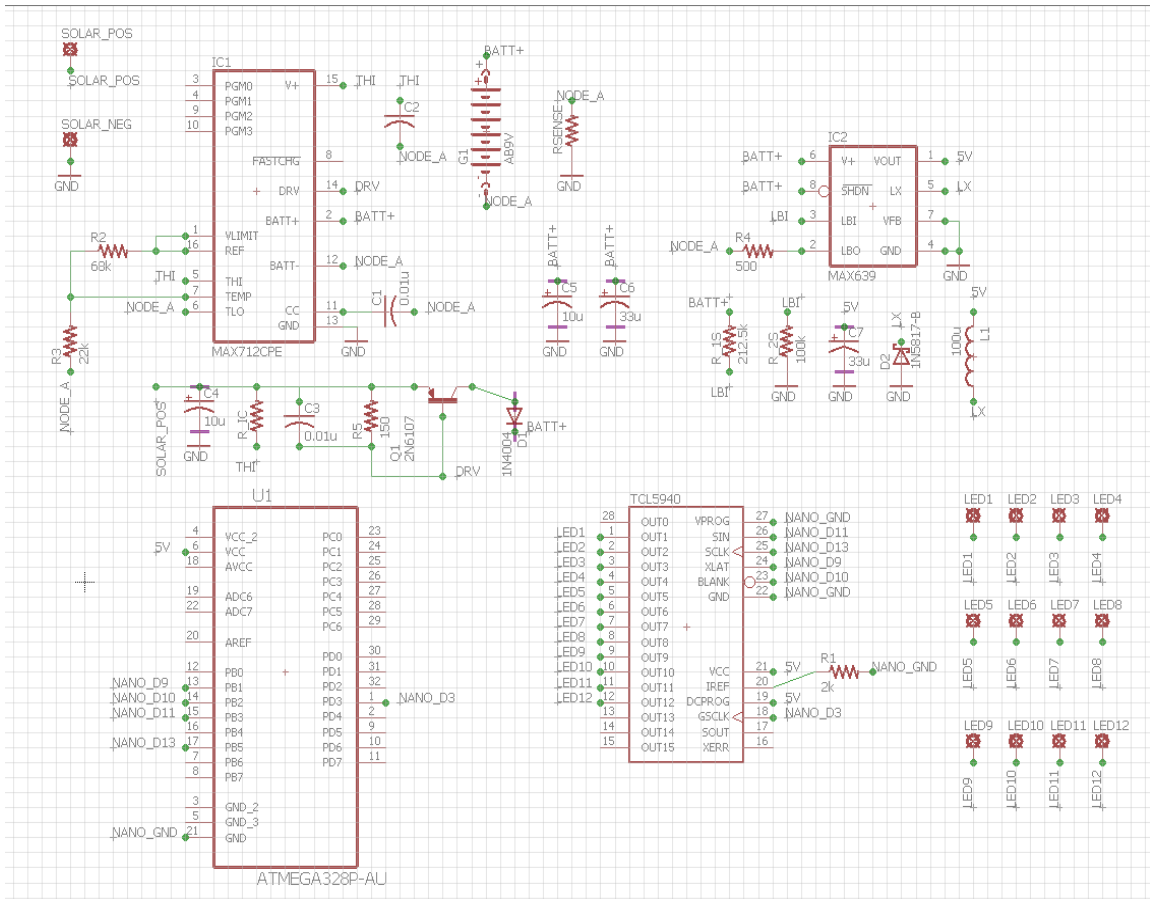


Figure 19: PCB Schematic using EAGLE

Now that the schematic has been created in EAGLE, the PCB design can be generated.

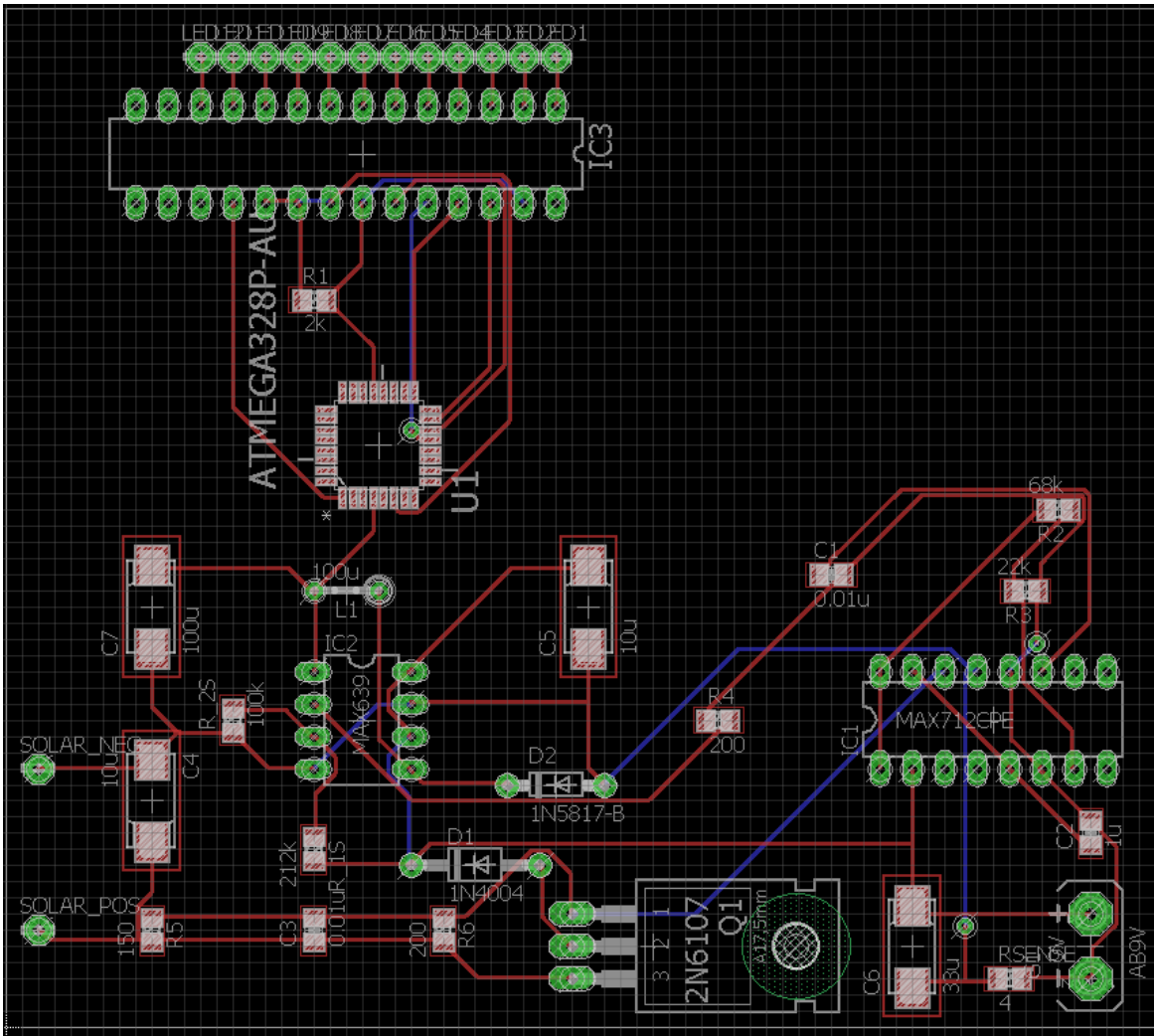


Figure 20: PCB Design

## 5.2.1b Battery

The group decided to go with the nine volt nickel metal hydride rechargeable battery to power the Solar Powered Flower Sculpture. The battery purchased looks as follows:



Figure 21: 9 V NiMH

## 5.2.2 Microcontroller Interface

The microcontroller will be the brains of the project, it needs to be able to both receive and send data from many different components. It will have to be interfaced with the LED driver, the battery, the Bluetooth shield. It also needs to be able to respond to command send from the user via the mobile app.

The Arduino Nano has 14 digital pins that can be used for either input or output. Communication with these pins is accomplished with the functions `pinMode ()`, `digitalWrite ()`, and `digitalRead ()`. The voltage for the signal coming into and out of the pins will have to be kept to 5 volts as that is what the pins are rated to. The max amperage for each pin is 40 milliamps, there is an internal resistor of twenty to fifty kilo ohms. In addition to general input and output the Arduino Nano have some pins that have special functions.

The serial zero and one pins are the (RX) and (TX) pins respectively. These are the receiver and transmitter pins. They are used to transmit and receive TTL serial

data. The receiver and transmitter pins are connected to the FTDI USB-to-TTL Serial chip.

This board also comes prepared to handle external interrupts. Pins two and three handle the external interrupts. They can be configured to be triggered on specific interrupts. They can be configured for low values, or rising edges, and falling edges, or simply a change in value. The `attachInterrupt ()` function can be used to handle these. The use of interrupts will help us solve any timing problems that may arise and are useful when setting up things to happen automatically on a microcontroller. Any input from the user via the mobile app will be monitored for with this feature. The interrupt service routine is like a special function that shouldn't be used to return anything.

The pulse width module pins are pins three through eleven. The `analogWrite ()` function will be used with these pins. This is so that we can write an analog value instead of a digital value. This gives us the option of operating the LEDs at specific brightness levels. This is accomplished by generating a steady square wave after a call to `analogWrite ()` function. The square wave will be maintained for the specified duty cycle until the next call to the `analogWrite ()` function. A call to the `digitalRead ()` or the `digitalWrite ()` function will also stop the maintaining of the square wave. These pins will be running at approximately 490 Hz.

Something that we will have to be mindful of is that the PWM outputs across all the pins won't be equal. The outputs on pins five and six will be higher than expected. This is the result of interactions between the `delay ()` function and the `millis ()` function. They share the same internal timer as the PWM outputs. This effect is mostly noticed on the low duty-cycle, where the LEDs connected to pins 5 and 6 might not completely turn off.

SPI communication are carried out through SPI pins 11 through 13. We won't be utilizing SPI communication in this project.

The microcontroller provides UART TTL (5V) for serial communication. Communication with a computer during programming will be done over a USB cable.

## 5.2.2a Interfacing with LEDs

Conceptually in the simplest sense an LED is just a two terminal device. This allows it to be characterized by two quantities: the current running through the device and its light output. The LED's light output is proportionally linear to its current either in milliwatts or photons per second. Because of this relationship between the current and light output it is helpful to think of LEDs as current-operated devices.

We can create a voltage – current graph to characterize a particular LED. This can be done by applying a particular voltage to the LED and using an ammeter to

record the current running through it. Completing a series of reading at staggered voltages will produce data points that can be plotted to produce a V-I graph.

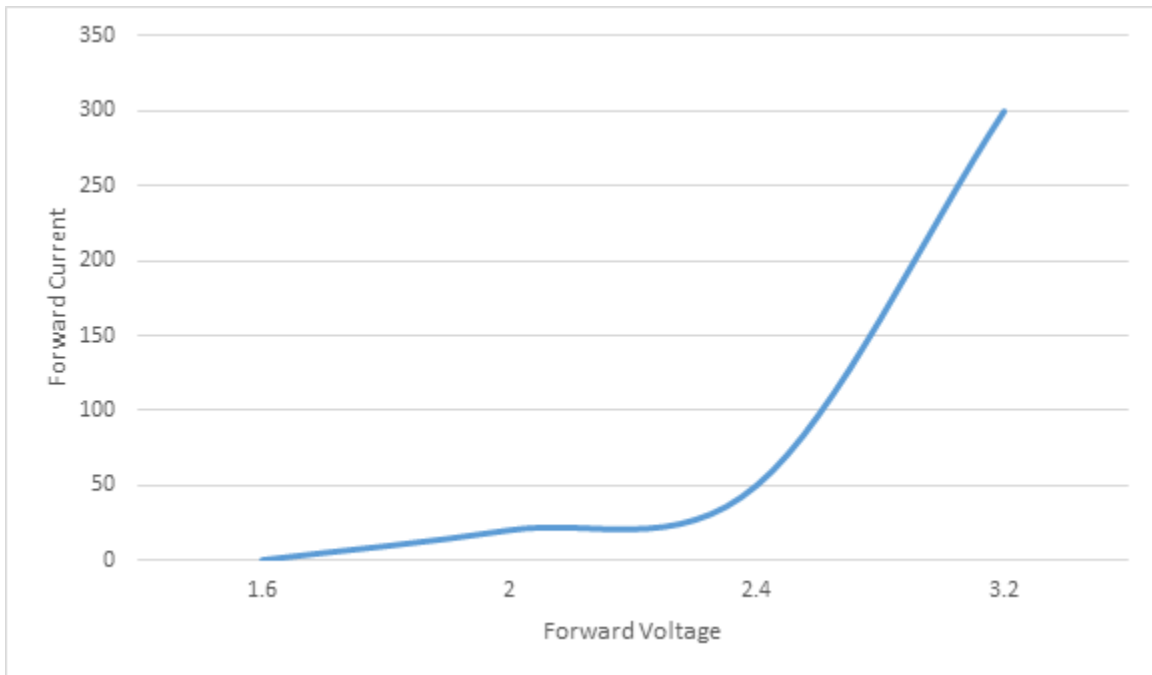


Figure 22: Voltage-Current Graph of LED (By Kibwe Williamson)

The relationship between voltage and current is a steep curve not a straight line. The curve is steep at the range of currents we will be operating on. The current has to more than double to get a twenty percent increase in voltage. This tells us that the LED has a high level of sensitivity to voltage.

Likewise, the opposite can be said about the relationship between the change in voltage and the current running through it. What this means to us is that we can model the LED's voltage as a constant and just deal with the voltage as a good approximation.

We could do a series combination of a resistor and an ideal diode if we wanted a more accurate model of the LED's V-I. The nonlinear characteristic of the curve is a result of the exponential V-I characteristic of the ideal diode. Also  $V = IR$  is dependent on  $I$ . When  $I$  is small,  $V$  is also small and when  $I$  is large,  $V$  is also large. This is the source of the curve, looks like a straight line at the higher voltages.



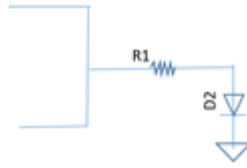


Figure 23: Adding Resistor before LED to Increase Current (By Kibwe Williamson)

Hooking the LEDs up required three connections, one for the power supply, one for the ground and one to carry the data from the Arduino. A connection of 5V was required to power the LEDs.

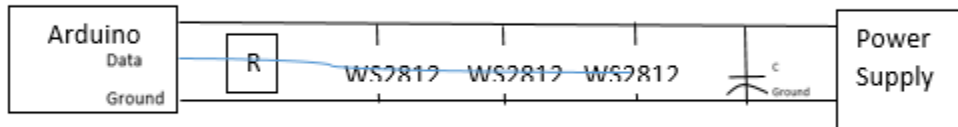


Figure 24: How the LEDs are hooked up to the Arduino. (By Kibwe Williamson)

The voltage drop for each strip could vary slightly by about 0.6 volts. This means that V-I characteristics of each individual strip could be anywhere on or between the two curves the line shown below.

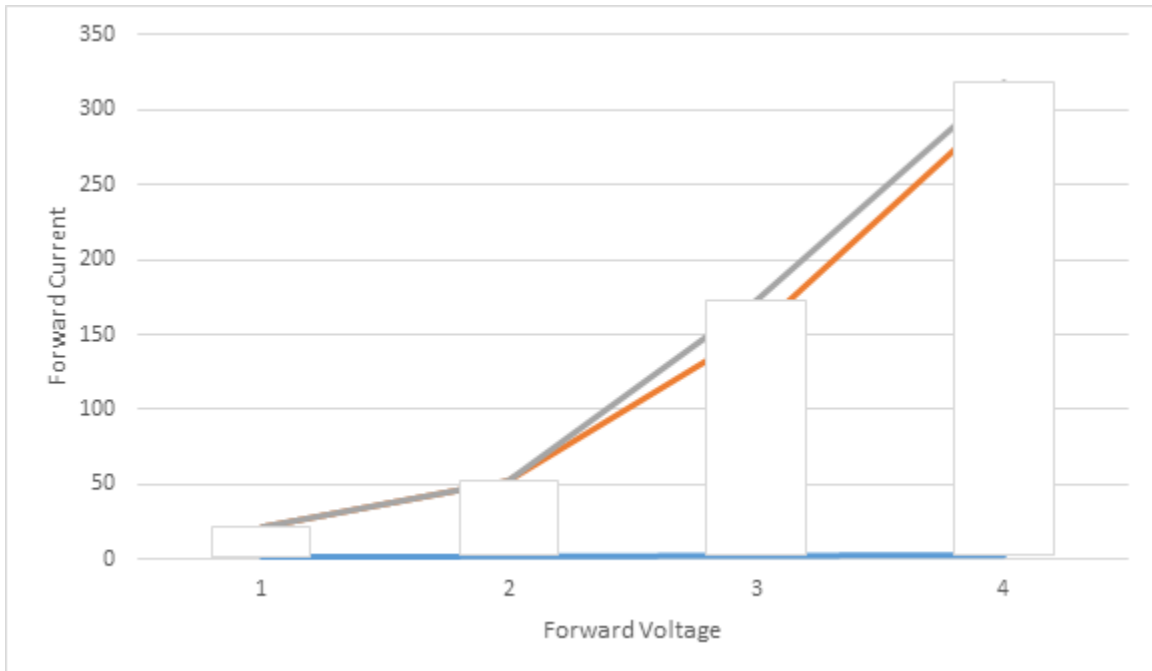


Figure 25: Voltage-Current Characteristics (By Kibwe Williamson)

This could cause a noticeable difference between the displays on each individual strip. We could minimize this effect by running off of a large voltage, decreasing the current sensitivity of the LEDs.

## 5.2.2b Pulse Width Modulation

The amount of current received by a LED during a sample period determines its brightness. Using Pulse Width Modulation is one way of controlling the average current per cycle that a LED receives. This is accomplished by varying the width of the pulse. Simulating an analog signal, we can control how much time a PWM digital signal is in high. Since a signal can only be either high or low at any moment, we can manipulate our display by changing the proportion of time the signal is in high versus low over a specific interval of time. Using this feature allowed us to simulate more colors than the red, green, blue and white that our LEDs came in. Our research indicated that the best way for us to get the type of displays we wanted was to use a microcontroller that featured PWM output.

## 5.2.3 Photovoltaic Cells

For the photovoltaic cells, the group had to decide to go with one large photovoltaic cell or to use multiple smaller photovoltaic cells. With the requirements specifications stating that the Solar Powered Flower Sculpture needs to be portable, it would not make sense to use one large photovoltaic cells. This being said, the group chose to series stack multiple photovoltaic cells. The photovoltaic cells chosen each have a maximum voltage rating of 4.5 volts, and the group decides to series stack six photovoltaic cells. This would give an overall maximum rating of 27 volts that could be delivered.

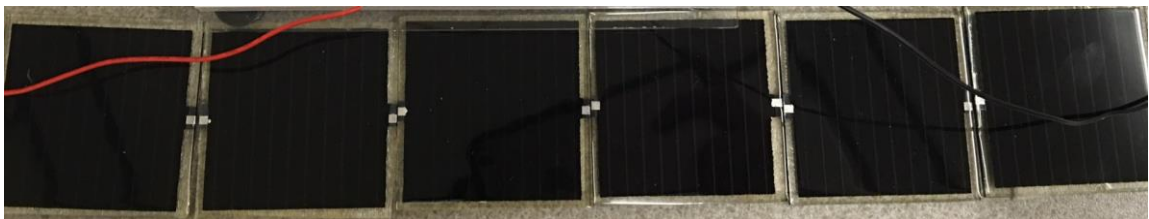


Figure 26: Photovoltaic Cells in Series

## 5.2.4 LED Circuitry

There are twelve individual NeoPixel RGBW LED strips that will be used for the Solar Powered Flower Sculpture. Each of these twelve LED strips needs to be connected to an input/output pin that contains pulse width modulation. Due to the ATmega328 not containing enough pulse width modulation input/output pins, an LED driver has to be integrated. This LED driver, the TLC5940, now allows up to

sixteen pulse width modulation channels, and is directly connected to the ATmega328 input/output pins. However, due to the TLC5940 being a current sinker, the LED strip circuitry will need the ground lead to connect to the TLC5940 output pins and the positive lead to the positive five voltage supply of the ATmega328 microchip. This design appears as follows:

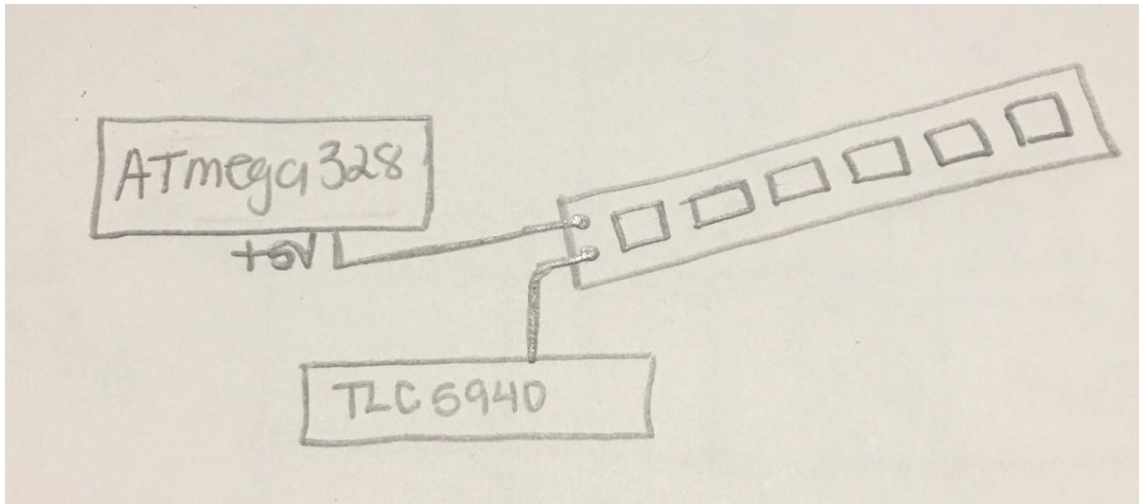


Figure 27: LED Strip Connections

## 5.3 Software

The software section details a thorough walkthrough of all components and operations of the Solar Powered Flower Sculpture's mobile application, Arduino microcontroller, and how they interact to enable functionality of the sculpture's LED display.

### 5.3.1 Application

As mentioned in earlier sections, the Android application is composed of four accessible, user-friendly display screens: a home screen, a clock mode screen, a lamp mode screen, and a party mode screen. The home screen ultimately serves as a hub for accessing the other three mode screens, as well as serving as a returning point (given from the user selecting the back icon) for any other point of the app. The home screen also allows the user to power on and off the system. The clock mode screen, upon opening, has the sculpture operate as a working clock system. From this screen, the time can be set and modified, and brightness and colors can be changed. The lamp mode screen changes the sculpture to act as an LED-powered lamp, with brightness and color options reachable on the menu. The party mode screen has the sculpture operate as a changing lights display, and the user can change between different patterns listed as well as manipulate brightness settings.

The reasons for our group implementing a mobile application into the project were both functional and aesthetic. For the clock component of the Solar Powered Flower Sculpture, having the user enter the desired time is a lot more customizable, less time-consuming, and less frustrating than manually entering and modifying the time on the sculpture itself. It also makes sense to develop a mobile application due to the numerous options users have for manipulating the sculpture. Between changing LED colors, adjusting brightness, and selecting different operating modes, using a different approach (such as installing buttons on the sculpture's base for accomplishing these functions) is not nearly as feasible. Overall, having a mobile app connects with our project's goal of delivering a satisfying user experience people of all sorts can connect with and enjoy. The application provides several different alternatives users can take, depending on the circumstances of the situation required. Rather than alienating people, having a mobile app display serves as an immediate source of recognition among users. The addition makes our group's desired goal of personal connection all that closer to achieve.

### **5.3.1a Platform**

Developed for Android mobile devices, our team decided to have the software component for our project utilize an application designed in the Android Studio Integrated Development Environment. Some members were already familiar with the Android Studio IDE, and are in possession of devices recent enough to run apps and programs designed for the project (installed with operating systems equivalent or exceeding Android 5.0 - "Lollipop"). Our team's programmers are also experienced in Java, and since Java is the official language of the Android Software Development Kit platform, that experience will come in handy as programs are being written with that language. Several functions the application is expected to perform will rely on imported Java classes. An example is when the Solar Powered Flower Sculpture is set to party mode, and the LEDs will cycle through a uniquely colored pattern loop. Depending on which pattern is selected, an array list being used to store a sequence of values that correspond to specific LEDs on the sculpture may be sent to the Arduino. The microcontroller would translate this request by having the LEDs illuminate in the order specified, on a continuous loop until an interrupt is received, (such as the user selecting a different pattern to be executed, the sculpture changing to a different mode of operation, of the system being powered off).

Our group plans to make use of several SDKs that will be downloaded to provide more varied resources at our disposal, for implementing the project software. In addition to building and running the software, the Android Studio platform is also intended to be used for debugging the written code, at various stages of the development cycle and testing plans. The mobile app will primarily act as a means of transmitting instructions to the Arduino board. Most operations will run on the microcontroller itself. There are a few exceptions, however. For the clock mode of the project, the movement of the LEDs simulating the "clock face" of the

sculpture, will be committed by the Arduino deciding which lights will illuminate in which order. Despite this, the mobile application will still update the time set by the user, so that as the sculpture will operate with respect to time, so will the app. In addition, if the sculpture were set to a different operation mode, and then set back to clock mode, the time previously set would need to be remembered by the application, so that information could be accurately conveyed back to the microcontroller receiver. This would obviously require some functionality with the app to maintain a running operation. Still, most of the possible Android interactions will only need to send data to the sculpture. From there, the microcontroller can determine what to do with the LED display. A more detailed analysis, and breakdown of what the application is capable of, is explained in Section 5.3.1b below. For observational reference purposes, a utility menu diagram, navigating the menu functions of the application, can be seen in Figure # below.

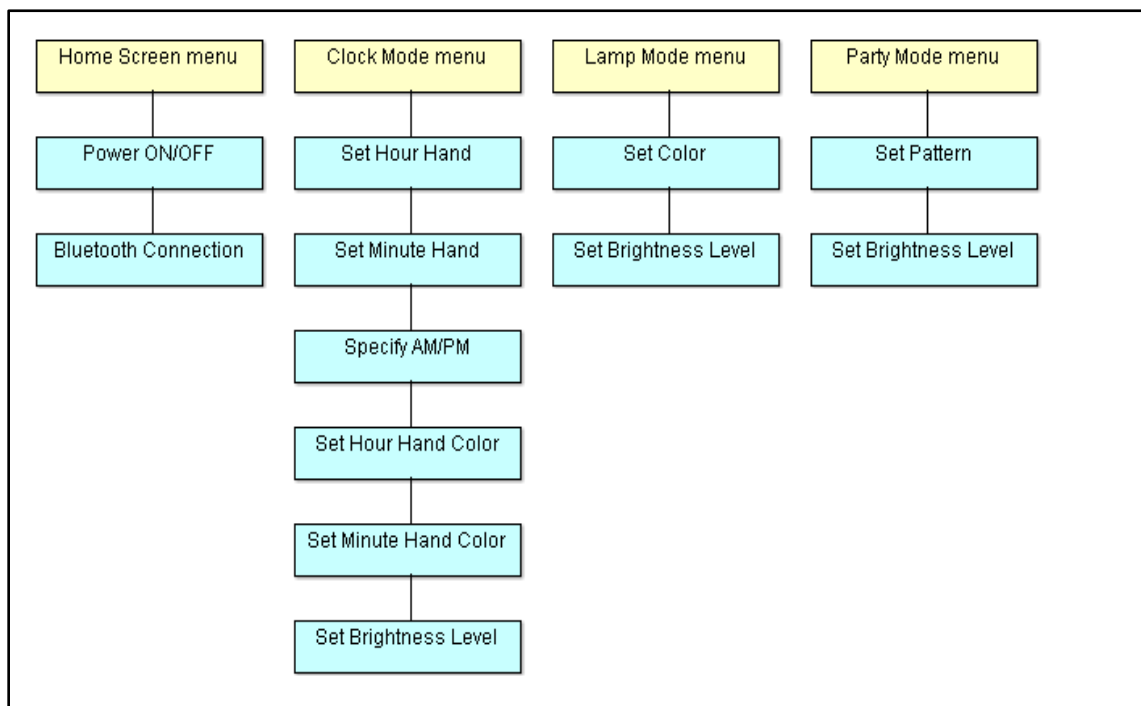


Figure 28: Application Utility Menu Structure (made by Nickolas DeVito)

### 5.3.1b Android Interface

By traversing the four menus of the mobile app, users will be able to accomplish all allowed operations through categorized buttons and pop-up menus. The expectation our group had while developing the project was that the application would support enough user-friendly qualities to ensure that users could open the app, complete all desired operations, and navigate all menus, without having used the app before, or being given any prior information on how to traverse it. This display is designed and intended for ease of use, and the forgiveness design principle is a key feature throughout. At any point from inside a mode menu

screen, the user can return to the home screen. Additionally, when a pop-up menu is opened, the user can exit by clicking anywhere outside the menu, returning to the screen as it appeared before the menu was brought up. At some points in the app, a second pop-up menu may be opened inside of an original pop-up menu. Clicking outside of the second pop-up menu will return to the first, at which point the user can click outside of this pop-up menu to return to the screen. This same touch mechanic is how operations from inside of pop-up menus are completed. In other words, if a user makes a change inside of a pop-up menu, clicking outside of the menu will execute that change.

Since users are expected to complete a large amount of operations in the mobile app, having in-app notifications for completing every action would become redundant and distracting. Therefore, notifications will only pop-up after a desired action has been confirmed, by clicking out of the menu. As an example, if confirmation notifications appeared for every time a user might cycle through the minute hand options, while trying to set the time for the app's clock mode, it would be distracting; however, after a user has selected the desired number, and all other options for the time have been decided on (besides the minute hand number, the user must select the hour hand number and AM/PM specification), now a notification may pop-up telling the user that time has been set. For further ensuring an action has been completed correctly, users can observe if their changes appear inside of the various pop-up menu sections in the app after pop-up menus have been closed, and by observing the statue itself for observable modifications.

Actions are also taken to ensure how operations are saved in the app. Whenever a menu is navigated away from, or the app itself is closed, the mode which the sculpture was last set to will be preserved, and all sections under this mode will retain the last changes made. Changes made to other modes will also be saved, although they will not be selected by the sculpture (as an example, if the hour hand for the clock mode of the sculpture was last set to the color red, and the color for the lamp mode was last set to yellow, both colors will be saved for their respective modes; however, only the mode which was last selected will be running on the sculpture). If the sculpture is powered off, settings in the app will still be saved as long as it is running. For the case where the app itself is opened up upon restarting, the default settings will be the first option available for every pop-up menu option. At no point will the Android application require users to type in information in order to obtain results. Instead, all direct manipulation operations rely on selections that the user can choose depending on the circumstances of the operation they wish to complete.

The home screen is essentially a multiple-item menu that displays four radio buttons, titled "CLOCK", "LAMP", "PARTY", and "POWER". The first three buttons change the mode of the solar sculpture and bring the user to new menu screens for those options. The power button opens a popup menu adjacent to button's location, that shows a single, binary slider (switch) with two options labeled "ON" and "OFF" for the sculpture's power function. Bluetooth connection may also be established from the home menu screen of the application (either as an icon on

the top of the app screen or by selecting the options button on the phone itself). As the phone app is only intended to connect to the Bluetooth transceiver on the Arduino microcontroller, (although Bluetooth connection to a computer is likely during testing), creating a complex menu for wireless Bluetooth options may not be necessary. Still, it is worth having some reliable area of the app where Bluetooth connectivity is addressed, such as during cases where connection may be lost and Bluetooth needs to be re-established. A preliminary estimation design of what the home screen for the application could appear like can be seen in Figure # below.

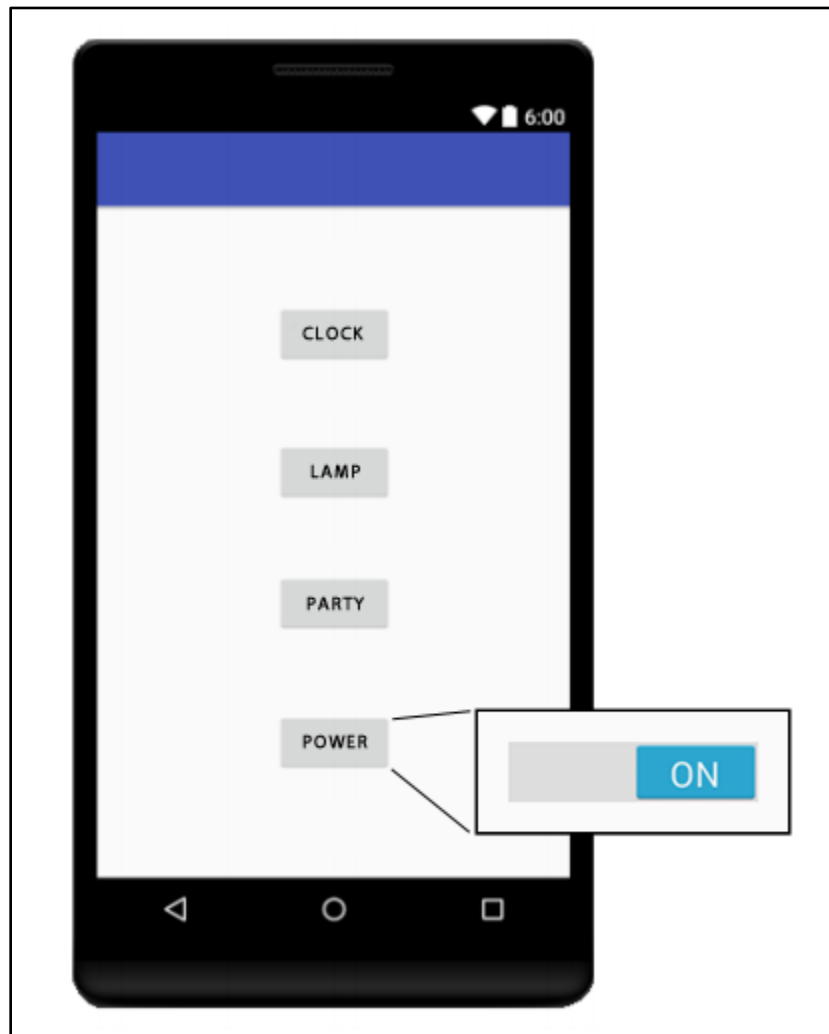


Figure 29: Application Home Screen Design (made by Nickolas DeVito)

The clock mode screen displays three radio buttons, similar in layout to the home screen, titled “SET TIME”, “COLOR”, and “BRIGHTNESS”. There is also a clickable button in the upper left corner of the screen, with a left-pointing arrow icon, titled “BACK”. In self-explanatory fashion, clicking this icon returns the user to the home screen. Returning to the clock mode screen buttons, clicking each

one opens a unique pop-up menu next to the respective options. The pop-up menu for setting time shows, in linear order: a button for the number that will be used as the hour place of the desired clock time (labeled "HOUR"), a permanent colon icon for traditional reference, a button for the number that will be used as the minute place of the desired clock time (labeled "MIN"), and a button for specifying AM or PM (labeled "AM/PM"). Clicking the hour place button opens an additional pop-up menu; a fixed vertical slider list with a range of numerical values from 1 through 12. The minute place opens a similar pop-up menu featuring a fixed slider list, but with a range of numerical values from 00 through 59 (two digits are emphasized due to the clock presentation). The AM/PM button opens a popup binary slider switch, with two list options appropriately labeled "AM" and "PM". Now, the pop-up menu for setting color lists two buttons titled "HOUR" and "MIN" from left to right, which specify the colors that will be conveyed by the hour hand and minute hand on the sculpture's LED clock face. These buttons bring up identical, fixed vertical slider lists featuring the series of colors the user can choose the LED to display. Having the user select the same color for both hands is allowed. Lastly, the brightness pop-up menu shows a single two-digit numerical value followed by a permanent percent sign icon. Underneath the percent value is a dynamic pop-up horizontal alpha slider, with a visible cursor, that allows the user to slide from left to right, changing the above numerical value from 1 through 100 in the process. A preliminary estimation design of what the clock mode screen for the application could appear like can be seen in Figure # below.



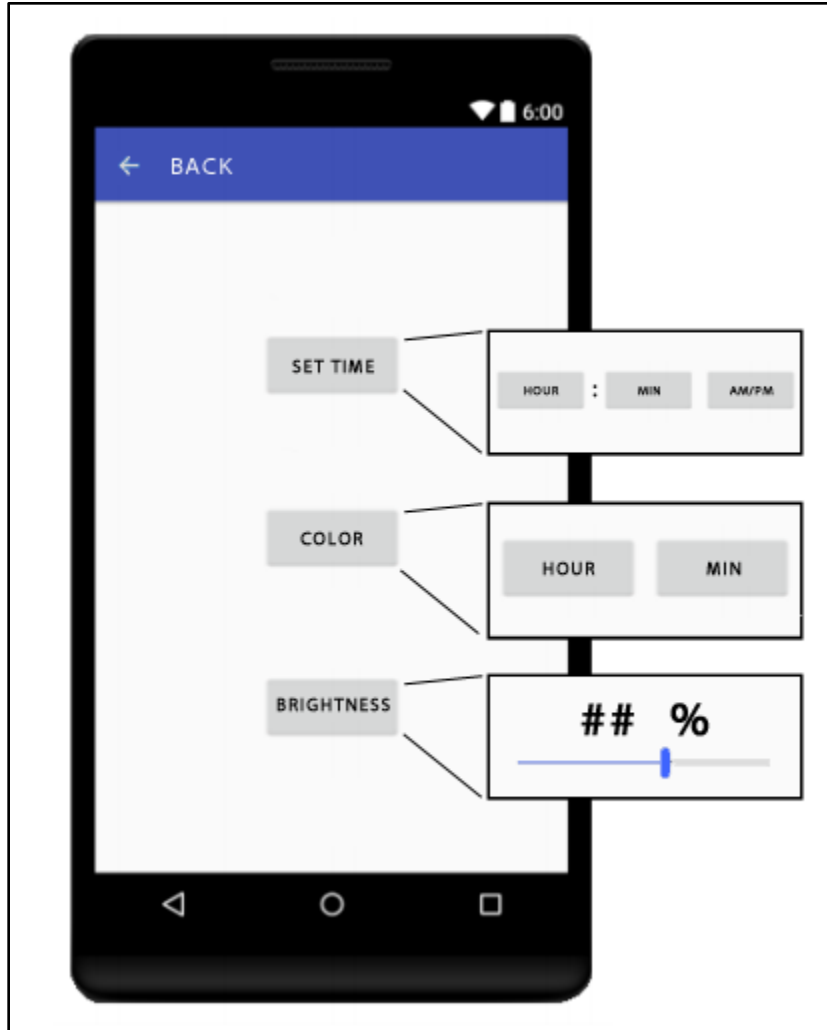


Figure 30: Application Clock Mode Screen Design (made by Nickolas DeVito)

The lamp mode screen displays two radio buttons titled “COLOR” and “BRIGHTNESS”. A back button, featuring an arrow icon in the upper left corner, is also present, and operates in the exact same way as on the clock mode screen. Clicking the color button opens a popup menu for a fixed vertical slider list that immediately lists all supported LED colors for the sculpture to take. The brightness button brings up a pop-up menu identical to the brightness button found on the clock mode screen. A preliminary estimation design of what the lamp mode screen for the application could appear like can be seen in Figure # below.

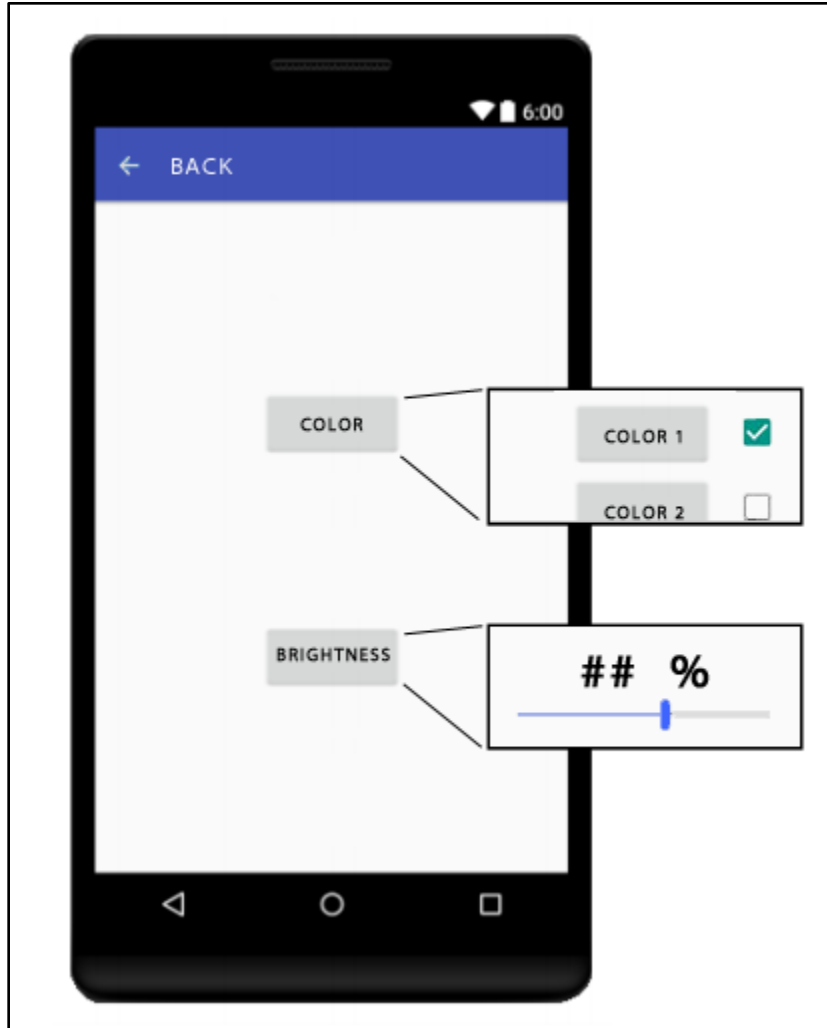


Figure 31: Application Lamp Mode Screen Design (made by Nickolas DeVito)

The party mode screen displays two radio buttons titled “PATTERN” and “BRIGHTNESS”. As with the clock and lamp mode screens, the party mode screen also contains a back button, identified with an arrow icon, in the upper left corner of the screen that operates the same way. Clicking the pattern button brings up a pop-up menu for a fixed vertical slider list, featuring all created patterns that the sculpture can repeat; these patterns are listed in numerical order (they are also numbered themselves). The brightness button brings up a pop-up menu that works in the same way as the option found on the clock mode and lamp mode screens. A preliminary estimation design of what the party mode screen for the application could appear like can be seen in Figure # below.

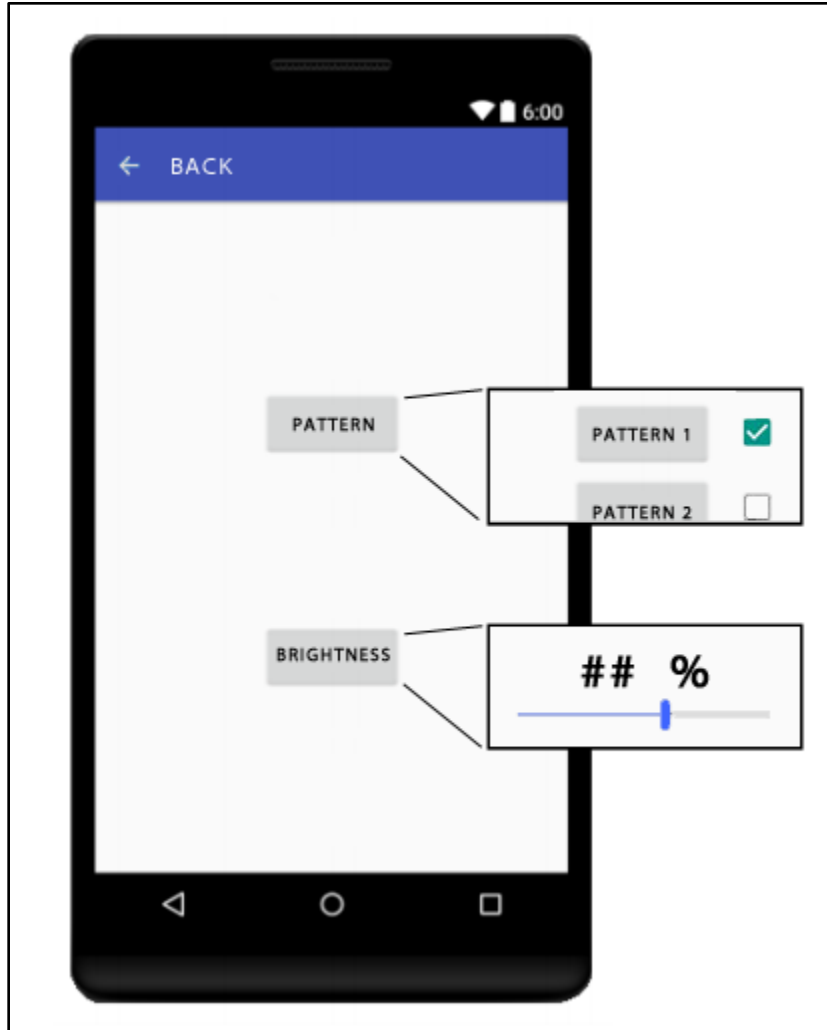


Figure 32: Application Party Mode Screen Design (made by Nickolas DeVito)

## 5.3.2 Microcontroller

The Arduino nano has a limit to how much current can be sunk by its I/O pins. To avoid killing the board we had to take this into account when interfacing the board with the other components.

Since we will be using addressable LEDs, we will be making use of the Fast Arduino Bitbang LED library or FAB\_LED. The FAB\_LED library provides powerful and easily implemented programming for both RGB and RGBW LEDs, on both AVR and ARM platforms. The library is compatible with a variety of models from the ws2812 to the apa102 to the sk612. The FAB\_LED library allowed us to manipulate individuals or arrays of pixels, declare which ports the LEDs were on, and send the pixels in a single command.

The FAB\_LED library is very efficient in terms of the amount of memory used to manipulate the LEDs. Adafruit's libraries will easily take up 3kB of memory, while the FAB\_LED can carry out the same implementation in less than 800B in comparison. It achieves this by streamlining, using zero or very little inline assembly. This maximizes the code's readability and portability. We will be using the NeoPixel LED from Adafruit on this project. This gives us the option of using the WS2812 Smart Pixels smart module or the Adafruit NeoPixel library. The Adafruit NeoPixel has a "getting started" tutorial which provided all the information we needed to interface the LEDs with the Arduino Nano. By making use of proper C++ constructs and GCC compiler optimization code compiling is further optimized. The library also eliminated the need for us to use a pixel buffer, which saved RAM, which could then be used to drive more pixels.

The FAB\_LED library provided us with a variety of options for setup and interfacing. The support for both three and four byte protocols RGB and RGBW, with the colors in any order. The support of predefined orders and types of pixels helped with the coding and raw uint8t types. That along with the automatic conversion of pixel types to LED strip order greatly facilitated the code development for the project. Being able to update the LED strips in parallel was essential for displays we wanted to implement.

## 5.3.2a Arduino IDE

To develop and troubleshoot the software for the microcontroller the Arduino 1.6.13 IDE was used. The Arduino Software (IDE) is based on Processing and Wiring. Processing is a sketchbook software and language designed for coding within for the visual arts. Wiring is an open-source framework for microcontrollers created for designers and artists. While most microcontroller systems are limited to the Windows environment, the Arduino IDE runs on Macintosh OSX and Linux systems as well. Since all of our members were inexperienced with microcontroller programming we wanted to use an IDE that was easy to use and learn, while still being powerful enough for advanced users. The Arduino software is an open source tool, allowing for experienced programmers to continually expand the libraries through C++.

The Arduino Integrated Development Environment contains multiple components used for writing code for the microcontrollers. There is a text console and an editor for writing code. There is also a message area and a toolbar with buttons that carry our common functions.

The Arduino Integrated Development Environment refers to programs as sketches and use the file extension .ion. The Arduino IDE uses the concept of a sketchbook to store programs. The IDE supports helpful functions like cutting and pasting and searching and/or replacing text. Feedback is given through a message area, besides errors feedback while saving and exporting the sketch is provided. The

current configuration of the board being used and the serial port it is connected to is displayed in the bottom right corner.

While developing and troubleshooting the software for the microcontroller the board is connected to a computer via a USB cable. The USB cable doubles as a conduit for the data being flashed to the board and the power supply for the board. Figure #. # shows how the Arduino is connected to the computer via a USB cable and a strip of LEDs for software development.

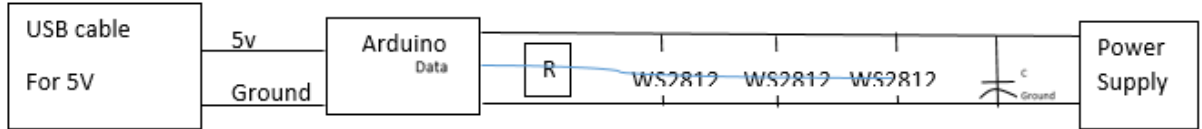


Figure 33: Arduino Connection

## 5.3.2b Bluetooth

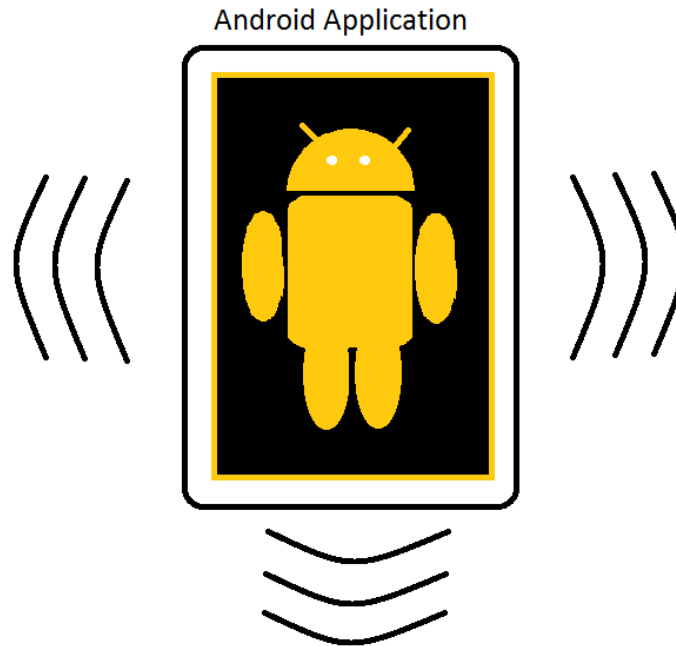


Figure 34: Android Symbol (made by Kelechi Ukachi-Lois)

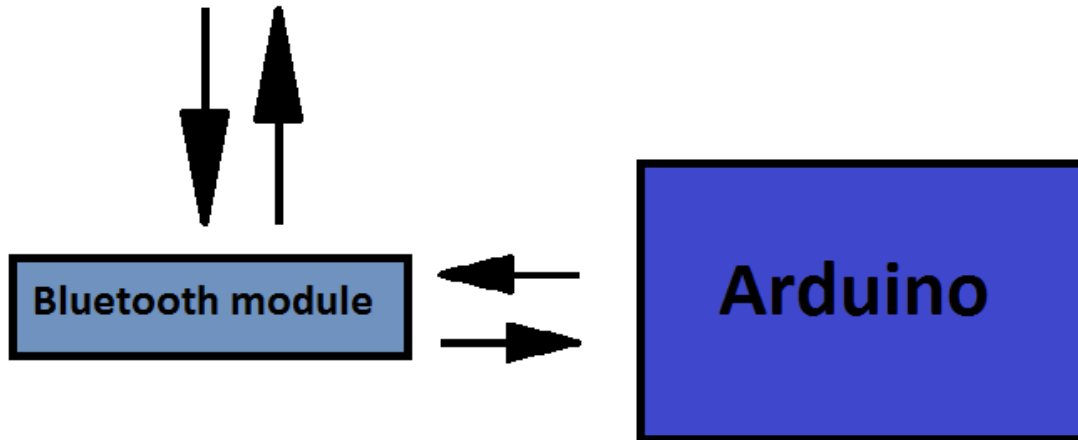


Figure 35: How Bluetooth Reacts with Arduino and Android Application

The Bluetooth software is a major integral part of the software component. It can be thought of as a pipeline. The Bluetooth software is used to tunnel information back and forth from the Arduino and the Android application. Being able to communicate to the Solar Powered Flower Sculpture through the Android application is a major feature of this project. The Bluetooth capabilities enable communication between both the Arduino and the Android application to become possible. Meaning, the Bluetooth software grants this project the ability to achieve one of its most important features and makes the Bluetooth software extremely valuable to the Solar Powered Flower Sculpture project. Since the Bluetooth acts as a pipeline between the Arduino and the Android application it is only logical that the software is on both the Arduino and the Android application.

From the perspective of the Android application, the design is very straight forward. In the Android application the Bluetooth capability is enabled in the code of the Android application. Through coding different abilities become available to the application. Readyng the application for Bluetooth communication involves quite a few steps. However, grouping these steps into sections would leave three clear sections. These sections are one, searching for devices, two, connecting to devices, and three, communicating with devices.

Below is an image taken from the prototype Solar Powered Flower Sculpture android application, showing the home screen with the Bluetooth option.



Figure 36: Prototype Android Application Interface (By Kelechi Ukachi-Lois)

The Bluetooth code in the Android application is designed to do two things when searching for devices. The first is to check for devices that have already been paired with the smartphone device. The application will generate a list of the paired devices. The second thing it be able to do is, search for available devices within the area. This step will generate a list of all the Bluetooth devices that were discovered. Once complete the application will have generated a list of both the paired and discovered devices.

When connecting to another Bluetooth device the Android application will also have a choice of two possible options. The connection step comes after the searching step. This means that there will be two lists that have already been generated. These lists are the list of paired devices and also the list of discovered devices. In order to connect simply select a device from one of these lists and

connect to it. The other option when connecting is to set the device discoverable so that another device can initiate the connection.

Below is an image taking from the prototype Solar Powered Flower Sculpture android application showing the Bluetooth screen.

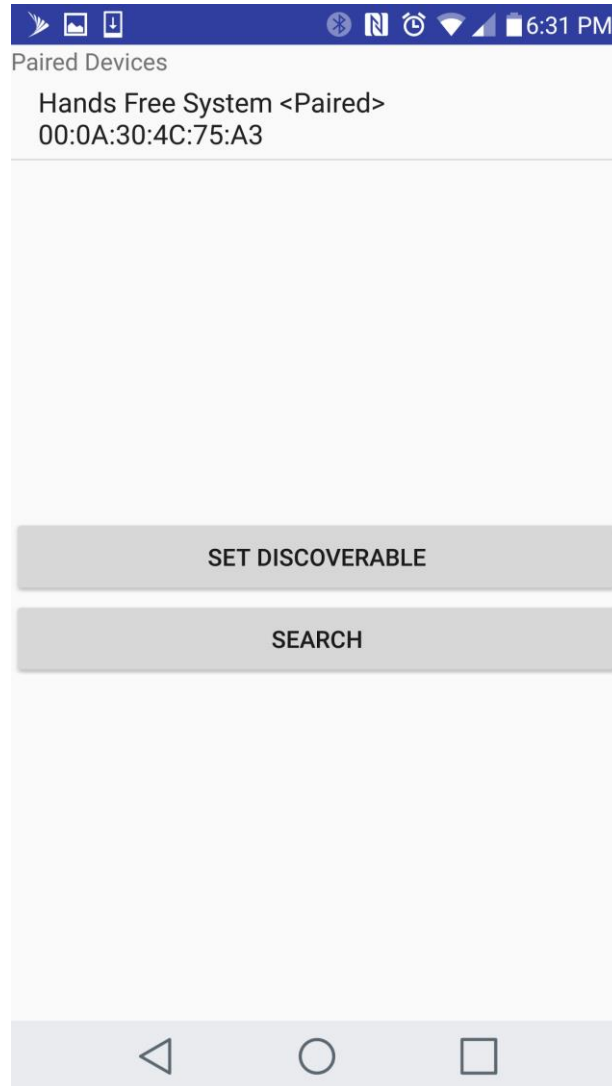


Figure 37: Bluetooth Pairing Screen from Application  
(By Kelechi Ukachi-Lois)

Once the devices are connected the next step is for the devices to communicate. Using a thread, the Bluetooth code will allow the Android application to communicate and send information through the Bluetooth code to the connected device. In this project the Android application will be communicating with the Bluetooth module connected to the Arduino.



The Arduino is connected to the Bluetooth module. The module has a transmitter and receiver on it which receives signals and can also send signals. These signals can be accessed through the Arduino code and be used to control different features. Based on the specific code that is received, different outcomes can be produced. For example, the user can program the Arduino to check for a sequence of letters or numbers to be received. If this specific sequence is received the user can initiate a function achieve a certain objective. This code gives the ability to receive signals which will control the Arduino remotely.

Understanding its purpose, Bluetooth's ability to connect is a major feature of the Bluetooth technology. Therefore, it is worth taking a closer look when it comes to the connection process of Bluetooth devices. The connection process for Bluetooth devices is a lengthy multi-step process. There is a generic process that is followed by the Bluetooth devices in order to form a connection. Of course, once the Bluetooth devices are connected, they form a Bluetooth network which is known as a piconet. The connection process can be divided into three progressive states. The general progressive phases are the Inquiry, the Paging phase (the connecting phase), and the Connection phase.

The initial stage in the connection process is the inquiry stage. During this phase it is assumed that the Bluetooth devices know nothing about each other. The Bluetooth devices must ask each other for information which is why this stage is known as the inquiry stage. When two Bluetooth devices know nothing about each other, one of the Bluetooth devices must run an inquiry to try and discover the other. If the other Bluetooth device is listening for a request, that Bluetooth device will respond with its address and possibly its name and other information which describes itself.

The second stage of the connection process is the paging process. During this phase is when the Bluetooth devices attempt to form a connection with each other. Of course, this paging process cannot be completed unless the inquiry process has already occurred. This is because the paging process needs information that is acquired in the inquiry process or it cannot complete its task. During the inquiry process the Bluetooth address of the other Bluetooth device is attained. Without the inquiry process the Bluetooth address of the Bluetooth device would be missing and without the Bluetooth address connection to the Bluetooth device is not possible.

The third phase of the connection process is the actual connection phase. The connection phase occurs after the paging phase (connecting phase). During the connection phase of the connection process the Bluetooth device has the option of doing many different things. Depending on usage and situation the Bluetooth devices can become active participants or they can choose to enter a low-power mode. During the connection phase there are four general modes the Bluetooth device can enter. This modes include active mode, sniff mode, hold mode, and park mode.

Intuitively, active mode is when the Bluetooth device is fully active. An active Bluetooth device is a Bluetooth device that is actively transmitting and or receiving data. Active mode is expected to be used when immediate use of the Bluetooth device is necessary. This could occur if a Bluetooth device was connected and immediately used to start transmitting and receiving data back and forth with another Bluetooth device. Different Bluetooth devices will have different needs and may not need to enter active mode immediately.

Another generic mode that is possible for a connected Bluetooth device is sniff mode. Sniff mode is another general mode that can be used in the connection phase of the connection process. Sniff mode is a power-saving mode in which the Bluetooth device sleeps but also listens for packets. The main difference between sniff mode and active mode is that active mode is continuously listening for packets whereas sniff mode takes breaks. Sniff mode will generally sleep and only listen for packets after a set interval of time. A common time interval is 100 ms of waiting in between each listen.

Hold mode is another general mode that a Bluetooth device can enter in the connection phase. Hold mode is another form of power saving. Hold mode works using intervals which is similar to sniff mode however, there is a main difference. During hold mode the Bluetooth device is fully asleep, meaning it will not listen or transmit packets during the hold mode interval. Hold mode is designed to be a temporary power saving feature. A master device has the slave the ability to make a slave device enter hold mode. Once the interval for the hold mode expires the Bluetooth device will wake up and enter active mode once again.

The other generic mode that a Bluetooth device can enter during the connection phase is park mode. Park mode is similar to both sniff mode and hold mode since it is also a power-saving mode. Park mode is the heaviest sleeper so to speak. Park mode is the deepest sleep mode of the three power- saving modes. Park mode is similar to hold mode except it is more extreme. During hold mode, it is as if the Bluetooth device is taking cat-naps however, park mode can be thought of as your Bluetooth device turning into sleeping beauty. During park mode the Bluetooth device sleeps and will stay inactive until the master Bluetooth device gives the command for the slave device to wake back up and become active again.

Another notable feature of Bluetooth is the Bonding and Pairing ability. When a certain affinity is shared between two Bluetooth devices, the Bluetooth devices can be bonded. Bluetooth devices that are bonded have a special ability to connect automatically when they come into range. An example of this is when a phone automatically connects to a car's Bluetooth system. This automatic connection is possible when the Bluetooth devices have a bond. The bonds required for bonding are acquired through pairing.

Pairing is when the Bluetooth devices share their addresses, names, and profiles. This information is then typically stored somewhere in memory. The Bluetooth

devices also share a common secret key. It is this secret key which allows them to bond whenever they are together in the future.

For security reasons pairing requires an authentication process. During the authentication process the user must validate the connection between the Bluetooth devices. Depending on the interface capabilities of the Bluetooth devices, the authentication process will vary slightly. In some cases pairing devices is a very simple process and only takes the press of a button, however, that is generally with Bluetooth devices such as headsets. This is because pairing is usually simpler for Bluetooth devices that do not have a user interface. Many times pairing devices will involve matching 6-digit numeric codes. In older version of Bluetooth such as, v2.0 and earlier, the pairing processes involves a slightly different method of authentication. Each Bluetooth device is to enter a common PIN code. "The PIN code can range in length and complexity from four numbers (e.g. "0000" or "1234") to a 16-character alphanumeric string." - (<https://learn.sparkfun.com/tutorials/bluetooth-basics>). Nevertheless, pairing and bonding are useful features that can be used with Bluetooth and the Solar Powered Flower Sculpture if deemed ideal and necessary to the Bluetooth capabilities of the Solar Powered Flower Sculpture.

## **6.0 Testing**

The following sections explain the methods that will be taken during the testing phase of the project. Basic hardware can be tested in Senior Design 1 and more difficult hardware and the software can be tested come next semester. As testing parts of the project is expected to cover a long period of time, all components with opportunities for viable testing will follow explicit rules. These rules describe how each component will be tested, how data will be accumulated and how that data can be read to improve the project. A combination of numerical and observational data will be recorded as a result of these tests, and multiple trials will commence to ensure accuracy of results.

### **6.1 Safety Procedures**

Extensive safety measures will be prepared before each testing phase is attempted. In situations where circuits are being tested, and there is a risk of circuit components burning/melting, attempts will be made to safely break the circuit (such as by turning off a voltage source or removing a component with rubber-covered tools). It is intended that these circumstances will be extremely rare, as thorough computer simulation testing will be made before live circuits are tested, and safe measurement ranges are calculated to avoid dangerous situations. In extraordinary cases where the situation could become dangerous, the testing will always involve at least two different group members working alongside each other, in addition to testing labs themselves featuring other individuals.

### **6.2 Breadboard Testing**

The actual breadboard will not be a component in our final design, but it will be an integral component in our prototypes. Their inclusion as a component in our prototype warrants their inclusion in our component testing phase as an error with a breadboard will have the same result as an error with any other component, such as a printed circuit board, effectively shutting down further development of the project. Using breadboards to test also allows for easier testing since printed circuit boards will require putting in a lot of money if one is not completely correct in its schematic.

#### **6.2.1 Testing Conditions**

These tests assume that jumper wires, Multi-meter probes, and the Multi-meter have been tested and are working correctly. These tests will require two jumper wires, one positive (red) wire and one grounded (black) wire, connected to both Multi-meter probes. The Multi-meter will be set to test resistance and voltage.

## 6.2.2 Measuring Approach

The breadboard will be tested for continuity and isolation. The “+” column for a bus will be tested first for continuity. The columns are tested for continuity by touching one of the probes to the top hole of the column and touching the other probe to the hole on the bottom of the column. If the column passes the test the Multi-meter should have a reading approximating zero. The continuity of the “-” column on a bus will be tested in the same way. When the “-” column is test the Multi-meter should display infinite resistance. Testing for isolation will be done by touching one probe to any hole in the “+” column and touching the other probe to a hole in the “-” column. This should also result in an infinite resistance reading, if the two bus columns are not connected electrically.

The row is tested for continuity and isolation, in much the same manner that the columns are tested. To test a row for continuity you place one probe in an opening, opening “A” for example. The other probe will be place in opening “E”. The Multi-meter should read zero if the rows have continuity. While keeping the first probe in “A”, move the other column in the same row, opening “J” for example. This will ensure that the row is isolated from adjacent rows. To test for isolation, you take one probe and touch it to “A”. The other probe is then placed in another hole labeled “A” in the row below. The Multi-meter should read infinite resistance if the rows are isolated from each other.

## 6.2.3 Implementation

### MAX712 Charge Controller

To breadboard test the MAX712 charge controller, the hand drawn schematics from earlier sections were taken into account. The chip that will be implemented on the breadboard looks as follows:



Figure 38: MAX712 Chip

To begin, all of the components were arranged on the breadboard to match the hand drawn schematics. This implementation on the breadboard will allow further

testing in a later section with the connection to the power supplies. The figures below show a side view and a top view of the charge controller circuit.

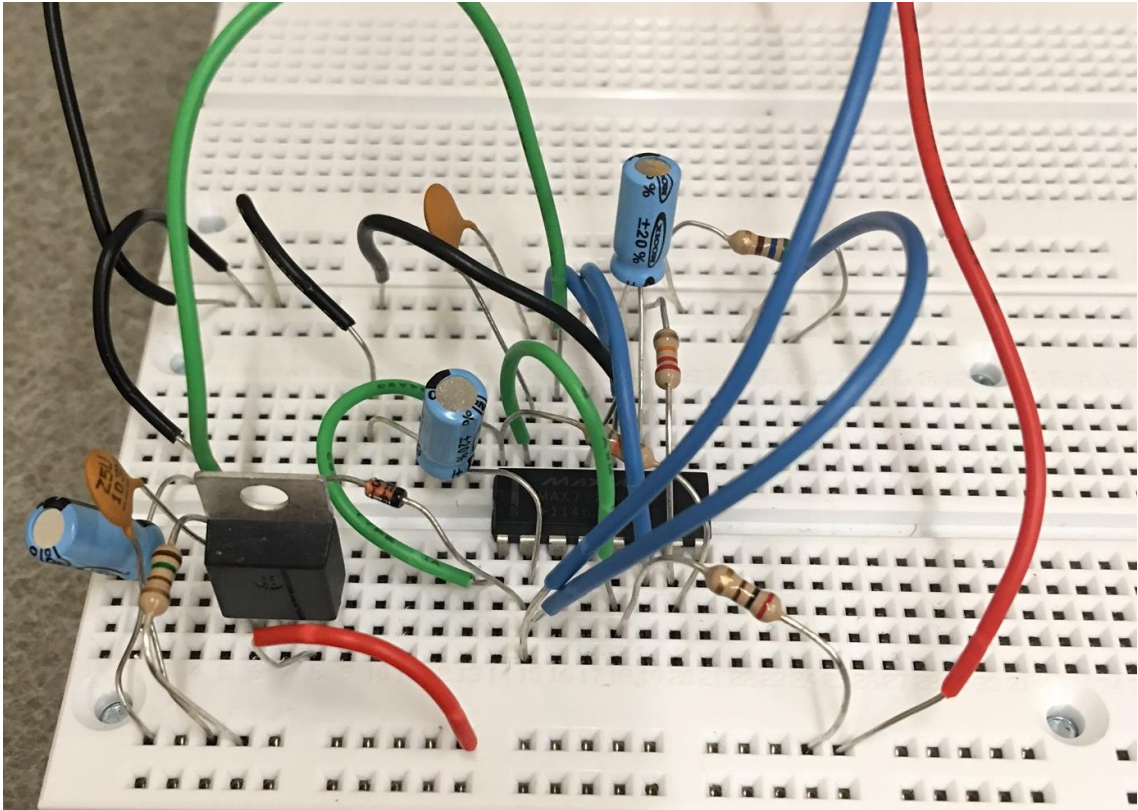


Figure 39: MAX712 Circuit - Side View

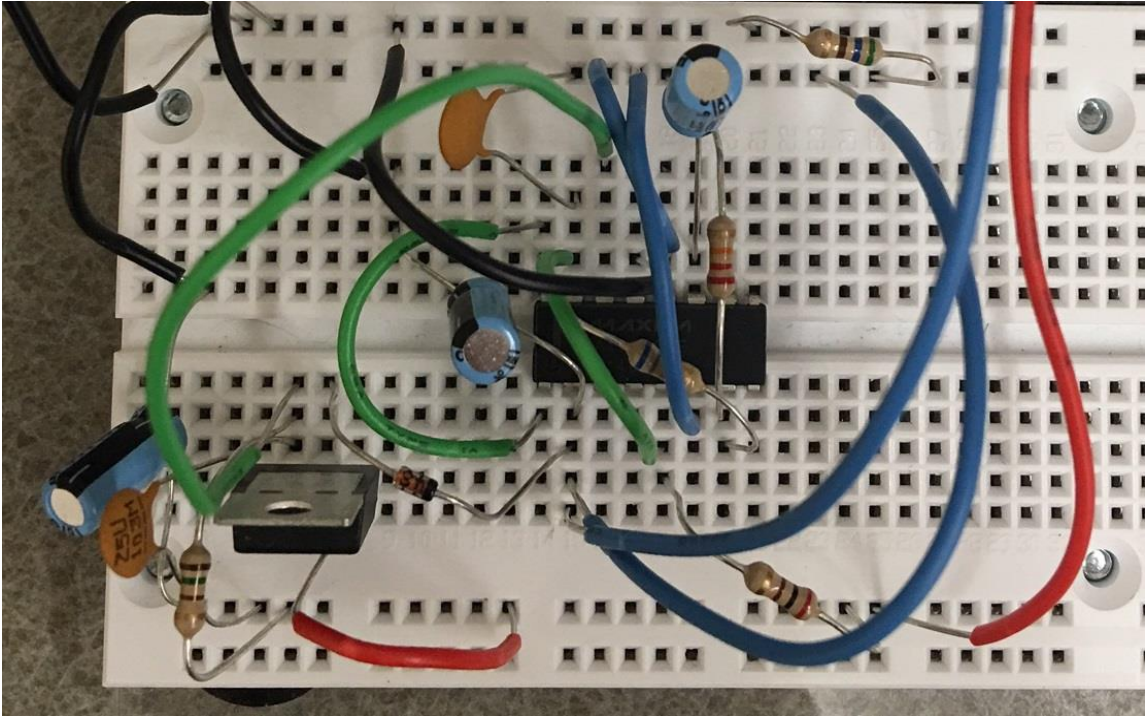


Figure 40: MAX712 Circuit - Top View

### **MAX639 Switching Voltage Regulator**

To breadboard test the MAX639 switching voltage regulator, the hand drawn schematics from earlier sections were taken into account. The chip that will be implemented on the breadboard looks as follows:

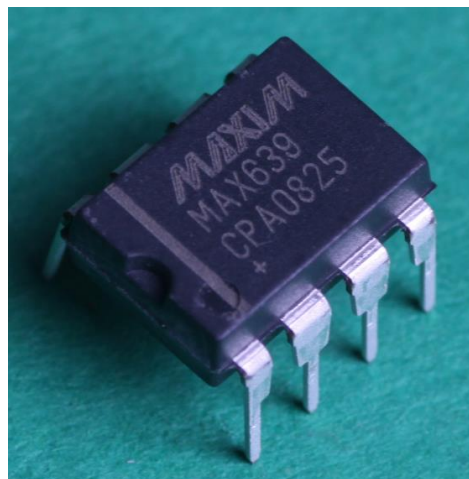


Figure 41: MAX639 Chip

To begin, all of the components were arranged on the breadboard to match the hand drawn schematics. This implementation on the breadboard will allow further

testing in a later section with the connection to the power supplies. The figures below show a side view and a top view of the switching voltage regulator circuit.

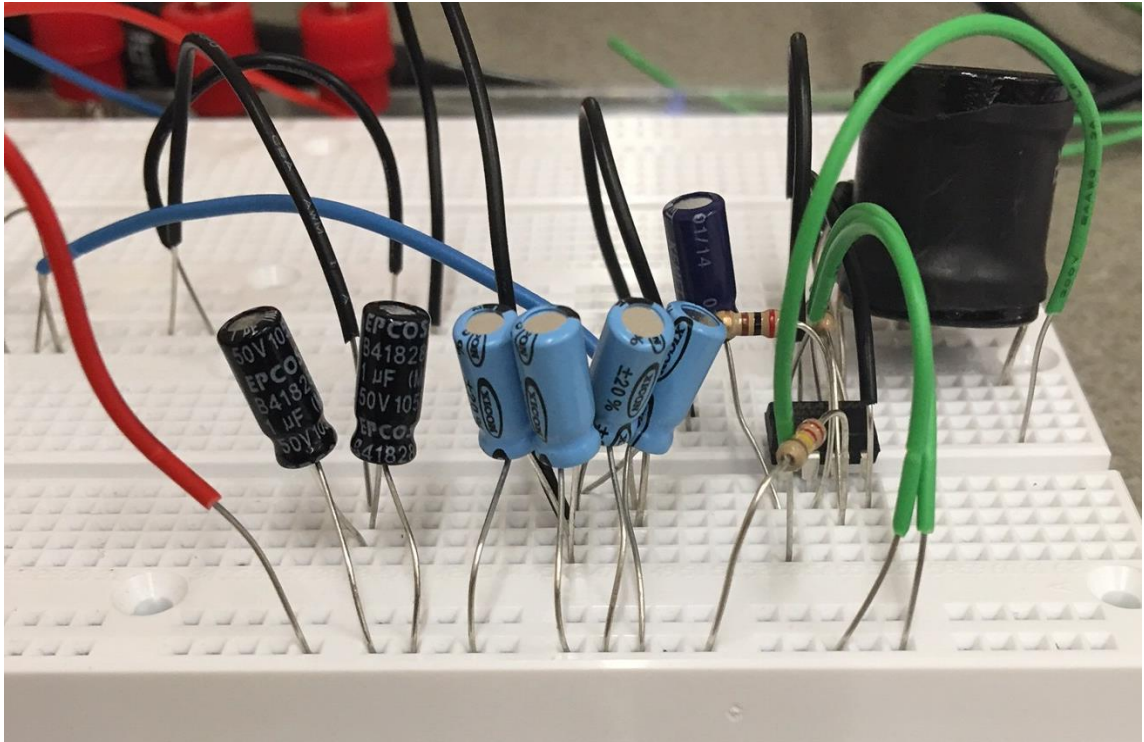


Figure 42: MAX639 Circuit - Side View



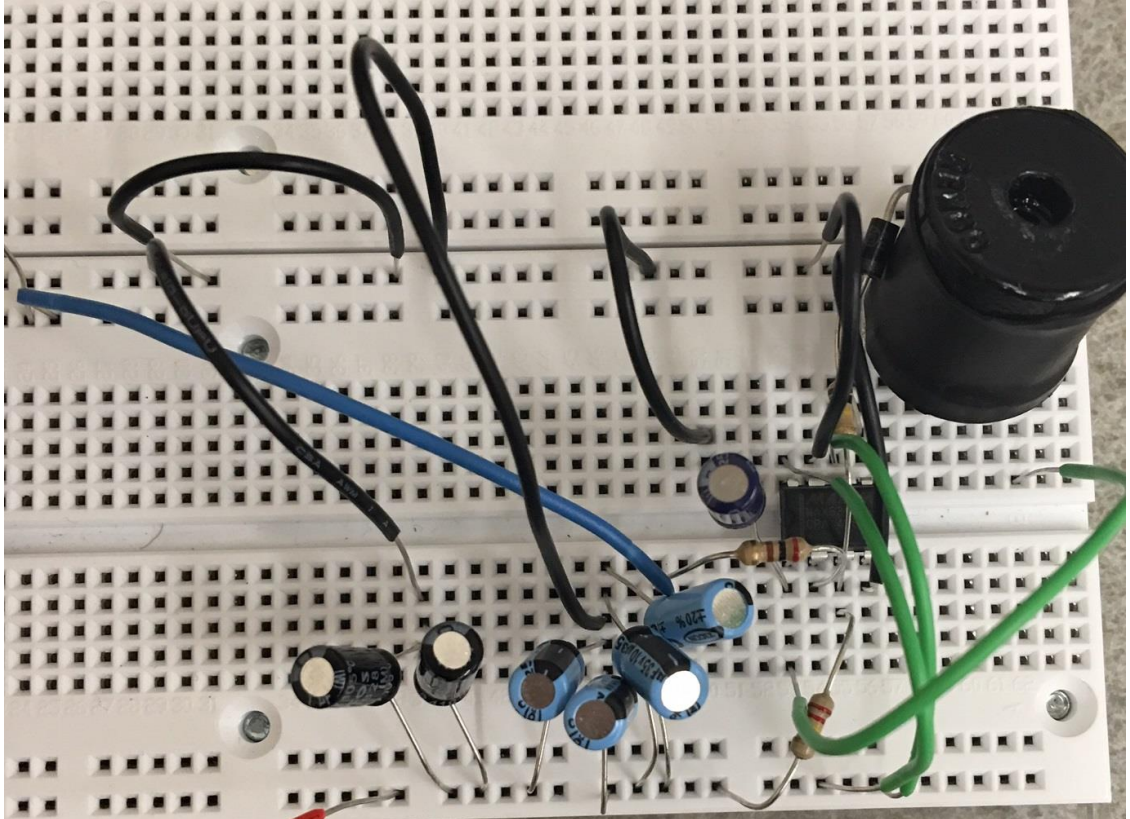


Figure 43: MAX639 Circuit - Top View

### **Arduino Nano and TLC5940**

To breadboard test the Arduino Nano and TLC5940, the hand drawn schematics from earlier sections were taken into account. The chips that will be implemented on the breadboard looks as follows:

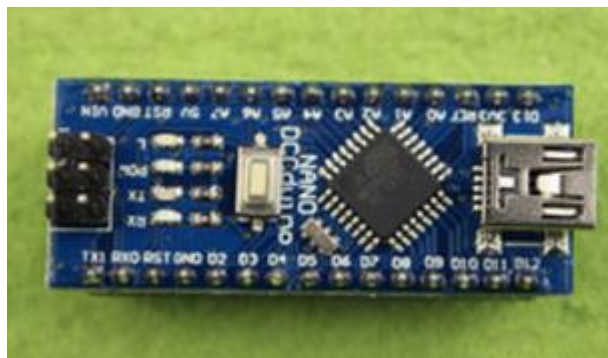


Figure 44: ATmega328 Microchip with Arduino Nano Breakout Board



Figure 45: TLC5940 Chip

To begin, all of the components were arranged on the breadboard to match the hand drawn schematics. This implementation on the breadboard will allow further testing in a later section with the connection to the power supplies. The figures below show a side view and a top view of the Arduino Nano and TLC5940 circuit.

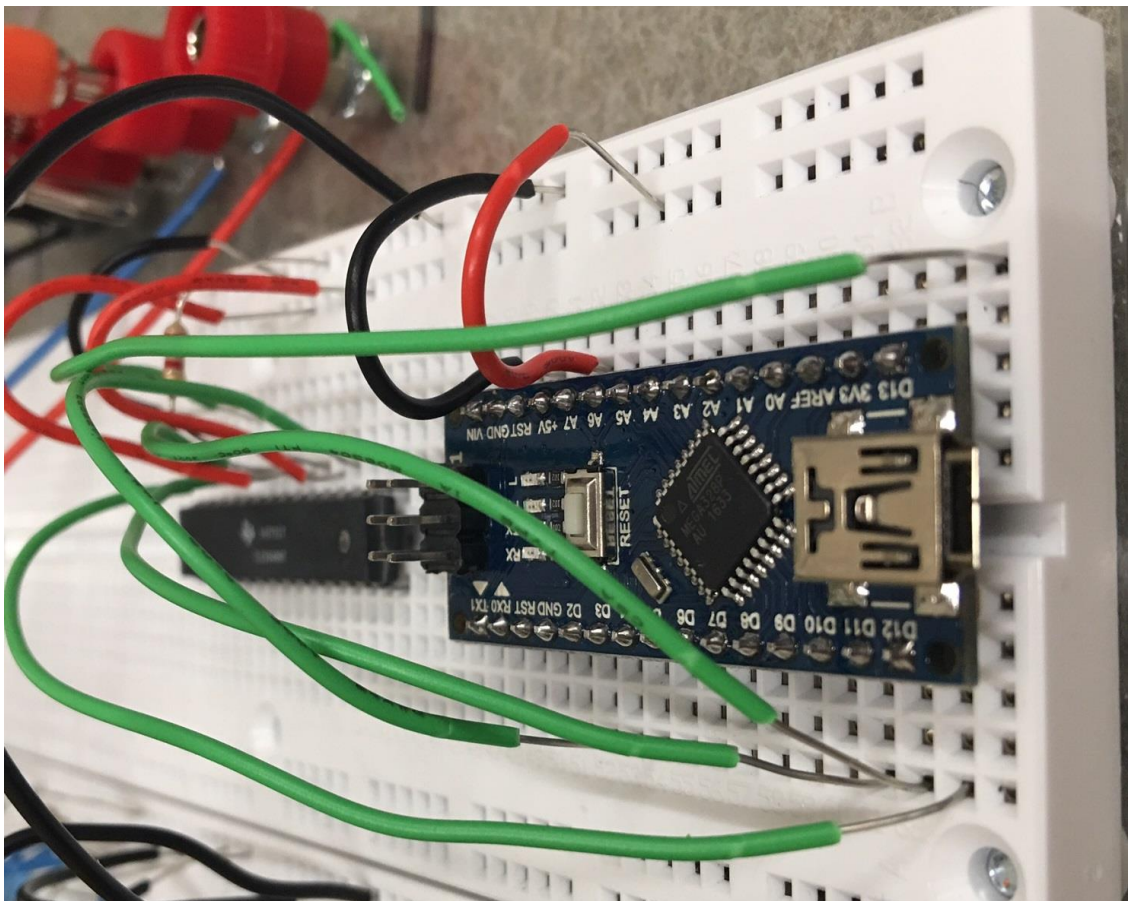


Figure 46: Arduino Nano - Side View

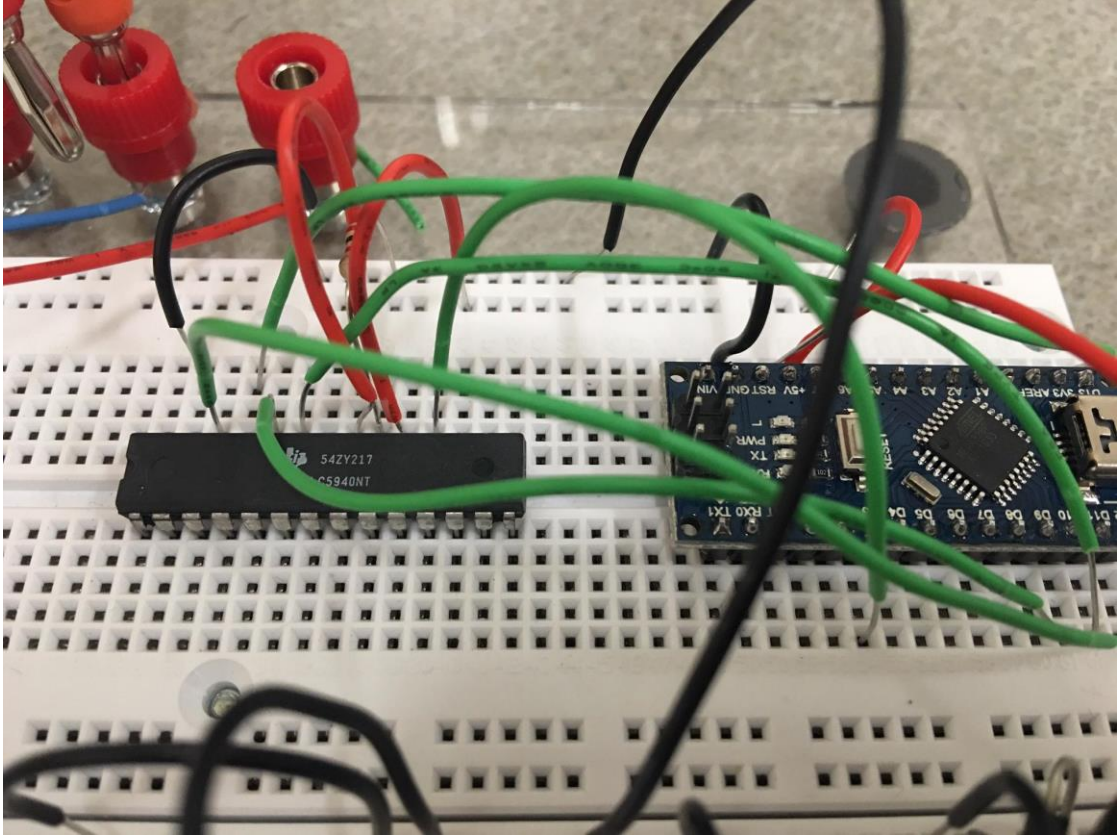


Figure 47: Arduino Nano and TLC5940 without LEDs

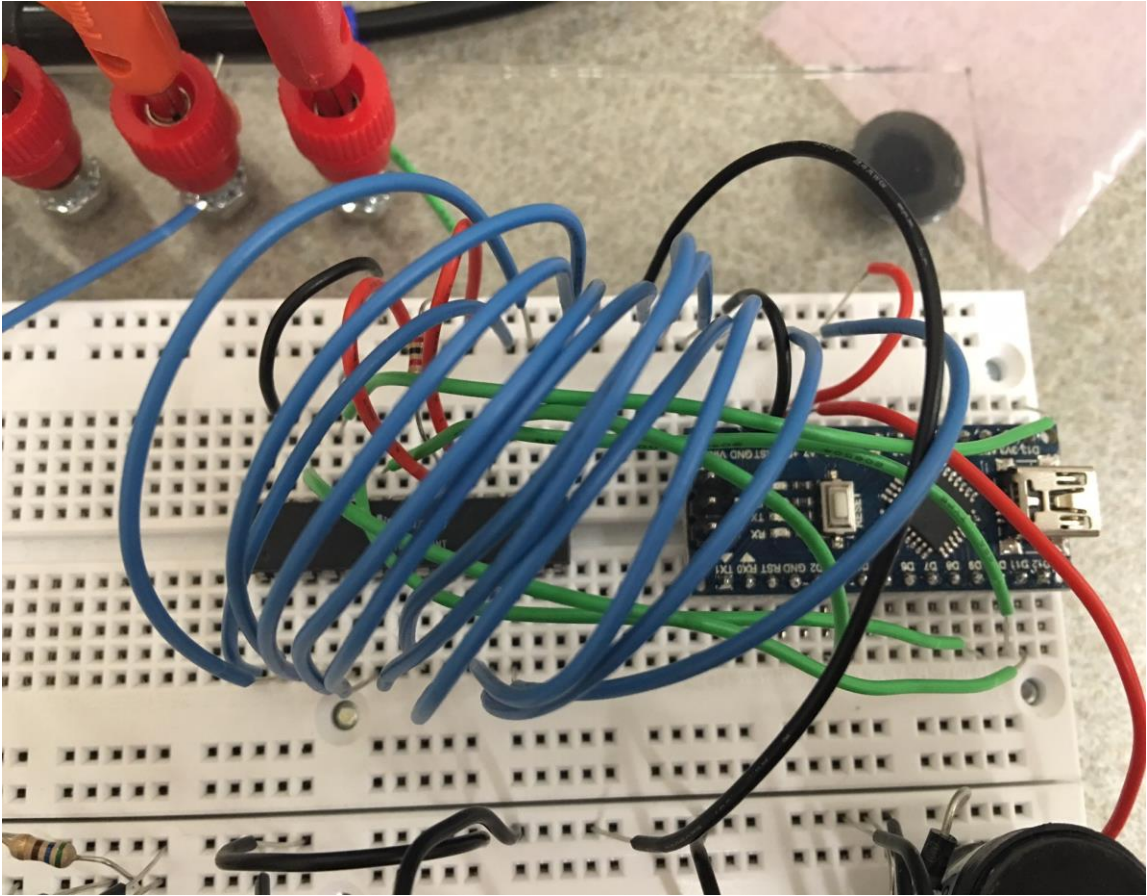


Figure 48: Arduino Nano and TLC5940 with LED Wiring

### **Combined System**

To breadboard test all of the individual component circuits in one combined circuit, the hand drawn schematics from earlier sections were taken into account. The combined circuit that will be implemented on the breadboard looks as follows:

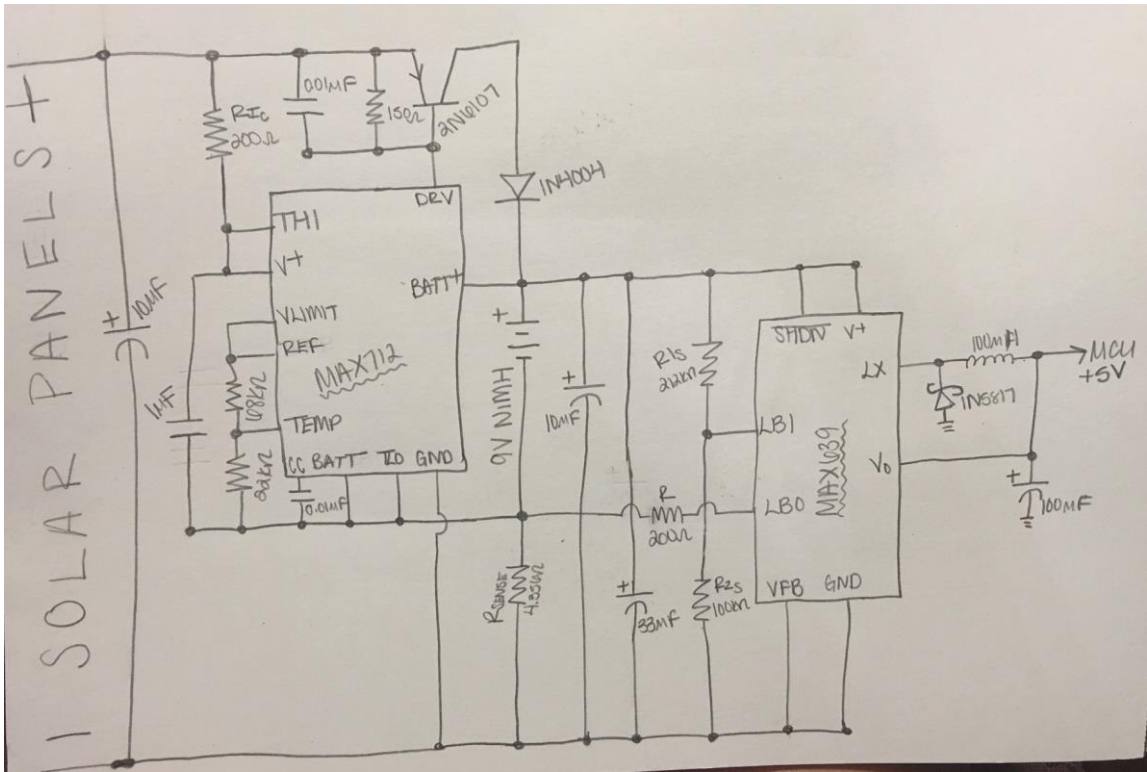


Figure 49: Combined System Circuit

To begin, all of the components were arranged on the breadboard to match the hand drawn schematics. This implementation on the breadboard will allow further testing in a later section with the connection to the power supplies. The figures below show a side view and a top view of the combined system circuit.

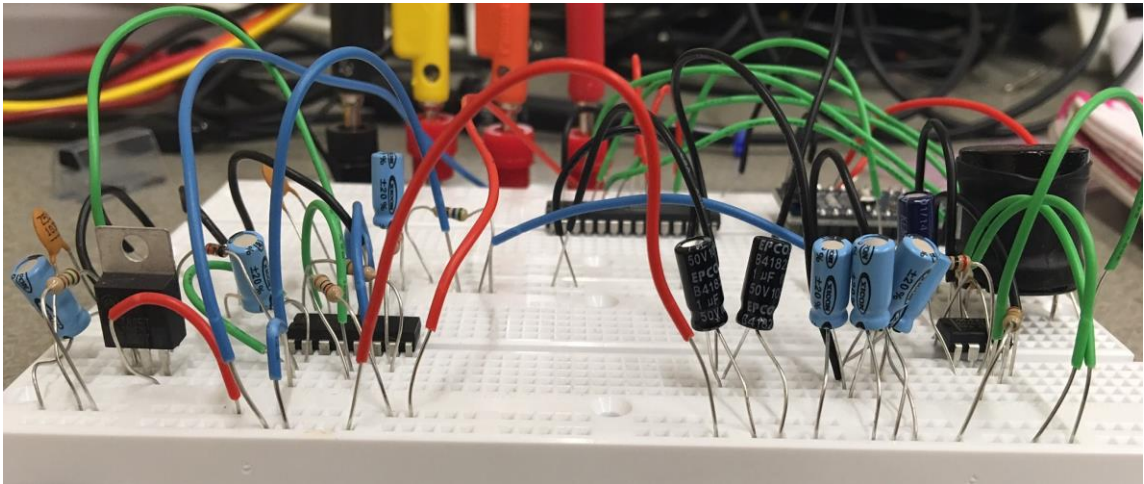


Figure 50: Combined System - Side View

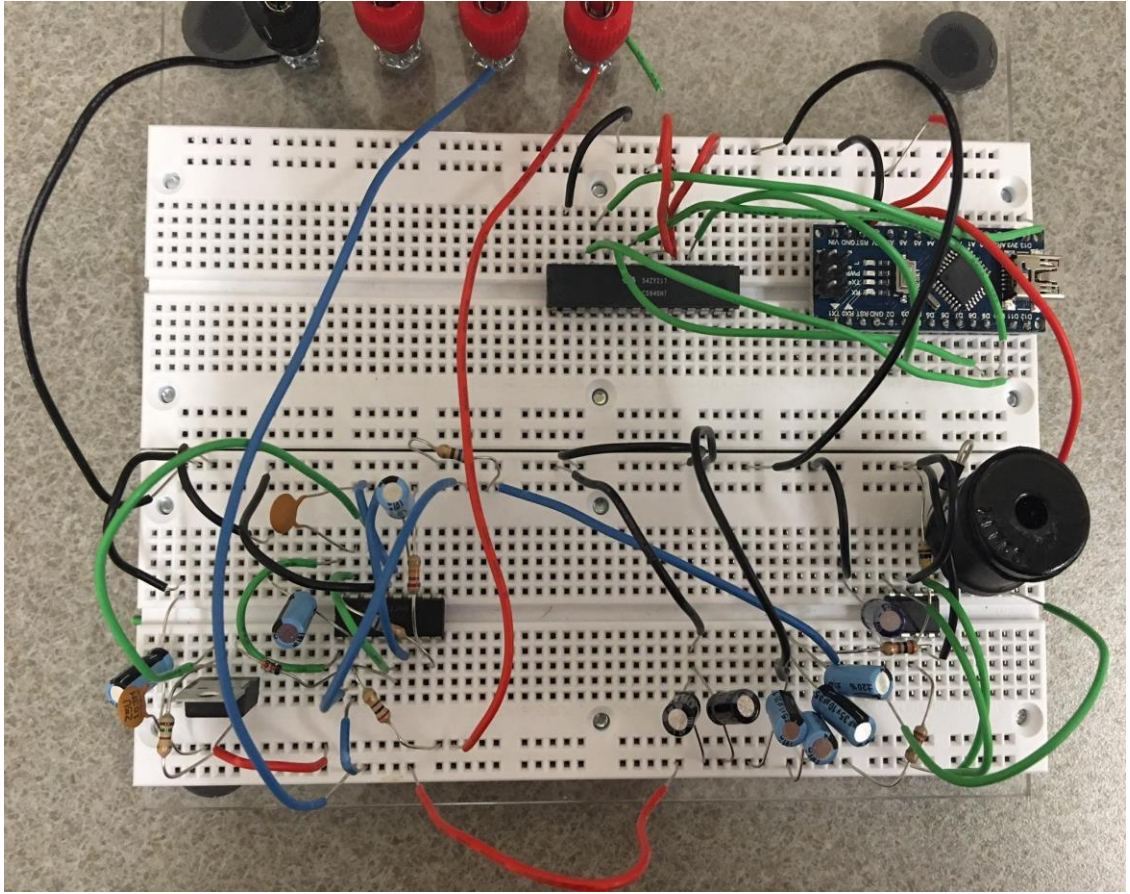


Figure 51: Combined System - Top View

**System with Photovoltaic Cells**

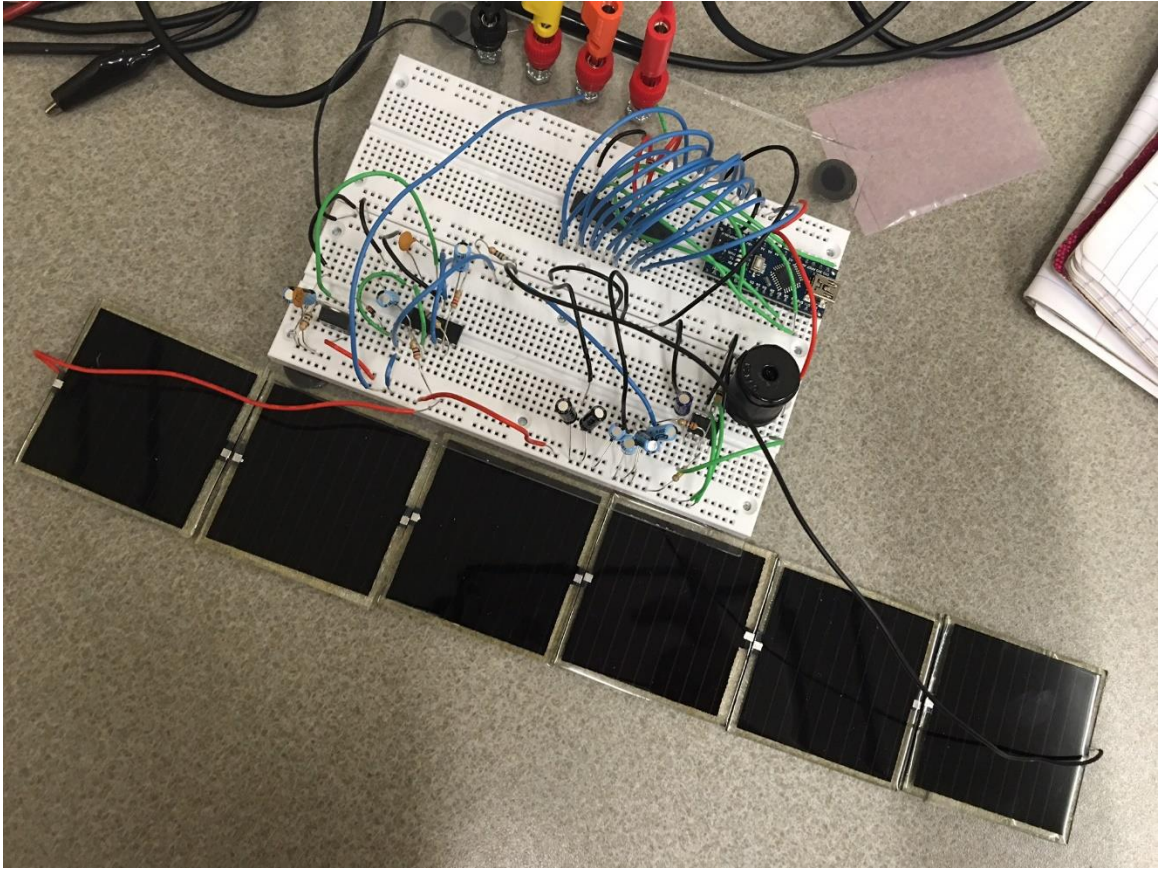


Figure 52: System with Photovoltaic Cells

The figure above has the same breadboard circuitry as the section above, but the difference is that the photovoltaic cells are what is powering the breadboard circuit. This is so since six photovoltaic cells are connected in series so that a large enough voltage is distributed throughout the system.

## 6.3 Power Supply Testing

The actual breadboard will not be a component in our final design, but it will be an integral component in our prototypes. Their inclusion as a component in our prototype warrants their inclusion in our component testing phase as an error with a breadboard will have the same result as an error with any other component, such as a printed circuit board, effectively shutting down further development of the project. Due to this dilemma, the breadboard is then connected to a power supply so to see if the circuit is capable of handling the input voltage.

## 6.3.1 Testing Conditions

To test a system with a power supply, the use of a Multi-meter is also needed. These tests assume that connection wires, Multi-meter probes, and the Multi-meter have been tested and are working correctly. These tests will require two connection wires, one positive lead (red wire) and one grounded lead (black wire) that will be connected to the positive probe and negative probe of the Multi-meter, respectively. The Multi-meter will then be used to measure the resistances and the voltages of a given circuit.

## 6.3.2 Measuring Approach

The power supply will be tested for continuity and isolation. The “+” column for a bus will be tested first for continuity. The columns are tested for continuity by touching one of the probes to the top hole of the column and touching the other probe to the hole on the bottom of the column. If the column passes the test the Multi-meter should have a reading approximating zero. The continuity of the “-” column on a bus will be tested in the same way. When the “-” column is test the Multi-meter should display infinite resistance. Testing for isolation will be done by touching one probe to any hole in the “+” column and touching the other probe to a hole in the “-” column. This should also result in an infinite resistance reading, if the two bus columns are not connected electrically.

The row is tested for continuity and isolation, in much the same manner that the columns are tested. To test a row for continuity you place one probe in an opening, opening “A” for example. The other probe will be place in opening “E”. The Multi-meter should read zero if the rows have continuity. While keeping the first probe in “A”, move the other column in the same row, opening “J” for example. This will ensure that the row is isolated from adjacent rows. To test for isolation, you take one probe and touch it to “A”. The other probe is then placed in another hole labeled “A” in the row below. The Multi-meter should read infinite resistance if the rows are isolated from each other.

After all of this, with the power supply turned on to the desired input voltage and connected to the breadboard, the Multi-meter can be used to read the voltage over any component. So long as the positive probe touches the point to be tested and the ground probe touches the grounded terminal or port. If these probes become switched, then the voltage displayed will be opposite of the desired measured value.



## 6.3.3 Implementation

### MAX712 Charge Controller

To begin testing with the power supplies connected, the breadboard schematic below was created.

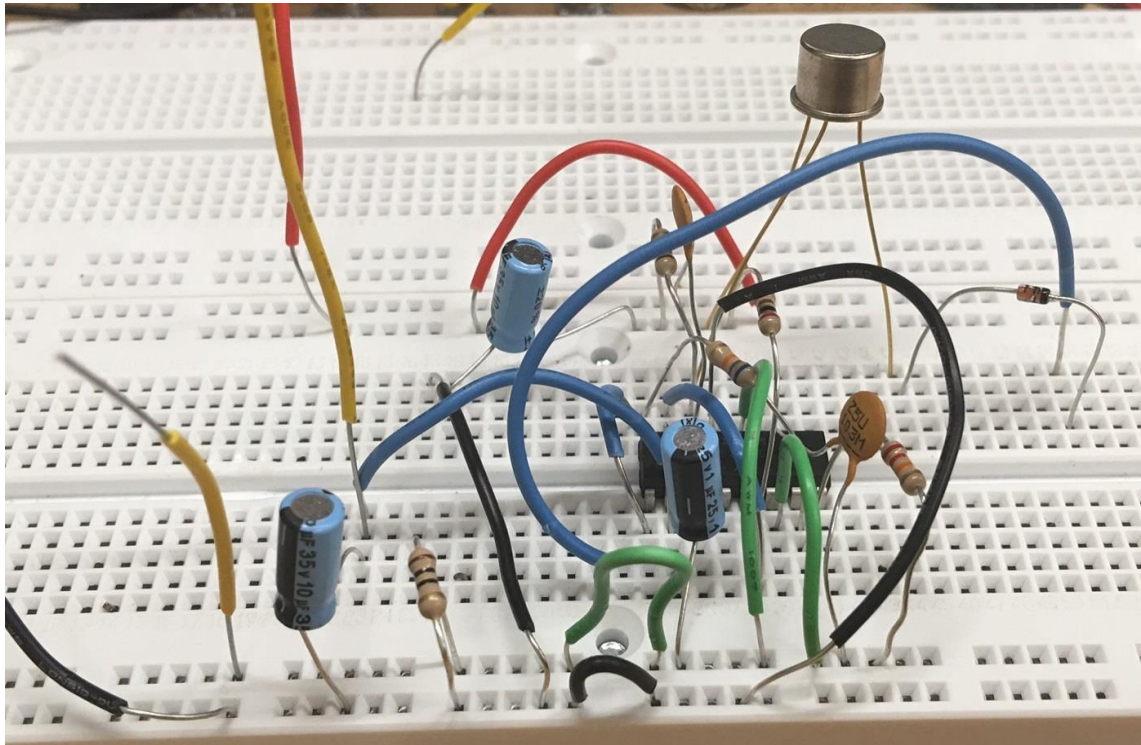


Figure 53: MAX712 Testing Schematic

With this breadboard schematic built, the values that will be used by the Solar Powered Flower Sculpture were simulated. This was done by assuming an average regulated input voltage of 20 volts. This value was used as an assumption since the maximum voltage the photovoltaic cells can draw in is about 27 volts, and the maximum voltage the power supply could output was 20 volts. This input voltage was transmitted to the circuit through the red wire that can be seen in the picture above. The black wires above are connected to the power supply ground terminal. The point of testing the MAX712 charge controller separately is to determine the potential average voltage that will be distributed to the nine volt rechargeable battery. This voltage can be determined by using a voltage Multi-meter to measure the voltage between the two yellow wires in the figure above. This measured voltage can be seen in the figure below:

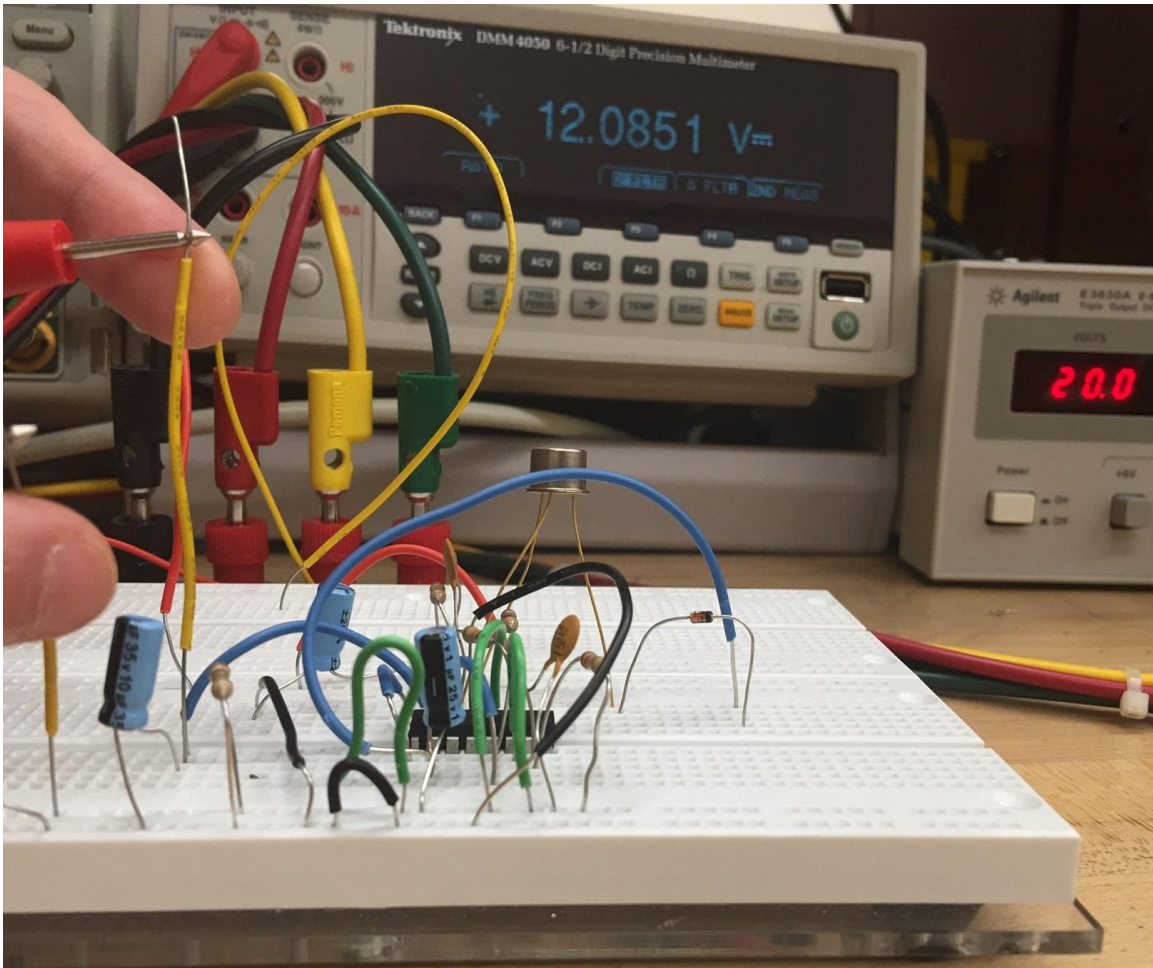


Figure 54: MAX712 Tested Values

As can be seen from the figure above, the measured voltage at the two yellow wires is about 12.1 volts. This measured value is definitely high enough to charge a discharged rechargeable battery, but is not high enough as to which would cause the rechargeable battery to explode.

### **MAX639 Switching Voltage Regulator**

To begin testing the MAX639 switching voltage regulator with power supply connections, the breadboard schematic was implemented below:

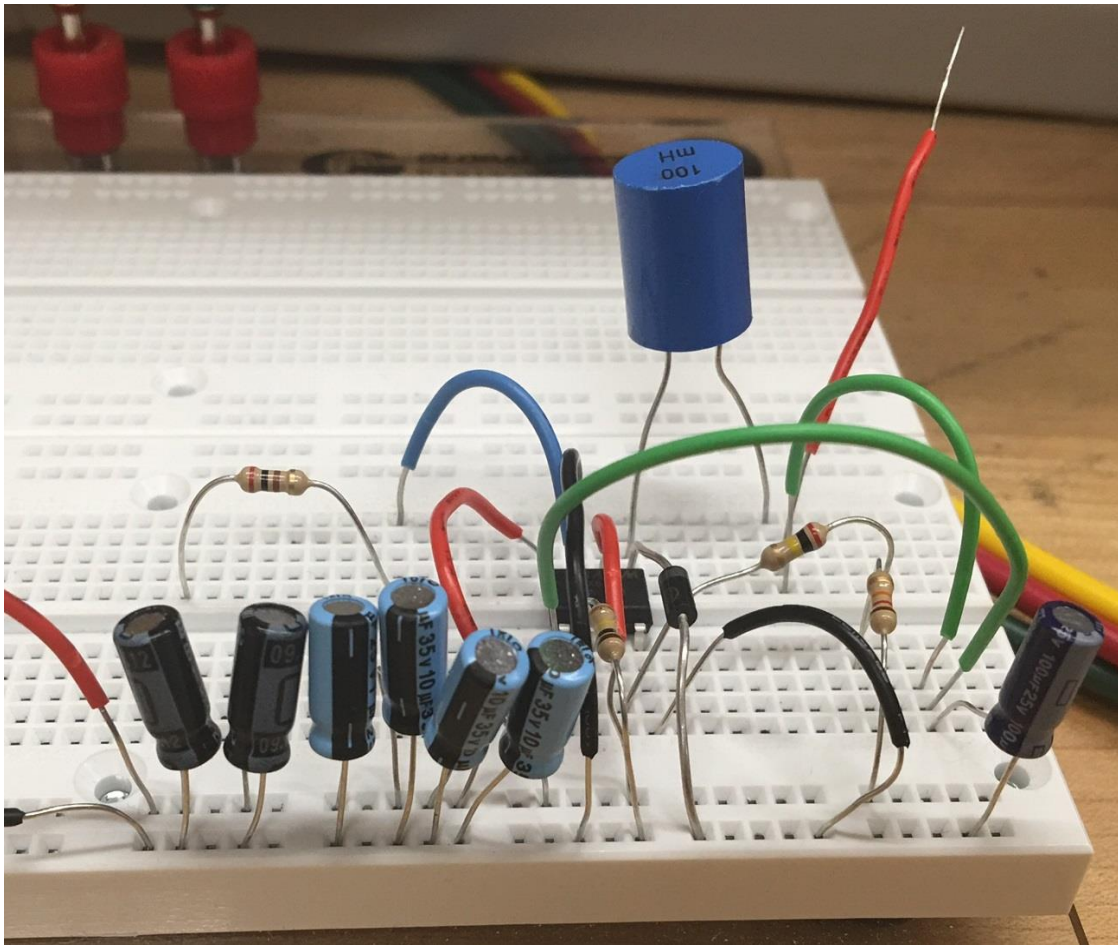


Figure 55: MAX639 Circuit to be tested

With the figure above built, the positive lead (red wire) can be connected to the power supply input voltage terminal and the ground lead (black wire) can be connected to the power supply ground terminal. With the breadboard circuit connected to the power supply terminals, the values needed for the Solar Powered Flower Sculpture can be simulated. The values assumed are a constant input voltage of nine volts, from the rechargeable battery, and from here that will be stepped down through the switching voltage regulator to a constant five volts. This five volts is what the Arduino Nano / ATmega328 microchip needs as an input voltage to operate.

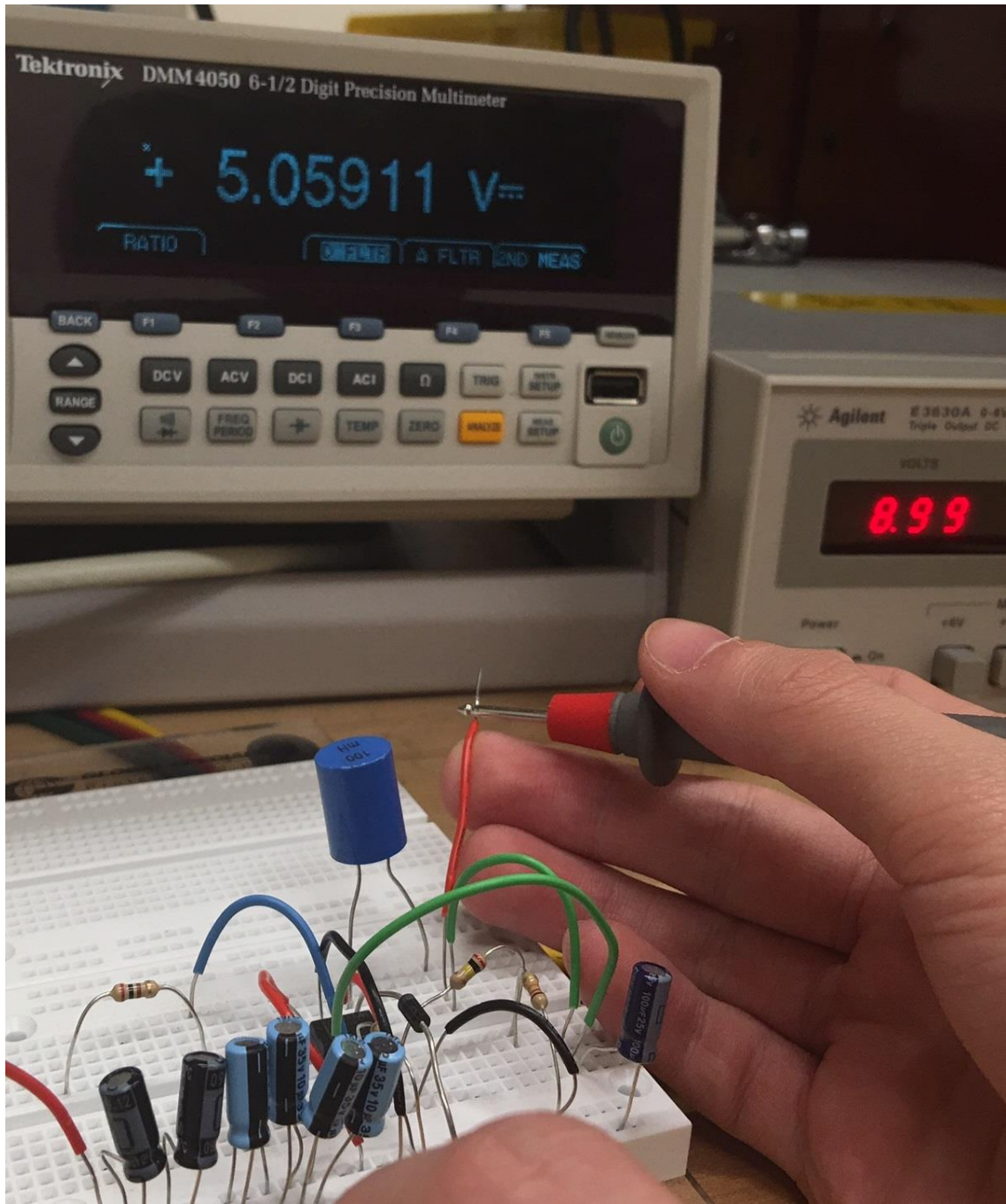


Figure 56: MAX639 Tested Values

As can be seen from the figure above, there is an input voltage of almost nine volts being sent through the positive lead wire. The Multi-meter is then used to test the voltage between the positive five volt input where the ATmega328 microchip will be connected and the ground. This measured value is tested at 5.06 volts which is allowed since the maximum operating voltage of the ATmega328 microchip at the positive five volts input is six volts.

## Arduino Nano and TLC5940

To begin testing the MAX639 switching voltage regulator with power supply connections, the breadboard schematic was implemented below:

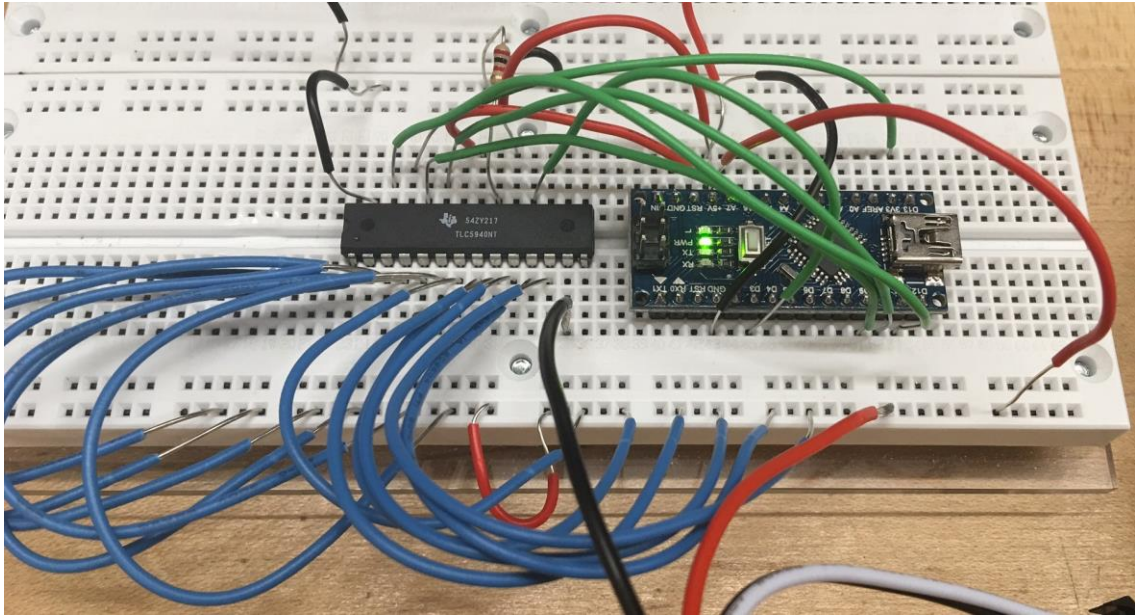


Figure 57: Arduino Nano and TLC5940 Test Circuit

From here the positive lead (red wire) is connected to the power supply voltage terminal and the ground lead (black wire) is connected to the ground terminal of the power supply. With the breadboard circuit now connected to the power supply, the assumed values for the Solar Powered Flower Sculpture can be implemented.

It is known that the operating voltage of the Arduino Nano / ATmega328 microchip is five volts. So, the power supply voltage terminal is set to five volts. This five volts is connected to the Arduino Nano positive five volt input and then also to the positive leads of the LED strips. The ground terminal is then connected to both component's ground inputs and then also to the negative led of the LED strips. The LEDs are connected in this fashion since the TLC5940 sinks current into all of the pulse width modulation output pins. The tested breadboard circuit appears as follows:

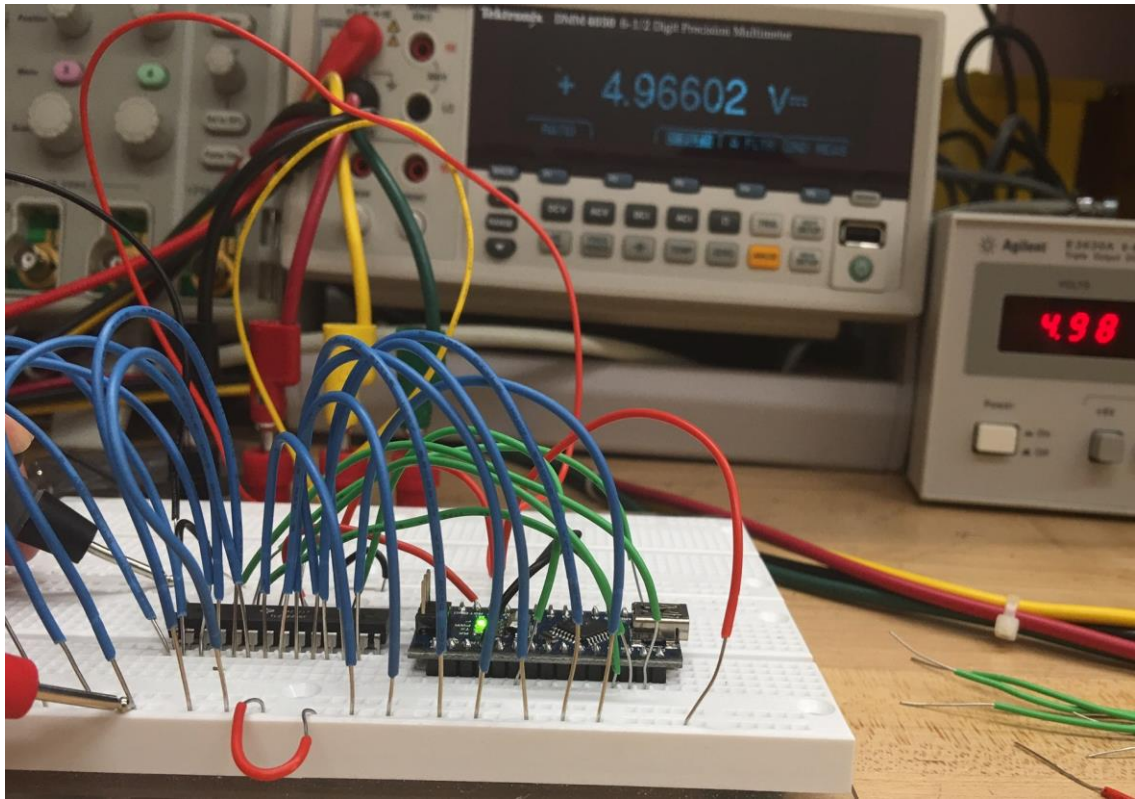


Figure 58: Arduino Nano and TLC5940 Tested Values

As can be seen from the image above, the input voltage from the power supply is at 4.98 volts. To test the voltage over the LED strips, the Multi-meter probes are used between the TLC5940 ground connection and then the positive lead since of the LEDs. The measured voltage across the LEDs is equal to 4.97 volts. This voltage would turn the LEDs on, however, the LEDs would not operate at full brightness.

## 6.4 LED Testing

Testing of the LEDs will have both a hardware and a software component. The LEDs will be connected to the Arduino Nano through the TLC5940 LED driver. Before we can do any testing it must be verified that there is power running through the Arduino Nano and that signals are being passed correctly to the TLC5940 LED driver. After we verify that the microcontroller and TLC5940 are receiving power the LEDs can be tested. The LEDs that are being tested are the Adafruit NeoPixel Digital RGBW LED Strip – White PCB 60 LED/m. These LEDs are smart LEDs with embedded microcontroller inside each LED. The brightness and color of each LED can be set individually with 8-bit precision. Because each LED-chip has a PWM built into it, once the color is set it should continue to PWM. The LEDs response to the signals by changing colors and brightness will indicate whether the LEDs are working properly or not. Test will be done via software modification on the microcontroller will be done first. Then the LEDs will then be tested with

commands sent wireless less through the mobile app to the microcontroller and then to the LEDs.

## **6.4.1 Testing Conditions**

These tests assume that the Arduino Nano, TLC5940 LED Driver, Bluetooth shield and mobile app has been tested and are working correctly. For each strip each LED-chip will be addressed and tested individually. For each LED-chip each red, white, green and blue LED will be tested individually. Then the LEDs will be tested together in different display patterns. The different patterns will consist of different on and off cycles, brightness combinations, and color combinations. At least three trials will be conducted during testing.

## **6.4.2 Measuring Approach**

The LEDs will be tested through the four modes of operation for the solar statue. All the results from the trials will be recorded for observation. It will be required that all tests result in zero errors within the trials for the test to be considered successful. If an error is indicated during test, then we will first determine if the error was a result of human error. If it is determined that the error was a result of human error, the error will be corrected and the test will be redone. If the error is not the result of human error, then will try to determine if the error is a result of manufacture defect. If it is determined that the part is defective, then it will be replaced and the test will be redone with its replacement. This process will be repeated until all trials result in successes.

## **6.4.3 Implementation**

To begin implementation, the NeoPixel Digital RGBW LED strips that the group purchased looks as follows:

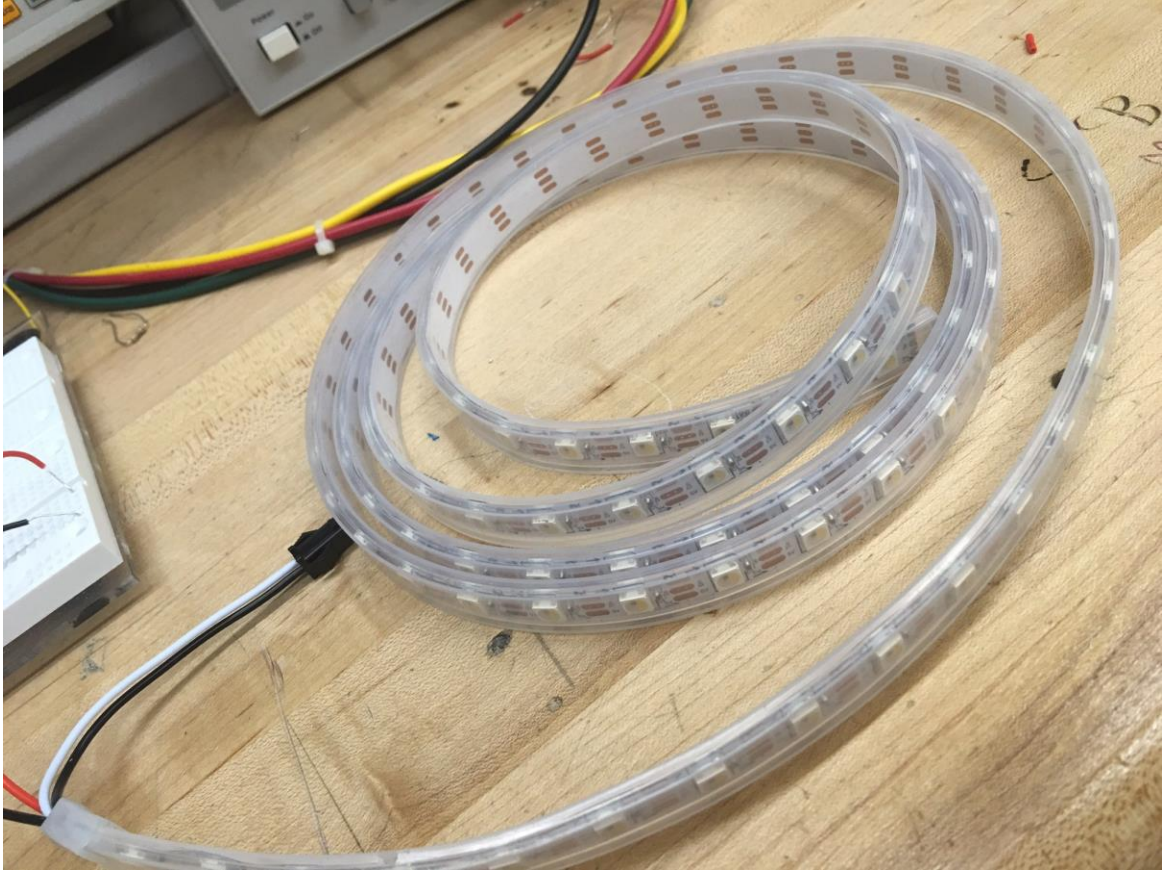


Figure 59: NeoPixel Digital RGBW LED Strips

From here the NeoPixel Digital RGBW LED strips were connected to the breadboard circuit. This is done by connecting the positive lead and the negative lead to the Arduino Nano / Atmega328 microchip and the TLC5940, respectively. This appears as follows:



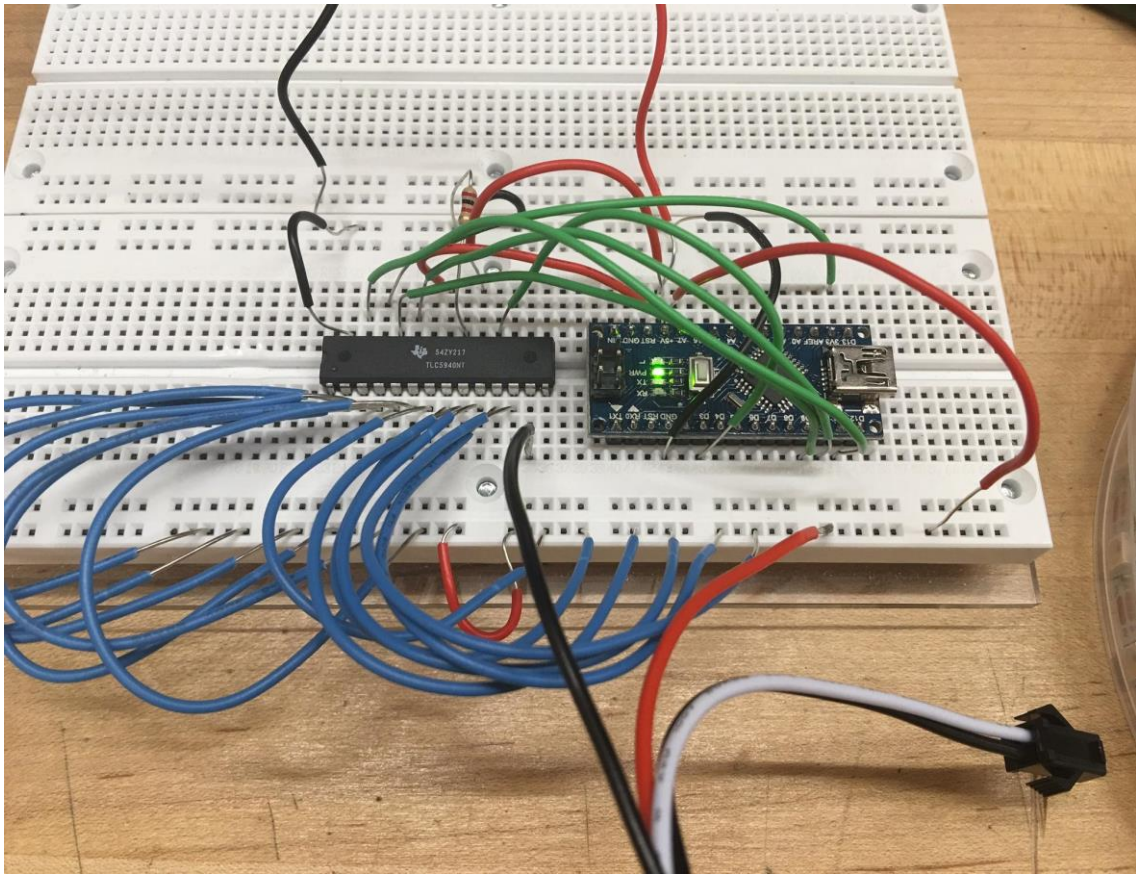


Figure 60: LED Strips Connection

The reason the positive lead of the NeoPixel Digital RGBW LED strips are not connected to the output pins of the TLC5940 is such since the TLC5940 output pins sink current from what is connected to them. This means that current is pulled into the pins instead of the current being released from the pins.

During testing the group was not able to get the LED strips to turn on. The group thinks this could be from one or two obstacles. The first potential obstacle is that the 1.5 meter wire was kept intact. This is so since none of the group members at that time have practice or are skilled in soldering. The 1.5 meter LED strip is meant to be cut in such a way that ten RGBW LEDs are left on an individual strip, which would allow twelve individual strips to be cut. The downside to trying to test the LED 1.5 meter strip is that there was potentially not enough current being pulled from the positive five volts output of the ATmega328 microchip to light up the whole strip. Looking back at the testing stage, this could have been altered or corrected by using a pull up resistor that would have increased the current enough to show the LED strip turn on.

The second obstacle that could have prevented the LED strips being turned on is that the pulse width modulation frequency had not been adjusted from the TLC5940. This would be due to the fact that coding for the NeoPixel Digital RGBW

LED strip has not yet been completed. With a completed and finalized code for correctly activating each and every individual LED on the LED strip, the testing could have been completed.

## **6.5 Wireless Module Testing**

Testing the wireless module involves checking of both hardware and software. The wireless module must be connected to the Arduino microcontroller. Before any testing can occur, power must be running through the microcontroller. After the wireless module is fully connected with power running through the microcontroller it can be tested. The wireless module that is being used in the Solar Powered Flower Sculpture is the HC-05 wireless Bluetooth module. The HC-05 wireless Bluetooth module is equipped with its own LED lights. These lights respond when another Bluetooth devices is attempting to either pair, attempting to connect, or has successfully connected to another Bluetooth device. These lights will give indication to whether or not the HC-05 wireless Bluetooth module is functioning correctly. The HC-05 wireless Bluetooth module will also be tested via software. HC-05 wireless Bluetooth module is to be connect with another Bluetooth device. Once connected a test will be run to see if the module is properly receiving and transmitting messages. Using the connected Bluetooth device, messages will be sent to the HC-05 wireless Bluetooth module. Response messages will also be sent from the HC-05 wireless Bluetooth module to the connected Bluetooth device. Once this is completed, each device will be checked in order to determine whether or not the information sent in the form of message were received. Based on which message are received it will be possible to determine whether the transmitting and receiving ports are adequately completing their tasks and functioning correctly.

### **6.5.1 Testing Conditions**

These tests assume that the Arduino Nano has already been tested and is working correctly. Upon connection the Bluetooth will be observed to see if the LED indicators built into the Bluetooth shield indicate that the device is receiving power from the Arduino Nano and is working properly. A Bluetooth terminal will be utilized to test the sending and receiving of data to and from the Bluetooth shield. After it is verified that the device is connecting and working properly, the Bluetooth shield will be test for receiving and sending data to and from the mobile app.

After initial testing follow up testing will be conducted with the Arduino Nano being connected to the TLC5940 and the LED strips. These test will assume that the TLC5940 and the LEDs have been tested and are working correctly. The Bluetooth terminal will be again utilized to test the sending and receiving of data. Command to activate the different display modes will also be tested in this configuration.

During the final testing phase the Bluetooth terminal will be replaced with the mobile app. All sending and receiving of data will be done via the mobile app.

Whether the correct displays and modes are being activated will determine whether the Bluetooth shield is working correctly or not.

## 6.5.2 Measuring Approach

The Bluetooth shield will be test for first the making of proper physical connection with the Arduino Nano. Then it will be tested for the transmission and receiving of simple data. Then it will be tested for proper communication resulting in the proper displays being activated. Finally it will be tested for proper communication with the mobile app. It will be required to pass all test with zero errors. If an error is detected at any stage of testing, it will be determined if human error is at fault. If it is determined that the error was a result of human error, the error will be corrected and the test will be redone. If the error is not a result of human error, we will try to determine if the device is faulty. If it is determined that the device is faulty. The device will be replaced and the test will be redone. This process will be repeated until all trials result in zero errors.

After the Bluetooth Shield is connected to the Arduino Nano and the onboard LEDs indicate that the Bluetooth Shield is working correctly the initial test of transmitting and receiving data can take place. Initial test will be conducted first which will entail just the transmitting and receiving of data, no specific functions will be tested. These test will consist of sending single characters and simple phrases to verify that the data isn't being corrupted during transmission. We will observe the display on the Bluetooth terminal to verify whether data is being transmitted correctly or corrupted.

After we verify that we can send and receive data without corruption, we can test the transmission of specific functions. With the TLC5940 LED driver and LED strips interfaced into the Arduino Nano we can test for the transmission of specific functions and commands. Instead of relying on just the Bluetooth terminal display, we will observe the LEDs for activation of proper displays when specific commands are send through the Bluetooth shield. This portion of the test assumes that the TLC5940 LED driver and the LED strips have already passed their tests and are working correctly. They will not be considered as a possible source of error if an error occurs during this portion of testing. It will be required that all testing at this phase result in zero error before we can move on to the next phase of testing. If an error is indicated at this phase, we will try to determine if the error is a result of human error. If we can determine that it is a result of human error, we will correct the error and redo the testing. If we can't find a human error, we will try to determine if the error is a result of faulty equipment. If we can determine that the error is a result of faulty equipment, we will replace the part and redo the testing. This process will repeat until we successfully complete this phase of testing without any errors.

The final stage of testing for the Bluetooth shield will be done with the mobile app. Commands to activate specific displays and modes will be send to the device via

the mobile app. It will be required that all test result show no errors. Again since it will be assumed that the Arduino Nano, the TLC940 LED driver, and LED strips have been tested and working correctly, any error will be assumed to be with communication between the app and the Bluetooth shield. If an error is indicated, we will try to determine if it is the result of a human error. If the error is the result of human error, the error will be corrected and the test will be redone. If the error is not a result of human error, we will try to determine if the error is a result of faulty equipment. If it can be determined that faulty equipment is the source of the error the faulty part will be replaced and the test will be redone. This process will be repeated until all trials result in no errors.

## **6.6 Microcontroller Software Simulations**

The Microcontroller Software Simulations were done on The Autodesk 123d Circuits platform. This platform comes with three main components, electronics lab, PCB Design + Manufacturing, and Circuit Scribe. The electronics lab simulates the Arduino software and board along with the components that would be used along with a breadboard. With this simulator we were able to test our code in real-time. This was the best way for us to make mistakes without risk of breaking our board. The app runs in the browser so there wasn't any software we needed to download, and the basic service is free. There is an option to upgrade but the basic service was enough for us to develop our project. The drop and drop interface for the components keep the learning curve down to a minimum for us. We were able to work as a team virtually via Autodesk circuits when our schedule didn't permit for us to meet up face to face. The PCB Design + Manufacturing components allows for the designing or importing of PCB boards. If components are not available, you are able to design you own and order them. The final component is the Circuit Scribe used to create circuit sketches.

### **6.6.1 Testing Conditions**

These tests assume that the computer the software will be running on and the software itself has been tested and is working correctly. A virtual mockup of our design, or as close to as possible, will be created. The software will be developed and the simulation will be executed. The simulated display will be observed for the proper execution of the code.

### **6.6.2 Measuring Approach**

The software will be tested for proper execution of each function. The functions for the different modes and displays will be tested. For the testing to be considered successful the simulated display will have to match that of what the final display will look like. If an error occurs, we will try to determine if the error is with the software being developed. If a problem with the software is discovered it will be corrected and the test will be repeated. If the software is not the source of the error,

it will have to be verified that the current simulator is capable of simulating the desired results from the software. If the current software is not capable of simulating the desired results than another simulator will have to be used or that particular test will have to wait for testing on the actual hardware.

## **6.7 Microcontroller and External Circuit PCB Interface**

The purpose of this testing phase is to accurately confirm all components of the created PCB are in working shape. Performance will be judged, and measurements will be made, which will then be analyzed by the group. These tests will be performed after the PCB design has already been implemented.

### **6.7.1 Testing Conditions**

There are several factors at stake that add to our project team wanting to avoid having to radically redesign or replace components in our completed PCB. For starters, ordering new specific parts is an expensive ordeal, and shipping time is unpredictable. In addition, a lot of additional work would have to be taken after the circuit has already built, and due to the soldering process, it is likely that additional components would have to be replaced if one was found to be faulty or otherwise ineffective. Therefore, our group will take all precautions during earlier testing phases to ensure that the microcontroller and circuit interfaces meet all requirements specified, and that the parts being implemented are accurately working. Testing the completed PCB will hopefully serve to verify all needed conditions found, and reinstate the consistency of results. Still, in the event of inaccurate results found being linked to components of the PCB design, changes may need to be made in order to produce a working interface. Rebuilding the PCB is not the ideal course of action, and preparations made in earlier testing phases leading up to the creation of a working PCB serve to make such an incident unlikely, but are group is prepared to do so in order to produce a final PCB that works with precise accuracy.

After a PCB with connected microcontroller and external circuit interface has been presented for testing, various components of the board will be measures, similar to the initial testing phase taken with the breadboard. Several trials will be completed during testing, to determine if results found are consistently accurate. The testing environment will be a closed lab setting without any external factors (such as harmful temperature and weather conditions) that may alter the effects of data. Only after all group members are satisfied with the readings recorded will we move on.

## 6.7.2 Measuring Approach

All data deemed necessary for analyzing by the group, will be recorded during testing for later reference and verification of findings. This includes voltage readings at various locations of the board, as well as current flowing through the system. These results will be compared with both measured expectations, and the results of the breadboard testing. In addition, more references of data may be decided by the group during the development phase, to be measured.

Ports at which the microcontroller connects to all LEDs will be measured to determine if the correct signals are being passed through as during earlier testing periods. A combination of using electronic testing equipment, and observing computer data readouts for the Arduino board, may be used. Power supply findings will also be tested when connected to the PCB, to verify data is as expected from earlier testing with these components. Any other components of the external circuit interface that directly or indirectly connect with the microcontroller on the PCB, such as the solar panels, may also be measured. Although similar testing has been done for just about every component leading up to the PCB creation, it is still necessary to continue testing. The group is literally creating a new board based on prior findings, so testing the PCB will be treated just as seriously (if not more so) to ensure data found meets the group's threshold. Only after testing of the PCB has been completed, and no concerning findings arise as a result of the testing, will the PCB be moved inside of the base of the Solar Powered Flower Sculpture?

## 6.8 Application Interactions Testing

The purpose of this testing phase is to determine how accurately the Solar Powered Flower Sculpture's mobile application functions. Each of the app's four modes will be tested to see how consistent they perform as desired. The ease of use of the application may be decided based on the reactions of human participants.

### 6.8.1 Testing Conditions

These tests assume that a secure Bluetooth connection has already been made between the mobile phone and Arduino microcontroller (the subject of a separate test phase). For the clock mode, a trial will consist of twelve different time, color, and brightness combinations, which are selected by the user. There will be three trials during testing. For the lamp and party modes, a trial will consist of all different color/pattern combinations being selected with different brightness settings; three trials will be run. After every trial, the system will be turned off and back on again with the power mode.

Five user participants with no prior awareness of our project or its mobile application may be chosen to test the application in a closed environment, based on their relative lack of computer/technology skills and availability. The testing will require them to complete a series of tasks relating to the app's four modes and its system interface. Tests may include navigating all four screens, being able to set the time on the clock, adjusting the color and brightness on the lamp, selecting a party pattern, and powering the solar sculpture on and off again. These tests will be done with the participants' free time and comfort prioritized, and results will be recorded for later evaluation.

## **6.8.2 Measuring Approach**

Regarding the four application mode tests, all results of the trials will be recorded for observation. For the application testing to be considered successful, it is required that zero faults are found within trials, including how the solar structure resumes its status after being powered back on from an off state. If an error is found during testing, and it is confirmed that the fault was not a result of human error while recording the test data, then it will be decided where the problem likely lies. This could require changes needing to be made to the Android Studio Java code or the Arduino microcontroller code. After changes have been made, and it is believed that the problem has been fixed, testing will resume from the beginning until all trials successfully pass.

The results of the five different user participants will be evaluated to determine if changes need to be made to the application interface. Instead of numerical statistics being compared, observational comments written during the testing phase will determine if something is completing a certain task on the app is particularly easy or difficult for users to perform. If concerning findings arise as a result of the testing, the app layout may be modified accordingly.

## **6.9 Final Testing Phase**

The purpose of this final testing phase is to determine the working functionality of the complete Solar Powered Flower Sculpture. This includes the finished PCB, with all connected components placed in their correct locations within the sculpture, and the mobile application. The purpose of this phase is primarily to judge aesthetic qualities of the finished design, as functionality of the device leading up to this phase is already assumed.

### **6.9.1 Testing Conditions**

Before testing, all features of the sculpture project would have had to be already implemented. This includes the addition of the 3D-printed petal cases encasing the LEDs, and the transportable flower base. It may also be necessary to include small structural or decorative additions to the sculpture, if it is felt they are needed

after analyzing it. The assumption being made is that no substantial additions to the project still need to be made at this point of testing. Having a final test period is reflective of verifying functionality and appearance of the sculpture before presentation. As a protective buffer, if problems are found, they can be corrected before signing off on the finished project design.

## **6.9.2 Measuring Approach**

All group members will be present to provide feedback on the sculpture. Data will be taken in the form of observations, with attention paid to any parts of the sculpture that appear suspicious. It may be necessary to test the sculpture in several different environments, such as indoors, outdoors at night, and outdoors during the day (with sun). This would be done to verify how the finished design cooperates with its solar panels. Ultimately, the group will decide if the Solar Powered Flower Sculpture meets all of the goals we set out to accomplish when the project was first developed, and that it is ready to present in a working state of functionality.



## 7.0 Administration

Information on all conflicts and issues, on a progress and barriers to progress had to be known by all of the group members. Any changes to the requirements or architecture had to be agreed upon by all of the group members.

## 7.1 Budgetary/Financial Data

Table 11: Estimated Budget for Sculpture

Parts for Sculpture		
Item	Quantity	Total Cost
LED Stripes	12	\$65.53
Photovoltaic Cells	6	\$17.05
PCB Manufacturing	1	\$35.00
3D Printed Petals	12	\$0.00
Wooden base (26.4 in x 26.4 in)	2	\$18.30
Wooden base (26.4 in x 4 in)	4	\$36.60
9 V Battery Pack	1	\$30.00
Rechargeable 9 V Battery	1	\$13.49

**Table 12: Estimated Budget for Testing**

<b>Parts for Testing</b>		
<b>Item</b>	<b>Quantity</b>	<b>Total Cost</b>
MAX639	1	\$16.49
MAX712	1	\$19.90
2N6107	1	\$6.95
1N4004	1	\$3.85
1N5817	1	\$1.79
Resistors	7	\$0.00
Capacitors	3	\$0.00
Electrolytic Capacitors	4	\$0.00
Inductor	1	\$0.00
Arduino Nano	1	\$7.99

## 7.2 Project Timeline

Table 13: Milestones for fall 2016

Fall 2016		
Description	Duration	Dates
Senior Design 1 Project Idea	1 week	August 22 - 26
Form Group / Project Idea	1 week	August 29 - September 2
Initial Project Document	1 week	September 5 - 9
Research / Begin Writing	3 weeks	September 12 - 30
Writing / Prototype Designing	3 weeks	October 3 - 21
Writing	2 weeks	October 24 - November 4
Table of Contents		November 4
Writing	1 week	November 7 - 11
Initial Draft		November 11
Code Development / PCB Design	2 weeks	November 14 - 25
Finish Document	1 week	November 28 - December 2
Review Document	1 day	December 5
Final Document		December 6

Table 14: Milestones for spring 2017

Senior Design II		
Description	Duration	Dates
Test Components	TBA	TBA
Build Prototype	TBA	TBA
Testing & Redesign	TBA	TBA
Finalize Prototype	TBA	TBA
Peer Presentation	TBA	TBA
Final Report	TBA	TBA
Final Presentation	TBA	TBA

### 7.2.1 Work Distribution

With three computer engineers and only one electrical engineer, distributing the work equally presented us with an initial challenge. No one was comfortable going out of their field of specialization. This was alleviated by a more or less shared workload approach. Outside of this, each group member was more or less assigned a specific role matching their skills, strengths, and specializations. Mahaley took the lead on all development involving hardware, power, and printed circuit board. To aid in the electrical design process, Kelechi bounced between helping oversee the electrical engineering roles and did some of the Android mobile application development. Kibwe took the lead on the aspects involving the microcontroller. Nickolas focused on the Android mobile application software.

The flow charts for the software development and the hardware development are found below. The software for the Solar Powered Flower Sculpture will be completed by Nickolas and Kibwe. The hardware for the Solar Powered Flower Sculpture will be completed by Mahaley and Kelechi.

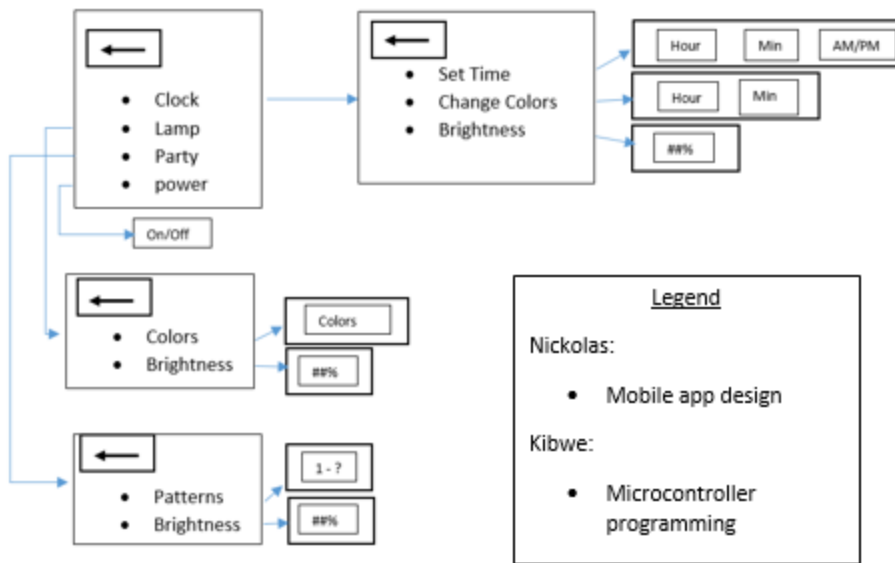


Figure 61: Software Logic Flowchart

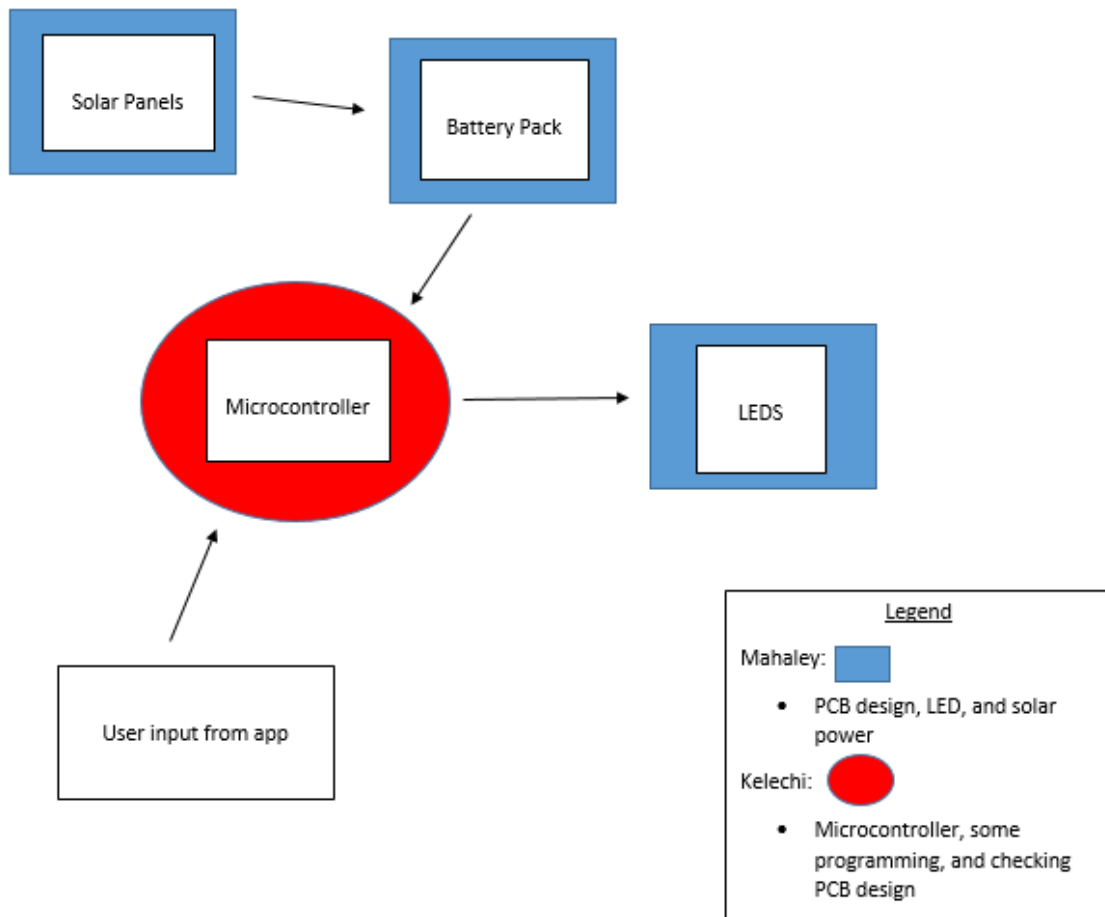


Figure 62: Hardware Block Diagram

## 7.3 Learned Materials and Skills

Throughout the research, design and developing phases of this project we were able to learn new materials, master new skills, and enhance ones that we gained through our education process here at the University of Central Florida. One of the most important skills that we gained was soldering. None of the group members had gained any experience with soldering through class work. This was a major concern going into the development and construction phase of the project, as that was a key skill needed to successfully complete the project.

Our research skills were definitely put to the test during the initial phase of this project. We had to relearn concepts that we hadn't used in a number of terms, and bring them together with a coherent understanding, as we compared various parts for inclusion into our project.

Typing was a skill that we had all neglected and is one of the skills least favored by people who go into the engineering field, we do engineering because we don't like having to write long papers. Each member didn't only have to brush up on their typing skills, but other skills needed to complete a technical document were also relied on to complete the first Senior Design document. We exercised our artistic skills and technical writing as we labored on our paper.

Mobile app development was a new skill for all of us as well. Some of us had tinkered with mobile app development from outside of class, but none of us had made a real app that did anything useful. That presented us with a unique challenge. Learning the skill quickly enough and sufficiently enough to get a working mockup by the end of the Senior Design 1 semester required the software developer to put in long hours researching and putting their new craft to work. Learning to adapt to the temperamental Android Studio, and manage it to cooperate was a new skill in its own right.

We have done some microcontroller programming in our labs, but nothing to the extent that this project required. Our labs were usually restricted to teaching the concepts and getting an onboard LED to blink or getting a digital display to show numbers. This introduced multiple components that we had to learn about as we were including them into the design of our project. The data manipulation was significantly more involved than our labs.

Working with wireless data transmission proved to be tricky for us. We had an idea of what a COM port was, but really didn't understand the nature behind a Bluetooth connection. Working with this project taught us the fundamentals of establishing a secure connection and troubleshooting the connection for any errors that might arise.

PCB might have been the most foreign on skills we had to learn in order to complete this project. When we first learned that we would have to design our own the group as a whole felt very intimidated by this challenge as we all had no idea of even where to begin to complete this task. Because it was so intimidating, this was one of the very first tasks we tackled. This required a lot of online research and visits to old professors as we learned why our initial designs would just kill our battery immediately. After a few weeks of trial and failure, the hardware team's proficiency and understanding of this material greatly increased, and we were able to get a viable design completed.

## 8.0 Conclusion

This document reflects the key points of our project and outlines its various functions. After completing this document, our group feels that we can successfully design and build the Solar Powered Flower Sculpture. Completion of the document also gave the group members a clearer picture and insight into what it will take to complete the project in a timely manner, and how each subsystem will have to work with each other. We feel that we are ready to move on the Spring semester of 2017 and complete our project. We spent a good portion of the Fall 2016 semester doing extensive research on components. A lot of our time went into delving deep into the finer details of how we would design and build each subsystem of the solar statue. After a slow start and some complications, we realized that good communication was the key to a successful project, and us reaching our goals.

During the research phase each member was able to learn a great deal about the components we will be working with, and were able to demonstrate a great deal of competence about the components that will be incorporated into our project, for each group member's fields of expertise and responsibilities. Each member was also able to show an overall understanding of the entire scope of the project. Each member let their skills shine during the design phase, allowing us to put together a very detailed design. The detailed outline of the design and construction gives each member a sense of their responsibility and accountability for the project.

The confusion and complications experienced at the onset of our project revolved around miscommunication and misunderstanding issues. Now we are able to work through our issues in a calm and professional manner. With familiarity, we became more comfortable with each other, and facilitated the improvement in communication. These improvements directly contributed to our group being able to combine our efforts and work as a team.

Throughout the project each team member had individual milestones that they had to reach. Each individual strived to reach their individual milestones, but would also be assisted by other group members when some assistance was needed. This strategy worked well, by helping us deal with any issues that had arisen, and kept us on schedule for meeting our deadlines. Throughout this whole process we were exposed to new material that was covered previously in the scope of our education at UCF. Through this experience we were able to develop new skill and techniques as well as learn new material not covered in our previous courses.

Now that we have a better idea of the scope and complexity of our project, we feel like our group can not only better anticipate potential issues and better prepare for them ahead of time, but also deal with them as they arise. As we move from the planning stage, into the prototyping stage, we plan on capitalizing on the momentum we built during the planning and research phase of this project. We are confident that we can meet our build requirements as well as add any features



we would like to include to enhance our design with the time remaining to us here at UCF. This has been a great opportunity for us to learn and grow. We feel fully confident that we are now prepared to move forward and complete our project.

## 9.0 Appendices

### Appendix A: References

1. "Can the Lead-acid Battery Compete in Modern Times?" *Basic to Advanced Battery Information from Battery University*. N.p., n.d. Web. 09 Nov. 2016.
2. "NSSN: ANSI Search Engine for Standards." *American National Standards Institute*, <https://www.nssn.org/search/IntelSearch.aspx>. Accessed 10 November 2016.
3. "Microcontrollers (MCU)." *Texas Instruments Incorporated*, [http://www.ti.com/lscds/ti/microcontrollers\\_16-bit\\_32-bit/msp/ultra-low\\_power/msp430frxx\\_fram/overview.page#MSP430FR2x](http://www.ti.com/lscds/ti/microcontrollers_16-bit_32-bit/msp/ultra-low_power/msp430frxx_fram/overview.page#MSP430FR2x) MCUs. Accessed 21 November 2016.
4. "Arduino/Genuino UNO." *Arduino*, <https://www.arduino.cc/en/Main/arduinoBoardUno>. Accessed 7 October 2016.
5. "Arduino MEGA 2560 & Genuino MEGA 2560." *Arduino*, <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>. Accessed 7 October 2016.
6. IgorF2. "Solar Charged Battery Powered Arduino Uno." *Instructables*, <http://www.instructables.com/id/Solar-Charged-Battery-Powered-Arduino-Uno/>. Accessed 7 October 2016.
7. Pmosconi. "Connect Arduino Arduino Uno to Android via Bluetooth." *Arduino*, <http://www.instructables.com/id/Connect-Arduino-Uno-to-Android-via-Bluetooth/>. Accessed 7 October 2016.
8. "Is Lithium-ion the Ideal Battery?" *Basic to Advanced Battery Information from Battery University*. N.p., n.d. Web. 10 Nov. 2016.
9. Sims, Gary. "I Want to Develop Android Apps - What Languages Should I Learn?" *Android Authority*, <http://www.androidauthority.com/want-develop-android-apps-languages-learn-391008/>. Accessed 3 December 2016.
10. Deyson. "How To: Create an Android App with Android Studio to Control LED." *Instructables*, <http://www.instructables.com/id/Android-Bluetooth-Control-LED/>. Accessed 12 October 2016.
11. Ford, Matthew. "pfod™ Parser Library for Arduino/non-Arduino microprocessors." *Forward Computing and Control Pty. Ltd.*, <http://www.forward.com.au/pfod/pfodParserLibraries/index.html>. Accessed 12 October 2016.
12. Ford, Matthew. "pfodDesignerV2™ Arduino/non-Arduino menus for Beginners Design Custom menus to switch Arduino outputs on and off Absolutely No Programming Required." *Forward Computing and Control Pty. Ltd.*, <http://www.forward.com.au/pfod/pfodDesigner/OnOffExample/index.html>. Accessed 12 October 2016.
13. Ford, Matthew. "pfodApp™ V2.0.1+ for Arduino™ V2.1+." *Forward Computing and Control Pty. Ltd.*, [http://www.forward.com.au/pfod/Android\\_pfodApp/pfodAppForAndroidGettingStarted.pdf](http://www.forward.com.au/pfod/Android_pfodApp/pfodAppForAndroidGettingStarted.pdf). Accessed 12 October 2016.
14. "Battery and Energy Technologies." *Nickel Cadmium NiCad Batteries*. N.p., n.d. Web. 10 Nov. 2016.
15. Moorthi, Mumu. "Field Experience with Large Nickel Metal Hydride (NiMH) Batteries in Stationary Applications." *INTELEC 06 - Twenty-Eighth International Telecommunications Energy Conference (2006)*: n. pag. Web.
16. Knier, Gil. "How do Photovoltaics Work?" *NASA*. NASA, n.d. Web. 10 Nov. 2016. <https://science.nasa.gov/science-news/science-at-nasa/2002/solarcells>.

17. "Poly (vinyl Chloride)." *Polymer Science Learning Center*. N.p., n.d. Web. <http://www.pslc.ws/macrog/pvc.htm>.
18. "Android Developers." *Creative Commons Attribution 2.5*. <https://developer.android.com/design/index.html>. Accessed 16 November 2016.
19. Kathy. "What Is Melamine? A Crash Course." *SafeMamacom*. N.p., 07 Oct. 2008. Web. 16 Nov. 2016.
20. "Types of Plastic." *Types of Plastic*. N.p., n.d. Web. 16 Nov. 2016.
21. Westhues, Jonathan. "Microcontroller Interfacing." *cq.cx*, N.p., Nov. 2006. Web. 21 Nov. 2016. <http://cq.cx/interface.pl>.
22. "Interfacing with Hardware." *Arduino*, <http://playground.arduino.cc/Main/InterfacingWithHardware#LED>. Accessed 21 November 2016.
23. "Teak." *The Wood Database*. N.p., n.d. Web. 29 Nov. 2016.
24. "Ipe." *The Wood Database*. N.p., n.d. Web. 29 Nov. 2016.
25. "Bluetooth Core Specification." *Bluetooth SIG Inc.*, <https://www.bluetooth.com/specifications/bluetooth-core-specification>. Accessed 21 November 2016.
26. "Facts About Oak Wood - Interesting Facts." *Interesting Facts*. N.p., 21 Aug. 2014. Web. 01 Dec. 2016.
27. "Bluetooth Architecture." *Tutorial-Reports.com*. <http://www.tutorial-reports.com/wireless/bluetooth/architecture.php>. Accessed 21 November 2016.
28. "Testing a Breadboard for Continuity and Isolation." *Simply Smarter Circuitry Blog*. <https://www.circuitspecialists.com/blog/testing-a-breadboard-for-continuity-and-isolation/>. Accessed 21 November 2016.
29. "IEEE 802.11." *Wikipedia*. Wikimedia Foundation, n.d. Web. 05 Dec. 2016
30. "Panasonic RF Modules." *Panasonic RF Modules | Mouser*. N.p., n.d. Web. 05 Dec. 2016
31. "Wi-Fi." *Advantech*. N.p., n.d. Web. 05 Dec. 2016
32. "Difference between Bluetooth and Wi-Fi." *Superadmin*. N.p., n.d. Web. 05 Dec. 2016.
33. "RF Module." *Wikipedia*. Wikimedia Foundation, n.d. Web. 05 Dec. 2016.
34. "Bluetooth." *Android Developers*. N.p., n.d. Web. 05 Dec. 2016.
35. "Electric Shock." *WebMD*. WebMD, n.d. Web. 05 Dec. 2016.
36. "The Fatal Current." *Electrical Safety*. N.p., n.d. Web. 05 Dec. 2016.
37. "BU-401a: Fast and Ultra-fast Chargers." *Battery University*. N.p., n.d. Web. 05 Dec. 2016
38. Expert, Home Services. "Electrical Safety Tips: How to Put Out an Electrical Fire." *Electrical Safety Tips*. N.p., n.d. Web. 05 Dec. 2016.
39. "Arduino Bluetooth Interfacing - Simplest Tutorial for Bluetooth Basics." *DIY Hacking*. N.p., 04 Oct. 2016. Web. 05 Dec. 2016.
40. "How It Works." *Bluetooth Technology Website*. N.p., n.d. Web. 05 Dec. 2016.
41. "Bluetooth / 802.15.1 Modules." *Mouser*. N.p., n.d. Web. 05 Dec. 2016.
42. "The Difference between Bluetooth 1.1 1.2 2.0 2.1 Module." *EBay*. N.p., 21 Sept. 2008. Web. 05 Dec. 2016.
43. "What Is the Difference between Bluetooth 4.0, 3.0 and 2.0?" *Quora*. N.p., n.d. Web. 05 Dec. 2016

44. "Bluetooth Basics." *Bluetooth Basics*. N.p., n.d. Web. 05 Dec. 2016.
45. "Our History." *Bluetooth Technology Website*. N.p., n.d. Web. 05 Dec. 2016.
46. "What Different with Bluetooth V3.0 & V4.0?" *Asus*. N.p., 01 Sept. 2015. Web. 05 Dec. 2016.
47. "Bluetooth Low Energy." *Wikipedia*. Wikimedia Foundation, n.d. Web. 05 Dec. 2016.
48. "The Effective Range of Bluetooth." *Techwalla*. Leaf Group LTD., n.d. Web. 05 Dec. 2016.
49. Gilbert, David. "Bluetooth 5.0: Here's Why The New Wireless Standard Matters." *International Business Times*. N.p., 17 June 2016. Web. 05 Dec. 2016.
50. "Bluetooth Low Energy vs. Classic Bluetooth: Choose the Best Wireless Technology For Your Application." *Medical Electronics Design RSS*. N.p., n.d. Web. 05 Dec. 2016.
51. "Bluetooth Core Specification." *Bluetooth SIG, Inc*. N.p., n.d. Web. 05 Dec. 2016.
52. "Bluetooth Basics." *Bluetooth SIG, Inc*. N.p., n.d. Web. 05 Dec. 2016.
53. "Physical Layer." *Wikipedia*. Wikimedia Foundation, n.d. Web. 05 Dec. 2016.
54. "Datasheet Bluetooth to Serial Port Module HC05." *Electronica 60 Norte*, <http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>. Accessed 5 December 2016.
55. "Bluetooth 4.0 BLE Module Datasheet." *JNHuaMao Technology Company*, [http://fab.cba.mit.edu/classes/863.15/doc/tutorials/programming/bluetooth/bluetooth40\\_en.pdf](http://fab.cba.mit.edu/classes/863.15/doc/tutorials/programming/bluetooth/bluetooth40_en.pdf). Accessed 5 December 2016.
56. "RN-42/RN-42-N Data Sheet." *Roving Networks*, <http://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/Bluetooth-RN-42-DS.pdf>. Accessed 5 December 2016.
57. "Product Data Sheet." *Guangzhou HC Information Technology Co., Ltd.*, <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>. Accessed 5 December 2016.

## Appendix B: Data Sheets

1. DATASHEET BLUETOOTH TO SERIAL PORT MODULE HC05  
<http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>
2. JNHuaMao Technology Company Bluetooth 4.0 BLE module Datasheet  
[http://fab.cba.mit.edu/classes/863.15/doc/tutorials/programming/bluetooth/bluetooth40\\_en.pdf](http://fab.cba.mit.edu/classes/863.15/doc/tutorials/programming/bluetooth/bluetooth40_en.pdf)
3. RN-42/RN-42-N Data Sheet  
<http://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/Bluetooth-RN-42-DS.pdf>
4. Guangzhou HC Information Technology Co., Ltd. Product Data Sheet  
<https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>