Developer Guide

Section 1: System Overview

The purpose of this project is to make a 5 by 5 by 7 LED matrix, in which each LED is able to be programmed individually to make animations. Furthermore, each LED is capable of 27 colors. The system works continuously, and has three programmable animations in addition to three that are pre-programmed. The system is compatible with any bluetooth device, and can switch between animations either via bluetooth or the graphical user interface, which is a Python script. This Python code is capable of uploading files to the Arduino Uno, which allows the customizable programs to be modified. The graphical user interface also includes a simulation feature, allowing users to preview their custom animations on the computer screen. Areas of improvement for the future include image and video parsing, and the ability to show smartphone app notifications when the user receives one. The system operates using a printed circuit board (PCB) and an Arduino Uno.

Section 2: Electrical Specifications

The system can be powered in either of two ways. The first is by powering the Arduino Uno using a standard 5 V USB cable. The second is by using the Arduino's built-in barrel jack, for which the recommended voltage is 7-12 V. The system draws a maximum power of 0.05 W. This is calculated when all 5 LEDs in a row (only one row is on at an instant) are white. This is well below the rating of the shift registers and the Arduino. Other relevant specifications are given in Table 1.

Table 1. Electrical Specifications

Component(s)	Voltage	Current	Power
Green & Blue LEDs	3.0 V	0.90 mA	2.7 mW
Green & Blue Resistors	2.0 V	0.90 mA	1.8 mW
Red LED	2.0 V	1.4 mA	2.8 mW
Red Resistor	3.0 V	1.4 mA	4.2 mW
System Total	5 V	10 mA	50 mW
USB Input	5 V	Maximum 200 mA	1 W
Barrel Jack	7 - 12 V	Maximum 200 mA	Maximum 1.4 W
Shift Register Inputs	Maximum 6 V	Maximum 70 mA	Maximum 0.42 W

Section 3: User Guide

This manual includes instructions to set up and operate the system. This is broken down into three parts: designing custom animations, changing the animation via the GUI, and changing the animation via bluetooth. Prior to these steps, the user needs to have a Windows OS with the command 'arduino' added to his or her PATH Environment Variables. They also need to have downloaded the files to their computer and they cannot alter the file structure inside the folder, although it will work regardless of the location of the folder. They also need to have installed a Python shell, whether that is IDLE, VSCode, Anaconda, or something else. While many of these programs have basic Python libraries pre-installed, the user may need to download the following Python libraries: tkinter, sys, numpy, PySerial, os, and time.

Section 3.1: Designing Custom Animations

1. Run visualizer.py, all windows will pop up in the upper left corner of your screen.
2. If you have not plugged the LED cube into a USB port, do so now.
3. Click 'Update COMs' and select the box next to the Arduino's COM port. This may not be obvious if the Arduino was not manufactured by an official Arduino manufacturer. When the box has been clicked, you will notice that the 'Animation' buttons will no longer be grayed out.
4. Hit Enter on your keyboard, click okay, or x out of the pop up.
5. Click 'Advanced Settings' to open the customization screen.
6. You may adjust the sliders for max frame count and speed at any time.
7. You may also click on the Animation button at the top to select which animation to customize. The default is animation 4. **Be warned that any animation not submitted before closing the GUI will be lost**.
8. Edit the animation, note that the color of the buttons reflects the color shown in the animation. **Be warned that you can overwrite any color in animation 4, 5, and 6 all day, but there is no undo button.** Animation Settings are however saved while the GUI is open.
    a. Manually
        i. Choose a color by clicking anywhere on a color in the "Possible Colors" list. A color has been selected if the box is checked. **Be warned that the animation cannot be manually changed without a color selected.**
        ii. Choose a layer by sliding the "Choose Layer" slider. The layers are stacked vertically (bottom to top).
        iii. Choose a frame by sliding the "Choose Frame" slider.
        iv. Click a button in the "Select LEDs Below" grid. Reminder, the bottom is toward you and the back is away from you (left/right are your left/right).
        v. Repeat i-iv to your heart's content.
    b. Command aided
        i. Choose an effect by sliding the "Effect" slider. 0 -> layer changes, 1 -> frame changes, and 2 -> whole animation changes
        ii. If layer level changes are being made complete step 8.a.ii and 8.a.iii, if frame level changes are being made complete step 8.a.ii, finally if animation level changes are being made ignore this step.

iii. If Sequential, Random, or clear are desired click one, otherwise click a color before clicking set color.
iv. Repeat i-iii to your hearts content.

9. It is advisable to demo the animation before submitting (though not necessary). To demo an animation on the GUI, select the 'Play Animation' button, which will take you to a black screen with buttons at the top.
    a. Selecting 'Play Animation' on this window will start the simulation. You may pause as needed with the "Pause Animation" check box. A checked box means it is paused and an unchecked box means it is playing. You may also stop the animation by clicking the "Stop Animation" button.
    b. Click 'Advanced Settings' to return to the advanced setting window.
10. When the animation(s) are as desired, click 'Submit.' An Arduino window will appear and close when the new animations are finished uploading. The LED Matrix will reset to the default rainbow color cycle animation. There will be a message that shows that all the animations have been uploaded. Type enter, click okay, or click the X to continue.
11. You may then click 'Basic Settings' to return to the original screen, from which the animation on the matrix can be selected.
12. Enjoy!

Section 3.2: Selecting an Animation from the GUI

1. Navigate to the 'Basic Settings' panel.
2. If you have not already done so, select the correct COM port from the list on the right. The animation buttons will no longer be grayed out. You may need to click 'Update COMs' button if the correct port does not appear.
3. Then you may click on the desired animation from the list:
    a. Animation 1: Rainbow Color Cycle
    b. Animation 2: Layers lighting up white, in sequence from bottom to top
    c. Animation 3: Vertical column swirl around the cube edge.
    d. Animations 4-6: Customizable animations
4. Click 'Okay' or close the pop-up window informing you of the animation choice.
5. Enjoy!

Section 3.3: Selecting an Animation over Bluetooth

1. Download a bluetooth serial application on your smartphone.
2. Connect to the HC-05 at a 9800 baud rate.
3. Send the value of the animation you want to play minus one (animations 1-6 are numbers 0-5). Send the value as a character with no line ending.
4. Enjoy!

Section 4: Design Artifact Figures

All electrical components are located inside of the enclosure. There are three inputs to the system, the GUI, Bluetooth, and power. The system can be controlled by either the GUI or by Bluetooth. All data will be parsed within the Microcontroller and eventually sent to the logic unit. The logic unit will translate the unique electrical input that it is receiving and be able to pinpoint a specific row of LEDs to light up precisely as was commanded. The only output of the system is an LED animation.

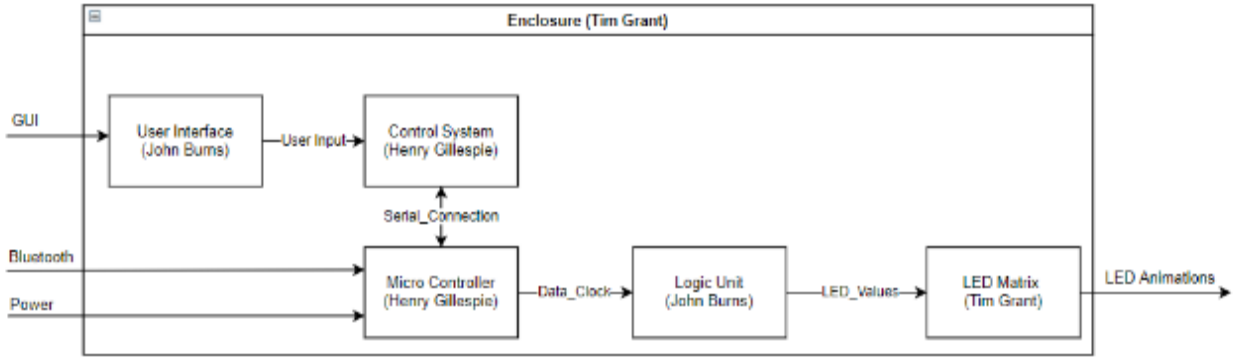

Figure 1: Block Diagram

The schematic was developed in Kicad. The electrical outputs that will be going to the LED matrix can be seen on the edges labeled J1-J13. The inputs to the PCB will be in the section with the label Microcontroller Connections. The inputs will come in from the Microcontroller, based on the input that the microcontroller sends, the logic unit will be able to light a row of LEDs based exactly how the microcontroller specified.

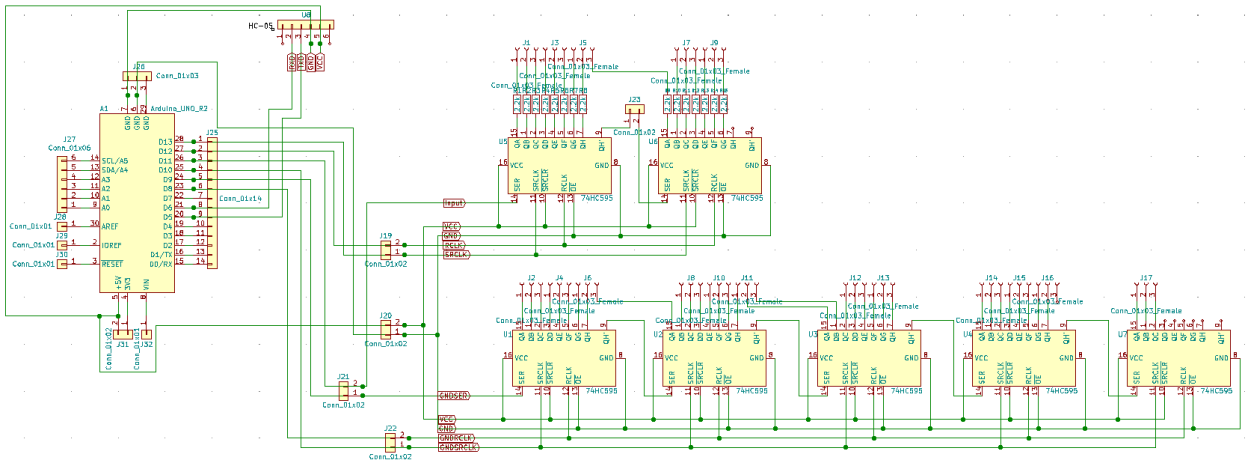

Figure 2: Full Schematic

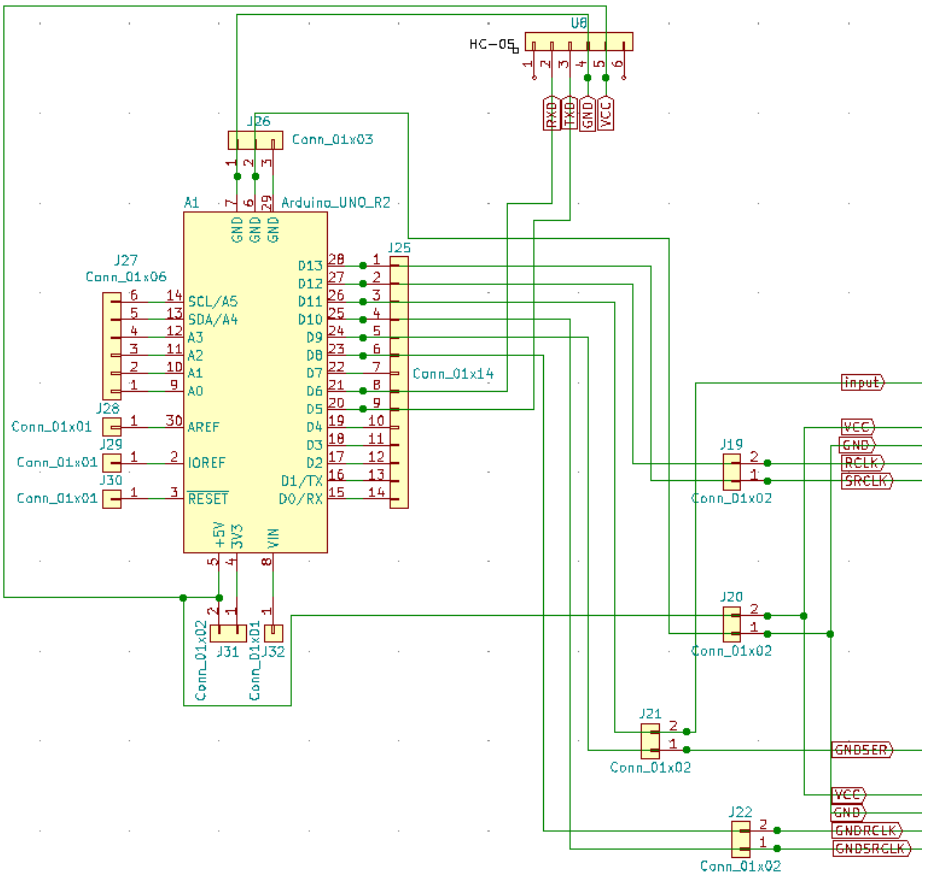

Figure 3: Left half of Schematic

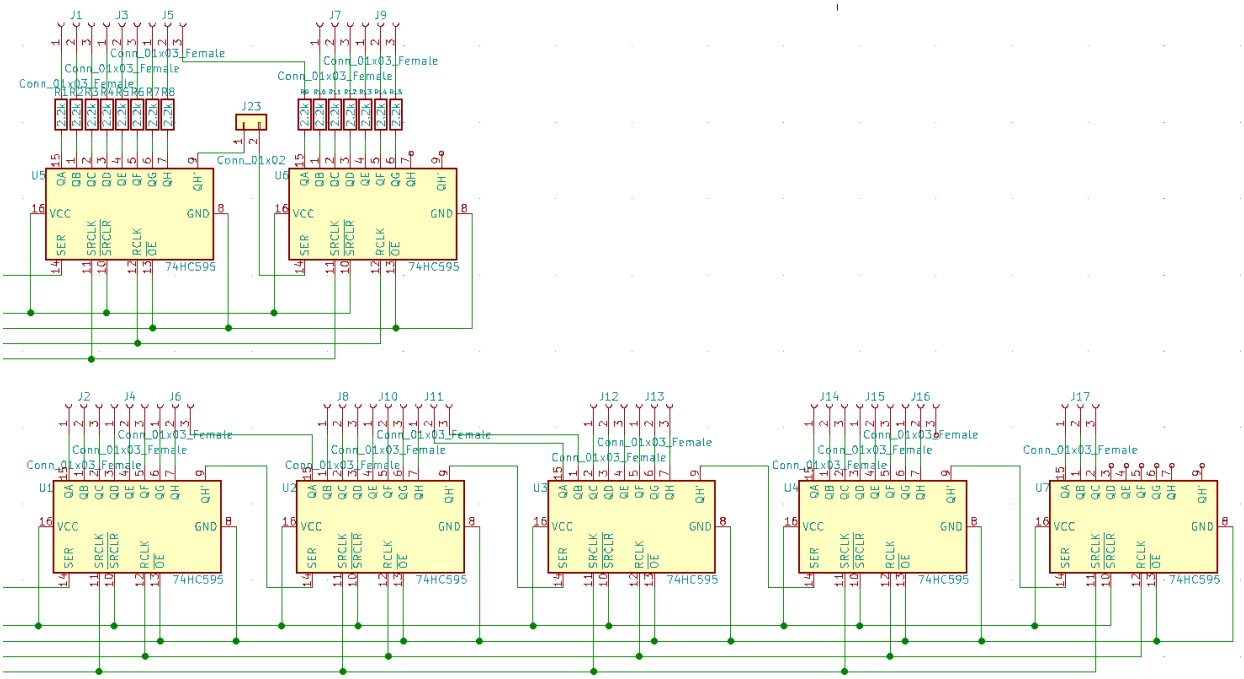

Figure 4: Right half of Schematic

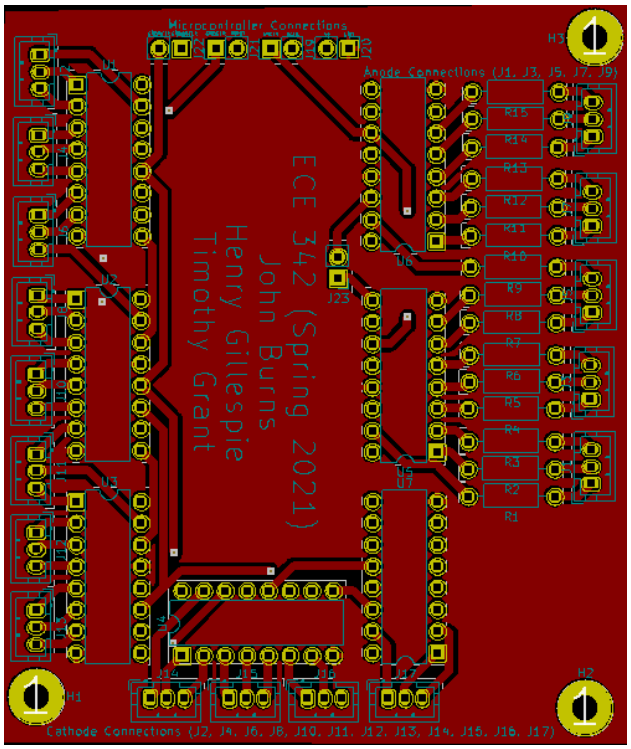

Figure 5: Top Layer of PCB

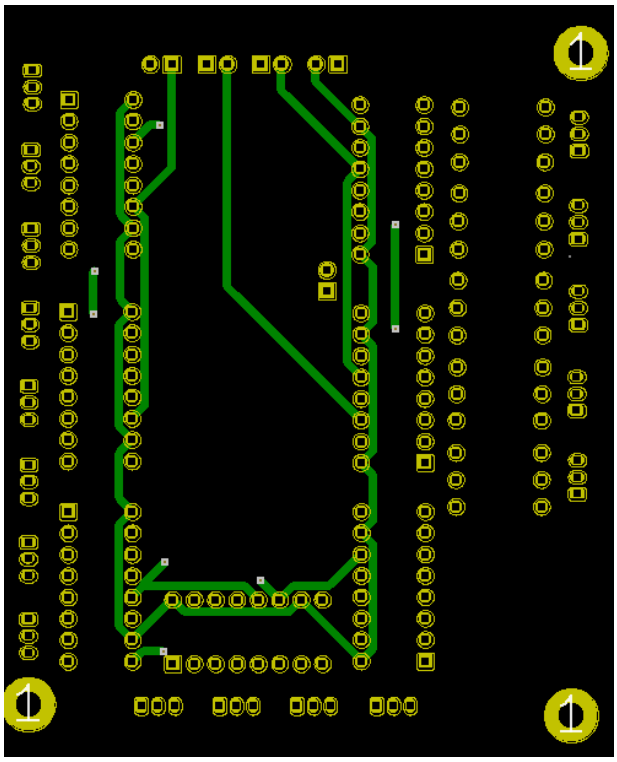

Figure 6: Bottom Layer of PCB

The enclosure base is built out of wood. The top of the enclosure is made out of plexiglass, to allow for clear unobstructed view of the LED matrix, while also allowing for protection of the Matrix from any outside objects that might come into contact with it. The hole in the side of the enclosure is where the USB can be connected and disconnected. The bottom of the wooden base is removable so that it is possible to open up the enclosure to rework any of the electronics located inside.

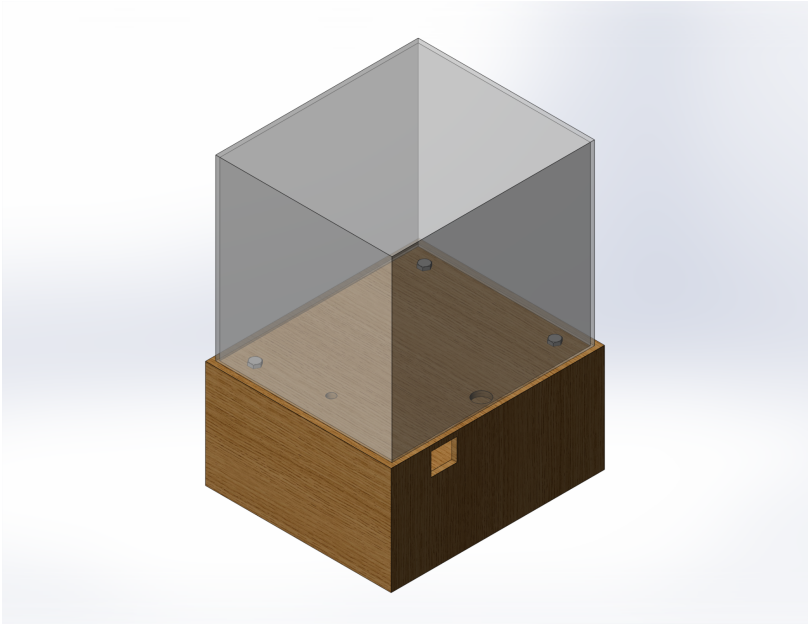

Figure 7: Model of Enclosure

Section 4.1: PCB Information

The PCB schematic was created by using a software called KiCad. KiCad allows the user to create pathways from component to component. Then from those schematics the physical PCB was manufactured by a company called JLCPCB. The PCB is 73 mm wide and 86 mm long. Some specific things of note are the 0 shaped formation of components, the holes, and the notes. The formation of the components was done this way to not only make the design more compact, but also keep its original clarity. The holes were added to allow for easy mounting. Finally, the notes are shown to help constructors understand what each section is. As a side note, J23 was included to make sure if something went wrong, there was a way to fix it without having to get a new board.

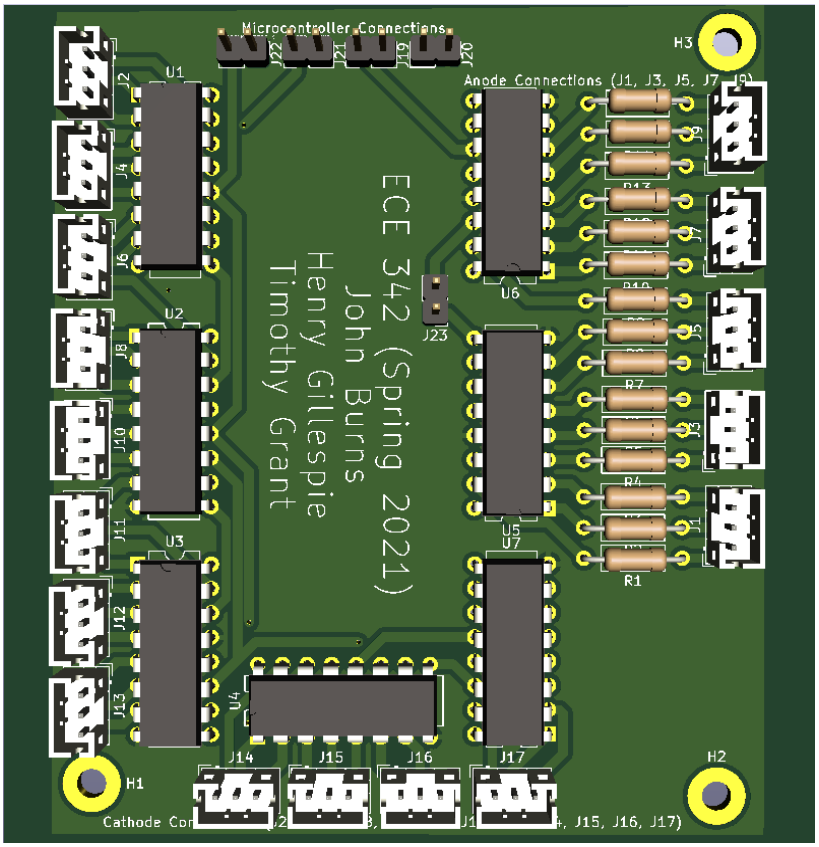

Figure 8: Layout of PCB

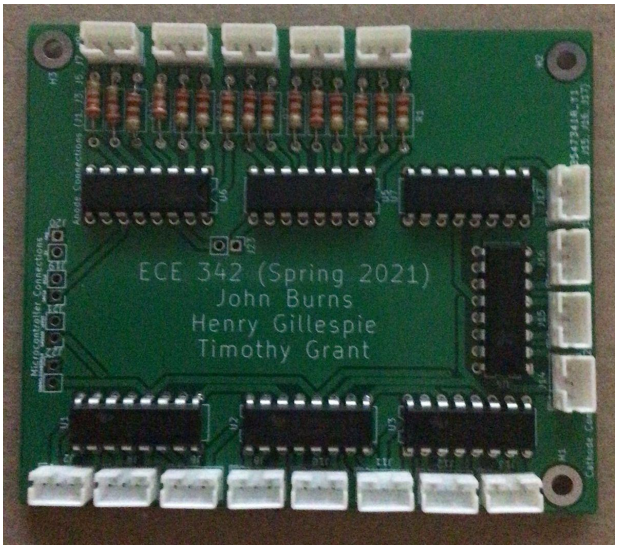

Figure 9: Picture of PCB

Section 4.2: Part Information

Table 2: Basic Information BOM

Material	Designation	Name	Quantity
1	N/A	Plywood	1
2	N/A	Bolt	3
3	N/A	¼-20 Nut	6
4	N/A	Wing Nut	3
5	N/A	Machine Screw	5
6	N/A	#10-32 Nut	5
7	N/A	#10 Wood Screw	3
8	N/A	Corner Brace	3
9	N/A	Gorilla Glue	0.01
10	N/A	RGB Common Cathode LED	175
11	N/A	PCB	1
12	A1	Arduino Uno w/ USB Connector	1
13	U8	Wireless Bluetooth Module	1
14	R1-R15	2.2kΩ Resistor	15
15	U1-U7	8-bit Shift Register	7
16	J19-J23	Pin Header	10
17	J1-J17	Connector Pin	17
18	N/A	Jumper Wires	62
19	N/A	Roll of Silver Plated Copper Wire (26 gauge, 100ft)	1
20	N/A	Roll of Solder (Lead Free, 100ft)	2

Table 3: Manufacturing Information BOM

Material	Manufacturer Number	Manufacturer	Value
1	NA	Home Depot	N/A
2	800080	Everbilt	N/A
3	801736	Everbilt	N/A
4	802371	Everbilt	N/A
5	803331	Everbilt	N/A
6	800272	Everbilt	N/A
7	807491	Everbilt	N/A
8	13619	Everbilt	N/A
9	NA	Home Depot	N/A
10	ED_YW05_RGB-4P-C_100Pcs	EDGELEC	N/A
11	NA	JLCPCB	N/A
12	A000052	Arduino	N/A
13	B01MQKX7VP	NewZoll	N/A
14	294-2.2K-RC	Xicon	2.2kOhm
15	74LV595N,112	NXP USA Inc.	N/A
16	826629-2	TE Connectivity	N/A
17	B3B-EH-A(LF)(SN)	JST	N/A
18	825	ADAFRUIT	N/A
19	CWIR-S003	KBeads	N/A
20	D96SCF192	MULTICORE (SOLDER)	N/A

Table 4: Material Specifications BOM

Material	**Physical Dimensions (L x W x H)**
1	8” x 48”
2	¼”-20 (Thread) x 4.5” (L)
3	¼”-20 (Thread)
4	¼”-20 (Thread)
5	#10-32 x 1”
6	#10-32
7	#10 x ¾”
8	1” x 1”
9	NA
10	5mm * 5mm * 29.5mm
11	73mm * 85mm
12	68.6mm * 53.4 mm
13	37.5mm * 16.5mm * 4mm
14	5.00mm x 12.00mm
15	20mm * 5mm
16	2.5mm * 2.5mm * 8mm
17	7mm * 5mm
18	205mm length
19	100ft * (0.157in diameter)
20	2m length