Modular LED Wall Panels: Design Document

Team 29: Adam Chung, James Prince, Kyle Salzberg
ECE 445
Professor: Victoria Shao
TA: Hanyin Shao

March 4, 2022

Contents

1 Introduction **3**
1.1 Problem . . . 3
1.2 Solution . . . 3
1.3 Visual Aid . . . 4
1.4 High-level Requirements . . . 5

2 Design **6**
2.1 Block Diagram . . . 6
2.2 Schematic . . . 7
2.3 PCB Design . . . 8
2.4 Power . . . 9
2.4.1 Description . . . 9
2.4.2 Requirements and Verification . . . 9
2.5 Core Tile Compute Unit . . . 9
2.5.1 Description . . . 9
2.5.2 Requirements and Verification . . . 10
2.6 Expansion Tile Compute Unit . . . 10
2.6.1 Description . . . 10
2.6.2 Requirements and Verification . . . 11
2.7 LED Array . . . 11
2.7.1 Description . . . 11
2.7.2 Requirements and Verification . . . 11
2.8 Expansion Port . . . 11
2.8.1 Description . . . 11
2.8.2 Requirements and Verification . . . 11
2.9 Tolerance Analysis . . . 12
2.10 Software Flowchart . . . 13

3 Cost and Schedule **14**
3.1 Cost . . . 14
3.1.1 Labor . . . 14
3.1.2 Parts . . . 14
3.2 Schedule . . . 15

4 Ethics and Safety **17**
4.1 Ethics . . . 17
4.2 Safey . . . 17

1 Introduction

1.1 Problem

In recent years, LED decorations and the IoT marketplace have grown immensely in popularity. However, many of the commercially available products are either overpriced or provide very little customizability. Currently, most LEDs are only available as strips, with the few purchasable modular LED panels unable to display text or images. The linearity of LED strips hinders the user's ability to create 2-D displays tailored to their specific room dimensions. On the other hand, the lack of text and images on current modular LED panels restricts the possibilities for dynamic displays that can provide useful information.

1.2 Solution

We will design and implement multiple modular LED panels that are capable of displaying customizable text or images. The modular design allows for the user to connect together as many panels as desired, in relatively any shape (excluding diagonally). For example, with four tiles, the user can create either a 2x2 square, a 4x1 line, or an "L" shape.

The design is focused around a "core panel" containing the main control unit, which can then be connected to "expansion panels" in the manner described above. The control unit will determine the configuration of the 4 panels, deciding the manner in which to place the text/images (e.g. text will scroll if too small to fit on one panel). Each panel will contain an array of 8x8 serially addressable LEDs which can be illuminated in any color. Utilizing a smartphone, the user will be able to choose what image, text, or dynamic display will be shown on the panels. At all times, the central panel will recognize the overall configuration, updating the display to fit within the set boundaries.

We also see our project as a great opportunity to create a platform for artists to create and design LED displays and designs. By implementing an API, any artist will be able to easily code, most likely in Python, any design of their choosing.

1.3 Visual Aid

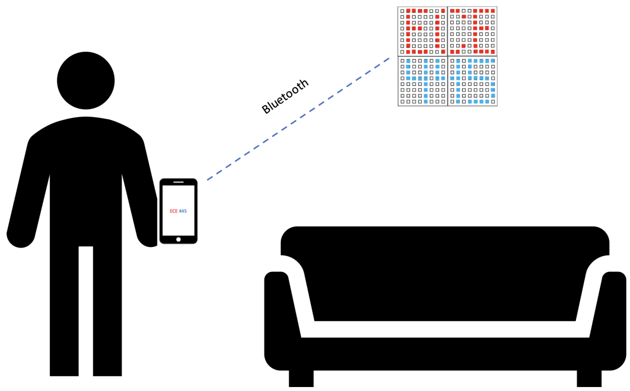

Figure 1: Configuration example 1 of four panels

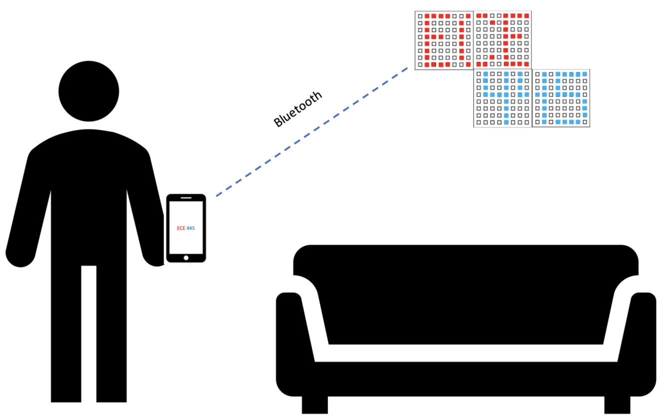

Figure 2: Configuration example 2 of four panels

1.4 High-level Requirements

- The four panels must be able to display text, images, and dynamic effects that can adapt to current configuration. If text is too small to fit within boundaries, it will scroll across panels.
- "Core panel" must automatically recognize any change to the configuration of expansion tiles, updating the display output to each tile to fit within the new boundaries in under 1 second.
- Panels must be able to be controlled through Bluetooth by a smartphone or other external device (must connect to Bluetooth device within 5-7 seconds).

2 Design

2.1 Block Diagram

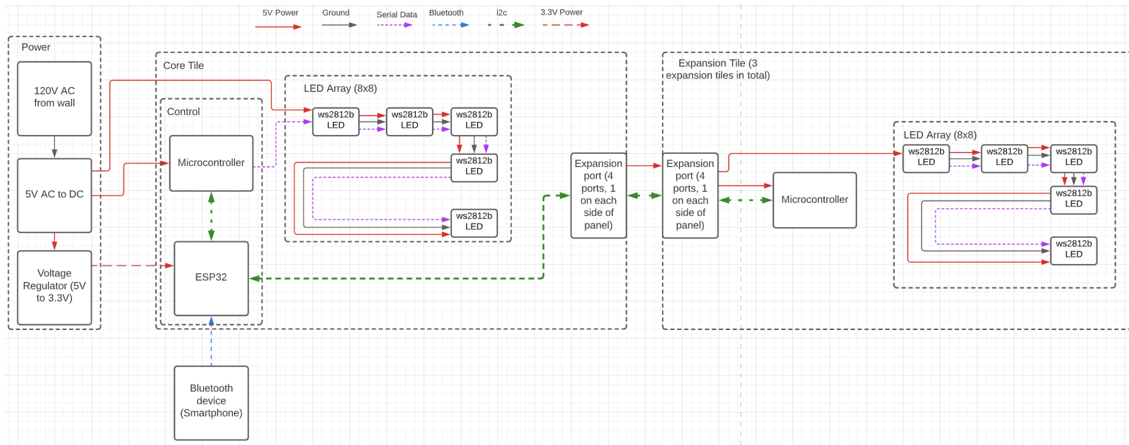

Figure 3: Block Diagram

2.2 Schematic

The following is the schematic for our design. We note that there is MOSFET [1] logic used to implement a level shifter. Since the ESP32 [2] runs on 3.3V and the ATMEGA [3] runs on 5V, we need this level shifter to successfully implement I2C. This logic is modeled off of the Philips Semiconductors Application Note AN97055 [4]

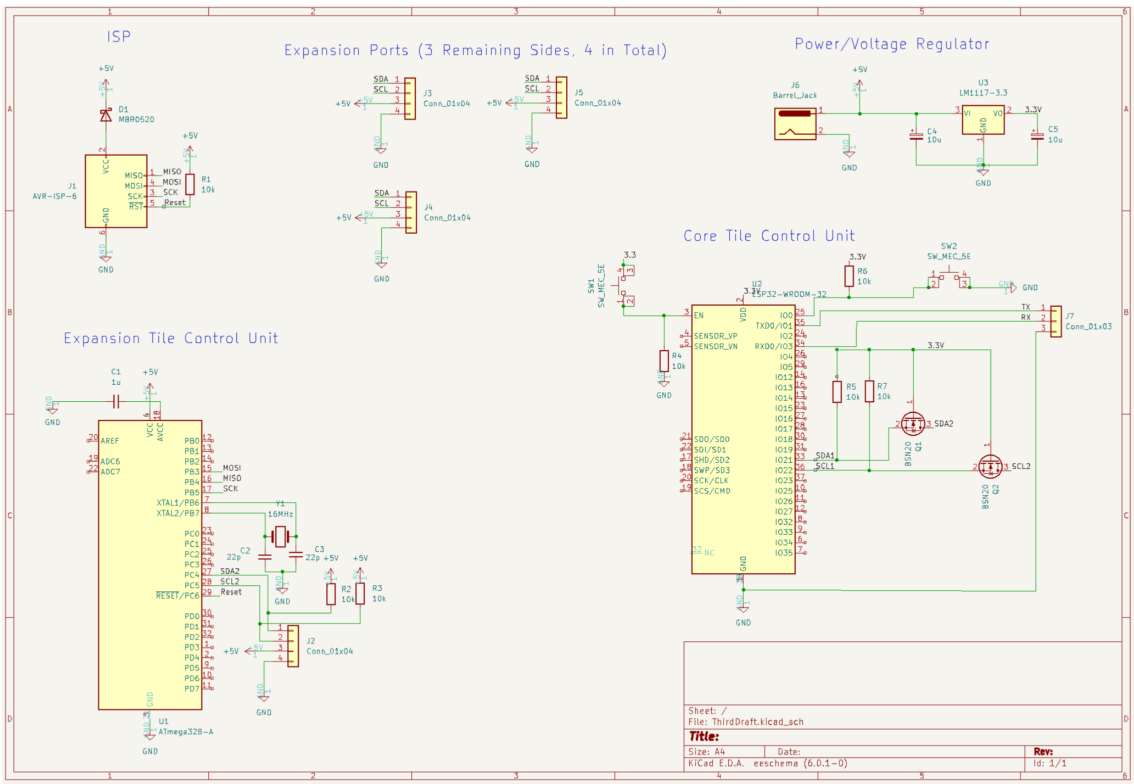

Figure 4: Schematic

2.3 PCB Design

The following is the first version PCB design for the core panel. The expansion panels' PCBs will have the same design, except the ESP32 will be removed, along with the buttons and external pins associated with the ESP32.

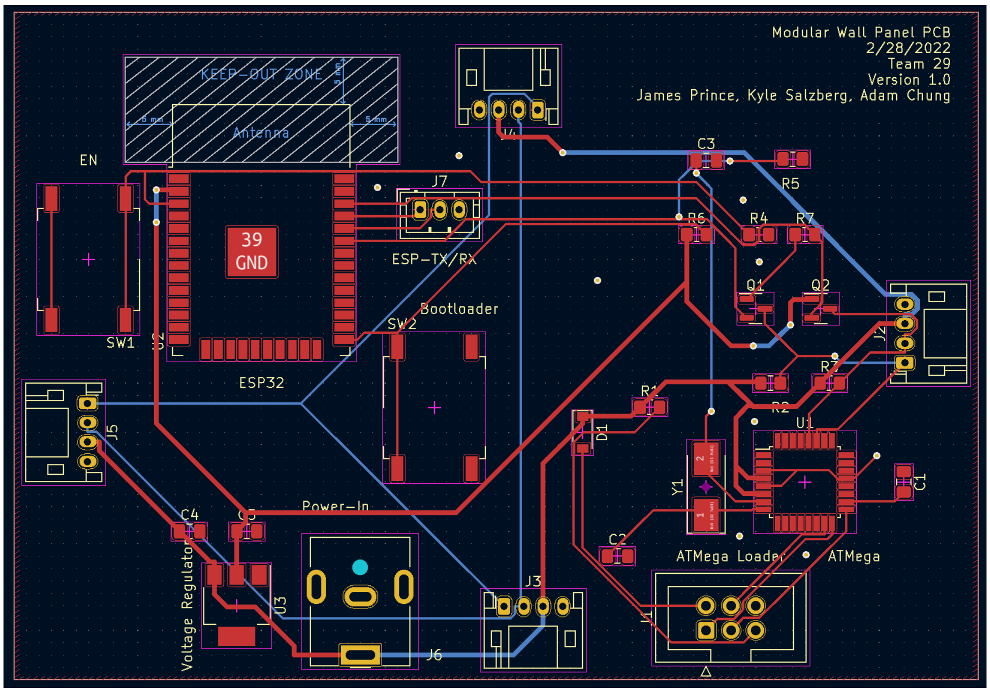

Figure 5: Version 1 Core Panel PCB Board Design

2.4 Power

2.4.1 Description

The power subsystem will be plugged into a standard North American outlet which outputs 120 V AC rated at 15 A. Utilizing a power adapter we will step down the 120 V to a 5 V DC supply. Additionally, we will use a voltage regulator [5] to step down the 5V supply to 3.3V which is the necessary VCC for the ESP32.

2.4.2 Requirements and Verification

Requirements	Verification
1. Our core tile must plug into a standard wall outlet (120V) using a reasonably sized adapter (ie phone/ laptop charger size) providing 5±0.5 V	1A. Use multimeter to confirm steady 5v output
2. Voltage regulator must provide 3.3±0.3 V to power the microcontroller	2A. Use an oscilloscope to verify that the voltage regulator is outputting 3.3±0.3 V
3. Core tile must distribute power between tiles, providing 5±0.5 V	3A. Verify that when all tiles are connected, each receives 5±0.5 V through oscilloscope measurements of voltage at input pins of expansion ports.
4. Current provided by voltage regulator must provide at least 500mA	4A. Use multimeter to ensure 500mA is being provided to the ESP32.

2.5 Core Tile Compute Unit

2.5.1 Description

The Core Tile Compute Unit is responsible for communication with external devices to receive text and images to display on all the panels. It also detects and updates the layout of all the core and expansion panels, and processes the LED outputs for each of the panels based on the current configuration. The ESP32 will facilitate the Bluetooth connection between our product and the user's smartphone. In addition, the ESP32 on the core tile unit will control the overall image being displayed on all the panels, and through I2C, will send a series of data packets to each microcontroller. The data packets will contain the data that is to be displayed (image/text) on the LEDs. Additionally, the ESP32 will poll the microcontrollers on the expansion panels to determine the current configuration of all panels, adjusting the image output accordingly. Finally, the core tile should process and display the desired image/text to its own LEDs since it itself is also a display panel with an LED matrix.

2.5.2 Requirements and Verification

Requirements	Verification
1. Must be able to establish Bluetooth connection with smartphone within 5-7 seconds	1A. Connect the phone to the compute unit. 1B. Send Bluetooth serial data from the phone to the ESP32. 1C. Verify ESP32 receives input and then sends data to expansion tile microcontrollers through I2C.
2. Must be able to process the algorithm to determine overall tile layout within 1 second.	2A. Verify display output with any arbitrary tile layout as specified in verification step 1A, 1B, and 1C. 2B. Update the tile layout after powering down the system. 2C. Repeat step 2A and verify the display output has adapted for the updated panel layout.
3. Must be able to process and send texts and images for each connected expansion tile within 30ms	3A. Connect one or more Expansion Units to the Core Unit and power the device. 3B. Connect the core panel to a Bluetooth device and send text and images as listed in step 1A. 3C. Verify LED output across panels and check each panel has texts and images correctly displayed across multiple panels.
4. Must be able to synchronize all the Expansion Units and refresh the LED output at 30 Hz.	4A. Use a slow-motion video to verify frames being updated at 30 Hz.

2.6 Expansion Tile Compute Unit

2.6.1 Description

Each expansion panel's microcontroller will receive a stream of packets through I2C from the core panel. Each tile will parse through the data stream and display its image on the LEDs.

2.6.2 Requirements and Verification

Requirements	Verification
1. Must receive data from either core tile or another Expansion Tile and parse contents	1A. Check on software the bit stream received is correct
2. Must be able to pass on data to other Expansion Tiles when packet is not addressed to it	2A. Verify that subsequent Expansion Tiles are displaying the expected LED output as listed in 3A.
3. Must be able to process then display packet data address to it onto the led array.	3A. Verify on singular expansion tile that LED matrix is displaying correct result. Compare to expected output on phone.

2.7 LED Array

2.7.1 Description

The LED [6] Array will be composed of an 8x8 array of WS2812B LEDs. These are serially addressable LEDs that are controlled by sending a bitstream indicating what color each LED should be illuminated as. The microcontroller only has to interface with the first LED, with each LED sending the bitstream to its subsequent neighbor.

2.7.2 Requirements and Verification

Requirements	Verification
1. Must be provided with 5V$\pm$0.3 V power	1A. Use multimeter to confirm steady 5V$\pm$0.3 V output.
2. Must display correct image or text	2A. Visually verify that each LED matrix is displaying the correct text/image.

2.8 Expansion Port

2.8.1 Description

The expansion port is essentially a connector that will transfer both voltage, ground, and data signals between each panel. The data will be transferred through the I2C protocol.

2.8.2 Requirements and Verification

Requirements	Verification
1. Must transfer 5$\pm$0.3 V power and clock and data on SDL/SDA lines to next panel.	1A. Use oscilloscope to confirm steady 5$\pm$0.3 V output and correct SCL/SDA values
2. Must physically stay connected without human assistance.	1A. Visually determine that panels are connected, and staying connected.

2.9 Tolerance Analysis

A significant part of our design will be the successful illumination of the LEDs as specified by the user. If the algorithm and control units that describe how the LEDs should be lit up are incorrect, this would pose a significant risk to the successful completion of our project. Furthermore, not delivering adequate power to the LEDs could result in diminished brightness. As more panels are added to the system, we want all LEDs to remain bright without needing an additional power supply.

Individual LED power usage: 3V, 60mA (at full brightness)
Panel size: 8x8; 64 LEDs per panel
$3V * 60mA = 180mW$
$180mW * 64 = 11.52W$

Based on our calculations, each panel will draw 11.52W of power at peak usage. We plan on using a power adapter with a supply of at least 50W. The current draw per panel may seem like it is too high for a basic power supply. However there are a few factors to consider. These calculations are for LEDs that are white and at full brightness. With our use cases, having every LED be white and at full brightness is not very likely. Along with this, the worst case for not providing enough current is dimming/discoloring of some LEDs. We do not think this will be a noticeable issue. Even if it does become a problem, we could use our software to limit brightness when power draw is too high. This would prevent the LEDs from any discoloring.

Furthermore, we conducted tests using 256 WS2812B LEDs powered by a raspberry pi 5V output. When setting LEDs to white at full brightness, we found no change in the first 64 LEDs (which is how many are in 1 panel), but towards the end of the strip they did become slightly more red. When testing with other colors there were no issues.

2.10 Software Flowchart

The following flow chart details the software/firmware that will be implemented in our design. After the user connects the panels in the manner that they desire, the ESP32 will poll each panel using a search algorithm, determining the overall configuration of the panels. Then, the user will connect their smartphone to the ESP32 through Bluetooth and enter in what they want to be displayed on the panels. Next, we will use the I2C protocol to send the specific data packets needed to illuminate the LEDs, with the ESP32 acting as the "master" and the microcontrollers acting as "slaves".

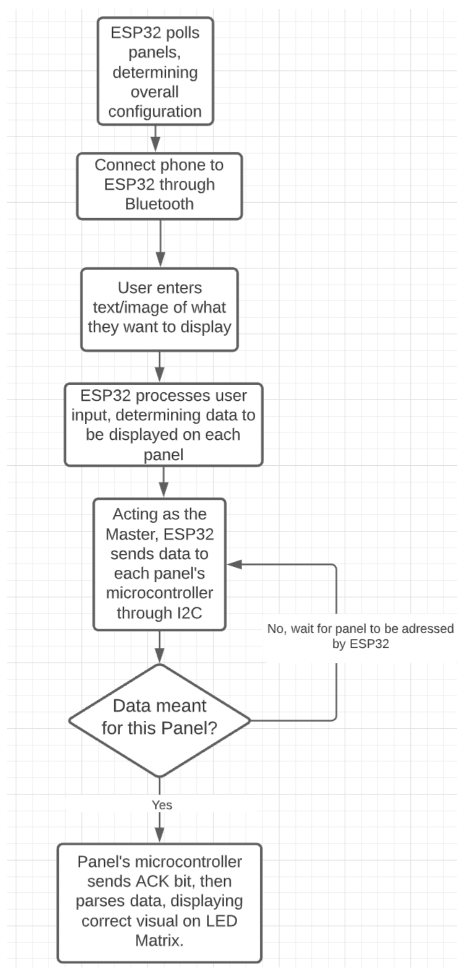

Figure 6: Software Flow Diagram

3 Cost and Schedule

3.1 Cost

3.1.1 Labor

Our hourly cost was calculated using the average hourly cost for an electrical engineer in 2020 in the United States ($49.71/hr) according to the Bureau of Labor Statistics [7].

Name	James Prince	Kyle Salzberg	Adam Chung
Rate	$49.71/hr	$49.71/hr	$49.71/hr
Overhead Cost	2.5x overhead	2.5x overhead	2.5x overhead
Hours Worked	100 Hours	100 Hours	100 Hours
Total Labor Cost	$12,427.5	$12,427.5	$12,427.5

3.1.2 Parts

Part	Part number	Unit price	Quantity	Total
ESP32	ESP32-WROOM-32	$4.20	1	$4.20
ATMEGA328	ATTINY85-20PU-ND	$2.46	4	$9.84
WS2812B LED Strip Individual Addressable Light (300 LEDs @ 60 LED/m)	WS2812B	$24.00	2	$48
Voltage regulator	LM117MPX-3.3	$1.77	1	$1.77
Barrel Jack Connector [8]	PJ-063AH	$1.59	1	$1.59
Capacitors	08053C104M4T4A	$0.01	6	$0.06
Resistors	652-CHP0805FX1002ELF	$0.47	11	$5.17
MOSFETs	BSN20-7	$0.36	2	$0.72
5V Barrel power adapter	ALITOVE 5V 15A AC	$29.99	1	$29.99
Micro-USB breakout	Adafruit-1833	$1.50	5	$7.5
Crystal Oscillator	ECS-160-8-36CKM-TR3	$0.36	1	$0.36
Zener Diode	MMSZ5226B-HE3-18	$0.24	1	$0.24
AVR-ISP-6	See name	$0.95	1	$0.95
Expansion Connectors	0470531000	$0.53	4	$2.12
Buttons [9]	5GSH93542	$5.81	2	$11.62
Box for LEDs and PCB	Custom	$10.00	4	$40.00
Total				$164.13

3.2 Schedule

Week	James Prince	Kyle Salzberg	Adam Chung
2/21	Attend Design Document Check and Finalize Design Document. Work specifically on the cost analysis and tolerance analysis.	Attend Design Document Check and Finalize Design Document. Work specifically on the Schematic and the Ethics and Safety.	Attend Design Document Check and Finalize Design Document. Work specifically on the R&V tables.
2/28	Complete Design Review and finish PCB design. Attend PCB board review. Specifically focus on the Control Panel subsection. Additionally, order parts for the project.	Complete Design Review and finish PCB design.Specifically focus on the Power subsection. Additionally, order parts for the project.	Complete Design Review and finish PCB design. Attend PCB board review. Specifically focus on the expansion panel unit. Additionally, order parts for the project.
3/7	Get approved for first round PCBway orders (by Tuesday). Begin to work on software code. Focus on code for the polling mechanism to figure out configuration of panels.	Get approved for first round PCBway orders (by Tuesday). Begin to work on low level software code (i2C interface).	Get approved for first round PCBway orders (by Tuesday). Begin to work on software code. Focus on code for fitting the letters/images on the panels.
3/14	Spring Break	Spring Break	Spring Break
3/21	Complete Soldering components on PCB. Also, continue working on specific focus of software code (configuration recognition). Get Approved for Second Round PCB Orders if necessary.	Complete Soldering components on PCB. Test PCB and modify PCB with any necessary changes. Get Approved for Second Round PCB Orders if necessary.	Continue working on Software code. Test code on completed PCB to see if lights illuminate as expected.

3/28	Complete Individual Progress Reports. Continue working on software.	Complete Individual Progress Reports. Perform verification tests on PCB	Complete Individual Progress Reports. Continue working on software.
4/4	Finalize software and begin working on demonstration. Make sure the project is completely verified and reliable.	Begin working on demonstration. Start preparing a printed out version of the block diagram, high level requirements, and RV table.	Finalize software and begin working on demonstration. Make sure the project looks polished and is functional.
4/11	Finalize and practice demonstration. Begin working on the final presentation. Specifically work on how to describe the software of the project during the presentation.	Finalize and practice demonstration. Begin working on the final presentation. Specifically work on how to describe the hardware of the project during the presentation.	Finalize and practice demonstration. Begin working on the final presentation. Specifically work on the design format of the presentation.
4/18	Complete Mock Demo.	Complete Mock Demo.	Complete Mock Demo.
4/25	Finalize project and demo completed design. Work on the final paper. Focus on describing the software components of the final paper.	Finalize project and demo completed design. Work on the final paper. Focus on describing the hardware components of the final paper	Finalize project and demo completed design. Work on the final paper. Focus on describing the software components of the final paper.
5/2	Finalize and turn in the final paper.	Finalize and turn in the final paper	Finalize and turn in the final paper.

4 Ethics and Safety

4.1 Ethics

While there is not an excessive amount of ethical issues that arise with our project, we still must take into consideration a few key ethical standards. Code I.1 of the 7.8 IEEE Code of Ethics states that we must "protect the privacy of others" [10]. With users of our product potentially entering in sensitive information to be displayed on the LED panels (e.g., text messages) we must take measures to ensure the privacy of this information. Since we don't plan on storing any user data, we must inform the user of this, but also ensure the user knows of the potential risk of sharing any private information.

Additionally, during the development of our project we must make sure to be honest and trustworthy with our teammates and any others who provide us with assistance. As Section 1.3 of the ACM Code of Ethics states, we must be "honest about [our] qualifications, and about any limitations in [our] competence to complete a task" [11]. This is a crucial standard to follow during development, for designing anything that is outside of our qualifications could lead to potential failure and harm. It is essential for us as teammates to communicate with each other and understand each person's area of expertise.

4.2 Safey

Issues of safety will mostly arise during our time in the lab. We should "never work in the laboratory alone" and will always "clean up after [ourselves]" [12] as stated by the ECE 445 lab rules. Working in the lab alone could lead to potentially dangerous situations if something happens and no TA or professor is present. Also, it is important to clean up after ourselves, for students who use the lab after us may or may not know the potential risks of any equipment we were previously using. Lastly, we will likely be soldering in the lab, so we must always keep to best practices while doing any soldering work.

When it comes to the safety of the user, our device isn't overtly dangerous. However, we must make sure that the device presents no fire hazards, and is safe for overall use. For example, if hanging on the wall, the panels must not be able to fall down or disconnect unintentionally. This could cause potential harm to the user if it were to strike them, so this scenario must be averted.

References

[1] Diodes Incorporated. ”N-CHANNEL ENHANCEMENT MODE FIELD MOSFET”. BSN20. 2019

[2] Espressif Systems. ESP32-WROOM-32. 2022.

[3] Atmel. ”8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash”. 7810D-AVR. 2015.

[4] Phillips Semiconductors Application Note A2907055, Bi-directional level shifter for I2C-bus and other systems. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/an97055.pdf. 1997

[5] Texas Instruments. ”LM1117 800-mA, Low-Dropout Linear Regulator”. SNOS412O. 2020.

[6] Worldsemi. ”WS2812B Intelligent control LED integrated light source”.

[7] Bureau of Labor Statistics, Electrical and Electronics Engineers, [Online]. Available: https://www.bls.gov/ooh/architecture-and-engineering/electrical-and-electronics-engineers.htm: :text. September 2021.

[8] CUI Devices. ”DC Power Jack”. PJ-063AH.

[9] APEM. ”Multimec 5G Switch”. 5GSH93542. 2006.

[10] IEEE Policies, Section 7.8 Code of Ethics, Standard I.1, [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. June 2020.

[11] ACM Code of Ethics and Professional Conduct, Section 1.3. [Online]. Available: https://www.acm.org/code-of-ethics. June 2018.

[12] ECE 445 Lab Rules. [Online]. Available: https://courses.engr.illinois.edu/ece445/lab/