

MODULAR DISPLAY

By

Dale Morrison (dalejm2)

Sean Halperin (seanmh3)

Yuzhe He (yuzhehe2)

Final Report for ECE 445, Senior Design, Spring 2026

TA: Wesley Pang

[May] 2026

Project No. 28

Abstract

The Modular Screen System is a reconfigurable display platform comprising 4 LCD tiles that together display a single composite image across multiple screens. Each tile connects to its neighbors through magnetic pogo pins that carry power to each tile. A central ESP32 microcontroller hosts a server that receives user-selected images wirelessly over Wi-Fi from a mobile or desktop device, and node tiles join the server to display a section of the image from their current position. During development, the team successfully demonstrated image transfer to the ESP32, tile-to-tile physical connectivity via pogo pin connectors, and layout detection, enabling the control tile to identify neighboring tile positions. The LCD screens were successfully driven. But image rendering was not successful. Future work will focus on resolving image segmentation and synchronization across the tile array.

Contents

1. Introduction.....	3
1.1 Purpose.....	4
1.2 Functionality.....	4
1.3 Subsystem Overview.....	4
1.1 High level requirement.....	6
2 Design.....	6
2.1 Equations & Simulations.....	6
2.2 Design Alternatives.....	6
2.3 Design Description & Justification.....	6
2.4 Subsystem Diagrams & Schematics.....	6
3. Costs.....	7
3.1 Cost.....	7
3.2 Schedule.....	7
4. Design Verification.....	7
4.1 Completeness of Requirements.....	8
4.2 Appropriate Verification Procedures.....	8
4.3 Quantitative Results.....	8
5. Conclusion.....	8
5.1 Accomplishments.....	9
5.2 Uncertainties.....	9
5.3 Ethical considerations.....	9
5.4 Future work.....	9
6. References.....	9

1. Introduction

This report addresses the lack of a flexible and reconfigurable way to display media across a flat surface. The Modular Screen System was developed as a solution, allowing users to freely configure multiple LCD display tiles into different arrangements to present a single composite image. The system focuses on four key areas: wireless communication, image processing, embedded systems, and server and client design.

1.1 Purpose

Large-on-the-table displays used in conference rooms, interactive activities, and tabletop role-playing games are typically expensive, difficult to use, and cannot be reconfigured once installed. Commercial displays require complex mounting systems and centralized control hardware. There is a need for a low-cost, spatially flexible alternative that non-specialists can assemble and rearrange without the need for specific tools.

This project fulfills that need by designing a modular screen built from display tiles. Each tile forms a self-contained unit with a 7-inch LCD panel driven by an ESP32-S3 microcontroller. Users can

physically place tiles next to each other and align them so their connection edges touch. Magnetic pogo pins embedded at the tile edges snap tiles together and enable them to communicate as a single display. A single master tile broadcasts a Wi-Fi access point, allowing any device to connect and send an image that propagates across all tiles. In the end, the system delivers a scalable, wireless, reconfigurable display using only low-cost components.

1.2 Functionality

The system provides the following high-level functionalities:

- ❖ **Tile Display Output:** Each tile independently drives a 1024-by-600 LCD over RGB. The ESP32-S3 and Arduino_GFX library manage the framebuffer and generate the pixel clock.
- ❖ **Backlight Power Conversion:** Each tile uses a voltage booster that increases the 5 V input to approximately 9.1 V for the LCD backlight, eliminating the need for a separate high-voltage supply.
- ❖ **3.3 V Logic Regulation:** A 3.3 V regulator converts the 5 V system rail to 3.3 V for the ESP32-S3 and digital peripherals.
- ❖ **Wi-Fi Access Point and Web Server:** The master tile hosts a Wi-Fi access point and deploys an HTTP web server, providing a color picker interface.
- ❖ **Tile Communication:** Tiles are mechanically and electrically connected through magnetic pogo pins that provide 5 V power, ground, and a shared communication bus between adjacent tiles.

1.3 Subsystem Overview

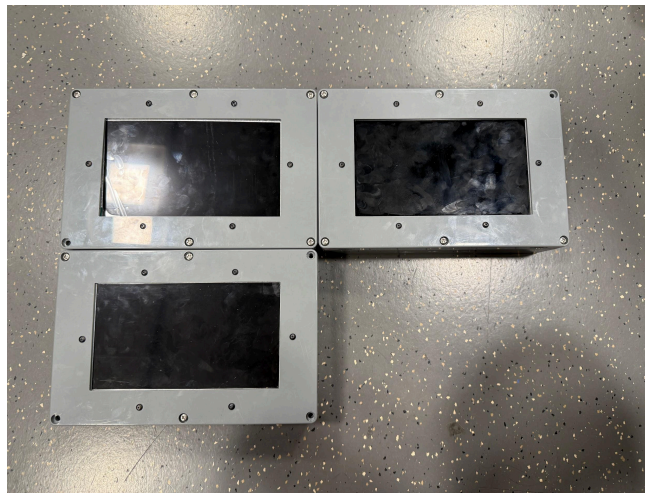


Figure 0: Final Physical Design

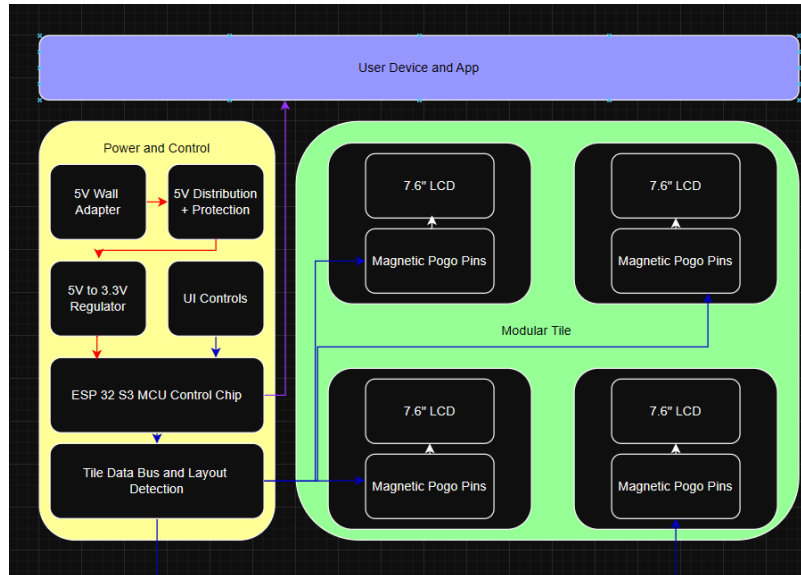


Figure 1: High Level Block Diagram

There are 3 main subsystems for the project: power and control, modular tile, and user device. The power part of the power and control subsystem uses a 5-volt connector to input the necessary voltage to the rest of the system. It also converts the 5 volts to the other two necessary voltages: 3.3 volts for the microcontroller and 9 volts for the screen. The control part of this subsystem is the MCU or ESP32-S3, which handles all display control and information collection. Therefore, the power and control subsystem controls all the power and communication throughout a tile. The modular tile subsystem has two main parts: connectivity and image display. The tiles must be connected using pogo pins and must be able to send power and ground across each tile. Each tile must also be able to display images on a screen. The last subsystem is the user device, which is the server created by the MCU, allowing the user to upload an image and send it to the screen. Each subsystem is connected using a PCB designed in KiCad that integrates all subsystems into a compact “motherboard,” which is placed inside a tile.

2 Design

2.1 Equations & Simulations

Boost Converter Output Voltage

The TPS61165DBV is a current-mode boost converter with a feedback pin whose reference voltage is 200 mV. The output voltage is set by a resistor divider from VOUT to FB. Using the feedback divider equation:

$$V_{out} = V_{ref} \times (1 + R1/R2)$$

where VREF = 0.2 V. With R1 = 100 kΩ and R2 = 4.7 kΩ, the nominal output is:

$$V_{out} = 0.2 \times (1 + 100 \text{ k}\Omega / 4.7 \text{ k}\Omega) = 0.2 \times 22.26 \approx 4.45 \text{ V}$$

The feedback resistor values were selected to achieve a target output of 9 V to drive the LCD screen's backlight.

Power Distribution

The AMS1117-3.3 must handle the dropout between a 5 V input and a 3.3 V output. The ESP32-S3 draws up to 240 mA during active Wi-Fi transmissions. The worst-case power dissipation in the LDO (low dropout regulator) is:

$$PLDO = (V_{in} - V_{out}) \times I_{load} = (5.0\text{V} - 3.3\text{V}) \times 0.240\text{A} = 0.408\text{W}$$

The AMS1117-3.3 is rated for 1 W. With a 0.408 W dissipation, the temperature rise is approximately $0.408 \times 165 \text{ }^\circ\text{C/W} = 67 \text{ }^\circ\text{C}$ above ambient, reaching approximately $92 \text{ }^\circ\text{C}$ at a $25 \text{ }^\circ\text{C}$ ambient, which is within the $125 \text{ }^\circ\text{C}$ maximum junction temperature. Copper pours on the PCB were used to improve heat spreading.

RGB Interface

The minimum pixel clock frequency to drive our LCD display at a refresh rate of 24 frames per second is calculated as:

$$fPCLK = H_{total} \times V_{total} \times FPS$$

where:

$$\begin{aligned} H_{total} &= H_{active} + H_{frontPorch} + H_{sync} + H_{backPorch} = 1079 \text{ Pixels} \\ V_{total} &= V_{active} + V_{frontPorch} + V_{sync} + V_{backPorch} = 624 \text{ Lines} \end{aligned}$$

At 24 fps: $fPCLK \geq 1079 \times 624 \times 24 \approx 16.2 \text{ MHz}$. A 9 MHz pixel clock was used during bring-up to reduce signal integrity risk.

2.2 Design Alternatives

The design went through several significant changes from the original proposal to the final implementation. These changes are documented below, along with their rationale.

Data Transfer: Serial/SPI to Wi-Fi

The original design envisioned a wired UART link between tiles, using the pogo pins for image data transfer, with a master-to-node design. This approach was abandoned for three reasons. First, routing high-speed SPI traces across the pogo pins introduced signal integrity concerns at the PCB edge. Second, the Wi-Fi peripheral of the ESP32-S3 offers a far more user-friendly interface and eliminates the need for any wiring. The third and final reason was speed: using Wi-Fi greatly increased the speed at which image data was transferred. Each tile now independently receives its image over Wi-Fi, simplifying the design.

User Interface: Dedicated App to Web Hosted Page

An early concept involved a custom mobile application to control the display. This was replaced with an HTML page served directly from the ESP32-S3 itself. The web server approach requires no app installation on the user's device, works on any operating system with a browser, and can be updated without recompiling a separate application. The web page approach provides the most accessibility to users and simplifies device programming, which is the main reason this change was made.

Image Configuration in the Web App

Rather than preloading images onto the MCU's flash memory, the final design allows the user to dynamically set the image displayed on the tile through the web interface. The web app converts the user's uploaded image to an RGB value, which the MCU uses to update the displayed image on the LCD screen. This approach eliminates the need for an on-board filesystem, simplifies the firmware, and reduces flash wear from repeated image writes.

MCU Selection: ESP32 C6 to ESP32-S3

The original bill of materials specified an ESP32 C6 MCU. After evaluating the display requirements, the design was upgraded to the ESP32-S3, which includes a dedicated hardware RGB LCD controller peripheral. This peripheral handles the pixel clock, HSYNC, VSYNC, and DE signals in hardware, enabling smooth rendering without specialized programming. The ESP32-S3 also supports OPI PSRAM, which provides the required frame buffer memory. Lastly, the ESP32-S3 had 50 GPIO pins, which connected to the LCD display to control the image, while the C6 had 23 pins [1], which were not enough to fully control the high-definition LCD.

Display Interface: FPC Connector Added

The original schematic did not include an on-board FPC connector; the intent was to use a separate breakout board. After PCB fabrication, it became clear that an integrated FPC connector was necessary for a clean, reliable connection to the 50-pin display cable, and with the change to the ESP32-S3, it was now possible [3]. A FPC connector was added to the KiCad PCB layout. This connector routes all 50 display pins directly to the ESP32-S3 GPIO pins and power rails on the PCB, eliminating the fragile wire-by-wire connections used during breadboard testing.

Pogo Pins: Data and Power to Power Only

The original design called for pogo pins to carry both power and data signals. After switching to Wi-Fi for all data transfer, the pogo pins were simplified to carry only the 5 V of power and GND. This reduced the number of required pogo pin contacts and eliminated the risk of digital noise from data signals being injected into the power rail. Each tile edge exposes two power pogo pins (5 V and GND), allowing tiles to be chained together in any physical arrangement while sharing a single power supply.

Removal of the TPS61165DBV Boost Converter

The original design included a TPS61165DBV boost converter to generate the 9 V required by the LCD backlight LED from the 5 V input supply. After testing, it was determined that a 9 V wall adapter could be provided directly to the backlight pins of each tile without exceeding current limits. The backlight draws approximately 60 mA at 9 V. Removing the boost converter reduced PCB complexity and

removed a potential source of electromagnetic interference on the board. It also made the full design cheaper. The backlight supply is now a separate dedicated input on each tile.

2.3 Design Description & Justification

Microcontroller

For the MCU, we chose the ESP32-S3. We chose this MCU for four main reasons. First, it includes a hardware-accelerated LCD RGB interface peripheral that manages DMA transfers to the panel without a CPU. Second, it supports 2.4 GHz Wi-Fi, eliminating the need for an external wireless module. Third, it offers 8 MB of PSRAM, enabling the framebuffer for our display, which approximately requires 1.2 MB of PSRAM rather than the limited 512 KB of internal SRAM. Fourth, the S3 has enough pins to drive the high-definition screen data. LCD display. The ESP32-S3 operates on a 3.3 V supply and is programmed using the onboard USB serial bridge.

Power Subsystem

The 5 V input is provided by a barrel connector from a 5 V, 2 A wall adapter. A P-channel MOSFET provides protection. The AMS1117-3.3 LDO generates the 3.3 V rail for the ESP32-S3 and all digital signals. 10 μ F capacitors are placed at the input and output of the LDO to suppress low-frequency noise. The TPS61165DBV boost converter, along with a 10 μ H inductor, resistors, and a Schottky diode, generates approximately 9 V for the LCD backlight LEDs. Lastly, the TVS diode protects the boost converter output against overvoltages.

Display and Interface

The LCD display panel is a 7-inch screen with a 1024 \times 600 resolution, driven via a 50-pin FPC connector [2]. The panel uses a 24-bit RGB parallel interface with separate HSYNC, VSYNC, DCLK, and DE signals. The MODE pin is set to always be high to select DE mode rather than SYNC mode. L/R and U/D mirror pins are both driven high to set the default scan direction. RESET is driven from GPIO46 and low for at least 1 ms during initialization. The full 24-bit data bus uses the ESP32-S3's GPIO pins 1–3, 7–12, 15–21, and 47–48 for the B, G, and R channels, respectively. Therefore, the MCU should be able to control the screen's images. VCOM is connected to GND through a 100-nF capacitor as recommended.

Wi-Fi and Web Server

On boot, the master tile initializes the display and then starts a Wi-Fi access point. The WebServer library serves an HTML page at the root endpoint that presents a color picker. A separate endpoint accepts a 24-bit hexadecimal value, converts it to the 16-bit RGB encoding used by the Arduino_GFX library, and then updates the entire display. The Arduino loop calls the server.handleClient() continuously to process incoming HTTP requests. This decision was made to ensure image data is sent quickly. Before switching to Wi-Fi, transferring an image took more than a minute. After the change, it took less than 5 seconds to send image data over to the screen.

2.4 Subsystem Diagrams & Schematics

Below is a summary of the key component selections for each subsystem, along with the full KiCad schematic files and PCB. If any additional information is needed regarding our components, we have included the data sheets for all components used within our references.

Label In KiCad	Part Number	Value / Description
U1	TPS61165DBV [4]	Boost Conv., 5V→9V
U2	AMS1117-3.3 [5]	LDO Reg., 3.3V/800mA
Q1	DMP2045U-7 [6]	P-ch MOSFET, 20V/4.3A
D1	82400102	TVS Diode
D2	SK54	Schottky, 5A/40V
L1	CD54 Series	10 μ H Inductor
FPC1	FH12A-50S-0.5SH [7]	50-pin FPC, Horizontal
R1	100 k Ω	FB top resistor
R2,R4,R5	10 k Ω	Pull-up / pull-down
R3	330 Ω	LED current limit
R6	0.9 Ω	Current sense shunt
C1	1 μ F	Input bypass
C2,C3,C5,C7,C9	10 μ F	Bulk bypass
C4	100 nF	VCOM filter
C6	4.7 μ F	Boost output
C8	220 nF	Soft-start / comp.

Table 1: Key Components per Tile

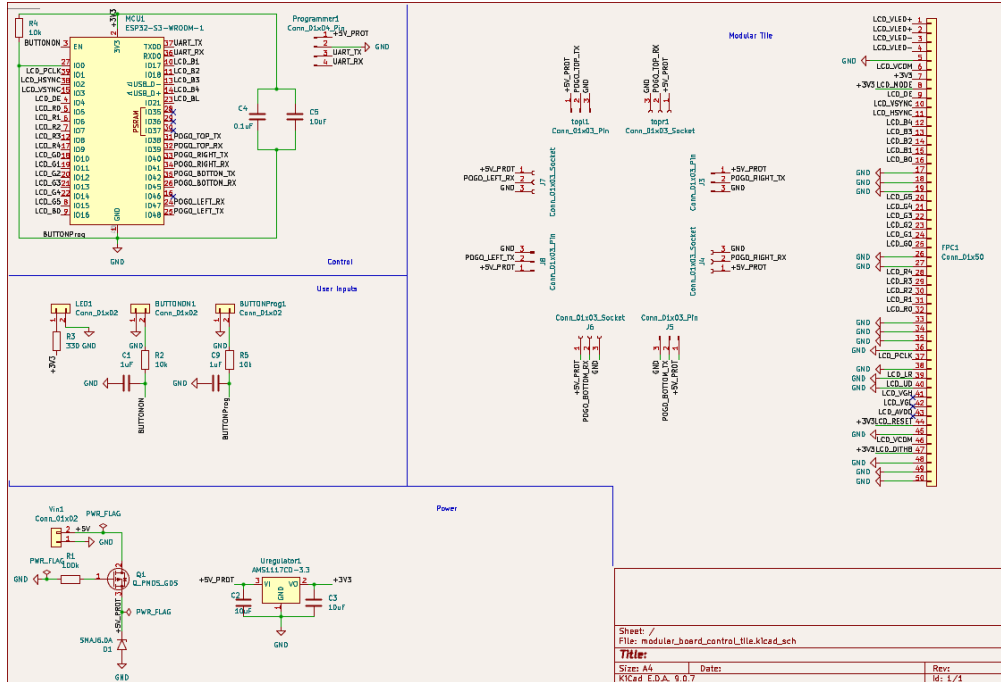


Figure 2: Ki-Cad Schematic

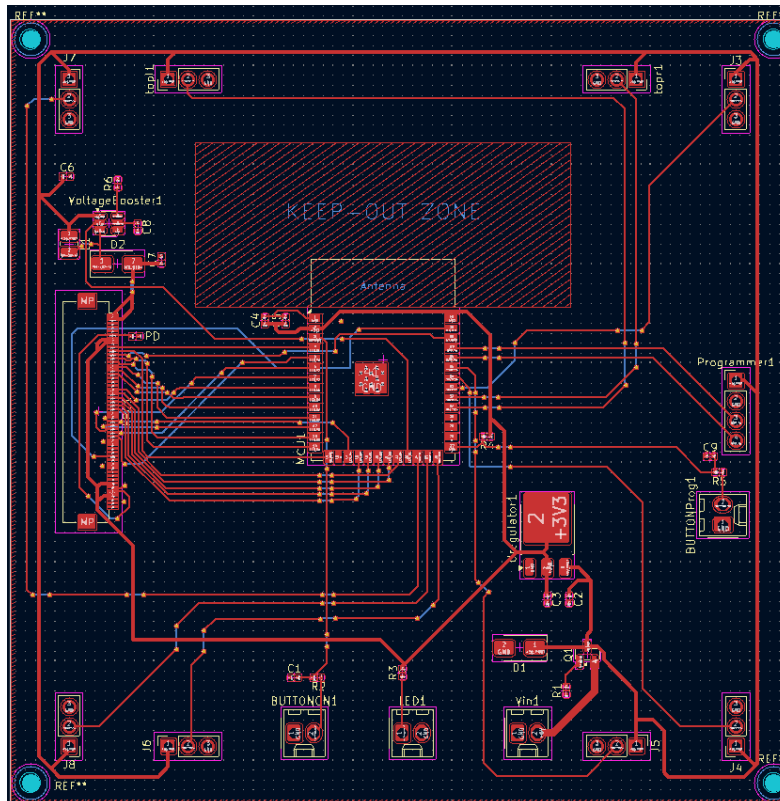


Figure 3: Ki-Cad PCB Design

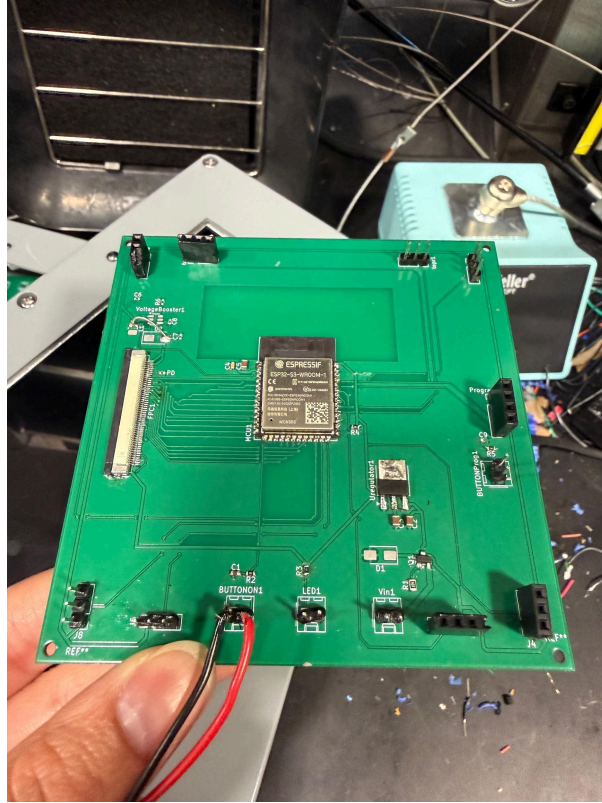


Figure 4: Physical PCB

3. Costs

3.1 Cost

Component	Qty	Unit Cost	Total Cost	Manufacture
ESP32-S3-WROOM-1 (16 MB, OPI PSRAM)	4	\$4.20	\$16.80	Mouser
ER-TFT070A2-6 7-inch IPS LCD	4	\$32.00	\$128.00	Eastrise
AMS1117-3.3 SOT-223 LDO	5	\$0.28	\$1.4	DigiKey
Hirose FH12A-50S-0.5SH FPC Connector	5	\$1.02	\$8.10	DigiKey
DMP2045U-7 P-ch MOSFET SOT-23	4	\$0.35	\$1.40	DigiKey
82400102 TVS Diode SOT23-6L	4	\$0.62	\$2.48	DigiKey
0603 SMD Resistor/Capacitor Kit	1 kit	\$12.00	\$12.00	Amazon

Component	Qty	Unit Cost	Total Cost	Manufacture
Pogo Pin Connectors (10-pack)	6	\$8.19	\$49.14	Amazon
CP2102 USB-UART Programmer	1	\$7.99	\$7.99	Amazon
PCB Fabrication (2-layer, 4 boards)	1 order	\$5.00	\$5.00	PCB Way
5 V 2 A Barrel Adapter	1	\$6.99	\$6.99	Amazon
9 V Wall Adapter (for backlight)	4	\$7.50	\$30.00	Amazon
Male/Female Pin Header Kit	1	\$8.99	\$8.99	Amazon
Total:			\$278.29	

Table 2: Part cost

Parts cost			\$ 278.29
Labor cost	Estimated salary: \$29 per hour	Estimated time: 70 hour each	\$ 15225
Machine shop cost	Estimated salary: \$30 per hour	Estimated time: 12 hours	\$ 360
Total			\$15863.29

Table 3: Total cost with part, labor, and machine shop

3.2 Schedule

This is the work each team member completed week by week.

Week number	Sean	Yuzhe	Dale
2/9	Create design section of project proposal		
2/16	finished the project design, start working on first PCB design	Making team contract	Worked Introduction and Conclusion and formatting of project proposal
2/23	Working on first round of PCB	Finding available parts	Part Procurement
3/2	Finish first round of PCB, attended design review	Brainstorming hardware design	Researching how to implement a system for sending images from iPhone to ESP
3/9	Change MCU to S3 redesigning PCB	Initialize client side on iphone	Beginning testing code for Bluetooth
3/23	Changing PCB design for find round	Finishing and testing code for bluetooth	Finishing and testing code for bluetooth
3/30	Ordering additional parts for the project	Pivoting from bluetooth to wifi data transfer	Researching libraries that would support image transfer over Wifi
4/6	Start soldering on PCB	Researching the use ability of softAP	Researching the use ability of softAP
4/13	Continue soldering PCB	Working on PCB testing	Implementing SoftAP and LittleFS

			code into our data pipeline
4/20	Testing board and pcb connection	Working on code testing and debugging	Working on Code and debugging
4/27	Trying to make screen work		

Table 4: Weekly Schedule

4. Design Verification

4.1 Completeness of Requirements

Main Power Requirements

- Must supply 5V to each tile continuously
- Must be within 100mV of desired voltage
- Overcurrent and short circuit protection

For the power system portion of the Screen System, our goal was to distribute power to each tile via a set of connections that would allow for the screens to be added and removed with ease. With this subsection of our project, we had three main requirements (listed above)

To fulfill our first requirement, we chose to assign one tile to be the “hub” of power in our system. This Main Tile was equipped with a 5 Volt Barrel connector, allowing it to be plugged into the wall. In order to distribute that power (and thus complete the requirement), we chose to use magnetic 3-pinned pogo connectors to securely connect the tiles to one another and also allow them to be rearranged easily. Each tile was equipped with a total of 8 connectors, all of which were connected to our PCB. 2 were mounted on each of the 4 faces of each tile, facing outward. Each face had a male and female connector to allow for the connection of any tile to any other tile. These connectors supported 5 Volts of power and created a network of power starting at the Main Tile and branching out to and through all peripheral tiles connected to the system.

To fulfill our second requirement, we purchased parts that supported the correct voltages for the main components of the PCB. After each round of PCB orders, we spend time measuring the voltages to ensure that our predictions of the assumed voltages were in alignment with the parts that were delivered to us. In order to make sure that there was consistency, we measured our voltages multiple times over a consistent instance of the system being plugged in, as well as looking at different tiles in different configurations. In our measurements, we saw an overall consistency that aligned with our tolerance of 100mV.

To fulfill our third requirement of preventing overcurrent and implementing general circuit protection, we used a MOSFET gate that shuts down when presented with too much power. This protected the rest of the system from being damaged if plugged in incorrectly and from any mistakes that we might have made in designing how power would flow throughout our system.

Main Control Requirements

- Detect tile layout within 5 seconds.
- Assign each tile its correct image section.

Our goal was to make sure that the Modular Screen System could send and receive image data quickly and that the information was reliable.

To fulfill our first requirement, we had to cycle through a few methods of data transmission before achieving a fast system. First, we wanted to send data through our pogo pins; we realized that the transmission would not work alongside power. Then we decided to make a shift towards Low Energy Bluetooth connection as the ESP32 we were using had Bluetooth capabilities, and we also wanted to conserve power. This method was not reliable, however, due to the fact that the transmission times were extremely slow and did not meet our requirements. We finally switched to transmitting our image data using the WiFi Capabilities of the ESP32, hosting a small server that would use HTTP requests to send our image files to the tiles within the desired time frame.

To fulfill our second requirement, we decided to send a copy of the full image along with a position over to each tile and task each of the ESP32s with deciphering which portion of the image to show on its screen.

Physical Tile Requirements

- 4 Tiles with 7.6" Displays
- Easy connection
- Brightness variation between tiles is at most 10%

We had a few main high-level requirements for our tiles to function fully to achieve the goal of a truly "Modular Screen System". These goals encompass the larger overarching goal of creating a screen with accessibility and visibility.

Our first and second requirements were fulfilled simply during the procurement process. We found the correct parts (the screens and the pogo pins) and coordinated with the machine shop to create a valid housing for the PCB and screen components of the project. Each housing was given enough space for a PCB and a screen and was mounted with pogo pins on each side. We also made sure that the Main Tile's housing had an opening for a barrel connector and an LED to fulfill its specific requirements.

We were not able to complete our third requirement of having a brightness variation of less than 10%. Our Screens did not fully function by the end of our time working, as we were only able to get the backlight turned on.

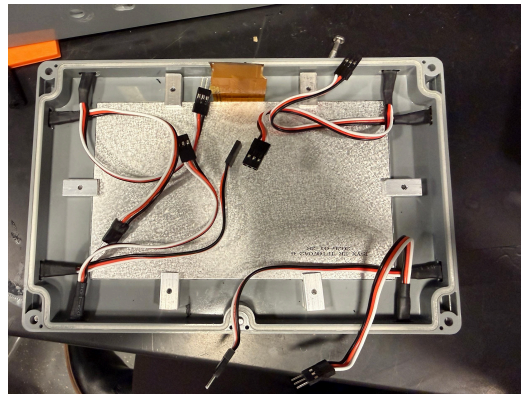


Figure 5: Physical Tile Layout

Application Requirements

- Upload images and deploy to the tile layout
- Confirm successful deployment to user
- System must continue if the app is closed

Our application requirements cover the goals that we had for the software portion of the project. Since we wanted the Screen system to be controlled via an application on a user's phone, ensuring clear communication between the 2 devices was necessary.

To fulfill our first requirement, we created a server that could be hosted on the ESP32 that served as an interface for the Modular Screen System. This interface took the form of a webpage that the user could go to on their browser when connected to the ESP32's network. On this webpage, the user would be allowed to upload an image from their camera roll or file system on their phone and then upload it to each of the ESP32s connected to the system. Our second requirement was fulfilled in the form of there being a confirmation requirement from all powered-on ESP32s before the phone could even send the image. Our page would wait for this confirmation from all boards and would then send out the image to be displayed, serving as a reliable way to ensure no tile was left out from receiving the user image.

Our third requirement was fulfilled through our use of a server being hosted on the ESP32. This made it so that as long as each tile was receiving power, the image would remain on the ESP32 regardless of whether the phone was on the webpage or not.

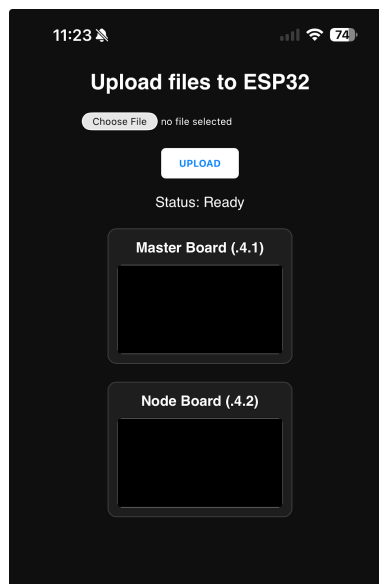


Figure 5: Application

4.2 Appropriate Verification Procedures

To verify our voltage is within limits, we measure the voltage delivered to each major part of the board and measure the time it takes for data transfer from user devices to make sure our device can handle image transfer in a short time.

For hardware we measure three major sections, first is the voltage to the chip, this is to make sure we deliver a consistent voltage to the chip, the desired voltage is 3.3V which we manage to deliver, second is the voltage to the fpc connector that power the screens, for this we need a consistent 5V to

the FPC connector which we also manage to perform, third is the voltage to pogo pins this part is to make sure there are voltage transferred between tile, need to be 5V.

For the software section, we need to make sure it takes less than five seconds.

4.3 Quantitative Results

Desired Voltage	Measured Voltage	Average Voltage
5 V	5.04 V	
5 V	5.02 V	
5 V	5.08 V	
5 V	5.0 V	5.035 V

Table 5: Voltage connected FPC connector

Desired Voltage	Measured Voltage	Average Voltage
3.3 V	3.68 V	
3.3 V	3.02 V	
3.3 V	3.12 V	
3.3 V	3.36 V	3.295 V

Table 6: voltage to Esp32

Desired Voltage	Measured Voltage	Average Voltage
5 V	5.048 V	
5 V	5.048 V	
5 V	5.048 V	
5 V	5.048 V	5.048 V

Table 7: Voltage transferred between tiles

Table 8: Time it take to transfer image from user device to esp32		
Desired Time	Measured Time	Average Time
5 sec	4 sec	
5 sec	2.1 sec	
5 sec	3.8 sec	
5 sec	4.5 sec	3.6 sec

Table 8: Time it take to transfer image from user device to esp32

5. Conclusion

5.1 Accomplishments

The team successfully designed and implemented a system where all tiles physically connect correctly through magnetic pogo-pin connectors, carrying power between tiles. Automatic layout detection was achieved, allowing the user to map the tile position on their phone. We use a SoftAP system, which uses a local server housed on the main tile to handle image transmission to the ESP32. Additionally, rather than 3D printing custom housings, commercial enclosures were purchased to house the displays, providing adequate protection and consistent spacing between tiles. However, image display on the tiles was not achieved, as the ESP32 was unable to successfully render and distribute image segments to the LCD screens.

5.2 Uncertainties

Currently, there are many things we still want to test. The main one is our displays, since wiring an ESP32 to the display directly is very complex, so we want to test if using a display with a built-in driver chip would be better for the project. We want to test if our current power system can transfer enough energy to more displays with more tiles.

5.3 Ethical considerations

One potential concern is user safety. If someone touches the power connectors while the system is connected, it could create a safety hazard. To reduce this risk, we plan to use low-voltage power and incorporate protective shielding around the connectors so they cannot be accessed while connected. Another concern is unauthorized access to the system through wireless communication. To prevent this, we will implement password protection or secure authentication methods to ensure that only authorized users can control the display panels.

5.4 Future work

For future development, we plan to improve synchronization between display modules and explore more cost-effective data transfer solutions to reduce the approximately \$80 cost of connectors. We plan to add eventual touchscreen functionality and support animation rather than still images. To complement that, we plan to have audio integration to have full video capabilities.

6. References

[1]

“ESP32-C6-MINI-1-N4 | DigiKey Electronics,” *DigiKey Electronics*, 2026.

<https://www.digikey.com/en/products/detail/espressif-systems/ESP32-C6-MINI-1-N4/17885796> (accessed Mar. 31, 2026).

[2]

“TFT LCD Display Datasheet,” *BuyDisplay*, Jul. 07, 2023.

https://www.buydisplay.com/download/manual/ER-TFT070A2-6_Datasheet.pdf (accessed Mar. 30, 2026).

[3]

“ESP32-S3-WROOM-1-N8R8 | DigiKey Electronics,” *DigiKey Electronics*, 2026.

<https://www.digikey.com/en/products/detail/espressif-systems/ESP32-S3-WROOM-1-N8R8/15295891> (accessed Mar. 31, 2026).

[4]

“TPS61165 High-Brightness, White LED Driver in WSON and SOT-23 Packages,” Texas Instruments Incorporated, Nov. 2007. Accessed: May 06, 2026. [Online]. Available:

https://www.ti.com/lit/ds/symlink/tps61165.pdf?ts=1778052236890&ref_url=https%253A%252F%252Fwww.google.com%252F

[5]

“AMS1117-3.3 | DigiKey Electronics,” *DigiKey Electronics*.

<https://www.digikey.com/en/products/detail/evvo/AMS1117-3-3/22482148> (accessed May 06, 2026).

[6]

“DMP2045U-7 | DigiKey Electronics,” *DigiKey Electronics*, Jul. 2021.

<https://www.digikey.com/en/products/detail/diodes-incorporated/DMP2045U-7/8275356?s=N4IgtCBcDaICIFkAKYAMAWArAVQLQHYQBdAXyA> (accessed May 06, 2026).

[7]

“FH12A-50S-0.5SH | DigiKey Electronics,” *DigiKey Electronics*.

<https://www.digikey.com/en/products/detail/hirose-electric-co-ltd/FH12A-50S-0-5SH/273548> (accessed May 06, 2026).