WordPal

Group 36

Nala Rivera
Hilda Deveaux
Daniel Ortiz-Gratacos

Daniel Ortiz-Gratacos
Computer Engineering

Hilda Deveaux
Computer Engineering

Nala Rivera
Computer Engineering

Motivation and Background

- Wordle is a popular daily word game that uses color based feedback to guide player guesses
- The game is normally played on phones or computers
- Our project brings Wordle into a standalone physical device with its own keyboard, display, and lighting feedback
- Our goal is to design a portable handheld device that recreates Wordle using real buttons and visual feedback

Goals

Basic

- Generation/selection of 5-letter words on device
- LCD display to show game progress and interface with user
- User input via on-device physical QWERTY keyboard
- Feedback via RGB LEDs diffused behind keyboard
- Wi-Fi capability via ESP32 (built in), for obtaining daily word
- Battery powered

Advanced

- Add a buzzer speaker for auditory feedback (and maybe personality)
- Allow exporting game results as serial text data via USB-A

Stretch

- Create wired local multiplayer through USB-A
- Touch screen
- Additional word games
- Multiple word lengths

Objectives

Basic

- Create a Network API to obtain the daily word if necessary
- Create an algorithm that pseudo randomly generates words (at least 100 possible words)
- Create an algorithm that keeps track of the rounds (default would be 6 rounds)
- Create code that allows the player to guess and for the guess to be checked for correctness
- Display colors that give hints to the player on the LCD
- Display the game playing interface on the LCD
- Light up relevant LEDs with the relevant colors that also give hints to the player (implement an LED multiplexer)
- Implement a keyboard of 26 letters in the QWERTY format, a backspace, and an enter key
- Create code that allows a button to interrupt and do a new game
- Implement a 5V regulator circuit
- Implement a power switch
- Create Wi-Fi configuration code
- Power from rechargeable batteries

Advanced

- Create the relevant code which allows exporting (implement serial functionality which could include the CP2102 chip and UART)
- Create the relevant code that utilizes the buzzer speaker (implement buzzer speaker)

Stretch

- Implement LEDs that light up relative to who wins or loses, and this be accomplished through the USB-A cable (and implement the relevant things for wired local multiplayer through USB-A)
- Create code that can have smaller or bigger word sizes for the word that needs to be guessed, and rounds adjusted accordingly
- Create code for other word games
- Implement Touch Screen

Table of Engineering Specifications

General	
Dimensions	< 8 in x 5 in x 2 in
Weight	< 200 g
Battery Capacity	> 2000 mAh
Battery Life	> 1 hour
Cost of Materials	< $200
Keyboard	
Response Time	Less than 100 ms
RGB LEDs	Addressable, <= 5V, >2 bpc
Display	
Response Time	Less than 100 ms
Brightness	> 250 nits

House of Quality

		Design Requirements	Response Time	Display Brightness	Battery Capacity	Weight/Dimensions	Cost
			-	+	+	-	-
Customer Requirements	Ease of Use	+	↑	↑	↓	↓	↑
	Portability	+			↓	↑	
	Battery Life	+	↓	↓↓	↑↑	↑	↓↓
	Charge Time	-			↓		↓↓
	Cost	-	↓	↓	↓		↑↑
Targets			<100 ms	> 250 nits	>1000 mAh	<= 8x5x2in, <=200g	<$200 excluding prototyping

Processor Technology Comparison

- Microcontroller far more cost-effective
- No need for the level of performance of the other two options
- Familiarity with developmental processes
- Microcontroller allows for simpler and more compact PCB
- Easy decision

Characteristic	MCU	SoC	FPGA
Cost	Affordable	Expensive	Very Expensive
Power Efficiency	Excellent	Fair	Poor
Performance	Fair	Good	Excellent
Flexibility	Fair	Fair	Excellent
Software Development	Moderate	Moderate	Difficult
PCB Development	Simple	Moderate	Difficult
Physical Size	Small	Medium	Large

Microcontroller Comparison

- Shared prior experience with MSP430 family
- Need memory for storing graphics and user data
- Need at least PRNG for random word selection
- ESP32 has Wi-Fi capabilities, allowing us to fetch words over online APIs
- MSP430 is ancient, RP235x brand-new, ESP32 is in the middle
- ESP32 popular with hobbyists
    - Greater wealth of information available

Characteristic	TI MSP430	RP2350/2354	ESP32
Chip Cost	≤$5	$1-2	$2-4
Dev Board Cost	$20-30	$7-8	$10-20
Speed	Up to 25MHz	Up to 150MHz	Up to 240MHz
Volatile Memory	Up to 66KB	520KB	520KB
Flash Memory	Up to 512KB	RP2350: none RP2354: 2MB	448KB ROM Up to 16MB
Serial Connectivity	UART, SPI, I2C, USB	USB, UART, SPI, I2C, HSTX	USB, UART, SPI, I2C
Other Peripheral Interfaces	Touch, temp sensor, ADC, LCD, PWM	Touch, temp sensor, ADC,	Touch, Ethernet, IR, PWM, I2S, ADC, DAC
Other Features	AES, RTC	TRNG, crypto	Wi-Fi, Bluetooth, TRNG, crypto, RTC, USB-OTG
Physical Size (mm)	Varies, commonly from 3x3-16x16	7x7 QFN, 10x10 SMD	Publicly available as ~20x20 WROOM module

Display Technology

- E-ink far more efficient, but far too slow and limited for our needs, on top of cost
- OLED more efficient than LCD, but much more expensive than an LCD of the same size or resolution

Characteristic	LCD	OLED	E-ink
Cost	$20-50	$50+	$50+
Power Efficiency	Decent	Great	Excellent
Brightness	Backlit, decent	Good	None
Contrast	Good	Excellent	Good
Colors	8bpc	8bpc	2-4 colors total

Battery Technology

Property	Lithium-ion	NiMH (AA)
Nominal voltage	3.7V	1.2V
Capacity (mAh)	2000-3000	2200-2800
Discharge rate	Up to 10C (risky)	Up to 3C
Weight	Moderate	Moderate
Shape / form	Cylindrical cell	Cylindrical cell
Safety Risk	Moderate	Relatively low

Power Table

Component	Voltage	Current	Power Consumption
ESP32-S3 MCU	3.3V	~ 350mA	1.16 W
TFT Display	3.3V	~ 100mA	0.33 W
RGB Key LEDs	5.0V	15mA * 26 = 400 mA	1.95 W
QWERTY Keyboard	3.3V	10mA	0.033 W
NiMH Batteries (2)	2.4V	2A maximum output	2500 mAh capacity

Programming Environment Comparison

Arduino Core for ESP32's downsides in mixed quality of libraries and second highest performance typically are not too problematic for this project, and the upsides may make it a suitable choice.

Criteria	Arduino Core for the ESP32	ESP-IDF	MicroPython
Ease of Use	Beginner Friendly	Intermediate/Advanced	Beginner Friendly
Programming Language Support	C, C++, Arduino Programming Language	C, C++	Python
Library Support	Many Libraries with Mixed Quality	Many Libraries with Better Average Quality relative to Arduino	Many Libraries with Better Average Quality relative to Arduino
Performance	Second Highest Performance Typically	Highest Performance Typically	Third Highest Performance Typically
Suitability for Prototype Development	Suitabe for Rapid Prototyping	More Suitable for Professional Grade Solutions	Suitable for Rapid Prototyping

Display Library Comparison

The choice might be more between TFT_eSPI and LovyanGFX. TFT_eSPI's performance is not that much of a downside, and it might be overall easier compared to LovyanGFX. There was the potential for TFT_eSPI not working out, but so far it has.

Criteria	TFT_eSPI	Adafruit GFX	LovyanGFX
Ease of Use	Beginner to Intermediate in terms of Setup Beginner in terms of most functions Beginner to Intermediate in terms of Sprites	Beginner to maybe Intermediate in terms of Setup Beginner in terms of most functions Beginner to Intermediate in terms of Canvas	Beginner to Advanced in terms of Setup Beginner in terms of most functions Beginner to Advanced in terms of Sprites
Feature Set / Capability	Graphics Primitives, Text, Fonts, Sprites, Some Anti-Aliasing, Partial Screen Updates, etc.	Graphics Primitives, Text, Fonts, Canvas, etc.	Graphics Primitives, Text, Fonts, Sprites, Color Formats, etc.
Performance	Second Highest Performance Typically	Third Highest Performance Typically	First Highest Performance Typically

LED Control Comparison

FastLED's bigger LED Compatibility does edge it out over the others. The ease of use is maybe not too bad for WordPal. The performance seems to be indefinite, so that may affect future usage of it.

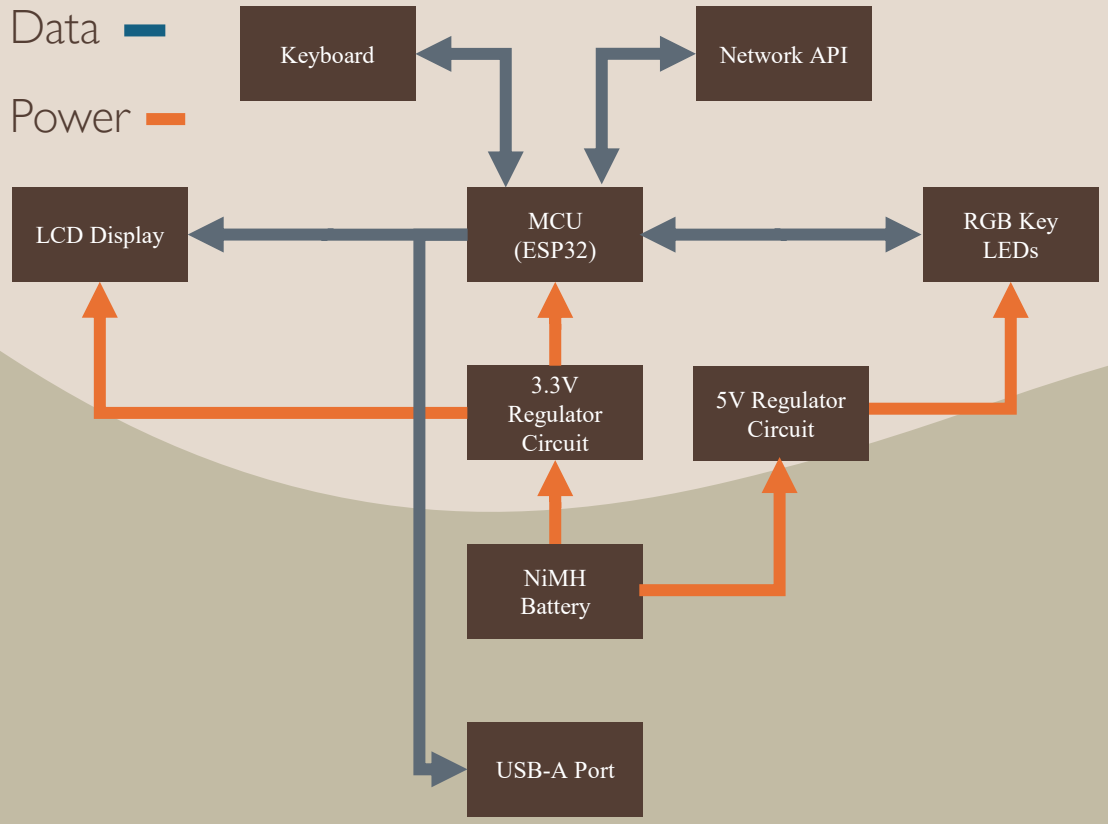
Criteria	FastLED	Adafruit NeoPixel	pololu-led-strip-arduino
LED Compatibility	Supported LED Chipsets include: WS2812B, APA102, SK6812, WS2813, and more Multiple LED Protocols Supported	Supported Chipset Families Include: WS2812, WS2811, SK6812, and more	Supported Strips: low-speed TM1804 (requires a lower version like 1.2.0), high-speed TM1804, SK6812, and WS2812B WS2811 LEDs Also Supported
Ease of Use	Beginner to Intermediate in terms of setup and code	Beginner to Maybe Lower Intermediate in terms of setup Beginner in terms of code	Beginner to Maybe Lower Intermediate in terms of setup Maybe Intermediate in terms of code
Performance	Theoretically Faster than Adafruit NeoPixel Experimentally Inconclusive	Theoretically Slowest Experimentally Inconclusive	Theoretically Fastest

Wi-Fi Software Stack and Optional Layers Comparison

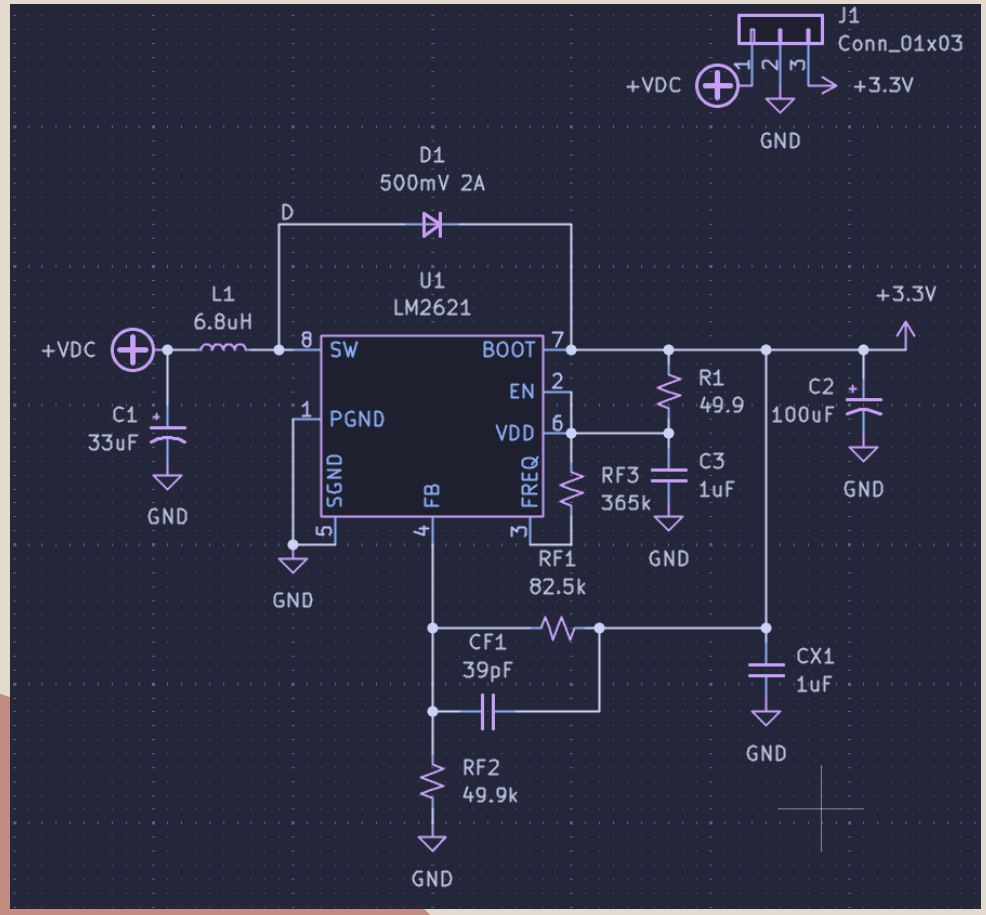
Wifi.h is a required choice, but using it by itself might make the most sense. AsyncTCP add unnecessary complexity for no practical gain. WiFiManager integration could be an additional feature added on.

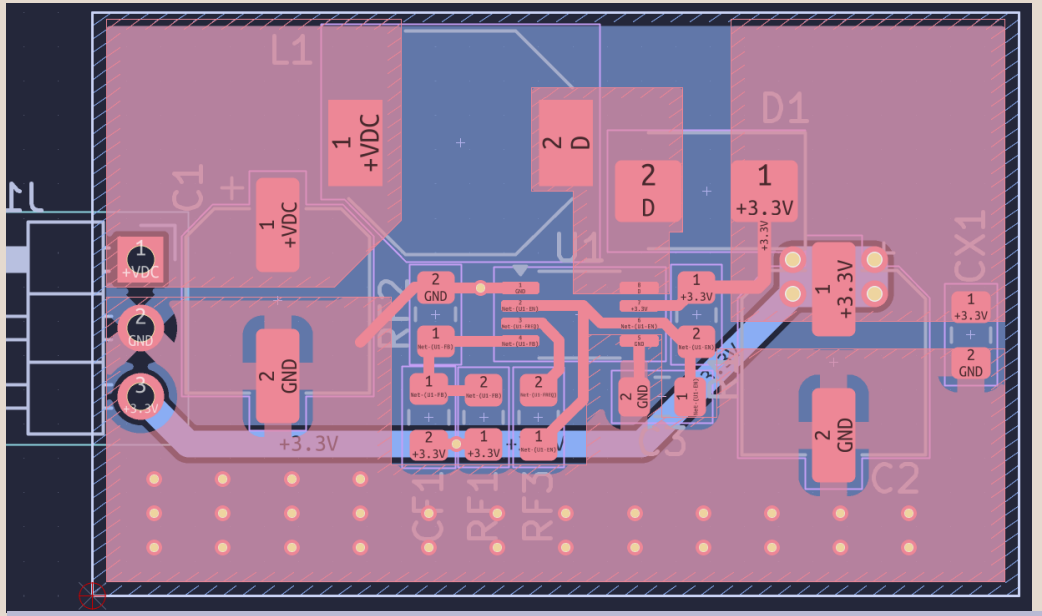
Criteria	Arduino-esp32 Wifi.h	WiFiManager	AsyncTCP
Ease of Use	Beginner in terms of setup and code	Beginner to Maybe Intermediate in terms of setup Beginner in terms of code	Beginner to Maybe Intermediate in terms of setup Maybe Intermediate to Advanced in terms of code
Feature Set	ESP32 WiFi can run in different WiFi modes: WiFi Station, Access Point, Access Point + Station Mode, and Promiscuous mode. It can scan WiFi networks, connect to a WiFi network, check ESP32 WiFi connection strength, and more.	It allows one to provide Wi-Fi credentials from the outside, do on-demand configuration, and add custom parameters.	AsyncTCP allows for multiple connections and makes up the base for ESPAsyncWebServer.
Dependencies	What's necessary in the Arduino Core	DNSServer-ESP32 library, WebServer-ESP32 library, Wifi.h, and What's necessary in the Arduino Core	Wifi.h, and What's necessary in the Arduino Core

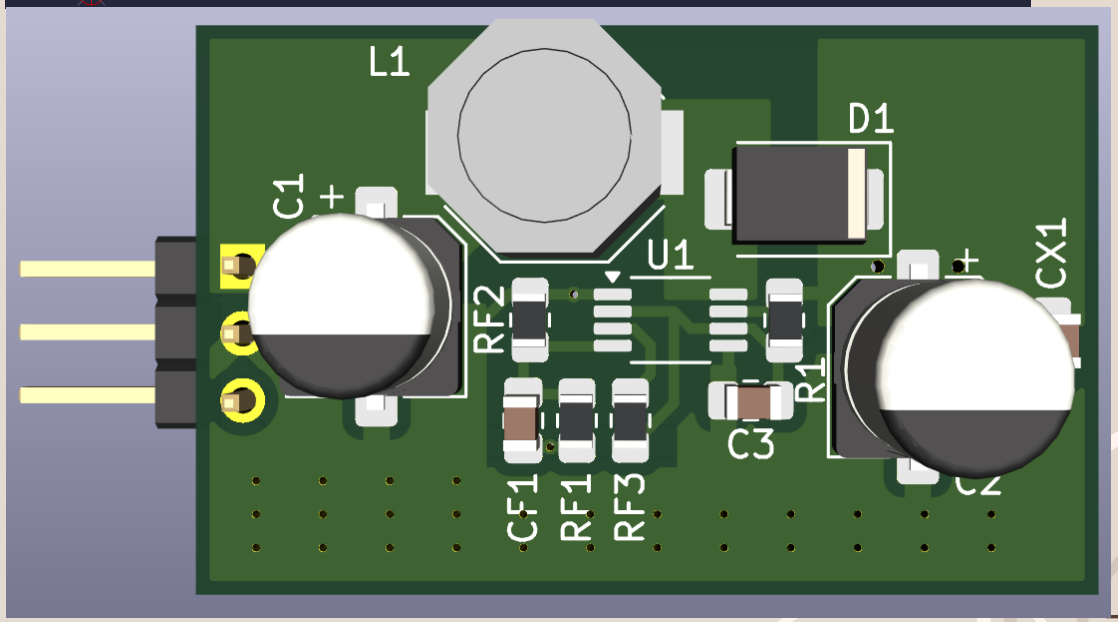
Hardware Block Diagram



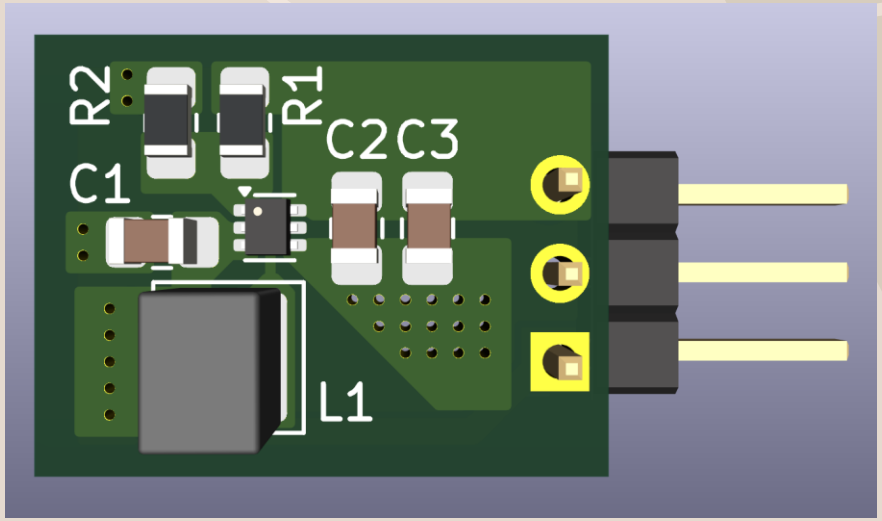
3.3V Regulator

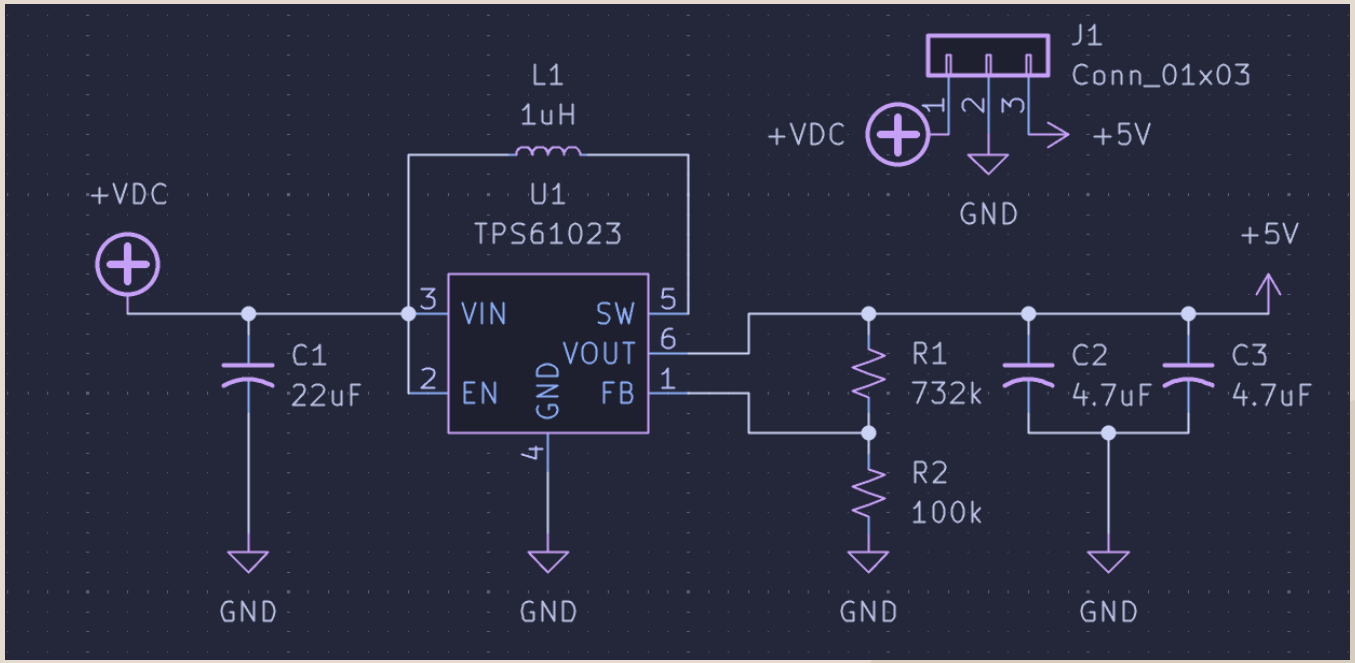


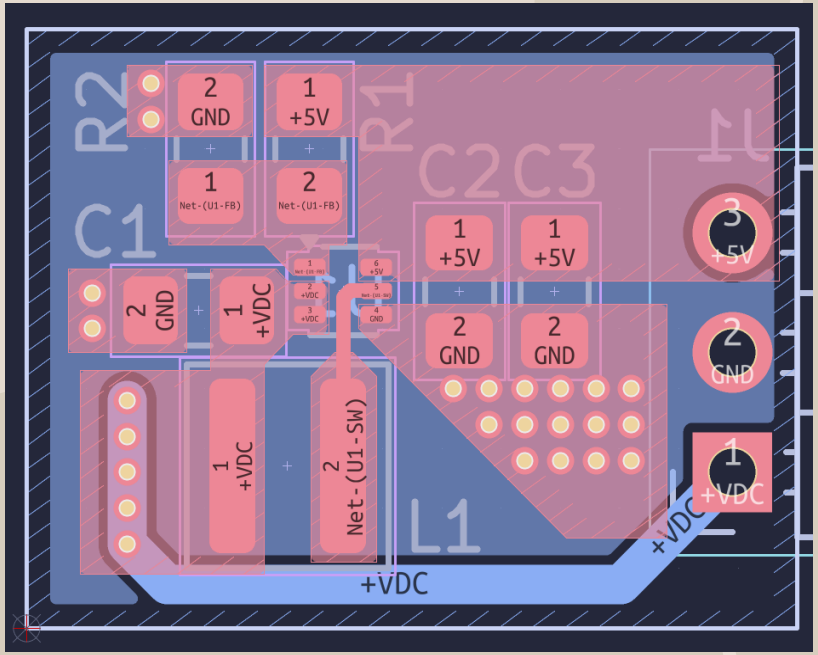




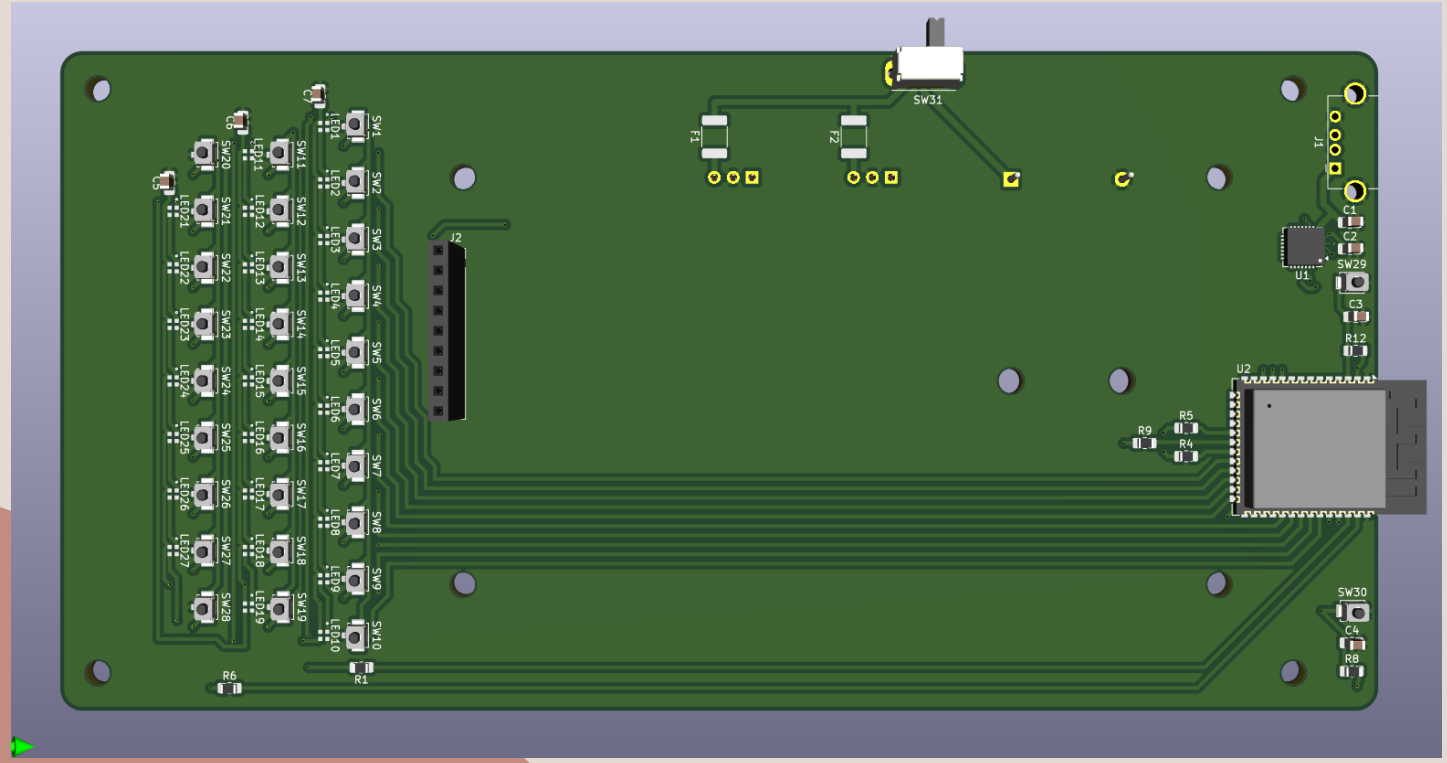
5V Regulator

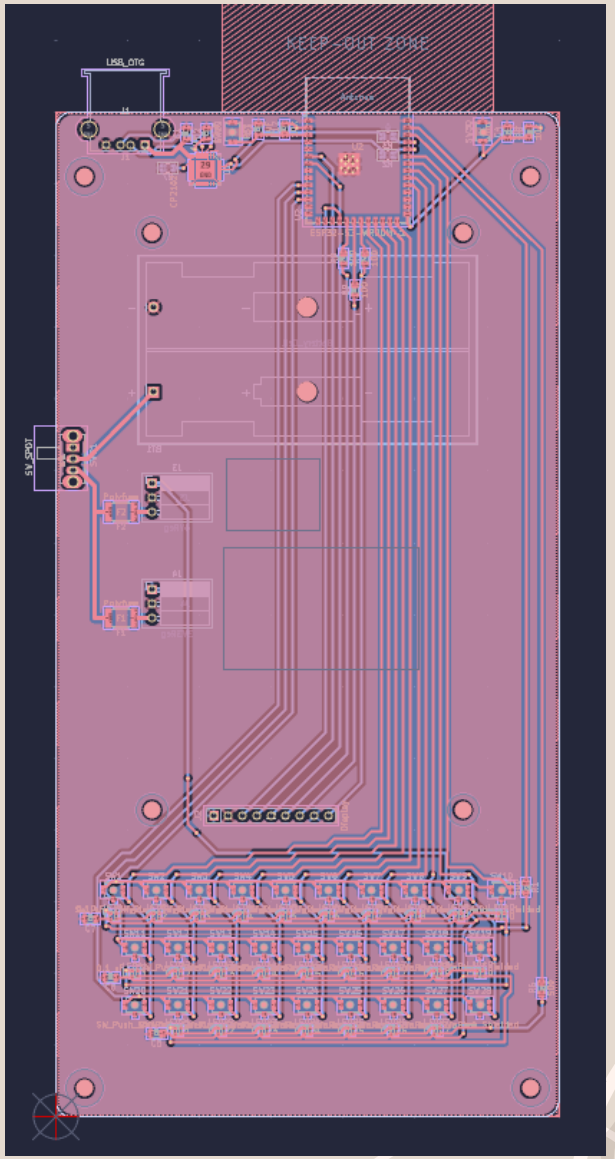




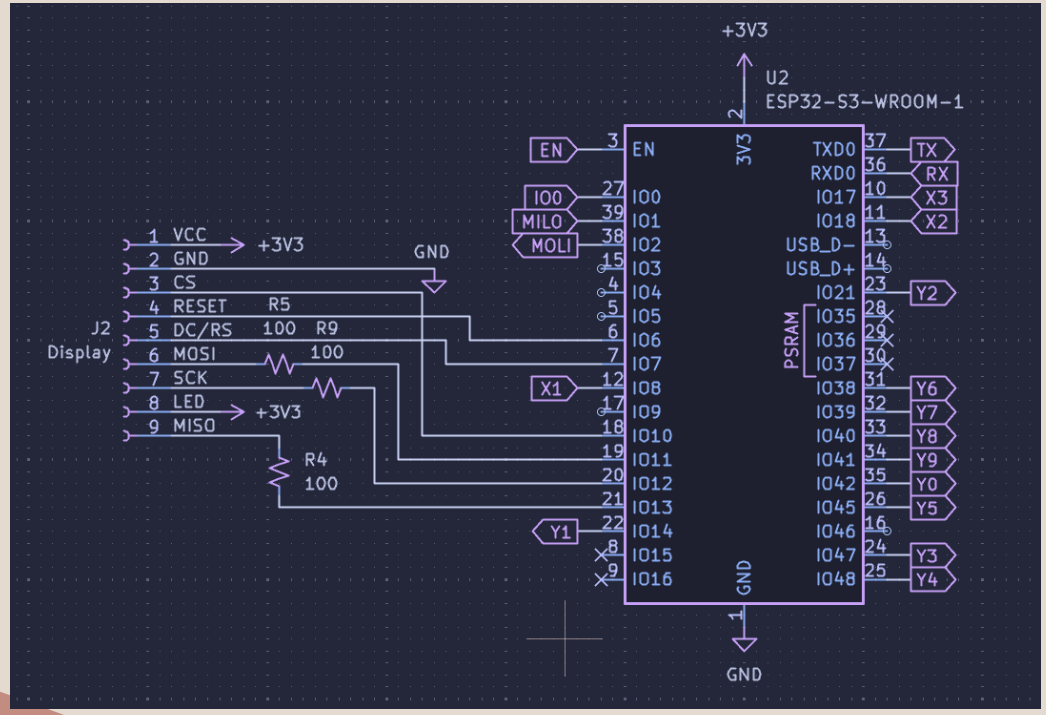


Main Board

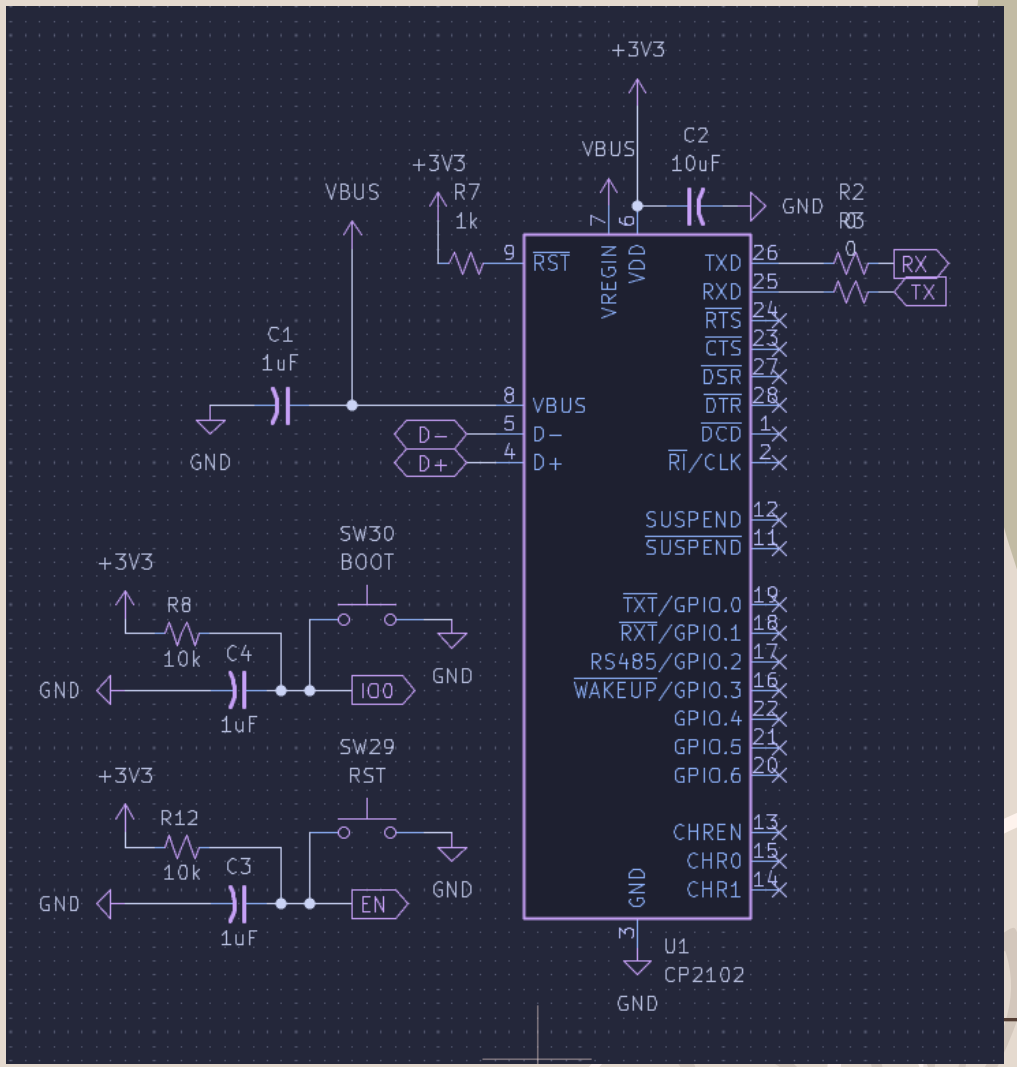

SW31
F1
F2
J1
J2
U1
U2
C1
C2
C3
SW29
R12
R5
R9
R4
SW30
C4
R8
R6
R1


KEEP-OUT ZONE
USB_OTG

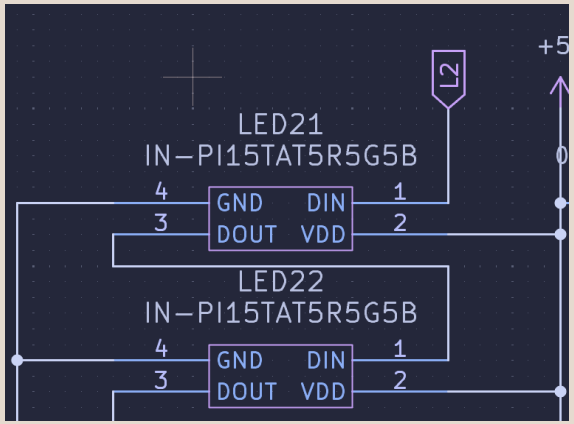
ESP32

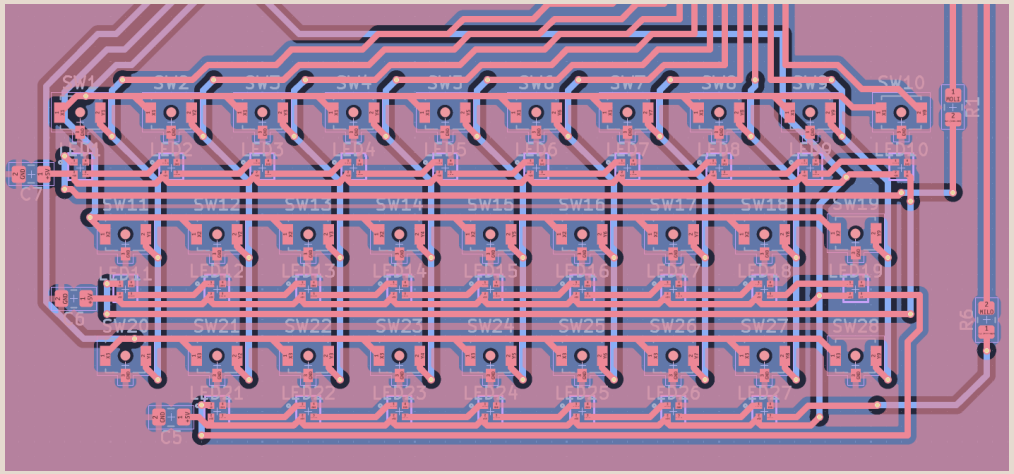


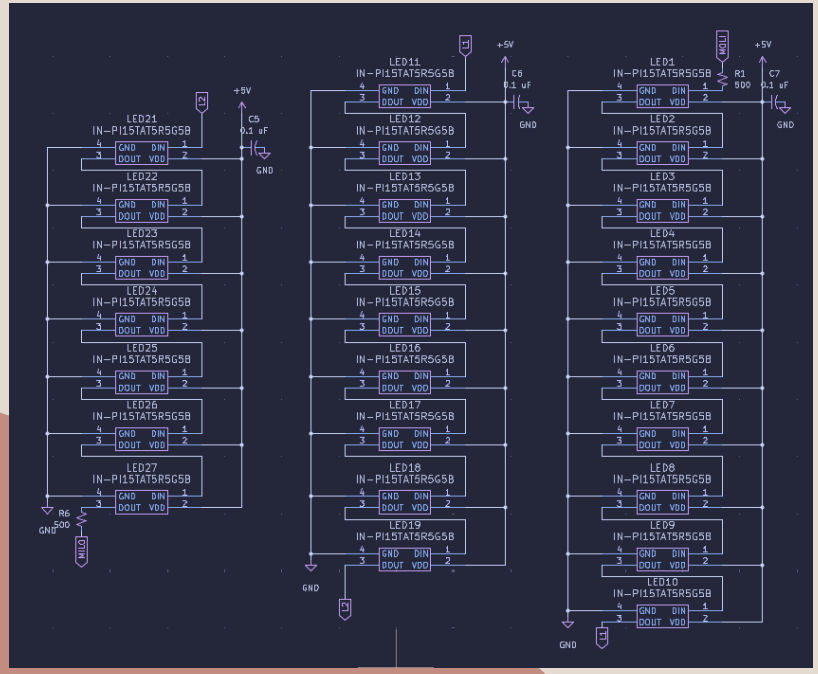
CP2102

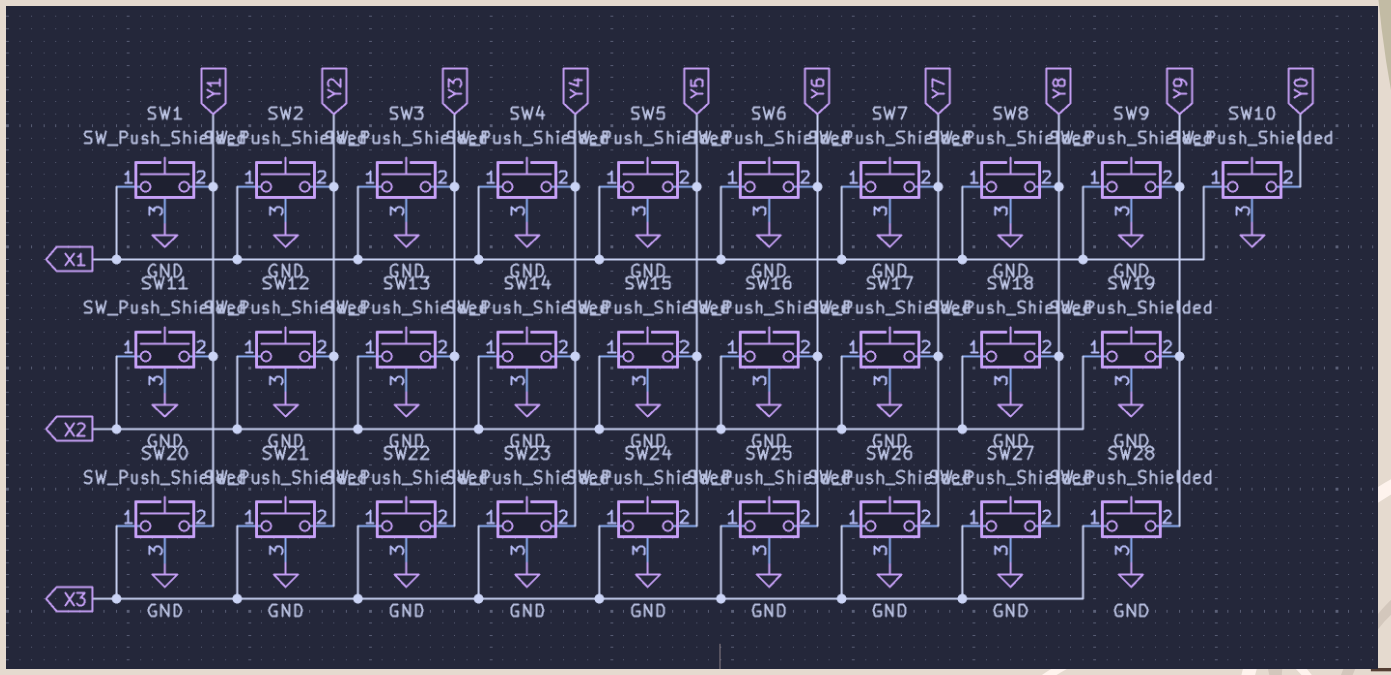


Keyboard and Lighting

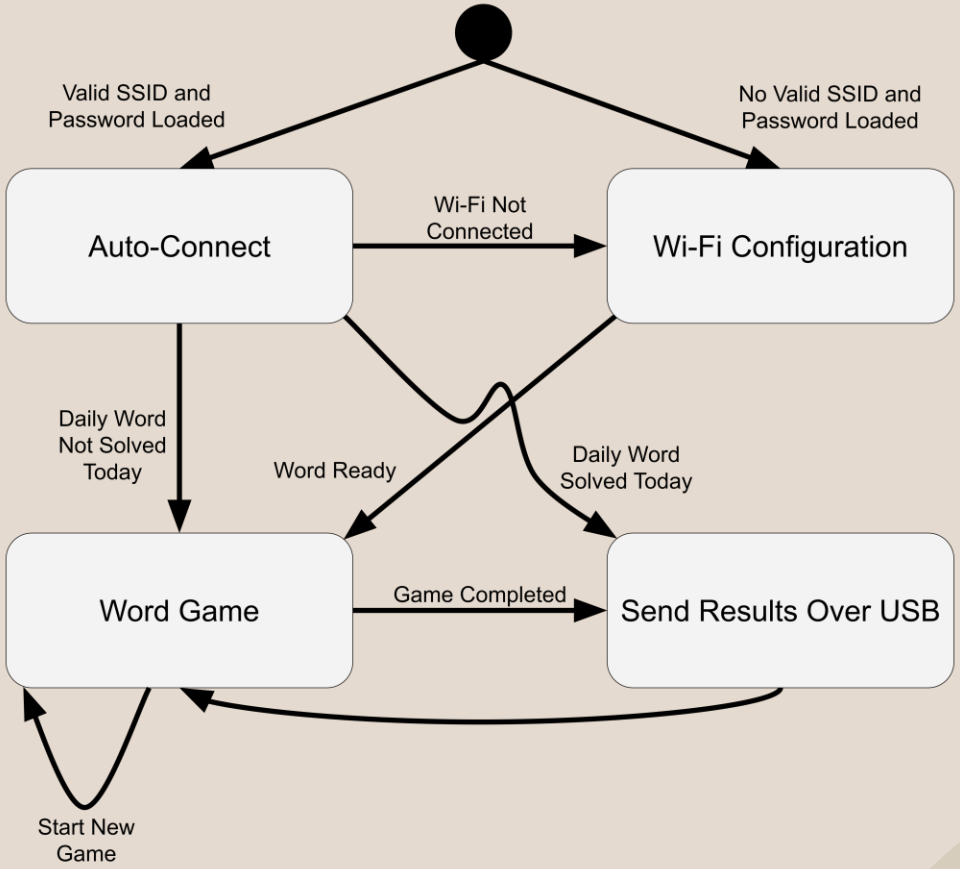


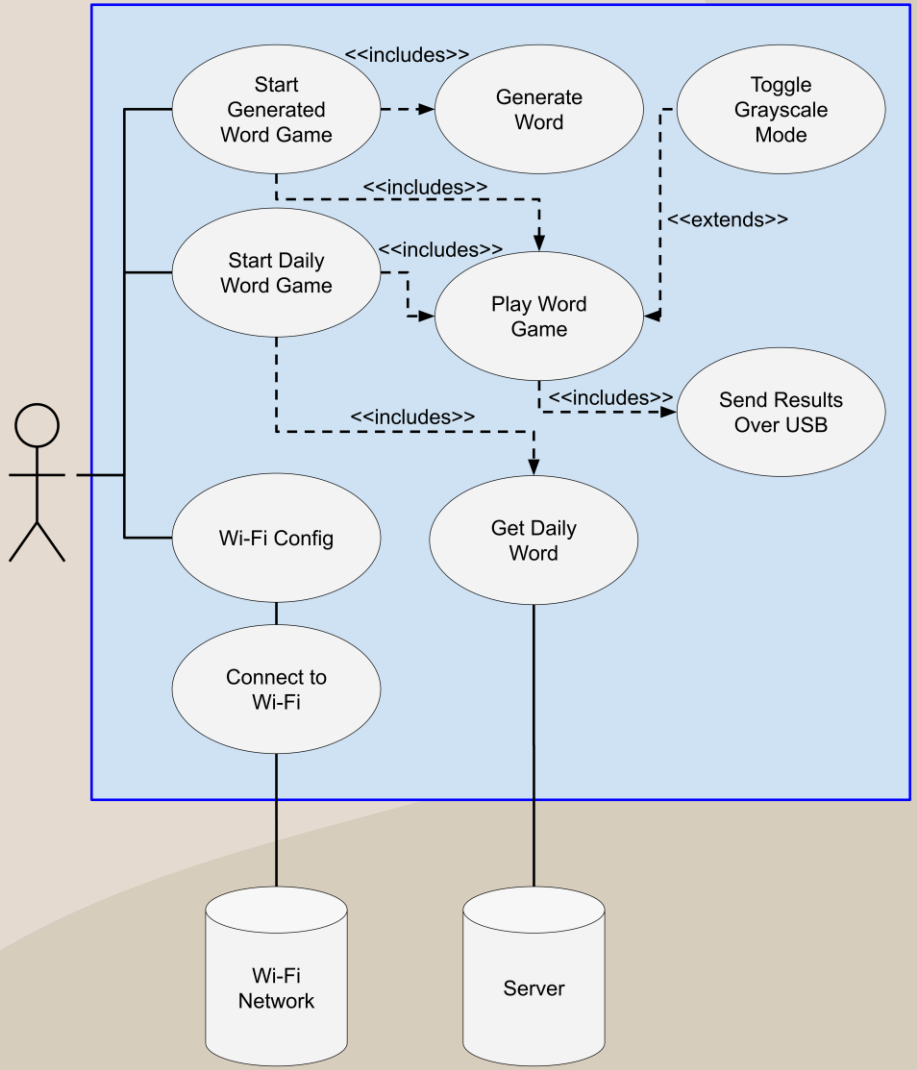




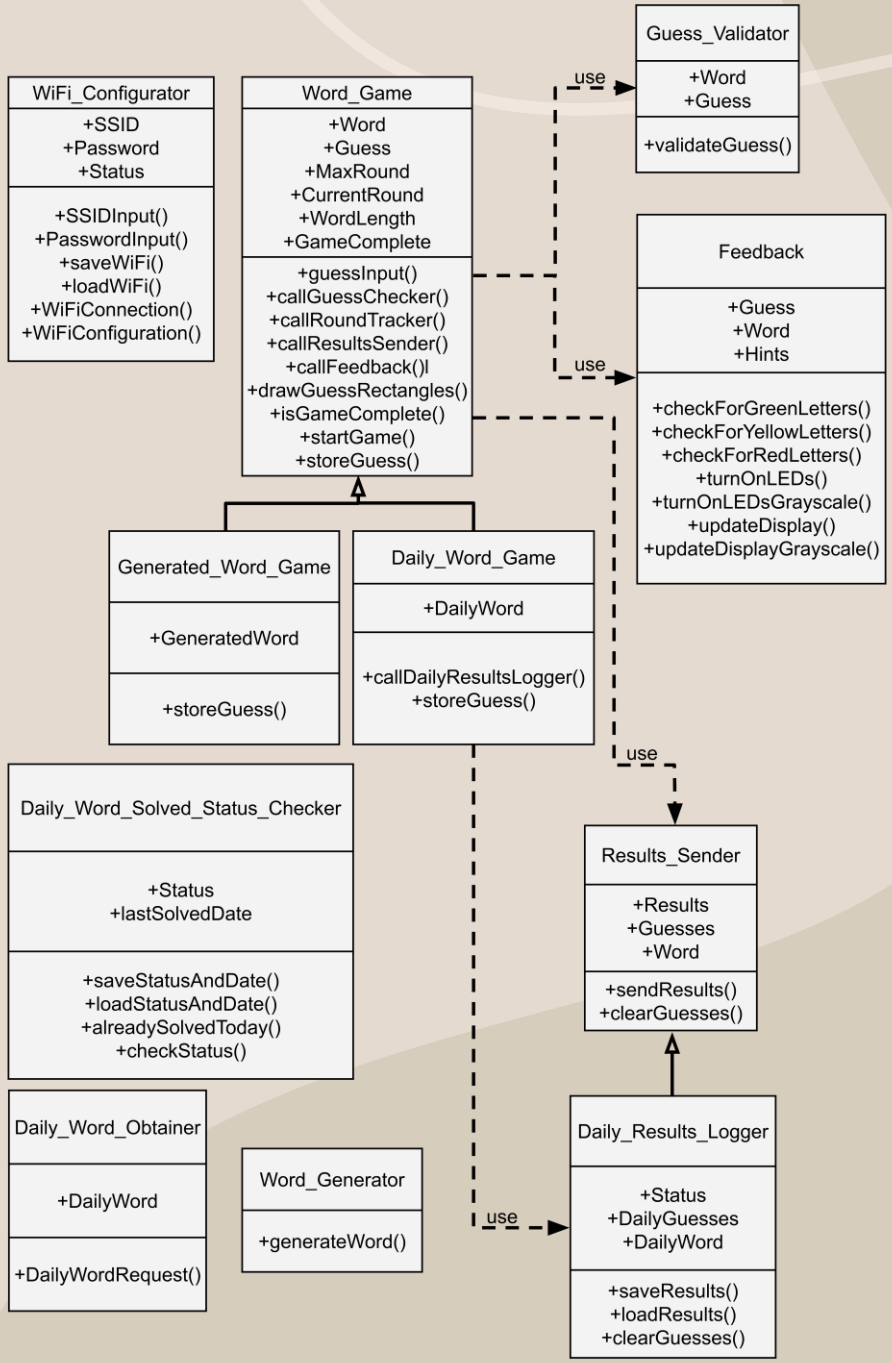


Software Design

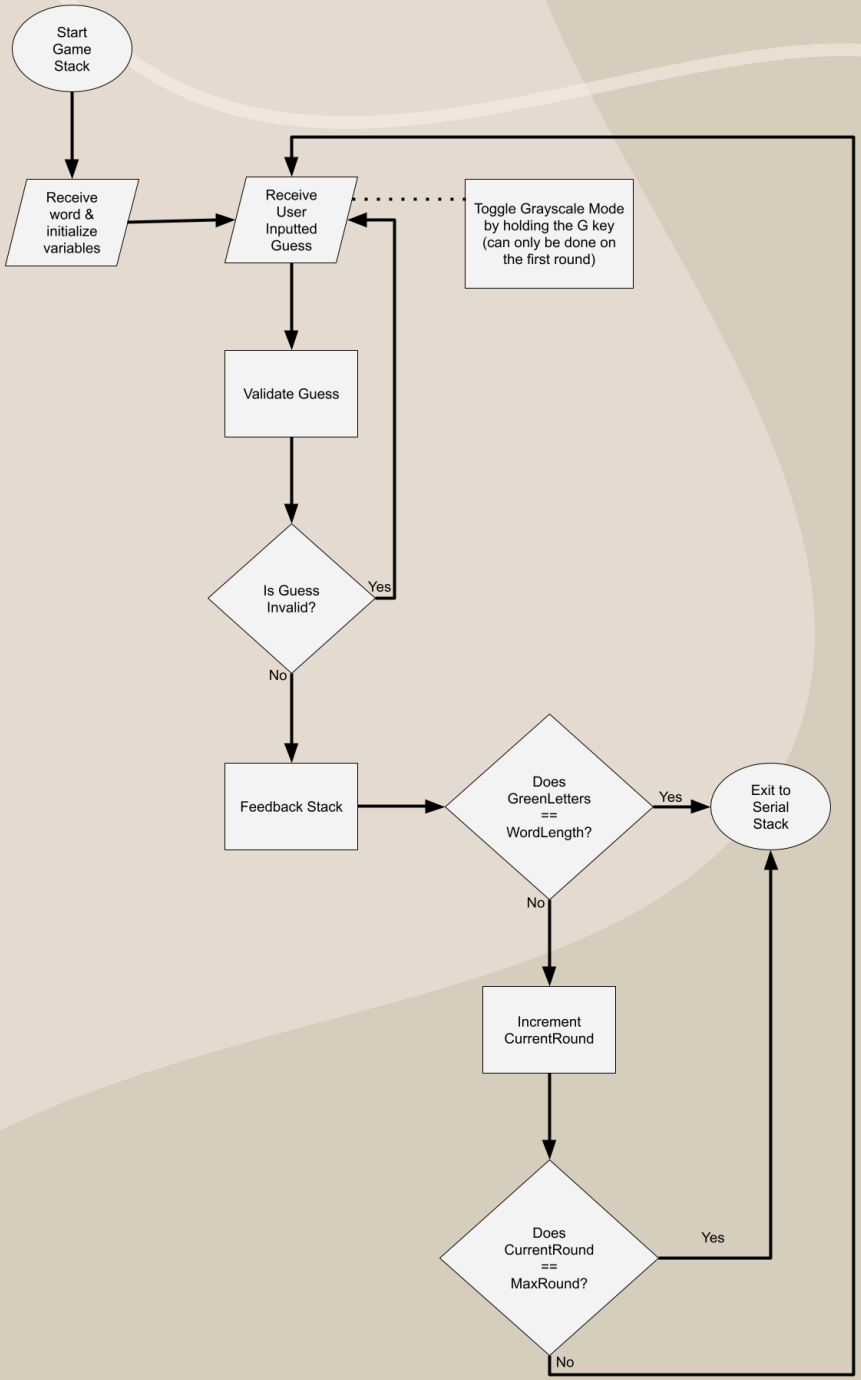




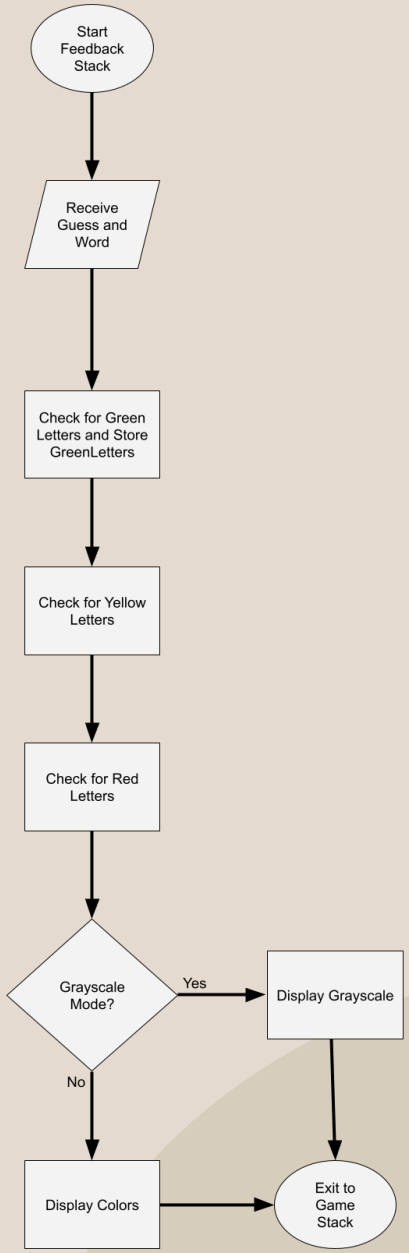
Software Design

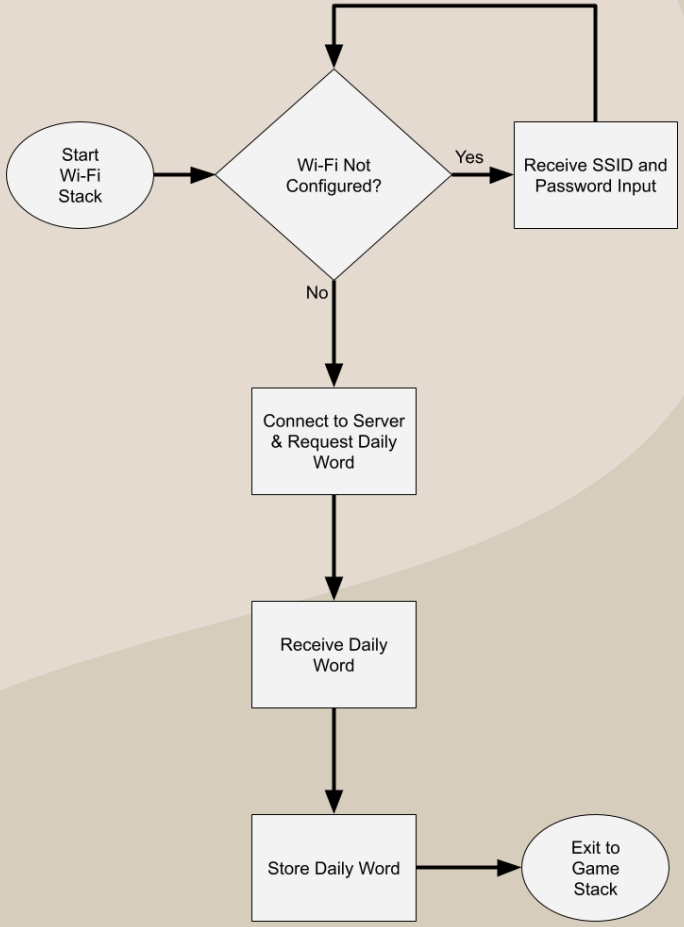


Software Design



Software Design

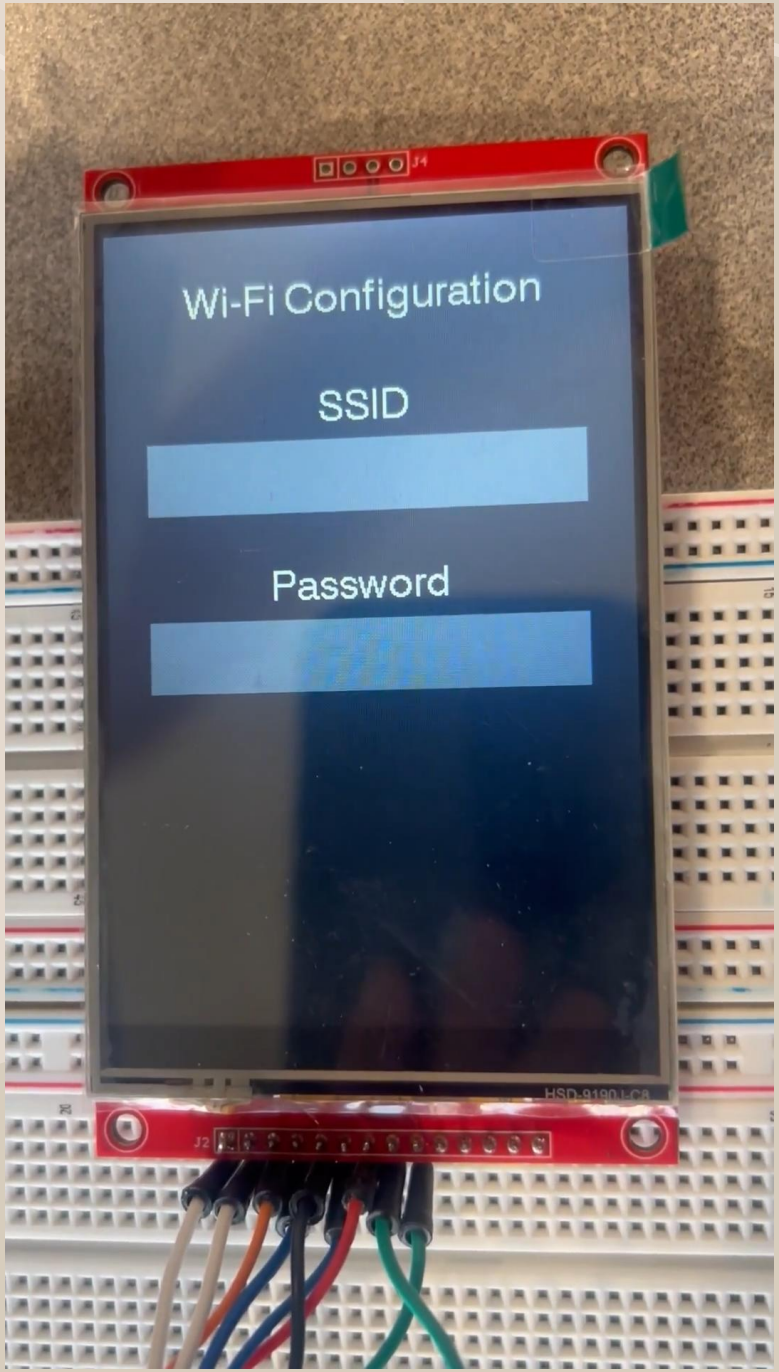




Software Design

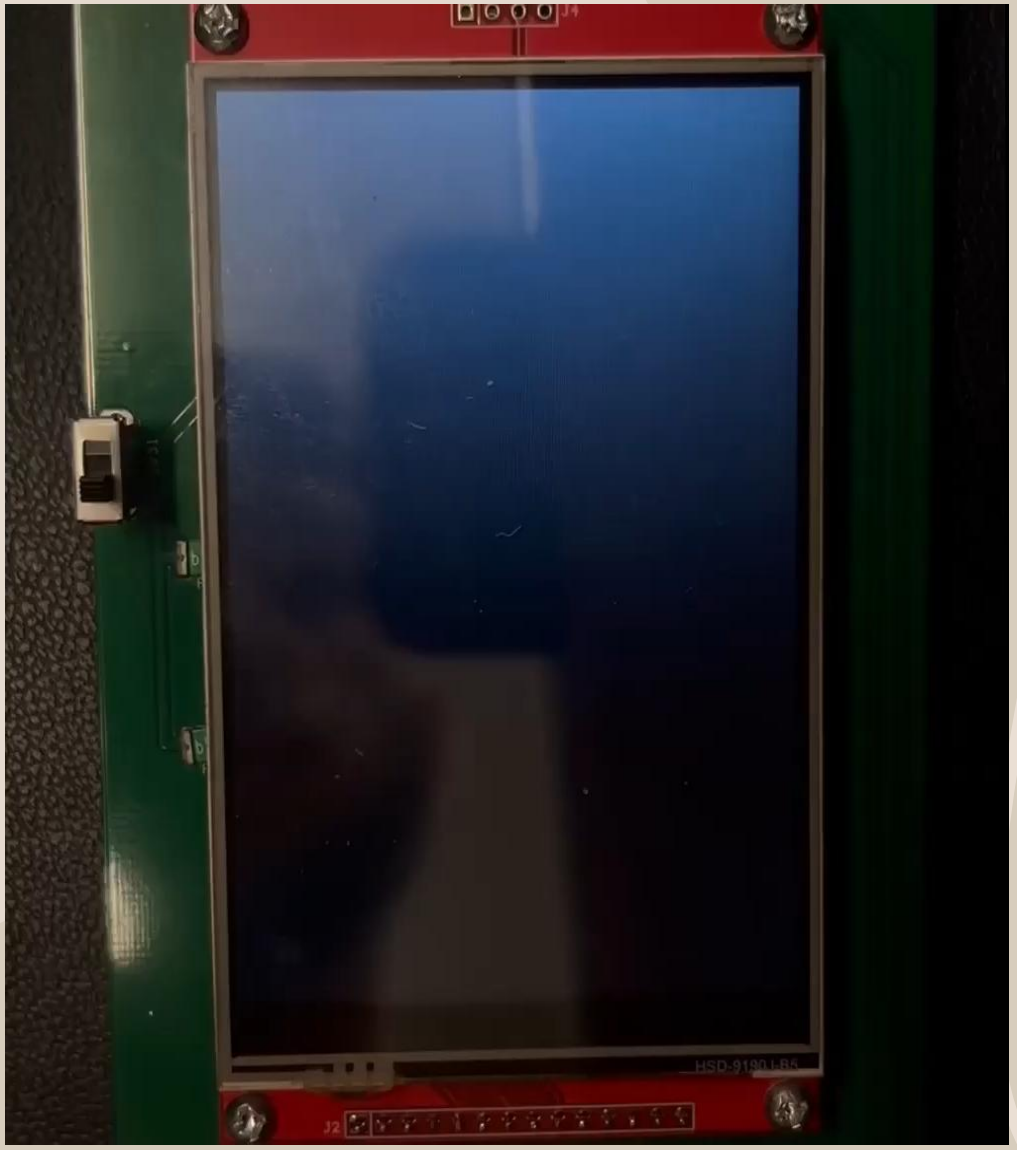
Prototyping and Testing

- The first video is from an earlier state of development and shows display operation with a functional Wi-Fi Configuration screen and a Daily Word Game being played



Prototyping and Testing

- The second video is from a later state of development, and it shows the Auto-Connect screen, Wi-Fi Configuration screen, and a bit of a Daily Word Game being played



Prototyping and Testing

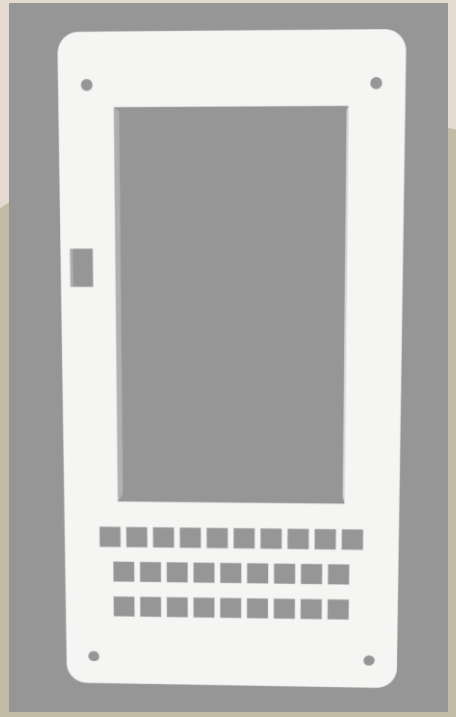
- Arduino's verification testing feature was used which helped with fixing up code.

- Different issues were encountered like Serial not working after including display code and game/feedback display code not displaying correctly

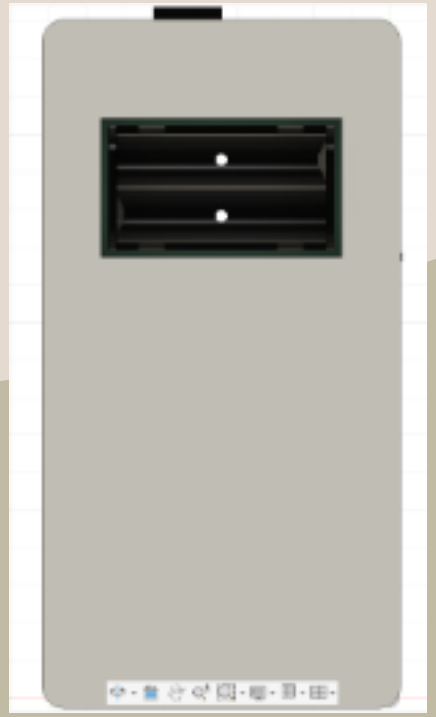
- These had different solutions. The Serial one was as simple as changing the port used on the Dev Kit and the latter involved multiple things like using fillScreen() instead of partial screen updating and letting the correct guess not leave early from the feedback stack

Enclosure Design

Top Case

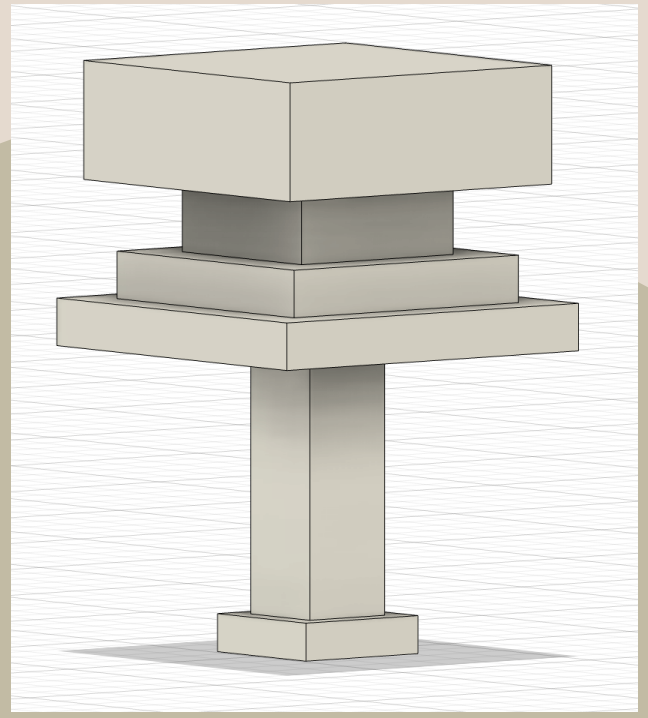


Bottom Case

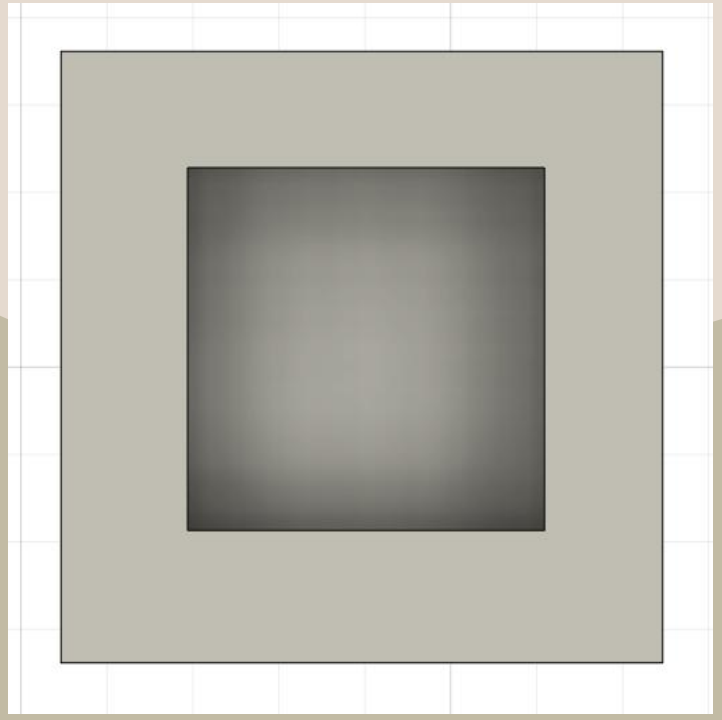


Key Integration

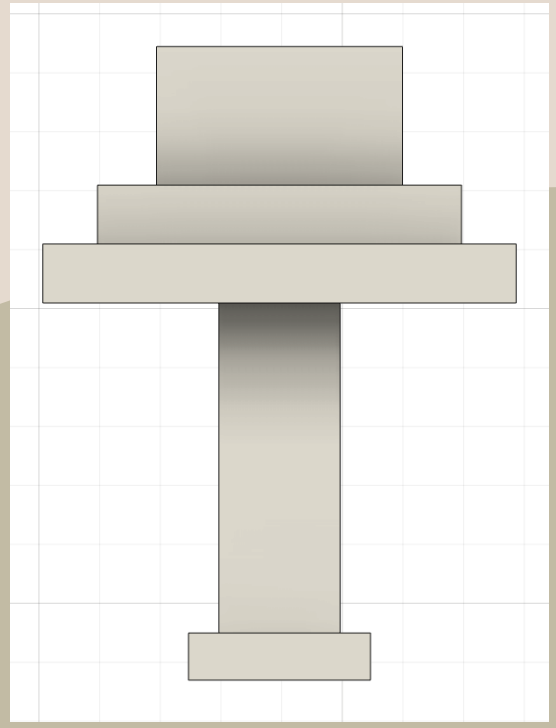
Keycap Assembly



Key Cutout (Top Case)



Key Stem



Budget

Component	Quantity	Price
PCB Fab. & Shipping	1	$183.65
ESP32-S3-DEVKITC-1-N8R8	2	$30
ESP32-S3 WROOM Module	3	$18.39
Tactile Switches	40	$14.35
TFT LCD Display	2	$38.32
USB Connectors	1	$6.99
Voltage Regulators	2	$80
RGB LEDs	40	$14.04
NiMH AA Battery Pack	2	$69.65
M3 Screws	1	$9.99
Enclosure (PLA Filament)	2	$46.99
Keycap Labels (Stickers)	1	$12.99
Total		$525.36

Work Distribution

Element	Primary Assignee
Schematic Design	Nala
PCB Design/Assembly	Nala
Software	Daniel & Hilda
Enclosure	Hilda

Thank You

Group 36