

# EduGRID MicroGrid Demonstrator Final Report

By

Ahmet Colak

Jason Hart

Srijan Kunta

Final Project Report for ECE 445, Senior Design, Spring 2026

TA: Abdullah Alawad

6 May 2026

Project No. 70

## Abstract

This report documents the design, implementation, and verification of the EduGRID MicroGrid Demonstrator, a tabletop educational system intended to teach users about electric grid protection, fault isolation, power factor correction, and grid stability. The system uses an ESP32-S3 microcontroller to run a software-defined grid simulation while physical switches, display feedback, LEDs, and a buzzer allow users to interact with different grid fault scenarios. The final board was designed to demonstrate how user actions affect fault clearing, load shedding, generation balance, and power factor correction. This report describes the final system architecture, major design decisions, changes from the original design, subsystem implementation, verification results, costs, schedule, and future improvements.

# Contents

- 1. Introduction ..... 1
  - 1.1 Problem ..... 1
  - 1.2 Solution ..... 2
  - 1.3 High Level Requirements ..... 2
- 2. Design ..... 2
  - 2.1 Block Diagram ..... 3
    - 2.1.1 Block Diagram Subsystems ..... 3
      - 2.1.1.1 Power Subsystem ..... 3
      - 2.1.1.2 Control Subsystem ..... 3
      - 2.1.1.3 Processing Subsystem ..... 3
      - 2.1.1.4 Status Devices ..... 4
      - 2.1.1.5 User Interface Subsystem ..... 4
      - 2.1.1.6 Design Alternatives ..... 4
    - 2.2 Subsystem Requirements ..... 4
      - 2.2.1 Power Subsystem ..... 4
      - 2.2.2 Controls Subsystem ..... 5
      - 2.2.3 Processing Subsystem ..... 6
      - 2.2.4 Status Devices Subsystem ..... 8
      - 2.2.5 User Interface Subsystem ..... 9
    - 2.3 Tolerance Analysis ..... 10
- 3. Design Verification ..... 12
  - 3.1 Power Subsystem Verification ..... 12
  - 3.2 Processing and User Interface Verification ..... 12
  - 3.3 Status Devices Verification ..... 12
  - 3.4 RC/RL Component Identification Verification ..... 12
- 4. Costs ..... 13
  - 4.1 Parts ..... 13
  - 4.2 Labor ..... 13
  - 4.3 Schedule ..... 14
- 5. Conclusion ..... 15

5.1 Accomplishments.....	15
5.2 Uncertainties.....	15
5.3 Ethical considerations.....	15
5.4 Future work.....	15
5 Citations .....	16
Appendix A Requirement and Verification Tables .....	17
Appendix B Requirement and Verification Results Tables .....	21

## 1. Introduction

The complexity of power grid protection and the role that power engineers play in designing it is not well supported or introduced in early education. Even within universities, grid infrastructure design and methods of protection for our grid is a subject where many students go through their whole educational careers without learning.

We propose a solution to this gap in education by creating a product aimed at all levels of education that give students varying levels of complex grid-scale problem solving opportunities. All of this will be achieved within a single tabletop package that provides numerous fault clearing objectives of increasing difficulty for the student, allowing for an intuitive hands-on approach to learn about the broad range of risks and corrective measures for our grid.

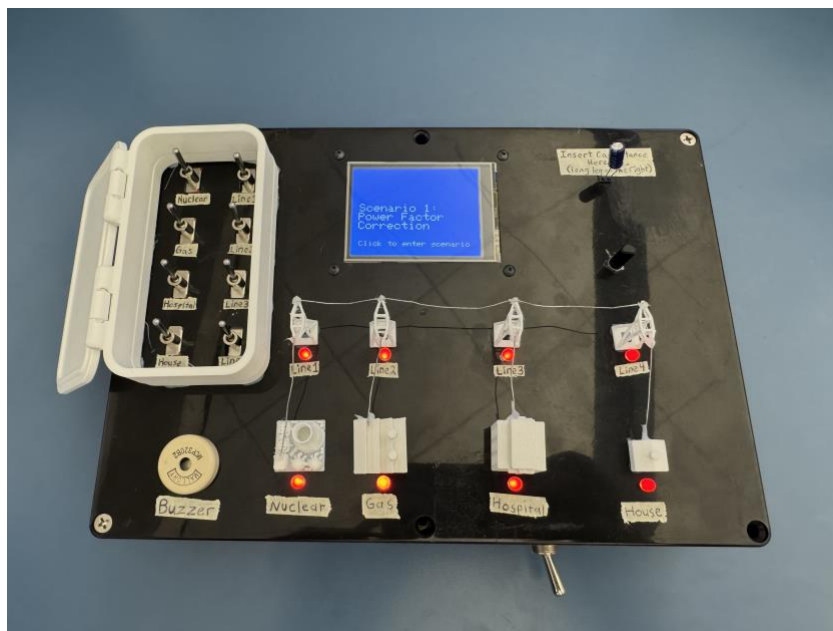


Figure 1: Board Design

### 1.1 Problem

Students often have limited understanding of how the electric power grid is designed to stay safe and reliable, especially the protection systems, breakers, relays, and fault isolation methods that prevent small failures from becoming large outages. Because these concepts are not taught in a fun and accessible way, many students do not see what power engineers actually do or why the field matters, which can reduce interest in pursuing power and energy careers. Recent large scale outages, such as the winter storm related failures experienced in Texas, show how grid reliability, planning, and protection directly affect daily life and the safety of the public, highlighting the need for clear, hands-on educational tool.

## 1.2 Solution

EduGRID is an interactive tabletop power grid that lets users visualize power flow from sources to loads while using physical controls to respond to grid events. The system simulates real power grid problems, including feeder short circuits, ground faults, arc faults, load imbalance, generation imbalance, and power factor correction. An ESP32-S3 state machine tracks the selected scenario, breaker states, fault status, LED states, and simulated grid values such as voltage, current, frequency, and power factor.

For the power factor correction scenario, users select capacitors or inductors, and the PCB identifies the component value using RC/RL time-constant measurements. The microcontroller uses the measured value to calculate the simulated power factor and displays feedback such as “Meets Target,” “Increase C / Decrease L,” or “Decrease C / Increase L.” By combining physical controls, display feedback, LEDs, and a buzzer, the system provides an intuitive way for students to learn how power engineers isolate faults, balance generation and load, and improve grid performance.

## 1.3 High Level Requirements

1. The user can select and clear the five available fault scenarios (permanent faults, transient faults, power factor correction, load shedding, selective generation).
2. The component identification circuit accurately computes and displays the value on the screen within 15% error.
3. The on-board display shows the relevant information for each fault scenario (power factor current state and target, frequency, voltage, and current simulation values).

## 2. Design

There were many alternatives possible for the design of this product. A few that we considered included a fully stepped down AC system relying on real time voltage and current measurements on multiple parts of the board to inject faults, check for cleared states, provide power factor correction measurements, and real time frequency monitoring. Another alternative was to create a software defined grid model that takes user input on buttons and components to update the model simulation and reflect changes to fault status visually. We chose to create a middle ground that involves a full DC based microgrid system that will include hardware and software defined solutions.

The hardware defined systems within the PCB will measure and compute the RC/RL time constant from user input capacitance and inductance through RL/RC circuit, OLED display of state-relevant grid quantities, and MOSFET logic to drive the buzzer system, LEDs, and current controlled rails. The software defined systems include our grid simulated measurements of voltage, current, frequency, load balancing, and power factor triangle.

## 2.1 Block Diagram

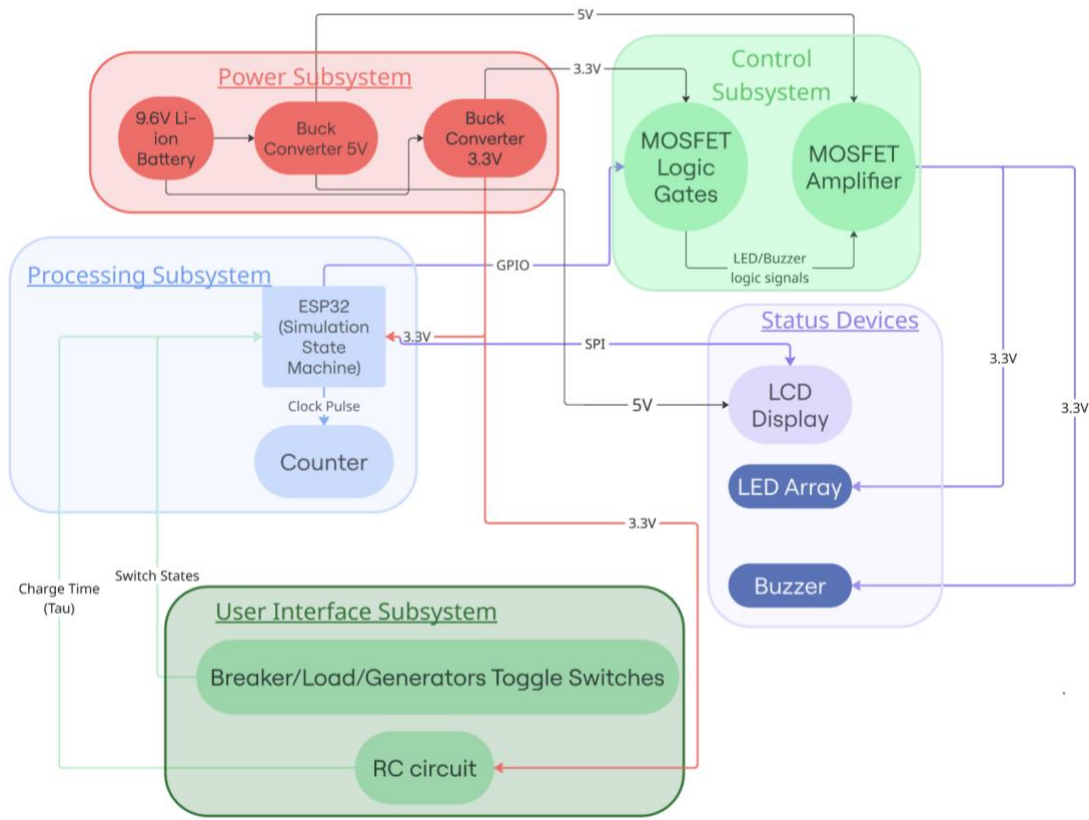


Figure 2: Block Diagram

### 2.1.1 Block Diagram Subsystems

#### 2.1.1.1 Power Subsystem

The power subsystem handles power distribution and DC/DC regulation. It provides stable regulated power rails for every subsystem. This allows the ESP32-S3, display, LEDs, noise devices, and indicators to operate reliably. This subsystem converts the main input supply of 6 AA batteries connected in series with a nominal voltage of 9.6V into a clean 5 V rail and a clean 3.3 V rail, with protection and power indication.

#### 2.1.1.2 Control Subsystem

The control subsystem contains the MOSFETS used for both low level logic control as well as the current amplification to drive the buzzer and bulk LED array. This ensures that the ESP32 does not experience any brownouts due to excess current draw. This subsystem communicates to the status devices subsystem to both drive the high current for the LEDs as well as to provide the buzzer component with the required voltage to sound only when the user has made a mistake in each fault scenario.

#### 2.1.1.3 Processing Subsystem

The processing subsystem acts as the brain of the system, storing the simulation's electrical values, feeder states, fault type, power factor, frequency, voltage, display text, and breaker status. It drives the control subsystem for MOSFET-based LED switching and the status display subsystem for the LCD, which

shows feeder voltage, frequency, and power factor based on physically added capacitance or inductance on the board.

#### **2.1.1.4 Status Devices**

The status devices subsystem contains the small LCD display, the LED array, and the buzzer. This subsystem is used to display all of the important information of the simulation. The LED array will be used to display which power lines and loads/generators are active on the grid. The LCD display will show information such as the power factor, voltage, frequency, and status information about the user's progress in the current level. The buzzer will sound if the user makes an incorrect choice for solving the given fault scenario.

#### **2.1.1.5 User Interface Subsystem**

The user interface subsystem is comprised of parts of the board that are controlled by the user. These include: the capacitors and inductors the user chooses for power factor correction, breakers the user must trip to separate faulty loads from sources, generator toggle switches that the user can decide to cut from the system to balance the power/frequency of the simulated grid.

#### **2.1.1.6 Design Alternatives**

Two main alternatives were considered: a stepped-down AC grid model and a fully software-based simulation. The AC model was closer to a real grid but created more safety, cost, and debugging concerns. The software-only option was simpler but did not provide enough hands-on interaction. The final design uses low-voltage hardware with software-defined grid behavior, giving the user physical interaction while keeping the system safe and practical.

## **2.2 Subsystem Requirements**

### **2.2.1 Power Subsystem**

The power subsystem converts the input from a six-cell AA battery pack into two independent regulated rails, +5.0 V and +3.3 V, used by the rest of the system. The battery pack is treated as a portable DC source with a nominal voltage of 9.6 V. To improve efficiency and isolate sensitive logic from higher-current loads, the subsystem uses two parallel TPS54202 buck converters, each connected directly to the battery input, rather than deriving the 3.3 V rail from the 5 V rail. The +5.0 V rail supplies the LED power stages, buzzer, and any 5 V peripherals, while the +3.3 V rail supplies the ESP32-S3, Display, and all control circuitry. This model reduces unnecessary conversion loss, prevents the processing subsystem from depending on the 5 V rail for regulation, and improves stability during transient load events such as simultaneous LED switching and activity on the ESP32-S3. The 5 V converter includes an enable-divider-based undervoltage setup so it disables when the battery is too low to regulate 5 V reliably, while the 3.3 V converter enable pin is tied directly to the battery so that the processing subsystem remains powered over the widest possible battery range. Each converter includes dedicated input bypass capacitors, bootstrap capacitors, an output filter network, and a feedback network sized for its target output voltage. Together, these circuit elements allow the subsystem to provide stable power rails that support reliable operation of every functional block in the design.

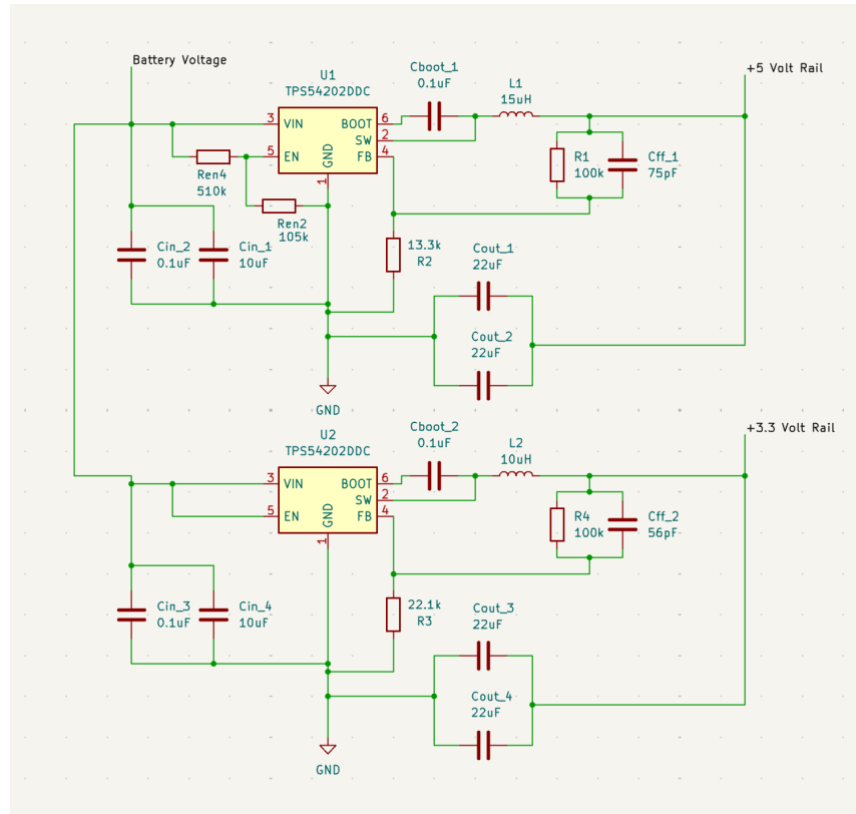


Figure 3: Power Subsystem KiCAD Schematic

(Requirement and Verification Tables in Appendix A)

### 2.2.2 Controls Subsystem

The control subsystem facilitates the function of our entire project by allowing the safe operation of our LED array, buzzer component, and the safekeeping of our microcontroller. Without this system, the current draw from our MCU will cause a brownout in the device. The control subsystem requires the sufficient supply of voltage from the power subsystem as well as the correct GPIO logic signals from the ESP32. Without these, then the logic and amplification capabilities of the control subsystem would not function. This would result in the status device subsystem failing to send the required current and voltage to the necessary LEDs and buzzer when needed.

- MOSFETs used to drive higher current for LEDs and buzzers (controlled by esp32) along with current limiting resistors to avoid esp32 brownout

The success of the controls subsystem is dependent on its communication with the status devices subsystem to deliver safe and reliable current flow to each device. Additionally, this subsystem is required to read control signals from our processing subsystem's MCU to coordinate the digital logic and amplification of current through the circuit.

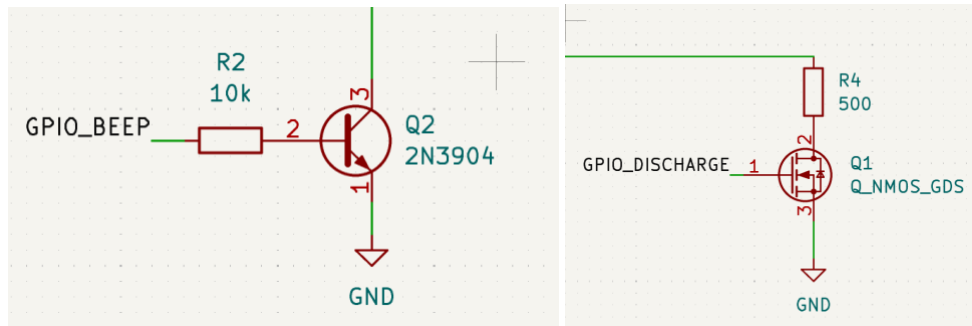


Figure 4: Control Subsystem KiCAD Schematic

### 2.2.3 Processing Subsystem

The processing subsystem allows the entire design to operate according to the user inputs and internal current state of the simulated model. The processing subsystem requires the functionality of the power subsystem to distribute a clean 3.3V rail to the ESP32 as well as requiring the simulated model and state machine to be accurate in terms of the current state and corresponding logic signals associated with each fault scenario.

Simulated fault scenarios (to be picked by the user)

- Ground fault (Line touches earth/ground) (Fallen tree), blown fuse
  - When the permanent fault scenario is activated, the MCU flags the power line feeder segment as faulted in the grid model and updates the board's display. Fault detection is implemented by checking whether the faulted line is down within the MCU, and the red LED illuminated accordingly. Protection is seen as successful when the user opens the correct nearest surrounding breakers to the feeder segment so the faulted section is optimally isolated, or else the red fault LED remains on and the buzzer alerts during solution check. The user controls the state of the breakers through toggle switches on the bottom right corner of the board, where each one is labeled with its corresponding breaker.
- Lightning-induced fault, arc faults
  - Transient fault scenarios where the user must wait until either the primary automatic breaking isolates the fault or the secondary breakers downstream (on a time delay) break the fault upon primary breaker failure (demonstrating real world operation to the user). If the user reclosed the breakers too early (before the simulated clearing delay), the fault reappears. If they wait and reclose at the correct time, the line returns back to functional.
- Undervoltage (voltage and frequency droop under heavy load)
  - A load shedding scenario where too many loads are on the grid (factories, hospitals, houses) and the frequency and voltage droop. The user must selectively load shed to raise the frequency back to 60 hz and correct the voltage droop. This voltage droop and frequency change will be displayed to the user as the scenario occurs. Successful

completion is determined when the correct load option (residential) is load shed (3D printed house piece is removed from the board) to preserve high priority loads just like in real grid operation.

- Generation disconnect to solve frequency droop
  - A frequency stability scenario where the frequency rises above 60 Hz because there is too much generation and not enough loads on the grid. The user selectively turns off generation sources that are quick to turn off and on (like natural gas turbines), and the MCU simulates the frequency response to load changes. Sources that are difficult to start up, like coal and nuclear behave differently than dispatchable sources like natural gas turbine generators. The display shows the resulting simulated frequency deviation and recovery. The user will correctly solve the scenario when they choose to disconnect the natural gas turbine from the set of generation sources on the board. If any other source is disconnected (nuclear or solar/wind) then the buzzer will sound.
- Power Factor Control
  - A power factor correction scenario where the user adjusts the system's reactive behavior by inserting capacitors/inductors into the board. The PCB physically measures the inserted L/C using DC RC/RL timing circuits by measuring the time it takes for an inserted voltage to reach 63.2% of the max across a known resistor to find tau. The MCU uses the measured values in a mathematical PF correction model referenced to a predefined target PF for the user to attempt to match. The display shows the computed PF and recommends actions ("increase C / decrease L" or "decrease C / increase L") until the target PF is met and the light goes green to show success.
- Correct isolation conditions
  - The microcontroller will initialize fault scenarios that will require the user to manually trip breakers nearest to the fault on all necessary ends to isolate the fault. Correct isolation will be detected using internal logic that determines if the nearest breakers in the zone of protection were tripped. If not, then the user will hear a buzzer sound upon checking their solution correctness while seeing red lights continue to illuminate the faulted region.
- For the fault scenarios involving automatic tripping, the user's success will be checked upon their decision of when to reclose the circuit. If they reclose the breakers too early before the primary and backup breaker protection has activated, then the buzzer will alarm. In this case, they must wait until the LEDs on the board show signs of effective simulated isolation before reclosing the breakers and hitting the test button to check accuracy.
- Our software defined symmetrical fault types include short circuit faults on various distribution lines, ground faults, or arc faults. The user will see the causes of these faults displayed to them via the on board display panel, whether caused by animal interference, vegetation, or lightning/earthquakes.
  - LEDs Indicates state of line (faulted, cleared, in progress) (Red, Green, Blue)
  - "In progress" signals that the automatic breaker scenario is attempting to isolate the fault using primary breaker protection before its time delay triggers the secondary breaker protection to trigger. Once it's isolated, the user is then

responsible for reclosing the correct breakers to bring the line back online. (For the automatic tripping demonstration scenario only)

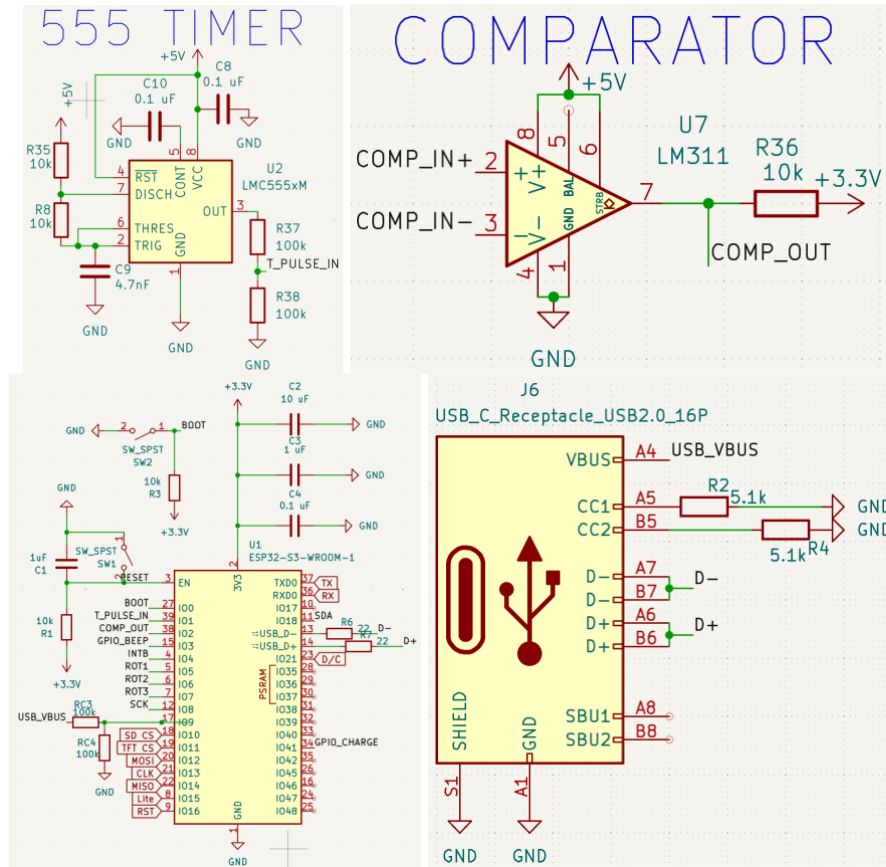


Figure 5: Processing Subsystem KiCAD Schematic

### 2.2.4 Status Devices Subsystem

The status devices subsystem is the primary user interface for the simulation, giving users real-time visual and audio feedback. It translates the internal logic of the simulation such as the power line status, electrical metrics, and game logic, and turns it into human-readable signals so the user can easily understand how their actions affect the simulation in real time.

The display must communicate with the ESP32 via the I2C protocol at a rate of at least 100 kbps, preferably 400 kbps for smooth display operation and to ensure that the display lag is minimal. All status components should operate within a 3.3V± 5% range to prevent damage to the ESP32 GPIO pins or cause logic level mismatches. The CUI CEM-1203 buzzer device must work properly and provide a Sound Pressure Level of at least 85 dB to ensure audibility of the sound.

This subsystem is functionally dependent on the 5V rail from the power subsystem to power the Display and 3.3V for the LEDs. If the voltage drops below 3.0V, the display will fail. It is also dependent on the processing subsystem to provide the SPI communication signals, (SCK, MOSI, MISO), as well as the required control lines (CS, D/C, RST) and GPIO triggers. Without the ESP32's SPI clock, data

transmission, and control logic, the display and status indicators would remain in an idle or undefined state.

This subsystem does not affect the rest of the subsystems too much. If this subsystem fails, the simulation simply goes “blind”, in that the user will be unable to see the state of the simulation demo, and will not understand what is going on, even if the simulation continues to run correctly. The display will be connected to the PCB via a 2x7 connector.

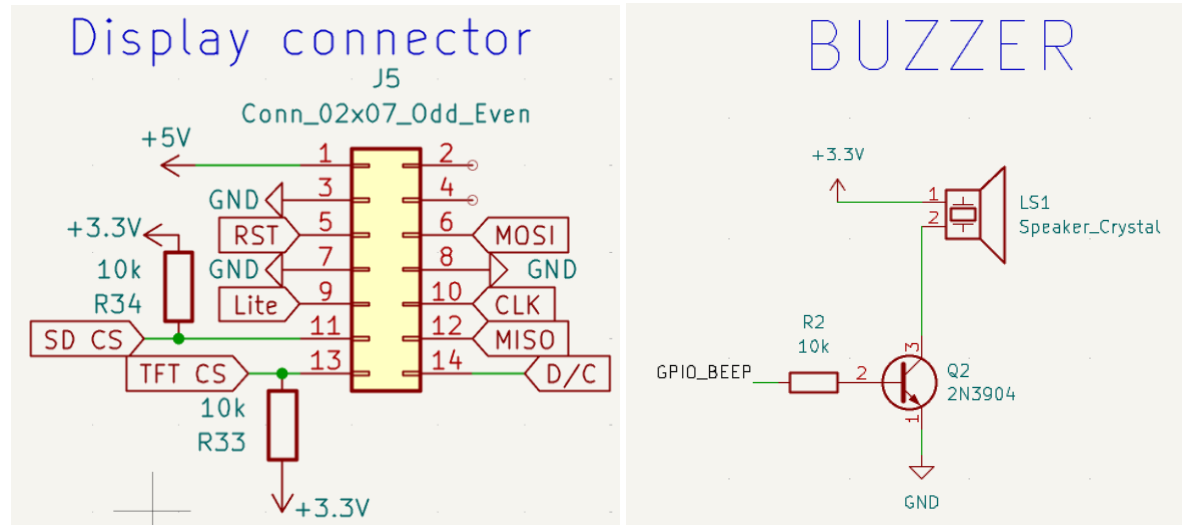


Figure 6: Status Devices Subsystem KiCAD Schematic

### 2.2.5 User Interface Subsystem

The user turns the rotary selector and chooses a fault scenario to initialize on the board, which is visible from the display. Depending on which of the five fault scenarios the user chooses, the MCU initializes the corresponding components on the board to show as faulted. The screen reflects this change and gives the user all the necessary information about the current state of the grid to correct the faults. The success of this subsystem requires the communication to the processing subsystem to allow for the current state of the simulation model as well as the board components to be accurately displayed and considered in the fault clearing checks.

For power factor correction, the PCB will physically measure the user chosen component using DC RC and RL time constant circuits within the PCB. A known resistor will be used with a step voltage input into it, and a 555 counter will capture the time it takes to reach 63.2% of the max voltage across the resistor. This will be done by having the MCU start the 555 and the comparator will have a known threshold voltage that it is comparing against the active voltage across the resistor in the RC/RL circuit. Once the comparator sees the resistor voltage hit the set threshold that is 63.2% of the step voltage applied, then the comparators output triggers the 555 timer to stop and the MCU reads the length of the pulse from the timer. This time is read in the simulation as tau and used to compute the capacitance or inductance based on the equation associated with a capacitors discharge pattern (gradual) or an inductors discharge pattern (instant). This will give us the tau value that the simulation model will use to

calculate the value of the capacitor or inductor that the user chose to add to correct the power factor. These measured C and L values will be used by the MCU to compute the new PF and reflect it on the display.

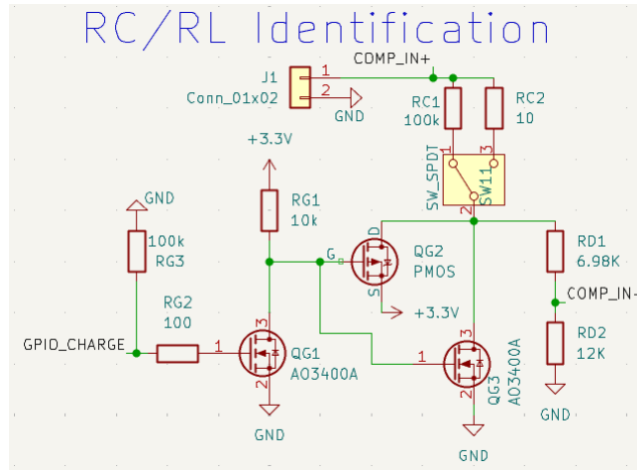


Figure 7: User Interface Subsystem KICAD Schematic

## 2.3 Tolerance Analysis

Because our PCB identifies user-inserted capacitors and inductors by measuring an RC/RL time constant, the measurable component range is limited by the microcontroller's ability to time short intervals. The ESP32 does not measure  $C$  or  $L$  directly, instead, it will timestamp when the RC or RL response reaches a fixed fraction of its final value the 63.2% point. This measured time is the time constant  $\tau$ , which is then converted into  $C$  or  $L$  using  $\tau = RC$  or  $\tau = L/R$ . If the selected capacitor or inductor is too small,  $\tau$  becomes very small and the current increases too quickly. In this case, the measured time is on the same order as the counter's tick size and software latency, so the measurement becomes inaccurate meaning one or two timer ticks of error becomes a large percentage of  $\tau$ . This directly increases the relative error in the computed component value and can cause the system to misclassify whether the user's  $C$  or  $L$  choice meets the power factor correction target. We must constrain the selectable capacitor and inductor values so that  $\tau$  remains sufficiently larger than the timer resolution, making sure that the 63.2% crossing occurs over a time interval that can be measured accurately.

RC step response for capacitance measurement

$$v_C(t) = V_{step} \left( 1 - e^{-\frac{t}{RC}} \right)$$

At the 63.2% point:

$$v_c(\tau) = 0.632 V_{step} \Rightarrow \tau = RC$$

Capacitance estimate from measured  $\tau$  and known  $R$ :

$$C = \frac{\tau}{R}$$

$$t = -RC \ln\left(1 - \frac{V_{th}}{V_{step}}\right) \Rightarrow C = \frac{t}{-R \ln\left(1 - \frac{V_{th}}{V_{step}}\right)}$$

A second tolerance related risk is possibilities of brownouts to the ESP32-S3 because it can not directly source the current needed for several LEDs without creating transient load steps that can cause a brownout. To prevent this, GPIO pins are used only as low current control signals with pull-up/pull-down resistors to guarantee defined startup states and avoid floating nodes. The LED current is handled by MOSFETs, which let the microcontroller control higher-current loads without stressing the GPIO improving rail stability and system reliability under worst case LED activity.

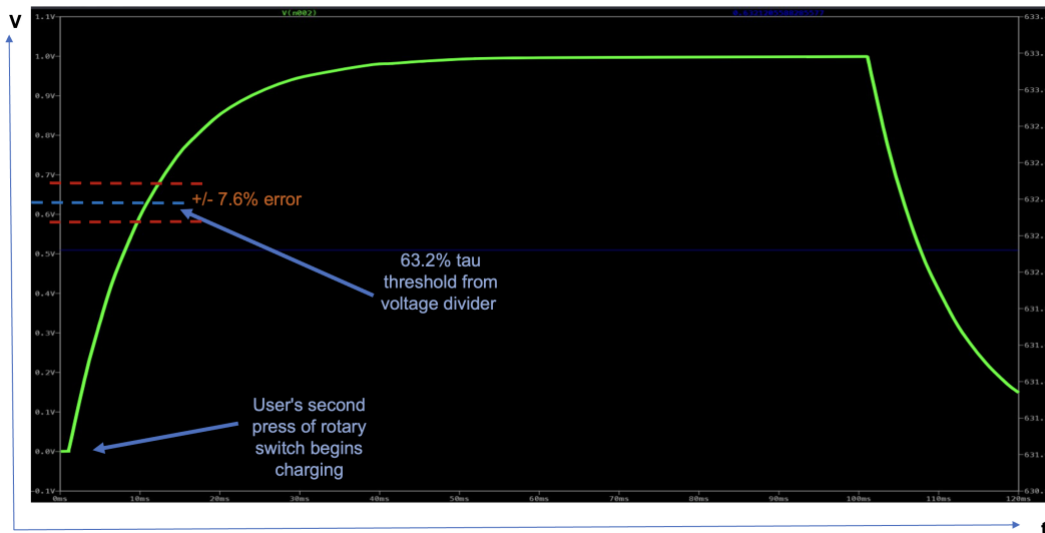


Figure 11: RC circuit 63.2% Simulation Full Graph

### 3. Design Verification

The requirements were verified by testing each subsystem against the requirements established during the design process. The verification process focused on confirming that the power subsystem supplied stable voltage rails, the ESP32-S3 correctly executed the simulation state machine, the user interface responded to physical inputs, the status devices provided clear feedback, and the RC/RL measurement circuit correctly identified user-selected reactive components.

The final system satisfied the three high-level project requirements. First, the user was able to select and clear the five available scenarios. Second, the component identification circuit measured and displayed the tested capacitance and inductance values within the required 15% tolerance. Third, the on-board display showed the relevant information needed for each scenario, including scenario status, corrective instructions, and simulated grid quantities.

(Verification Tables are in Appendix A)

#### 3.1 Power Subsystem Verification

The power subsystem was tested by powering the board from the battery input and measuring the regulated voltage rails during operation. The 5 V rail measured 5.03 V, which met the required 4.5 V to 5.5 V range. The 3.3 V rail measured 3.29 V, which met the required 3.10 V to 3.50 V range. The ESP32-S3 remained powered and did not reset during scenario switching and status device updates.

#### 3.2 Processing and User Interface Verification

The processing and user interface subsystems were tested by selecting each scenario and checking the ESP32-S3 state machine response. All five scenarios were selected successfully. The system recognized the breaker, load, and generator inputs, updated the display, and produced the correct success or failure response based on the user's action.

#### 3.3 Status Devices Verification

The status devices were tested during scenario operation. The display showed the correct scenario information for all five scenarios. The LEDs indicated the tested grid states, and the buzzer activated for incorrect responses while remaining off for correct responses.

#### 3.4 RC/RL Component Identification Verification

The component identification circuit was tested using known capacitors and inductors. The capacitors tested were 10  $\mu\text{F}$ , 33  $\mu\text{F}$ , and 47  $\mu\text{F}$ . The inductors tested were 15 mH, 10 mH, and 5 mH. For each component, the ESP32-S3 measured the RC or RL response, calculated the component value, and displayed the result.

All tested components were measured within the required 15% tolerance. This verified that the RC/RL measurement circuit was accurate enough for the power factor correction scenario.

## 4. Costs

### 4.1 Parts

Parts Costs Table

Part Names & purchase link	Part	Manufacturer	Unit Cost (\$)	Quantity	Actual Cost (\$)
<a href="#">switch</a>	EG1218	E-Switch	\$0.72	1	\$0.72
<a href="#">Buck converter</a>	TPS54202DDCR	Texas Instruments	\$1.17	2	\$2.34
<a href="#">Voltage Regulator</a>	TLV75533PDBVR	Texas Instruments	\$0.36	1	\$0.36
<a href="#">diode</a>	SS14TR-ND	Onsemi	\$0.29	1	\$0.29
<a href="#">Red LED</a>	WP7113SURDK14V	Kingbright	\$0.66	10	\$4.53 (Bulk)
<a href="#">N-MOSFET</a>	CSD15380F3	Texas Instruments	\$0.32	1	\$0.32
<a href="#">P-MOSFET</a>	CSD25501F3	Texas Instruments	\$0.38	1	\$0.38
<a href="#">ESP32-S3</a>	1965-ESP32-S3-ND	Espressif Systems	\$1.85	1	\$1.85
<a href="#">Display Board</a>	1528-1469-ND	Adafruit	\$29.95	1	\$29.95
<a href="#">Capacitor 200uF</a>	647-UTH2G201MND	Nichicon	\$5.40	1	\$5.40
<a href="#">Capacitor 0.1uF</a>	C320C104K5R5TA73	KEMET	\$0.30	10	\$1.77 (Bulk)
<a href="#">Rotary selector</a>	PEC11R-4220F-N0012	Bourns Inc.	\$1.89	1	\$1.89
<a href="#">Buzzer</a>	CUI CEM-1203	Same Sky	\$0.68	1	\$0.68
<a href="#">Switch</a>	100SP1T1B4M2QE	E-Switch	\$2.63	6	\$15.78
<a href="#">Power MOSFET</a>	CSD15380F3	Texas Instruments	\$0.32	10	\$1.94 (Bulk)
<a href="#">MOSFET</a>	CSD25501F3	Texas Instruments	\$0.38	10	\$2.32 (Bulk)
<a href="#">Binary Counter</a>	595-CD4040BE	Texas Instruments	\$1.02	1	\$1.02
	<b>Total</b>				<b>\$71.50</b>

### 4.2 Labor

Labor Costs Table

Name	\$ per hour	Hours per week	Total Weeks	Total Cost
Ahmet Colak	\$35	10	16	\$5600
Jason Hart	\$35	10	16	\$5600
Srijan Kunta	\$35	10	16	\$5600
Machine Shop	\$0	0.5	2	\$0
				\$16800

### 4.3 Schedule

Week	Work	Team member distribution
7 3/2	Order all necessary parts, finish first PCB design, design review, then submit for second round PCBway orders. Work on breadboard implementation	Ahmet: power and mosfets Jason: RC/RL, LEDs, buzzer Srijan: MCU, display, switches
8 3/9	Breadboard demo basic/partial functionality, Third round PCBway order if needed	Ahmet: power and mosfets Jason: RC/RL, LEDs, buzzer, combining PCB Srijan: MCU, display, switches, software, dev board testing
9 3/23	AFTER SPRING BREAK: get parts, solder PCB, build box, Fourth round PCBway order if needed	Ahmet: power and mosfets, box Jason: RC/RL, LEDs, buzzer, switches Srijan: MCU, display, soldering, software
10 3/30	Individual progress reports, software state machine, display UI, box setup, RC/RL circuit setup	Srijan: software state machine, display UI Ahmet: box setup, power Jason: RC/RL
11 4/6	Progress Demo, Team Contract assignment, combine software with RC/RL	Srijan and Jason: RC/RL software combination Ahmet: Power Testing
12 4/13	Get everything working together	All
13 4/20	Mock demo	All
14 4/27	Final demo	All
15 5/4	Final paper, presentation	All

## 5. Conclusion

### 5.1 Accomplishments

We successfully created a project that meets our high-level requirements by allowing the user to select, view, and solve 5 different grid fault scenarios. The 9.6V battery supply correctly feeds the buck converters to be stepped down to stable 3.3V and 5V rails, and the component identification circuit measures and computes values within our 15% error margin. The final functionality of our project delivers on our desired effect in providing a range of complex problem-solving scenarios to students interested in power engineering.

### 5.2 Uncertainties

Within the component identification circuit, we saw errors in component values of up to 7%. To expand our product's capability in question complexity, we would need to shrink this error. We found that the main sources of this error came from our code's implementation on the tau calculation. We defined the clock frequency variable ahead of time based on initial measurements. A better strategy for the future would be to have the microcontroller take a live measurement of the clock frequency as it runs so as to avoid any discrepancy in the calculation. Another source of error was the resistors that we used to make the 63.2% threshold for the comparator. If we had chosen resistors with tighter tolerances and more accurate values, we would have seen a resistor divider with a far closer 63.2% division. We found that the voltage drop across our current divider gave us 1.98V, while our ideal divided voltage should be 2.0856V since we fed the divider 3.3V, and expect 63.2% of that at the measured node.

### 5.3 Ethical considerations

EduGrid is tied to the IEEE Code of Ethics through point 2: the ethical obligation to improve people's understanding of the capabilities and societal implications of technology [1]. Our system is meant to be educational, so it must be an accurate representation of real-world systems that does not misinform users. In addition, the system should operate at low voltage with no exposed conductors within the user's reach, and electrical design choices (current limiting, protected rails, safe connectors) should reduce the risk of overheating, short circuits, or accidental contact. These safety protocols fulfill point 1 of the code, which requires a prioritization of safety and health of the public [1].

### 5.4 Future work

Improvements that we plan to implement would be to create a randomized selection of parameters and values for the given fault scenarios to improve the educational effectiveness of our product. Additionally, we plan to increase the number of fault scenarios by having the switches and LEDs repurposed for new fault conditions.

## 5 Citations

- [1] IEEE, "IEEE Code of Ethics," IEEE.org, June 2020. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>.
- [2] IPC, "IPC-2221A: Generic Standard on Printed Board Design," IPC.org, May 2003. [Online]. Available: [https://www-eng.lbl.gov/~shuman/NEXT/CURRENT\\_DESIGN/TP/MATERIALS/IPC-2221A\(L\).pdf](https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A(L).pdf).
- [3] OSHA, "Occupational Safety and Health Standards: Subpart S - Electrical," OSHA.gov, Jan. 2007. [Online]. Available: <https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910SubpartS>.
- [4] University of Illinois Urbana-Champaign, "ECE 330: Review 1—Single-Phase AC Circuits (Complex Power, PF, Reactive Power of L/C)," *Illinois.edu* (course notes), [Online]. Available: <https://courses.physics.illinois.edu/ece330/sp2021/review/review1.pdf>
- [5] OpenStax, "University Physics Volume 2—Power in an AC Circuit (Power Factor)," *OpenStax*, [Online]. Available: <https://openstax.org/books/university-physics-volume-2/pages/15-4-power-in-an-ac-circuit>.
- [6] Texas Instruments, B. Hauke, "Basic Calculation of a Buck Converter's Power Stage (Rev. B)," Application Report SLVA477B, Aug. 2015. [Online]. Available: <https://www.ti.com/lit/pdf/slva477>. [Accessed: Feb. 13, 2026].
- [7] Texas Instruments, "Power Supply Decoupling and Layout Considerations," Application Report SPRA906, *Ti.com*, [Online]. Available: <https://www.ti.com/lit/an/spra906/spra906.pdf>

## Appendix A Requirement and Verification Tables

**Power Subsystem Verification Table**

Requirement	Test Equipment	Verification Procedure	Pass Criterion / Recorded Result
The Power Subsystem shall operate from an input source between 6.0 V and 9.6 V applied to the Battery Voltage input.	Bench power supply	Replace the battery pack with a bench supply. Apply 6.0 V, 7.5 V, 9.0 V, and 9.6 V to the Battery Voltage net. Confirm both rails start and remain present.	+5 V and +3.3 V rails are present and within their specified ranges at all four input voltages.
The subsystem shall regulate the +5.0 V rail to 4.5 V to 5.5 V for load currents from 0 mA to 600 mA.	Bench power supply, resistor bank	Apply nominal input voltage. Sweep the +5 V rail load from 0 mA to 600 mA. Measure rail voltage at each load point.	Measured +5 V rail remains between 4.5 V and 5.5 V over the full load range.
The subsystem shall regulate the +3.3 V rail to 3.10 V to 3.50 V for load currents from 0 mA to 600 mA.	Bench power supply and a resistor bank.	Apply nominal input voltage. Sweep the +3.3 V rail load from 0 mA to 600 mA. Measure rail voltage at each load point.	Measured +3.3 V rail remains between 3.20 V and 3.40 V over the full load range.
During a transient event equivalent to 15 LEDs switching simultaneously, the +3.3 V rail shall not drop below 3.20 V, and the ESP32-S3 shall not brown out or reset.	Oscilloscope	Load the firmware that toggles the maximum LED set simultaneously. Monitor +3.3 V with an oscilloscope during switching and confirm processor operation continues.	+3.3 V stays at or above 3.20 V and no ESP32 reset or brownout occurs.

### Control Subsystem Verification Table

Requirement	Test Equipment	Verification Procedure	Pass Criterion / Recorded Result
The current amplifying MOSFETS sends 3.3V +/- 0.3V through to complete the circuit for our buzzer upon GPIO buzzer signal	Power supply, multimeter	Apply 3.3V through the buzzer and trigger the gate of the transistor to complete the circuit through drain and source. Measure current flow and verify correct voltage value across transistor.	Multimeter shows voltage drop of 3.3V across the buzzer, ensuring the device is no longer open circuited.
Digital logic transistors compute accurate truth tables upon their configuration	Power supply, multimeter,	Apply all valid input combinations to the logic inputs for a 2-input gate: 00, 01, 10, 11. For each case, measure the output node and compare the result to the expected truth table for the configured gate.	Pass if the measured output matches the intended truth table for all input combinations. For a 3.3 V logic system, a logic HIGH shall be $\geq 2.5$ V and a logic LOW shall be $\leq 0.5$ V.
The power MOSFET section will not have excessive voltage drop above 15% and not exceed a temperature of more than 70 degrees C.	Power supply, temperature sensor, multimeter	Operate the LED/buzzer load in the worst-case state for 1 minute. Measure the MOSFET case temperature at the end of the test.	Pass if the measured MOSFET temperature remains $\leq 70^{\circ}\text{C}$ after 1 minute of continuous operation.

### Processing Subsystem Verification Table

Requirement	Test Equipment	Verification Procedure	Pass Criterion / Recorded Result
The MCU recognizes the rotary encoder selection and updates both the display and the state machine accordingly	stopwatch and oscilloscope	Rotate the selector to each scenario position. Measure the time from selector movement to the correct scenario state appearing on the Display.	Pass if the displayed scenario and active state match the selector position for all scenarios, and

Requirement	Test Equipment	Verification Procedure	Pass Criterion / Recorded Result
			update time is $\leq$ 250 ms.
The counter starts upon MCU signal and stops upon the comparator trigger signal to measure RC/RL time constant within +/- 20% accuracy.	known capacitors/inductors, and oscilloscope	Insert known reference $C$ and $L$ values. Trigger the measurement routine. Compare the MCU measured value to the reference value. Repeat across the supported range.	Pass if computed $C$ and $L$ values are within +/- 20 % of the reference for all tested values. Record measured error.
The MCU recognizes the current state of the toggle switches on the board and updates its internal variables accordingly.	multimeter probe, stopwatch	Toggle each breaker/load/generator switch individually and verify that the displayed state output matches the physical switch position. Measure the update delay.	Pass if every switch state is read correctly and internal state updates within 100 ms.

**Status Devices Subsystem Verification Table**

Requirement	Test Equipment	Verification Procedure	Pass Criterion / Recorded Result
The display updates with grid information necessary to solve each scenario upon user inputted selection	Oscilloscope and stopwatch	Power the board and use the scenario selection input to cycle through each fault scenario. After each selection, observe the display and verify that the correct scenario name, fault information, and relevant values are shown. Measure the delay between user input and correct screen update.	Pass if the correct grid information for each selected scenario appears on the display within 250 ms of user input for all tested scenarios.
The scenario specific LEDs are red when the fault is active, blue if the fault is in a waiting state, and green when cleared.	programmed test scenarios, visual inspection	Run each scenario and place the system into the three required states: active fault, waiting/in-progress, and cleared. Observe the corresponding LED color for each state. Repeat for all applicable scenarios.	Pass if the LEDs display red during active fault, blue during waiting/in-progress, and green when the fault is cleared for all tested scenarios.

Requirement	Test Equipment	Verification Procedure	Pass Criterion / Recorded Result
The buzzer makes an audible sound upon a failure to clear the fault when the user hits the check status button.	audio observation	Start a fault scenario and intentionally clear it incorrectly. Press the check status button and observe whether the buzzer sounds. Repeat after correctly clearing the fault to confirm that the buzzer does not sound in the success case.	Pass if the buzzer sounds whenever the user checks status while the fault is cleared incorrectly, and remains silent when the fault is cleared correctly

### User Interface Subsystem Verification Table

Requirement	Test Equipment	Verification Procedure	Pass Criterion / Recorded Result
The breaker, load, and generator toggle switches both update the state machine variables as well as the on-board LEDs within 500ms.	stopwatch or oscilloscope, visual inspection	Toggle each breaker, load, and generator switch individually. Verify that the internal simulation state changes correctly and that the corresponding LED/output state updates. Measure the delay from switch movement to visible update.	Pass if all switches are correctly recognized and the LED/state update occurs within 500 ms for all tested inputs.
The RC/RL circuit correctly identifies the capacitor or inductor inserted by the user within 5 seconds.	known capacitors and inductors, stopwatch	Insert known test capacitors and inductors into the measurement socket. Trigger the measurement routine and observe the time required for the MCU to compute the measured component value. Compare the result to the reference value.	Pass if the system produces a valid component measurement within 5 s for all tested values.
The rotary selector shall allow the user to select one of the fault scenarios, and the selected scenario shall be recognized by the MCU and shown on the display within 500 ms.	stopwatch or oscilloscope, visual inspection	Rotate the selector to each fault scenario position. Verify that the display updates to the correct scenario and that the MCU enters the corresponding state. Measure the time between selector movement and display update.	Pass if each selector position correctly maps to the intended scenario and the display updates within 500 ms.

## Appendix B Requirement and Verification Results Tables

### Power Subsystem Verification Results Table

Requirement	Verification Procedure	Measured Result	Status
Board shall operate from the battery input.	Powered the board from the battery input and observed operation.	All active subsystems operated correctly.	Pass
5 V rail shall remain between 4.5 V and 5.5 V.	Measured the 5 V rail with a digital multimeter.	5.03 V	Pass
3.3 V rail shall remain between 3.10 V and 3.50 V.	Measured the 3.3 V rail with a digital multimeter.	3.29 V	Pass
ESP32-S3 shall not brown out or reset.	Ran the system through the scenario sequence.	No reset or brownout occurred.	Pass

### Processing and User Interface Verification Results Table

Requirement	Verification Procedure	Measured Result	Status
User shall select all five scenarios.	Selected each scenario and observed the response.	5 out of 5 scenarios selected correctly.	Pass
ESP32-S3 shall recognize user inputs.	Toggled breaker, load, and generator switches.	All tested inputs were recognized.	Pass
State machine shall check user responses.	Tested correct and incorrect responses.	Correct responses passed; incorrect responses triggered failure feedback.	Pass
User input shall update within required time.	Observed delay between input and system response.	Response occurred within the timing limit.	Pass

### Status Devices Verification Results Table

Requirement	Verification Procedure	Measured Result	Status
Display shall show scenario information.	Checked display output for all five scenarios.	Display worked for 5 out of 5 scenarios.	Pass
LEDs shall indicate grid or fault state.	Observed LED behavior during tested states.	LEDs responded correctly.	Pass
Buzzer shall sound for incorrect actions.	Entered incorrect responses.	Buzzer activated correctly.	Pass
Buzzer shall stay off for correct actions.	Entered correct responses.	Buzzer remained inactive.	Pass

### Component Identification Verification Results Table

Component Type	Reference Value	Acceptable 15% Range	Measured Result	Percent Error	Status
Capacitor	10 $\mu$ F	8.50 $\mu$ F to 11.50 $\mu$ F	9.72 $\mu$ F	2.8%	Pass
Capacitor	33 $\mu$ F	28.05 $\mu$ F to 37.95 $\mu$ F	34.1 $\mu$ F	3.3%	Pass
Capacitor	47 $\mu$ F	39.95 $\mu$ F to 54.05 $\mu$ F	45.8 $\mu$ F	2.6%	Pass
Inductor	15 mH	12.75 mH to 17.25 mH	14.2 mH	5.3%	Pass
Inductor	10 mH	8.50 mH to 11.50 mH	10.7 mH	7.0%	Pass
Inductor	5 mH	4.25 mH to 5.75 mH	4.81 mH	3.8%	Pass

# PCB KiCAD Image

