ECE 445

SENIOR DESIGN LABORATORY

Design Document

ClassroomClarity

A Portable Teacher Support Hub

Team No. 21

MADDIE DONKU
(mdonku2@illinois.edu)

KAITLIN GOWENS
(kgowens2@illinois.edu)

JESSE GRUBER
(jgruber5@illinois.edu)

TA: Aishee Mondal
Professor: Michael Oelze

March 6, 2025

Abstract

This document provides a detailed description of our project, ClassroomClarity, and the designs currently being implemented and tested. In this document, we breakdown the physical and circuit design, the estimated cost and timeline, as well as the ethics and safety of our project.

Contents

1 Introduction

The following sections provide an overview of the problem we are focusing on, as well as our solution to said problem.

1.1 Problem

Within the classroom there are two sides to learning: students and teachers. For students, it is critical that they can ask questions in order to fully participate in the learning process. However, many students are too anxious to raise their hands or when they do the teacher is so engrossed in their lecture that they go unnoticed. For teachers, they not only have to stay aware of the classroom while teaching the material but also gauge student understanding to lead a successful lesson.

While there are online tools such as Mentimeter, these platforms require professors to use time outside of class to create slides and also take up screen space on the lectern. Another issue with the variety of sites used for student engagement is that there is no uniformity for the students. Cell phones and laptops can become clogged with numerous bookmarks for these applications for different classes. Lastly, professors may need an easily detectable, portable, physical alert to remind them to look at questions that students have posted, which cannot be provided by online means.

Professors and students can benefit from a tool that will easily show them how the class is handling material and any questions that may arise. A hub that is consistent between classes will simplify the learning experience for both students and professors.

1.2 Solution

Our solution is a physical hub for the teacher and a phone app for the students that works together to bring clarity into the classroom. The hub includes indicator lights that will visually relay how the students are reporting their understanding of the class material through the app. The app will also provide options to the students on how they would like to submit a question: The "raise-hand" feature, which will notify a professor when a student wants to vocally ask their question, or a text submission which will display on the hub's screen. Any submitted questions will be added to a queue which the teacher can scroll through and clear answered questions using a dial and a button, respectively. In addition, the teacher has the option for the hub to vibrate when a question is asked to remind the teacher to look at the hub. There is also an option to put the hub on silent mode where instead of vibrating, another LED will light up when a question is asked. The hub will also have a specific passcode that must be entered into the app to control who has access to the hub.

1.3 Visual Aid

Figure 1 shows an in-context diagram of the ClassroomClarity hub system. The central hub connects via Bluetooth to the phone application. The app collects data on student understanding and engagement by allowing students to select their level of understanding. It also provides a space to submit questions. This data is sent to the hub which displays the first question in the queue and reflects the average engagement level with the LED display. The hub, when it receives question data, will either vibrate or light an LED depending on the selected mode. The central hub also has a button and dial which allows the teacher to scroll through the question queue and clear answered questions.

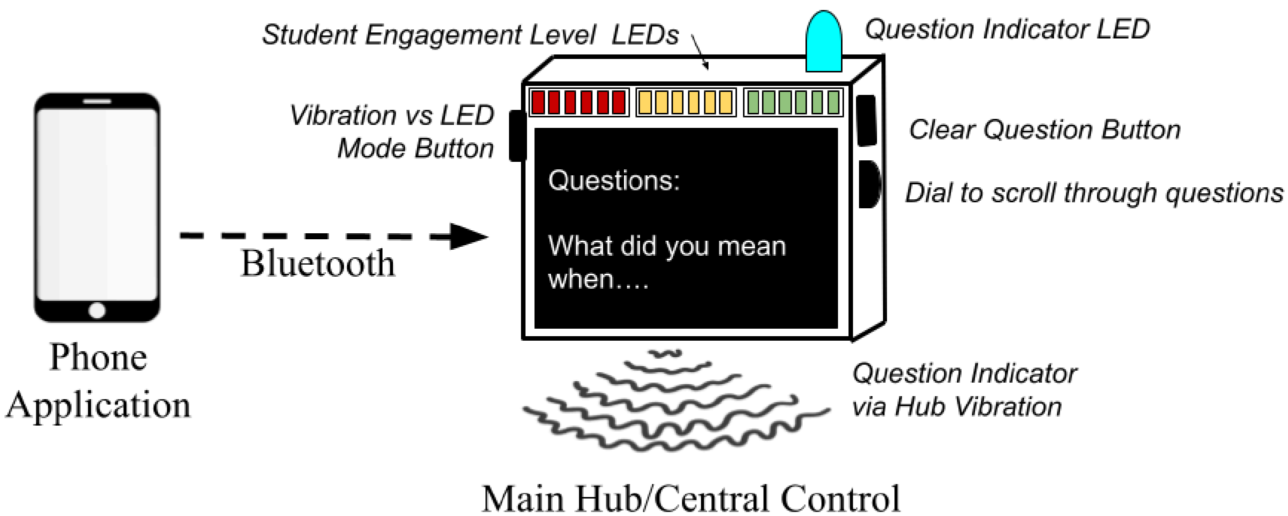

Figure 1: High-level diagram of the ClassroomClarity design

1.4 High-Level Requirements

In order to deem the project successful, we want to reach the following goals.

1. Questions can be sent from the phone app to the hub question queue, which should be able to hold at least 5 questions, such that at least 1 question should be displayed by the central hub if the queue is not empty.

2. When a question is received by the hub, the hub should vibrate for 1 second and light the indicator LED within 30 seconds of receiving it. LEDs should turn off within 30 seconds of the question queue becoming empty.

3. Students can select their understanding level on the app and the LEDs on the hub should change within 30 seconds of the selection to match the new average level of understanding.

2 Design

2.1 Block Diagram

Figure 2 displays the various subsystems necessary for the creation of ClassroomClarity. These include the power management, control, and feedback subsystems along with the mobile application. The power management subsystem is responsible for converting AC power from an outlet to 5 VDC and 3.3 VDC to be used throughout the circuit. The control subsystem consists of the various devices that produce signals to coordinate the hub's operation. Next, the feedback system involves all the methods of displaying data and signals to the professor. Finally, the mobile application is how students can interact with the main hub through question submission and understanding rating, as well as how professors can change certain settings.

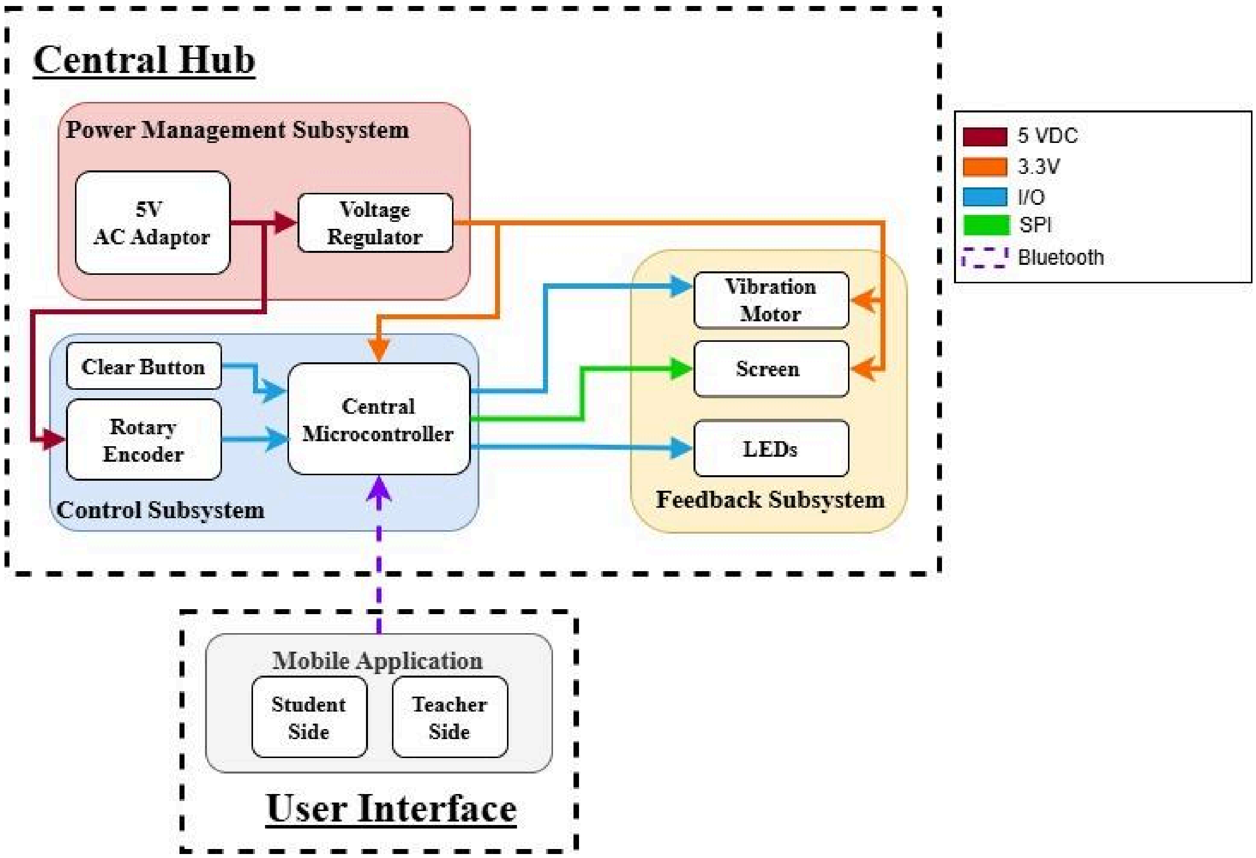

Figure 2: Block Diagram of ClassroomClarity System

2.2 Physical Design

In order to maintain simplicity and portability, the base of the physical design is a box. The box holds the circuitry inside, only showing the screen display, button, LEDs and rotary encoder for the instructor to interact with. Attached to the box is a flexible cover that will provide protection to the screen during transport as well as act as a stand during use. The flexibility of the cover allows the instructor to position the device at various angles so they can choose what best suits their needs. Figure 3 contains a visual of the physical design while Figure 4 provides the key dimensions of the casing. Finally, Figure 5 gives a peek inside the casing to demonstrate how the PCB and screen will be attached to the casing walls. The screen will slide onto mounts that stick off the wall and be secured using a nut. The PCB will be screwed on top of mounts that stick off the lid to provide both ventilation and easy access.

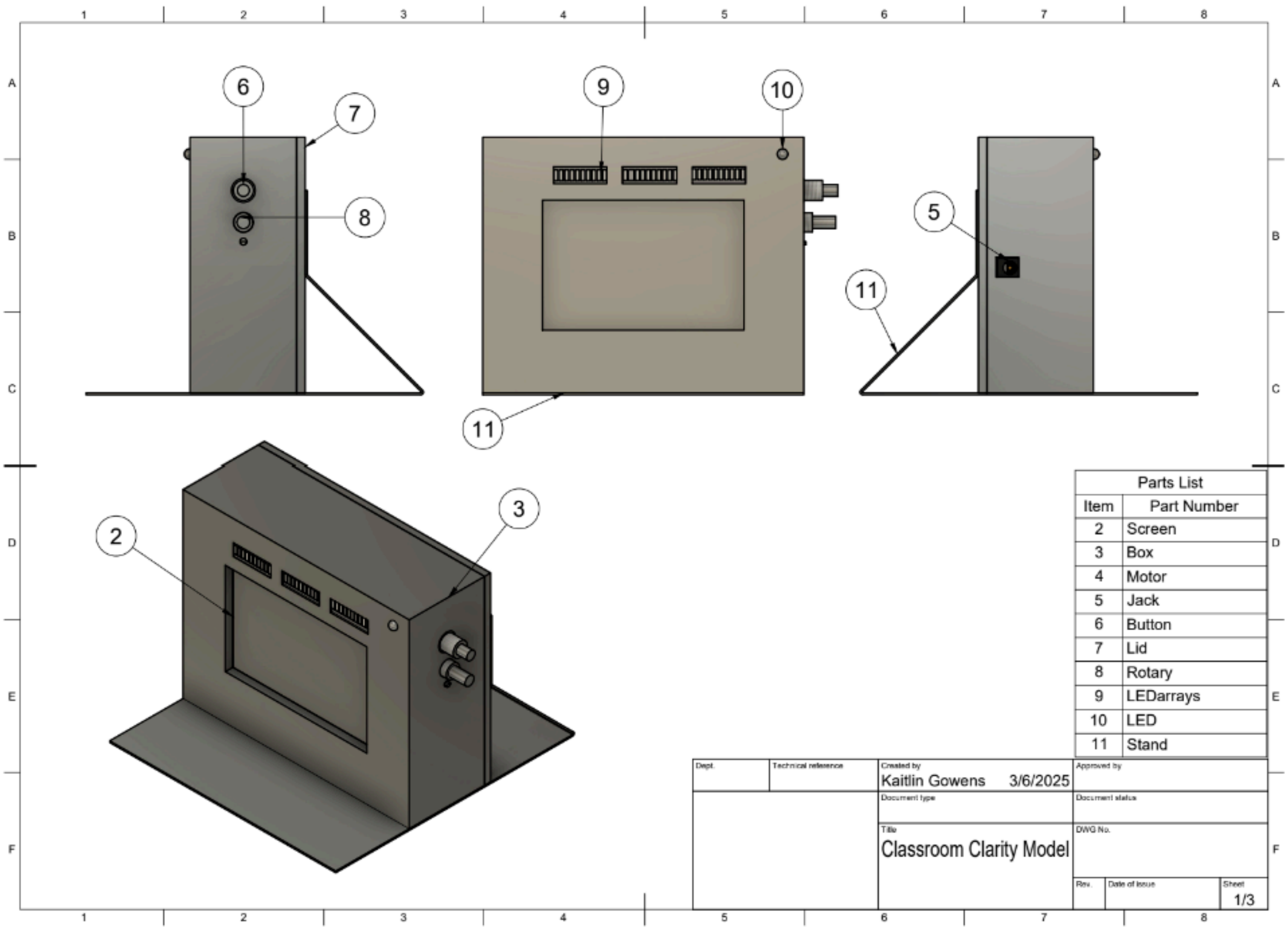

Figure 3: 3D Cad Model of ClassroomClarity casing

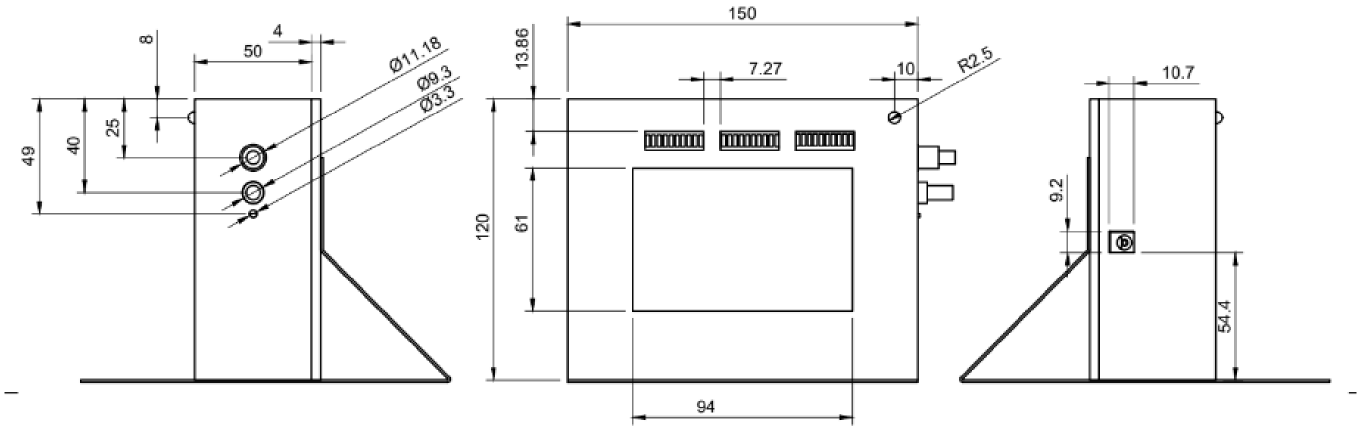

Figure 4: Key dimensions of the ClassroomClairy casing

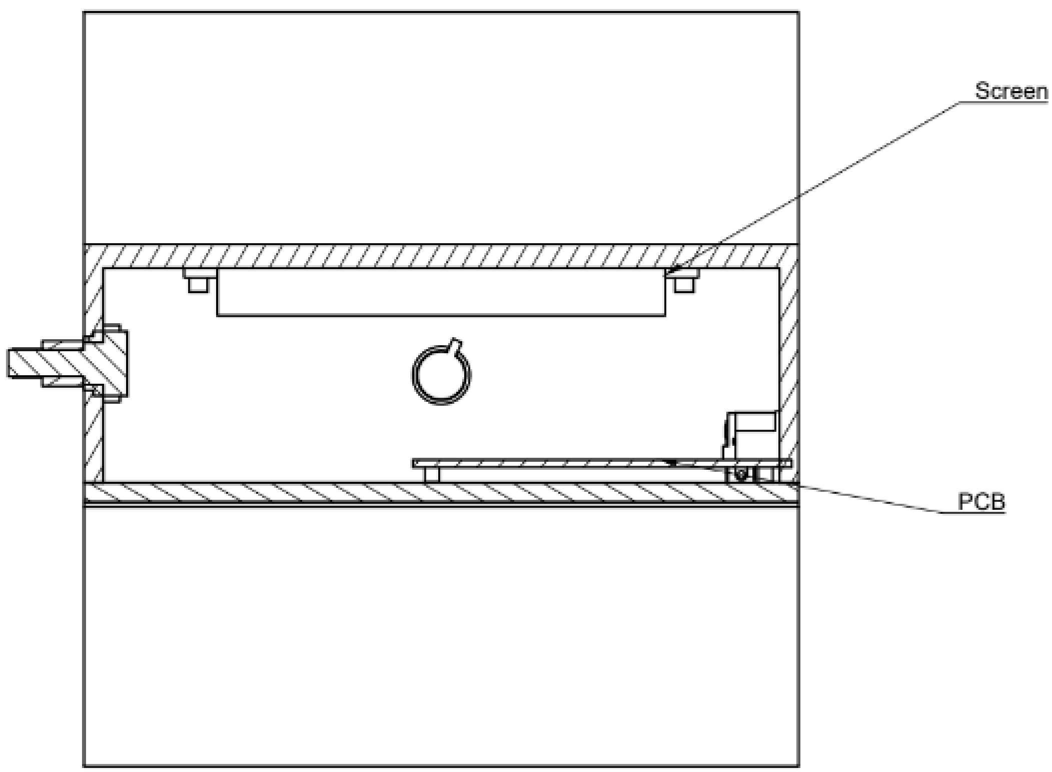

Figure 5: Top view of the ClassroomClarity casing with the top removed to reveal the PCB and screen attachment system

2.3 Functional Overview and Requirements

The following section will provide a description of each subsystem, detailing its specific purpose and functionality.

2.3.1 Power Management Subsystem

The power management subsystem, as seen in Figure 6, provides power to all components of the hub and is essential for the device's functionality. Without proper power delivery from this subsystem, the hub itself would function unreliably or not at all. The subsystem consists of a AC/DC wall adapter (WSU050-1500-R) and a DC/DC converter module (TPSM84203EAB). These components were chosen to match the needs of the ESP32-WROOM microcontroller as well as the rest of component ratings.The adapter converts 120V AC from a wall outlet and provides 5V, 1.5A to the DC/DC converter module. The converter will then provide 3.3V to the hub's control and feedback subsystems. All components have been selected to operate with the 3.3V from the converter module except for the rotary encoder, which will be supplied directly with 5V volts from the wall adapter. Calculations to ensure adequate power delivery to the entire system are provided in the Tolerance Analysis section of this document. Table 1 enumerates the specific requirements of this subsystem and the methods to verify their functionality.

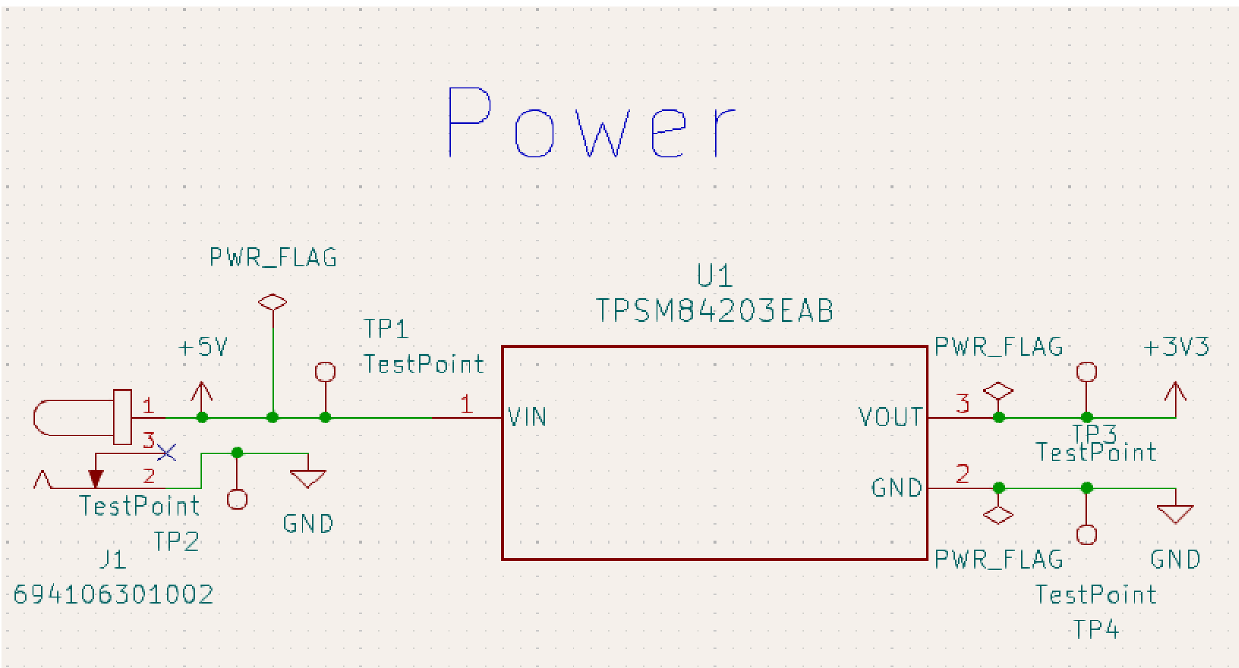

Figure 6: KiCad schematic of power management subsystem

Table 1: Requirements and verifications of the power management subsystem

Requirements	Verifications
1. The AC DC wall adapter must supply at least 1.5A to the hub and 5V +/- 0.5V to the DC DC converter module.	• Ensure adequate connection of adapter to barrel jack • Probe 5V and ground test points corresponding to adapter output with multimeter • Confirm that measured voltage is within 4.5-5.5V

2. The adapter must be able to safely provide a stable supply of power at the very least the length of one lecture (50 minutes).	● Plug in clarity hub to wall adapter and ensure adequate connection ● Probe 5V and ground test points corresponding to adapter output with oscilloscope ● Observe voltage and current signals from oscilloscope for the duration of a lecture period ● Use infrared thermometer to measure adapter temperature at both the wall outlet and barrel jack connections and ensure temperature is below 43°C (limit for safe human touch) ● Confirm that there are no voltage or current discrepancies during testing period
3. The DC DC converter module must take the input from the AC DC adapter and output 3.3V +/- 0.3V.	● Ensure adequate connection of adapter to barrel jack ● Probe 3V3 and ground test points corresponding to converter output with multimeter ● Confirm that measured voltage is within 3.0-3.6V

2.3.2 Control Subsystem

The control subsystem, illustrated below in Figure 7, is responsible for receiving signals via Bluetooth from the application, lighting LEDs, displaying questions to an LCD, and processing physical inputs. Because of its plethora of GPIOs, the ESP32-S3-WROOM-1 was chosen to implement our design. The GPIO pins output about 2.64V and 40mA. These will be used to light the LEDs and turn on the vibration motor. Also, the Bluetooth signals received from the application will be sent to the LCD via the SPI pins on the microcontroller.

The control system houses 3 buttons: clear (40-2388-01), chip enable (PTS645SL43-2 LFS), and program (PTS645SL43-2 LFS). The clear button will allow a professor to mark a question answered and remove it. Next, the chip enable button will turn off the chip whenever it is pressed and leave the chip on when unpressed. Finally, the program button will initiate programming when it is pressed with the chip enable button. Internal pull-up resistors of 45kΩ are utilized to avoid a floating input on the clear button.

Additionally, the microcontroller will receive inputs from a rotary encoder (PEC16-4120F-N0012-ND). The rotary encoder will have a debouncing/pull-up circuit before its signals are passed to the microcontroller. It features a quadrature output, meaning the A and B outputs are square waves shifted by 90 degrees. The microcontroller will have interrupts on both A and B and will compare each read state to the last. Comparing how the states of A and B change determines the rotation of the encoder, allowing a professor to scroll through questions. Figures 8-10 show the circuit schematics necessary for the controls to function.

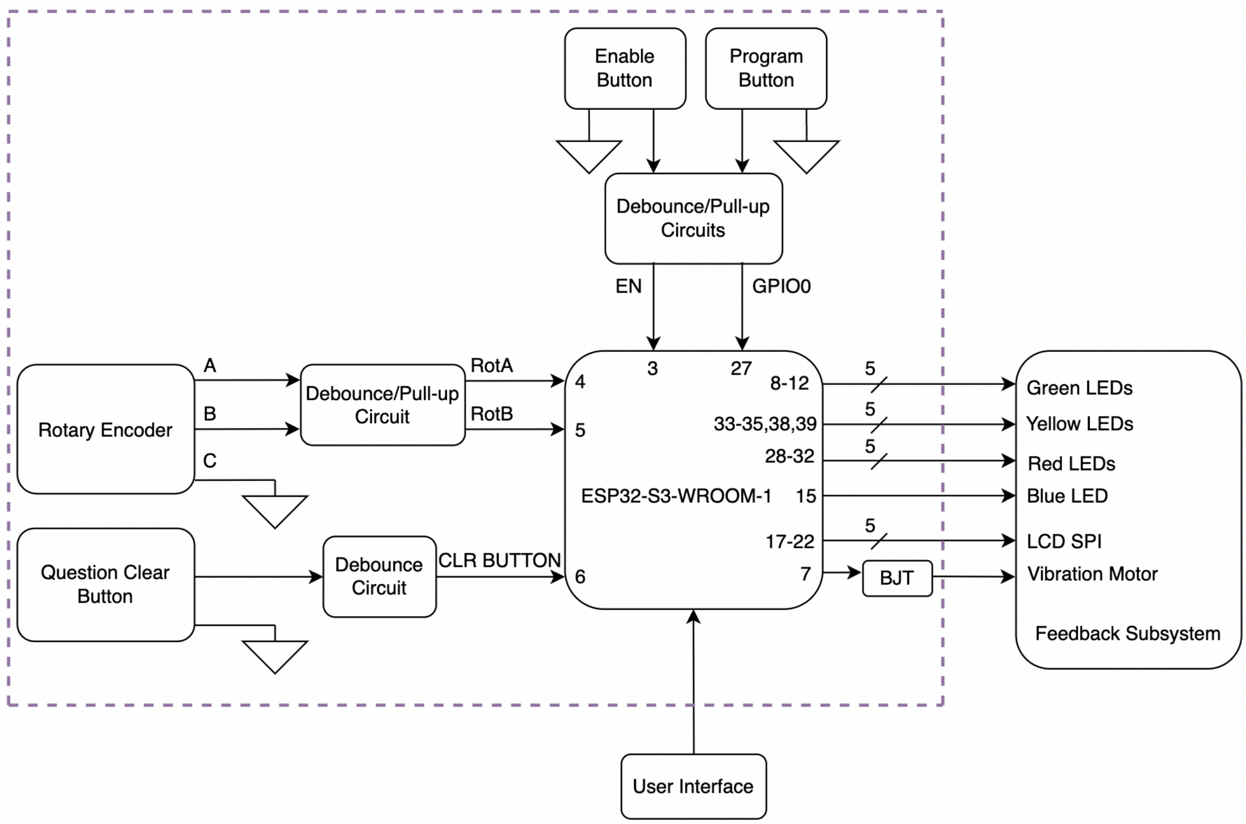

Figure 7: Block diagram of control subsystem

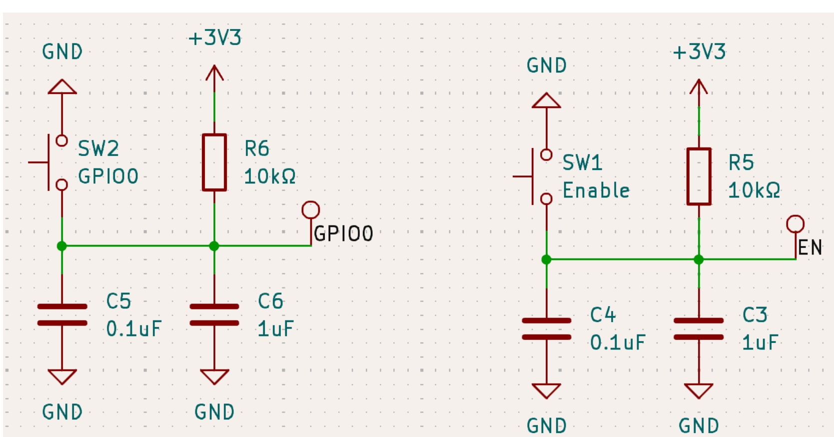

Figure 8: Enable and program button debounce\pull-up KiCad schematic

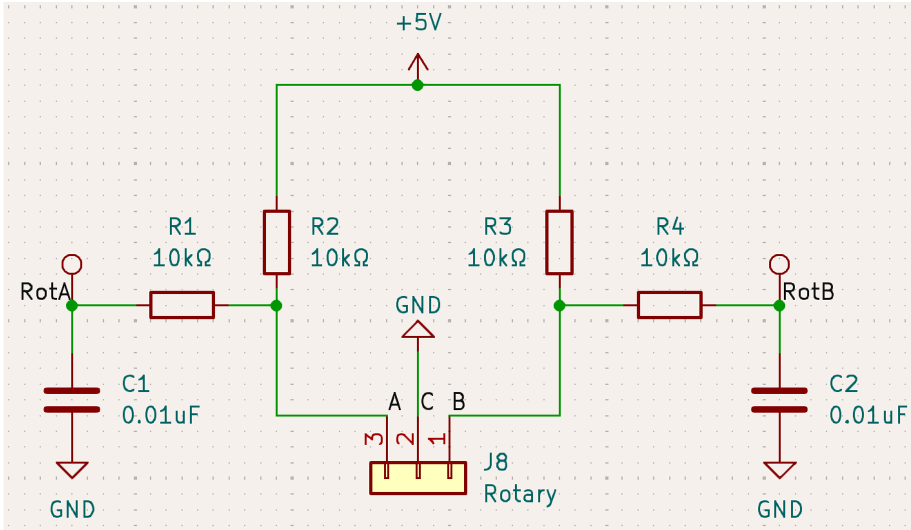

Figure 9: Rotary encoder debounce/pull-up KiCad schematic

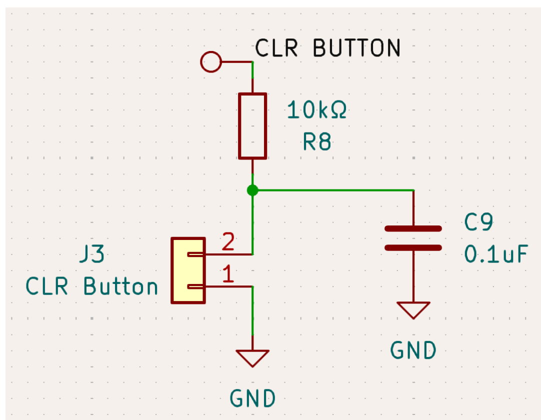

Figure 10: Clear button debounce KiCad schematic

Table 2: Requirements and verifications of the control subsystem

Requirements	Verifications
1. The unpressed input voltage of each button GPIO must fall between 2.5-3.6V, and the pressed input voltage must be between -0.3-0.8V within one second of pressing the button.	• Probe test points of each button using a multimeter • Confirm the "unpressed" voltage is between 2.5-3.6V • Confirm the "pressed" voltage is between -0.3-0.8V within a second of pressing the button
2. The rotary encoder processed signals must have a maximum between 2.5-3.6V and a minimum between -0.3-0.8V.	• Probe test points of processed signals RotA and RotB using an oscilloscope • Begin turning the encoder until voltage is high on RotA and RotB • Confirm the maximum voltage is between

	2.5-3.6V and the minimum voltage is between -0.3-0.8V
3. The high output voltage of the green, yellow, and red LED GPIO pins must be greater than 2.6V and the low output voltage must be less than 1.5V.	● Probe the headers of the LEDs using a multimeter ● In the microcontroller code, set the LED GPIO pins high ● Upload the code to the microcontroller ● Confirm the header voltage is greater than 2.6V ● In the microcontroller code, set the LED GPIO pins low ● Upload the code to the microcontroller ● Confirm the header voltage is less than 1.5V
4. The high output voltage of the vibration motor GPIO pin must be greater than 2V and the low output voltage must be less than 1V.	● Probe the test point of the vibration motor control signal ● In the microcontroller code, set the vibration motor GPIO pin high ● Upload the code to the microcontroller ● Confirm the voltage is greater than 2V ● In the microcontroller code, set the vibration motor GPIO pin low ● Upload the code to the microcontroller ● Confirm the voltage is less than 1V in the microcontroller code

2.3.3 Feedback Subsystem

The feedback subsystem is the method in which a professor will interact with the information students input to the application. This subsystem consists of three arrays of 10 LEDs in green, yellow, and red to gauge student understanding, a single LED to indicate a question, a vibration motor to alert and remind a professor that there is a question, and an LCD screen to present the question to a professor.

First, the arrays of LEDs will illustrate the average level of understanding based on the information sent in by the application. The green array (LTA-1000G) represents general understanding, the red array (DC10EWA) represents a low level of understanding, and the yellow (DC10YWA) array is in between. One GPIO pin will drive two LEDs, allowing 5 pairs of LEDs to be lit to represent the resulting average of class understanding per understanding level. In addition to the arrays of LEDs, a single LED will be lit whenever there is a question transmitted to the hub. These LEDs must respond to any relevant changes from the application within 30 seconds of data being sent by the application.

Additionally, a vibration motor (316040004) will act as a reminder to a professor that there is a question on the hub. The vibration motor must vibrate within 30 seconds of a question being posted to the hub. Furthermore, as long as there is an unanswered question posted, it must vibrate every 2 minutes as a reminder. Lastly, an LCD (MSP4002) will display the questions posted by the students.

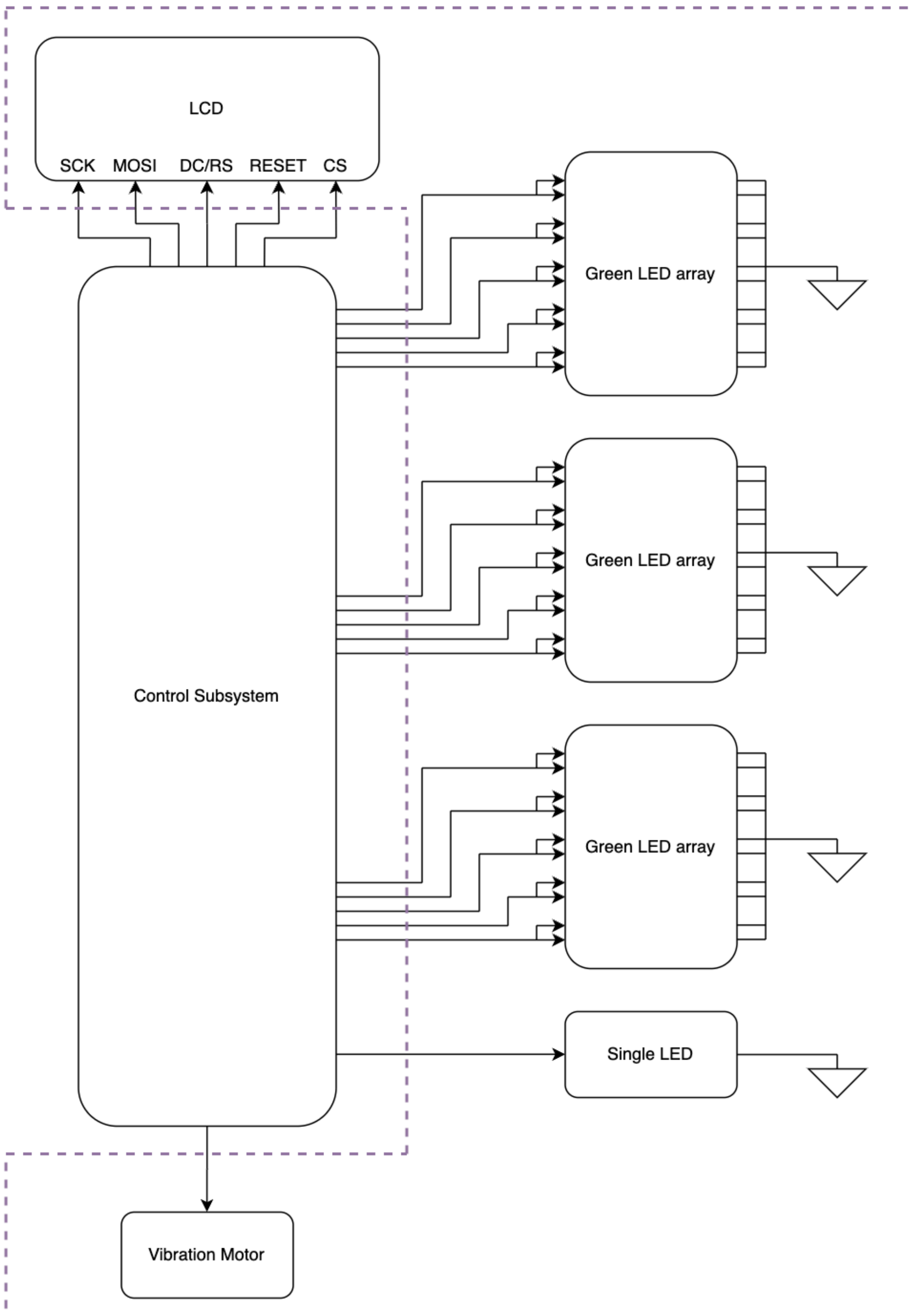

Figure 11: Block diagram of the feedback subsystem

Table 3: Requirements and verifications of the feedback subsystem

Requirements	Verifications
1. LEDs should react to any changes in "Understanding Rating" data sent by the app within 30 seconds.	● In the microcontroller code, add a "t_start" variable right after the get data function. Add a "t_end" variable after the change to the "Understanding Rating" LEDs GPIO signal. Both of these variables should store the time at occurrence. ● At the end of the code, add a line to print out t_end - t_start to the serial display. ● Upload the code to the microcontroller and restart the device and reconnect the app. ● Select a new understanding rating button. The hub's LED should update to reflect the new value within 30 seconds.
2. Signal A of the rotary encoder must lead signal B when turning the encoder clockwise. Signal B must lead signal A when turning counterclockwise.	● Probe test points of processed signals RotA and RotB using an oscilloscope triggered on a rising edge ● Turn the encoder clockwise and confirm signal A's waveform leads signal B's ● Turn the encoder counterclockwise and confirm signal B's waveform leads signal B's
3. The LCD should display a maximum of 200 characters with a black background and white font	● In the microcontroller code, utilize the TFT_eSPI library included in the Arduino IDE to run a provided demo ● Upload the code to the microcontroller ● Confirm the demo's operation on the screen
4. The vibration motor should vibrate within 30 seconds of a question being sent from the app	● In the microcontroller code, add a "t_start" variable right after the get data function. Add a "t_end" variable after the change to the vibration motor GPIO signal. Both of these variables should store the time at occurrence. ● At the end of the code, add a line to print out t_end - t_start to the serial display.

	● Upload the code to the microcontroller and restart the device and reconnect the app. ● Send a question to the system. The vibration motor must begin vibrating within 30 seconds.

2.3.4 Mobile App User Interface

The mobile app is used as a user interface for both the students and the instructor. Overall there are 5 possible screens a user can encounter: A welcome screen, two login screens depending on choice of student or instructor, and two functionality screens also depending on the choice of student or instructor. Figure 12 demonstrates the flow between the various screens. The welcome screen consists of two buttons that allow the user to choose whether they would like to login as a student or a teacher. If “Student” is chosen, the user is prompted to enter their display name as well as the student password their instructor will provide them with. If “Instructor’ is chosen, the user is prompted to input the separate instructor specific password. When the correct password is entered, teachers will be brought to a page that acts as the settings control for the central hub. Here they can change the font size and turn on silent mode which removes vibration notifications. On the student side, students can select their level of understanding by pressing one of the buttons on a 1-10 scale. Students can also submit questions in one of two ways. Firstly, students can type out their question in the provided text field and press the submit button which will send a copy of their response to the central hub for the teacher to read. Secondly, they can press the “Raise Hand” button which will instead submit their display name to the center for the teacher to see, allowing the professor to then call on them in-person.

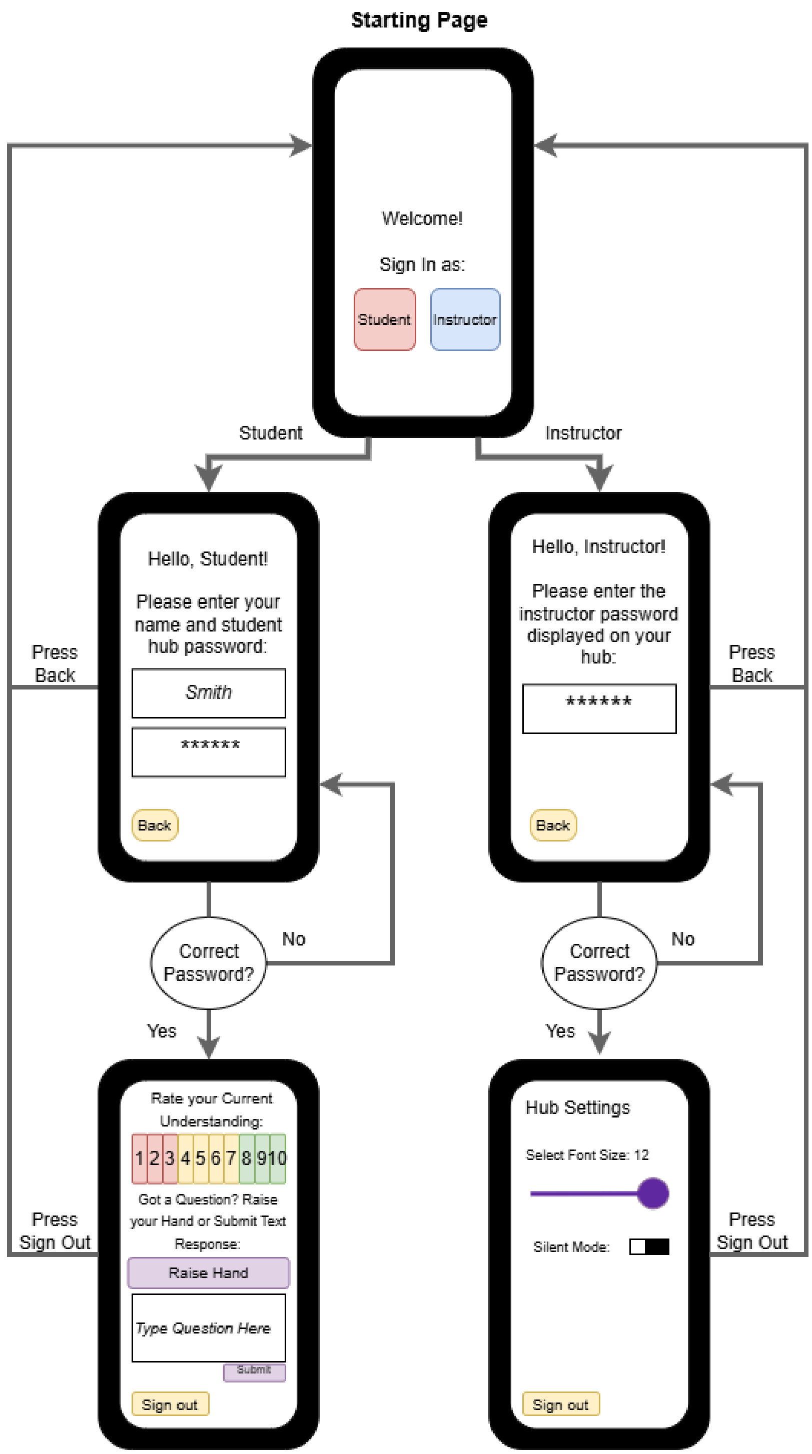

Figure 12: Chart depicting the flow between the various screens a user can interact with

Android Studio was chosen to develop the mobile app using the Flutter UI framework. Flutter allows for straightforward creation of various UI components such as buttons, switches, sliders, and text fields which serves the needs of the app well. It also supports the performance of standard smartphones as it runs at 60 FPS or 120 FPS depending on the refresh rate of the individual devices. The Android Studio and Flutter combination also allows for simultaneous development of IOS and Android applications.

In order to send the data collected by the app to the central hub, a Bluetooth connection between the user's phone and the central hub is formed. The ESP32 microcontroller has its own built-in Bluetooth module that supports BLE 5.0 which is compatible with most modern Android and Apple devices. It transmits 1Mbps at a frequency range of 2402-2480 MHz [3]. To avoid needing to connect to the hub's Bluetooth via the mobile device's settings, the flutter_blue dart package will allow the device to scan for nearby BLE devices and connect directly from the app [2]. This package also is what prepares the data packages for sending and receiving. More information on specific Bluetooth communication controls will be discussed in section 2.4.1.

Table 4 lists the requirements necessary to deem the mobile app successful in the scope of the project, as well as the tests used to verify it meets the requirements.

Table 4: Requirements and verifications table for the mobile app user interface

Requirements	Verifications
1. The app must bluetooth connect to ESP32 within 10 seconds of initiating a connection request	● Within the code, place a variable, "t_start", before at the beginning of the connection protocol and other variable, "t_end", after connection process completes that stores the current time ● At the end of the code, add a line to print out t_end - t_start to the serial display ● Upload the code to the microcontroller and restart the device ● On the app, try to connect to the microcontroller. ● Once it finishes connecting, read the serial display and confirm the time was <10s.
2. The app should be able to send and receive data with a less than 1% error rate per transmission.	● In the microcontroller software, add serial printouts that print out the received data. ● Upload the new code to the microcontroller and connect the mobile app to the microcontroller Bluetooth. ● On the app, select a different "Understanding Rating" button ● Check serial display to confirm the

	received understanding rating matches ● On the app, press the "Raise Hand" button ● Check serial display to confirm the received display name matches ● On the app, submit a text question ● Check serial display to confirm the received question matches
3. The app should respond to any user interactions within 1 second.	● Restart the app ● Select a new "understanding rating" button ● Confirm the app reflects the change within 1 second of pressing the button.
4. The app should hold any data ready to be transmitted until the central hub's signals it is ready to accept new data.	● Connect 2 devices to the hub ● Send in two different text questions at the same time. ● Use the rotary encoder to scroll through the questions queue and confirm both questions were received.

2.4 Software Design

The following section goes into detail of the design considerations for implementing signal control and bluetooth communication within the microcontroller. The code is written in Arduino IDE which provides a vast amount of feature libraries and has a built-in compiler for easy upload to the microcontroller.

2.4.1 Bluetooth Communication

The ESP32 has built-in Bluetooth Low Energy (BLE) which is ideal for connecting to smart devices and is power efficient. BLE also works at longer ranges, approximately 100 meters, compared to classic Bluetooth which is only about 10 meters. Conveniently, the Arduino IDE contains various BLE libraries that work well with the ESP32 microcontroller. The BLE libraries that will be used in this project are BLEDevice, BLEUtils, and BLEServer [11]. These libraries are what allow the ESP32 to be set up as a BLE server, handle UUIDs and data, as well as interact with the mobile app clients. In order to prepare the ESP32 to act as a BLE server, there are a series of steps that must be executed, shown in Figure 13.

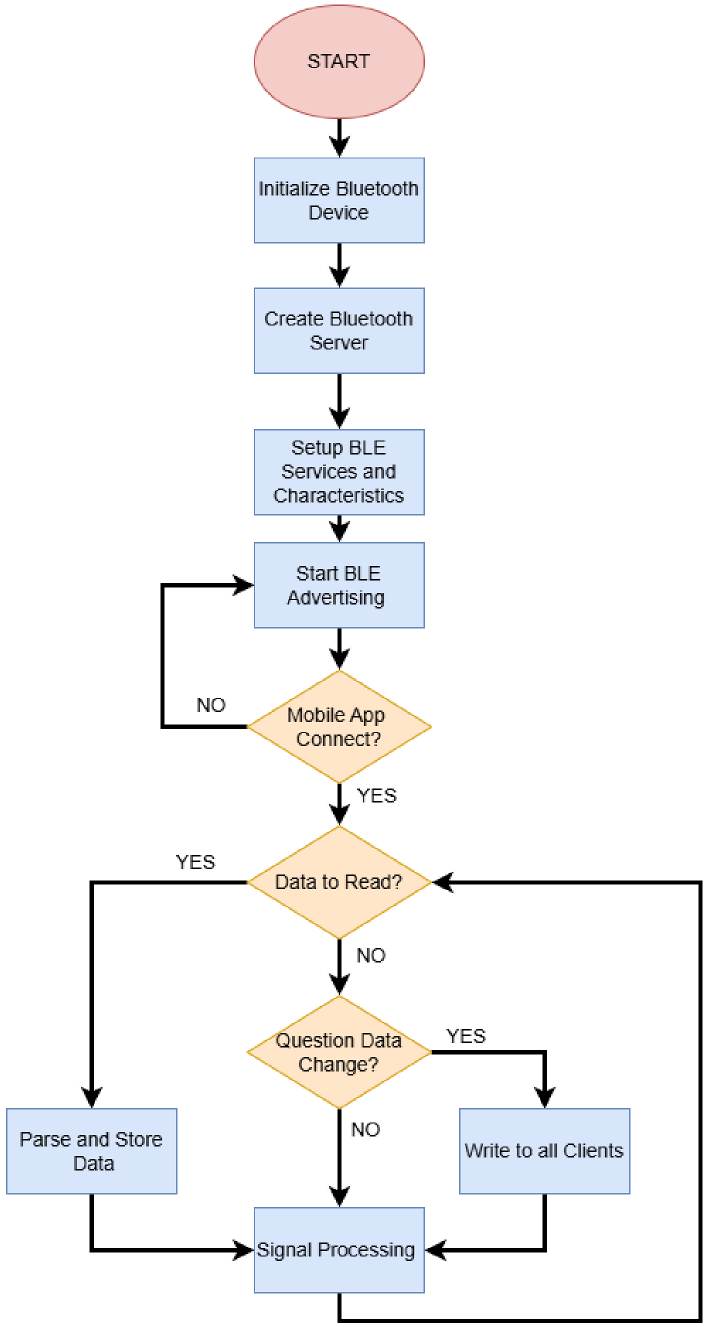

Figure 13: Flowchart depicting the setup and data management of a BLE device

First, there is the setup that creates the bluetooth device and server as well as define the types of transmitted data, aka characteristics. This project aims to both read data coming from the app and write data to the app, therefore it is important to specify more than one characteristic. There are three different types of data transfer in the scope of the project: Integer data for the understanding rate from the app, string data for the questions submitted from the app, and string list that compiles all the asked questions. In order to efficiently handle the various expected data, each of these should be their own characteristic with a unique UUID and read/write property.

Once this is setup the ESP32 is prompted to begin announcing it is ready for a client device, aka the smartphone with the mobile app, to connect. In order to finish the connection, the user will be prompted to input the passkey in the app and only once that matches will the mobile connect to the Bluetooth. Finally, the ESP32 checks for any new data input or output. If a new question has been added to the question queue, then the ESP32 will notify all the connected clients that there is new data so the apps will add the new question to their lists. If the app has sent any new data for the understanding ratings or a new question, the ESP32 will process the incoming data and use it in the signal processing updates. The signal process code will be discussed more in the following section. This checking for data updates is repeated until the hub is turned off.

2.4.2 Signal Control

The ESP32 microcontroller is in charge of sending and accepting the correct signals for proper functionality of the LEDs, vibration motor, rotary encoder, button, and screen display. Figure 14 shows an overview of the logic flow of the software and this section will dive into a more detailed description of how the code deals with each function.

Understanding Rating: As described in the feedback subsystem, the LED arrays are used to display the average level of understanding. When student users change their level of understanding the app sends both the new and old rating to the hub. The old rating is necessary in order to calculate the average rating value shown in Equation 1.

$$prevAvg - prevRating + newRating = currAvg \quad (1)$$

Once the microcontroller detects a change in the rating, it updates the amount of LEDs lit up by changing the GPIO signals to HIGH or LOW. With 15 LEDs available to show the rank level, the amount of GPIO signally HIGH can be calculated using Equation 2.

$$\frac{15\ LEDs}{10\ levels}(currAvg) = \#\ GPIOs\ HIGH \quad (2)$$

Notification System: The notification system is a combination of a vibration motor and a LED. The instructor can set in their app whether or not the hub will be in silent mode (no vibration) and the app will send that data to the microcontroller. Within the software, there is a list variable that holds all the questions that have yet to be answered. So long as this list has a length greater than zero, the LED's GPIO will be HIGH, therefore lighting the LED. If the hub is not in silent mode, then a timestamp of the last time the vibration motor went off will be saved. If the current time gets to be more than 2 minutes, the motor's GPIO will be brought HIGH for ~1 second before going LOW again. Then, the timestamp will be updated.

Question Display: The question display control deals with interpreting the signals for the clear button, the rotary encoder, and the screen. When the list holding the unanswered questions has a length greater than 0, the screen displays the question at index 0. This can be done using the TFT_eSPI library included in the Arduino IDE which after setting the correct SPI pin connections of the ESP32, allows for easy printing, rotating, and sizing of the display. When the microcontroller detects a change in signal at the rotary encoder GPIO, then the screen is reset to now display the question at the next index. If there is no following index, the index loops back around to the zeroth position. The button uses an internal pullup resistor so that its GPIO reads LOW when it is pressed. When the microcontroller detects the button state change from HIGH to LOW, it removes the question that is at the current display index and displays the question at the following index. Once again if there is no following index, it loops back to the zeroth position. If the list becomes empty, then no question is displayed. Another software feature to mention is the debounce of the button. Once a button state change is detected, there is a 200ms delay that stops a single button press from being considered multiple.

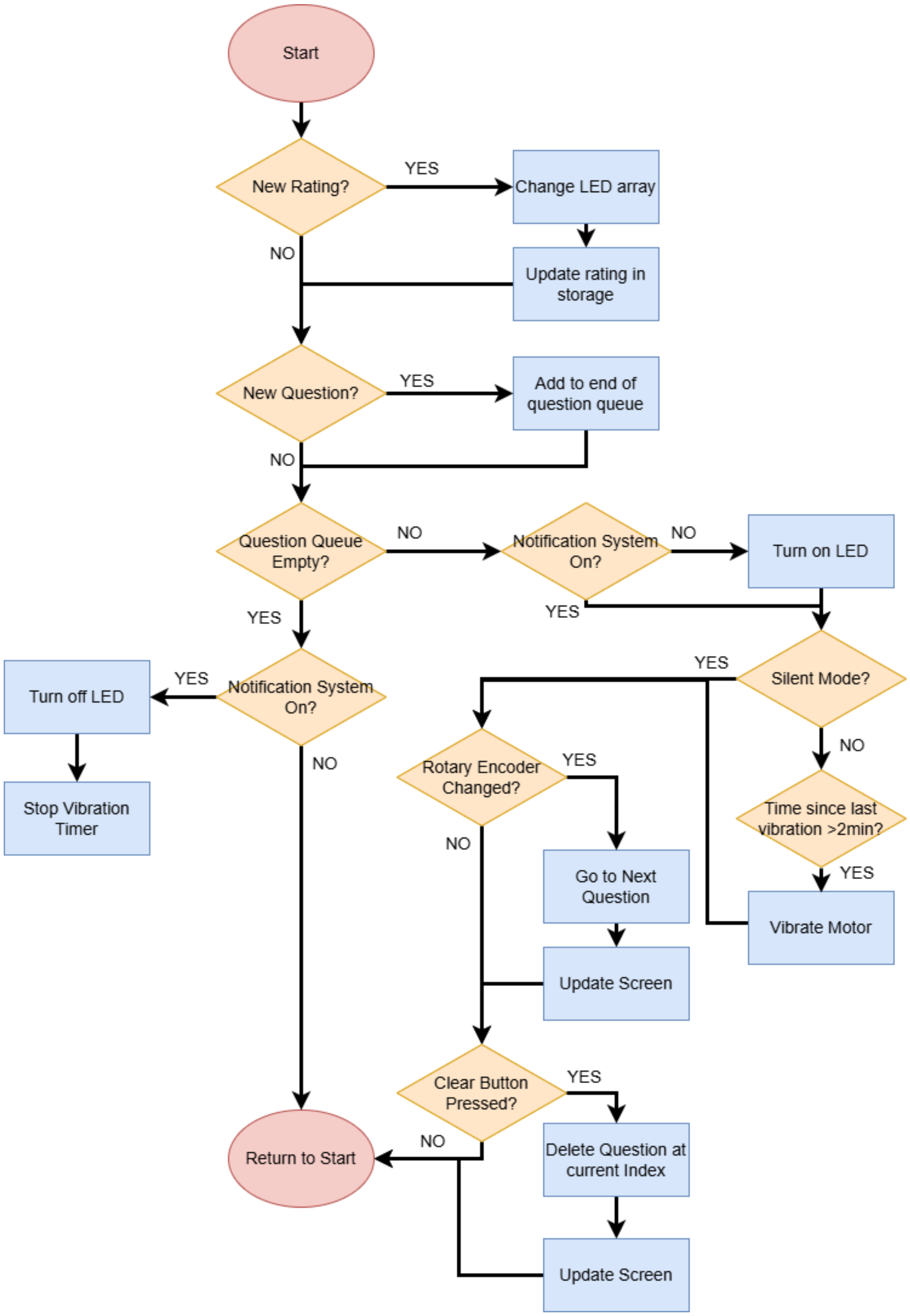

Figure 14: Flowchart depicting the control logic of the microcontroller for LED, vibration motor, rotary encoder, button, and screen functionality

2.5 Tolerance Analysis

The biggest failure point for this project is if power is not properly delivered to the different components of the hub. Failure can occur if incorrect power conversions are made, leading to either too much or too little power to the components. This can cause the circuit to simply not function, or it can lead to much larger issues due to overheated components or blown capacitors. Careful calculations are required to find suitable conversion rates based on the peak voltage ratings of the components.

Table 5: Voltage requirements for Hub components

Component	Peak Voltage	Peak Current
Microcontroller (ESP32-S3-WROOM-1) [3]	3.3V	500mA
TFT LCD (MSP4022) [4]	5V	100mA-500mA
Vibration Motor (316040004) [5]	3.5V	104mA
LED - Blue [6]	4V	20mA

Table 6: Maximum ratings for relevant Hub components

Component	Max Voltage	Max Current
Voltage Regulator (TPSM84203EAB) [7]	30V	1.5A
Rotary Encoder (PEC16-4120F-N0012-ND) [8]	5V	1mA
Push Button (40-2388-01) [9]	14V	2A

By analysing the peak voltages given in Table 5, the AC adapter must supply at least 5V to the circuit for functionality. This voltage is well within the maximum ratings for all components in the circuit, some without significant peak values were included for their maximum ratings in Table 6. Next, it is important to ensure the adapter will be able to provide adequate current to the circuit. The TFT LCD had inadequate documentation for its peak current so a range was found by comparing some similar models. The highest current will be chosen from the range to ensure sufficient current can be drawn. To simplify calculations, all LEDs are assumed to be blue as this color has the highest peak requirements. These estimates will result in a higher total current requirement than what is likely necessary, but leaves room for adjustment since this current can be stepped down. This is found by the following equation:

$$I_{total} = I_{microcontroller} + I_{screen} + I_{motor} + I_{LEDs} \tag{1}$$

$$I_{total} = 500mA + 500mA + 104mA + 19 * 20mA \quad (2)$$

$$I_{total} = 1.484A \quad (3)$$

The total current calculated in Equations 1-3 gives a value that is within the bounds of the chosen voltage regulator. Now, with the voltage and current requirements known, a proper ac adapter, such as the WSU050-1500-R [10], can be chosen with confidence that it fits the ratings required to properly supply the circuit and ensure successful project completion.

3 Cost and Schedule

3.1 Cost Analysis

Table 7 goes through the cost considerations for the production of the ClassroomClarity project. Assuming a salary of around $88,000 per year, which is about the average salary of an electrical engineering graduate from UIUC, an hourly rate of $42/hr can be calculated [12]. Estimating that each team member works about 6 hours per week for 14 weeks total, the total cost of labor sits at about $10,584. At the time of this document's creation, the parts listed in Table 7 are what have been purchased for the creation of the clarity hub. The total amount spent on these components totals to $69.06. Combining these two costs, the grand total for the production of the Classroom Clarity hub is $10,653.

Table 7: Cost analysis for ClassroomClarity production

Labor					
Team Member	*$/hr*	*Hours Worked/week*	*Weeks Worked*	*Total Cost*	
Maddie Donku	$42	6	14	$3,528	
Kailin Gowens	$42	6	14	$3,528	
Jesse Gruber	$42	6	14	$3,528	
				Total Labor Costs:	$10,584
Parts					
Description	*Manufacturer*	*Part Number*	*Quantity*	*Unit Cost*	Total Cost
AC/DC Adapter	Triad Magnetics	WSU050-1500-R	1	$5.88	$5.88
Rotary Encoder	Bourns Inc.	PEC16-4120F-N0012-ND	2	$1.69	$3.38
Vibration Motor	Seeed Technology Co., Ltd	316040004	3	$1.20	$3.60
Barrel Jack (old)	Same Sky	PJ-037A	3	$1.05	$3.15
Barrel Jack (new)	Würth Elektronik	694106301002	2	$0.92	$1.84
Push Buttons (large)	Judco Manufacturing Inc.	40-2388-01	2	$2.35	$4.70
BJT	STMicroelectronics	BD139	3	$0.67	$2.01
LCD Screen	Hosyond	MSP4002	1	$18.99	$18.99
Programmer	DSD TECH	SH-U09C	1	$12.49	$12.49
Micro USB Connector	Molex	473460001	2	$1.01	$2.02
DC DC Converter	Texas Instruments	TPSM84203EAB	2	$5.50	$11.00
				Total Part Costs:	$69.06
				Grand Total:	**$10,653**

3.2 Schedule

Week of	Task	Group Member(s)
2/17	Divide tasks	All
	App and bluetooth research	Kaitlin
	Control signals and programmer research	Jesse
	Power management research	Maddie
2/24	Component selection and ordering	All
	1st Draft of PCB modeling on KiCAD	Maddie
	App Design	Kaitlin
	Control signal organization and planning	Jesse
3/3	Order 1st PCB	Maddie
	Document power management subsystem, budget, timeline, and risks	Maddie
	Document control and feedback subsystems and tolerance analysis	Jesse
	Document user interface subsystem and software design	Kaitlin
	Prepare for breadboard demo	All
	Get basic student side components functional(Understanding rating buttons, text submission, raise hand button)	Kaitlin
	Begin CAD modeling for enclosure	Kaitlin
3/10		All
	Get Bluetooth connection between app and microcontroller functional	Kaitlin
	Finalize GPIO pin configurations	Jesse
	Get all feedback systems working with test data set from microcontroller	All
	Refine CAD prototype	Maddie
3/24	Assemble PCB	Maddie
	Test PCB	All
	Discuss PCB changes/improvements	All
	Edit and order 2nd PCB	Maddie
	3D print casing prototype	All
	Get teacher side of app fully functional	Kaitlin
3/31	Get microcontroller software working so all feedback systems reactly properly to incoming data	Jesse
	Assemble full prototype	All
	Discuss prototype improvements	All
4/7	Order Final PCB (if neccessary)	Maddie
	Add login page with functional student and teacher login + password	Kaitlin
	Edit and print final casing design	Maddie
4/14	Final assembly of clarity hub	All
	Begin Final Report and Presentation	All
	Make app aesthetically pleasing	Kaitlin
4/21	Mock Demo	All
	Finalize presentation	All
4/28	Final Demo	All
	Complete Final Report	All
	Mock Presentation	All
5/5	Final Presentation	All

Figure 15: ClassroomClarity Design Timeline by week

4 Ethics & Safety

This section holds a brief discussion of the ethics and safety related to the project and how we plan to uphold them. The code of ethics this discussion draws from is a combination of the IEEE Code of Ethics [1] and the ACM Code of Ethics and Professional Conduct [2]. As engineers tasked with innovation, we understand the impact we can have on the lives of the communities we serve. That is why we strive to reach the highest standards of safety and ethics, such as the following:

1. **To treat all people fairly and respectfully, which includes not discriminating, harassing, or injuring others [1],[2].**

   Throughout the course of this project, we will be working as a team to create the final product. In order to work the most efficiently and produce the best results, we must strive to listen to each other and respect any and all ideas that are shared. So far in the project we have made efforts to uphold this by taking turns speaking during meetings so that all voices are heard. We have also put into action a choosing process that is based on facts and proof to ensure personal bias doesn't create an unfair decision-making process.

2. **To seek, accept, and offer honest criticism by acknowledging and correcting errors while remaining true to the facts and data available to us [1]**.

   We understand that as individuals we do not know everything, which is why we plan to enter this project with an open mind to new ideas and to listen to the advice we are given. One of the main sources of critiques for the project will likely come from our weekly TA meetings. As such, we plan to guarantee that at least one member every week will take notes during the meeting so we can be sure to review and correct any mistakes that were brought up.

   All of the team members will also be keeping a lab notebook that will contain all the data gathered throughout the project. This will provide us with a base to refer to in order to guarantee that the shared information remains true.

3. **To continue to develop our technical skills and to only accept tasks that we are prepared and qualified for [1].**

   So far, we have already made strides to improve the technical skills we will need for this project. The full team has completed both the KiCad and soldering assignments which will provide good experience for when we build our PCB. Similarly, in preparation for designing the PCB, we all attended Dylan Wagner's presentation to hear advice from someone working in the industry.

As the project progresses, we expect to come across more areas where we lack experience but are necessary for the project; therefore, we plan to continue learning by researching experts in those areas and reading about what they have to say about the topic. We will also ask for advice from the TAs and professors to learn from their experience as well.

4.1 Risk Analysis

The ClassroomClarity hub is a low voltage, low current system but there are still several important considerations to take into account when assessing potential risks.

The primary concern with any electronic device is electrical shock to the user. This risk is mitigated by utilizing an AC adapter with built in short circuit, overcurrent, and overvoltage protection. Additionally, all circuitry and exposed wiring are fully enclosed within the hub's casing to prevent accidental contact and ensure user safety.

Electronic components inherently have losses which generate heat. Heat that is not properly managed can cause burns or discomfort upon contact with the user. Aside from having these components inside the casing of the hub, ventilation is placed on the back of the hub to ensure proper airflow through the device, therefore cooling the components and preventing the case from overheating.

Outside of physical harm, security risks can also pose a threat to hub users. People unenrolled in the class may try to access the hub to post inappropriate questions or skew engagement data. To prevent unwanted users from joining the class, a password was implemented for both teacher and student access.

By addressing these risks, the ClassroomClarity hub is designed to be safe, reliable, and user-friendly, ensuring a seamless experience for both students and professors.

References

[1] IEEE. “IEEE Code of Ethics.” (2020), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/11/2025).

[2] "flutter_blue package," pub.dev, Feb. 28, 2024. [Online]. Available: https://pub.dev/documentation/flutter_blue/latest/

[3] Espressif Systems. “ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U Datasheet.” (2021), [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf (visited on 03/06/2025)

[4] LCD WIKI. “4.0inch SPI Module MSP4022&MSP4023 User Manual.” (2021), [Online]. Available: http://www.lcdwiki.com/res/MSP4022/4.0inch_SPI_Module_MSP4022&MSP4023_User _Manual_EN.pdf (visited on 02/12/2025).

[5] Seeed Technology Co., Ltd. “Product Specification.” (2016), [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2226/1027_Spec.pdf (visited on 03/06/2025).

[6] Kingbright. “WP5603QBDL/SD/G Oval Solid State Lamp.” (2018), [Online]. Available: https://www.kingbrightusa.com/images/catalog/SPEC/WP5603QBDL-SD-G.pdf (visited on 03/06/2025).

[7] Texas Instruments. “TPSM84203, TPSM84205, TPSM84212 1.5-A, 28-V Input, TO-220 Power Module.” (2017), [Online]. Available: https://www.ti.com/lit/ds/symlink/tpsm84203.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1741317560705&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%\253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Ftpsm84203 (visited on 03/06/2025).

[8] Bourns. “PEC16 - 16 mm Incremental Encoder.” (2023), [Online]. Available: https://www.bourns.com/docs/Product-Datasheets/pec16.pdf (visited on 03/06/2025).

[9] Judco Manufacturing Inc. “40-2388-01.” (2006), [Online]. Available: https://www.digikey.com/en/products/detail/judco-manufacturing-inc/40-2388-01/185674 (visited on 03/06/2025).

[10] Triad Magnetics. “WSU050-1500-R.” (2021), [Online]. Available: https://catalog.triadmagnetics.com/Asset/WSU050-1500-R.pdf

[11] Arduino. “ArduinoBLE.” (2024), [Online]. Available: https://docs.arduino.cc/libraries/arduinoble/ (visited on 03/06/2025)

[12] The Grainger College of Engineering. “Post-Graduation Success.” (2023), [Online]. Available: https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/electrical-engineering#:~:text=Post%2DGraduation%20Success&text=The%20average%20salary%20between%20for,average%20signing%20bonus%20of%20%245%2C000.&text=*Post%2Dgraduate%20data%20was%20collected,self%2Dreported%20survey%20data%20sources. (visited on 03/06/2025)