Zachary Milan

**UM-LED Design Proposal**

*This document outlines the planned design and functionality of UM-LED during its early design phase, prior to detailed PCB design, hardware assembly, and firmware implementation. It serves as a proposal for the intended system and reflects my independent design process and planning.*

## 1. Project Overview

UM-LED is a remote-controlled LED display in the shape of the University of Michigan's Block-M, intended to be a visually striking addition to a room or workspace. The system resembles a consumer product, offering features such as adjustable brightness, multiple LED modes, auto-dimming, and rechargeability. The display rests on a flat surface, supported in an upright position by an attached stand.

The project consists of two parts:

- LED display – a Block-M configuration of individually-controlled LEDs, capable of displaying a variety of patterns with varying brightness.
- Remote Control – a handheld controller with tactile feedback that allows users to wirelessly toggle power, cycle through modes, and adjust brightness of the display.

Together, the display and remote create an interactive device that feels polished, responsive, and fun to use.

## 2. Concept and Purpose

UM-LED is inspired by my pride for the University of Michigan (Go Blue!) and the desire to create something unique for my workspace. At the same time, it serves as a platform to sharpen my embedded systems skills by taking a mere idea through the entire development process, from concept and planning, to PCB schematic and layout, to firmware and mechanical integration, finishing with a polished, consumer-style product.

The design draws from consumer products I admire and use regularly, such as Apple's use of capacitive touch and haptic feedback, and the simplicity of my rechargeable wireless speaker. These influences shape both the system's interface and the overall user experience.

Figure 1 shows early concept art of UM-LED, illustrating the Block-M display and associated remote, highlighting the consumer-style appearance with simple, intuitive user interfaces for each.

Figure 1: Rough Concept Art of UM-LED

## 3. System Features

UM-LED includes several features across both the remote and display that enhance the user experience and make the system unique.

### Remote Control

- Wireless control of LED display via Bluetooth Low Energy (BLE)
- Brightness control via a capacitive-touch slider
- Haptic feedback upon brightness adjustments
- Mode, power, and auto-dim buttons
- Status indicators for charging, low battery, and button presses
- Rechargeable via USB-C

### LED Display

- Through-hole LEDs arranged in the Block-M pattern (yellow fill, blue outline)
- Cyclable LED patterns/modes (heartbeat, snake, demo cycle, etc.)
- Local mode, power, and auto-dim buttons
- Local brightness control via rotating knob
- Status indicators for charging, battery level, auto-dim mode, and button presses
- Rechargeable via USB-C
- A microSD card for logging the display's power usage over time

## 4. Functional Requirements

UM-LED's functional requirements are listed below, and are achieved through a combination of component selection, hardware design, and firmware optimizations.

### Remote Control

- Recharge time: ≤ 5 hours via USB-C
- Battery life: ≥ 1 week per charge under normal use
- BLE range: 3-5m indoors
- Haptic feedback latency: ≤ 100ms
- Brightness control: ≥ 20 levels

### Display

- Recharge time: ≤ 10 hours via USB-C
- Battery life: ≥ 1 week per charge under normal use
- Supports ≥ 5 LED animation modes
- Restores last brightness and mode on startup

### System Level

- Automatic BLE reconnection after sleep/wake cycle
- Remote inputs update the display within 200ms
- Power usage data is automatically stored when microSD card is present

## 5. System Architecture

The UM-LED system is implemented on two custom printed circuit boards (PCBs) – the display and the remote. Each PCB consists of its own microcontroller, dedicated power system, and necessary peripherals to provide the intended features and functionality.

## Remote Architecture

The remote PCB is built around an STM32 microcontroller and integrates multiple user input and feedback features. Brightness is adjusted using a capacitive touch slider implemented through the STM32's native TSC peripheral, while dedicated tactile pushbuttons provide mode, power, and auto-dim control. A linear resonant actuator (LRA), driven by a haptic driver IC, delivers haptic feedback during brightness adjustments. Wireless communication with the display is handled by a BLE module, and a reset button and SWD header are included for firmware development and debugging. Additional user feedback includes LED indicators for button presses, low battery, and charging status.

Power is supplied by a single-cell 3.7V Li-ion battery, rechargeable via a USB-C receptacle and dedicated charging IC. The battery powers a 3.3V rail – regulated by an LDO – and the haptic driver directly. Figure 2 illustrates the remote's full architecture.

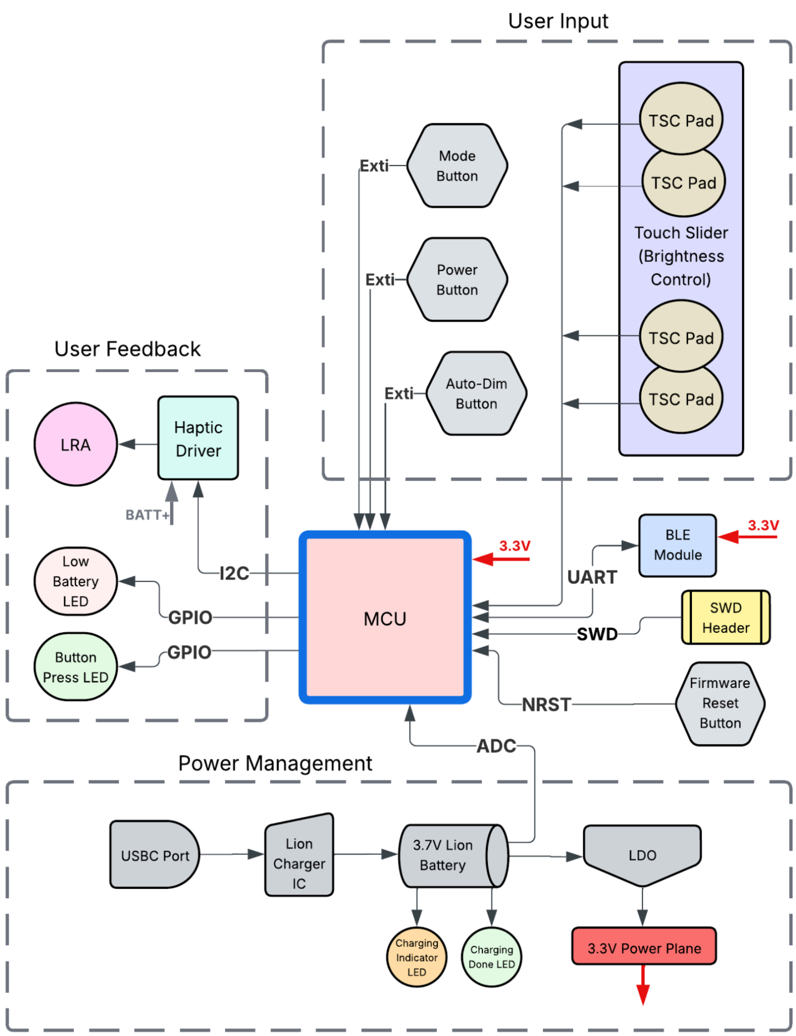


Figure 2: Remote PCB System Architecture

## Display Architecture

The display PCB is built around an STM32 microcontroller and drives the Block-M LED configuration through daisy-chained constant-current LED drivers. Local user input is provided by pushbuttons (mode, power, auto-dim) and a rotary potentiometer for brightness adjustment via ADC readings. An ambient light sensor supports automatic dimming functionality. Wireless control is handled by a BLE module – receiving control data from the remote – and a microSD card slot is included for power-usage logging. A 5-segment LED bar indicates battery status, while other LED indicators exist for button presses and charging status.

Power is supplied by a single-cell 3.7V Li-ion battery, rechargeable via a USB-C receptacle and dedicated charger IC. The battery supplies two voltage rails – a 5V rail via a boost converter (powering the LED configuration), and a 3.3V rail via an LDO for the MCU and peripheral circuitry. A power switch exists to completely cut battery power to the system. Figure 3 illustrates the display's full architecture.

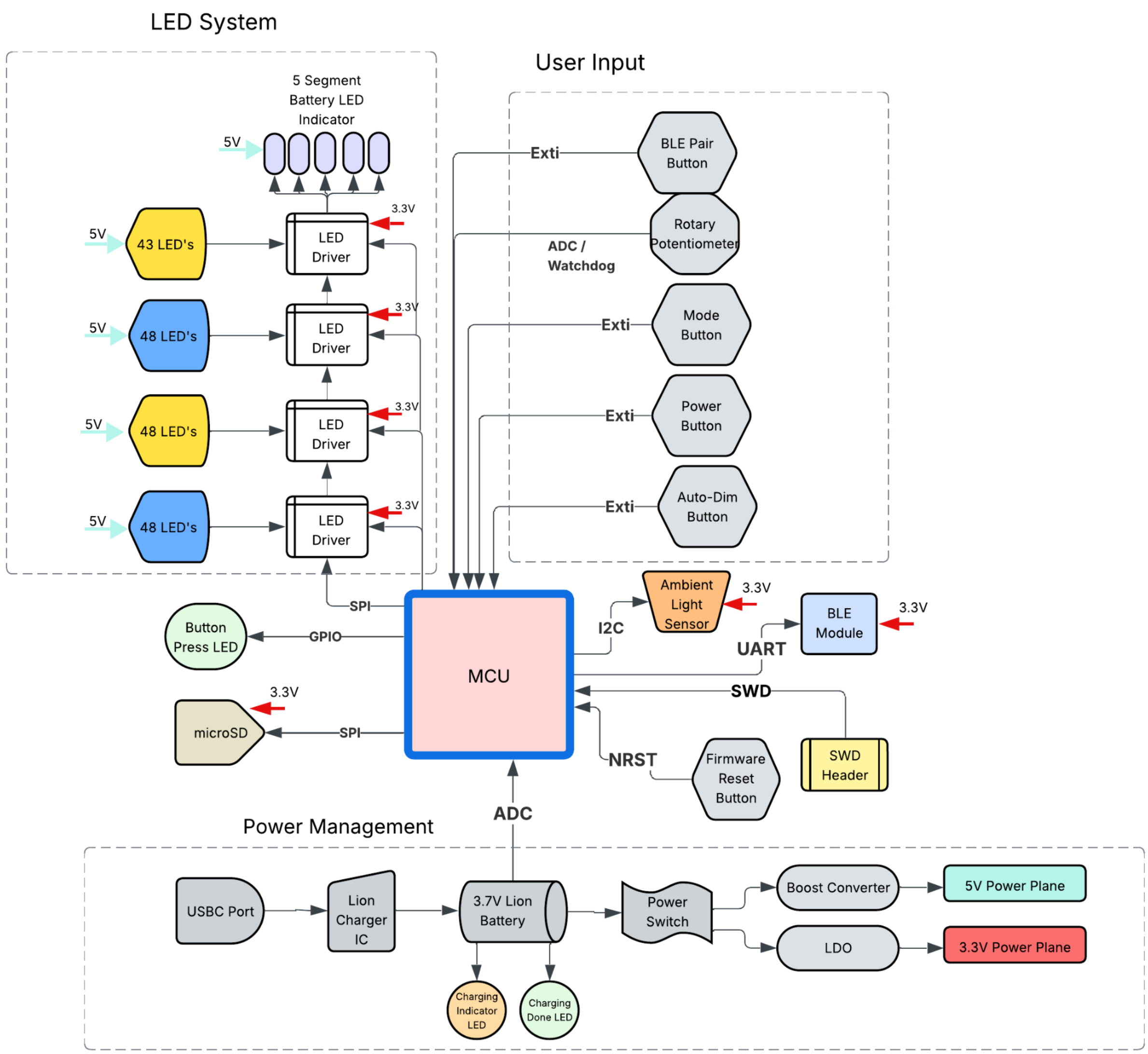


Figure 3: Display PCB System Architecture

## 7. Major Components

Table 1 lists major components and their functionality and usage across the remote and display boards.

Table 1: Major Components

| Component | Description & Functionality | Used In |
|---|---|---|
| STM32F303RCT6 MCU | Microcontroller for both remote and display boards; chosen for high pin count, built-in TSC peripheral, and variety of serial communication peripherals. | Remote, Display |
| IS31FL3248 LED Driver | 48-channel constant-current-sink LED driver IC. Chosen for high channel count, SPI interface, and constant-current capability. | Display |
| BM71BLES1FC2 BLE Module | BLE module for both remote and display. Chosen for its on-board antenna, UART interface, auto-operation feature, and configurable GPIOs. | Remote, Display |
| DRV2605L Haptic Driver | Haptic driver IC to control LRA for haptic feedback. Chosen for its I2C interface and built-in haptic effects library. | Remote |
| MCP73831 Linear Charge Controller | Linear charge management controller for remote's Li-ion battery. Chosen for its simplicity and ease-of-use, as well as programmable charge current and charge status pin. | Remote |
| MAX77715 Autonomous USB-C Charger | Charge management controller for display's Li-ion battery. Chosen for its fast charge current, automatic power source detection, and automatic input-current limit configuration. | Display |
| TLV755P LDO | Low-dropout regulator to supply 3.3V rails. Chosen for its popularity and simplicity, and reliability in a past project. | Remote, Display |
| MP3432 Boost Converter | Synchronous boost converter to supply 5V rail on the display. Chosen for its capability to output a high amount of current. | Display |
| VG0832013D LRA | Linear resonant actuator providing haptic feedback on the remote. Chosen for its small profile and popularity. | Remote |

## 8. Implementation Details

### Remote Firmware

*Capacitive Touch Slider* – The STM32's TSC peripheral samples touch pads simultaneously. Values are smoothed/averaged, extreme readings mitigated, and the final result mapped to a discrete brightness level.

*Buttons* – Mode, power, auto-dim buttons generate interrupts upon press. Button debouncing and long-press detection occur in software.

*Haptic Feedback* – A linear resonant actuator (LRA) is driven by a dedicated haptic driver IC, controlled by the STM32 via I2C and a TRIG pin. The TRIG pin activates a predetermined haptic effect, occurring in sync with brightness changes sent over BLE.

*BLE* – The BLE modules function in "Auto-Operation" mode, acting as a UART bridge between boards for transmitting control data such as brightness, mode, auto-dim, and power.

### Display Firmware

*LED Control* – The LED drivers are initialized and controlled via SPI. Each of the driver's 48 channels is controlled individually via PWM registers, allowing for flexible patterns and a wide range of brightness. Multiple drivers are daisy-chained to support a high number of LEDs. Additionally, enabling the BLANK pin allows for immediate shutoff of all LEDs connected to a driver.

*Local Controls* – Similar to the remote, buttons on the display provide user access to mode, power, and auto-dim controls and generate interrupts upon press. Button debouncing and long-press detection occur in software. For brightness control, a rotary potentiometer provides an analog input, which firmware maps to a brightness scale for the display.

*Ambient Light Sensor* – In auto-dim mode, the display periodically takes readings from an ambient light sensor and updates global LED brightness accordingly.

### Power Management

*MCU Sleep/Wake* – Both boards utilize low-power or sleep modes when idle, and wake upon user input or BLE events.

*Battery Indicators* – Voltage readings of the battery are taken via ADC. Firmware compares readings to determined thresholds and updates the display's 5-segment bar or remote's low-battery LED accordingly.

*Charging* – Charging ICs on both the remote and display handle safe battery charging without the need for firmware intervention.

### Programming & Configuration

*MCU Programming* – The STM32 on both boards are programmed via SWD. Each board exposes SWD pins through a footprint compatible with a Tag-Connect cable connected to a ST-Link programmer.

*BLE Configuration* – The BLE modules are configured through UART using Microchip's PC GUI tool. Low-profile test loops are provided on both boards to provide programming access for the module.

## 9. Proposed Physical Design

The proposed physical design of both the remote and display establishes the placement of user-interface elements in order to support preliminary STM32 pin-mapping and simplify PCB routing. These illustrations do not include all major components listed in Table 1, but emphasize user-interface and feedback elements that are given priority for pin assignments. Figures 4 and 5 illustrate the physical layout of the remote and display.

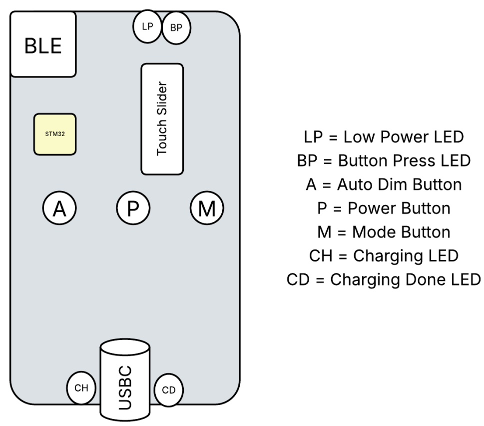


Figure 4: Remote Physical Layout

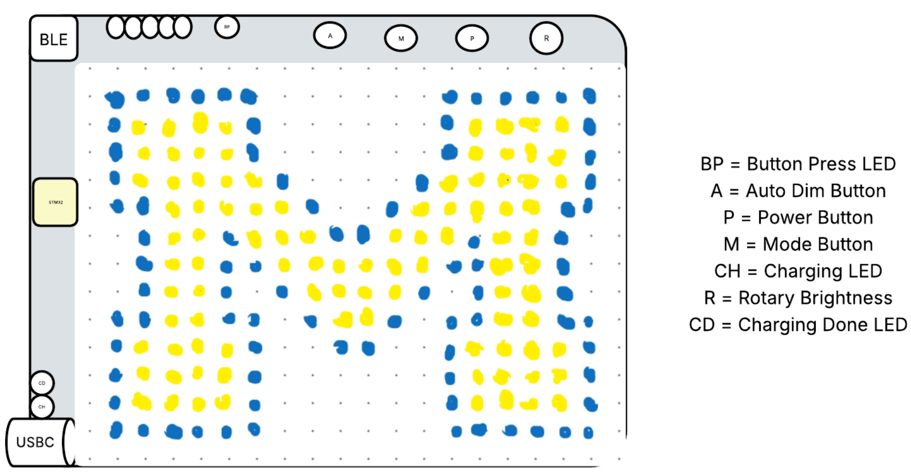


Figure 5: Display Physical Layout

## 10. Anticipated Technical Challenges

UM-LED involves several technical hurdles spanning hardware, firmware, wireless communication, and mechanical design. Table 2 details anticipated technical challenges along with a brief description of each.

Table 2: Technical Challenges

| Technical Challenge | Description |
|---|---|
| LED Driver Thermal Management | Managing heat dissipation from multiple constant-current LED drivers under high load (stemming from high brightness levels). |
| Switching Noise Suppression | Reducing switching noise from the boost converter and LED drivers on the display, which could interfere with serial communication and affect power integrity. |
| Reliable Wireless Communication | Ensuring automatic BLE reconnection between remote and display after waking, and achieving low-latency responses between remote user input and a display update. |
| Capacitive Touch Algorithm | Developing firmware to translate readings from a limited number of touch pads to numerous brightness levels and avoiding sudden jumps to extreme values. |
| MCU Power Management | Designing an efficient sleep/wake control flow for both boards in order to balance responsiveness during use with extended battery life when idle. |
| Power Logging to microSD Card | Implementing accurate and reliable logging of power usage data, including determining an appropriate sampling frequency and logging duration. |
| Mechanical Integration | Designing and producing polished 3D enclosures for both boards that expose the user interface and hide internal hardware. |

## 11. Stretch Goals / Future Development

UM-LED has potential for several upgrades once the base system is complete. Extended functionality can be implemented through firmware updates, additional hardware, or both. Table 3 details possible stretch goals and a description for each.

Table 3: Stretch Goals

| Stretch Goal | Description |
|---|---|
| Mobile Companion App | Develop a mobile app to control the display over BLE, offering an alternative to the physical remote and enabling extended control functionality. |
| Audio-sensitive LEDs | Add hardware and firmware to allow the display to react dynamically to ambient sound. |
| Battery Percentage Estimation | Provide a precise battery percentage readout on the display, potentially via a small LCD or OLED, instead of a 5-segment LED bar. |
| Extended Capacitive Slider Functions | Enable the remote's capacitive slider to perform different actions depending on the mode (e.g. brightness adjustment *and* direct LED control) |
| Improved Mounting Options | Develop solutions for flexible mounting, such as wall brackets or magnetic attachments, in addition to the default stand. |

## 12. Project Timeline

### Target Completion

Full functionality and features of UM-LED (excluding the 3D enclosure) implemented by October 31, 2025.

### Current Status (as of September 17, 2025)

Remote schematic complete; PCB layout in progress. Next step: Completing layout and ordering parts.

### Looking Ahead

Table 4 details a rough timeline for future development, and Figure 8 on the following page illustrates this timeline graphically.

Table 4: Future Development Timeline

| | |
|---|---|
| Mid-September | Finalize remote schematic, complete remote layout, and submit for fabrication; order associated parts. |
| Late-September | Begin and complete display pin-mapping, schematic, and layout; submit for fabrication and order associated parts. Begin early firmware setup/development. |
| Early-October | Assemble and bring-up remote PCB. |
| Mid-October | Assemble and bring-up the display PCB. Expand firmware development to support BLE communication and local inputs on remote and display. |
| Late-October | Integrate core firmware features (brightness slider, mode cycling, auto-dim, microSD card logging). |
| October 31 | Demonstrate full system functionality on both boards. |
| November | Design and print custom 3D enclosures for the remote and display. |
| December | Implement stretch features (optional) |

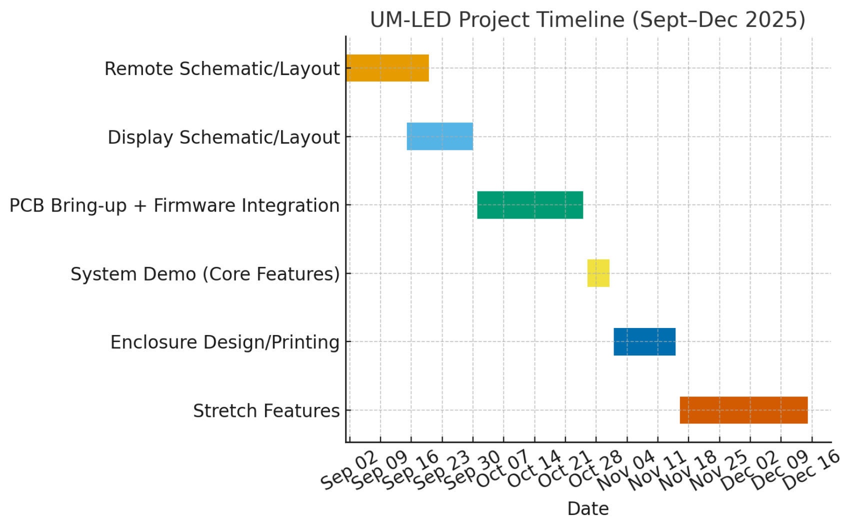


Figure 8: UM-LED Implementation Timeline

*Note: The UM-LED development schedule is subject to change due to technical challenges or unforeseen circumstances. The above serves as an estimated timeline.*