Focus Dial

ECE 445 Design Document – Spring 2026

Team #45

Ahan Goel (ahang5)

Amogh Mehta (amoghm3)

Ben Loo (bloo2)

Professor: Craig Shultz

TA: Frey Zhao

Contents

1 Introduction

1.1 Problem

Sustained attention is increasingly difficult to achieve in modern digital environments characterized by constant notifications, multitasking, and on-demand connectivity. Although structured focus methods such as the Pomodoro technique have proven effective for improving productivity and reducing cognitive fatigue, their implementation typically relies on screen-based tools. Smartphone applications, browser extensions, and desktop timers require users to interact directly with devices that are themselves sources of distraction. This design conflict creates friction. Activating or checking a timer often exposes users to notifications, messages, or unrelated applications. As a result, focus tools often increase the likelihood of interruption. Many users abandon structured focus methods not because they lack effectiveness, but because they are difficult to sustain within distraction-heavy environments. Beyond productivity, fragmented attention contributes to stress, burnout, and decreased overall well-being. In academic and professional settings, task-switching reduces work quality and increases completion time. There is a need for a dedicated, low-friction, hardware-based solution that enables structured focus management without requiring repeated screen interaction.

1.2 Solution

The proposed Focus Dial is a dedicated hardware device designed to enable intentional focus management through tactile interaction rather than screen navigation. The device allows users to activate, modify, and deactivate the Focus Mode or Do Not Disturb settings on a paired personal device via Bluetooth communication. The primary interface consists of a rotary encoder with an integrated push-button. The user rotates the dial to select the timer duration or focus state and presses to confirm the action. Upon confirmation, the device transmits a command to the paired device and provides immediate feedback through an LED ring, a display, and a haptic motor. All core functionality, including timer tracking, state management, and feedback, operates locally on the device without requiring internet connectivity. Optional local network integration enables communication with devices such as smart lighting systems. By relocating focus control to a purpose-built physical interface, the Focus Dial reduces interaction with distracting digital environments while preserving the benefits of structured time management techniques.

1.3 Visual Aid

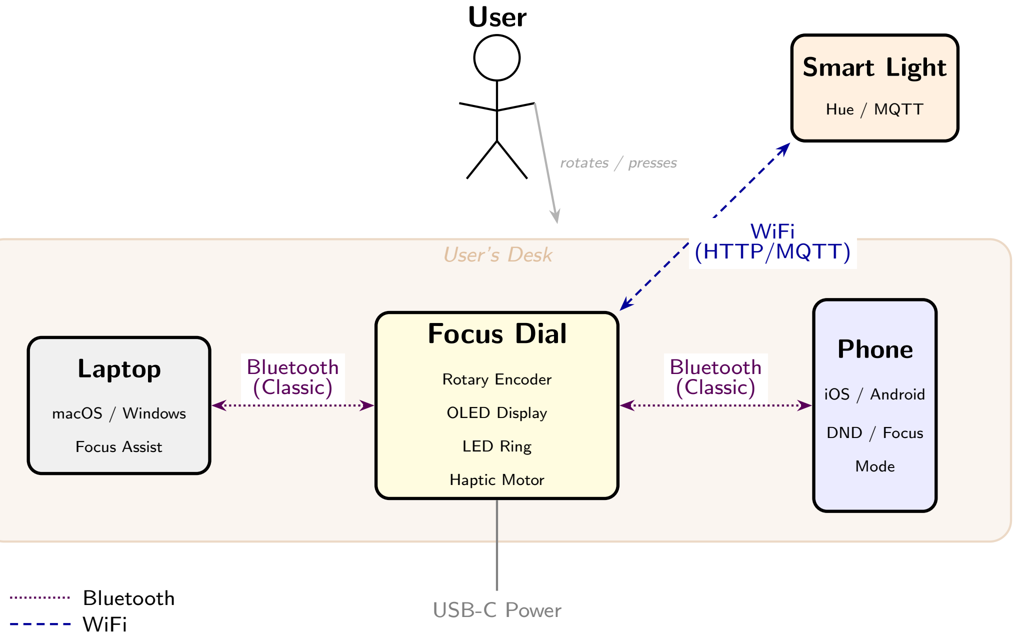

Figure 1: Visual aid showing Focus Dial in a typical use context. The user interacts with the dial on their desk; the dial communicates wirelessly with paired devices.

1.4 High Level Requirements

To consider our project successful, the Focus Dial must fulfill the following:

1. The system shall initiate, modify, or terminate a focus session within 1 second of a user's

rotary input, ensuring low-friction interaction that does not interrupt or distract the user from their task.

2. The system must communicate focus state changes to a paired device via Bluetooth and complete the activation or deactivation of the target focus mode.

3. The device must provide immediate feedback (visual via display and LED ring, and tactile via haptic motor) that accurately reflects the current focus state and timer status at all times.

4. The device shall track and display focus and break intervals with a timing accuracy of ±1 second per 30-minute session, ensuring reliable implementation of structured time management techniques

5. The device shall maintain continuous operation for at least 8 hours of active use, enabling users to complete multiple focus sessions throughout a typical work or study day without interruption.

2 Design

2.1 Physical Design

The Focus Dial is housed in a cylindrical enclosure approximately 10 mm in diameter and 60 mm in height, designed to sit stably on a desk surface. The top is a dial that the user rotates and presses to interact with the device. Surrounding the dial is a translucent diffuser ring that sits above the 15-LED WS2812B ring, providing ambient visual feedback visible from all angles.

A circular cutout on the front face of the enclosure exposes the SPI-driven OLED display, which shows timer state and system status information. The rear of the enclosure contains a recessed USB-C port for power input and programming. Internally, the main PCB is mounted horizontally at the base of the enclosure using M2 standoffs. The motor is mounted centrally beneath the encoder shaft to deliver haptic feedback directly through the knob. Passive ventilation slots are located near the base perimeter to dissipate heat from the ESP32 module and LED ring.

2.2 Block Diagram

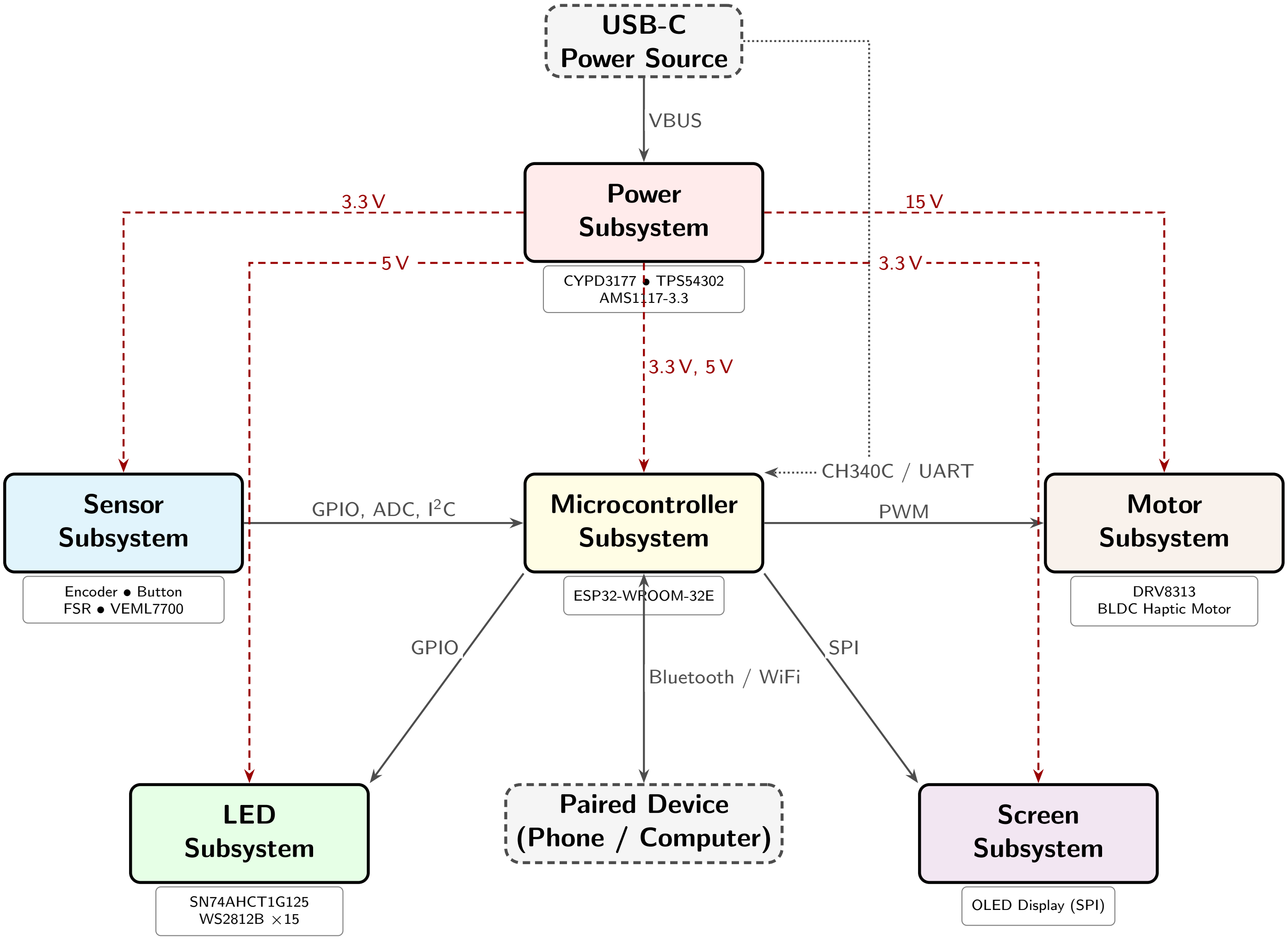

Figure 2: Focus Dial system block diagram

2.3 Subsystems

2.3.1 Power Subsystem

The Power Subsystem is responsible for distributing, regulating, and converting electrical power to all other subsystems. It accepts external power via a USB-C connector and negotiates a 15 V Power Delivery (PD) profile using a dedicated USB-C PD controller. The negotiated 15 V rail serves as the primary system supply. From this 15 V input, a synchronous buck converter generates a regulated 5 V rail, and a low-dropout (LDO) regulator derives a stable 3.3 V rail from the 5 V output. These rails power the microcontroller, sensor Subsystem, screen Subsystem, and associated peripherals. High-current loads such as the motor subsystem are supplied directly from the regulated 15 V rail.

Isolating power negotiation and regulation into a dedicated subsystem ensures:

- Protection against voltage fluctuations
- Stable logic operation
- Minimal latency due to brownout or resets
- Compliance with High-Level Requirement 2 (low-latency interaction)

Requirement	**Verification**
• When a USB-C PD compliant source is connected, the Power Subsystem shall negotiate and stabilize at 15.0 V±5% within 2.0 s at ambient temperature 20°C–25°C.	• Connect a PD-capable USB-C source ($\geq$ 45 W) to J1. • Connect a digital multimeter across TP1 (VBUS) and TP2 (GND) as labeled in the power schematic. • Connect a 100 MHz oscilloscope probe to TP1 with ground referenced to TP2. • Begin oscilloscope capture prior to cable insertion. • Insert the USB-C cable and start a stopwatch simultaneously. • Measure the time required for VBUS to reach steady-state voltage. • Record steady-state voltage and stabilization time in a laboratory data table. • Repeat for five independent trials. Pass condition: All trials must show voltage between 14.25–15.75 V and stabilization time $\leq$ 2.0 s. Results will be presented as a table of measured values and a representative oscilloscope capture of the VBUS ramp.

Table 1: Power Subsystem Requirement and Verification

2.3.2 Microcontroller Subsystem

The microcontroller subsystem serves as the computational core of the Focus Dial system. It performs real-time acquisition of user input signals, executes the device state machine, manages timekeeping for focus intervals, coordinates feedback outputs, and handles communication scheduling with external interfaces. Electrically and logically, it is the central integration point between all subsystems.

The subsystem is powered by the regulated 3.3 V rail generated by the power subsystem. It consumes a typical operating current of approximately 120 mA and a peak current of 250 mA during simultaneous peripheral activity (SPI display update + PWM + Bluetooth transmission). Brown-out detection is configured at 3.0 V to prevent undefined operation under transient supply disturbances.

Requirements	**Verification**
• When the Microcontroller Subsystem detects a valid rotary encoder press event and the device is paired to a host device, the Microcontroller Subsystem must transmit a focus activation command and update all local feedback (OLED and LED ring) within 1.0 second of user input under nominal operating voltage (3.3 V ± 5%).	• – Ensure the Focus Dial is powered via USB-C and paired to a smartphone with Bluetooth enabled. – Ensure the system is in an idle state and not currently in Focus Mode. – Using a stopwatch, press the rotary encoder to initiate a focus session. – Measure the time between physical button press and: ∗ LED ring state change ∗ OLED display update ∗ Focus Mode activation on the host device – Repeat this procedure for 10 independent trials. – Record all measured times in a table. Pass if all measured activation times are less than or equal to 1.0 second.

Table 2: Microcontroller Subsystem – Focus Activation

2.3.3 Sensor Subsystem

The Sensor Subsystem captures physical user input and converts it into electrical signals for processing by the microcontroller subsystem. It consists of a rotary encoder, a push-button mechanism, and optional force or strain sensing elements. These sensors enable intentional, tactile interaction without reliance on external screens.

The rotary encoder generates two phase-shifted digital signals that allow the microcontroller to determine both rotational magnitude and direction. Interrupt-driven edge detection ensures accurate capture of up to 200 detents per second without missed transitions. The push-button input is debounced in firmware using a 5 ms filtering window to eliminate false triggers.

All sensor interfaces operate at 3.3 V logic levels and are designed to maintain signal integrity under motor-induced electrical noise. This subsystem directly supports High-Level Requirement 1 by enabling precise, low-latency physical interaction and supports Requirement 2 by ensuring input detection occurs within 10 ms of user action.

Requirements	**Verification**
• The subsystem shall detect rotary encoder increments/decrements without missed counts up to a user rotation rate of 200 detents/s.	• Rotate the knob at increasing speeds while logging counts; compare against a known reference count over 100 rotations. Pass if missed count error $\leq 1\%$.
• A single press interaction shall be recognized with debounce time $\leq 5\,\text{ms}$ and no false triggers.	• Perform 100 presses under normal use and vibration; count false positives/negatives. Pass if false triggers $= 0$ and miss rate $\leq 1\%$.

Table 3: Sensor Subsystem - Requirements & Verification

2.3.4 Screen Subsystem

The Screen Subsystem provides low-latency visual feedback that communicates system state independently of any external device. It displays timer duration, countdown progress, focus/break state, pairing status, and error conditions. The display is not used for primary interaction. Rather, it serves as a state indicator that reinforces tactile input.

This subsystem supports High-Level Requirement 1 by eliminating the need for phone-based interaction and Requirement 3 by providing clear visual confirmation of system state.

All display updates must occur within 100 ms of a state change to maintain perceptual alignment with haptic and LED feedback. An SPI-controlled LED module was selected to ensure sufficient refresh bandwidth and minimize update latency.

Requirements	**Verification**
• The screen subsystem must display contextual system information provided by the microcontroller.	• Change system states (e.g., timer start, timer stop, focus mode toggle). • Confirm that the displayed information updates accordingly. • Confirm that the information shown matches internal system state.
• The system shall display the elapsed timer value with a maximum deviation of 1 second over a continuous 60-minute operating period.	• Start a 60-minute timer and compare the displayed time to a calibrated reference clock at both the start and end of the test period.

Table 4: Screen Subsystem - Requirements & Verification

2.4 Tolerance Analysis

End-to-End Latency Budget

The most critical performance constraint is High-Level Requirement 1: the system must initiate, modify, or terminate a focus session within 1 second of user input. This requires a detailed analysis of the end-to-end signal path from rotary encoder input to focus mode activation on the paired device. The total latency is decomposed as:

$$T_{\text{total}} = T_{\text{input}} + T_{\text{fw}} + T_{\text{BT}} + T_{\text{host}} \leq 1.0 \text{ s}$$

1. Input Detection Latency (T_{input}): The rotary encoder generates signals on two GPIO pins. The ESP32 detects edges via hardware interrupts. At the maximum specified rotation rate of 200 detents/s, the minimum time between encoder edges is:

$$t_{\text{edge}} = \frac{1}{200 \text{ detents/s}} = 5.0 \text{ ms}$$

The ESP32 runs at a clock frequency $f_{\text{clk}} = 240$ MHz. The interrupt service routine (ISR) reads two GPIO pins and updates a counter. This requires approximately 20 instructions, giving an ISR execution time of:

$$t_{\text{ISR}} = \frac{20}{240 \times 10^6 \text{ Hz}} \approx 83 \text{ ns}$$

For the push-button confirmation, a firmware debounce window of $t_{\text{debounce}} = 5$ ms is applied. Including interrupt latency overhead (approximately 1 μs on the ESP32), the worst-case input detection latency is:

$$T_{\text{input}} = t_{\text{debounce}} + t_{\text{ISR}} \approx 5.0 \text{ ms}$$

2. Firmware Processing Latency (T_{fw}): After input detection, the firmware must execute the state machine transition, update the internal timer state, queue the display update (SPI write to OLED), update the LED ring (serial data to WS2812B chain), and prepare the BLE notification payload. The ESP32's dual-core architecture allows parallel execution of communication and I/O tasks.

The SPI display update for a partial screen refresh at 10 MHz SPI clock with a 128×64 pixel buffer requires:

$$t_{\text{SPI}} = \frac{128 \times 64 \times 1 \text{ bit/pixel}}{10 \times 10^6 \text{ Hz}} = 0.819 \text{ ms}$$

The WS2812B LED chain update for 15 LEDs (24 bits per LED at 800 kHz data rate) requires:

$$t_{\text{LED}} = \frac{15 \times 24}{800 \times 10^3 \text{ Hz}} = 0.45 \text{ ms}$$

Including state machine execution (∼0.1 ms) and Bluetooth payload preparation (∼0.2 ms), and noting that SPI and LED updates can overlap with Bluetooth scheduling on the second core:

$$T_{\text{fw}} \approx \max(t_{\text{SPI}}, t_{\text{LED}}) + t_{\text{state}} + t_{\text{prep}} \approx 0.819 + 0.1 + 0.2 = 1.12 \text{ ms}$$

3. Bluetooth Transmission Latency (T_{BT}): The ESP32 Bluetooth Classic stack uses the Serial Port Profile (SPP) for data transfer over an RFCOMM channel. Once the SPP connection is established, data can be transmitted immediately without waiting for a connection interval. The ESP32 Bluetooth Classic radio operates at a baseband rate of 3 Mbps (EDR). For a typical command payload of 20 bytes, the transmission time is:

$$t_{\text{packet}} = \frac{20 \times 8}{3 \times 10^6 \text{ Hz}} \approx 0.053 \text{ ms}$$

However, the dominant latency component is the SPP/RFCOMM protocol overhead and scheduling. Empirically, the round-trip latency for an SPP write on ESP32 is approximately

20–40 ms. Using the worst case:

$$T_{\text{BT}} = t_{\text{SPP}} + t_{\text{packet}} = 40 + 0.053 \approx 40 \text{ ms}$$

4. Host Processing Latency (T_{host}): The paired device (smartphone or computer) must receive the Bluetooth SPP data, parse the command, and activate the corresponding focus mode via the operating system API. On iOS, this is handled through a Shortcuts automation. Empirical measurements of Bluetooth-triggered automations on iOS 17 and Android 14 show typical response times of:

$$T_{\text{host}} \approx 100\text{–}300 \text{ ms}$$

We use the worst-case value of $T_{\text{host}} = 300$ ms.

Total Worst-Case Latency:

$$T_{\text{total}} = T_{\text{input}} + T_{\text{fw}} + T_{\text{BT}} + T_{\text{host}} = 5.0 + 1.12 + 40 + 300 = 346.12 \text{ ms}$$

This yields a worst-case total latency of approximately **346 ms**, which provides a margin of:

$$\text{Margin} = 1000 - 346.12 = 653.88 \text{ ms}$$

This 65.4% margin comfortably satisfies High-Level Requirement 1, even accounting for occasional Bluetooth retransmissions (adding up to one additional SPP retry of ~40 ms) or host-side scheduling delays. Under such degraded conditions:

$$T_{\text{degraded}} = 346.12 + 40 = 386.12 \text{ ms} \ll 1000 \text{ ms}$$

The system remains well within the 1-second latency requirement under all anticipated operating conditions.

2.5 Cost Analysis

Parts Cost

Description	Manufacturer	Qty	Ext. Price	Link
Integrated Circuits				
ESP32-WROOM-32E-N4 WiFi/BT Module	Espressif	1	$4.84	DigiKey
CH340C USB-UART Bridge, SOIC-16	WCH	1	$0.65	DigiKey
DRV8313PWPR BLDC Motor Driver	TI	1	$4.26	DigiKey
VEML7700 Ambient Light Sensor (I2C)	Vishay	1	$1.24	DigiKey
SN74AHCT1G125 Buffer/Level Shifter	TI	1	$0.14	DigiKey
TPS54302 3A Buck Regulator	TI	1	$1.29	DigiKey
CYPD3177 USB-C PD Controller	Infineon	1	$2.01	DigiKey
AMS1117-3.3 LDO 3.3V 1A	AMS/EVVO	1	$0.27	DigiKey
Discrete Semiconductors				
WS2812B Addressable RGB LED	Worldsemi	15	$5.25	DigiKey
S8050 NPN Transistor (TO-92)	Unisonic	2	$0.20	DigiKey
DMP3013SFV P-CH MOSFET	Diodes Inc.	1	$0.87	DigiKey
LED Indicator, 0805 SMD	Generic	1	$0.15	DigiKey
Passive Components				
100μF Electrolytic Capacitor, Radial	Generic	2	$0.60	DigiKey
100nF Ceramic Capacitor, 0805 (C2–C27)	Generic	18	$1.80	DigiKey
Ceramic Caps (470nF, 10μF, 1μF, 47μF)	Generic	8	$2.40	DigiKey
10μH Inductor, 0603	Generic	1	$0.30	DigiKey
Resistors, various values, 0805	Generic	25	$2.50	DigiKey
Connectors & Switches				
USB-C Receptacle 16P, SMD	HCTL	1	$0.70	DigiKey
STEMMA QT / JST SH 4-Pin Connector	JST	2	$1.00	DigiKey
JST ACH 3-Pin Connector (Motor)	JST	1	$0.60	DigiKey
Pin Headers & Sockets, 2.54mm	Generic	3	$0.90	DigiKey
Force Sensing Resistor	Interlink	1	$7.00	DigiKey
DPST Push Toggle Switch	C&K	1	$2.50	DigiKey
Tactile Switch 6x6mm, SMD (Boot/Reset)	Generic	2	$0.50	DigiKey
Mechanical & Misc				
M2 Mounting Hardware	—	4	$0.50	—
Test Points, 1mm pad	—	19	$1.90	DigiKey
		Total Parts Cost	**$40.07**	

Table 5: Bill of Materials

Labor Cost

Team Member	**Hourly Rate**	**Hours**	**Total**
Ahan Goel	$40/hr	100	$4,000
Amogh Mehta	$40/hr	100	$4,000
Ben Loo	$40/hr	100	$4,000
Total Labor Cost			**$12,000**
Grand Total (Parts + Labor)			**$12,040.07**

Table 6: Labor Cost Estimate ($40/hr × 2.5× multiplier = effective cost)

2.6 Schedule

Table 7: Schedule for Project Progression

Week	**Task**	**Person**
Wk 1	Finalize architecture, R&V table	Everyone
Wk 2	Schematic + power bring-up plan	Ben & Amogh
Wk 3	PCB layout + enclosure CAD	Ahan
Wk 4	Firmware bring-up (inputs/UI)	Everyone
Wk 5	BLE host integration + latency test	Everyone
Wk 6	Full integration + verification tests	Everyone

2.7 Risk Analysis

The most critical failure mode for the Focus Dial is wireless communication loss (Bluetooth or Wi-Fi disconnection). If the dial loses connection with the paired smartphone, it risks creating a ”false positive” where the physical dial indicates focus mode is active, but the user's phone is actually allowing distracting notifications - or conversely, the phone is stuck in Do Not Disturb after the user tried to turn it off. The safe fallback behavior requires a Fail-Safe UI Indication. The device must detect the dropped connection via missed heartbeat packets, immediately change the LED ring to a distinct warning state (ex. pulsing amber), and update the OLED to notify the user. It should locally cache the intended state and automatically attempt to reconnect, ensuring that the physical interface never misrepresents the phone's actual status.

3 Ethics and Safety

Discuss societal impact, applicable standards, ethics, safety concerns, and mitigation procedures.

3.1 Societal Impact

The Focus Dial makes a meaningful contribution to public health, safety, and welfare by addressing the modern crisis of digital interaction.

- Public Health and Social/Cultural Factors: Constant digital interruptions contribute to cognitive fatigue, anxiety, and decreased attention spans. By providing a low-friction, tactile way to manage focus, the Focus Dial promotes healthier digital habits and better mental well-being. It shifts the culture of interaction from "always-on" to "intentionally engaged".
- Economic Factors: Distraction costs the global economy billions annually in lost productivity. A standalone, affordable hardware tool that seamlessly improves user efficiency has strong market viability and potential economic benefits for its users.
- Environmental Factors: Unlike cloud-dependent voice assistants that require massive data centers to process simple commands, the Focus Dial prioritizes local network communications (Bluetooth, Wi-Fi). This edge-computing approach reduces continuous data transmission, saving energy.
- Global Factors: The physical, tactile nature of a rotary dial transcends language barriers, making it an accessible solution globally compared to text-heavy software interfaces or voice assistants that struggle with accents.

3.2 Applicable Engineering Standards

- IEEE 802.11 (Wi-Fi): Governs the local wireless networking protocols used by the ESP32 to communicate with IoT devices (MQTT/HTTP).
- IEEE 802.15.1 (Bluetooth): Standardizes the Bluetooth communication used for pairing the dial with smartphones or computers.
- FCC Part 15 (Subpart C): Regulates intentional radiators (like your Wi-Fi/Bluetooth antennas) to ensure they do not cause harmful electromagnetic interference (EMI) to other devices.
- RoHS (Restriction of Hazardous Substances): Ensures the electronic components and custom PCBs we source do not contain dangerous levels of lead, cadmium, or mercury.

3.3 Ethical Considerations

Our project closely aligns with the core tenets of the IEEE and ACM Codes of Ethics, particularly, regarding the protection of public welfare, privacy, and system reliability.

- Privacy and Data Protection (ACM 1.6/IEEE 1): A major ethical strength of your project is its commitment to privacy. By relying on local network protocols rather than cloud-based data harvesting, you respect the user's right to privacy. It is an ethical imperative to ensure that the companion app or local IoT server does not unnecessarily log or transmit behavioral data.
- Honest Claims and Reliability (IEEE 3/ACM 1.2): The system must accurately reflect its state. If the LED ring indicates that the user is in ”Focus Mode”, the device must have successfully triggered that mode on the user's phone. False positives could lead to users missing important messages, assuming that they were protected by the system.
- Critical Ethical Concern - Emergency Bypasses: Activating ”Do Not Disturb” (DND) inherently blocks incoming communications. An ethical concern is whether your physical dial allows for or respects DND overrides for emergency contacts (ex. medical alerts,

family emergencies). Your design must ensure that OS-level emergency bypass settings are not overridden or corrupted by the Focus Dial's Bluetooth commands.

3.4 Safety Considerations and Mitigation Procedures

While a low-voltage IoT controller is generally low-risk, integrating custom hardware, moving parts, and potentially a portable power source introduces specific safety concerns that require strict mitigation.

3.4.1 Electrical Safety Concerns

- Short Circuits: Exposed PCB traces or improper wiring near the metallic components of the rotary encoder could cause short circuits, leading to component failure or localized burning.
- Thermal Dissipation: The ESP32 and the WS2812B LED ring can draw significant current and generate heat. Improper enclosure design could trap heat, melting plastics or burning the user.

3.4.2 Mechanical Safety Concerns

- Pinch Points: The clearance between the rotary dial knob and the base enclosure could pinch fingers or catch hair/jewelry during rapid turning.

3.4.3 Documented Mitigation Procedures

- Thermal Mitigation: Limit the maximum brightness of the LED ring in firmware to restrict the current draw. Ensure that the enclsoure features passive ventilation slots near the ESP32 antenna and LED ring to allow the heat to escape.
- Mechanical Mitigation: The enclosure will be designed with a tight tolerance (less than 1mm) between the rotary knob and the base to eliminate pinch points.

References

[1] IEEE, “IEEE Code of Ethics,” 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html