



UNIVERSITY OF
ILLINOIS
URBANA - CHAMPAIGN

Focus Dial

Electrical & Computer Engineering

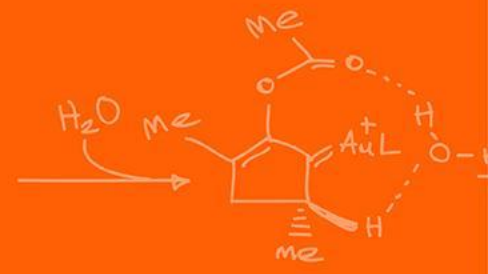
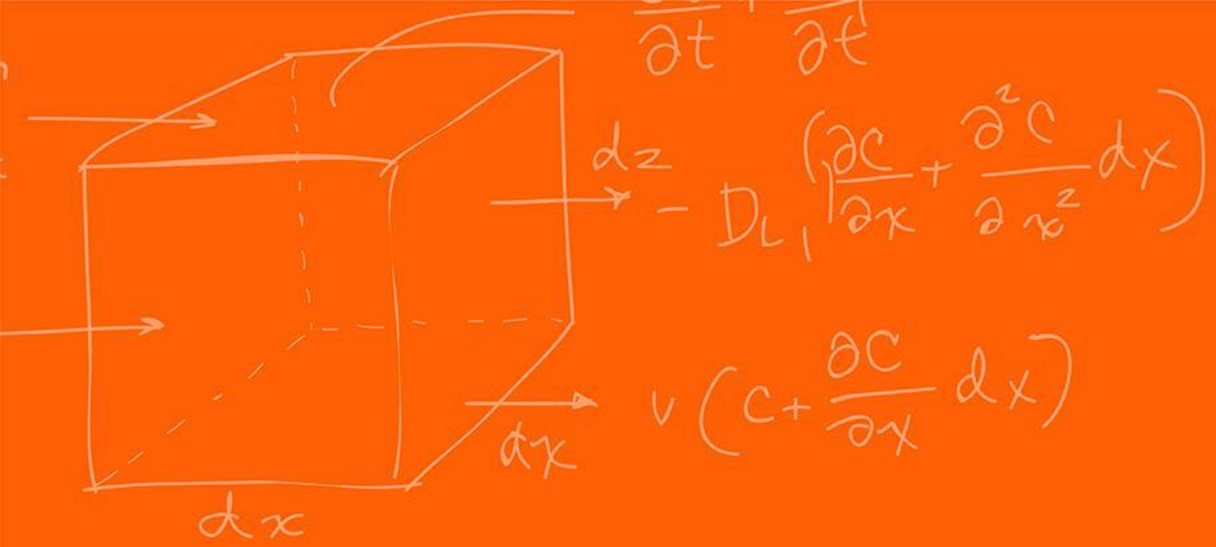
ECE 445: Team 45

Ahan Goel | Amogh Mehta | Benjamin Loo

May 1, 2026

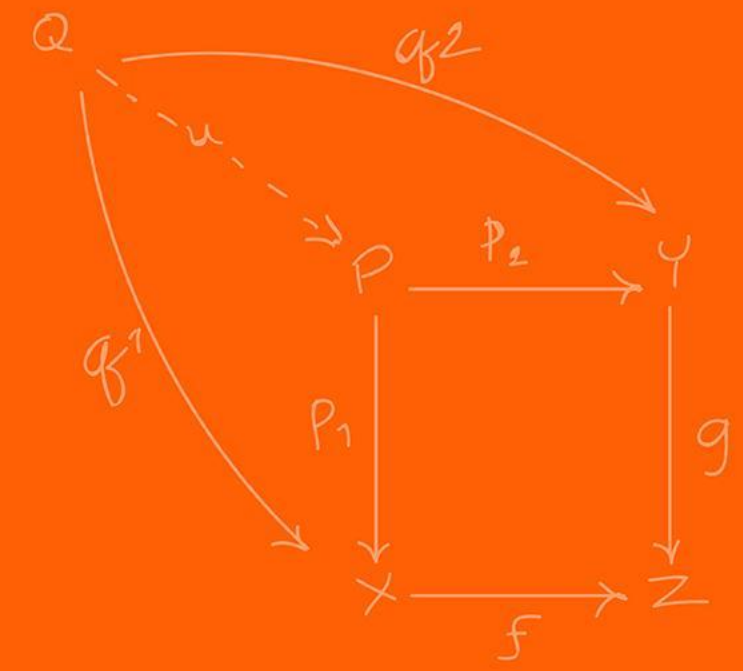
- **Digital devices are the primary source of distraction during work and study**
- **Existing focus tools (apps) require interacting with those same devices to activate them.**
- **This interaction inherently raises the risk of distraction.**

- **Develop a standalone, low-friction hardware interface for focus control.**
- **Accurately translate rotary input into time-based focus sessions.**
- **Enable seamless wireless activation of device-level focus modes.**
- **Provide clear, multimodal feedback of system state.**
- **Support local, privacy-preserving operation and IoT integration.**

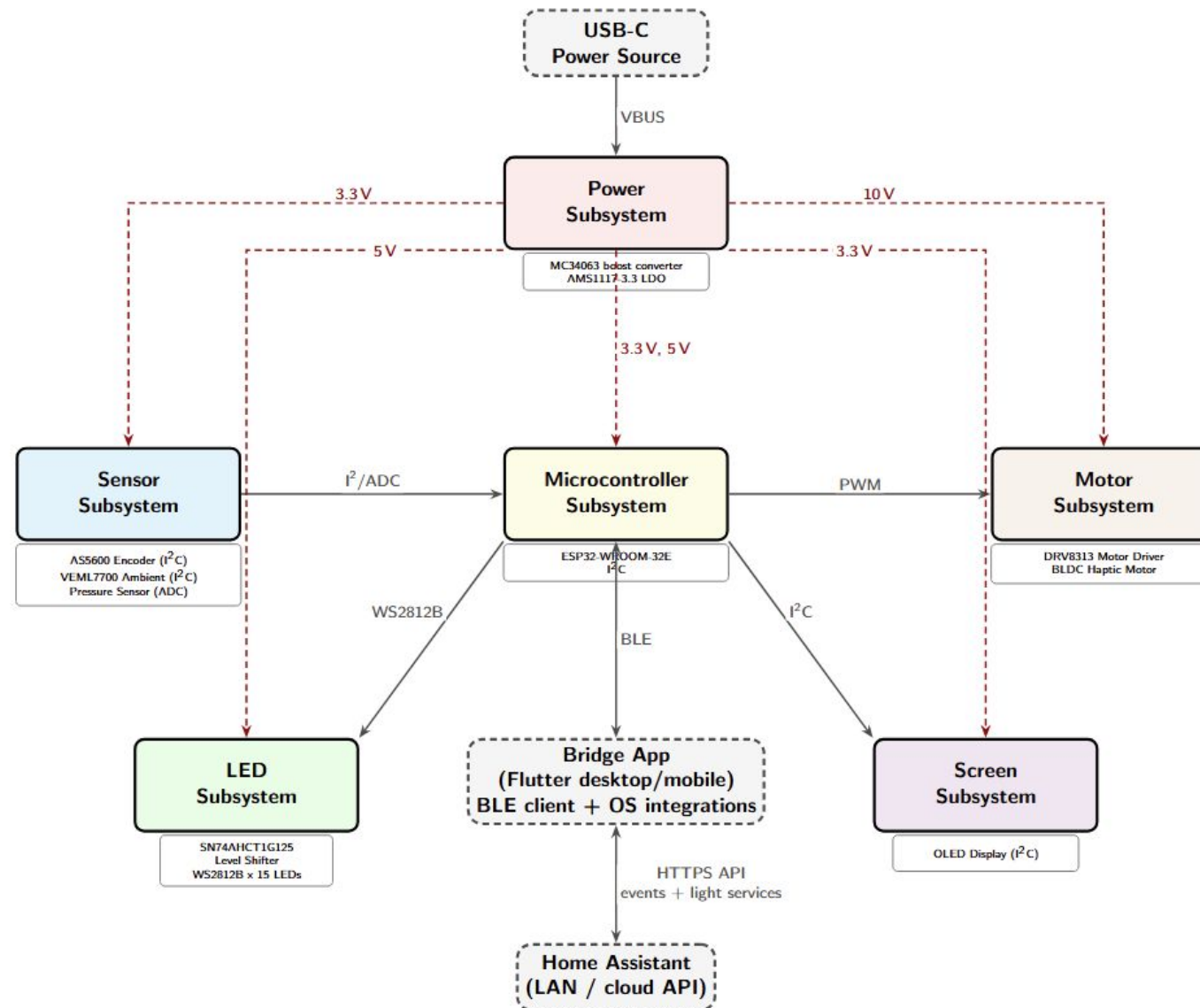


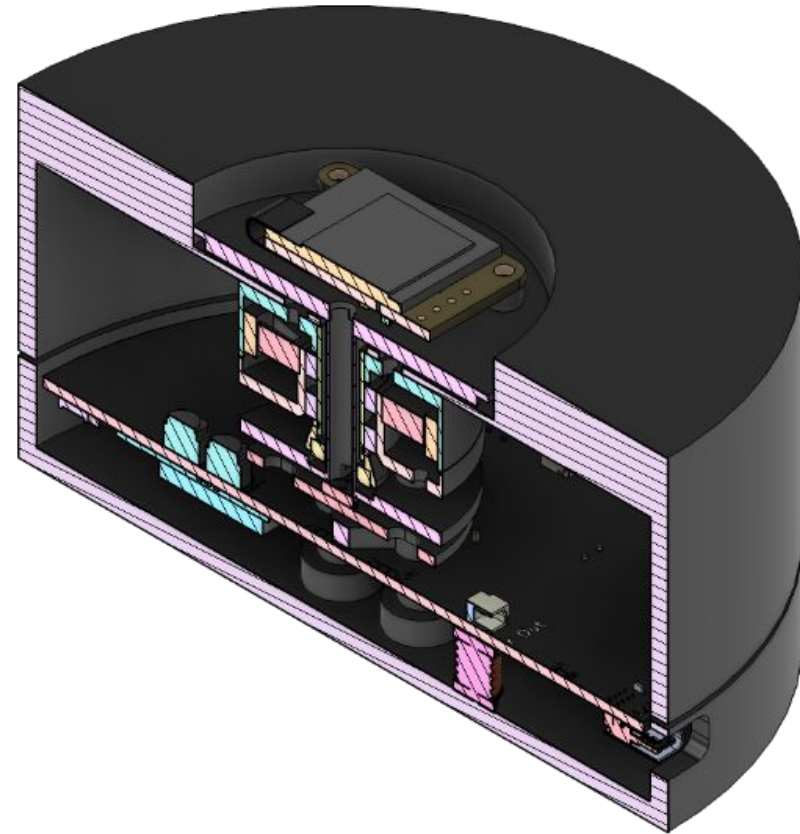
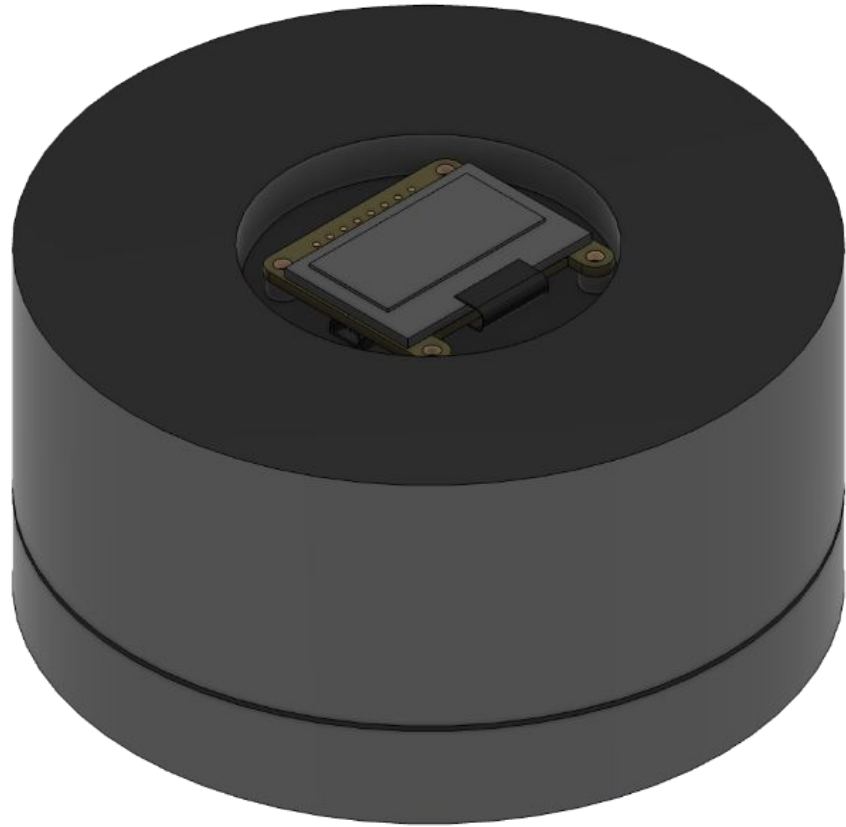
Block Diagram

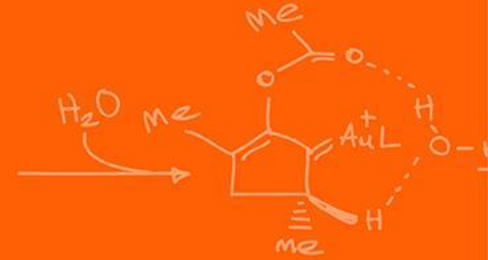
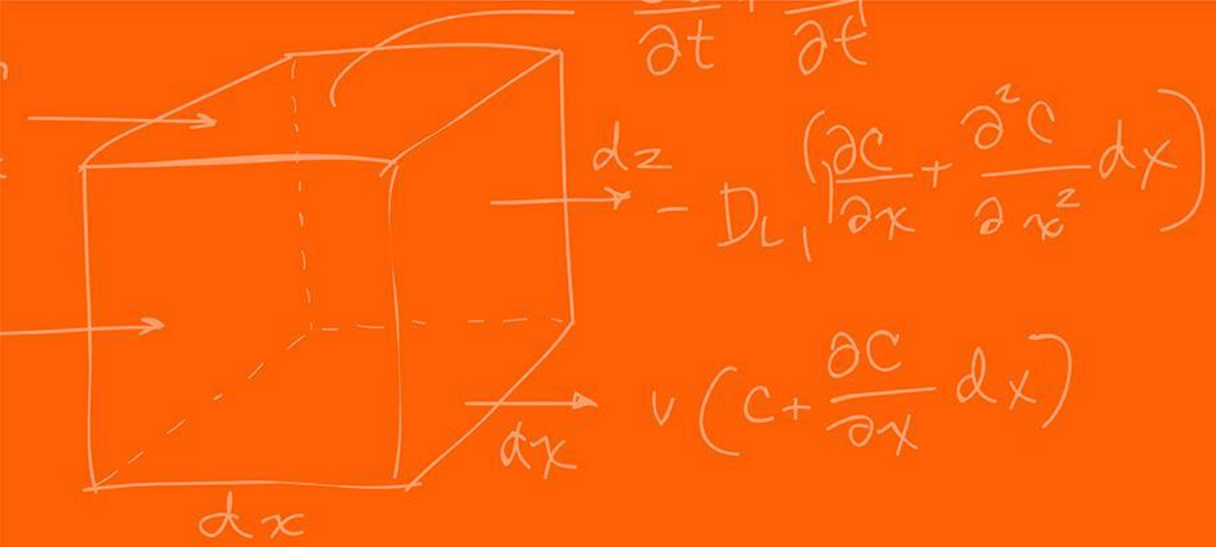
Overview of subsystems



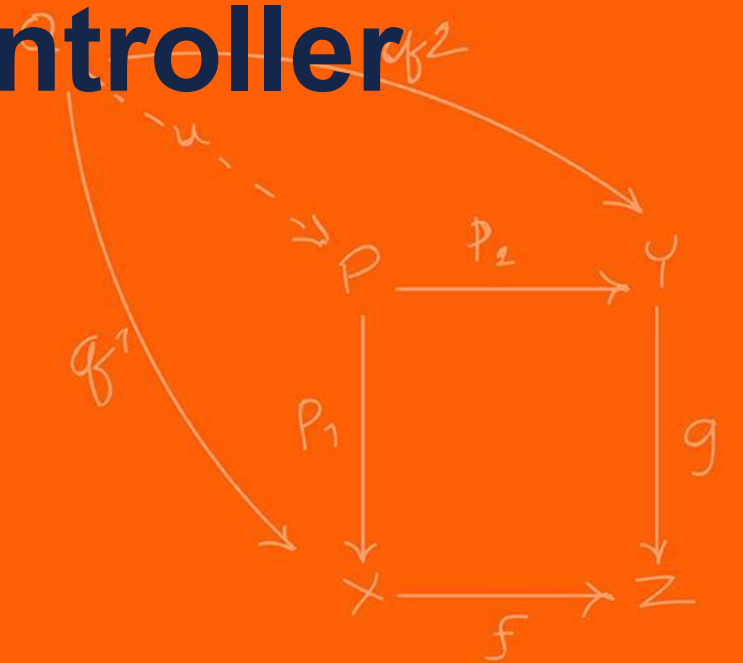
Block Diagram



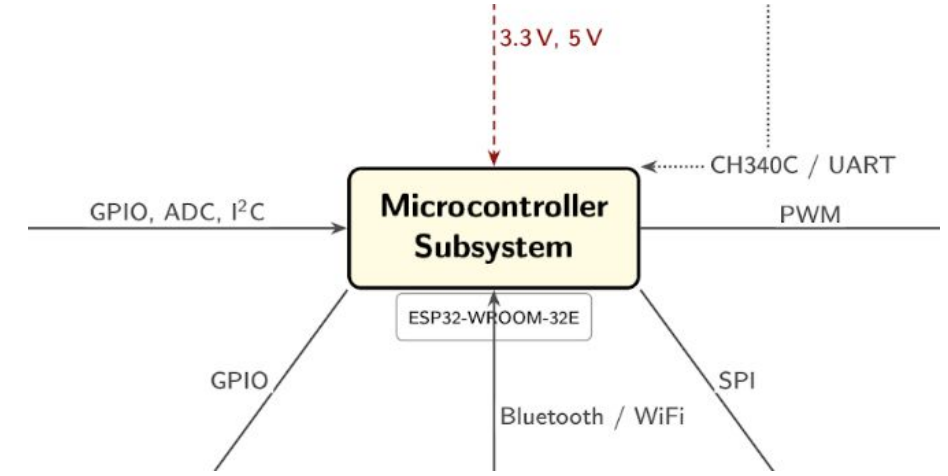
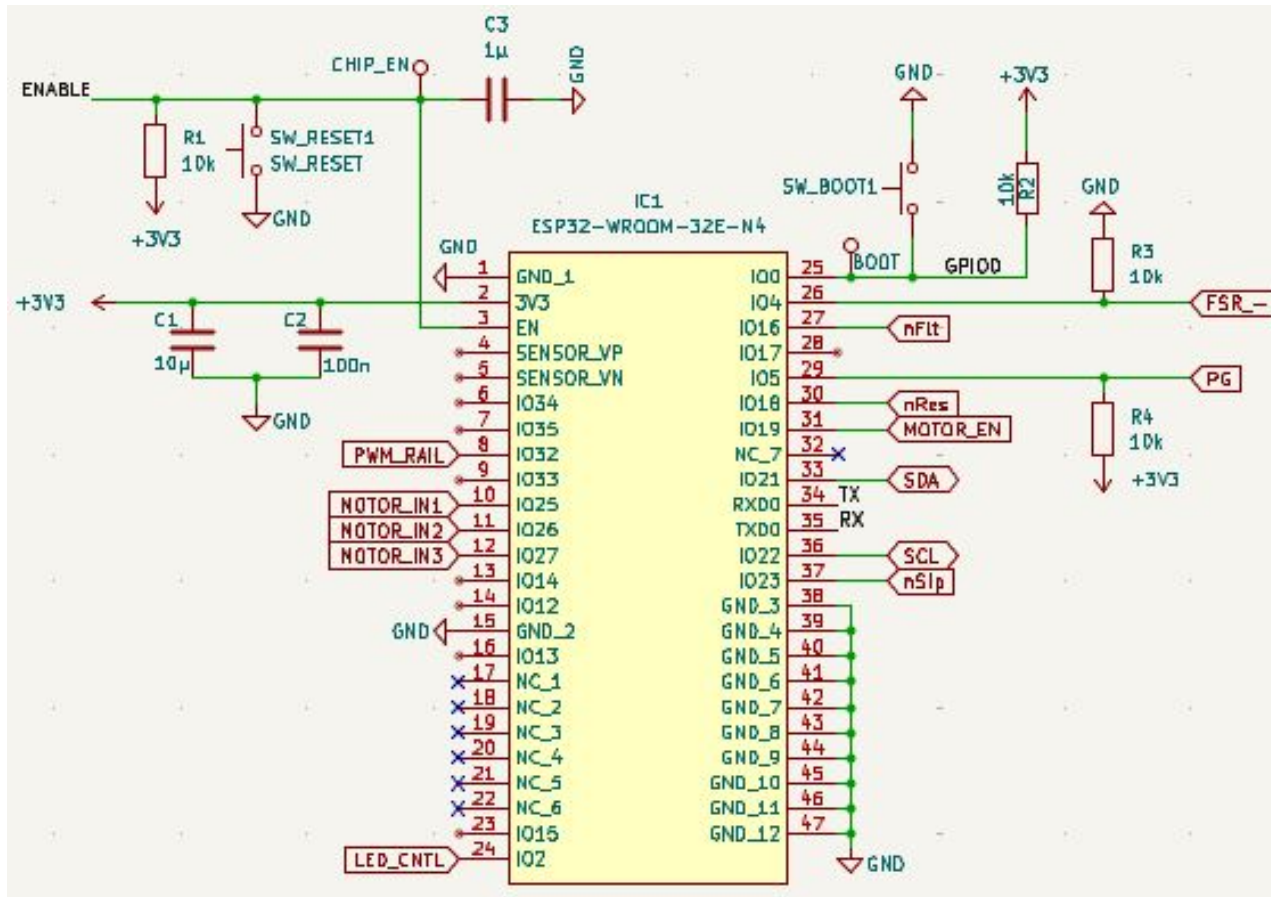




Subsystem #1: Microcontroller



Subsystem #1: Overview

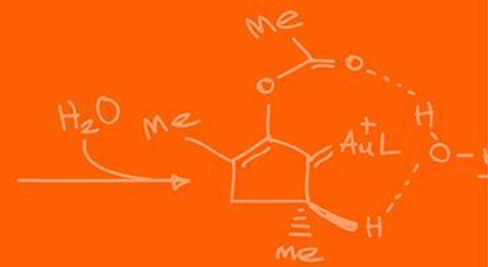
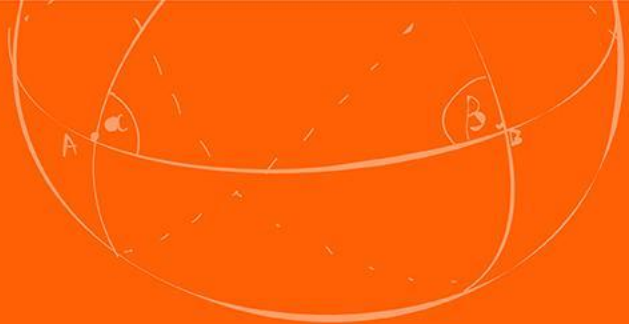
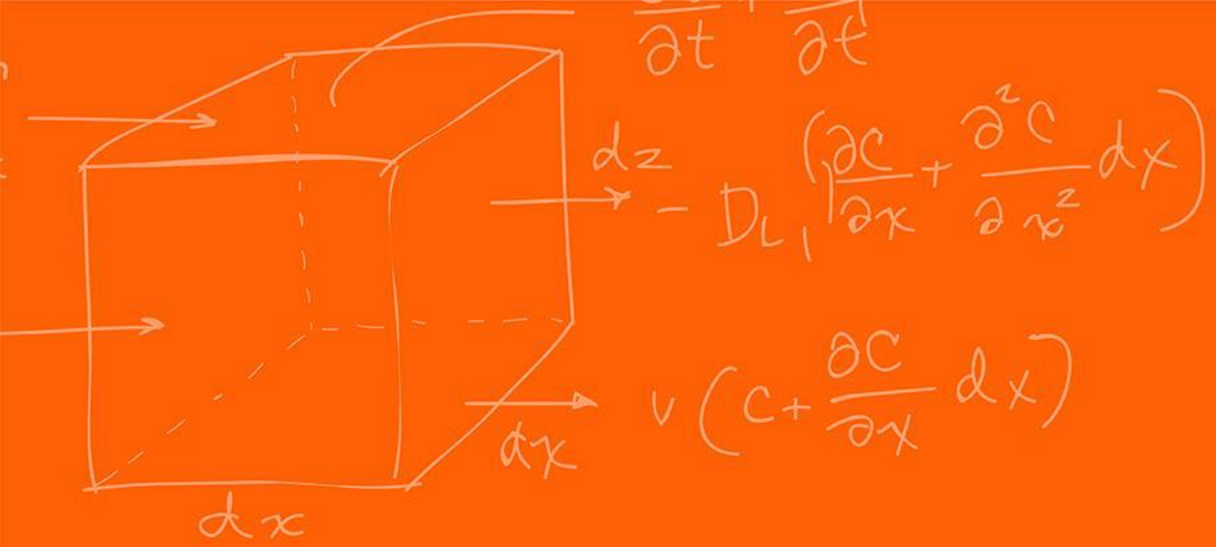


Requirements

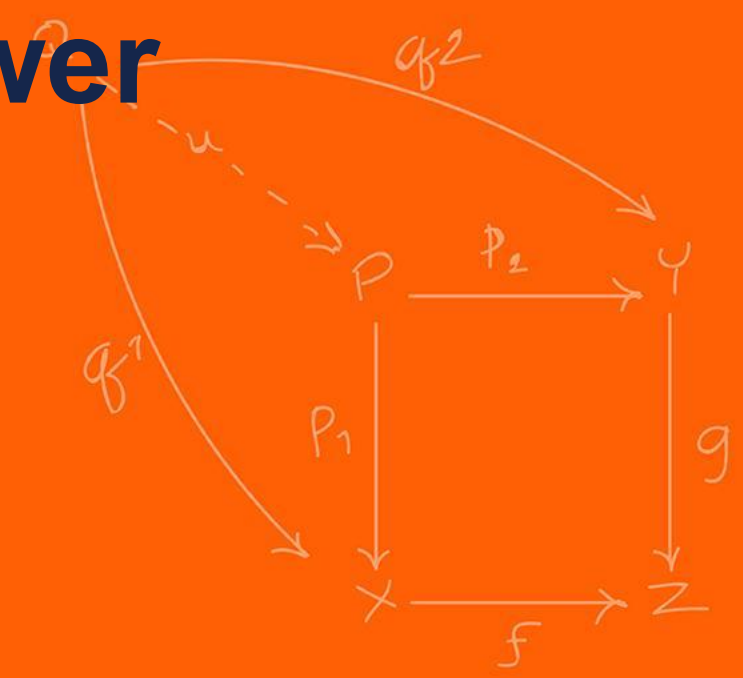
- Must transmit activation/deactivation commands to all connected devices.
- Must update the OLED display and LED light ring with the appropriate information.

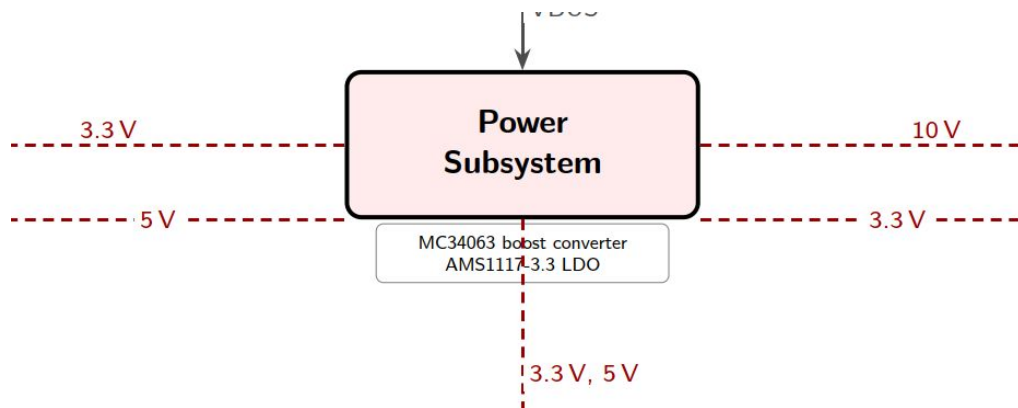
Verifications

- Confirm that all devices that are connected to the Focus Dial are synchronized in terms of state with the Focus Dial within 1 second.
- Verify that the OLED display and LED light ring are updated within 0.1 second.



Subsystem #2: Power





Functionality:

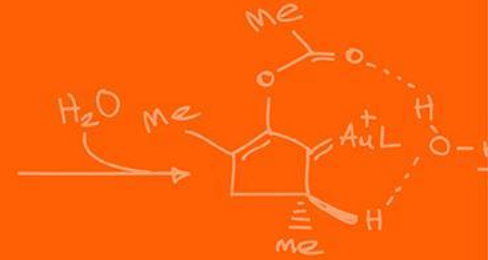
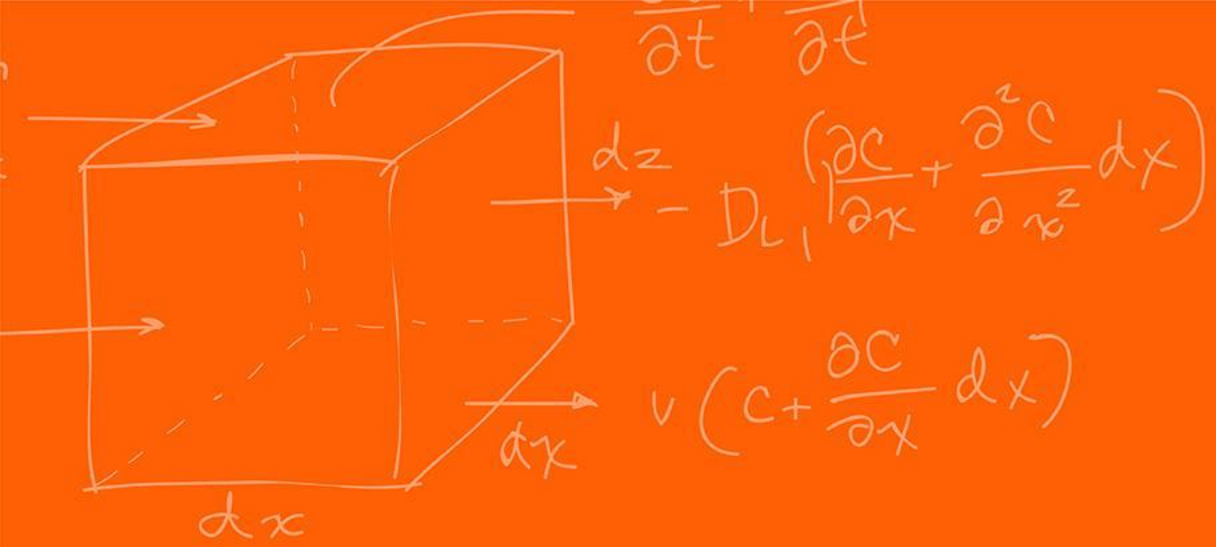
- Generates low-voltage for the ESP32 and associated systems and high-voltage for the motor.
- Accommodates safety features such as overrotation on the motor.
- Ensures protection against voltage fluctuations, stable logic operation, and minimal latency due to brownout or resets

Requirements

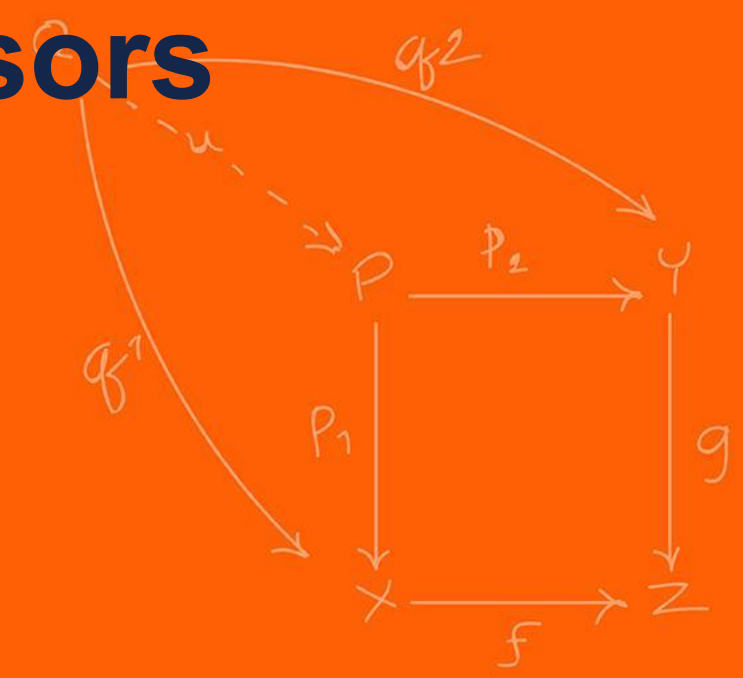
- When a USB-C power source is connected, the VBUS rail shall successfully negotiate and stabilize at the appropriate operating voltage.
- The boost converter shall regulate and supply the required high voltage to the motor.
- The LDO regulator shall provide a stable low-voltage supply to the ESP32 microcontroller.

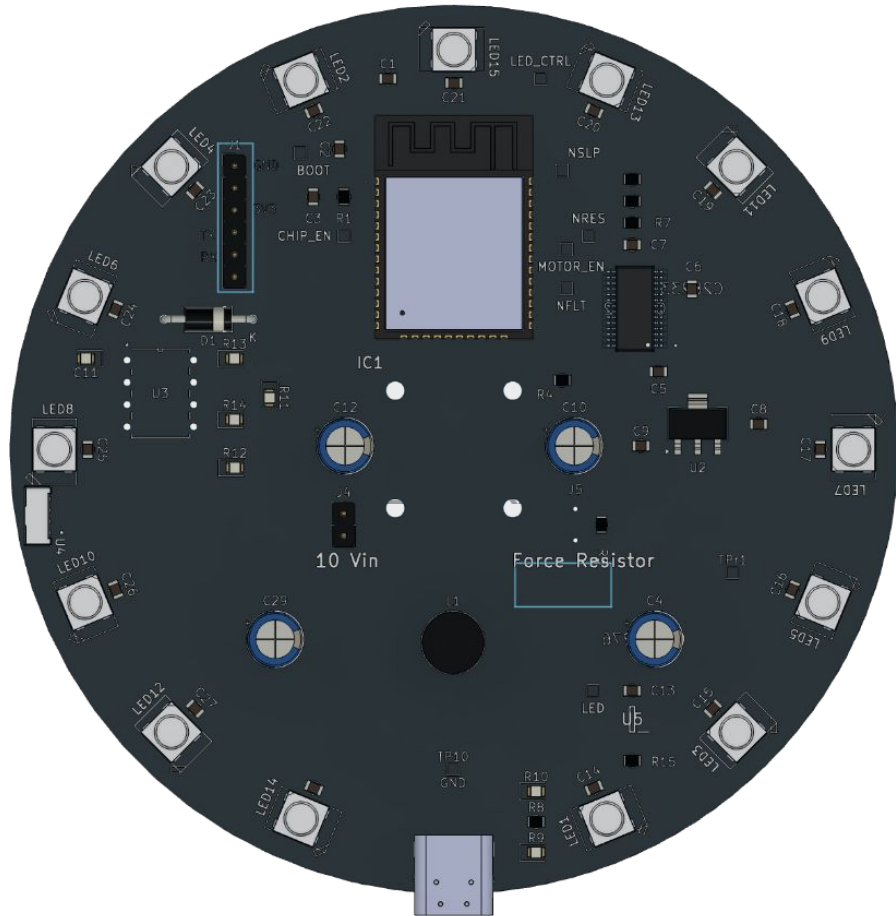
Verification Procedure

- Connect a USB-C power source and measure voltage across any NeoPixel 5V pad and TP1 (GND).
- Repeat for 3.3V and 10V voltage rails.
- Measure the time required for VBUS to achieve a steady-state voltage (should be less than 1 second).



Subsystem #3: Sensors

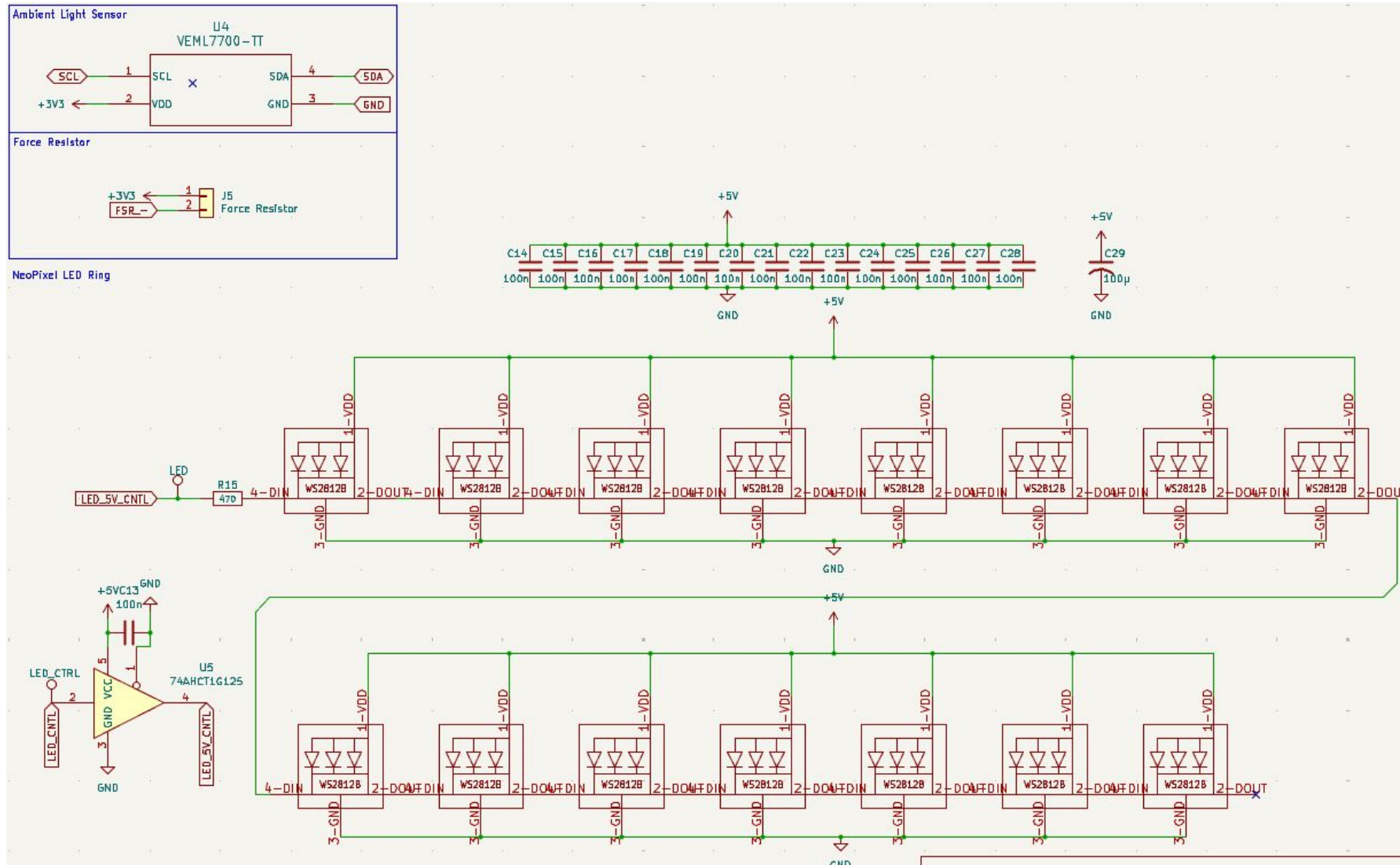




Functionality:

- **Pressure Sensor:** Controls device state via a haptic “press”
- **Motor Encoder:** Tracks rotational position of the BLDC motor to provide tactile feedback
 - Tactile feedback via counter rotation
- **Ambient Light Sensor:** Adjusts LED brightness to scale with the room’s light level

Subsystem #3: Overview

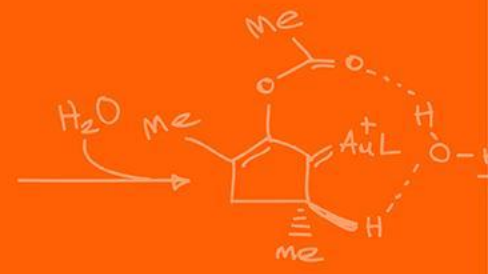
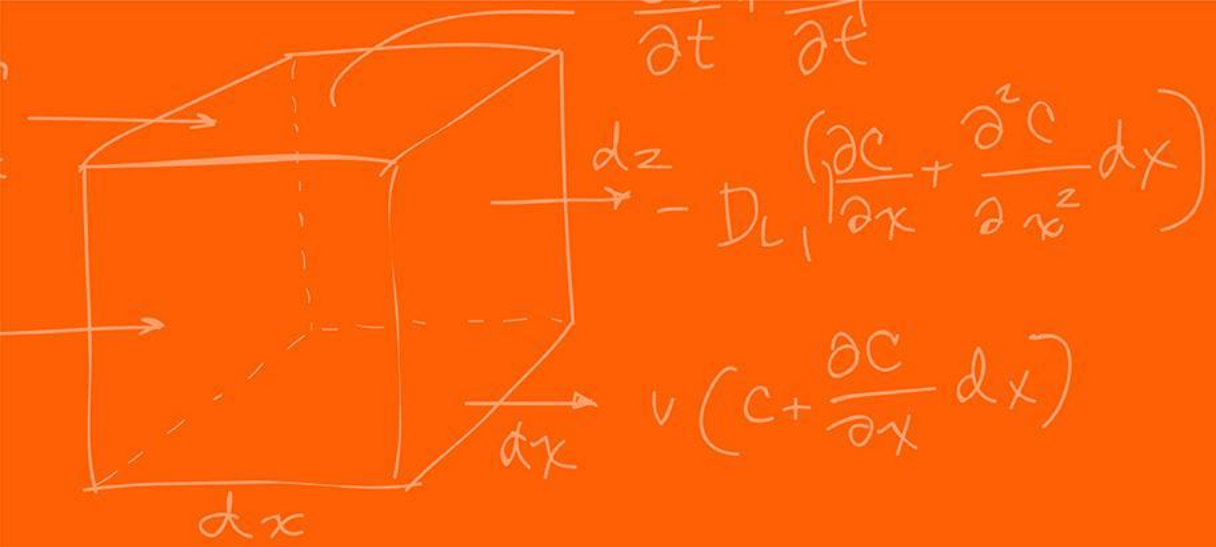


Requirements

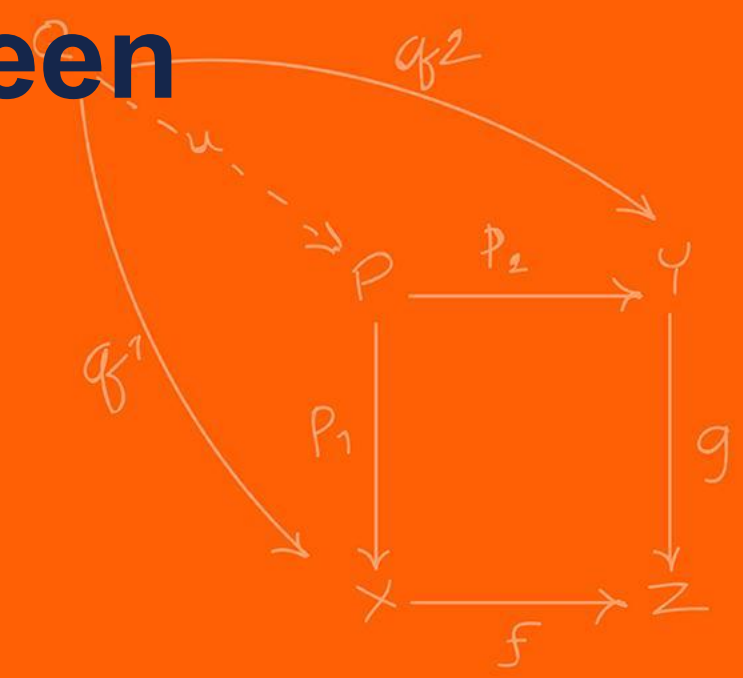
- Use pressure sensor input to reliably detect user interaction and map it to corresponding device state changes.
- Utilize motor encoder feedback to determine motor position and generate appropriate counter-torque for haptic feedback
- Adjust NeoPixel LED brightness based on ambient light sensor readings to maintain appropriate visibility.

Verifications

- Apply pressure to the sensor and confirm that the system transitions states with at least 95% accuracy over repeated trials.
- Verify using raw encoder data that the motor produces consistent counterforce behavior across multiple trials.
- Expose the device to different lighting conditions and confirm that LED brightness changes at device startup/device wake.



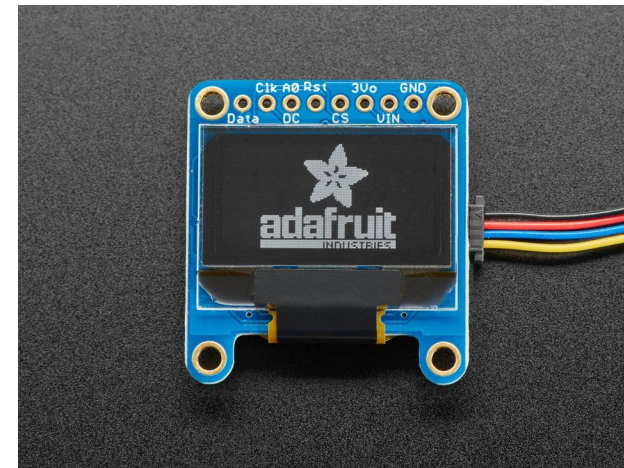
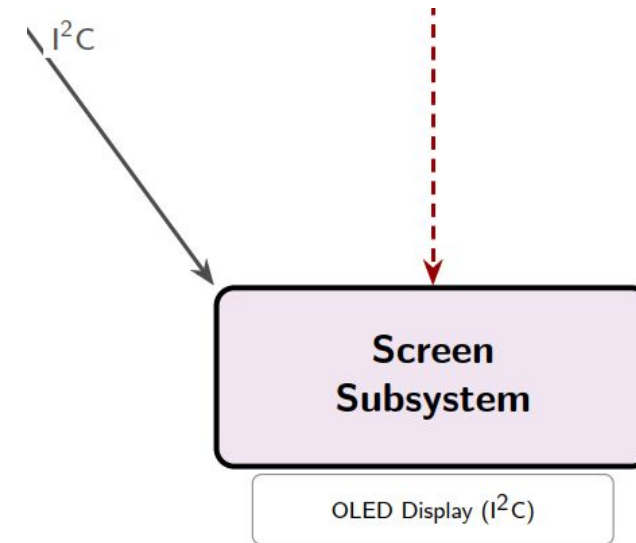
Subsystem #4: Screen



Subsystem #4: Overview



- Provides low-latency visual feedback to communicate system state independently of external devices.
- Displays timer duration, countdown progress, focus/break mode, pairing status, and error conditions.
- Acts as a primary state indicator, reinforcing user interactions and complementing tactile input through clear visual cues.

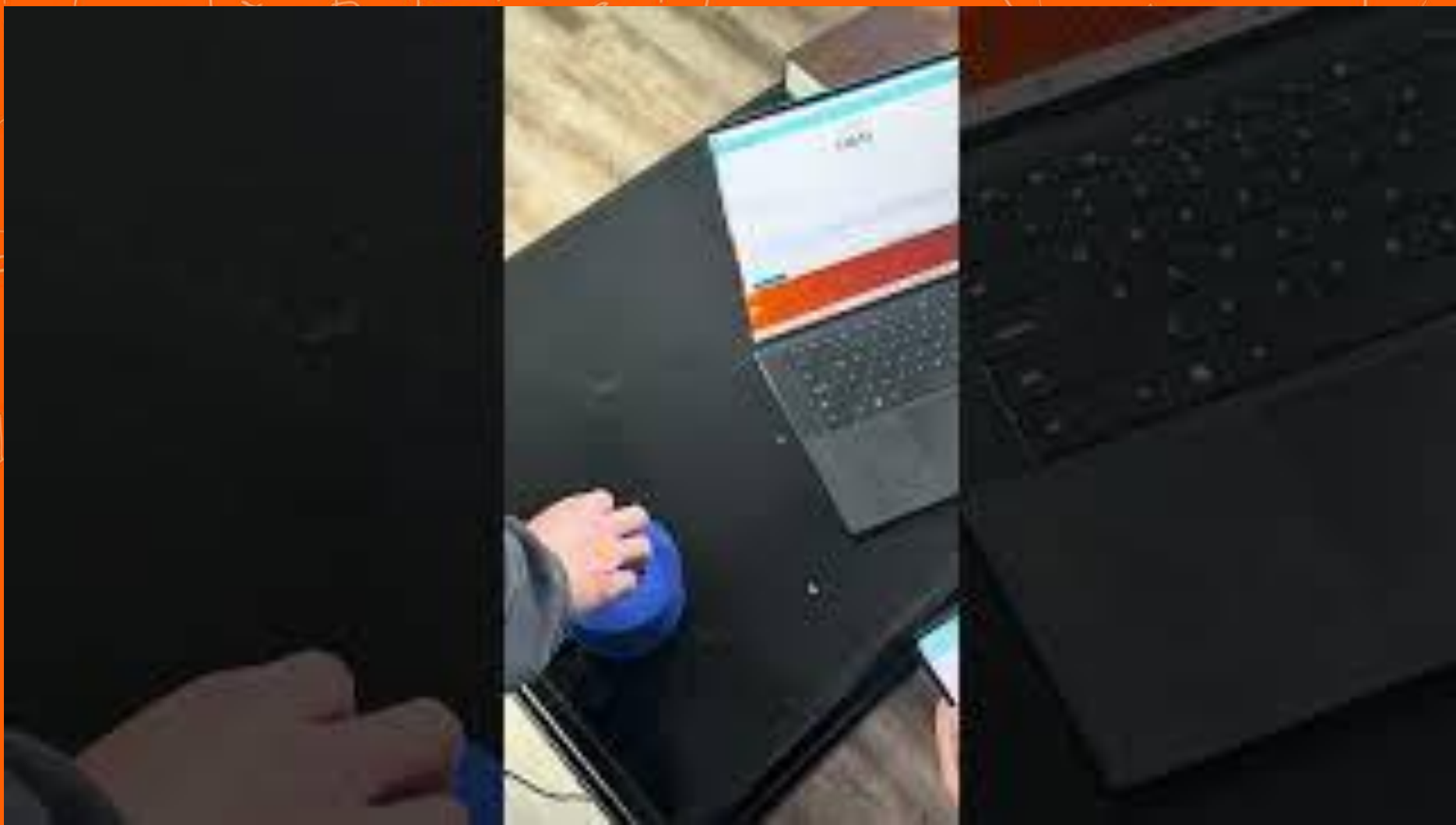


Requirements

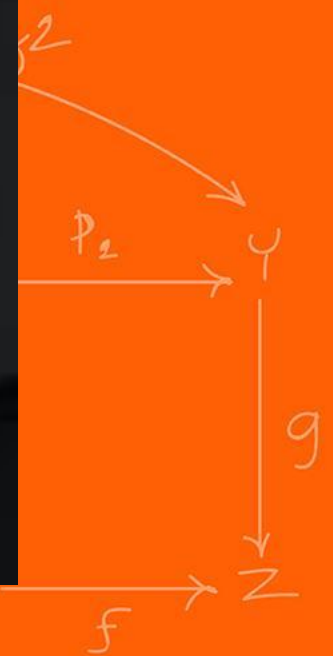
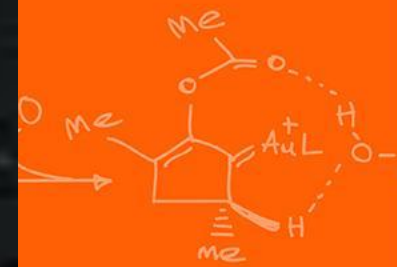
- Provide visual updates that reflect system state changes, independent of external device connectivity.
- Present accurate, real-time information including timer duration, countdown progress, operational mode (focus/break), and error conditions.

Verifications

- Trigger known system state changes (ex. timer start/stop) and measure the time between the event and the corresponding display update using a timestamped reference; confirm latency is below 100 ms.
- Induce each system state and verify that the displayed information matches the expected value with 100% correctness across test cases.



$$\frac{\partial z}{\partial t} = \left(\frac{\partial c}{\partial x} + \frac{\partial^2 c}{\partial x^2} \right) dx$$



Ethics:

- Promotes healthier digital habits and better mental well-being.
- Ensure that the companion app does not unnecessarily log or transmit behavioral data.
- Respects DND overrides for emergency contacts

What did you learn?

- PCB Design + verification + prototyping.
- Organizing a project from inception to delivery.
- Timelining a project + meeting deadlines and having deliverables
- Working with sensors, motors, and mechanical design outside of the EE/CE wheelhouse.
- Hardware + Software Integration.
- Power System Design.

What would you do differently if you redesigned your project?

- Condense power subsystem + increase voltage for maximum efficiency
- Plan overall design immediately.
 - Use some sort of slip ring to transmit signals + power, plan clicking mechanism better
- Smaller PCB is achievable
 - Embed a different MCU to simplify design (doable as the software aspect can compensate)
- Get luckier with part availability (encoder that could have been embedded was sold out)
- Use a different motor as the motor that was chosen was not strictly necessary by the end

1. Enable different configuration modes via the app
2. Add IoT connections for other external devices such as LEDs or speakers
3. Develop a low-power state for the 'Sleep Mode'.
4. Incorporate end-to-end encryption and multi-factor authentication to ensure secure communication between the app and hardware.
5. Size + Weight + Material can be adjusted for a final end-user product

Thank You

Questions?

Contact Information

Ahan Goel: ahang5@illinois.edu | Amogh Mehta: amoghm3@illinois.edu | Benjamin Loo: bloo2@illinois.edu