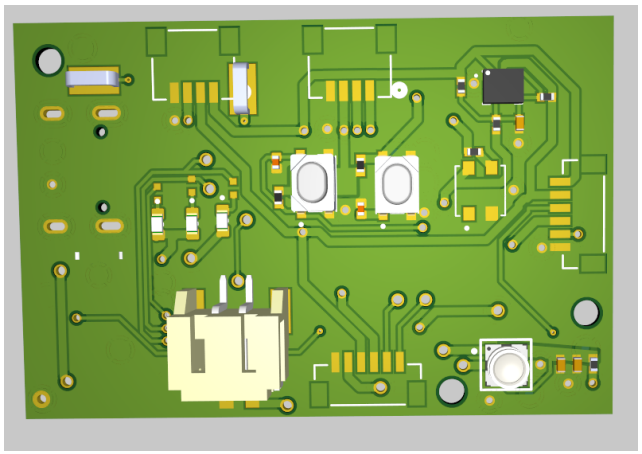
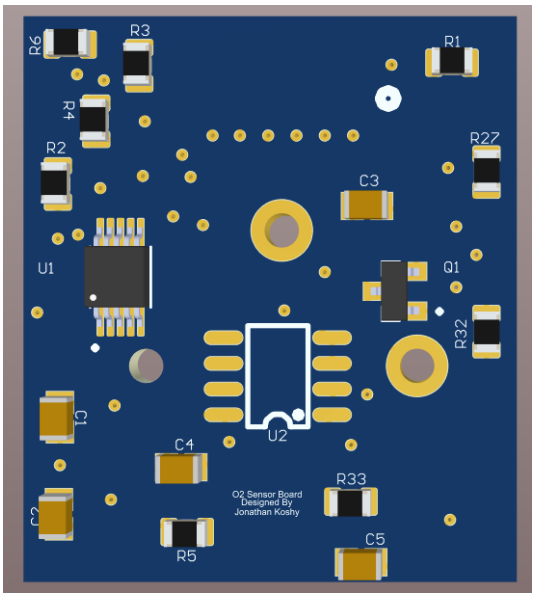
ESP32-S3 Wearable Health and Respiratory Monitoring System

(a) Wearable enclosure prototype, side view

(b) Wearable enclosure prototype, front airflow interface



(a) Custom ESP32-S3 sensor and power PCB



(b) Oxygen sensor PCB

Overview

Designed and prototyped a wearable embedded system for continuous cardiovascular and respiratory monitoring. The system integrates custom low noise PCBs around an ESP32-S3 to acquire synchronized biomedical and environmental sensor data and stream it in real time.

Hardware Design

- Designed and assembled multilayer PCBs integrating pressure, optical PPG, gas, and inertial sensors with the ESP32-S3 in a compact wearable form factor.
- Optimized analog layout and decoupling to reduce noise and improve wireless signal integrity while reducing board weight by twenty percent.
- Implemented USB C power management with ESD protection and verified manufacturability using

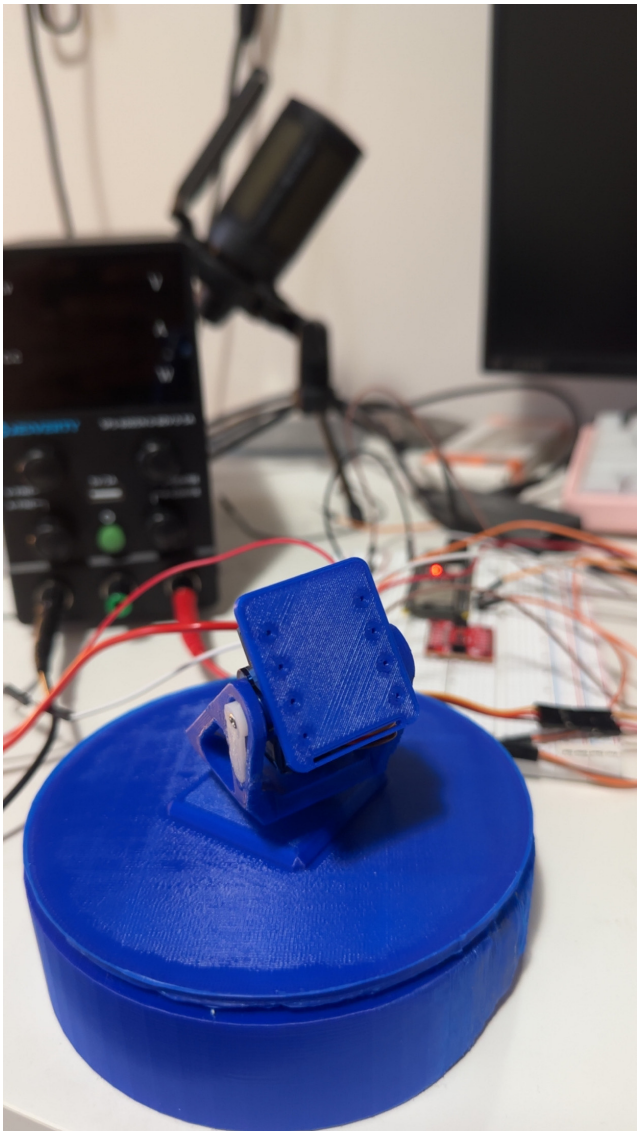
DFM checks.

Firmware and Data Pipeline

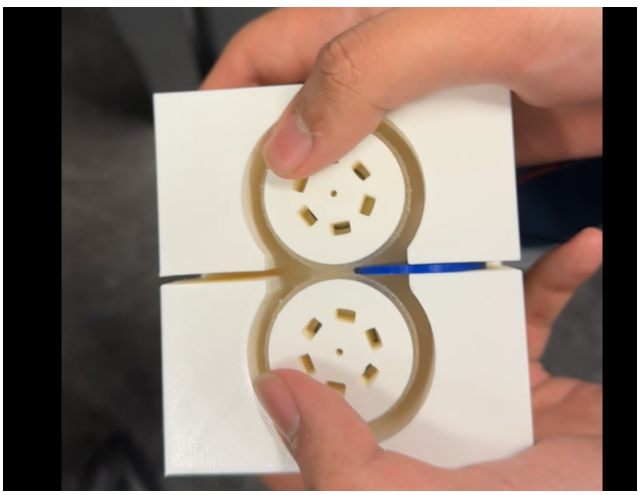
- Developed multi sensor firmware in FreeRTOS using ESP-IDF to poll I^2C and UART devices and stream synchronized data over BLE with end to end latency under fifty milliseconds.
- Implemented power efficient dual core task scheduling on the ESP32-S3 to support continuous high rate signal acquisition without packet loss.
- Calibrated cardiac, respiratory, and gas sensors against lab instruments to keep measurements within plus or minus two percent of reference values.

Vision-Guided Autonomous Disk Launcher

GitHub



(a) Pan tilt mechanism prototype



(b) Dual flywheel disk launch mechanism

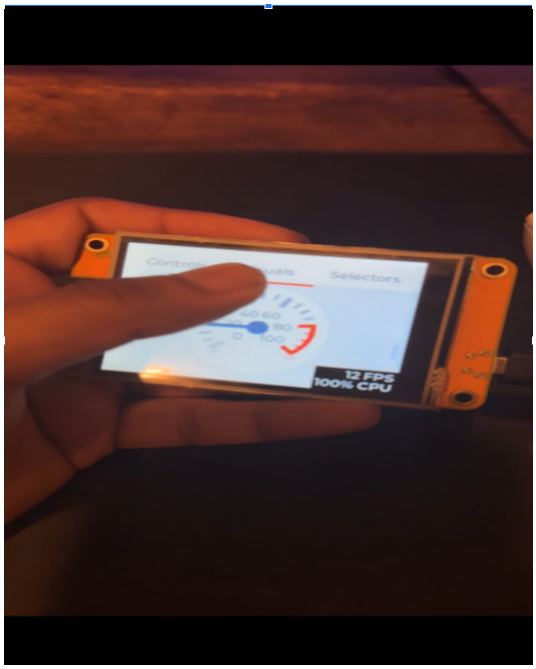
Overview

Built an autonomous vision guided launcher that detects, tracks, and engages targets using a Raspberry Pi for perception and an ESP32-S3 for real time motion and firing control.

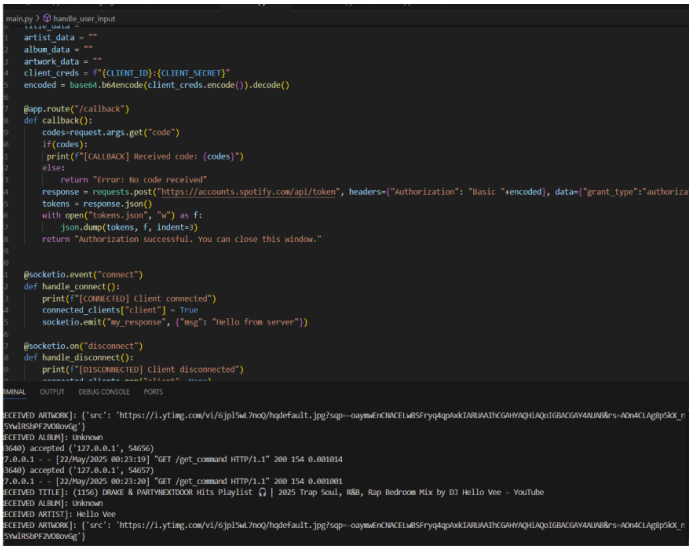
System Architecture

- Ran object detection on a Raspberry Pi and transmitted target offsets to the ESP32-S3 over a lightweight serial protocol.
- Converted bounding box positions into angular setpoints and applied closed loop pan and tilt control using custom PID logic.
- Integrated TF Luna LiDAR to validate target distance and gate the firing sequence based on range.
- Drove dual brushless DC motors and flywheels with sequenced spin up and release timing to launch disks consistently.

ESP32-Based Embedded Dashboard for Media and System Monitoring GitHub



(a) ESP32 touch display interface



(b) Python backend snapshot feed

Overview

Designed and built a real time embedded dashboard on an ESP32 with a 480x320 touchscreen display. The system runs a multitasking FreeRTOS environment and communicates with a Python backend over Wi Fi to visualize system metrics, control media playback, and display live Discord call data.

Embedded Architecture

- Implemented a multi task FreeRTOS design separating UI rendering, socket communication, system monitoring, and serial debugging.
- Pinned LVGL rendering to a dedicated core to maintain responsive touch interaction under high update rates.
- Used static allocation and bounded queues to avoid heap fragmentation and stack overflow.

UI and Graphics Pipeline

- Built a dark themed, modular UI using LVGL 9.4 with XML based layouts and reusable global styles.
- Implemented base64 JPEG decoding into RGB565 buffers for direct rendering with minimal memory overhead.

- Added animated progress bars, media transitions, and dynamic widgets optimized for a 480x320 display.

Backend Communication and Protocol

- Designed a bidirectional TCP socket protocol using newline delimited JSON for robustness and extensibility.
- Implemented buffered message queues to batch updates and prevent UI thread blocking.
- Added UART fallback for debugging and recovery when Wi Fi bandwidth was constrained.

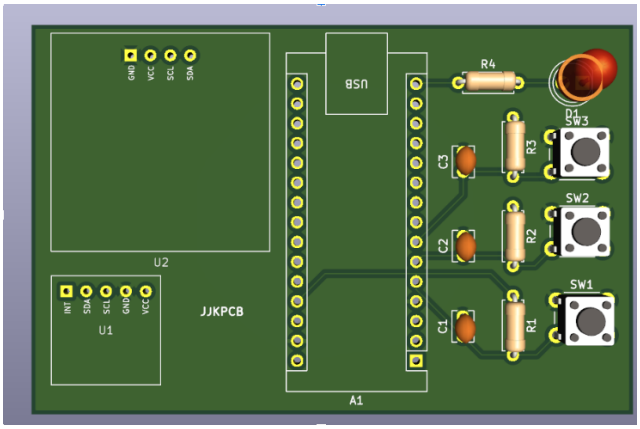
System Monitoring and Media Control

- Displayed live CPU usage, memory utilization, idle percentage, and top processes collected via a Python backend.
- Enabled touchscreen control of Spotify and YouTube playback including play, pause, seek, and queue navigation.
- Integrated Discord voice metadata including active users, mute state, and profile images.

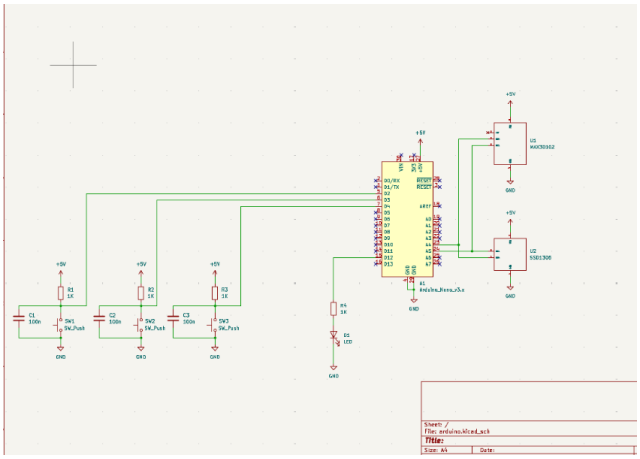
Engineering Challenges

- Resolved LCD color inversion and rendering artifacts through display calibration and pixel format correction.
- Reduced UI latency by throttling snapshot updates and applying differential screen refreshes.
- Optimized JPEG decode paths to reuse buffers and minimize dynamic allocation.

Heartbeat Monitor PCB Design



(a) PCB render



(b) Circuit schematic

- Designed a custom PCB around the MAX30102 for heart rate and oxygen saturation sensing.
- Implemented I^2C communication with a microcontroller to compute and display live BPM and SpO_2.
- Optimized component placement and decoupling to reduce noise and improve signal quality.