Lab 1: Individual Part Task 1

Ubiquitous Computing

Tiemor Amjad

Date: 06.11.2025

1. Task 1: Enhanced Posture Detector and Environmental Alarm System

Task 1 required a significant improvement to the basic Posture Detector developed in Exercise 4 by incorporating noise level detection, expanding posture analysis to include Roll and Yaw, and implementing a complex, timed alarm system.

1.1. Implementation Steps and Methodology

To achieve the extended functionality, the following steps were taken:

Step 1: Integrated Noise Threshold Detection (Microphone)

- **Objective:** Utilize the PDM microphone to detect if the ambient noise level exceeds a threshold unsuitable for studying.
- **Steps Followed:** The PDM setup from Exercise 3 was adapted. Instead of simply plotting the raw samples, the `loop()` function was modified to calculate the **Root Mean Square (RMS)** value over the collected audio samples. The RMS value provides a more stable representation of the average audio power compared to checking peak amplitude.
- **Threshold:** After empirical testing, a stable threshold of **RMS > 1500** was set. If the RMS exceeded this, the `is_noise_too_high` state flag was set to `true`.

Step 2: Expanded Posture Analysis (Roll and Yaw)

- Objective: Monitor posture not just for forward/backward leaning (Pitch), but also for lateral tilting (Roll) and twisting (Yaw).
- Steps Followed: The existing Madgwick filter implementation was sufficient, but the detection logic in the main loop was expanded.
    - Pitch: Alarm triggered Pitch |Pitch- (-90°)| > 10°
    - Roll: Alarm triggered if Roll > 15° (significant side-to-side lean).
    - Yaw: Alarm triggered if Yaw > 25° (excessive twisting motion).
- Logic: A unified `is_posture_bad` flag was used to combine any of the three axis violations.

Step 3: Complex Timed Alarm System

- **Objective:** Implement a state machine to manage the alarm, including blinking red/blue LEDs for different failure modes and a 10-second cool-down period before switching back to the "Green" (OK) state.
- **Implementation:** A `millis()`-based timing system was used instead of `delay()`. This **non-blocking approach** allows the sensors to continue reading and the posture/noise checks to run even while the alarm is blinking or in the cool-down phase.
    - **Alarm States:**
        - Blinking Red: `is_noise_too_high == true`.

        ■ Blinking Blue: `is_posture_bad == true`.
        ■ Blinking Yellow: Both conditions are true (Red + Green = Yellow).
    ○ Cool-Down Logic: If both `is_noise_too_high` and `is_posture_bad` become false, a timestamp (`last_alarm_time`) is recorded. The system remains in a blinking state for 10 seconds after this timestamp before transitioning to the solid Green LED.

1.2. Challenges Encountered and Resolution

Challenge: Noise RMS Calculation Stability

The raw PDM sample data is very noisy. Initial simple averaging resulted in highly fluctuating noise measurements and false alarms. The resolution involved switching from simple average to Root Mean Square (RMS) calculation over a buffer of 512 samples. This provided a far more stable and representative measure of the overall volume/power of the ambient sound, significantly reducing false positives.

Challenge: Non-Blocking Alarm Timing

Using the blocking `delay()` function for the alarm blinking in the initial draft prevented continuous sensor reading, leading to unresponsive posture detection during an alarm event. The resolution was to replace all delays in the alarm function with a non-blocking, `millis()`-based state management approach. This allowed the LED state to change periodically without freezing the main `loop()`, ensuring real-time monitoring.

Challenge: Madgwick Angle Drift

Over time, the estimated Roll, Pitch, and Yaw angles tended to drift slightly, especially Yaw, due to magnetic interference and slight gyroscope bias. This is inherent to IMU-based filters. The resolution involved accepting a larger tolerance band (e.g., 10° for Pitch) and focusing the detection on rapid changes and sustained bad posture, rather than absolute angular accuracy.

1.3. Further Improvement Ideas

The device represents a robust system, but its functionality can be expanded to fully realize its potential as an Ubiquitous Computing tool:

1. **Haptic Feedback (Actuator Integration):** Currently, the alarm is visual (LED). A significant improvement would be to add a small **vibration motor**. This provides discrete feedback that alerts only the user, making it ideal for a shared environment like a library or office.
2. **Wi-Fi Cloud Logging and Analytics:** Leveraging the Wi-Fi capability (Task 2), the device could upload hourly summaries of posture and noise data to a cloud database (e.g., Firestore). This would enable the user to track long-term trends, identify peak times for bad posture/noise, and receive analytical reports, turning raw data into actionable insights.

3. **Machine Learning Core Utilization (MLC):** The Nano RP2040's IMU includes an MLC. Instead of using fixed geometric thresholds (Roll/Pitch/Yaw), the device could be trained to **recognize specific *patterns* of bad posture** (e.g., slouching versus leaning) using the MLC, freeing up the main microcontroller for other tasks.

Conclusion

The individual task expanded the fundamental skills acquired in the lab. Task 1 led to the development of a sophisticated, non-blocking environmental monitor that simultaneously analyzes noise levels and complex tri-axial posture using sensor fusion, integrating a timed alarm system.