Exercise 3: Event & State Machine — Security System

Objectives

- Introduce real-time state machines
- Handle buttons, LEDs, and files
- Structure a robust real-time application

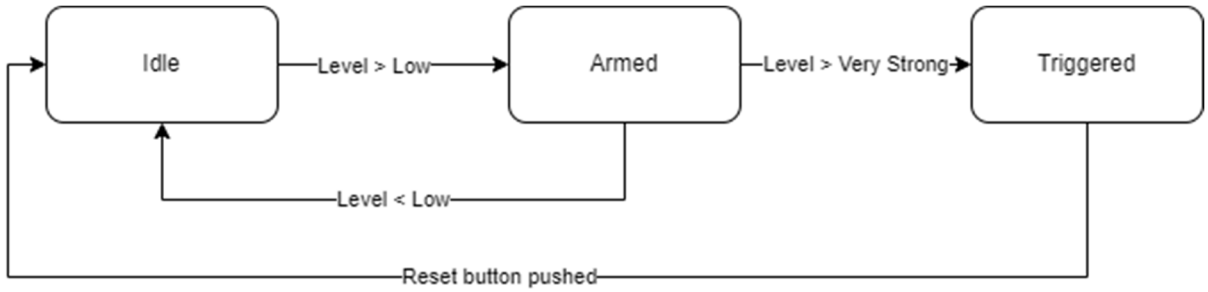
Tasks

1. **Implement a Real-Time State Machine**

Your RT application must implement the following states:

- Idle
- Armed
- Triggered

Use a type-def enum for clean architecture.



2. **Read and Process the Accelerometer**

✓ Read the x, y, and z accelerometer values.
✓ Compute a global vibration magnitude, e.g.:

$$Magnitude = \sqrt{x^2 + y^2 + z^2}$$

3. **Use this magnitude to drive a 4-LED bargraph:**

- Low magnitude → only LED 0 (inf. 2g)
- Medium → LEDs 0–1
- Strong → LEDs 0–2
- Very strong → LEDS 0–3

(Exact thresholds are up to you.)

4. **Intrusion Detection Logic**

When the system is in the Armed state:

- If the magnitude exceeds a chosen threshold →
  - ✓ Transition to Intrusion Detected
  - ✓ Turn on all the LED in solid state
  - ✓ Send a UDP packet to a predefined IP/Port (e.g. 192.168.1.10:5000)

- ✓ Packet content example:
  "INTRUSION DETECTED - <timestamp>"
- ✓ Append a line to a log file stored on the RT target (/home/lvuser/alerts.log)
  Format example: <timestamp>, INTRUSION, magnitude=<value>

5. **6. Reset the System**

A button (physical or front-panel interactive) must:

- ✓ Turn off all LEDs
- ✓ Return the state machine to Idle

Bonus (Optional Improvements)

- Add a 5-second arming countdown displayed on LEDs.
- Log the raw magnitude every second while armed.
- Add a moving average filter to reduce noise.
- Build a simple PC application to receive the UDP packets.