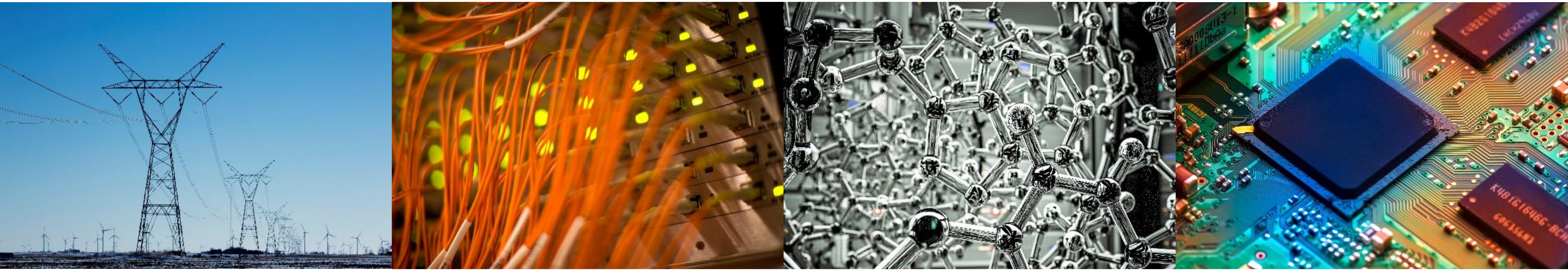
E-Bike Theft Detection System

Group 71: Kacper Bakun, John Hanley, Paul Harris
Professor: Craig Schultz
TA: Yulei Shen



ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

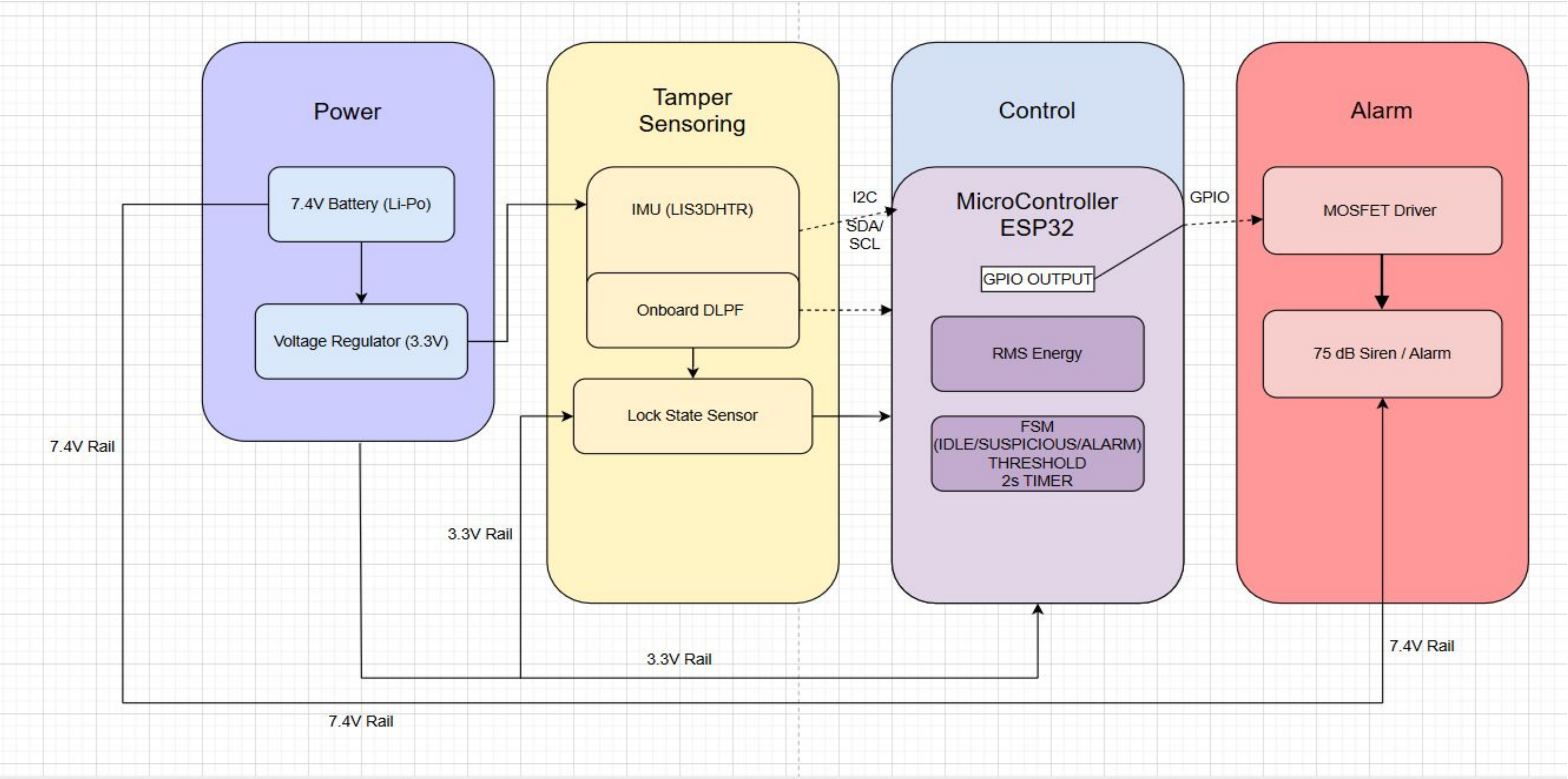
Introduction: Problem Statement

- **The Limitation of Physical Locks:** Traditional mechanical security only delays theft; it lacks the real-time deterrence necessary to stop an attempt once it begins.
- **Operational & Financial Damage:** Ride-sharing fleets face persistent brute-force attacks. Even unsuccessful theft destroys docking infrastructure, driving up maintenance costs.
- **Stalling Sustainable Transit:** High replacement costs force companies to raise rental prices and halt service expansion, acting as a barrier to green transportation.
- **The Core Gap:** Current security focuses entirely on stronger locks rather than intelligent, active sensing that responds to tampering as it happens.

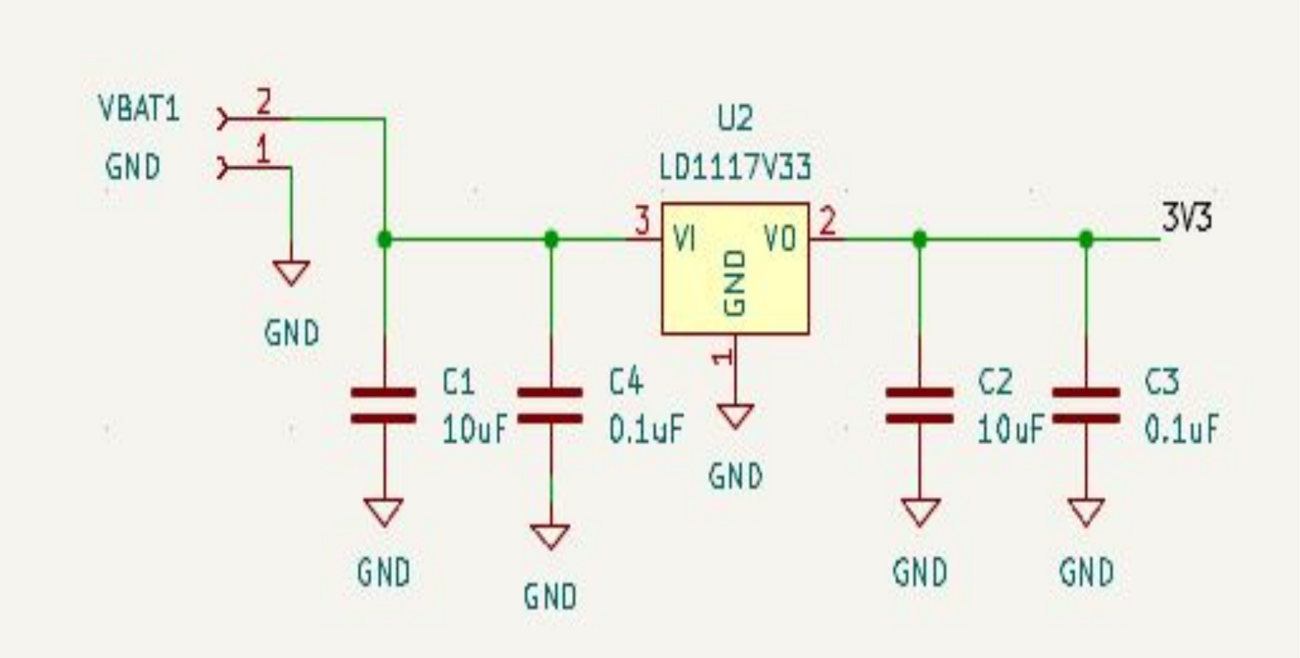
Solution: E-bike Theft Detection Alarm System

- **Active Real-Time Deterrence:** A custom embedded PCB utilizes an IMU (LIS3DHTR) mounted to the frame to continuously monitor for motion signatures unique to theft.
- **Intelligent Noise Filtering:** The system applies a Digital Low-Pass Filter (DLPF) and Root Mean Square (RMS) energy calculations to effectively ignore wind, bumps, and vibrations.
- **State Machine Decision Logic:** Sensor data is processed through a Finite State Machine to ensure only sustained tampering triggers a response.
- **Alarm Intervention:** Once tampering is identified, the device activates a 75 dB siren to actively deter the thief and alert bystanders.

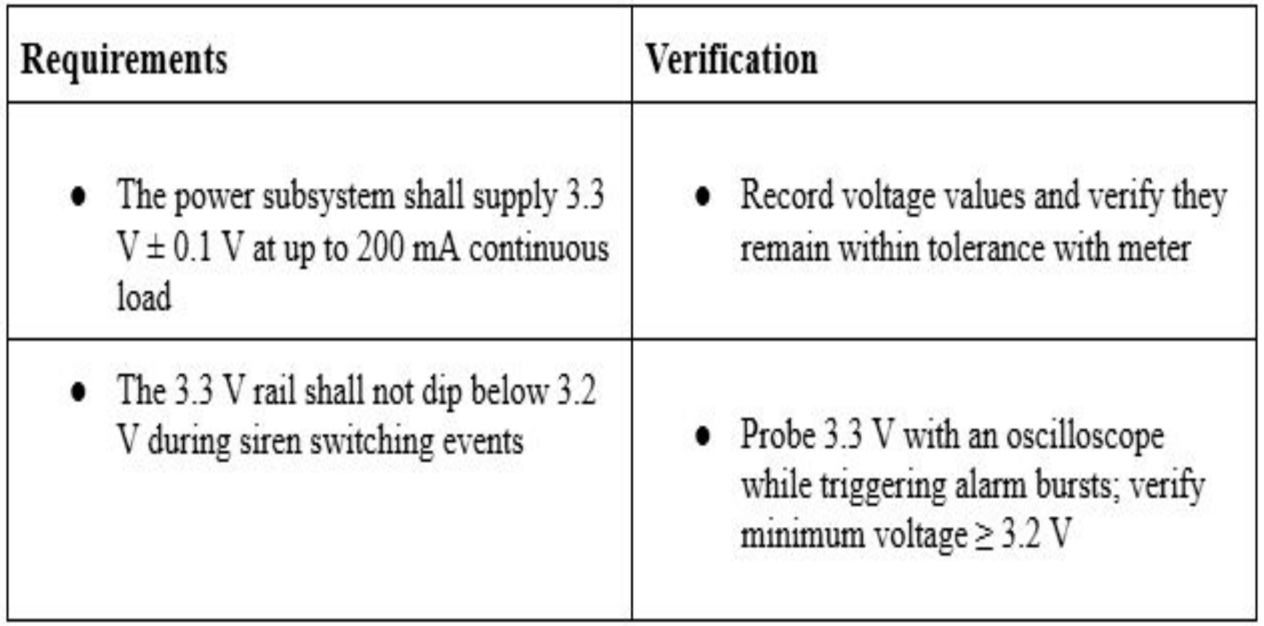
Block Diagram



Power Subsystem



Power Subsystem PCB Schematic



Requirements	Verification
• The power subsystem shall supply 3.3 V ± 0.1 V at up to 200 mA continuous load	• Record voltage values and verify they remain within tolerance with meter
• The 3.3 V rail shall not dip below 3.2 V during siren switching events	• Probe 3.3 V with an oscilloscope while triggering alarm bursts; verify minimum voltage ≥ 3.2 V

R&V Table

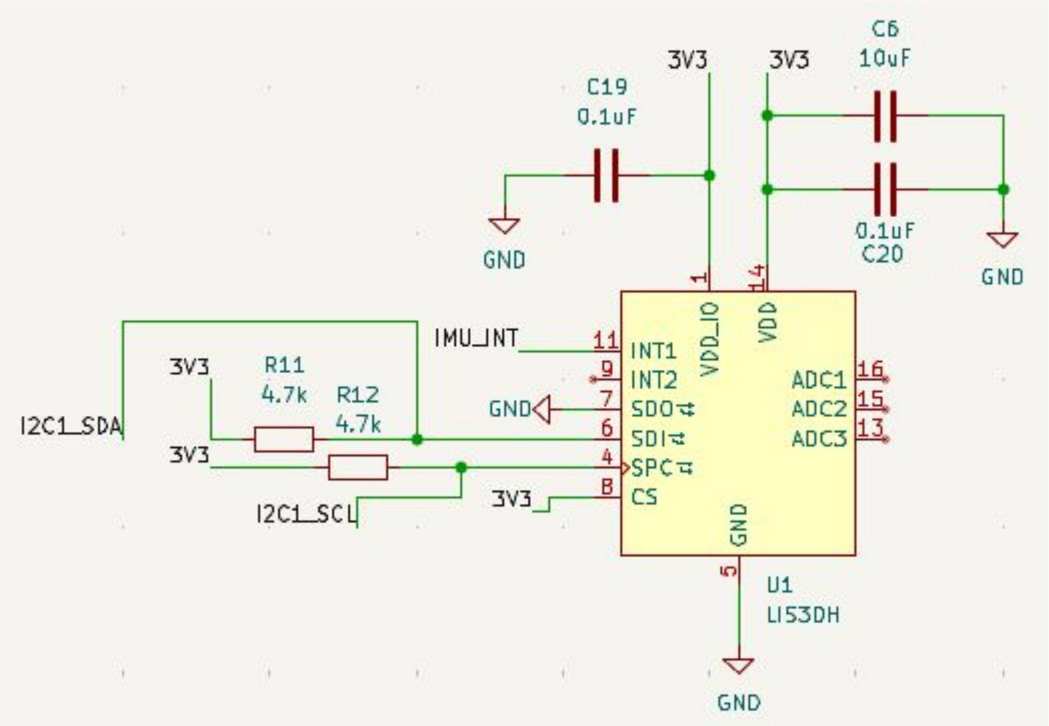
Power Subsystem: Voltage Measurements

Total Power Supplied:

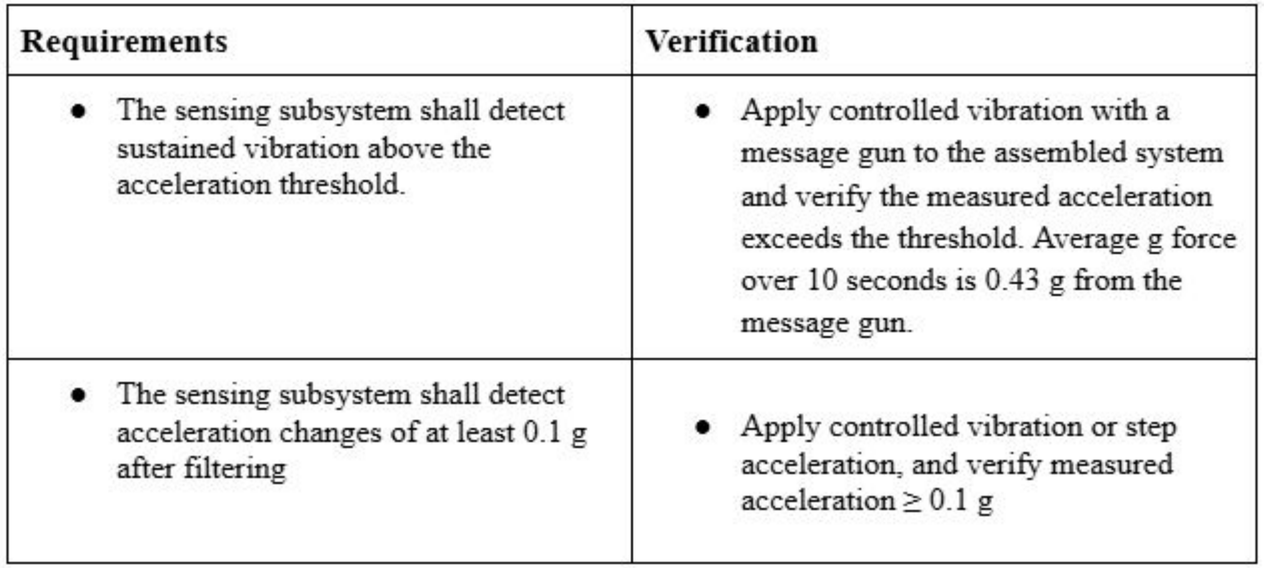
Regulated voltage value:

Regulated Voltage during switching event:

Sensing Subsystem



Sensing Subsystem PCB Schematic

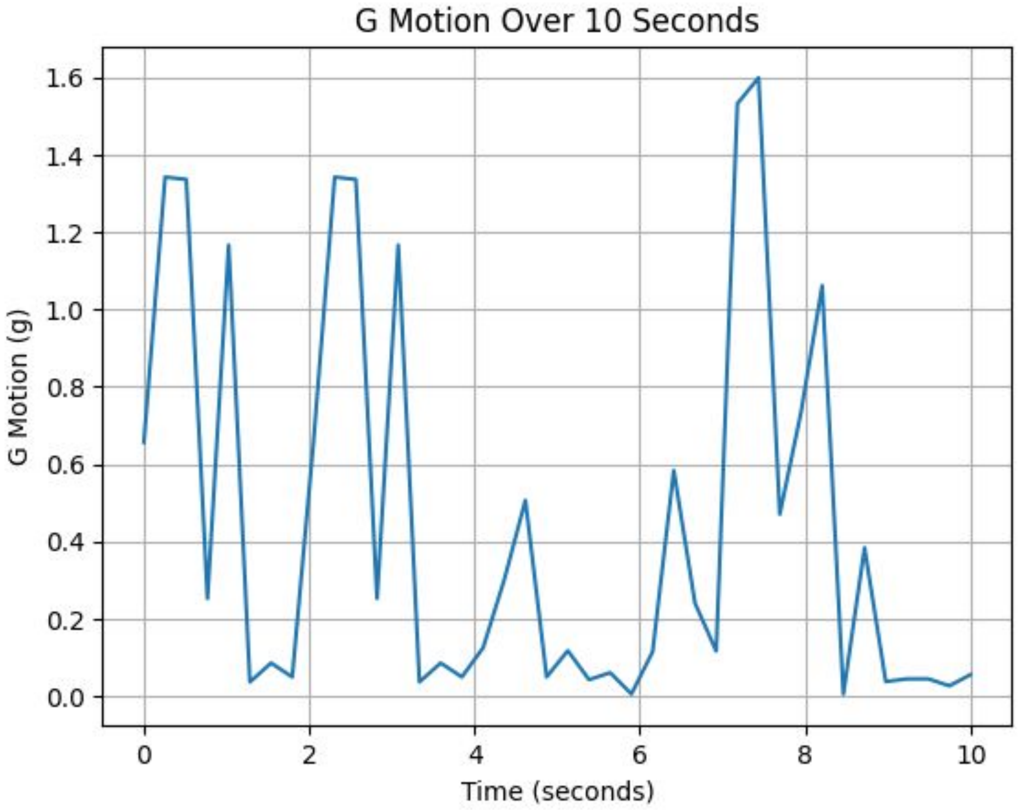


Requirements	Verification
• The sensing subsystem shall detect sustained vibration above the acceleration threshold.	• Apply controlled vibration with a message gun to the assembled system and verify the measured acceleration exceeds the threshold. Average g force over 10 seconds is 0.43 g from the message gun.
• The sensing subsystem shall detect acceleration changes of at least 0.1 g after filtering	• Apply controlled vibration or step acceleration, and verify measured acceleration ≥ 0.1 g

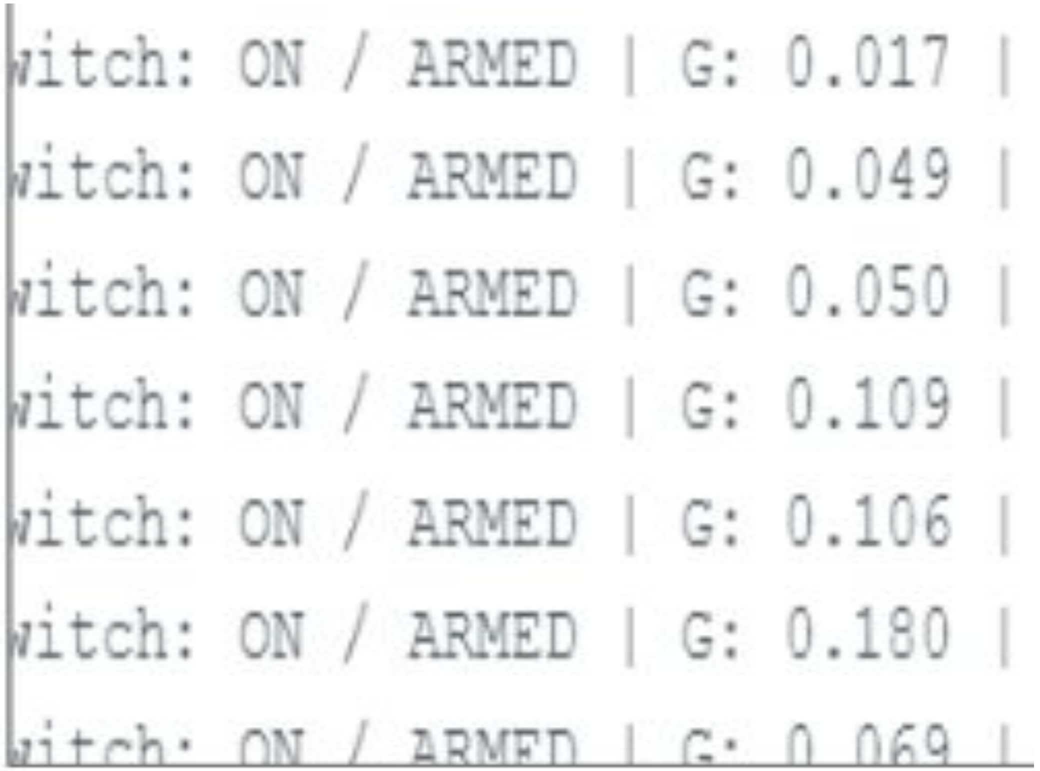
R&V Table

Sensing Subsystem: Acceleration Data

Average G value(.43 g) recorded from massage gun:

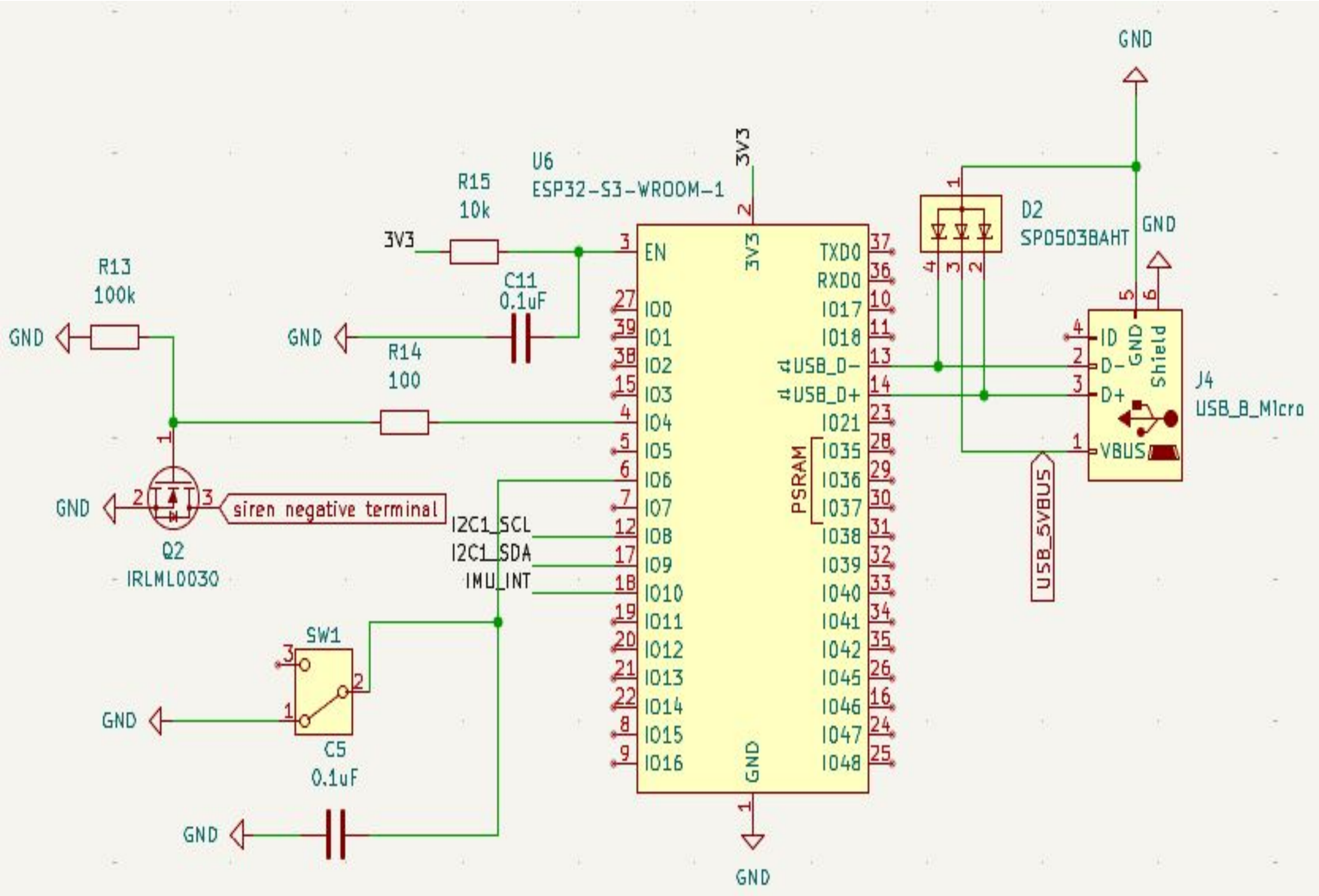


System detection of g values of approximently .1 g:



```
witch: ON / ARMED | G: 0.017 |
witch: ON / ARMED | G: 0.049 |
witch: ON / ARMED | G: 0.050 |
witch: ON / ARMED | G: 0.109 |
witch: ON / ARMED | G: 0.106 |
witch: ON / ARMED | G: 0.180 |
witch: ON / ARMED | G: 0.069 |
```

Control Subsystem

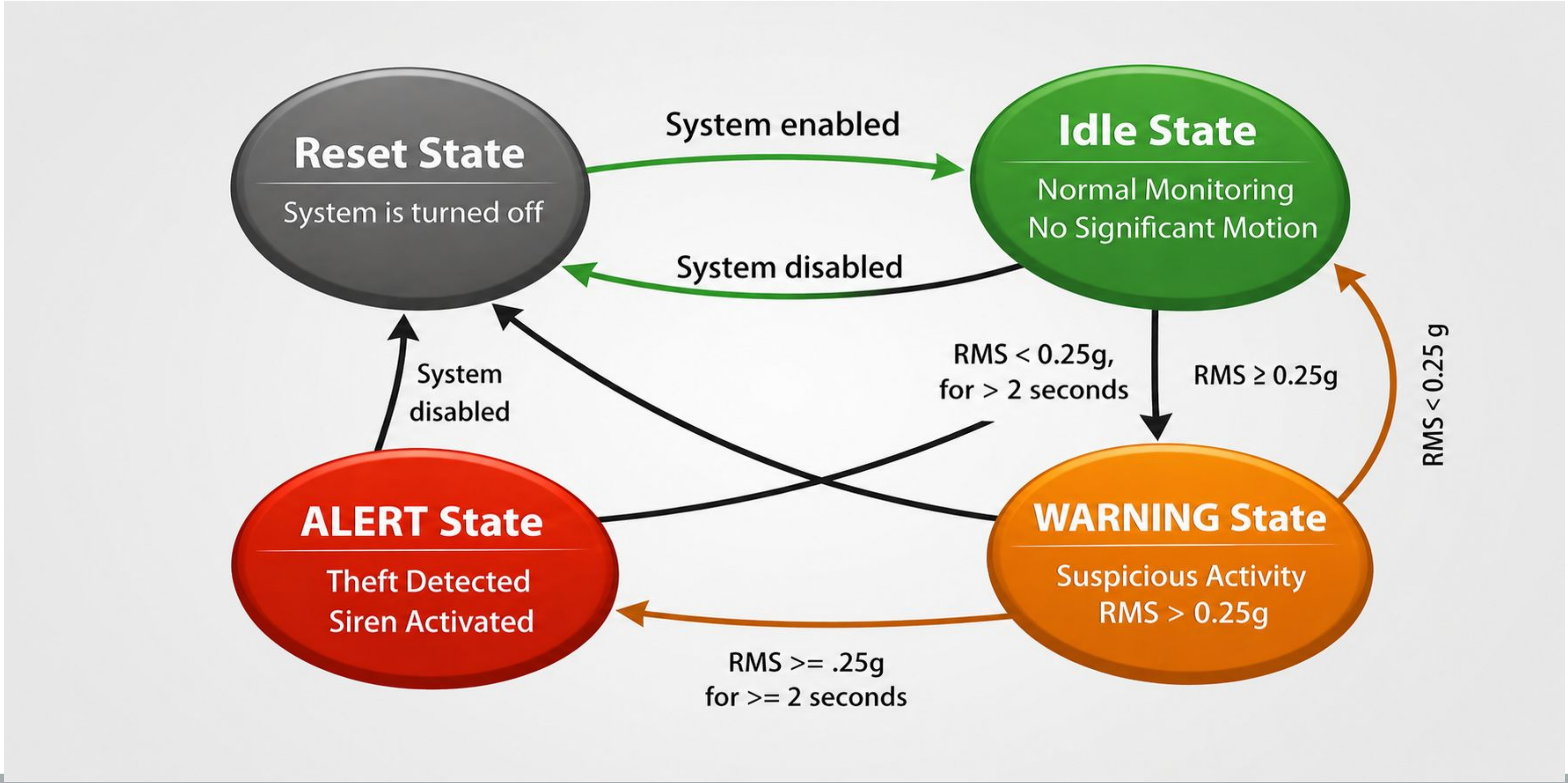


Control Subsystem PCB Schematic

Requirements	Verification
- The FSM shall transition to Alarm within 2.0 s ± 0.2 s of sustained tamper motion	- Apply a repeatable shaking stimulus to the mounted system, which maintains an average of over .25 g. - Use a GPIO pin to indicate FSM Alarm state. - Repeat tests across multiple trials, and compute the average.
- The system shall achieve ≥ 90% tamper detection accuracy over 10 trials tamper trials and 10 non tamper trials	- A tamper trial is defined as sustained shaking motion of over .25 g back and forth for over 2 seconds - A non-tamper trail consists of either a rapid spike of over .25 g for less than a second or any amount of shaking with a force of .25 g or less for any period of time.
- The firmware shall boot into a safe non-alarm state (Idle)	- Verify alarm remains inactive until valid detection occurs.

R&V Table

Finite State Machine



Reset State

System is turned off

Idle State

Normal Monitoring
No Significant Motion

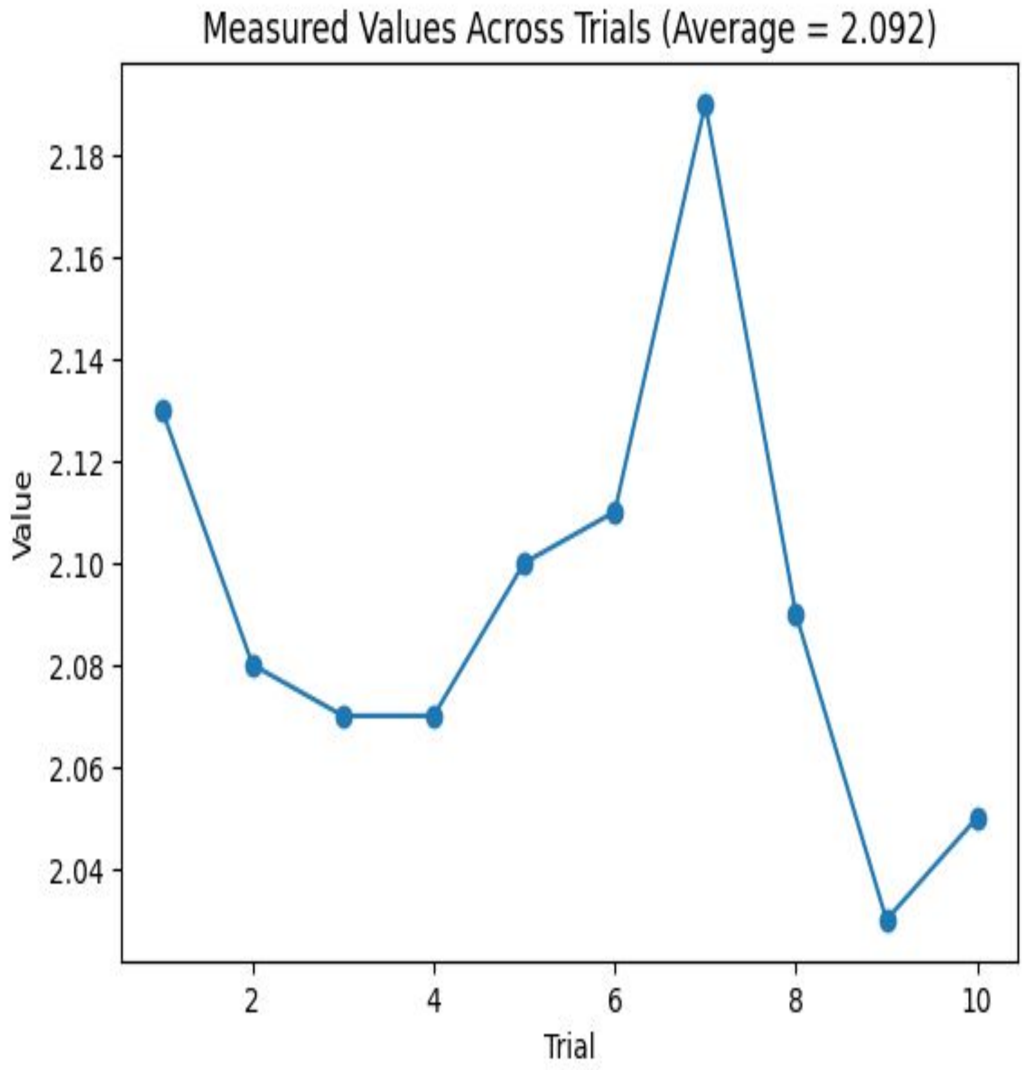
WARNING State

Suspicious Activity
RMS > 0.25g

ALERT State

Theft Detected
Siren Activated

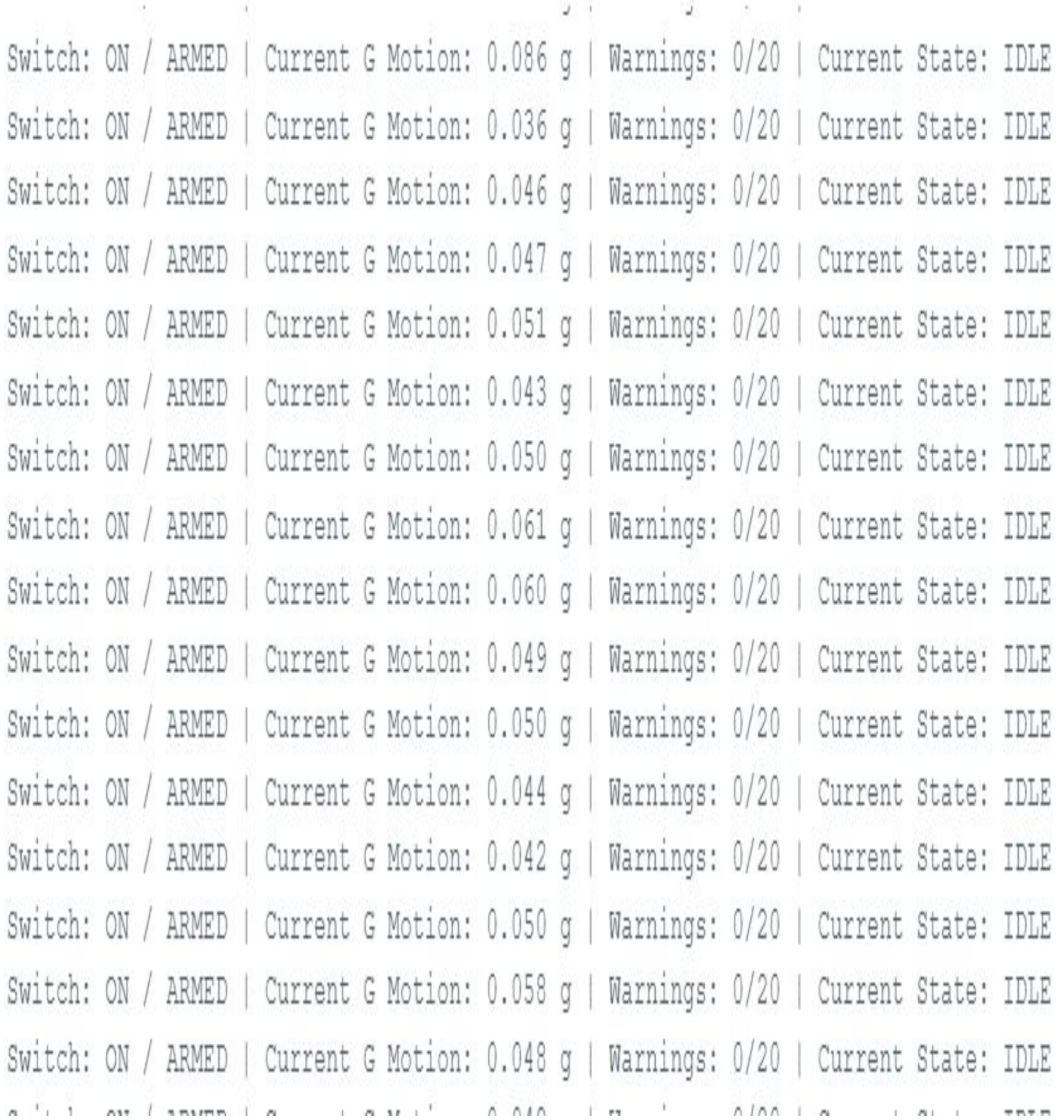
Control Subsystem FSM Data



FSM transition Timing

- 100% accuracy across 20 tamper trails
- 95% accuracy across 20 non tamper trials

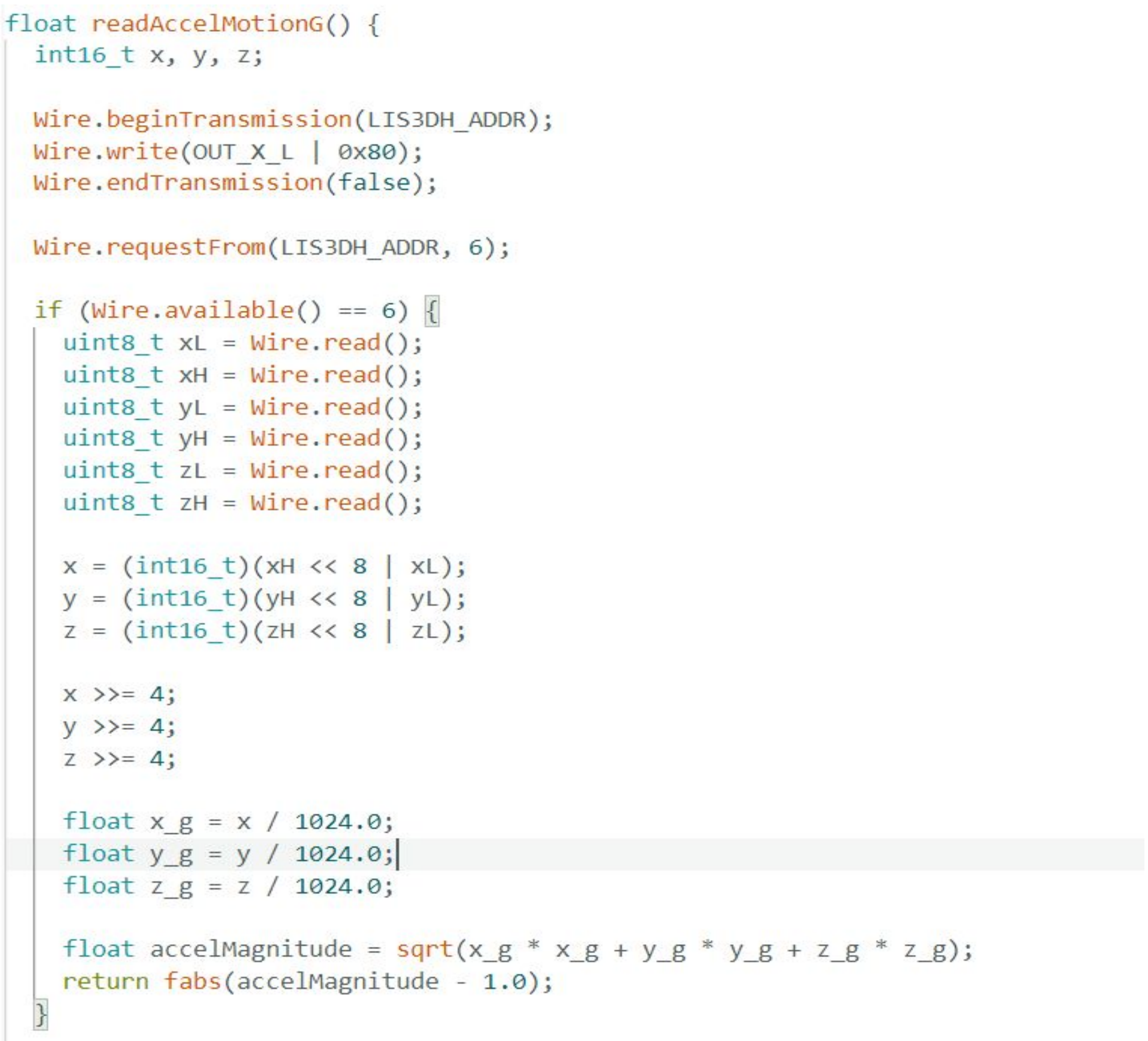
Test Accuracy



```
Switch: ON / ARMED | Current G Motion: 0.086 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.036 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.046 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.047 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.051 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.043 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.050 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.061 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.060 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.049 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.050 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.044 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.042 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.050 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.058 g | Warnings: 0/20 | Current State: IDLE
Switch: ON / ARMED | Current G Motion: 0.048 g | Warnings: 0/20 | Current State: IDLE
```

Idle State Bootup

Code Flow



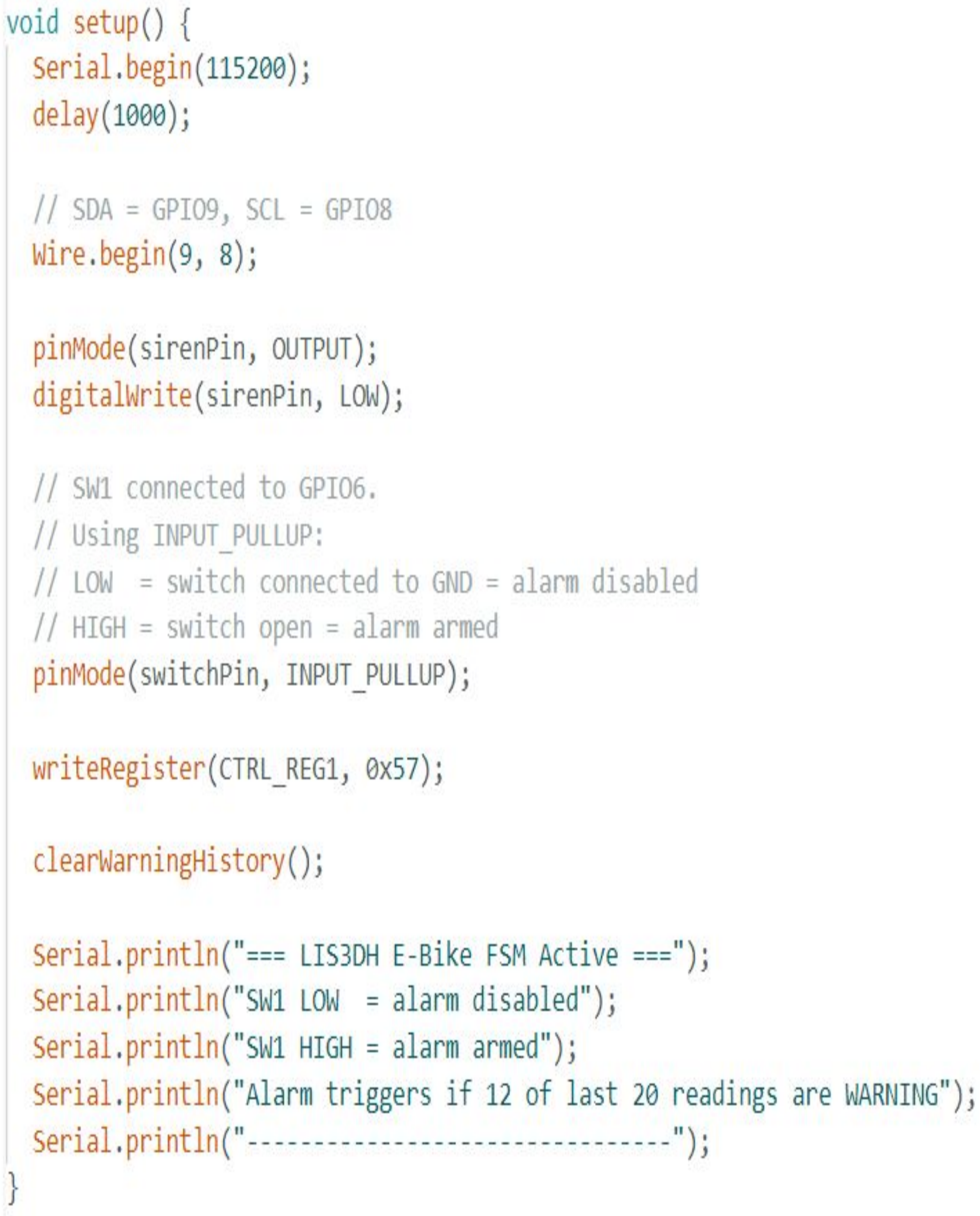
```
float readAccelMotionG() {
  int16_t x, y, z;

  Wire.beginTransmission(LIS3DH_ADDR);
  Wire.write(OUT_X_L | 0x80);
  Wire.endTransmission(false);

  Wire.requestFrom(LIS3DH_ADDR, 6);

  if (Wire.available() == 6) {
    uint8_t xL = Wire.read();
    uint8_t xH = Wire.read();
    uint8_t yL = Wire.read();
    uint8_t yH = Wire.read();
    uint8_t zL = Wire.read();
    uint8_t zH = Wire.read();

    x = (int16_t)(xH << 8 | xL);
    y = (int16_t)(yH << 8 | yL);
    z = (int16_t)(zH << 8 | zL);

    x >>= 4;
    y >>= 4;
    z >>= 4;

    float x_g = x / 1024.0;
    float y_g = y / 1024.0;
    float z_g = z / 1024.0;

    float accelMagnitude = sqrt(x_g * x_g + y_g * y_g + z_g * z_g);
    return fabs(accelMagnitude - 1.0);
  }
```

Acceleration Calculation

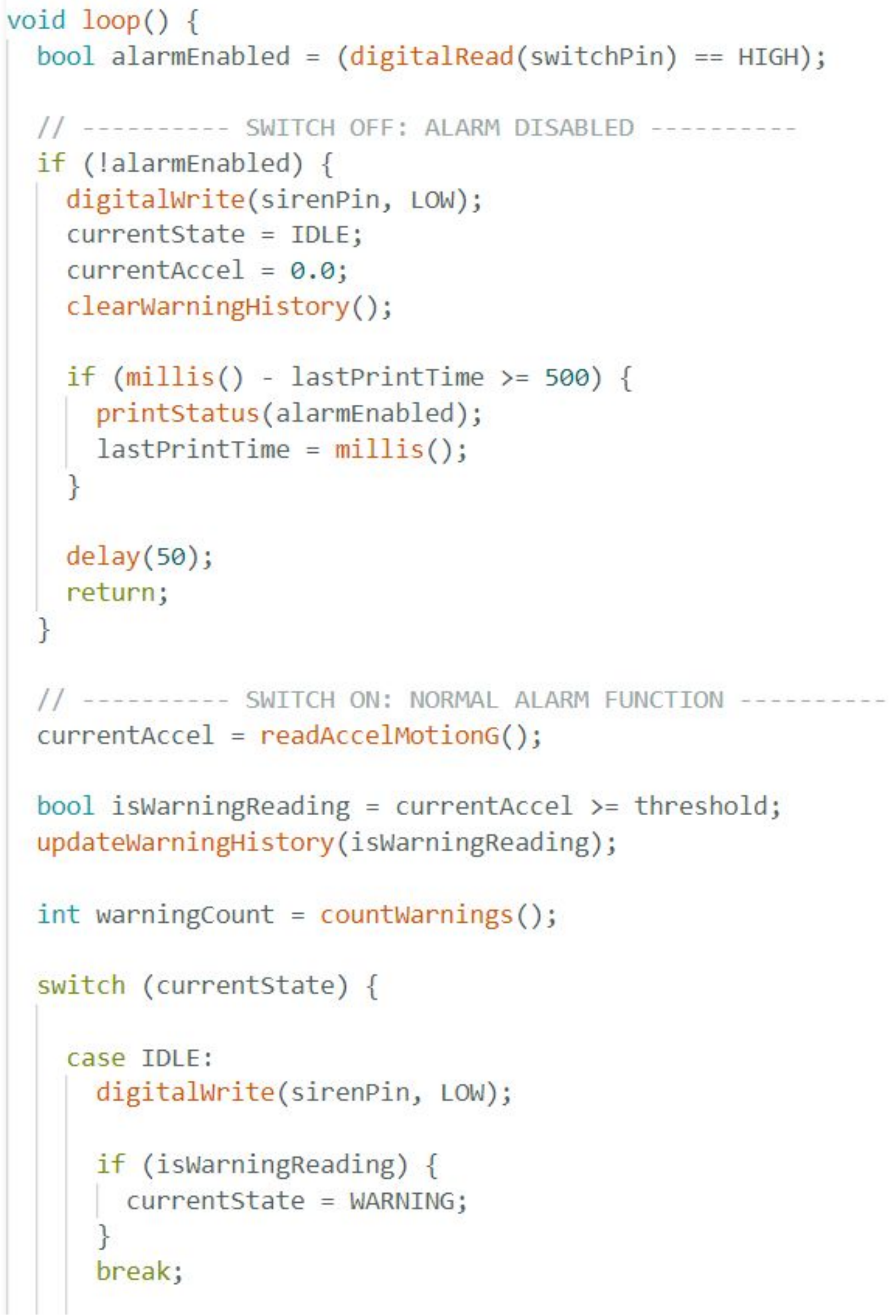
```
void updateWarningHistory(bool isWarning) {
  warningHistory[historyIndex] = isWarning;
  historyIndex = (historyIndex + 1) % HISTORY_SIZE;

  if (historyCount < HISTORY_SIZE) {
    historyCount++;
  }
}

int countWarnings() {
  int count = 0;

  for (int i = 0; i < historyCount; i++) {
    if (warningHistory[i]) {
      count++;
    }
  }

  return count;
}

void clearWarningHistory() {
  for (int i = 0; i < HISTORY_SIZE; i++) {
    warningHistory[i] = false;
  }

  historyIndex = 0;
  historyCount = 0;
}
```

warnings count

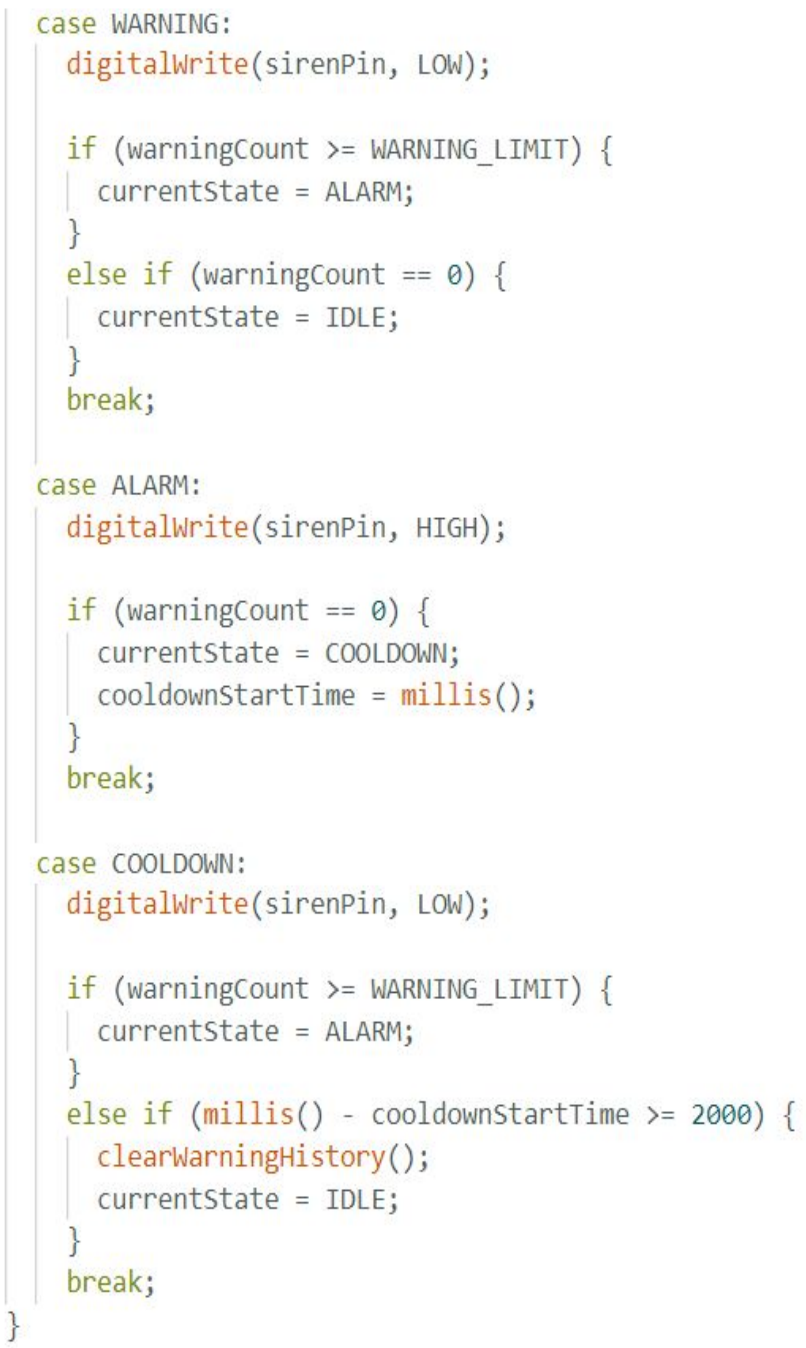
Setup & Loop



```
void setup() {
  Serial.begin(115200);
  delay(1000);

  // SDA = GPIO9, SCL = GPIO8
  Wire.begin(9, 8);

  pinMode(sirenPin, OUTPUT);
  digitalWrite(sirenPin, LOW);

  // SW1 connected to GPIO6.
  // Using INPUT_PULLUP:
  // LOW  = switch connected to GND = alarm disabled
  // HIGH = switch open = alarm armed
  pinMode(switchPin, INPUT_PULLUP);

  writeRegister(CTRL_REG1, 0x57);

  clearWarningHistory();

  Serial.println("=== LIS3DH E-Bike FSM Active ===");
  Serial.println("SW1 LOW  = alarm disabled");
  Serial.println("SW1 HIGH = alarm armed");
  Serial.println("Alarm triggers if 12 of last 20 readings are WARNING");
  Serial.println("--------------------------------");
}
```

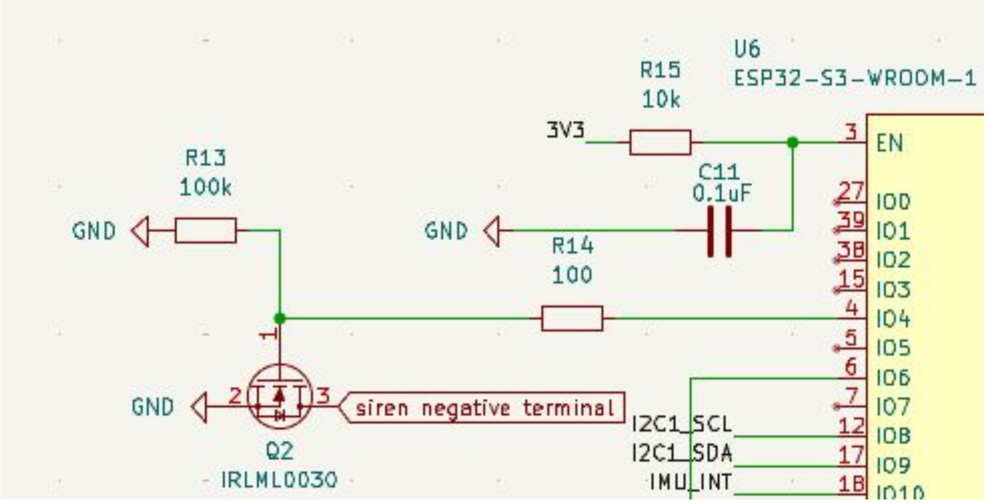


```
void loop() {
  bool alarmEnabled = (digitalRead(switchPin) == HIGH);

  // ---------- SWITCH OFF: ALARM DISABLED ----------
  if (!alarmEnabled) {
    digitalWrite(sirenPin, LOW);
    currentState = IDLE;
    currentAccel = 0.0;
    clearWarningHistory();

    if (millis() - lastPrintTime >= 500) {
      printStatus(alarmEnabled);
      lastPrintTime = millis();
    }

    delay(50);
    return;
  }

  // ---------- SWITCH ON: NORMAL ALARM FUNCTION ----------
  currentAccel = readAccelMotionG();

  bool isWarningReading = currentAccel >= threshold;
  updateWarningHistory(isWarningReading);

  int warningCount = countWarnings();

  switch (currentState) {

    case IDLE:
      digitalWrite(sirenPin, LOW);

      if (isWarningReading) {
        currentState = WARNING;
      }
      break;
```

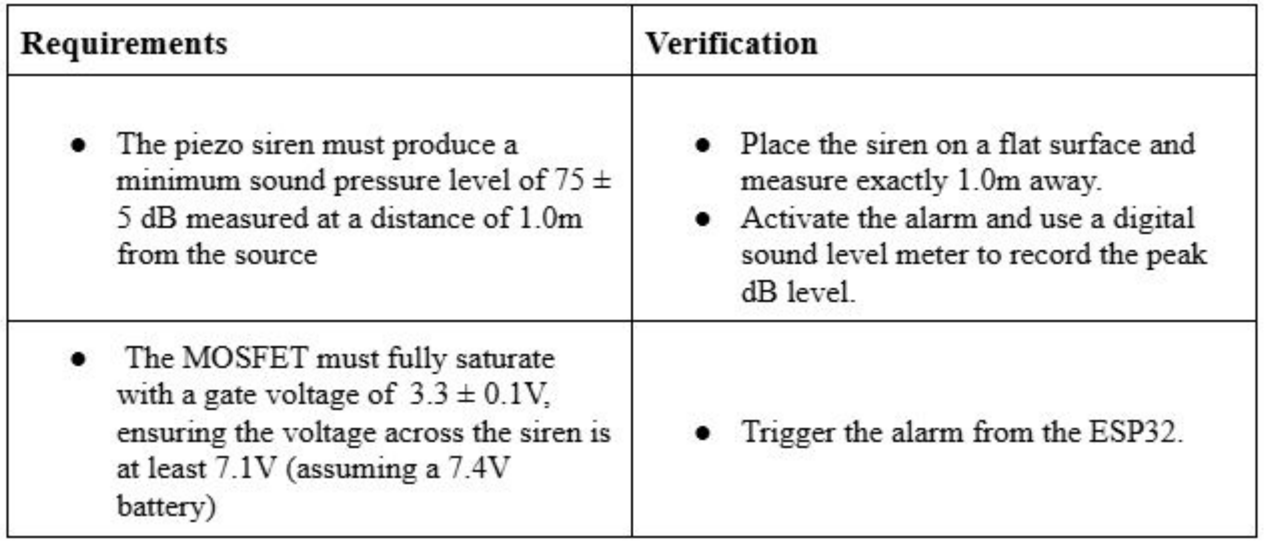


```
    case WARNING:
      digitalWrite(sirenPin, LOW);

      if (warningCount >= WARNING_LIMIT) {
        currentState = ALARM;
      }
      else if (warningCount == 0) {
        currentState = IDLE;
      }
      break;

    case ALARM:
      digitalWrite(sirenPin, HIGH);

      if (warningCount == 0) {
        currentState = COOLDOWN;
        cooldownStartTime = millis();
      }
      break;

    case COOLDOWN:
      digitalWrite(sirenPin, LOW);

      if (warningCount >= WARNING_LIMIT) {
        currentState = ALARM;
      }
      else if (millis() - cooldownStartTime >= 2000) {
        clearWarningHistory();
        currentState = IDLE;
      }
      break;
  }
```

Alarm Subsystem



Alarm Subsystem PCB Schematic



Requirements	Verification
• The piezo siren must produce a minimum sound pressure level of 75 ± 5 dB measured at a distance of 1.0m from the source	• Place the siren on a flat surface and measure exactly 1.0m away. • Activate the alarm and use a digital sound level meter to record the peak dB level.
• The MOSFET must fully saturate with a gate voltage of 3.3 ± 0.1V, ensuring the voltage across the siren is at least 7.1V (assuming a 7.4V battery)	• Trigger the alarm from the ESP32.

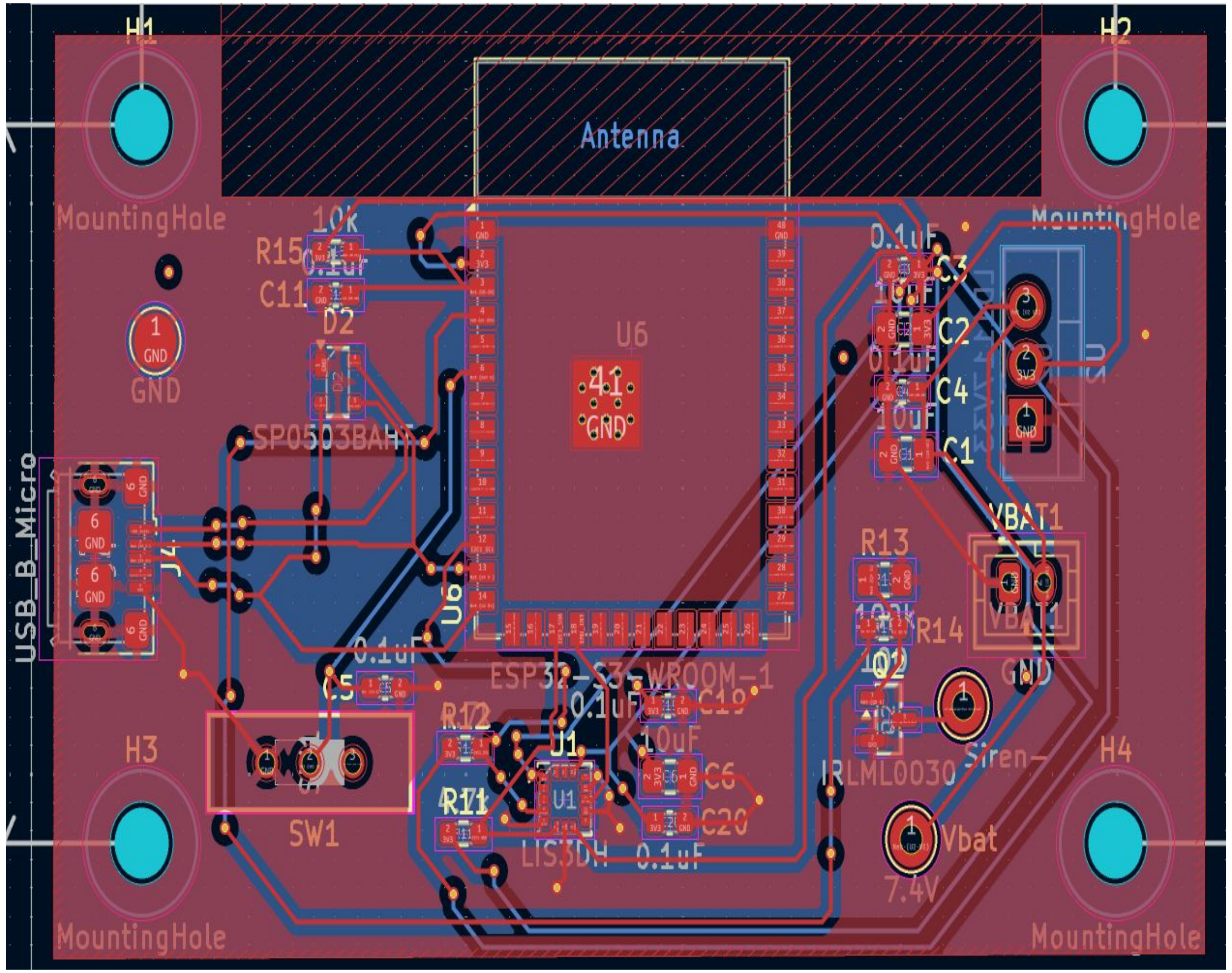
R&V Table

Alarm Subsystem Verfication

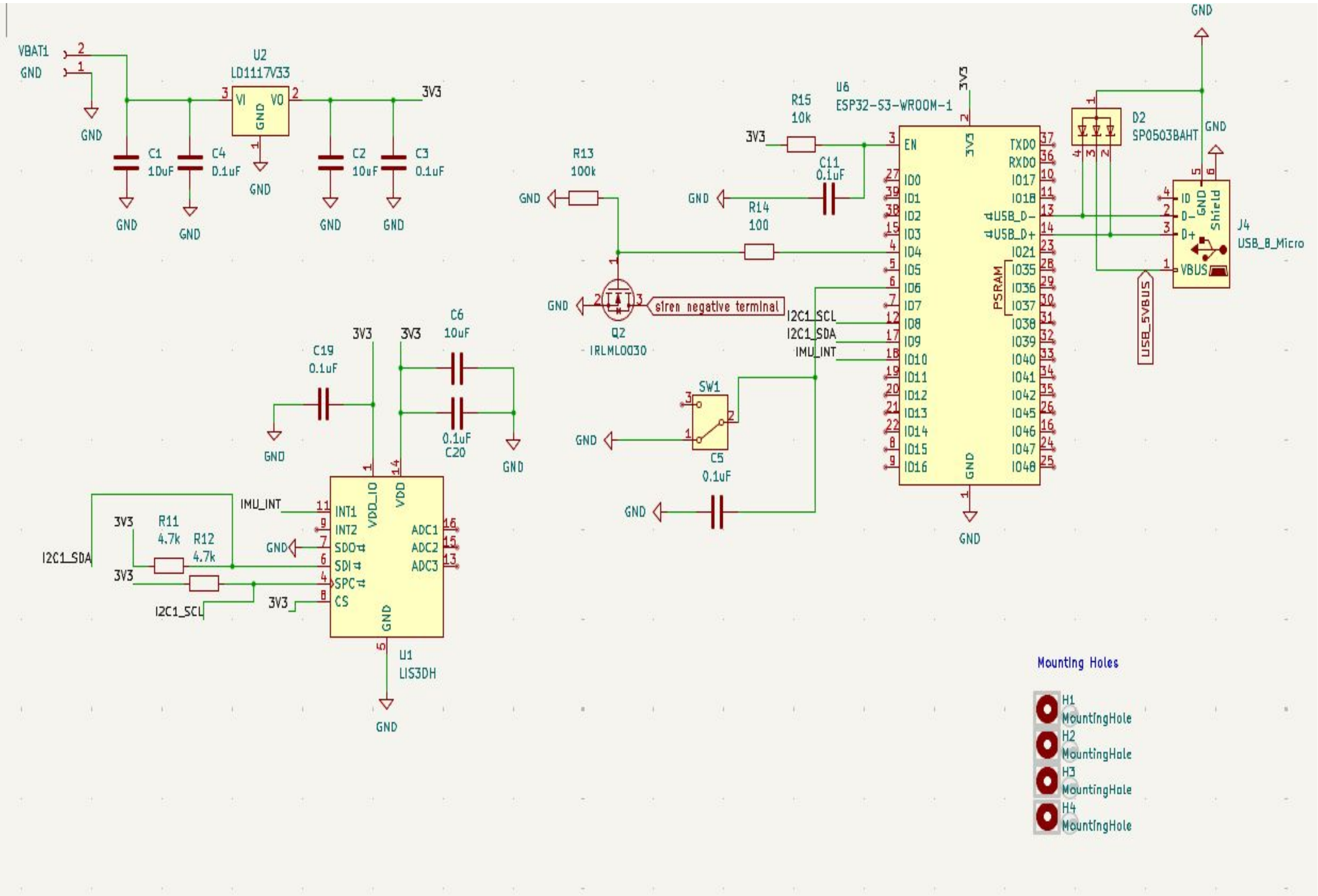
alarm decibel value(1 m away)

MOSFET voltage saturation

Final Design

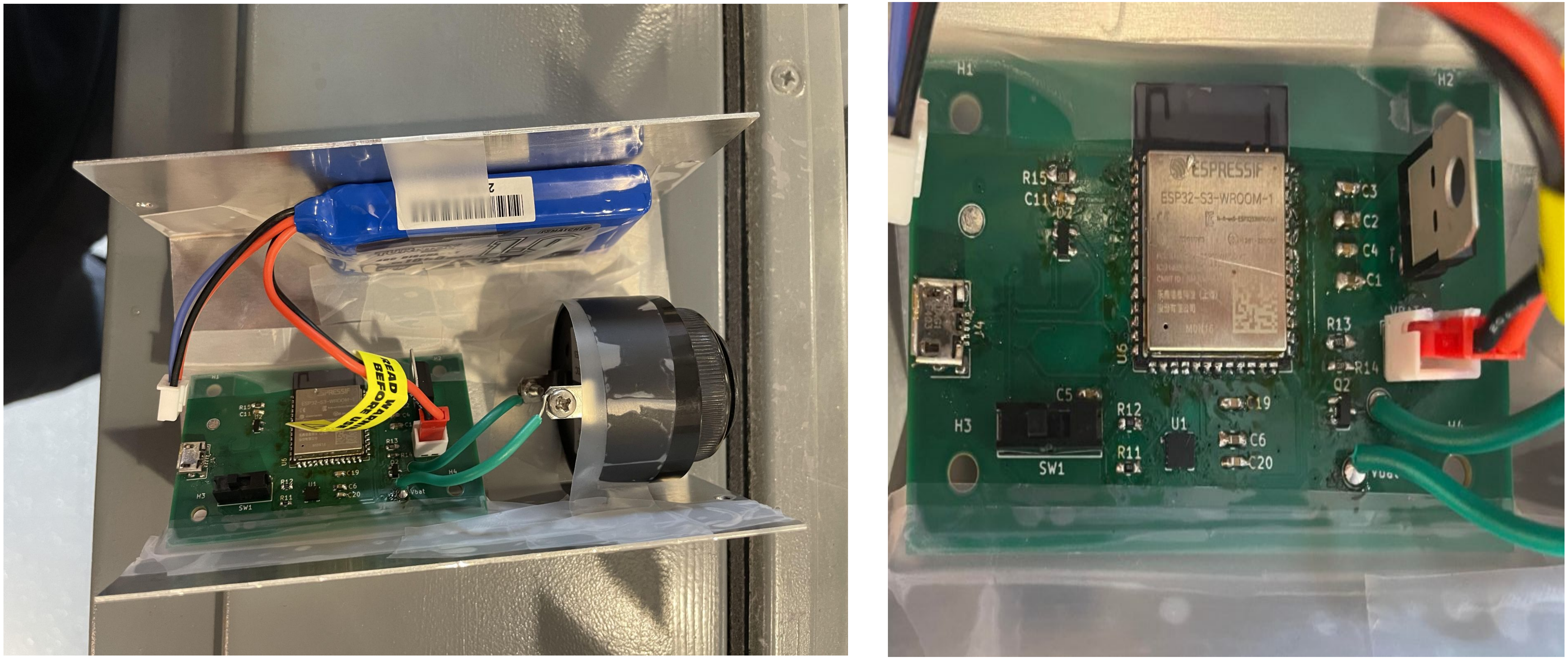


Final PCB Board Design

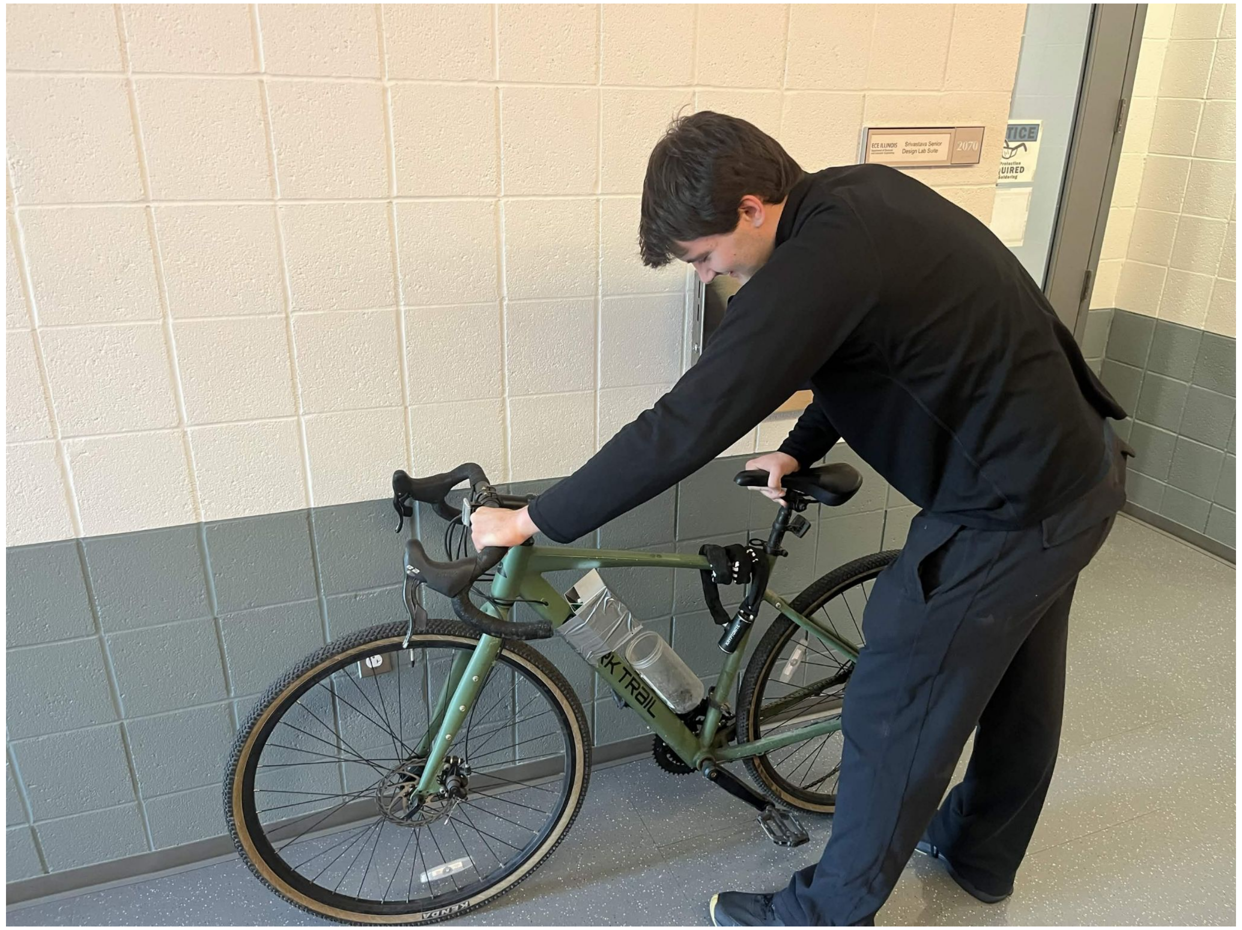


Full Circuit Schematic

PCB Enclosure



Pictures of Bike - Theft Attempt

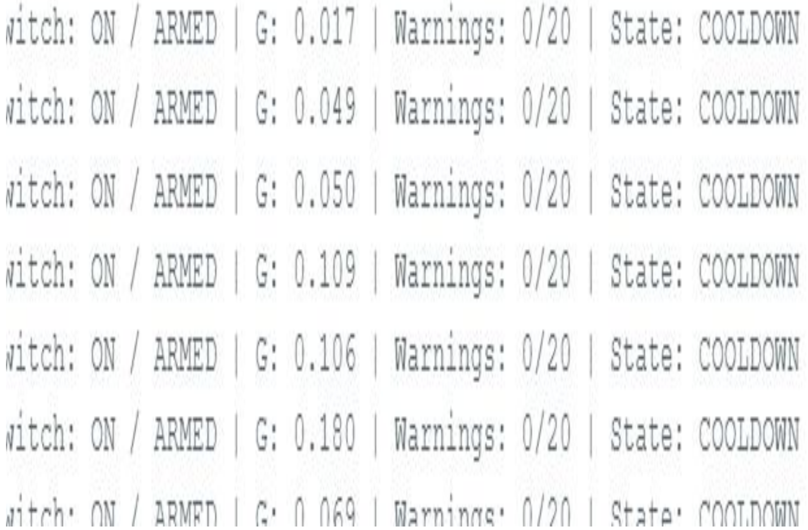


High Level Requirements

- **Power Stability:** Maintained 3.3V ± 0.1V under continuous operation → ensured reliable sensing and immediate alarm response
- **Motion Sensing:** I2C communication stable; detected ≥0.1g changes with filtering → minimized false alarms while achieving ≥90% accuracy
- **Detection Performance:** Achieved ≤2s+/ .2 latency with real-time logic → consistently met ≥90% accuracy in tamper vs. non-tamper tests

Power Stability:

Motion Sensing:



```
witch: ON / ARMED | G: 0.017 | Warnings: 0/20 | State: COOLDOWN
witch: ON / ARMED | G: 0.049 | Warnings: 0/20 | State: COOLDOWN
witch: ON / ARMED | G: 0.050 | Warnings: 0/20 | State: COOLDOWN
witch: ON / ARMED | G: 0.109 | Warnings: 0/20 | State: COOLDOWN
witch: ON / ARMED | G: 0.106 | Warnings: 0/20 | State: COOLDOWN
witch: ON / ARMED | G: 0.180 | Warnings: 0/20 | State: COOLDOWN
witch: ON / ARMED | G: 0.069 | Warnings: 0/20 | State: COOLDOWN
```

Detection Performance:

- 100% accuracy across 20 tamper trails
- 95% accuracy across 20 non tamper trials

Challenges

- **Late-Stage Component Pivots:** Discovering voltage mismatches forced us into a late IMU swap, which resulted in a rapid redesign of both our firmware and PCB layout.
- **Footprint & Sourcing Errors:** Minor layout oversights such as awkwardly placing the USB connector, sizing the battery connector incorrectly, and confusing the specific voltage regulator model created major assembly issues.
- **Hardware Damage & Soldering:** We lost valuable time to fried components while soldering and testing, particularly when working on the siren setup without dedicated ground pads for safe probing.

Successes

- **Rapid System Recovery:** After encountering a severe hardware setback, we successfully resoldered and tested the entire final PCB within a single week of receiving the updated board.
- **Full System Integration:** Despite the challenges, we delivered a fully functional theft detection system that successfully unifies the ESP32, IMU, power regulation, and high-current siren.

Conclusions from the Project: What we Learned

- **Time Management is Critical:** Delays in hardware development compounded quickly. Missing early PCB ordering windows meant we could not place our first PCB order until the third round. A stricter schedule would have saved us weeks of soldering and debugging time.
- **PCB Layout & Part Accuracy:** Minor layout oversights created unnecessary headaches. We learned that verifying exact physical footprints before manufacturing is just as important as the schematic design.
- **The Necessity of Subsystem Testing:** Skipping the breadboard testing phase for our initial IMU delayed the discovery of a critical voltage mismatch. However, rigorously unit testing our other three subsystems was the sole reason we successfully recovered and completed the project on time.

Recommendations for Further Work

- **Upgraded Enclosure:** Transition from the prototype metal box to a custom 3D-printed, weather-resistant container to better protect the PCB from physical impact and potential weather conditions.

- **Smart Docking Integration:** Replace the physical arm/disarm switch with a magnetically triggered sensor designed to automatically arm/disarm the system when locked into a Divvy-style docking station.

- **Self-Sustaining Power Architecture:** Introduce a battery management system with a charging circuit to allow for continuous operation without the need for battery replacements.

Ethics

- We strictly adhered to the IEEE Code of Ethics mandate to prioritize public welfare by designing an active deterrence system that operates without introducing physical or electrical hazards to the rider.

- To comply with the ACM directive of avoiding harm, the system's alarm is explicitly capped at 85 decibels to deter thieves without causing hearing damage or physical distress to innocent bystanders.

- We actively mitigate the risk of creating a public nuisance by utilizing digital low-pass filtering and RMS energy calculations to mathematically eliminate false alarms caused by environmental factors.

Thank you
Questions?