Multi-Level Security System

*Ryan Miller* | *Calvin Friedrich*

5/16/2025

**Table of Contents:**

|  |  |
| --- | --- |
| Title Page | Page 1 |
| Table of Contents | Page 2 |
| Executive Summary | Page 3 |
| Background:   * Problem Statement * Requirements | Page 3 – 4 |
| Design:   * Solution Overview * Design Details * Sub-system Diagram * Bill of Materials * Design Verification * Summary of Outcomes | Page 5 - 9 |
| Power Budget | Page 9-10 |
| Lessons Learned and Reflection | Page 10 |
| References | Page 11 |

**Executive Summary:**

This project demonstrates the design and implementation of an autonomous, microprocessor-based security system that monitors the environment and motion cues to protect a defined space. When the ambient light drops below a predefined threshold, the system automatically arms itself, ready to sense motion via a passive infrared (PIR) sensor. Once motion is detected, an ultrasonic sensor measures object proximity. As the object moves closer, the system will display higher threat levels using LCD, LEDs, and a buzzer. If an object crosses a critical distance threshold, a loud audible alarm triggered. An accelerometer continuously monitors for tampering or unexpected movement of the device itself, adding to the security integrity. Throughout operation, the system displays the smallest measured distance until manually reset, providing a record of the nearest approach. Overall, this solution demonstrates robust sensor integration and real-time threat assessment suitable for autonomous perimeter monitoring.

**Background:**

**Problem Statement:** Traditional security systems often rely on manual activation and provide only binary intrusion alerts, lacking nuanced threat assessment and tamper detection. This project aims to create a self‐arming, light‐sensitive security system that differentiates between motion and proximity threats using PIR and ultrasonic sensors, maintains the closest approach record, and contains an onboard accelerometer to detect unauthorized movement of the device itself.

**REQUIREMENTS**  
The system SHALL:

1. **Processor and Operation**

* **Operate using an ARM Cortex M4 processor**
* **Function autonomously after initial setup/reset and power-on**
* **Perform real-time sensor data acquisition, processing, and decision-making without noticeable lag**

1. **Power and Battery**

* **Be powered from a standard 120 VAC outlet**
* **Optimize power by initializing sensors only when needed**

1. **Light and Tamper Detection**

* **Use a photoresistor to detect ambient light and automatically arm the system when lights are off**
* **Implement tamper detection via the LIS3DSH accelerometer; trigger an audible alarm upon displacement**

1. **Proximity Detection and Threat Level Assessment**

* **Utilize an HC-SR04 ultrasonic sensor to monitor object proximity against a predefined threshold**
* **Trigger the alarm/buzzer and display “DANGER ALERT—HIGH THREAT” on the LCD when the threshold is crossed**

1. **Message Display**

* **Display “Armed” on power-up and “Danger” when a threat is detected**
* **Provide real-time proximity readings or threat-level indicators on the LCD**

1. **Alarm and Response**

* **Activate the WT-1205 buzzer whenever proximity or accelerometer tamper thresholds are exceeded**
* **Differentiate low, medium, and high threat levels with distinct audio and visual cues**

1. **LED Proximity Indicators**

* **Arrange LEDs to represent proximity bins**
* **Light LEDs progressively as objects move closer, indicating threat severity**

1. **System Reset and Manual Override**

* **Include a manual override (e.g., button press or external command) to reset/deactivate the system**
* **Automatically return to Armed after the alarm clears and no threat is detected**

1. **User Interface and Feedback**

* **Provide clear feedback via LCD, LEDs, and buzzer on system status and detected threats**
* **Allow configuration of settings (e.g., threshold distance, sensitivity) through a simple interface or external method**

**Design:**

**Solution Overview**

The security system is organized around a central STM32F407 microcontroller that facilitates sensor readings, state transitions, and actuator outputs. It operates as a finite state machine with three primary states—**Sleep**, **Armed**, and **Active**—and transitions automatically based on ambient light, motion, and distance measurements. The system stays in sleep when the lights are on in the environment. Once the lights turn off the system enters an armed state and will start detecting for motion. The active state happens following motion being detected, and will monitor for proximity threats as well as tamper detection, sending alerts to the buzzer and LED matrix if a threat is perceived.

**Design Details:**

1. **Core Architecture**
   * **Microcontroller (STM32F407)**: Hosts the state machine, processes sensor inputs via ADC and timers, and drives LEDs and buzzer.
   * **Power Supply**: A regulated 5 V wall plug transformer that powers all sensors and the MCU
2. **Sensor Suite & Interfaces**
   * **Ambient Light Sensor (Photoresistor + ADC)**
     + Continuously sampled every 100 ms; falling below the light threshold triggers the transition from Sleep to Armed.
   * **Passive Infrared (PIR) Motion Sensor (GPIO Interrupt)**
     + Generates a rising-edge interrupt on motion; on detection in Armed, the system enters Active and cannot return to Sleep.
   * **Ultrasonic Distance Sensor (TIM3 Input Capture & Trigger PWM)**
     + Measures echo pulse width to compute distance using the speed of sound
     + Used to display the accurate distance of the object
   * **3-Axis Accelerometer (SPI)**
     + Polled at 10 Hz; sudden acceleration events above ±0.3 g trigger an immediate audible alarm. The accelerometer alarm is only able to trigger in Armed and Active states.
3. **Output & Feedback**
   * **LED Indicators (PWM)**
     + Mapped to proximity bins:
       - Blue External LED: Fades on/off entire time system is on
       - Orange External LED: Toggles on/off when motion is detected; stays on when high threat level is reached.
       - Red External LED: Toggles on/off when proximity <30cm (enters high threat).
       - On Board Blue LED: Turns on when lights turn off (ARMED + ACTIVE states).
       - On Board Green LED: Flashes when accelerometer movement exceeds threshold.
     + PWM generated by TIM9 channels, drives Blue LED fader.
   * **Buzzer/Alarm (PA5 via TIM2/4 PWM)**
     + Activated in Active state with a 2 kHz tone; various buzz rates triggered by interrupt service, dependent on which peripheral does the triggering.
4. **State Machine Logic**
   * **Sleep**: Default on power-up; Blue LED fades on/off to show user system is functioning, buzzer silent. Light Detected.
   * **Armed**: Entered when room light < threshold; LED indicates PIR sensor is detecting for motion.
   * **Active**: Entered when motion is detected; turns on ultrasonic sensor for proximity readings; enables buzzer to trigger. Orange LED toggles on/off at fixed rate. Can not leave this state. High proximity alert and tamper alarm are enable in this state.

**(Figure 1) Sub-System Block Diagram:**

A diagram of a machine

AI-generated content may be incorrect.

**(Figure 2) Bill of Materials and Total Design Cost:A screenshot of a computer

AI-generated content may be incorrect.**

The complete bill of materials consists of an STM32F407 microcontroller ($48.50), a PIR motion sensor ($2.50), an ultrasonic sensor ($2.50), a photo-resistor ($0.70), an LCD display ($3.90), a speaker ($0.50), assorted resistors, diode, transistor, wiring, and headers, for a total cost of $59.55.

**(Figure 3) Design Verification Table:**

| Test Category | Method | Acceptance Requirements: | Results & Reflections |
| --- | --- | --- | --- |
| 1. Firmware Verification | • Keil debugger stepping through FSM states • Logic analyzer capture of GPIO & timer waveforms | • Correct state transitions • PIR interrupts, ultrasonic pulses, PWM outputs align with expected timing | • All states and variables verified • Minor timing adjustment made to PWM duty cycle after scope review |
| 2. “Valuable Object” Security | • Slide/lift a small object under the armed device • Observe PIR & ultrasonic triggers driving Active state | • Any motion or proximity immediately enters Active • Correct LED, buzzer, LCD indications | • System reliably entered Active • “Dead zones” in distance thresholds identified and tightened to prevent stealth removal |
| 3. Prolonged-State Stability | • Leave device in Sleep, Armed, Active for extended periods • Monitor for unintended exits, hangs, or increased power draw | • No unintended state changes • No software hangs • Power draw within budget | • Watchdog and ISR routines proved reliable for tests over 24hrs • No unintended resets or excessive current draw detected |
| 4. Threshold Calibration | • Sweep ambient light (bright→dark) and record ADC at arming threshold • Apply calibrated taps/tilts to accelerometer, record g-force response | • Light threshold ≈ 200 lux reproducible • Accelerometer sensitivity ≈ 0.3 g yields minimal false alarms | • Thresholds repeatable across multiple trials • Tuned light ADC cutoff and accelerometer filter to reduce nuisance triggers |
| 5. Tamper-and-Lift Test | • Physically lift/rotate enclosure in Armed/Active • Verify immediate alarm response | • Immediate buzzer + red LED activation upon tampering | • SPI polling rate and interrupt latency sufficient • No delay in tamper detection |
| 6. Reset-Behavior Trials | • Trigger manual and automatic resets under various scenarios (light only, Armed, Active) • Check reset requires deliberate button press | • No reset on transient light changes in Active • Manual button press required for disarming | • Push button Reset  • Prevented accidental resets during brief motion or high-current events |

**Summary of Outcomes**  
We verified the system by first using the Keil debugger and a logic analyzer to step through each finite-state transition, confirm PIR interrupts, ultrasonic trigger/echo timing, and PWM outputs, then tightened minor duty-cycle timing after review. We slid and lifted a “valuable” object beneath the armed device to ensure any motion or proximity immediately drove the system into Active state with correct LED, buzzer, and LCD indications—eliminating stealth “dead zones.” Extended Sleep, Armed, and Active runs demonstrated no unintended state exits, software hangs, or excessive power draw thanks to our watchdog and ISR routines. Ambient light sweeps established a reproducible arming threshold at ≈200 lux, and calibrated accelerometer taps confirmed a 0.3 g sensitivity with minimal false alarms. Tamper tests (lifting/rotating the enclosure) produced immediate buzzers and red-LED alarms, validating our SPI polling and interrupt latency. Finally, reset trials across light-only, Armed, and Active scenarios proved that the system resists transient light changes and requires a deliberate button press to disarm. Overall our tests proved that our system was working successfully, and with in our requirement’s specifications.

**Power Budget Analysis:**

**(Figure 4) Power Budget Analysis Spread Sheet:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**Assumptions**

* The system is powered by a regulated 5 V DC source.
* Ambient temperature is 25 °C.
* PWM outputs are disabled (off) when not actively driving LEDs or buzzer.
* The LCD remains continuously active and does not enter a low-power idle state.
* Every security event (armed, motion detected, proximity alert) is displayed on the LCD.
* All component calculations (e.g., voltage, current) assume a 5 V supply rail.
* The device follows a 24 h cycle, with ambient lighting “on” for exactly 8 h (disarmed) and “off” for 16 h (armed).
* Currents are expressed in milliamps (mA) and time durations in hours (h).
* Total daily charge (mAh/day) is converted to energy (Wh/day) using

**Run Time on a 9 V Battery Before Recharge**

Based on a 9 V battery capacity of 500 mAh and our measured current draw:

* Minimum Runtime (Armed/Active Mode): 1.71 days × 24 h/day = 41.0 hours
* Maximum Runtime (Disarmed/Sleep State Mode): 2.31 days × 24 h/day = 55.4 hours

Thus, the system will operate continuously for approximately 41 hours under full-load (armed) conditions, and up to 55 hours on unarmed standby before the battery requires recharging.

**Lessons Learned and Reflections**

In reflecting on our project, we learned that clear, continuous communication, especially around pin assignments, timer channels, and function interface, is critical to preventing integration conflicts. Allocating dedicated blocks of time for early modular integration exposed mismatches in initialization order and function calls before they became larger issues. Going forward, we would fully develop and unit-test each subsystem in isolation, then bring them together in a focused integration sprint guided by a detailed Gantt chart with specific deliveries and deadlines. Our greatest hurdles, overlapping GPIO/timer assignments, tangled initialization routines, and iterative hardware revisions, were overcome by merging one module at a time using step-through debugging and by holding brief daily stand-up meetings to review progress and demonstrate new code. This approach not only accelerated troubleshooting but also fostered a more cohesive, adaptable team workflow.

Check out our GitHub Repository: [ryan-miller33/dawg-security-system: Calvin Friedrich | Ryan Miller Security System Project](https://github.com/ryan-miller33/dawg-security-system)

**References:**

**[1] Digi-Key Electronics, “Digi-Key Electronics,” Online:** [**https://www.digikey.com/**](https://www.digikey.com/) **[Accessed: May 15, 2025].**

**[2] Adafruit Learning System, “PIR Passive Infrared Proximity/Motion Sensor – How PIRS Work,” Online:** [**https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work**](https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work) **[Accessed: May 15, 2025].**

**[3] ScienceDirect, “Photocell or photoresistor,” Online:** [**https://www.sciencedirect.com/topics/engineering/photoresistors#:~:text=A%20photocell%20or%20photoresistor%20is,light%20will%20cause%20higher%20resistance**](https://www.sciencedirect.com/topics/engineering/photoresistors#:~:text=A%20photocell%20or%20photoresistor%20is,light%20will%20cause%20higher%20resistance) **[Accessed: May 15, 2025].**

**[4] GitHub, “GitHub: Where the world builds software,” Online:** [**https://github.com/**](https://github.com/) **[Accessed: May 15, 2025].**

**[5] STMicroelectronics, Programming manual: STM32 Cortex®-M4 MCUs and MPUs (PM0214 Rev 10), March 2020.**

**[6] STMicroelectronics, Discovery kit with STM32F407VG MCU – User manual (UM1472 Rev 7), October 2020.**

**[7] STMicroelectronics, STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs – Reference manual (RM0090 Rev 19), February 2021**

**[8] C. Priscilla, “SPI Protocol: A Complete Guide from Basics to Advanced,” Online:** [**https://pantech.ai/spi-protocol-a-complete-guide-from-basics-to-advanced**](https://pantech.ai/spi-protocol-a-complete-guide-from-basics-to-advanced) **[Accessed: May 16, 2025]**