Journal of Electrical Engineering, Electronics, Control and Computer Science – JEEECCS, Volume 11, Issue 40, pages 1-6, 2025

Design of an Arduino-Based Intelligent Access Control System with Priority Levels

1st Iorga Rusten-Andrei
Faculty of Electronics, Communications and Computers
National University of Science and Technology Politehnica Bucharest
Pitesti, Romania
iorgaandreir@gmail.com

2nd Bizon Nicu
Department of Electronics, Computers and Electrical Engineering
National University of Science and Technology Politehnica Bucharest
Pitesti, Romania
nicubizon@yahoo.com

3rd Dragusin Sebastian-Alexandru
Department of Electronics, Computers and Electrical Engineering
National University of Science and Technology Politehnica Bucharest
Pitesti, Romania
dragusin.sebi@yahoo.com

Abstract –This paper presents the design and validation of an intelligent, low-cost access control system built on an Arduino Mega platform that enforces multi-factor authentication and priority-based policies for institutional environments. The solution integrates RFID (Radio Frequency Identification) tags, fingerprint verification, and PIN (Personal Identification Number) entry via a keypad, augmented with a PIR (Passive Infra-Red) sensor and audible/visual indicators for tamper feedback. A modular firmware architecture coordinates an authentication pipeline with configurable retry limits, lockout intervals, and role/priority rules that gate sensitive actions (e.g., password change, RFID enrollment) behind elevated privileges. Hardware interfacing is described for the RC522 RFID reader, optical fingerprint sensor, 4×4 keypad, relay/servo door actuator, and an I²C (Inter-Integrated Circuit) LCD (Liquid Crystal Display), alongside the finite-state control logic and the data structures governing user records and policy enforcement. Experimental evaluation encompasses per-factor latency, success and false-reject rates, and robustness under fault scenarios (power loss, sensor mismatch, replay attempts). Results indicate sub-second on-device decisions, reliable operation across more than 1,000 transactions, and measurable security gains from combining factors with priority checks, while maintaining the bill of materials within an educational/SME (Small and Medium-sized Enterprises) budget. The architecture thus constitutes a reproducible reference for secure, scalable access control in laboratories, classrooms, and administrative facilities.

Keywords- multi-factor authentication; priority-based access control; security systems; sensors; low-cost embedded systems

I. INTRODUCTION

In modern industrial and institutional environments, MFA (multi-factor authentication) has become a baseline control for physical and logical access, yet persistent threats, such as credential phishing, relay/man-in-the-middle attacks, tailgating, and insider misuse, continue to expose gaps, underscoring the need for layered, priority-aware authorization, resilient device attestation, and continuous monitoring to secure end-to-end entry workflows [1].

Physical access control in small to medium-sized facilities (schools, laboratories, SMEs) must balance security, usability, and cost. Commercial solutions often provide strong security guarantees but at a price point and integration complexity that exceed educational or constrained budgets. Conversely, low-cost hobbyist builds tend to sacrifice policy expressiveness (e.g., role/priority handling), auditability, or reliability under real-world operating conditions [2].

This work addresses that gap by engineering an embedded, Arduino-based access control system that delivers multi-factor authentication and priority-aware authorization while remaining affordable, maintainable, and extensible. The proposed system integrates complementary sensing and actuation modalities: contactless identification (RFID), knowledge-based verification (PIN via keypad), and biometric confirmation (fingerprint), coordinated by a microcontroller platform with deterministic real-time behavior. A modular I/O (Input/Output) layer manages door hardware (relay/solenoid), status indicators (LED (Light-Emitting Diode) / buzzer / LCD), and environmental monitoring (e.g., motion sensing) to support tamper detection and user feedback. Critically,

authorization decisions are governed by a priority-based policy engine that encodes hierarchical roles (e.g., administrator, staff, visitor) and context-dependent overrides (e.g., emergency unlock), enabling fine-grained control without imposing undue latency on entry workflows.

From a systems perspective, the authors emphasize local autonomy for safety and availability (decisions occur on-device even during network loss), secure credential handling (rate limiting, lockout windows, and multi-attempt policies), and event logging for forensic analysis and compliance. The architecture is deliberately component-agnostic, allowing drop-in replacement of sensors, door actuators, or communication modules as requirements evolve, and it accommodates future upgrades such as encrypted telemetry, remote fleet management, and integration with attendance systems.

Contributions. This paper offers: (i) a reproducible hardware architecture for multi-modal, low-cost access control; (ii) a priority-aware authorization model implementable on resource-constrained MCUs; (iii) a robust firmware design with anti-tamper, lockout, and audit logging mechanisms; and (iv) an experimental evaluation covering authentication latency, success/denial behavior across policy tiers, and resilience under fault scenarios (power glitches, network unavailability).

Paper organization. Section II reviews related work on embedded access control, multi-factor authentication, and low-cost sensing. Section III details the hardware architecture (microcontroller platform, peripherals, and door interface). Section IV presents the system software design, including the priority-based policy engine and communication stack. Section V reports results from functional and stress testing in a lab setting. The conclusions summarize the findings and outline future work toward encrypted telemetry, centralized management, and expanded analytics

II. Literature Review

The authors [3] propose a microcontroller-centric IoT (Internet of Things) smart door lock that unifies voice commands, fingerprint biometrics, and PIN entry within a single control stack; the design integrates anti-tamper sensing and basic cryptographic protection for credentials/commands, and the prototype evaluation evidence functional correctness and improved usability relative to single-factor locks.

Vel'as et al. [4] present a rigorous, practice-oriented study on RFID-based access control "permeability," introducing a purpose-built test rig to quantify reader throughput and read reliability at varying stand-off distances, and framing results within EN 60839-11-1 risk classes and deployment environments; their experiments (6,000 trials) show a clear degradation of successful reads with increased distance, offering actionable parameters for security managers sizing lanes and shift changes.

The paper [5] surveys security in smart-home IoT, formalizing core goals: authentication, authorization, confidentiality, integrity, and availability and classifying threats into passive (eavesdropping/monitoring) and active (modification, DoS (Denial-of-Service), malware) attacks. It analyzes representative scenarios (e.g., energy-usage data leakage, smart-meter tampering, interference at HEMS (Home Energy Management System) / ESI (Energy Services Interface) interfaces, and manipulation of consumption reports), then derives corresponding requirements such as robust user/device authentication, network monitoring / IDS (Intrusion Detection System), message integrity protection, availability planning, and confidentiality via cryptography and key management. The authors also review countermeasures from the literature and even attempt coarse forecasts of cyber-attack trends to motivate future security architectures for smart homes.

Levshun et al. [6] present a security-by-design methodology for microcontroller-based systems that combines a set-based hierarchical relational model with four design algorithms (requirements formation, component composition, abstract model synthesis, and detailed model synthesis). Their attacker model parameterizes access, knowledge, and resources, which is then mapped to classes of attack actions and corresponding security controls; the approach aims to balance protection levels against resource constraints. A software implementation (Python + PostgreSQL + Tkinter) demonstrates the workflow on a perimeter-monitoring mobile-robot system, where they evaluate design time as a function of attacker parameters and discuss trade-offs and extensibility beyond specific platforms. The work emphasizes integrating security elements alongside hardware/software building blocks and link models early in the lifecycle to reduce architectural weaknesses.

[7] present a microcontroller-based, multi-layer security system for an industrial complex that integrates fire detection (photoelectric smoke sensing with LDR (Light-Dependent Resistor) and LM324 comparator) and temperature monitoring (LM35), intrusion detection and doorway people counting (IR (Infra-Red) transmitter/receiver beams), CCTV (Closed-Circuit Television) via IP (Internet Protocol) cameras, and an AFIS (Automated Fingerprint Identification System) for access control; a web portal provides remote, authenticated monitoring, while an 8051-family AT89C52 MCU (Microcontroller Unit) orchestrates the hardware modules. The paper details circuit-level designs for each subsystem, emphasizes affordability and modularity, and outlines future enhancements such as backup power and additional sensing modalities.

The Zenodo preprint [8] presents a low-cost, Arduino-based password-protected door-locking system that integrates an Arduino UNO with a 4×4 keypad for credential entry, an LCD for user feedback, a servo-driven bolt, and an optional GSM (Global System for Mobile Communications) module to notify owners upon repeated failed attempts; the workflow includes lockout after a configurable number of incorrect PINs and an on-device password-change routine, with design and validation carried out in Tinkercad and a comparative discussion against RFID, biometric, and smartphone locks, plus future work on encryption and two-factor schemes.

III. Control System Hardware Architecture

This section details the embedded platform that realizes the priority-aware, multi-factor access workflow. The design targets low cost, high availability, and maintainable wiring, while providing clean electrical interfaces for sensing, human input, actuation, and status feedback.

- **Central controller** – The system is built around an Arduino Mega 2560 (ATmega2560 MCU), chosen for its abundant GPIO (General-Purpose Input/Output), multiple hardware UARTs (Universal Asynchronous Receiver/Transmitter), and native SPI (Serial Peripheral Interface) / TWI (Two-Wire Interface) (I²C) buses. The Mega's on-board $5V$ regulator and USB (Universal Serial Bus) interface simplify development and bench testing; all field peripherals are referenced to a single ground domain to minimize measurement offsets [9].
- **Human input (PIN entry)** – A 4×4 matrix keypad connects to digital lines configured for row/column scanning. Internal pull-ups and debounced, time-windowed reads prevent ghosting and brute-force bursts; the firmware enforces lockout and attempt counters tied to user priority levels [10].
- **RFID credential (badge)** – An RC522 ($13.56MHz$) RFID reader is interfaced via SPI (SCK (Serial Clock), MISO (Master In, Slave Out), MOSI (Master Out, Slave In), SS (Slave Select)). SS is assigned to a dedicated Mega chip-select to avoid bus contention. The reader's $3.3V$ rail is decoupled locally ($100nF + 10\mu F$) and its IRQ (Interrupt Request) line is polled to shorten card-present latency [11].
- **Presence sensing (anti-tailgating)** – A PIR HC-SR501 motion sensor is connected to a digital input with an interrupt-capable pin for prompt wake and for logging entry/exit motion events. The module's retrigger and sensitivity are tuned to the doorway geometry to reduce false positives [12].
- **Biometric credential (fingerprint)** – The module (FPM10A) interfaces over a hardware UART (TX (Transmit) / RX (Receive), $5V$, GND (Ground)) and performs on-sensor enrollment, template storage, and 1:1/1:N matching, returning status codes to the MCU; firmware enforces attempt counters and lockout windows, while local decoupling and short twisted-pair routing on the UART reduce noise, keeping biometric templates off the main bus for better security and lowering MCU load [13].
- **User feedback** – A 20×4 I²C LCD provides real-time prompts and audit messages (SDA (Serial Data) / SCL (Serial Clock) on the Mega's TWI pins). Three status LEDs (GREEN/YELLOW/RED with $1k\Omega$ series resistors) indicate authentication state, priority evaluation, and fault/lockout conditions. An active buzzer (logic-level) delivers audible cues for success/error and for dwell-time warnings [14].

The complete electrical interconnection is depicted in Figure 1: Hardware schematic of the intelligent access controller, showing the Arduino Mega 2560, RC522 RFID reader over SPI (SCK/MISO/MOSI/SS), 4×4 keypad matrix, PIR input, FPM10A fingerprint sensor connected via UART (TX/RX/GND), I²C 20×4 LCD (SDA/SCL), status LEDs and active buzzer, plus power rails and auxiliary headers.

The fabricated controller used in experiments is shown in Figure 2: Assembled PCB (Printed Circuit Board) of the priority-aware access controller, integrating the 4×4 keypad, PIR sensor, FPM10A fingerprint module (UART), I²C 20×4 LCD, status LEDs, buzzer, Arduino Mega 2560, and the RC522 RFID module, with neatly routed SPI/I²C/UART harnesses for maintainable wiring.

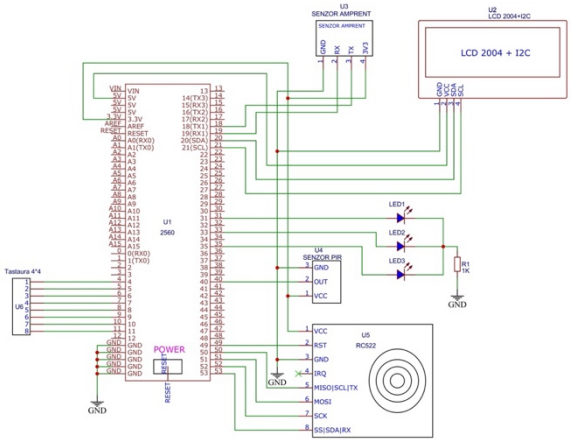

Figure 1. Hardware schematic of the intelligent access controller

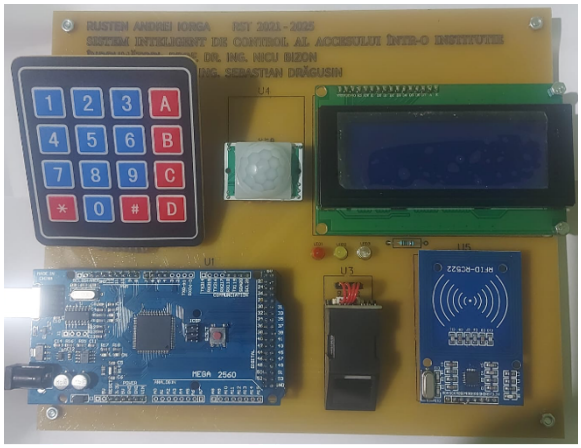

Figure 2. Assembled controller PCB used in testing

IV. System Software Design

The firmware was developed in Arduino IDE (Integrated Development Environment) (C/C++) following a modular, event-driven architecture with a non-blocking main loop and a compact state machine [15].

Hardware resources are abstracted through dedicated services: *KeypadSvc* (matrix scanning and debouncing), *RFIDSvc* (SPI session with MFRC522, UID (Unique Identifier) parsing and anti-replay

checks), *FingerSvc* (UART driver for the FPM10A reader, capture/match/enroll routines), *PirSvc* (interrupt-driven motion events with edge filtering), and *UiSvc* (LiquidCrystal_I2C display, tri-color status LEDs, and audible feedback via buzzer). Timing is orchestrated with *millis()* to avoid *delay()*, enabling concurrent polling of inputs, UI (User Interface) refresh, and enforcement of lockout windows. Sensitive artifacts: hashed PIN, authorized RFID UIDs, enrolled fingerprint IDs (Identifiers), attempt counters and CRC (Cyclic Redundancy Check) guards persisted in EEPROM (Electrically Erasable Programmable Read-Only Memory) to survive power cycles; a lightweight watchdog resets the device on stalls to guarantee availability. All serial links (USB console, UART to FPM10A) are rate-limited and framed to mitigate buffer overflows, while SPI/I²C transactions are wrapped with retries and backoff to tolerate transient bus errors.

From a control-policy perspective, the system implements a priority-aware multi-factor model with three authentication strata. **Level 1** (biometric) relies on the FPM10A fingerprint match and is considered the highest trust anchor; **Level 2** (knowledge) uses a keypad PIN validated against a salted hash; **Level 3** (possession) authenticates an RC522 RFID badge by UID and optional sector read. The main loop evaluates user intents and routes execution through four guarded flows. In the arming/monitoring flow, the controller enters a supervised mode in which PIR motion events are latched and reported with visual and acoustic alarms; disarming accepts any single valid factor to minimize operator friction, yet still records the credential used for audit. In the change-PIN flow, the current PIN must be supplied before a Level-1 biometric confirmation authorizes replacement; the operation is atomically committed to EEPROM after checksum verification. In the change-RFID flow, the currently registered tag must be presented, then a valid PIN confirms intent before the new UID is enrolled. In the enroll-fingerprint flow, a valid PIN followed by a valid tag elevates the session to permit biometric enrollment; two or three captures are averaged to improve template quality, with rollback on poor-quality frames.

Across all configuration flows, the firmware enforces three-attempt limits per factor with timed lockout windows and cool-down counters; these are implemented as monotonic timers to remain effective after resets. The LCD provides deterministic prompts and outcome messages, while LEDs (green/yellow/red) encode success, pending review, and fault/lockout states. All significant events like attempts, matches, failures, arm/disarm transitions are timestamped (relative time) and emitted on the serial audit channel for subsequent forensic analysis or integration with supervisory systems.

Figure 3 present top-down software flow for the multi-factor, priority-based access controller. The diagram shows system initialization, a non-blocking main loop, and four operational branches (arm/monitor, change PIN, change RFID, enroll fingerprint), each enforcing hierarchical verification: validating the current credential and then a higher-priority factor under three-attempt limits with timed lockouts.

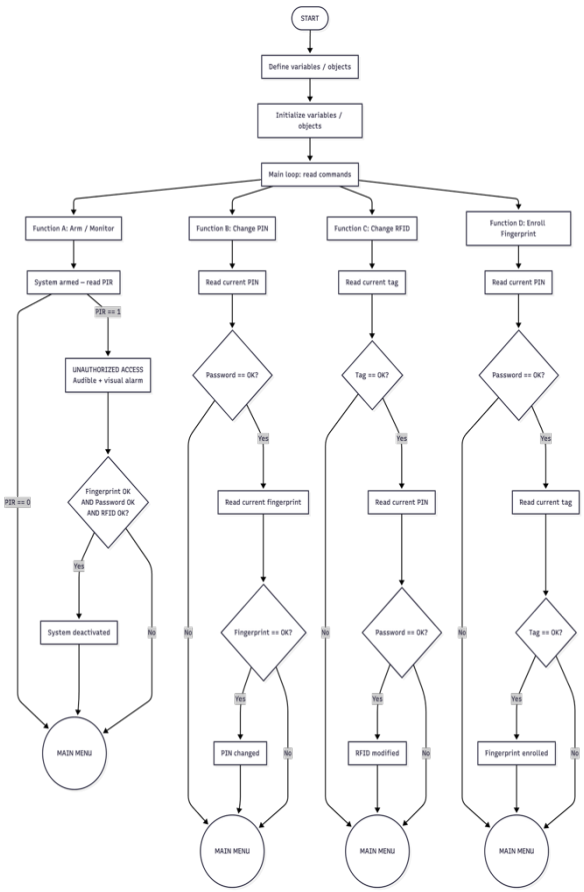

Figure 3. Software flowchart of the intelligent access control system

V. RESULTS

The proposed access-control prototype was evaluated through a staged campaign combining unit, integration, performance, regression, and in-situ tests. The evaluation verified correctness across the hardware–software boundary, quantified responsiveness under repeated use, and validated security behaviors, including attempt limits, timed lockouts, and enforcement of the privilege hierarchy.

In the present prototype, up to 162 biometric users can be enrolled (limited by the FPM10A sensor's on-device template store), while RFID/PIN user records are bounded by the ATmega2560's 4 KB EEPROM, e.g., with a 32-byte record format the controller accommodates 128 users, expandable via external I²C FRAM (Ferroelectric Random-Access Memory) if needed.

First, functional (unit) tests were executed on each module in isolation. The 4×4 keypad was scanned with debounced, non-blocking reads and verified for correct key codes and absence of ghosting. The RC522 reader correctly recognized authorized UIDs and rejected unknown tags. The fingerprint sensor identified previously enrolled templates and captured new ones reliably during enrollment. The 20×4 I²C LCD was checked for legibility and update latency, while buzzer and RGB (Red, Green, and Blue) status LEDs were exercised for success/error signaling. The PIR (HC-SR501) was tuned for sensitivity and retrigger timing and validated for stable motion events.

Integration tests confirmed inter-module coherence. Concurrent events were injected, for example, PIR motion during an authentication failure and state transitions were verified: the system enters alarm when armed and exits only upon presentation of a valid credential. LED states and acoustic cues aligned with the access scenario (authorized, unauthorized, or disarming).

Performance tests focused on latency and robustness. Alarm latency from PIR trigger to audible/visual alert was consistently $< 500ms$. Fingerprint verification and PIN processing times were measured over repeated trials to assess worst-case response under sustained use; the system maintained stable timings without UI jitter. Stress trials toggled arming/disarming and navigated menus rapidly to detect lockups, and none were observed.

To ensure stability after firmware changes, regression tests were executed after each refactoring (e.g., introducing fully non-blocking PIN entry). Menu integrity, message coherence, and peripheral compatibility were preserved, and no feature regressions were observed.

Real-world scenarios were executed to validate behavior under realistic conditions. An unauthorized passage in front of the PIR while the system was armed correctly raised the alarm. Sequences of invalid inputs: three incorrect PIN attempts followed by an unauthorized tag, triggered the expected lockouts and automatic return to the main menu. Administrative operations adhered to the hierarchical policy: changing the PIN required the current PIN plus a valid fingerprint; adding an RFID tag required the current PIN and a known authorized tag; enrolling a new fingerprint required completion of both prior steps. All authentication paths enforced three-attempt limits and timed lockouts, effectively mitigating brute-force trials.

Finally, non-functional assessments addressed usability and ergonomics. LCD readability remained adequate under varied ambient light; buzzer intensity was appropriate for local alerting; the physical layout on the PCB afforded easy keypad access and clear silkscreen labeling. Overall, the prototype demonstrated reliability, security, and low-latency operation, with coherent handling of complex access, lock, and update scenarios and with clear headroom for future extensions (e.g., encrypted telemetry and centralized logging).

Limitations include unsecured on-device storage (no secure element), absence of secure boot/firmware attestation, susceptibility of low-cost optical fingerprint sensors to certain presentation attacks, and cleartext local buses (SPI/I²C). Mitigations, TLS (Transport Layer Security)-protected telemetry, encrypted credential stores (secure element/ TPM (Trusted Platform Module)), rate-limiting with exponential backoff, liveness detection, and signed OTA (Over-The-Air) are outlined as future work.

This paper contributes: (i) a priority-aware, multi-factor policy that binds administrative operations to higher-trust factors (e.g., PIN change requires current PIN plus fingerprint); (ii) a non-blocking, fault-tolerant firmware architecture with attempt counters, timed lockouts, and watchdog-safe state transitions; (iii) a component-agnostic device abstraction layer enabling drop-in replacement of sensors/actuators; and (iv) a reproducible test protocol (unit, integration, performance, regression, and in-situ) with quantitative latency and reliability metrics.

As illustrated in Figure 4, a typical test stage captures the controller's end-to-end behavior: sensor input, decision logic, and user feedback, under controlled authentication scenarios.

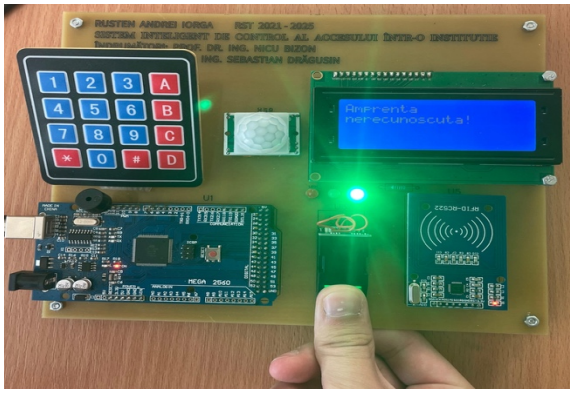

Figure 4. Representative result from a system test stage

CONCLUSION

This work has demonstrated a priority-aware, multi-factor access controller on Arduino Mega that fuses a fingerprint sensor (FPM10A, UART TX/RX), a 4×4 PIN keypad, and an RC522 RFID reader into a cohesive, low-cost embedded platform. The biometric factor operates as the top-tier authenticator: administrative actions (e.g., PIN change, RFID enrollment) require successful fingerprint verification followed by the lower-tier credential, enforcing strict hierarchical control. Functional and integration tests confirmed sub-second response to PIR events, reliable template matching on the FPM10A, correct UID recognition on RFID, and non-blocking user interaction on the I²C LCD with synchronized LED/buzzer feedback. Clean electrical partitioning (SPI/I²C/UART) and a unified ground domain yielded stable operation and maintainable wiring.

Limitations remain. On-device storage constrains audit depth and biometric template protection; the current prototype lacks secure boot and encrypted non-volatile storage, and EMI (Electromagnetic Interference) / ESD (Electrostatic Discharge) hardening is minimal [16]. Spoof-resilience for the fingerprint pipeline (e.g., presentation attack detection) has not been systematically characterized, and long-run behavior under brownouts or noisy supplies requires further study [17]. While local autonomy improves availability, the absence of centralized policy management and remote attestation limits multi-door deployments.

Future work targets three fronts. (i) Security & operations: adopt encrypted telemetry (MQTT (Message Queuing Telemetry Transport) or HTTPS (Hypertext Transfer Protocol Secure) / TLS with mutual auth), OTA updates, signed firmware, and encrypted/cancellable biometric templates on EEPROM / FRAM; add watchdog-gated recovery,

tamper switches, and structured audit export to SIEM (Security Information and Event Management). (ii) Biometric robustness: calibrate FMR/FRR (False Match Rate / False Reject Rate) trade-offs for the FPM10A, implement presentation-attack countermeasures (ridge/valley quality checks, time-of-contact dynamics, multi-capture voting), and evaluate spoof media (gelatin/silicone) in a controlled protocol; align template handling with recognized interchange formats. (iii) Analytics & fleet: introduce a central controller for role/attribute-based policies, time schedules, anti-passback, and anomaly detection over access logs; expose REST (Representational State Transfer) / MQTT APIs (Application Programming Interfaces) for attendance / HR (Human Resources) integration. In parallel, improve board-level resilience (filtering, shielding, surge protection, battery buffering and brownout handling) to raise overall confidentiality, integrity, availability, and auditability without sacrificing the system's cost and maintainability advantages.

REFERENCES

[1] S. A. Dragusin, N. Bizon, R. N. Bostinaru, F. M. Enescu, R. M. Teodorescu, and C. Savulescu, "Analysis of Vulnerabilities in Communication Channels Using An Integrated Approach Based on Machine Learning and Statistical Methods," *Proceedings of the 16th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2024*, 2024, doi: 10.1109/ECAI61503.2024.10607483.

[2] I. E. Kezron, "Cybersecurity Strategies for Resource Constrained SMEs and Health Providers," *Iconic Research And Engineering Journals*, vol. 8, no. 5, pp. 1215–1224, 2024, Accessed: Nov. 13, 2025. [Online]. Available: https://www.irejournals.com/paper-details/1706570

[3] A. Ahmed, S. Abdulkadir, J. Mohamed, S. A. Ali, M. Abdi, and S. A. Kahie, "Design and Implementation of an IoT based Smart Door Lock System," *2023 2nd International Conference on Multidisciplinary Engineering and Applied Science, ICMEAS 2023*, 2023, doi: 10.1109/ICMEAS58693.2023.10379324.

[4] A. Veľas, M. Boroš, R. Kuffa, and F. Lenko, "Testing of Permeability of RFID Access Control System for the Needs of Security Management," *Applied Sciences 2024, Vol. 14, Page 4227*, vol. 14, no. 10, p. 4227, May 2024, doi: 10.3390/APP14104227.

[5] W. Ali, G. Dustgeer, M. Awais, and M. A. Shah, "IoT based smart home: Security challenges, security requirements and solutions," *ICAC 2017 - 2017 23rd IEEE International Conference on Automation and Computing: Addressing Global Challenges through Automation and Computing*, Oct. 2017, doi: 10.23919/ICONAC.2017.8082057.

[6] D. Levshun, A. Chechulin, and I. Kotenko, "Design of Secure Microcontroller-Based Systems: Application to Mobile Robots for Perimeter Monitoring," *Sensors 2021, Vol. 21, Page 8451*, vol. 21, no. 24, p. 8451, Dec. 2021, doi: 10.3390/S21248451.

[7] N. David and G. Ajah, "A Microcontroller Based Security System," *Scholars Journal of Engineering and Technology*, vol. 2, no. SJET, pp. 868–873, 2014, Accessed: Nov. 13, 2025. [Online]. Available: www.saspublisher.com

[8] Harishni L, Srisha M, and Dr Sandhya S, "Design and Simulation of an Arduino-Based Password-Protected Door Locking System," *Int J Eng Adv Technol*, vol. 14, no. 2, pp. 25–31, Dec. 2024, doi: 10.35940/IJEAT.B4550.14021224.

[9] Arduino Docs, "Arduino® Mega 2560 Rev3." Accessed: Nov. 13, 2025. [Online]. Available: https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf

[10] SparkFun Electronics, "4x4 Matrix Membrane Keypad." Accessed: Nov. 13, 2025. [Online]. Available: https://cdn.sparkfun.com/assets/f/f/a/5/0/DS-16038.pdf

[11] NXP, "MFRC522 Standard performance MIFARE and NTAG frontend." Accessed: Nov. 13, 2025. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf

[12] HandsOn Tech, "HC-SR501 Passive Infrared (PIR) Motion Sensor." Accessed: Nov. 13, 2025. [Online]. Available: https://www.handsontec.com/dataspecs/SR501%20Motion%20Sensor.pdf

[13] Ja-bots.com, "Fingerprint Sensor Module with Arduino (FPM10A)." Accessed: Nov. 13, 2025. [Online]. Available: https://ja-bots.com/wp-content/uploads/2022/03/FPM10A-Guia-Arduino.pdf?srsltid=AfmBOoq4kY9Iirb1zUB2w-FjIIjnR7ibbOygSG0yiqd5DWIqSYkBYdjD

[14] HandsOn Tech, "I2C Serial Interface 20x4 LCD Module." Accessed: Nov. 13, 2025. [Online]. Available: https://www.handsontec.com/dataspecs/I2C_2004_LCD.pdf

[15] D. K. Halim, T. C. Ming, N. M. Song, and D. Hartono, "Arduino-based IDE for embedded multi-processor system-on-chip," *Proceedings of 2019 5th International Conference on New Media Studies, CONMEDIA 2019*, pp. 135–138, Oct. 2019, doi: 10.1109/CONMEDIA46929.2019.8981862.

[16] S. A. Dragusin, N. Bizon, and R. N. Bostinaru, "A Brief Overview Of Current Encryption Techniques Used In Embedded Systems: Present And Future Technologies," *15th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2023 - Proceedings*, 2023, doi: 10.1109/ECAI58194.2023.10194034.

[17] S. A. Dragusin, N. Bizon, and R. N. Bostinaru, "Comprehensive Analysis of Cyber-Attack Techniques and Vulnerabilities in Communication Channels of Embedded Systems," *Proceedings of the 16th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2024*, 2024, doi: 10.1109/ECAI61503.2024.10607432.