



Design and implementation of an Arduino-based automated barricading and intelligent access-control system

Dr. Ranjan Kumar Singh¹, Jyoti Sharma², Gagan³

¹ Professor, Global Institute of Technology and Management (GITM), Gurugram, Haryana, India

² Assistant Professor, Global Institute of Technology and Management (GITM), Gurugram, Haryana, India

³ Global Institute of Technology and Management (GITM), Gurugram, Haryana, India

DOI: <https://doi.org/10.66856/njmrd.2025.10.2.11084>

Abstract

The increasing demand for secure and efficient access management in residential complexes, institutional premises, industrial facilities, and parking areas has encouraged the development of low-cost automated barricading systems. This paper presents the design and implementation of an Arduino-based automated barricading and intelligent access-control system. The proposed system combines radio-frequency identification (RFID) for user authentication, ultrasonic sensors for vehicle detection and obstacle monitoring, a servomotor-driven barrier mechanism, and a liquid-crystal display for system-status indication. An Arduino Uno based on the ATmega328P microcontroller coordinates the sensing, authentication, decision-making, and actuation processes. A finite-state-machine control algorithm is implemented to manage system states, including idle, authentication, barrier opening, vehicle passage, barrier closing, and fault conditions. The proposed design emphasizes affordability, modularity, ease of maintenance, and operational safety. Prototype testing demonstrates that the system can authenticate registered RFID cards, operate the barricade automatically, and prevent barrier closure when an obstacle is detected. The system is suitable for small-scale access-control applications and can be extended through wireless communication, license-plate recognition, cloud logging, and mobile-based authorization.

Keywords: Arduino Uno, automated barricade, access control, RFID, ultrasonic sensor, servomotor, embedded system, finite-state machine

Introduction

Access-control systems are used to regulate the movement of people and vehicles into restricted areas. Conventional access-control arrangements commonly rely on security personnel, manually operated gates, or mechanical locks. Although these methods are simple to deploy, they may result in increased labour requirements, delayed access verification, inconsistent operation, and limited monitoring capability.

Automated barricading systems address these limitations by combining electronic authentication with electromechanical gate control. Such systems are widely used in parking facilities, residential communities, schools, offices, warehouses, and industrial sites. Commercial systems, however, may be expensive and difficult to customize for small organizations or educational applications. A low-cost microcontroller-based solution can provide the essential functions of authentication, barrier actuation, vehicle detection, and obstruction protection at a substantially lower cost.

This work presents the design and implementation of an automated barricading and intelligent access-control system using an Arduino Uno microcontroller. The system uses an RFID reader to identify authorized users and ultrasonic sensors to detect vehicles and obstacles. When a valid RFID card is detected, the controller activates a servomotor to raise the barricade. The barrier remains open while a vehicle is detected in the passage area and closes only after the passage is clear. During closing, the system continuously monitors for obstacles and reopens the barrier if an object is detected.

The main contributions of this work are as follows:

1. Design of a low-cost, modular, and programmable automated barricade system.
2. Integration of RFID-based authentication with ultrasonic vehicle and obstacle detection.
3. Development of a finite-state-machine control algorithm for reliable operation.
4. Implementation of an obstruction-detection mechanism to improve user and vehicle safety.
5. Evaluation of the prototype in terms of authentication, response, actuation, and safety performance.

Related Work

Automated access-control systems have been implemented using several technologies, including passwords, keypads, RFID cards, biometric authentication, Bluetooth communication, and automatic number-plate recognition. Keypad-based systems are inexpensive but require users to remember passwords and may be vulnerable to password disclosure. Biometric systems provide stronger identity verification but generally require more expensive sensors and additional processing capability.

RFID-based access control is attractive for small and medium-scale installations because it provides contactless identification with relatively simple hardware. Passive RFID cards do not require an internal power source and can be issued to authorized users at low cost. However, RFID authentication alone does not verify whether a vehicle has safely passed through the barricade. Therefore, vehicle-presence and obstruction sensors are required.

Ultrasonic sensors are commonly used for short-range object detection because they are inexpensive, easy to interface with microcontrollers, and relatively independent

of visible lighting conditions. Their performance may be affected by target shape, surface texture, temperature, and acoustic interference. These limitations can be reduced through appropriate sensor placement, filtering, and repeated measurements.

The proposed system combines RFID authentication and ultrasonic sensing in a single embedded architecture. Compared with more advanced vision-based systems, the proposed design requires less computational power and is easier to implement in resource-constrained environments.

System Requirements

The system was designed according to the following functional requirements:

1. Detect the presence of an RFID card.
2. Compare the card identification number with a stored authorization list.
3. Open the barricade when an authorized card is detected.
4. Keep the barricade open while a vehicle is present in the passage area.
5. Detect obstacles during barrier closing.
6. Reopen or stop the barricade when an obstruction is detected.
7. Indicate system status using LEDs, a buzzer, and an LCD.
8. Deny access to unauthorized RFID cards.
9. Return automatically to the secure state after a completed passage.
10. Provide a platform that can be expanded with wireless communication and event logging.

Proposed System Architecture

The proposed system consists of five primary subsystems:

1. Authentication subsystem.
2. Sensing subsystem.
3. Processing and control subsystem.
4. Barrier actuation subsystem.
5. User-interface and indication subsystem.

The overall operation is coordinated by an Arduino Uno microcontroller. The RFID reader supplies the card identification number to the controller. Ultrasonic sensors detect vehicle presence and obstacles. Based on the authentication result and sensor conditions, the controller drives the servomotor to open or close the barricade.

a. Hardware Architecture

The major components of the prototype are listed in Table I.

Table 1: Components of the Prototype

Component	Function
Arduino Uno	Central processing and control
ATmega328P	Embedded microcontroller on Arduino Uno
MFRC522 RFID reader	Contactless card identification
RFID cards or tags	User credentials
HC-SR04 ultrasonic sensors	Vehicle and obstacle detection
Servo motor	Barrier-arm actuation
16 × 2 LCD	Status and user messages
Red and green LEDs	Access-status indication
Buzzer	Audible notification
External power supply	Power for controller and actuator
Barrier arm	Physical access-control mechanism

The MFRC522 RFID module communicates with the Arduino Uno through the serial peripheral interface (SPI). The ultrasonic sensors are connected through digital trigger and echo pins. The servomotor is controlled by a pulse-width modulation signal. The LCD may be connected through an I2C interface to reduce the number of required microcontroller pins.

b. Authentication Subsystem

The RFID reader operates at **13.56 MHz** and detects compatible passive RFID cards. Each card has a unique identifier (UID). The UID is read by the Arduino and compared with a list of authorized UIDs stored in program memory or EEPROM.

The authentication process is expressed as

$$A = \begin{cases} 1, & \text{if } UID \in \mathcal{U}_{\text{authorized}} \\ 0, & \text{otherwise,} \end{cases}$$

Where A is the authentication result and $\mathcal{U}_{\text{authorized}}$ is the set of authorized RFID identifiers.

If $A = 1$, the controller permits access and initiates the barrier-opening sequence. If $A = 0$, access is denied and the system remains in the secure state.

c. Vehicle-Detection Subsystem

Two ultrasonic sensors are used in the proposed system. The first sensor detects the approach or presence of a vehicle, while the second sensor is positioned near the barricade to detect obstacles during operation.

The distance measured by an ultrasonic sensor is calculated using

$$d = \frac{v_{\text{sound}} t_{\text{echo}}}{2},$$

Where d is the distance between the sensor and the object,

v_{sound} is the speed of sound, and t_{echo} is the measured echo duration. The factor of two accounts for the outgoing and returning acoustic paths.

At approximately **20°C**, the speed of sound is approximately **343 m s⁻¹**. In practical implementations, the distance can be approximated in centimeters by

$$d_{\text{cm}} \approx \frac{t_{\text{echo}}}{58},$$

where t_{echo} is expressed in microseconds.

An object is considered present when

$$d \leq d_{\text{threshold}},$$

Is selected experimentally according to the physical arrangement of the sensors.

d. Barrier-Actuation Subsystem

A servo motor is used to rotate the barrier arm between the closed and open positions. In the prototype, the closed position is represented by approximately **0°**, while the open position is represented by approximately **90°**.

The controller does not immediately move the servo to the final position. Instead, the angular command is incremented

gradually to reduce mechanical shock. If θ_k represents the commanded angular position at sampling instant k , the position update may be written as

$$\theta_{k+1} = \begin{cases} \min(\theta_k + \Delta\theta, \theta_{\text{open}}), & \text{opening,} \\ \max(\theta_k - \Delta\theta, \theta_{\text{closed}}), & \text{closing.} \end{cases}$$

Here, $\Delta\theta$ is the angular increment applied during each control cycle.

Control-System Design

a. Finite-State-Machine Model

A finite-state machine was used to organize the system behaviour. The primary states are

$$S = \{S_{\text{idle}}, S_{\text{auth}}, S_{\text{opening}}, S_{\text{hold}}, S_{\text{closing}}, S_{\text{fault}}\}.$$

The states are defined as follows:

- S_{idle} : The barrier is closed and the system waits for an RFID card.
- S_{auth} : The detected UID is validated.
- S_{opening} : The barrier is raised.
- S_{hold} : The barrier remains open while vehicle passage is monitored.
- S_{closing} : The barrier is lowered after the passage area becomes clear.
- S_{fault} : An abnormal condition, timeout, or repeated obstruction is detected.

The principal transitions are:

1. $S_{\text{idle}} \rightarrow S_{\text{auth}}$ When an RFID card is detected.
2. $S_{\text{auth}} \rightarrow S_{\text{opening}}$ When the UID is authorized.
3. $S_{\text{auth}} \rightarrow S_{\text{idle}}$ When the UID is unauthorized.
4. $S_{\text{opening}} \rightarrow S_{\text{hold}}$ When the open position is reached.
5. $S_{\text{hold}} \rightarrow S_{\text{closing}}$ After the vehicle has cleared the passage.
6. $S_{\text{closing}} \rightarrow S_{\text{idle}}$ When the closed position is reached.
7. $S_{\text{closing}} \rightarrow S_{\text{opening}}$ When an obstacle is detected.
8. Any state may transition to S_{fault} if a timeout or sensor failure is detected.

b. Barrier-Closing Safety Logic

The barrier is not allowed to close when an object is located inside the safety region. Let d_o be the distance measured by the obstacle sensor. The closing command is permitted only when

$$d_o > d_{\text{safe}}.$$

If

$$d_o \leq d_{\text{safe}},$$

The controller immediately stops the closing motion and commands the barrier to reopen. A time delay is also

applied before closing to ensure that a vehicle has fully cleared the passage.

The clearance condition can be expressed as

$$C = \begin{cases} 1, & d_v > d_{\text{clear}} \text{ continuously for } t_{\text{clear}}, \\ 0, & \text{otherwise,} \end{cases}$$

Where d_v is the measured vehicle distance, d_{clear} is the clearance threshold, and t_{clear} is the required clearance duration.

c. Fault Handling

Fault conditions may include:

- Failure to detect the RFID reader.
- Invalid or corrupted sensor readings.
- Excessively long barrier movement.
- Continuous obstacle detection.
- Loss of actuator response.
- Repeated unauthorized access attempts.

In a fault condition, the controller keeps the barrier in a safe position, activates the buzzer, and displays a warning message. A manual reset or authorized maintenance procedure may be required before normal operation resumes.

Software Implementation

The firmware was implemented using the Arduino Integrated Development Environment. The program is organized into initialization, sensing, authentication, decision-making, actuation, and fault-handling routines.

A simplified control algorithm is given below:

Text

Initialize RFID reader, ultrasonic sensors, LCD, LEDs, buzzer, and servo

Set barrier position to CLOSED

Set system state to IDLE

Repeat continuously:

Measure vehicle and obstacle distances

If state is IDLE:

Scan for RFID card

If card is detected:

State = AUTH

If state is AUTH:

Read card UID

If UID is authorized:

Display ACCESS GRANTED

State = OPENING

Else:

Display ACCESS DENIED

Activate warning indication

State = IDLE

If state is OPENING:

Increment servo angle

If open angle is reached:

State = HOLD

If state is HOLD:

Monitor vehicle passage

If passage is clear for the required time:

State = CLOSING

If state is CLOSING:

If obstacle is detected:

Stop closing

State = OPENING

Else:

Decrement servo angle

If closed angle is reached:

State = IDLE

If a timeout or sensor fault occurs:

Activate alarm

State = FAULT

A non-blocking software structure is preferred so that RFID scanning, sensor monitoring, display updates, and actuator control can operate without unnecessary delays. The use of timer-based scheduling also improves responsiveness during barrier closing.

Prototype Implementation

A prototype was assembled on a laboratory-scale barricade model. The RFID reader was mounted near the entry point so that an authorized user could present a card without physical contact. One ultrasonic sensor was placed before the barrier to detect an approaching vehicle, while the second sensor was positioned within the barrier zone to detect obstacles.

The servomotor was mechanically coupled to the barrier arm. The actuator power supply was separated from the microcontroller supply to reduce voltage fluctuations caused by motor current. A common ground was maintained between the controller and actuator supply.

The LCD displayed messages such as:

- System ready
- Scan rfid card
- Access granted
- Access denied
- Barrier opening
- Vehicle passing
- Obstacle detected

The green LED indicated valid access, while the red LED indicated denied access or a fault condition.

Experimental Methodology

The prototype should be evaluated using repeatable tests rather than isolated demonstrations. The following tests were performed or are recommended for validation:

1. **Authorized-card test:** Present each registered RFID card and record successful authentication.
2. **Unauthorized-card test:** Present unregistered cards and verify that the barrier remains closed.
3. **Read-range test:** Measure the maximum reliable distance between the RFID card and reader.
4. **Actuation test:** Measure the time required to open and close the barrier.
5. **Vehicle-clearance test:** Verify that the barrier remains open while a vehicle is detected.

6. **Obstacle test:** Place an object beneath the barrier during closing and verify reopening.
7. **Repeated-cycle test:** Operate the system for multiple cycles to identify intermittent faults.
8. **Power-interruption test:** Interrupt and restore power to determine whether the barrier returns to a safe state.

The authentication success rate can be calculated as

$$\eta_{\text{auth}} = \frac{N_{\text{successful}}}{N_{\text{attempts}}} \times 100\%.$$

The obstacle-detection accuracy can be calculated as

$$\eta_{\text{obstacle}} = \frac{N_{\text{correct detections}}}{N_{\text{actual obstacle events}}} \times 100\%.$$

The mean response time is

$$\bar{t} = \frac{1}{N} \sum_{i=1}^N t_i$$

Where t_i is the response time measured during the i test and N is the total number of trials.

Because measured performance depends on the specific servo, mechanical barrier, sensor placement, power supply, and test environment, numerical results should be reported only from actual experimental measurements. The following table provides a suitable format for reporting results.

Results and Discussion

The implemented prototype demonstrates the feasibility of integrating authentication, sensing, and actuation in a single Arduino-based control system. Valid RFID cards initiate the opening sequence, whereas invalid cards do not cause any barrier movement. This provides a basic level of access restriction suitable for controlled environments.

The use of a finite-state-machine architecture improves the clarity and reliability of the firmware. Each operational state has a defined purpose and a limited set of valid transitions. This reduces the possibility of contradictory actuator commands, such as opening and closing the barrier simultaneously.

The ultrasonic sensors provide an inexpensive method for monitoring vehicle presence and obstacles. However, sensor readings may fluctuate because of vehicle shape, sensor alignment, environmental temperature, and acoustic reflections. To reduce false transitions, the controller should use multiple readings, median filtering, or a short confirmation interval before changing state.

The barrier safety mechanism is particularly important. Authentication alone cannot guarantee safe operation because an authorized vehicle or pedestrian may still remain under the barrier. Continuous obstacle monitoring during the closing sequence therefore provides an additional safety layer.

The major limitations of the prototype are its short-range RFID communication, dependence on ultrasonic sensor performance, limited user database capacity, and absence of encrypted communication. In addition, the use of an RFID

UID as the sole credential may not be sufficient for high-security applications because some low-cost RFID systems are vulnerable to UID cloning. For critical installations, the system should be extended with cryptographic authentication, a secure database, or a second authentication factor.

Security and Reliability Considerations

Although the proposed system is intended for low-cost applications, several design considerations are necessary for reliable deployment.

First, authorized RFID identifiers should not be stored only in publicly accessible program code. They should be stored in protected memory, and access to the programming interface should be restricted. Second, the controller should distinguish between a temporary sensor error and a confirmed obstacle. Third, the actuator should have mechanical end stops or position feedback to prevent excessive torque at the ends of travel.

A watchdog timer can be enabled to recover the controller from firmware lockups. Brownout detection can also be used to reset the microcontroller when the supply voltage falls below a safe operating level. In larger installations, event records should be transmitted to a central server for auditing and maintenance.

The system should also include a manual release mechanism to allow emergency evacuation or maintenance access during power failure. For outdoor use, the electronics should be enclosed in a weather-resistant housing and protected against moisture, dust, and electromagnetic interference.

Cost and Scalability

The proposed architecture is suitable for small-scale installations because it uses commercially available components and open-source development tools. The design is modular, allowing individual components to be replaced or upgraded without redesigning the entire system.

Possible extensions include:

- ESP32-based Wi-Fi or Bluetooth connectivity.
- GSM-based remote alerts.
- Cloud-based access logs.
- Keypad or biometric secondary authentication.
- Automatic number-plate recognition.
- Solar-powered operation.
- Battery backup.
- Multiple barriers managed by a central controller.
- Web-based authorization management.
- Secure over-the-air firmware updates.

For large-scale systems, the Arduino Uno may be replaced by a more capable microcontroller or industrial programmable logic controller. Nevertheless, the proposed design provides a useful foundation for educational prototypes, small parking areas, laboratories, and controlled residential entrances.

Conclusion

This paper presented the design and implementation of an Arduino-based automated barricading and intelligent access-control system. The system combines RFID-based user authentication, ultrasonic vehicle and obstacle detection, servomotor-based barrier actuation, and finite-state-machine control. The architecture is inexpensive, modular, and relatively simple to implement.

The proposed design improves upon manually operated barricades by reducing operator involvement and providing automated access decisions. The obstacle-detection mechanism also improves operational safety by preventing the barrier from closing when a vehicle or object is detected in the protected region.

The prototype is appropriate for low- and medium-security applications. However, the use of unencrypted RFID identifiers and basic ultrasonic sensing limits its suitability for high-security installations. Future work will focus on cryptographic RFID authentication, wireless monitoring, cloud-based event logging, improved position feedback, and integration with automatic number-plate recognition. Experimental results should be reported using measurements collected from the completed prototype so that the performance claims remain reproducible and technically valid.

References

1. Arduino. Arduino Uno Rev3 Technical Specifications, Arduino Documentation, accessed, 2026.
2. Microchip Technology Inc. ATmega328/P 8-Bit AVR Microcontroller with 32K Bytes In-System Programmable Flash, datasheet.
3. NXP Semiconductors. MFRC522 Standard Performance MIFARE and NTAG Frontend, product data sheet.
4. Finkenzeller K. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, 3rd ed. Hoboken, NJ, USA: Wiley, 2010.
5. HCSR04. Ultrasonic Ranging Module HC-SR04 Datasheet.
6. Bishop RH. The Mechatronics Handbook, 2nd ed. Boca Raton, FL, USA: CRC Press, 2008.
7. Boxall J. Arduino Workshop: A Hands-On Introduction With 65 Projects. San Francisco, CA, USA: No Starch Press, 2013.
8. Pressman RS, Maxim BR. Software Engineering: A Practitioner's Approach, 9th ed. New York, NY, USA: McGraw-Hill, 2019.
9. International Organization for Standardization. ISO/IEC 14443: Identification Cards—Contactless Integrated Circuit Cards—Proximity Cards, ISO/IEC Standard.
10. IEEE. IEEE Editorial Style Manual, IEEE Publication Services and Products Board, Piscataway, NJ, USA.