Jurnal Vocational Teknik Elektronika dan Informatika
http://ejournal.unp.ac.id/index.php/voteteknika
Universitas Negeri Padang, Indonesia

ISSN 2302-3295 (Print)
ISSN 2716-3989 (Electronics)
Vol. 13, No. 4, 2025

Performance Analysis of an ESP32-Based Smart Home System: A Laboratory Scale Study

Ryan Fikri*[1], Agariadne Dwinggo Samala[2], Delsina Faiza[3], Almasri[4]

[1,2,3,4]Universitas Negeri Padang, Padang, Indonesia

**Corresponding Author*: ryanfikri@ft.unp.ac.id

Abstract - ***This study developed and evaluated an integrated Internet of Things (IoT)-based access control system that unifies residential gate operation, garage door automation, and garage lighting within a single platform. Conventional home automation deployments typically address only one function at a time, resulting in fragmented control and limited interaction between subsystems. Compared to prior smart home implementations that focus on single subsystems, the proposed platform jointly manages gate, garage door, and lighting and reports scenario-based reliability metrics. To address this gap, the proposed system used an ESP32 microcontroller as the central controller, ultrasonic distance sensors for vehicle detection, servo motors for mechanical actuation, and a relay module for lighting control. A mobile application built with the Blynk platform provided real-time monitoring and remote control via smartphone. The prototype was tested under three scenarios-manual mobile control, fully automatic sensor-triggered operation, and combined operation-with 20 repeated cycles per scenario. Performance metrics included servo actuation time, communication latency between the mobile application and the ESP32, sensor accuracy, and operational reliability. The gate and garage door achieved opening times of approximately 1.0-1.2 s and 1.5 s, respectively, while end-to-end communication latency remained between 300 ms and 480 ms across all tests. Ultrasonic distance measurements showed a maximum error of 1.8 cm and an average error below 1.2 cm, with no system failures in any scenario. These results demonstrate that the integrated design is technically feasible, reliable, and suitable as a cost-effective foundation for residential access control. Future work will focus on scaling the prototype toward full-size installations and extending integration with additional smart home services.***

Keywords—***Internet of Things (IoT), ESP32, Blynk, Smart home, Access control.***

I. INTRODUCTION

The rapid growth of the Internet of Things (IoT) has enabled pervasive connectivity and intelligent services across industrial, urban, healthcare, and residential domains [1]-[3]. By combining embedded sensing, wireless communication, and cloud or edge computing, IoT platforms support real-time monitoring and control with relatively low hardware and deployment costs [1], [2]. In built environments, sensors and microcontrollers are widely used to improve comfort, safety, and energy efficiency by capturing environmental and behavioral data [3]-[5].

In the smart home context, IoT-based solutions have been proposed for automation of lighting, appliances, and security systems [6]-[8]. Smart lighting and energy management systems integrate networked sensors and actuators to adapt indoor illumination, air quality, and ventilation to occupancy and environmental conditions [7]-[9]. At a larger scale, smart campus and smart building deployments illustrate how IoT architectures can coordinate multiple services and devices through unified monitoring and control platforms [10]-[13]. These developments demonstrate the potential of IoT to provide fine-grained control across heterogeneous devices in residential settings.

For individual households, many studies have focused on specific subsystems, such as smart fences and gates, door locks, and room-level automation. Smart home deployments include secure automation and monitoring architectures [6], sustainable smart fences using scanning methods [14], and intelligent door locks based on computer vision or sensor-enhanced authentication [15], [16]. Other works analyze vulnerabilities and attack surfaces in connected households through penetration testing [17] and propose hybrid authorization models tailored to smart home IoT. Parallel efforts explore ESP32-based monitoring and general automation frameworks with mobile or web interfaces [18], [19]-[23], as well as Android- and voice-controlled systems for elderly or general users [22]-[29].

DOI: https://doi.org/10.24036/voteteknika.v13i4.136492

Received	:	2025-08-07
Revised	:	2025-11-14
Accepted	:	2025-11-19
Published	:	2025-12-29

Table 1 summarizes representative categories of smart home and access-control studies, together with their primary focus and typical limitations.

Table 1. Representative smart home and access control studies

Category	Representative focus	Typical limitation
Smart lighting and energy systems	IoT-based lighting, energy, and ventilation	Limited integration with physical access mechanisms
Smart door/gate and fence systems	Door locks, gate/fence control and security	Often standalone, not linked to indoor automation
General home automation frameworks	Multi-appliance control and monitoring	Room- or appliance-centric, limited access integration

Although these implementations demonstrate the feasibility of IoT-based automation, they frequently remain function-specific and fragmented. Smart lighting and energy systems are often deployed independently of gate and garage door controls [7]-[9], [14], whereas smart door locks and fence solutions rarely integrate with indoor lighting or broader home automation frameworks [14]-[16]. As a result, homeowners may rely on multiple applications and controllers, leading to inconsistent user experiences and limited coordination between subsystems during daily use.

Security and access control further complicate smart home design. Studies have shown that connected households can expose significant attack surfaces due to weak authentication, insecure communication protocols, or misconfigured devices [6], [17]. Hybrid authorization models and secure access-control schemes have been proposed to address these challenges, and LoRa- or cloud-based architectures have been explored to improve communication reliability and coverage [6]. Nevertheless, many practical systems still prioritize basic functionality over holistic security, tight integration, and systematic performance evaluation under realistic operating scenarios [17], [20], [24].

Low-cost Wi-Fi-enabled microcontrollers such as the ESP32 have emerged as attractive platforms for smart home monitoring and control [18], [21]-[23]. ESP32-based devices support real-time sensing, local processing, and Internet connectivity, and have been used for home monitoring [18], energy monitoring, and multi-device IoT deployments [23]. Mobile and voice interfaces further increase usability for everyday users [21], [22], [27], [29]. However, most existing systems remain constrained to specific rooms, appliances, or single-function services and do not provide tightly integrated control over residential access points—such as gates and garage doors—alongside context-aware lighting within a single IoT architecture. In addition, performance evaluations are often limited to qualitative demonstrations or a narrow set of test conditions, offering limited insight into actuation time, communication latency, sensing accuracy, and operational reliability across different modes of operation.

In summary, there is a lack of IoT-based smart home solutions that (i) jointly manage gate, garage door, and lighting as an integrated residential access system; and (ii) quantitatively evaluate the resulting platform under realistic operating scenarios. To address these gaps, this paper proposes an integrated IoT-based control system that unifies gate operation, garage door automation, and garage lighting within a single ESP32-based platform accessed through a Blynk mobile application. Ultrasonic distance sensors are employed to detect vehicle presence and trigger automated access sequences, while servo motors and relay modules perform mechanical actuation and lighting control. The system supports manual mobile control, fully automatic sensor-triggered operation, and a combined mode that leverages both approaches.

The main contributions of this work are as follows:

1) The design and implementation of a low-cost IoT architecture that integrates gate, garage door, and lighting control into a unified ESP32-based smart home access system;
2) The development of a Blynk-based mobile interface that enables real-time monitoring and multi-mode control suitable for everyday residential use; and
3) An experimental evaluation under representative operating scenarios, quantifying end-to-end communication latency, actuation time, sensor accuracy, and operational reliability over repeated test cycles.

These contributions demonstrate the technical feasibility of integrated residential access and lighting control using affordable IoT hardware and highlight opportunities for future extensions in scalability, security, and interoperability with broader smart home ecosystems.

II. METHOD

A. Research Framework

The work developed and evaluated an integrated IoT-based residential access control system that combined three subsystems: front gate control, garage door automation, and garage lighting control. A NodeMCU ESP32 board served as the central controller and interfaced with an ultrasonic distance sensor, two servo motors, and a relay module. A mobile application built with the Blynk platform provided user interaction and enabled remote monitoring and control via Wi-Fi. The overall concept followed common IoT-based smart home architectures that integrate sensing, actuation, and networking in a single platform [19], [21], [25].

The research framework consisted of four main stages: system design, prototype implementation, functional verification, and performance evaluation. System design defined the hardware and software architecture, including sensor placement, actuator configuration, and communication flows between the ESP32, the Blynk cloud, and the mobile application. Implementation covered hardware assembly and development of the ESP32 firmware and Blynk dashboard. Functional verification checked the behavior of each subsystem in manual, automatic, and combined modes. Performance evaluation used defined test scenarios to quantify actuation time, communication latency, sensor accuracy, and operational reliability.

B. System Architecture

The overall system architecture is illustrated in Fig. 1. A NodeMCU ESP32 module acts as the central processing and communication unit. It connects to the home Wi-Fi network and establishes a session with the Blynk cloud server. The user interacts with the system through the Blynk mobile application, which sends control commands and receives real-time status updates via the cloud, as is common in recent IoT-based home automation platforms [21], [22], [27], [29].

On the device side, the ESP32 reads the distance from an ultrasonic sensor mounted near the driveway. Based on this measurement and the selected operating mode, it generates control signals for two servo motors that actuate the gate and garage door mechanisms in the prototype. A relay module controlled by a digital output pin drives the garage lighting circuit, allowing the ESP32 to switch the lamp on or off. A low-voltage DC power supply powers all components, while the relay contacts are used to interface with the mains-powered lamp.

In terms of data flow, user commands generated on the smartphone are transmitted to the Blynk cloud and forwarded to the ESP32 over Wi-Fi. The ESP32 processes these commands together with sensor readings and internal state information, then updates actuator outputs accordingly. In the opposite direction, the ESP32 periodically sends sensor values and subsystem states back to the Blynk dashboard so that the user can monitor the current condition of the gate, garage door, and lighting [18], [21], [23].

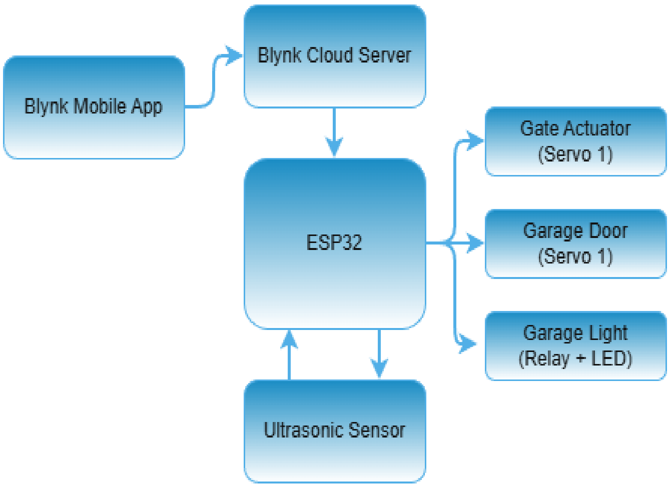

Fig. 1 Overall architecture of the integrated gate, garage door, and lighting control system.

The system supports three operating modes:

1) Manual mode: the user directly commands gate, garage door, and light operation from the mobile application;
2) Automatic mode: vehicle presence detected by the ultrasonic sensor automatically triggers a predefined access sequence;
3) Combined mode: both automatic triggers and manual commands are enabled, allowing the user to override or complement automatic behavior.

C. Hardware Design

The hardware design is organized into four functional modules: central control, sensing, actuation, and power/interface. Table 2 lists the main components used in the prototype.

Table 2. Main hardware components of the prototype

Module	Component	Function
Central control	NodeMCU ESP32 development board	Wi-Fi-enabled microcontroller, main controller
Sensing	Ultrasonic distance sensor	Vehicle/presence detection near gate/garage
Actuation	Servo motor 1 (gate mechanism)	Mechanical actuation of gate prototype
	Servo motor 2 (garage door mechanism)	Mechanical actuation of garage door prototype
	Single-channel relay module	On/off control of garage lighting
Power/interface	DC power supply, wiring, connectors	Power delivery and signal interconnection

1. Central Control Unit
   The NodeMCU ESP32 development board functioned as the main controller. It provided a dual-core processor, integrated Wi-Fi, and multiple GPIO pins for connecting to sensors and actuators. These characteristics, together with low cost and a mature software ecosystem, have made ESP32 a common choice in smart home, energy monitoring, and general IoT devices [18], [23], [30], [31]. In the prototype, the board was powered by a regulated DC supply. GPIO pins were configured as digital inputs for reading the ultrasonic sensor echo and as digital or PWM outputs for driving the servo motors and relay.
2. Sensing Subsystem
   The sensing subsystem employed an ultrasonic distance sensor to detect the presence of a vehicle in front of the gate and garage entrance. The sensor periodically emitted ultrasonic pulses and measured the time of flight of the reflected signal. The echo time was converted into a distance value using the speed of sound. The sensor was connected to the ESP32 via a trigger pin and an echo pin. The firmware generated the trigger pulse, measured the echo pulse width, and computed the corresponding distance. To improve robustness, several consecutive measurements were averaged, and values that were clearly inconsistent due to noise or transient reflections were discarded. Distance-based sensing, similar to this arrangement, has been used in earlier smart fence and door lock systems for access control [14]-[16].
3. Actuation Subsystem
   The actuation subsystem consisted of two servo motors and a relay module. One servo motor was mechanically coupled to the prototype gate mechanism, while the second servo drove the garage door mechanism. A PWM signal from an ESP32 GPIO pin controlled each servo. Specific duty cycle values corresponded to closed and open positions. The prototype required only these two discrete positions.
   Garage lighting was controlled by a single channel relay module. The relay coil was driven by a digital output of the ESP32, and the relay contacts were connected in series with the lamp and the mains supply. This arrangement allowed the ESP32 to switch the light on and off without direct exposure to high voltages. The relay module provided electrical isolation between the low voltage control circuit and the mains powered load.

4. Power Supply and Wiring
   All low voltage components (ESP32, ultrasonic sensor, servo motors, and relay coil) were powered by a DC power supply rated to deliver sufficient current for peak servo loads. A common ground reference was shared across all modules to ensure reliable signal levels. High current lines for the servos were routed separately from sensor and communication lines to reduce interference. All mains wiring for the lamp was physically separated and insulated from the low voltage circuitry.

The overall interconnection of the hardware modules is illustrated in Fig. 2. A NodeMCU ESP32 board serves as the central controller, with the HC-SR04 ultrasonic distance sensor connected to its digital pins for the trigger and echo signals. A 16×2 character LCD is interfaced to the ESP32 to provide local feedback on distance and system status, supported by a pair of indicator LEDs that signal operating conditions, including normal operation and warning or error states. Two SG90 servo motors are driven from separate PWM-capable pins to actuate the gate and garage door mechanisms. At the same time, a relay module connected to another digital output is used to switch the garage lamp. All devices share standard 5 V and ground rails from the DC power supply, and the wiring is arranged so that the ESP32 can coordinate sensing, actuation, and local indication in a single, compact prototype.

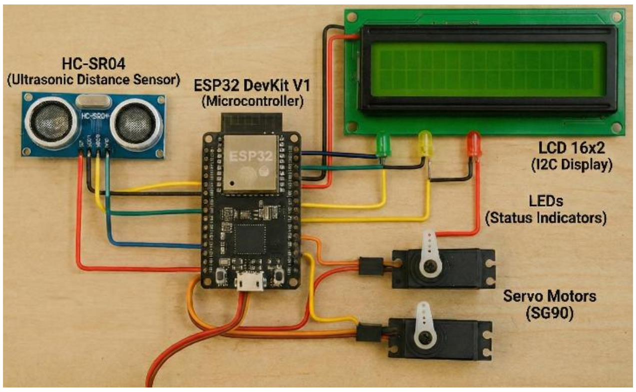

Fig. 2 Hardware wiring diagram of the ESP32 based gate, garage door, and lighting control prototype

D. Software Design

The software design comprised two layers: the ESP32 firmware and the Blynk based mobile user interface.

1. ESP32 Firmware
   The firmware was developed using the Arduino framework for ESP32. Its main tasks were to initialize hardware resources, maintain network and Blynk connections, acquire sensor data, implement the control logic for each operating mode, and update actuator states and dashboard information.
   The high-level control flow is illustrated in Fig. 3. After startup, the ESP32 configured all GPIO pins, initialized the serial interface for debugging, loaded configuration parameters such as distance thresholds and timing delays, and connected to the Wi-Fi network and Blynk cloud. Once initialization completed, the firmware entered a continuous loop with the following steps:
   a. Read the current operating mode and user commands from Blynk virtual pins.
   b. Acquire a distance measurement from the ultrasonic sensor and apply basic filtering.
   c. Update the internal state machine according to the selected mode (manual, automatic, or combined), sensor readings, and recent user commands.
   d. Drive the servo motors and relay to the required states (open or close gate, open or close garage door, switch light on or off).
   e. Send status updates, including sensor values and subsystem states, to the Blynk dashboard.
   f. Monitor the Wi-Fi and Blynk connection and attempt reconnection if a failure was detected. In the event of network loss, the system moved to a safe state by stopping motion and maintaining the last positions.

Timing functions in the Blynk and Arduino libraries were used to schedule periodic tasks such as sensor sampling and dashboard updates. Simple debouncing and timeout mechanisms were implemented to prevent repeated triggering and ensure that each open-close cycle was completed before a new command was processed.

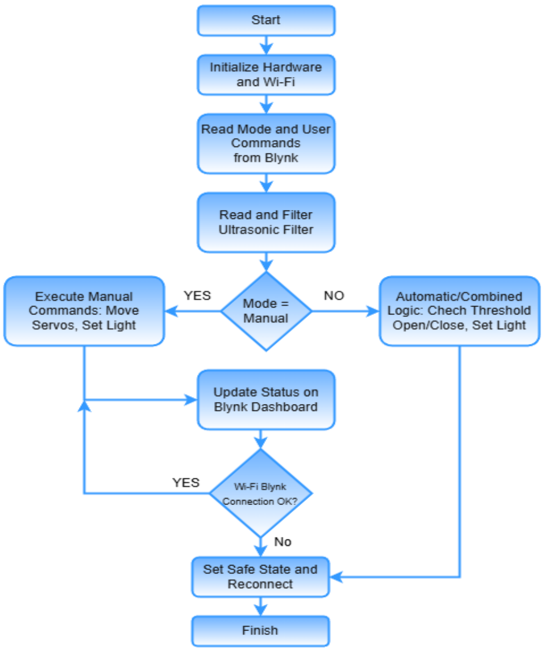

Fig. 3 High level control flow of the ESP32 firmware.

2. Blynk Mobile Interface
   The mobile interface was created in the Blynk application. The dashboard included push buttons to open and close the gate and garage door, a switch to select the operating mode (manual, automatic, or combined), a button or switch to toggle the garage light, and indicator widgets that displayed the current state of each subsystem and the measured distance. Dashboards of this type are commonly used in IoT-based home automation to provide smartphone control and feedback [21], [22], [27], [29].
   Each widget was linked to a Blynk virtual pin. When the user interacted with the dashboard, the corresponding virtual pin value was sent to the ESP32. The firmware reads these values, updates internal state variables, and sets actuator outputs. Conversely, the ESP32 periodically wrote to other virtual pins to reflect the latest sensor readings and subsystem states, providing real-time feedback to the user.

E. Operating Modes and Control Logic

The control logic was modeled as a finite state machine with three operating modes: manual, automatic, and combined.

1. Manual Mode
   In manual mode, all actions were initiated by the user through the mobile application. Pressing the gate or garage door button sent a command to the ESP32 to move the corresponding servo to the open or closed position. The lighting relay was controlled by a separate button that toggled the lamp state. Ultrasonic sensor data were still collected and sent to the dashboard, but they did not trigger automatic movement.
2. Automatic Mode
   In automatic mode, the system used the ultrasonic sensor to detect vehicle presence and execute a predefined access sequence. When the measured distance fell below a configurable threshold, the ESP32 interpreted this condition as a vehicle approaching the gate. The controller then:
   a. opened the gate by driving the gate servo to the open position;
   b. opened the garage door by driving the second servo;
   c. switched on the garage light via the relay.

   After a configurable delay that allowed the vehicle to enter the garage, the system initiated the closing sequence: the garage door and gate were closed, and the light was switched off. Minimum and maximum timing constraints were enforced between consecutive triggers to prevent rapid oscillation caused by transient changes in distance.
3. Combined Mode
   In combined mode, automatic triggers and manual commands were both active. The ultrasonic sensor could trigger the automatic open close sequence, but the user could override the system at any time via the mobile application. When a manual command was issued, it temporarily took priority over automatic logic. After the manual action was completed and a short priority interval had elapsed, automatic behavior resumed if the sensor readings still satisfied the trigger conditions. This mode provided flexibility by combining autonomous operation and user control.

F. Experimental Setup and Evaluation Metrics

The prototype was installed in a laboratory-scale setup that emulated a residential driveway and garage entrance. The ultrasonic sensor was mounted at a fixed height and angle facing the vehicle approach path. The gate and garage door mechanisms were implemented as scaled physical models driven by the servo motors. The garage lamp was a mains-powered light controlled by the relay module.

Three test scenarios were defined:

1. Scenario 1 (manual control): the system operated in manual mode. The gate, garage door, and light were controlled only through the Blynk mobile application.
2. Scenario 2 (automatic control): the system operated in automatic mode. Vehicle presence was emulated by moving a test object into and out of the sensor detection zone, and allowing the automatic open-close sequence to run without manual intervention.
3. Scenario 3 (combined control): the system operated in combined mode. Automatic triggers were active while the user occasionally issued manual override commands.

For each scenario, 20 complete open-close cycles were performed, similar to other experimental evaluations of IoT-based home automation systems [21], [25], [30]. During each cycle, the following metrics were measured:

a. actuation time: the time from command issuance or sensor trigger to completion of gate and garage door movement;
b. end-to-end communication latency: the time between pressing a button on the mobile application and the corresponding actuator response (for manual mode);
c. distance measurement accuracy: the difference between the distance reported by the ultrasonic sensor and a reference distance measured with a ruler or tape;
d. operational reliability: the ratio of successful cycles to the total number of attempted cycles, where a successful cycle was defined as one in which all planned actions (opening, waiting, closing, lighting control) were completed without communication failure or mechanical interruption.

The recorded data were used to compute average values and observed ranges. These results form the basis for the analysis of the integrated access control system's feasibility, responsiveness, and robustness in the next section.

III. RESULTS AND DISCUSSION

A. *Functional Performance*

The prototype was implemented and tested following the procedure described in the methodology. The evaluation covered three operating scenarios, each executed for 20 complete cycles: manual control via the Blynk mobile application (S1), fully automatic operation triggered by the ultrasonic sensor (S2), and combined operation with automatic triggering and manual overrides enabled simultaneously (S3). The test scenarios and conditions are summarized in Table 3.

Table 3. Test Scenarios and Conditions

Scenario ID	Description	Number of Cycles	Condition / Trigger Mechanism	Expected Output
S1	Manual control via the Blynk mobile app	20	User sends a command through Blynk	The gate, garage door, and lamp respond manually
S2	Automatic sensor-triggered operation	20	Ultrasonic sensor detects objects < 30 cm	Gate opens, garage door opens sequentially, lamp ON
S3	Combined operation of the gate, door, and lighting	20	Sensor detection and Blynk commands are used simultaneously	The gate, garage door, and lamp operate in integrated mode

In Scenario S1, the user issued commands through the Blynk dashboard to open and close the gate and garage door and to switch the garage lamp on and off. All commands were received by the ESP32 and executed correctly; the actuators moved to the intended positions, and the lamp followed the requested state. In Scenario S2, the gate opened automatically when the ultrasonic sensor detected an object within 30 cm, and closed again after 10 s without further detections. To avoid mechanical conflict, the garage door was actuated only after the gate reached its fully open position, so that neither mechanism moved at the same time. The garage lamp was switched on during access and could also be controlled from the mobile interface.

Scenario S3 verified the combined behavior. The ultrasonic sensor remained active, but the user could override or complement the automatic sequence from the mobile application. During the tests, manual commands issued while the automatic logic was active did not produce inconsistent states (for example, the gate and garage door were never commanded to move in opposite directions at the same time). After the manual action finished, the system resumed normal automatic operation.

Overall, the functional tests confirmed that the three subsystems—gate, garage door, and lighting—can be coordinated within a single ESP32-Blynk framework and controlled through a single mobile interface without observable functional conflicts or instability.

B. *Sensor Accuracy*

The HC-SR04 ultrasonic sensor was evaluated by comparing its readings with reference distances ranging from 10 to 50 cm, with a step size of 10 cm. For each position, several measurements were taken and averaged, and the absolute error was calculated with respect to the actual distance. The results are summarized in Table 4.

Table 4. Comparison of measured and actual distances

Actual distance (cm)	Measured distance (cm)	Error (cm)
10	10.8	0.8
20	21.0	1.0
30	30.9	0.9
40	41.2	1.2
50	51.8	1.8

Across the 10-50 cm range, the maximum absolute error is 1.8 cm at 50 cm, while the average error remains below approximately 1.2 cm. The error increases slightly with distance, consistent with the expected behavior of low-cost ultrasonic sensors: reflections at greater distances are more affected by beam divergence and environmental noise.

For the intended application, the critical requirement is the ability to distinguish between “vehicle present” and “no vehicle” at the gate and garage entrance. With the detection threshold set at 30 cm, the observed error is small compared to the distance difference between an empty driveway and a vehicle within the trigger zone. Consequently, the probability of false triggering or missed detection due to measurement noise is low, and the HC-SR04 sensor is suitable for vehicle detection in this context.

C. Response Time and Communication Latency

To characterize the system's dynamic behavior, two main metrics were measured: servo actuation time and communication latency between the mobile application and the ESP32 controller. The measurement methods are described in Section II-F, and the observed system performance metrics are summarized in Table 5.

Table 5. System performance metrics, including actuation time and latency

Component / Metric	Measurement description	Observed value/range
Gate servo actuation time	Time from command signal to fully open position	1.0 - 1.2 s
Garage door servo actuation time	Time from command signal to fully open position	1.5 s
Communication latency (ESP32-Blynk)	Time delay between app command and start of ESP32 response	300 - 480 ms
Operational reliability (20 cycles)	Successful cycles without system failure	100%

The gate servo required approximately 1.0-1.2 s to move from the closed to the fully open position. In comparison, the garage door servo required around 1.5 s, reflecting the higher mechanical load in the garage door mechanism. These times are typical for SG90-class servos used with small mechanical loads and are acceptable for residential-scale access control prototypes.

Communication latency between the Blynk mobile interface and the ESP32 was measured as the interval between a button press in the application and the moment the servo began to move. Over multiple test cycles, the latency values remained between 300 and 480 ms. The average latency was approximately 378 ms, as illustrated by the red dashed line in the latency graph in Fig. 4. Although small fluctuations occurred when Wi-Fi signal strength varied, all samples stayed within the specified range. From a user’s perspective, this sub-second delay is generally imperceptible during gate and garage door operations and does not reduce usability.

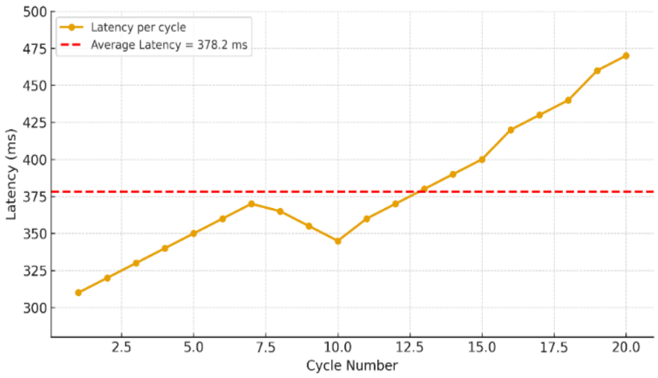

Fig. 4 Graph of system latency across test cycles.

D. *Scenario-Based Performance and Stability*

In addition to individual metrics, the system was evaluated per-scenario to quantify stability and reliability. Each of the three scenarios (S1-S3) was executed for 20 full cycles; a cycle was considered successful if all intended actions (opening and closing the gate and garage door, and correct lamp behavior) were completed without communication failures or firmware errors. The performance per scenario can be summarized as in Table 6, which combines the scenario definitions from Table 3 with the performance metrics of Table 5.

Table 6. Performance per scenario

Metric	S1	S2	S3
Description	Manual (Blynk mobile app)	Automatic (sensor-triggered)	Combined automatic and manual
Number of test cycles	20	20	20
Successful cycles	20	20	20
Success rate (%)	100	100	100
Average gate actuation time (s)	1.1	1.1	1.1
Average garage door time (s)	1.5	1.5	1.5
Communication latency (ms)	378	N/A	378

For Scenario S1, all 20 cycles completed successfully, demonstrating that the gate, garage door, and lamp can be controlled reliably through the Blynk dashboard. The average actuation times for the gate and garage door were within the specified ranges. The average communication latency was close to 378 ms.

In Scenario S2, the opening and closing sequences were triggered exclusively by the ultrasonic sensor and internal timing. The exact actuation times were observed, but communication latency did not play a critical role because movement was initiated locally by the ESP32, and the mobile application served mainly as a monitoring interface. Again, all 20 cycles were completed without failure.

Scenario S3 combined automatic operation with manual overrides. Here too, the system achieved a 100% success rate: manual commands issued during automatic operation were executed correctly, and the control logic then returned to normal automatic behavior. Latency values for manual commands in S3 remained within the 300-480 ms range.

Across all scenarios, no system crashes, firmware hangs, or loss of control were observed during the 20-cycle tests. Minor latency variations due to Wi-Fi signal fluctuations did not result in any incomplete cycles. The latency trend across cycles, shown in Fig. 4, confirms that the delay remained stable and within the expected bounds throughout the experiment.

These results indicate that, under the tested indoor conditions with a stable 20 Mbps Wi-Fi connection, the integrated system exhibits good short-term stability and reliable operation for the three defined scenarios.

E. Comparative and Critical Discussion

The numerical results obtained in this study do not attempt to surpass the best sensor accuracy or latency values reported in the literature. Still, they fall within the typical ranges for the same class of components. A maximum ultrasonic distance error of 1.8 cm and an average error below about 1.2 cm are consistent with previous investigations of HC-SR04 sensors in short-range detection applications. Likewise, the measured communication latency of 300-480 ms is comparable to values reported in other ESP32-Blynk-based smart home prototypes.

The main contribution of this work is therefore architectural and applicative rather than purely numerical. Whereas many earlier implementations focus on a single device or subsystem—such as a gate-only, a lighting circuit-only, or a single monitored door—the present system integrates three key residential access functions (gate, garage door, and garage lighting) on a single ESP32-Blynk platform, coordinated by ultrasonic detection. The experimental results show that this integration can be achieved without compromising response time, accuracy, or short-term reliability, and that the combined behavior can be evaluated quantitatively under realistic operating scenarios.

At the same time, the limitations of the prototype must be recognized. The mechanical structure uses small SG90 servo motors and scaled-down gate and door mechanisms; transferring the design to full-scale residential installations will require actuators with higher torque, more robust linkages, and additional safety mechanisms. The tests were conducted indoors with a stable network connection and over a limited number of cycles, so long-term behavior and performance under degraded Wi-Fi conditions remain to be studied. Security aspects, such as encrypted communication and strong user authentication, were beyond the scope of the present work but are essential for real deployments in smart homes.

Within these boundaries, the results demonstrate that an integrated gate, garage door, and lighting control system based on a NodeMCU ESP32 controller, HC-SR04 ultrasonic sensing, servo actuators, and a Blynk mobile interface can deliver acceptable accuracy, sub-second responsiveness, and high short-term reliability, providing a solid foundation for further development toward full-scale residential automation.

As summarized in Table 7, most existing smart home implementations either focus on monitoring or on general-purpose automation without explicitly targeting integrated access control for a gate, garage door, and lighting within a single experimental setup. Works such as [18] provide detailed measurements of end-to-end latency for ESP32-based monitoring nodes, [21] quantifies latency reductions achieved through edge-computing architectures, and [19] and [25] demonstrate flexible multi-device automation and NodeMCU-based frameworks, respectively. However, these systems typically do not report scenario-based reliability metrics or coordinated multi-actuator behavior for access control. In contrast, the present work integrates gate, garage door, and garage lighting control on a single ESP32-Blynk platform. It reports quantitative results on sensor accuracy, actuation time, communication latency, and success rate across three operating scenarios, thereby complementing the existing literature with a focused, experimentally validated access-control use case.

Table 7. Comparison with selected related smart home works

Work	Functions covered	Platform/network	Latency / key metric	Quantitative evaluation?
Babiuch and Postulka [18]	Smart home monitoring (environmental and energy-related data)	ESP32-based sensor nodes, Wi-Fi, IoT backend	Measures end-to-end and transmission latency for different configurations	Yes - latency and packet-related metrics
Stolojescu-Crisan	General-purpose smart home	qToggle platform, ESP8266/ESP8285 nodes,	Focus on scalability and flexibility;	Yes - number of devices,

et al. [19]	automation (lighting, climate, irrigation, security)	Raspberry Pi hub, Wi-Fi	no detailed user-to-actuator latency for a specific access-control case	resource usage, case-study behavior
Yar et al. [21]	Smart home automation with edge-computing support (data aggregation and control at the edge)	IoT devices at the edge, edge server, cloud backend, Wi-Fi/Internet	Evaluates processing delay and network latency reduction when offloading to edge nodes	Yes - simulation-based evaluation of latency and network load
Islam et al. [25]	Home automation framework (lighting and small appliances via mobile app)	NodeMCU (ESP8266), MQTT broker, Wi-Fi, Android app	Describes overall responsiveness, but without detailed per-scenario latency and reliability	Partial - functionality and basic timing discussed

IV. CONCLUSIONS

This study developed and evaluated an integrated IoT-based access control system that combines gate operation, garage door automation, and garage lighting control in a single platform. A NodeMCU ESP32 was used as the main controller, interfacing with an HC-SR04 ultrasonic sensor for vehicle detection, SG90 servo motors for mechanical actuation, and a relay module for lamp control. A Blynk-based mobile application provided a unified dashboard for monitoring and manual control over Wi-Fi. The system was implemented in a laboratory-scale prototype and tested under three operating scenarios: manual control, automatic sensor-triggered operation, and combined automatic and manual operation.

The prototype met its functional objectives in all three scenarios. The ultrasonic sensor exhibited a maximum distance error of 1.8 cm and an average error below about 1.2 cm for target distances between 10 cm and 50 cm, which is sufficient for reliable discrimination between the presence and absence of a vehicle in the trigger zone. The gate and garage door actuators required approximately 1.0-1.2 s and 1.5 s, respectively, to complete an opening movement. At the same time, the end-to-end communication latency between the Blynk application and the ESP32 ranged from 300 to 480 ms, with an average of around 378 ms. Across 20 test cycles for each scenario, the system achieved a 100 percent success rate, with no incomplete sequences or firmware failures observed. These results indicate that the proposed architecture provides adequate accuracy, sub-second responsiveness, and stable short-term operation for integrated residential access control at prototype scale.

Despite these positive results, several limitations should be noted. The mechanical structure is based on scaled-down gate and garage door models driven by small SG90 servo motors, so the reported actuation times and mechanical behavior cannot be transferred directly to full-size installations without redesigning the actuators and linkages and adding safety interlocks. All tests were carried out indoors under relatively stable Wi-Fi conditions and over a limited number of cycles, so long-term behavior and performance under fluctuating network quality and outdoor environmental factors were not assessed. Furthermore, the work focused on functional integration and basic performance metrics; communication security, encryption, and user authentication were not analyzed in detail, even though these aspects are critical in real smart home deployments.

Future work will address these limitations in several directions. On the hardware side, the system will be scaled up to full-size gate and garage door mechanisms, using higher-torque actuators, more robust mechanical structures, and appropriate safety features such as limit switches and obstruction detection. From the networking and system side, future versions will incorporate a local fallback mode that allows essential access control functions to continue when cloud connectivity is lost, and will harden the communication stack through encryption and token-based authentication. In addition, extended field trials in real residential environments with longer observation periods will be carried out to evaluate reliability under varying network and environmental conditions. Finally, further research will explore integration with additional smart home services, including security alarms, camera-based monitoring, and energy management, as well as support for multiple users and multi-node deployments, so that the proposed ESP32-based platform can evolve into a more comprehensive residential automation solution.

REFERENCES

[1] B. Mishra and A. Kertész, "The use of MQTT in M2M and IoT systems: A survey," IEEE Access, vol. 8, pp. 201071-201086, 2020, doi: 10.1109/ACCESS.2020.3035849.

[2] M. Mansour, A. Gamal, A. Ahmed, L. Said, A. Elbaz, N. Herencsar, and A. Soltan, "Internet of Things: A comprehensive overview on protocols, architectures, technologies, simulation tools, and future directions," Energies, 2023, doi: 10.3390/en16083465.

[3] M. Al-Kahtani, F. Khan, and W. Taekeun, "Application of Internet of Things and sensors in healthcare," Sensors, vol. 22, 2022, doi: 10.3390/s22155738.

[4] E. Senoo, E. Akansah, I. Mendonça, and M. Aritsugi, "Monitoring and control framework for IoT, implemented for smart agriculture," Sensors, vol. 23, 2023, doi: 10.3390/s23052714.

[5] A. Sharma, S. Sharma, and D. Gupta, "A review of sensors and their application in Internet of Things (IoT)," International Journal of Computer Applications, 2021, doi: 10.5120/ijca2021921148.

[6] M. Al-Gburi and L. Abdul-Rahaim, "Secure smart home automation and monitoring system using Internet of Things," Indonesian Journal of Electrical Engineering and Computer Science, 2022, doi: 10.11591/ijeecs.v28.i1.pp269-276.

[7] M. Poyyamozhi, B. Murugesan, N. Rajamanickam, M. Shorfuzzaman, and Y. Aboelmagd, "IoT—A promising

solution to energy management in smart buildings: A systematic review, applications, barriers, and future scope," Buildings, 2024, doi: 10.3390/buildings14113446.
[8] P. Chiradeja and S. Yoomak, "Development of public lighting system with smart lighting control systems and Internet of Things (IoT) technologies for smart city," Energy Reports, 2023, doi: 10.1016/j.egyr.2023.10.027.
[9] M. Mahbub, M. Hossain, and M. Gazi, "IoT-cognizant cloud-assisted energy efficient embedded system for indoor intelligent lighting, air quality monitoring, and ventilation," Internet of Things, vol. 11, p. 100266, 2020, doi: 10.1016/j.iot.2020.100266.
[10] T. Domínguez-Bolaño, V. Barral, C. Escudero, and J. García-Naya, "An IoT system for a smart campus: Challenges and solutions illustrated over several real-world use cases," Internet of Things, 2024, doi: 10.1016/j.iot.2024.101099.
[11] N. Swamy and S. Kota, "An empirical study on system level aspects of Internet of Things (IoT)," IEEE Access, vol. 8, pp. 188082-188134, 2020, doi: 10.1109/ACCESS.2020.3029847.
[12] A. Javed, A. Malhi, T. Kinnunen, and K. Främling, "Scalable IoT platform for heterogeneous devices in smart environments," IEEE Access, vol. 8, pp. 211973-211985, 2020, doi: 10.1109/ACCESS.2020.3039368.
[13] K. Shafique, B. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," IEEE Access, vol. 8, pp. 23022-23040, 2020, doi: 10.1109/ACCESS.2020.2970118.
[14] H. Purwanto, R. Nugraha, F. Ferdiansyah, D. Dewi, R. Sofian, and M. Rizaldy, "Sustainable smart home IoT to open and close the house fence using a scanning method," International Journal of Advanced Computer Science and Applications, 2023, doi: 10.14569/IJACSA.2023.0141072.
[15] Z. Zhu and Y. Cheng, "Application of attitude tracking algorithm for face recognition based on OpenCV in the intelligent door lock," Comput. Commun., vol. 154, pp. 390-397, 2020, doi: 10.1016/j.comcom.2020.02.003.
[16] M. Mehmood et al., "Invisible touch sensors-based smart and disposable door locking system for security applications," Heliyon, vol. 9, 2023, doi: 10.1016/j.heliyon.2023.e13586.
[17] F. Heiding, E. Süren, J. Olegård, and R. Lagerström, "Penetration testing of connected households," Comput. Secur., vol. 126, p. 103067, 2022, doi: 10.1016/j.cose.2022.103067.
[18] M. Babiuch and J. Postulka, "Smart home monitoring system using ESP32 microcontrollers," Internet of Things, 2020, doi: 10.5772/intechopen.94589.
[19] C. Stolojescu-Crisan, C. Crisan, and B. Butunoi, "An IoT-based smart home automation system," Sensors, vol. 21, 2021, doi: 10.3390/s21113784.
[20] S. Ameer, J. Benson, and R. Sandhu, "Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT," IEEE Transactions on Dependable and Secure Computing, vol. 20, pp. 4032-4051, 2023, doi: 10.1109/TDSC.2022.3216297.
[21] H. Yar, Z. Khan, M. Sajjad, and Z. Kastrati, "Towards smart home automation using IoT-enabled edge-computing paradigm," Sensors, vol. 21, 2021, doi: 10.3390/s21144932.
[22] M. Khan et al., "Smart Android-based home automation system using Internet of Things (IoT)," Sustainability, 2022, doi: 10.3390/su141710717.
[23] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and implementation of ESP32-based IoT devices," Sensors, vol. 23, 2023, doi: 10.3390/s23156739.
[24] N. N., M. Ahsan, M. Based, J. Haider, and E. Rodrigues, "Smart monitoring and controlling of appliances using LoRa based IoT system," Designs, 2021, doi: 10.3390/designs5010017.
[25] S. Islam, M. Hasan, M. Akter, and M. Uddin, "Implementation and analysis of an IoT-based home automation framework," Journal of Computer and Communications, 2021, doi: 10.4236/jcc.2021.93011.
[26] M. Murad, O. Bayat, and H. Marhoon, "Design and implementation of a smart home system with two levels of security based on IoT technology," Indonesian Journal of Electrical Engineering and Computer Science, 2021, doi: 10.11591/ijeecs.v21.i1.pp546-557.
[27] A. Avi, M. Rana, M. Bedar, and M. Talukder, "An Android application and speech recognition-based IoT-enabled deployment using NodeMCU for elderly individuals," Bulletin of Electrical Engineering and Informatics, 2023, doi: 10.11591/eei.v12i5.5062.
[28] W. Khalid, M. Jamil, A. Khan, and Q. Awais, "Open-source Internet of Things-based supervisory control and data acquisition system for photovoltaic monitoring and control using HTTP and TCP/IP protocols," Energies, 2024, doi: 10.3390/en17164083.
[29] P. Netinant, T. Utsanok, M. Rukhiran, and S. Klongdee, "Development and assessment of Internet of Things-driven smart home security and automation with voice commands," IoT, 2024, doi: 10.3390/iot5010005.
[30] C. Hassan et al., "Design and implementation of real-time kitchen monitoring and automation system based on Internet of Things," Energies, 2022, doi: 10.3390/en15186778.
[31] H. El-Khozondar et al., "A smart energy monitoring system using ESP32 microcontroller," e-Prime - Advances in Electrical Engineering, Electronics and Energy, 2024, doi: 10.1016/j.prime.2024.100666.