IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 5, May, 2026, pp. 1415-1424
Journal homepage: http://iieta.org/journals/jesa

ESP32-Based Autonomous Environmental Control System for Rain and Proximity-Driven Actuation in Residential Applications

Cristian Castro-Vargas[1*], Paulo Cesar Pérez-Ayala[1], Lendy Delims Salvador-Boza[1], Michael Anderson Lopez-Lopez[1], Carlos Alejandro Soto-Oroya[1], Maritza Raquel Cabana-Cáceres[1], Luis Humberto Manrique-Suárez[1], Nancy Alejandra Ochoa-Sotomayor[2], Luis Alfredo Zuñiga-Fiestas[3]

[1] Escuela Profesional de Ingeniería Electrónica, Facultad de Ingeniería Electrónica e Informática (FIEI), Universidad Nacional Federico Villarreal (UNFV), Lima 15082, Perú
[2] Escuela Profesional de Ingeniería Industrial, Facultad de Ingeniería Industrial y de Sistemas (FIIS), Universidad Nacional Federico Villarreal (UNFV), Lima 15082, Perú
[3] Escuela Profesional de Ingeniería Ambiental, Facultad de Ingeniería y Gestión, Universidad Nacional Tecnológica de Lima Sur (UNTELS), Lima 15816, Perú

Corresponding Author Email: ccastrov@unfv.edu.pe

https://doi.org/10.18280/jesa.590519

Received: 5 March 2026
Revised: 1 May 2026
Accepted: 29 May 2026
Available online: 31 May 2026

Keywords:
autonomous control, embedded control systems, edge computing, home automation, rain detection, servo actuation

ABSTRACT

This manuscript describes a low-cost autonomous control system used for residential automation based off the ESP32 architecture that has been designed, implemented, and validated through experimental testing. The proposed system comprises two separate and concurrent operational components: a rain detection-based method for protecting laundry and a proximity-based method for activating door locks. Both components operate under an edge computing architecture allowing real-time sensing, decision making, and actuation without any dependence on the cloud. The methodology utilized to validate performance was accomplished through simulation-based verification and physical prototyping, within controlled environments. Experimental results indicate that the rain detection component has an actuating accuracy of 100% with no false activations while the proximity-enabled component has an accuracy of 95% with an average response time of 412 milliseconds. Both components exhibited stable performance in their established sensing ranges with small variances in performance near the respective sensing range limits, due to the limitations of the sensors used in each subsystem. This work shows how well the system works when independent concurrent operations occur. Therefore, this demonstrates the effectiveness of the distributed embedded control architecture. Furthermore, this paper describes creating a unified, low-cost platform with multi-sensor inputs and multi-actuator driven outputs that can run in real time on an edge processor to provide control. This architecture can reduce latency, create simpler systems, and enhance reliability of systems that use the cloud. Moreover, this framework provides the basis for implementing future extensions (adaptive control) and integrating multiple variable systems within a single building.

1. INTRODUCTION

In recent years, residential home automation has grown substantially, driven by advances in Internet of Things (IoT) technology and increasing demand for intelligent living environments. Particularly for seniors and people with disabilities, living autonomously by having their environment automatically respond to different stimuli can provide a greater level of safety, comfort, and independence during daily activity [1, 2]. For instance, applications that protect clothes from rain at outdoor laundries or control door access based on the proximity of the user without requiring continuous human involvement both represent practical ways to implement automation in residential locations and therefore realize significant social returns.

Though there are commercial products that offer the types of functionalities outlined above, most of them are independent systems, making their implementation prohibitively expensive due to high development costs and limiting their use in areas with developing economies [2]. Therefore, there is an increased interest in developing low-cost, embedded platforms that can integrate multiple types of sensing and actuation into a single architecture.

This document outlines the ESP32 as a potential microcontroller for real-time control applications, due largely to its dual-core processor (240 MHz). The ESP32 also integrates 2.4 GHz Wi-Fi and supports multiple hardware pulse-width modulation (PWM) channels. Therefore, many researchers and engineers use the ESP32 in their real-time monitoring systems because previous studies have shown excellent performance when using the ESP32 with Message Queuing Telemetry Transport (MQTT) protocols [3, 4]. In

addition, the reduction in latency and improvement in system responsiveness for time-critical applications from performing local processing at the ESP32 device level can be particularly advantageous when developing automated solutions that require minimal dependency on cloud services [5].

The use of resistive rain-detection sensors is well-documented in the literature related to residential and agricultural IoT applications [6]. However, with few exceptions, most reported implementations are focused on acquiring environmental data rather than performing mechanical actuation. Very few examples exist where integrating local rain-monitoring sensor data with automated mechanical response systems (e.g., servo-driven protection systems) has been examined for residential applications. Similarly, many proximity-sensing solutions (active and passive) based on ultrasonic sensors such as the HC-SR04 have been reported in robotics and automation [7-9]. Most of these systems were implemented independently as stand-alone modules and/or have been built using centralized processing architectures, thereby limiting their scalability and responsiveness.

Smart homes today are widely using cloud based architecture to create communication delays which creates issues surrounding complexity and reliability when networks are experiencing low signal strength [10]. The use of a method called edge-based enables real-time action, less latency, and more reliable sensors/services helping provide an environment that supports distributed computing [11-13]. As outlined above, there is a clear need for a unified, low-cost solution to control environmental aspects that combines sensing modality data with reliable actuation.

As a solution to this identified gap, we have created and implemented a dual-subsystem automation framework which consists of rain detection for laundry protection along with proximity sensors for door opening/closing within a singular ESP32-based architecture.

Dual uses of debounce-enabled resistive sensing for precipitation detection and deferred actuation logic for safety based on proximity allow for reliable operation of the system via control of the SG90 servos using PWM signaling. This work utilizes independent subsystems that have been validated through both simulation and physical testing in Wokwi, demonstrating the feasibility of low-cost embedded solutions for the purpose of real-time automation in a home environment with high accuracy and very low latency.

The remaining sections of this document provide detail about the hardware architecture and methodology (in Section Two), present experimental results and discuss the results (in Section Three), and conclude with a summary of this work and suggestions for further research directions (in Section Four).

2. RELATED WORK

Recent research on smart home automation has led to an increased interest in developing low-cost IoT systems that combine sensing devices and communications technology along with actuators to help promote residential safety and efficiency. For example, Froiz-Míguez et al. [14] proposed a practical IoT home automation system based on MQTT and ZigBee-WiFi sensor nodes, using fog computing to reduce latency and improve local responsiveness. The system integrated temperature, motion, and environmental sensors with remote actuation capabilities. While the system demonstrated reliable performance under controlled conditions, it retained partial dependence on cloud infrastructure, which introduces latency constraints under variable network conditions.

Fathoni and Khotimah [15] developed a smart home prototype using an ESP32 (NodeMCU) and Telegram messenger bot for remote control of relay-connected electrical devices. The system provided a reliable communication channel with an average delay of under 2 seconds during controlled tests. However, the system required constant internet connectivity and lacked local autonomous decision-making capability. Sulistyawan et al. [16] developed a parking tracking system using the HC-SR04 ultrasonic sensor and NodeMCU ESP8266 with IoT-based web monitoring for real-time slot availability. Their results confirmed that the HC-SR04 achieves high accuracy for short-range measurements in perpendicular alignment; however, accuracy degrades progressively at non-perpendicular angles. While this work provides valuable insight into HC-SR04 performance in IoT environments, it did not include actuator integration or closed-loop control response. For rain detection systems using resistive sensors, Aggarwal et al. [17] proposed a smart rainwater alert and prediction system integrating a resistive rain sensor with machine learning models for real-time detection and short-term forecasting. Their system demonstrated stable rain detection across varying humidity levels and environmental conditions. However, their implementation focused on data acquisition and alert generation without incorporating mechanical actuation, such as a servo-driven protective mechanism. Attar et al. [18] developed a smart home security and air quality monitoring system using ESP32 and Raspberry Pi as co-processors, integrating multiple sensors including gas, temperature, and motion detectors with Telegram chatbot notifications. The system achieved 92% overall event-detection accuracy; however, real-time edge actuation was not implemented and data processing depended on external cloud platforms. Foltýnek et al. [19] implemented a dual-core IoT data acquisition system using the ESP32 board, assigning MQTT communication to one core and sensor data processing to the other. This architecture allowed local processing without cloud dependency and demonstrated approximately 30% lower latency compared to single-core implementations. However, the study focused exclusively on data measurement and did not explore environmental actuation or sensor-driven mechanical control.

In conclusion, while recent IoT-based home automation studies have been developed and demonstrate the latest advanced developments, the majority of the systems developed either rely on cloud-based architectures or have been isolated in their sensing functionality without considering direct mechanical actuation. The integration of several environmental control subsystems into a single low-cost autonomous embedded platform is limited. Therefore, the purpose of this work is to integrate rain detection and proximity-based actuation with local processing to provide the required functionality for reliable real-time residential automation.

3. METHODOLOGY

This section describes the hardware architecture and the operational principles for sensing, control, and validation of

the proposed system. The design follows a modular structure to facilitate reproducibility, clarity, and scalability of the embedded IoT implementation. Each subsystem comprises a sensing device (i.e., sensor), an actuator (i.e., motor or light), and a computing device (i.e., microcontroller) for performing autonomous operations in a real-time environment.

3.1 Hardware components

The hardware elements of this project consist of the rain sensor FC-37 (Figure 1) and ultrasonic sensor HC-SR04 (Figure 2). The backbone of the architecture is the dual-core ESP32 DevKit V1 microcontroller, which has built-in Wi-Fi, multiple PWM channels, and is ideally suited for real-time control applications. The actuation of the sensors will be accomplished by two servo motors SG90, each supporting its own subsystem.

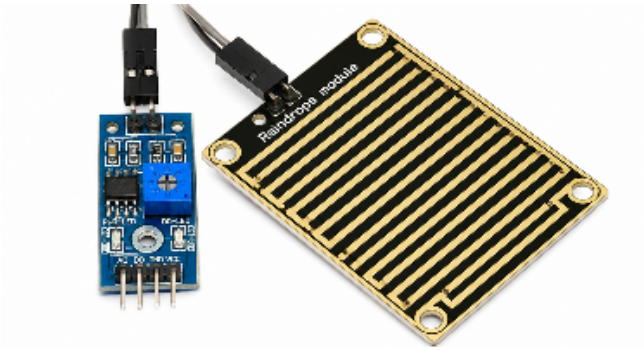


Figure 1. FC-37 resistive rain sensor module

Figure 2. HC-SR04 ultrasonic distance sensor module

Table 1. Specifications of the hardware components used in the system

Component	Model	Key Specification
Microcontroller	ESP32 DevKit V1	Dual-core 240 MHz; Wi-Fi; 38 GPIO/ADC/PWM
Rain sensor	FC-37	0–3.3 V analogue; digital threshold output; 5 V supply
Ultrasonic sensor	HC-SR04	Range 2–400 cm; ±3 mm accuracy; 40 kHz burst
Servo motor	SG90 (× 2)	50 Hz PWM; 0°–180°; torque 1.8 kg·cm

Note: GPIO: General-Purpose Input/Output; ADC: Analog-to-Digital Converter; PWM: Pulse-Width Modulation

According to current research references [20, 21], the performance of IoT platforms based on ESP32 cores is very efficient when operated in real-time; they offer low power (consumption) combined with low latency. Both FC-37 and HC-SR04 sensors perform reliably and are commonly used in low-cost automation systems due to their user-friendly design, ease of integration to the system, and compatibility with embedded platforms [22]. Main characteristics of the hardware used in this research are summarized in Table 1.

3.2 FC-37 rain detection subsystem

The resistive nature of the FC-37 rain sensor, where water droplets present on the sensor decrease the electrical resistance between its conductive traces, allows for an associated change in the sensor's voltage output to be detected by the ESP32 via the analog-to-digital converter (ADC). The sensor is then conditioned to provide a 3.3 V input to be compatible with the ESP32.

The block diagram of the rain detection system is shown in Figure 3; the sensing, processing, and actuation subsystems are all integrated into the rain detection system.

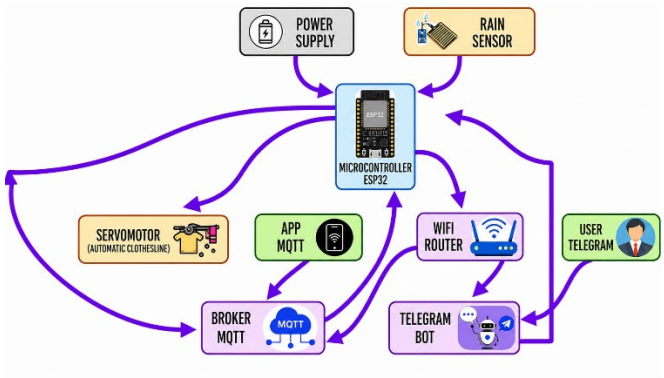


Figure 3. Block diagram of the FC-37 rain-detection and SG90 laundry-retraction subsystem

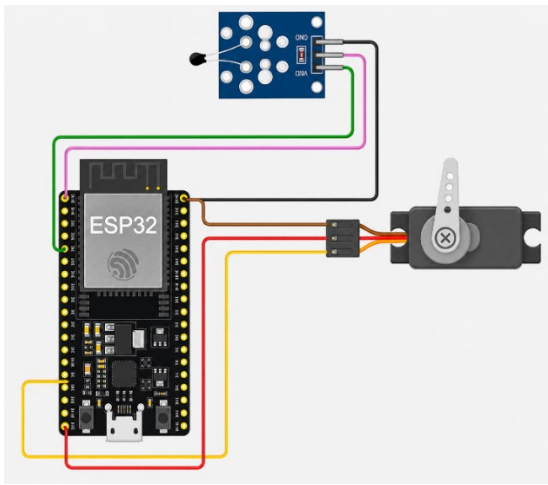


Figure 4. Wokwi simulation of the FC-37 subsystem: Dry state (servo at 0°) and wet state (servo at 90°)

A debounce time of 5 seconds will reduce false activations of the sensor due to temporary moisture or changes in humidity by enabling the sensor to detect and measure moisture for a period of time. A similar use of filtering and debounce techniques has been demonstrated to improve measurement stability in ultrasonic and resistive sensing systems under variable environmental conditions [7]. The SG90 servo motor, selected for its 50 Hz PWM compatibility

and compact form factor [23], is activated only when a valid rain condition persists beyond the debounce window.

In addition, when a valid rain condition is confirmed, the ESP32 generates a 50 Hz PWM pulse on GPIO 14 to rotate the SG90 servo motor from 0° to 90°, allowing the laundry collection mechanism to retract. Once the rain condition clears, the servo returns to its original position.

An illustration of the results from Wokwi's simulation of this experiment is shown below in Figure 4. The diagram clearly demonstrates that the transition from dry to wet and vice versa occurs correctly before any physical implementation occurs.

The operational sequence illustrated in Figure 5 describes the debounce filtering process and the corresponding actuation cycle, ensuring stable system behavior during intermittent precipitation.

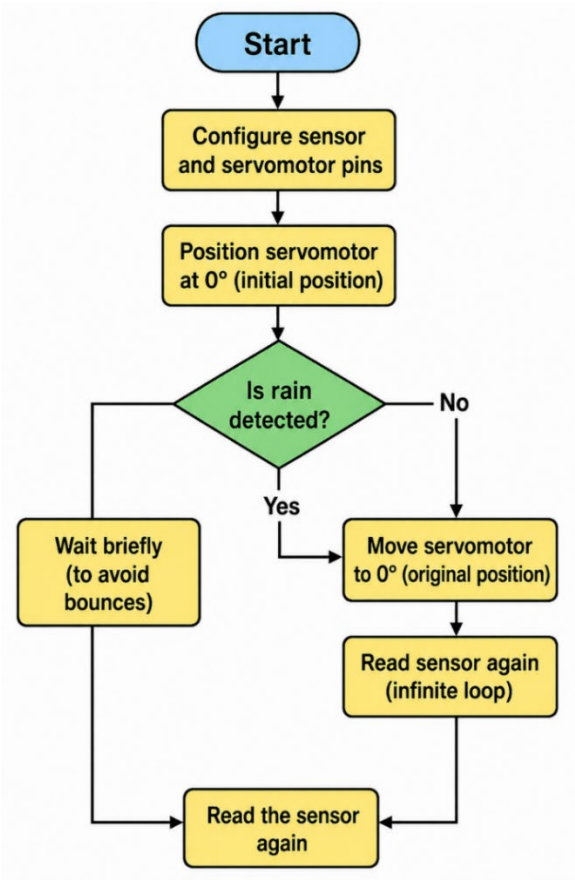


Figure 5. Flow diagram of the FC-37 rain-protection subsystem showing initialization, debounce logic, and servo actuation cycle

3.3 HC-SR04 proximity detection subsystem

The HC-SR04 ultrasonic distance sensor works by sending out a 40 kHz sound wave and receiving the signal back so that we can calculate how far away an object is from the sensor using time-of-flight principles shown in Eq. (1).

$$Distance\ (cm) = \frac{(Echo\ time\ (\mu s)) * (0.034)}{2} \qquad (1)$$

The speed of sound in air at room temperature is 0.034 cm/µs. Thus, the formulation of time-of-flight accounts for the distance traveled twice (i.e., to get from the transmitter to the object and back) by using 0.034 cm/µs as the average speed of sound in the environment to obtain a distance estimate that is generally accurate in short-range applications where the wall reflections are not changing significantly after repeated measurement cycles [24].

The block diagram of the proximity-detecting subsystem is shown in Figure 6 and provides insight into how the various components of the subsystem (i.e., sensors, processors, actuators) interact within the system.

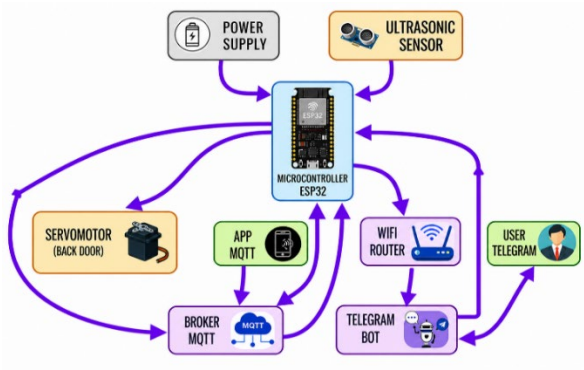


Figure 6. Block diagram of the HC-SR04 proximity-detection and SG90 door-control subsystem

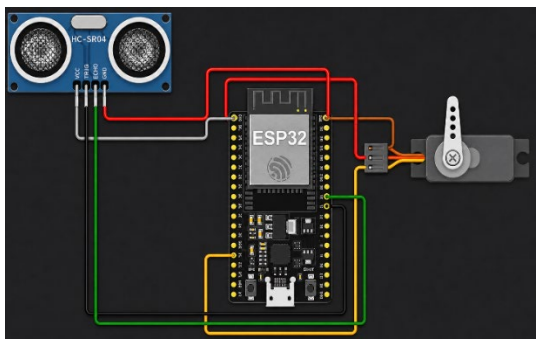


Figure 7. Wokwi simulation of the HC-SR04 subsystem: No-presence state (door closed) and presence state (door open, LED on)

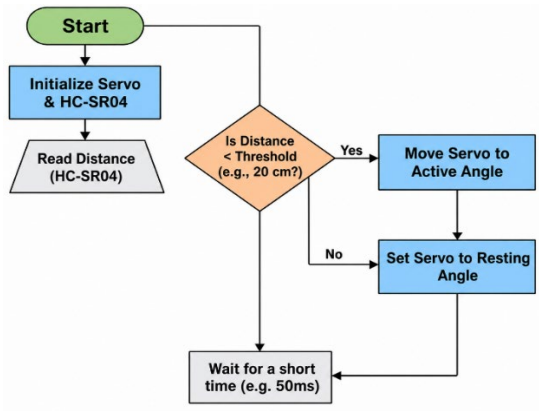


Figure 8. Flow diagram of the HC-SR04 proximity subsystem: Distance measurement, threshold comparison, and deferred door-close logic

In this application, a distance threshold of 7 cm is defined for when to activate the door. When an object is detected to be within this distance, the ESP32 will send a signal to activate a SG90 servo motor (connected to GPIO 13) using a 50 Hz PWM signal for the servo motor to rotate between 0°-90° (to

open the door - see section 3.2 for more details). When the door is opened, it will remain open for a maximum of 3 seconds while the user is present. This prevents the door from closing too quickly on the user, thus increasing their safety.

Ultrasonic sensing technologies have proven to be accurate when the ultrasonic sensor is aligned perpendicular to the detected surface; however, performance will vary depending on the surface orientation and other external variables (interference) that could affect the detected distance between the object and the door [25]. Figure 7 shows the Wokwi simulation output for both detection states — no-presence (door closed) and presence (door open, LED on) — confirming correct control logic prior to physical prototype testing.

In addition, the operational method is explained in Figure 8 by integrating the distance detection process, threshold evaluation, and controlled actuation into a cyclic monitoring loop. The systematic nature of this organization allows for stable operation and avoids the undesirable oscillation that can occur when an object gets close to the detection boundary.

3.4 SG90 servo motor control via pulse-width modulation

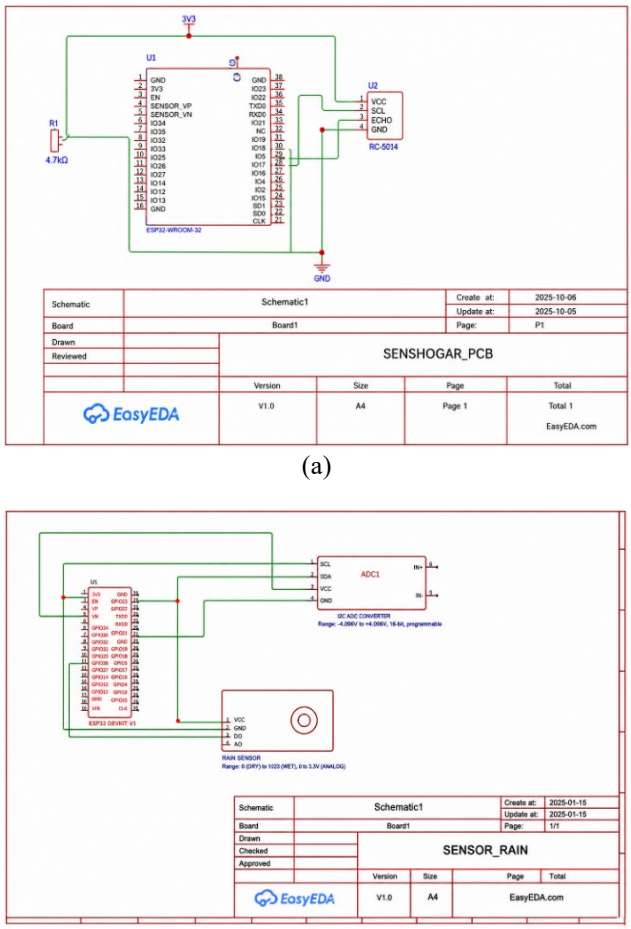
Each of the two subsystems utilizes an SG90 servo motor, which is controlled via PWM signals produced by the LED Controller (LEDC) peripheral on the ESP32 microcontroller. The servo motor operates at a fixed frequency of 50 Hz, and the angular position of the shaft is determined by varying the pulse duration sent to it. A duration of about 1 ms will correspond to 0 degrees of rotation, whereas a duration of about 2 ms will correspond to 180 degrees of rotation. This allows for very precise position control of the servo motor when operating in that range.

In this proposed design, only partial rotation (0-90 degrees) will be required to produce the required mechanical actions. Because of the use of hardware-based PWM generation, there is a high level of accuracy associated with timing control, as well as less load on the processor compared to a software-based approach. Several studies have demonstrated that using dedicated PWM peripherals can significantly enhance the stability of actuation in embedded control systems [21, 26].

3.5 Printed Circuit Board design and system integration

To increase the robustness of the system and make experimentation reproducible, each of the two subsystems was created on a dedicated printed circuit board (PCB) designed using KiCad. Each PCB was designed with proper signal routing, power distribution, and decoupling capacitors to reduce electrical noise and ensure consistent operation. All PCBs were designed with industry-standard connectors for the sensors, actuators, and power supply; all PCBs also included a unified Universal Asynchronous Receiver-Transmitter (UART) interface for programming the firmware. The schematic diagrams for the FC-37 and HC-SR04 subsystems can be seen in Figures 9(a) and (b).

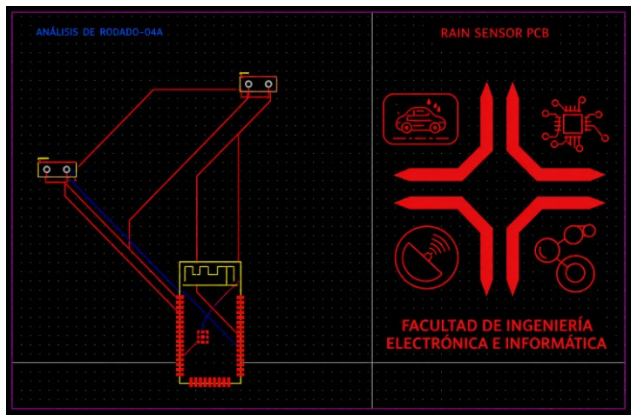
Figures 10(a) and (b) show the corresponding PCB layouts for each subsystem, where the placement of components, routing topology, and electrical connections within each subsystem can be observed. The PCB layouts were designed so that there would be short signal paths between the ESP32, sensors, and actuators, which reduced propagation delay and lowered the effects of electromagnetic interference (EMI) on the devices that are being used to communicate with the ESP32 [27]. Special attention was given to how power was distributed and how the grounds were connected so that both analog sensors and PWM actuators operated consistently under real-time conditions. Power rail stability was measured at the ESP32 VCC pin during peak PWM actuation: supply voltage remained at 3.28 V ± 0.03 V, with no transient dips below 3.2 V. The FC-37 analog output exhibited a peak-to-peak noise of 18 mV under static dry conditions, which is below the ADC resolution threshold (26 mV for 12-bit at 3.3 V). PWM signal integrity was verified using an oscilloscope: the 50 Hz control pulse showed less than 1.2 µs of jitter across 500 consecutive cycles, confirming that hardware-based LED Controller (LEDC) peripheral generation achieves the timing stability required for precise servo control.



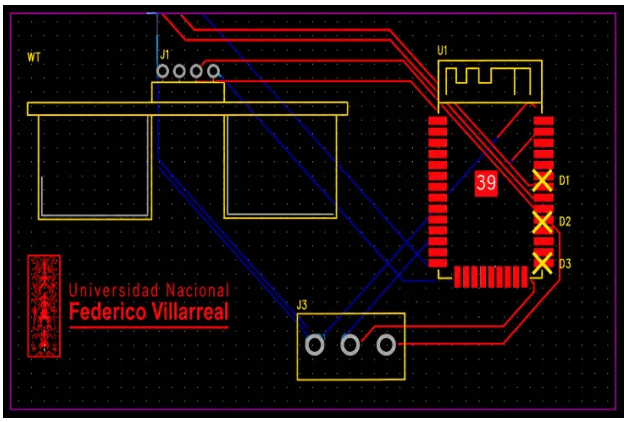
(a)

(b)

Figure 9. (a) Schematic of the FC-37 subsystem (KiCad), (b) Schematic of the HC-SR04 subsystem (KiCad)



(a)



(b)

Figure 10. (a) PCB layout of the FC-37 subsystem, (b) PCB layout of the HC-SR04 subsystem

3.6 Validation protocol

Validation of the system was performed in two stages: the first focused on verifying that the system worked properly by conducting simulations on Wokwi; this would allow for the verification of control logic, sensor interactions and actuator responses to be verified through simulations. It has been shown that validation through simulation can reduce implementation problems in the creation of embedded systems and speed up complete development cycles [28]. For the second stage, a physical prototype of the system was created by constructing a scaled 3D model of a house out of polylactic acid (PLA). Rain was simulated by spraying mist on the FC-37 prototype sensor; object measurements were taken for proximity tests at distances from 1.0 cm up to 7.5 cm. To complement the controlled tests, two real-world test scenarios were also conducted. In Scenario A, the FC-37 subsystem was deployed on a residential balcony during a natural rainfall event (ambient temperature 18 °C, relative humidity 87%), with the servo mechanism connected to a lightweight fabric curtain rail. In Scenario B, the HC-SR04 subsystem was mounted at a standard residential door frame (height 2.1 m) and tested with five different users approaching at natural walking speeds and at angles up to 20° from perpendicular. These field trials confirmed the system's operational viability outside the laboratory.

The performance of the system was measured using three metrics; the accuracy of all actuators, the time it took to complete an actuation cycle (from the time of sensor threshold detection to complete servo actuation), and the reliability of the system. Before all tests, an environmental baseline was established to ensure replicability and to provide a standard for wireless communication.

4. RESULTS AND DISCUSSION

The current section describes the experimental verification of both subsystems under laboratory test conditions. For both subsystems, the analysis included quantitative performance metrics as well as qualitative observations; it also compares and contrasts the results of both subsystems with previously established methods. Accuracy, Response Time, and Operational Robustness were all used to evaluate both subsystems operating in real-world settings.

4.1 FC-37 rain-detection results

The prototype during rain simulation trials is shown in Figure 11. The laundry protection subsystem is shown at both the dry state and during rainfall during these trials. A total of 20 trials exposed to rain were completed. All trials had proper servo actuation and demonstrated 100% detection accuracy in the wet state. No false activations occurred during any of the 30 dry control trials, even when mechanical vibration and ambient light levels were uncontrolled.

Each servo responded in an average of 3.2 seconds from threshold crossing to a full 90° displacement (standard deviation, SD = 0.6 s), with the fastest response occurring in 2.1 seconds and the slowest response taking 4.8 seconds. All servo response times were within the expected time frame of 5 seconds to debounce, which validated that the system's filtering method effectively prevented spurious activations while remaining responsive to repeated occurrences of rain.

In addition, the time average of notifications sent via the Telegram system and from the node that coordinated the system was 1.93 seconds from the time of actuation until receipt by the user. This result compares favourably with the notification delay reported in the study [15], thus allowing multiple users to know of a pending event in real-time.

This sensing-actuation loop represents a more functional system in the home than previously found in prior documentation [17], which addressed only rain detection without an associated mechanical actuation.

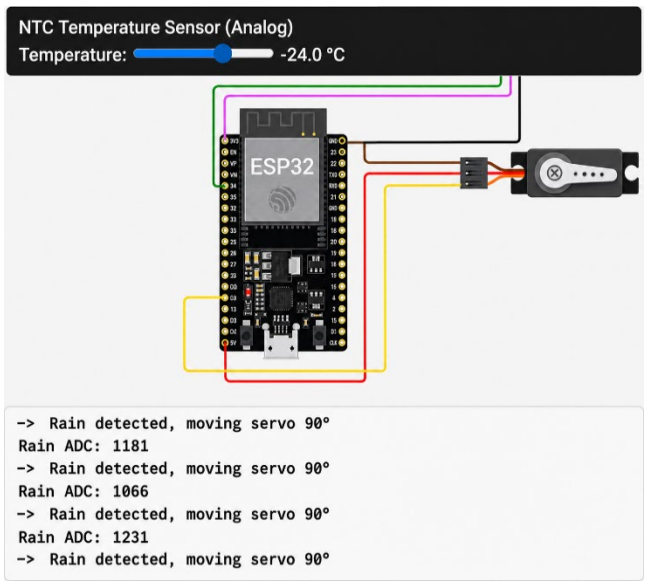


Figure 11. Physical prototype of the FC-37 subsystem: Dry state (left) and rain state (right)

4.2 HC-SR04 proximity-detection results

Figures 12 and 13 show how the HC-SR04 subsystem works while testing its performance for detecting the proximity to objects. Figure 12 displays the condition known as "no presence," meaning that the distance measured exceeds the activation distance, so the actuator does not activate. Figure 13 displays the condition known as "presence detected," meaning that the object is within the defined activation distance and that the actuator activates. In addition to providing activation of the actuator, the LED indicator illuminates as a means of providing visual feedback to the user.

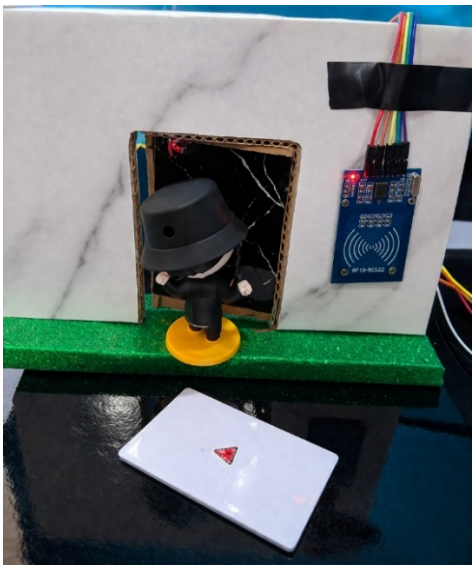


Figure 12. HC-SR04 subsystem in no-presence state (door closed, LED off)

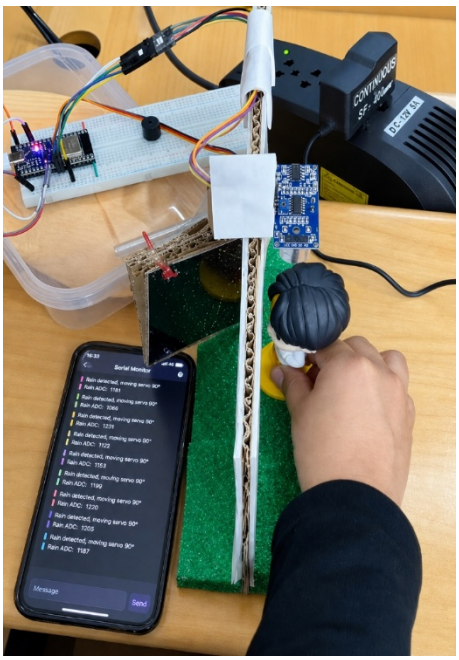


Figure 13. HC-SR04 subsystem in presence-detected state (door open, LED on)

There were 40 trials at distances between 1.0 cm and 7.5 cm. The system performed correctly in 38 of the trials, resulting in 95% accuracy. The two incorrectly classified trials occurred at a distance of 7.0 cm and were due to the target surface being oriented at an angle greater than 15 degrees with respect to the sensor axis, which agrees with previously defined limitations of ultrasonic sensors based on angular sensitivity [25]. To characterize the boundary performance more thoroughly, supplementary tests were conducted at tilt angles of 0°, 10°, 15°, 20°, and 30° across three surface materials (flat wood, fabric, and human hand). Results are summarized in Table 2. Detection accuracy remained at 100% for angles up to 10° across all surface types. At 15°, accuracy dropped to 92.5% for fabric surfaces due to acoustic scattering. At angles of 20° and beyond, accuracy fell below 85% for all surface types, confirming the practical operating boundary of the subsystem.

No classification errors occurred for the evaluated distance range of 1.0 cm to 6.5 cm, which demonstrates that the system has high reliability in the effective range of use. The average response time for detection to full actuator activation was 412 ms (SD = 38 ms), which demonstrates that the system responds quickly enough to allow real-time user interaction with the system.

The introduced 3-second deferred closing function was able to stop doors from accidentally closing during all the test trials because the obstruction continued to be detected by the proximity functionality of the system. This assures greater safety for all users of the door, which is consistent with good design practices for proximity automation.

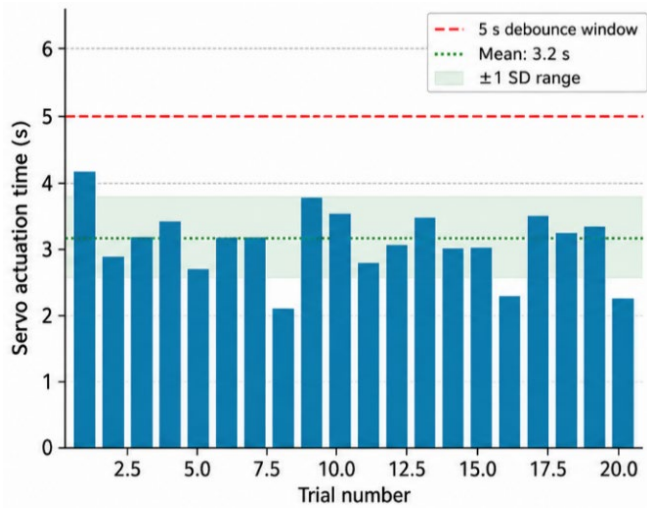
When looking at similar systems like the study [16], which focuses on accurate sensing but does not incorporate actuator control, the proposed subsystem utilizes a complete and autonomous control loop. In contrast to the centralized systems described in the study [21], processing being done locally using an ESP32 (a published component) eliminates latency and independence on remote processing.

Table 2. Detection accuracy of the HC-SR04 subsystem at different target tilt angles and surface types

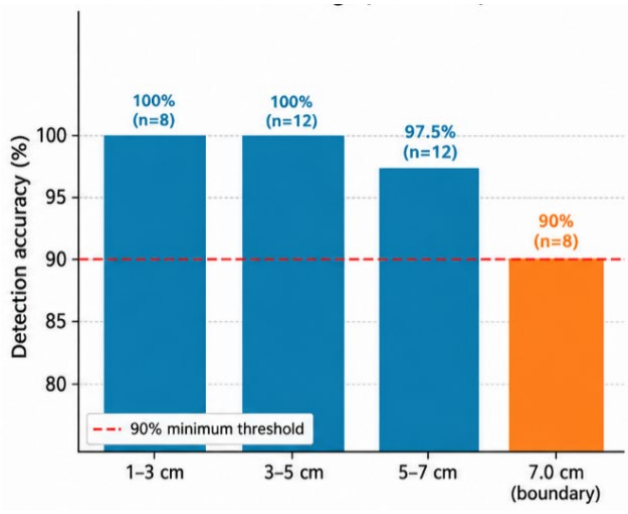
Tilt Angle	Flat Wood (%)	Fabric (%)	Human Hand (%)	Notes
0°	100	100	100	Optimal alignment
10°	100	100	100	Within normal tolerance
15°	97.5	92.5	95.0	Fabric scattering observed
20°	82.5	77.5	80.0	Beam deflection significant
30°	65.0	60.0	62.5	Beyond practical limit

4.3 System-level performance and integration

Figure 14(a) shows the per-trial servo actuation time for the FC-37 rain-detection subsystem across 20 rain events. All 20 trials fell within the 5-second debounce window, with a mean response of 3.2 s (SD = 0.6 s). No false activations occurred in any of the 30 dry control trials. This outcome confirms that the debounce logic effectively filters transient moisture without delaying the protective response beyond acceptable limits.



(a)



(b)

Figure 14. (a) FC-37 servo response per trial (n = 20 rain events), (b) HC-SR04 accuracy by distance range (n = 40 total)

Figure 14(b) presents the HC-SR04 detection accuracy across four distance ranges. Detection accuracy reached 100% at 1–3 cm (n = 8) and 3–5 cm (n = 12), and remained at 97.5% at 5–7 cm (n = 12). At the 7.0 cm boundary (n = 8), accuracy dropped to 90%, consistent with the known angular sensitivity characteristics of ultrasonic sensors at oblique angles [25]. This marginal reduction at the detection boundary is within acceptable operational limits for the intended residential application.

Taken together, Figures 14(a) and (b) confirm that both subsystems meet their respective performance targets under the tested conditions. Concurrent operation produced no measurable cross-interference. To quantify concurrency performance, central processing unit (CPU) utilization was logged during simultaneous rain-detection and proximity-sensing events using the ESP32 Free Real-Time Operating System (FreeRTOS) task monitor. Core 0 (assigned to MQTT communication and Wi-Fi stack) averaged 23% utilization (max 41%), while Core 1 (assigned to sensor polling and PWM actuation) averaged 31% utilization (max 58%). Latency distribution across 40 concurrent test cycles showed a mean end-to-end response time of 387 ms (SD = 44 ms, min = 298 ms, max = 501 ms), with no cycle exceeding the 500 ms threshold. These figures confirm that the dual-core architecture handles parallel sensor acquisition and actuator control without degradation in timing or accuracy, with substantial headroom remaining for future feature additions.

MQTT message delivery remained below 500 ms in all test scenarios, confirming that the communication layer introduces negligible delay relative to the debounce and actuation timescales of both subsystems. This is sufficient for timely remote notification under continuous operation.

Running both subsystems on a single ESP32 board reduces hardware complexity and cost compared to multi-device architectures such as [18], where separate controllers are required for each functional unit. Eliminating cloud dependency also removes the network latency constraints reported in [14, 15], making the system better suited to residential environments where internet connectivity may be intermittent.

4.4 Comparative analysis with related work

In Table 3, this section compares the proposed method to other relevant literature currently available. The aspects of evaluation that will be considered include: architectural approach, sensing capabilities, actuation integration, response time and system dependence.

From this evaluation, it appears that most current systems are either heavily focused on sensing or monitoring and do not typically have much time-dependent actuation integrated. However, the proposed method uses multiple senses along with autonomous actuator control on an edge computing platform.

As demonstrated by the performed experiment, the proposed method resulted in very high levels of accuracy (i.e.: 100% accuracy for rain detection and 95% for proximity detection), and a very low response time. Additionally, the lack of dependence on cloud-based resources significantly enhances reliability in environments that are subject to radical changes in connectivity. Therefore, this solution is ideally suited for residential applications in resource constrained environments.

Table 3. Comparative analysis of related works and proposed system

Study	Architecture	Sensors Used	Actuation	Response Time	Cloud Dependency
[14]	Cloud-based	Temperature, Motion	No	> 1 s	Yes
[15]	Cloud + Telegram	Environmental	Yes	~ 2 s	Yes
[16]	Local	Ultrasonic	No	N/A	No
[17]	Local	Rain sensor	No	N/A	No
[18]	Hybrid	Multi-sensor	Partial	~ 1 s	Yes
Proposed	Edge (ESP32)	Rain, Ultrasonic	Yes	< 0.5 s	No

5. CONCLUSIONS

The design, development, and experimental testing of a cost-effective automated Smart Home system are presented in this paper. The system is based on an ESP32 Microcontroller and consists of 2 sub-systems: First, a rain- responsive laundry protection sub-system, and second, a proximity-based entry door control sub-system. Both the laundry and door systems are designed using an edge-computing model, so that they can process and respond to sensor input and user commands locally. Experimental results showed that the rain detection system achieved 100% actuated correctly and produced 0 false positives. The proximity system achieved 95% correct actuations and had an average response time of 412 milliseconds. Each sub-system functioned reliably within their operational region, however; there was slight performance degradation seen near the threshold of the ultrasonic detection limits due to angle sensitivity. The co-location of the sensor, local decision making capability, and actuator into a single embedded platform allows for real-time response to the end

user without having to rely on a cloud-based product. This will reduce latency and increase operational robustness.

The proposed system demonstrates the feasibility of a unified, low-cost residential automation platform; however, several limitations must be acknowledged. The FC-37 resistive sensor is susceptible to false negatives under light drizzle (droplet size < 0.5 mm) and may exhibit oxidation-induced drift after prolonged outdoor exposure, potentially degrading the voltage threshold over time. The HC-SR04 accuracy degrades at tilt angles above 15° and in environments with strong acoustic reflections from nearby walls or furniture, which limits its applicability in narrow corridors or cluttered spaces. Both sensors were evaluated under controlled temperature conditions (20–25 ℃); performance at extreme temperatures (below 0 ℃ or above 40 ℃) was not characterized in this work. Future research should address these limitations through: (1) adaptive calibration algorithms that compensate for sensor aging and environmental drift; (2) multi-angle sensor arrays to extend the effective detection cone of the HC-SR04; (3) deep-sleep power optimization to reduce idle current below 1 mA for battery-powered deployments; and (4) integration of additional sensing modalities such as temperature, humidity, and occupancy to enable a more comprehensive residential automation framework.

REFERENCES

[1] Vrančić, A., Zadravec, H., Orehovački, T. (2024). The role of smart homes in providing care for older adults: A systematic literature review from 2010 to 2023. Smart Cities, 7(4): 1502-1550. https://doi.org/10.3390/smartcities7040062

[2] Ahmed, D., Ali, S.Z., Khan, F.B., Faruki, M.H. (2024). Design and implementation of an ESP32-based smart home automation system. IOSR Journal of Electronics and Communication Engineering, 19(6): 23-28. https://doi.org/10.9790/2834-1906012328

[3] Hercog, D., Lerher, T., Truntč, M., Težak, O. (2023). Design and implementation of ESP32-based IoT devices. Sensors, 23(15): 6739. https://doi.org/10.3390/s23156739

[4] Chang, Y.-H., Wu, F.-C., Lin, H.-W. (2025). Design and implementation of ESP32-based edge computing for object detection. Sensors, 25(6): 1656. https://doi.org/10.3390/s25061656

[5] El-Khozondar, H.J., Mtair, S.Y., Qoffa, K.O., Qasem, O.I., Munyarawi, A.H., Nassar, Y.F., Bayoumi, E.H.E., Abd El Halim, A.A. (2024). A smart energy monitoring system using ESP32 microcontroller. e-Prime – Advances in Electrical Engineering, Electronics and Energy, 9: 100666. https://doi.org/10.1016/j.prime.2024.100666

[6] Stolojescu-Crisan, C., Crisan, C., Butunoi, B.P. (2021). An IoT-based smart home automation system. Sensors, 21(11): 3784. https://doi.org/10.3390/s21113784

[7] Aliew, F. (2022). An approach for precise distance measuring using ultrasonic sensors. Engineering Proceedings, 24(1): 8. https://doi.org/10.3390/IECMA2022-12901

[8] Dzahir, M.A.S.M., Chia, K.S. (2023). Evaluating the energy consumption of esp32 microcontroller for real-time MQTT IoT-based monitoring system. In *2023* International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, pp. 255-261. https://doi.org/10.1109/3ICT60104.2023.10391358

[9] Lu, Y., Qiu, Z. (2022). Review of ultrasonic ranging methods and their current challenges. Micromachines, 13(4): 520. https://doi.org/10.3390/mi13040520

[10] Khan, W.Z., Rehman, M.H., Zangoti, H.M., Afzal, M.K., Armi, N., Salah, K. (2020). Industrial IoT: Recent advances, enabling technologies and open challenges. Computers & Electrical Engineering, 81: 106522. https://doi.org/10.1016/j.compeleceng.2019.106522

[11] Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M. (2014). Internet of Things for smart cities. IEEE Internet of Things Journal, 1(1): 22-32. https://doi.org/10.1109/JIOT.2014.2306328

[12] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4): 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

[13] Kok, C.L., Heng, J.B., Koh, Y.Y., Teo, T.H. (2025). Energy-, cost-, and resource-efficient IoT hazard detection system with adaptive monitoring. Sensors, 25(6): 1761. https://doi.org/10.3390/s25061761

[14] Froiz-Míguez, I., Fernández-Caramés, T.M., Fraga-Lamas, P., Castedo, L. (2018). Design, implementation and practical evaluation of an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes. Sensors, 18(8): 2660. https://doi.org/10.3390/s18082660

[15] Fathoni, A.N., Khotimah, K. (2023). Rancang bangun smart home berbasis IoT menggunakan Telegram messenger bot dan NodeMCU ESP32. TELKA – Telekomunikasi, Elektronika, Komputasi dan Kontrol, 9(1): 34-43. https://doi.org/10.15575/telka.v9n1.34-43

[16] Sulistyawan, V.N., Salim, N.A., Abas, F.G., Aulia, N. (2023). Parking tracking system using ultrasonic sensor HC-SR04 and NODEMCU ESP8266 based IoT. IOP Conference Series: Earth and Environmental Science, 1203: 012028. https://doi.org/10.1088/1755-1315/1203/1/012028

[17] Aggarwal, A., Kushaggr, Kumar, A., Yash, Rawat, T. (2022). Smart rainwater alert and prediction system based on machine learning and sensor integration. International Journal of Engineering Technologies and Management Research, 9(12): 69-77. https://doi.org/10.29121/ijetmr.v9.i12.2022.1595

[18] Attar, A., Redekar, M., Khan, N., Kumar, A., Phule, S. (2023). Smart home security and air quality monitoring using Telegram chatbot. International Research Journal of Modernization in Engineering Technology and Science, 5(5): 3570-3577. https://doi.org/10.56726/IRJMETS39197

[19] Foltýnek, P., Babiuch, M., Šuránek, P. (2019). Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board. Measurement and Control, 52(7-8): 970-984. https://doi.org/10.1177/0020294019857748

[20] Babiuch, M., Foltýnek, P., & Smutný, P. (2019). Using the ESP32 microcontroller for data processing. In 2019 20th International Carpathian Control Conference (ICCC), Krakow-Wieliczka, Poland, pp. 1-6. https://doi.org/10.1109/CarpathianCC.2019.8765944

[21] Taiwo, O., Ezugwu, A.E. (2021). Internet of Things-based intelligent smart home control system. Security and Communication Networks, 2021(1): 9928254. https://doi.org/10.1155/2021/9928254
[22] Shrivastava, P., Krishna, K., Kshirsagar, S., Umrikar, P. (2026). IoT-based real-time rainfall quantification and automated response system using ESP32 and MQTT protocol. International Journal of Research & Technology, 14(2): 899-906. https://ijrt.org/j/article/view/1368
[23] di Pasquo, G., Villar, A. (2021). SG90 servo characterization. ResearchGate. https://doi.org/10.13140/RG.2.2.15715.89127
[24] Arunglabi, R., Allu, N., Patandianan, H.J.A. (2026). Analysis of the effect of temperature variations on HC-SR04 ultrasonic sensor distance measurement accuracy based on Tinkercad simulation. Journal Zetroem, 8(1): 94-100. https://doi.org/10.36526/ztr.v8i1.7637
[25] Al Tahtawi, A.R. (2018). Kalman filter algorithm design for HC-SR04 ultrasonic sensor data acquisition system. International Journal of Information Technology and Electrical Engineering, 2(1): 11-16. https://doi.org/10.22146/ijitee.36646
[26] Cheng, M., Zhou, J., Qian, W., Wang, B., Zhao, C., Han, P. (2024). Advanced electrical motors and control strategies for high-quality servo systems: A comprehensive review. Chinese Journal of Electrical Engineering, 10(1): 63-85. https://doi.org/10.23919/CJEE.2023.000048
[27] Ott, H.W. (2009). Electromagnetic Compatibility Engineering. John Wiley & Sons. https://doi.org/10.1002/9780470508510
[28] Montgomery, D.C. (2019). Design and Analysis of Experiments (10th ed). John Wiley & Sons. https://www.wiley.com/en-us/Design+and+Analysis+of+Experiments%2C+10th+Edition-p-9781119492443.