ASM Sc. J., 21(2), 2026
https://doi.org/10.32802/asmscj.2006.1007

Low-Cost IoT-Based Smart Door Lock with Real-Time Telegram Integration

R. Omar[1], M.A. Mohammed Ahmed[1], J. Abdul Sukor[2*], M.Z. Hussin[1] and J. Jalani[1]

[1]*Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia Parit Raja, 86400 Batu Pahat, Johor, MALAYSIA*

[2]*Department of Electrical Engineering Technology, Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia Pagoh Campus, Johor, MALAYSIA*

This paper presents a cost-effective Internet of Things based smart door lock and alarm system designed to enhance residential security through remote access and automated threat detection. The system is built using the ESP32 CAM microcontroller, which supports image capture and wireless communication, and is integrated with a cloud-based messaging interface using Telegram. Multiple sensors including light, sound, and motion detectors enable adaptive responses to environmental changes and potential intrusions. The user can control the locking mechanism remotely and receive visual confirmation of visitors. Embedded safety features such as automatic re-locking and alert notifications improve system reliability. Experimental validation shows that the system responds to user commands within five seconds and completes a full operation cycle in under one minute. With an estimated total cost of less than RM150.00, this system is considerably more affordable compared to similar systems currently available in the market. The results confirm the practicality of developing low-cost and scalable smart security solutions using open-source components and widely available communication platforms for home applications.

Keywords: IoT; smart home security; ESP 32-CAM; intrusion detection

I. INTRODUCTION

The concept of the smart home, also referred to as a "connected home," has seen significant growth over the past decades due to advancements in communication technologies and embedded systems (Mujeye, 2024). In a smart home environment, household appliances and systems such as lighting, heating, ventilation, air conditioning (HVAC), security, and entertainment are connected and can be remotely controlled via a centralised system, often using smartphones or computers (Paraschiv *et al.*, 2021; Shin *et al.*, 2019). This technological shift aims to improve user comfort, energy efficiency, and security while offering greater convenience, especially to elderly and physically impaired individuals (Yan *et al.*, 2014).

Among the various applications of smart home systems, home security plays a critical role. Traditional door locking mechanisms rely heavily on physical keys and manual operation, which are prone to several issues including lost keys, lock tampering, unauthorised access, and the need for physical presence to grant entry (Al-Ali & Al-Rousan, 2004; Nakandhrakumar *et al.*, 2022). These limitations have led to the development of Internet of Things (IoT)-based smart door locks that allow users to monitor and control access to their homes remotely and securely (Ahmed *et al.*, 2024).

IoT-enabled smart locks incorporate sensors, actuators, and microcontrollers to provide real-time access control and monitoring capabilities. Users can lock and unlock doors using mobile applications, receive alerts about visitor activity, and integrate the system with video or image capturing devices such as Wi-Fi-enabled cameras (Kurniasyah &

*Corresponding author's e-mail: jumadi@uthm.edu.my

Rakhmawati, 2024; Goyal *et al.*, 2024). These systems help mitigate common risks associated with traditional locks while enhancing accessibility for users who are not physically present at home.

In particular, the ESP32 microcontroller has emerged as a popular platform for IoT-based home security projects due to its built-in Wi-Fi and Bluetooth capabilities, low power consumption, and affordability (Mutiarani & Ritonga, 2025; Singh *et al.*, 2024). It allows developers to integrate wireless communication with sensors such as Passive Infrared (PIR) sensors, sound sensors, light sensors, and solenoid locks for a fully functional smart locking system (Dey *et al.*, 2016). The system can also be programmed using the Arduino Integrated Development Environment (IDE), which simplifies the process of code development and hardware interfacing (Anwar & Kishore, 2016).

To improve user interaction and accessibility, messaging platforms such as Telegram are often employed. Telegram bots can send notifications, capture and deliver images from the ESP32 camera, and receive user commands to trigger lock/unlock actions (Patil *et al.*, 2024). This allows homeowners to control the locking mechanism even when connected to different Wi-Fi networks, improving flexibility and control (Sharma *et al.*, 2022).

Additionally, modern smart lock systems incorporate alarm features that respond to motion or noise events near the entrance. For instance, PIR sensors can be configured to activate only during nighttime, based on readings from a light sensor, while a sound sensor can detect abnormal noises like glass breaking. When a threat is detected, a buzzer can be triggered as an audible warning to the homeowner (Rahman *et al.*, 2015; Mahalingam *et al.*, 2024).

Smart door locks also contribute to energy efficiency by reducing the need for constant physical checks and improving home automation integration (Balaji *et al.*, 2023; Sarbishaei *et al.*, 2024). Furthermore, with the widespread adoption of 5G and cloud computing, IoT-based smart security systems are expected to offer even faster and more reliable performance in the future (Skouby & Lynggaard, 2014; Dzogovic *et al.*, 2019).

Despite their benefits, smart door locks must address several challenges, including internet dependency, data privacy, and cyber-security risks. Nonetheless, continued research and innovation in this domain are helping to overcome these concerns using encryption protocols, secure authentication, and advanced communication frameworks (Al-Fuqaha *et al.*, 2015; Simatupang & Tambunan, 2022).

This project focuses on the design and implementation of a smart door lock system using the ESP32 microcontroller, a 12VDC solenoid lock, and multiple environmental sensors. The total development cost of this system is below RM150, making it significantly more affordable compared to similar products available in the market, such as the Smart Door Lock Face Recognition priced at RM550 and the Smart WiFi Door Lock priced at RM2000. The proposed solution aims to provide a cost-effective, accessible, and secure alternative to traditional door locks, contributing to the growing field of smart home automation (Sahat *et al.*, 2018; Zia *et al.*, 2025; Seera & Behera, 2025; Anik *et al.*, 2025).

II. METHODOLOGY

The methodology for developing the IoT-Based Smart Door Lock System consists of five stages: architectural design, circuit development, hardware integration, software implementation, and system testing. Each step is carefully planned and executed to ensure the system achieves security, usability, and real-time control objectives.

A. *System Development*

The development of the IoT-based smart door lock system follows a structured methodology encompassing conceptual design, hardware assembly, software integration, and system validation. As shown in Figure 1, the process begins with system planning, where specifications were outlined based on objectives such as remote-control capability, real-time monitoring, and secure access via mobile devices.

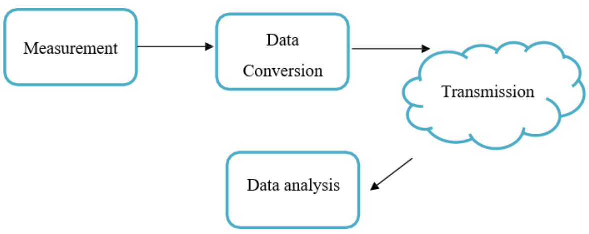


Figure 1. Flow of system development methodology

After defining requirements, a component selection phase was conducted. The ESP32-CAM microcontroller was chosen for its integrated camera and Wi-Fi features, which support both video streaming and control signals. A 12V solenoid lock was selected for its mechanical robustness, and a relay module was integrated for controlling power to the lock. The hardware configuration was then connected to a voltage regulation system using an LM7805 to deliver appropriate power levels. The software design proceeded in parallel, where the firmware was developed using Arduino IDE, and the mobile user interface was built with Blynk IoT platform. Cloud integration and device authentication were handled using Blynk's secure APIs and unique token generation.

Prototyping was done on a mock door assembly using a wooden panel, simulating a real-world application scenario. This was followed by system integration and testing, where the entire smart lock system was evaluated for reliability, latency, and usability. Finally, data was collected to assess performance under different Wi-Fi conditions and system response times. This methodology ensures that each component and function is tested both independently and within the integrated system, providing a modular and scalable approach to smart home security development.

B. System Architecture

The system architecture outlines the interaction among hardware modules, network components, and user interfaces in the smart door lock system. As illustrated in Figure 2, the core component is the ESP32-CAM, which connects both the relay module and 12V solenoid lock and communicates wirelessly with the Blynk IoT platform via Wi-Fi.

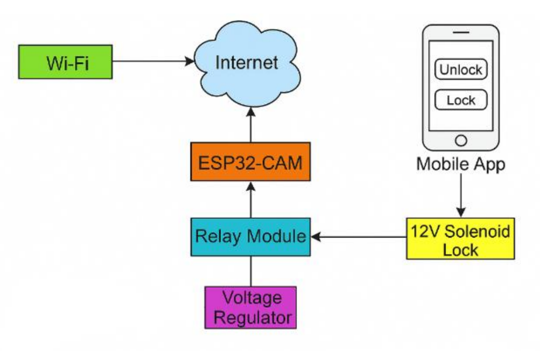


Figure 2. System architecture of the IoT-based smart door lock system

The ESP32-CAM acts as both the controller and video capture device. When a user accesses the mobile app, they can view a live feed from the camera and activate the lock or unlock function through button widgets. These commands are routed via Blynk's cloud server to the ESP32-CAM, which then toggles a relay connected to the 12V solenoid lock.

A voltage regulator circuit ensures that the ESP32-CAM receives a stable 5V while the solenoid operates on 12V. The architecture also includes status feedback, which allows users to monitor the lock state (locked or unlocked) directly on the app. From a networking standpoint, the system connects to a local Wi-Fi access point with internet access, enabling global control regardless of the user's physical location. The ESP32 continuously listens to input signals from Blynk, and the embedded logic ensures that only authenticated users can control the system.

The architecture is modular, allowing future integration with biometric scanners or keypad input. The system also supports over-the-air firmware updates (OTA), further reinforcing its scalability. This layered architecture provides the foundation for reliable, secure, and user-friendly smart door access control in residential environments. Together, these mechanical designs and the 3D printing workflow resulted in a low-cost, lightweight, and anatomically accurate robotic finger system. The integration of modular components and additive manufacturing not only ensured ease of prototyping and assembly but also allowed for future scalability and patient-specific customisation.

C. Hardware Development

The development of the smart door lock system required careful integration of several hardware components to ensure robust functionality and user-friendly operation. The core of the system was built around the ESP32-CAM microcontroller, selected for its integrated Wi-Fi and camera capabilities. As shown in Figure 3, the final assembly included the ESP32-CAM module, a 12V DC solenoid lock, a relay module, a voltage regulator (LM7805), and a compact PCB for stable connections.

Power was supplied through a dual-voltage configuration: 12V DC for the solenoid and 5V DC regulated for the ESP32-CAM and its peripherals. The solenoid lock was mounted to a wooden panel simulating a real door frame. The ESP32-CAM was mounted in a 3D-printed case and positioned to provide a clear visual of the entryway. The relay module acted as a digital switch, activated via GPIO from the ESP32 to energise the solenoid lock.

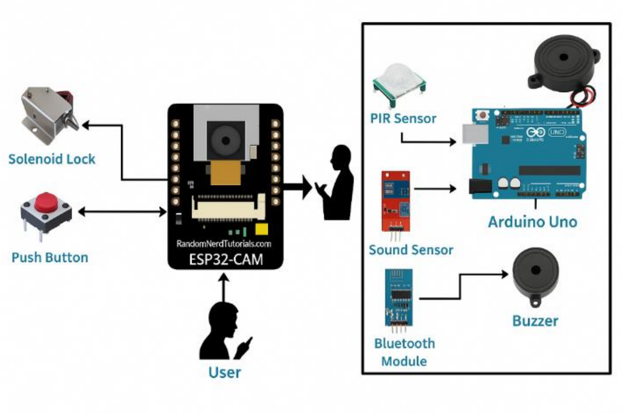


Figure 3. Complete hardware assembly of the smart door lock system

A custom PCB board was used to minimize wire clutter and soldering errors. This board included a rectifier diode to protect against back current from the solenoid and capacitors to stabilise power fluctuations. Heat sinks were attached to the LM7805 voltage regulator to manage thermal load during extended operation.

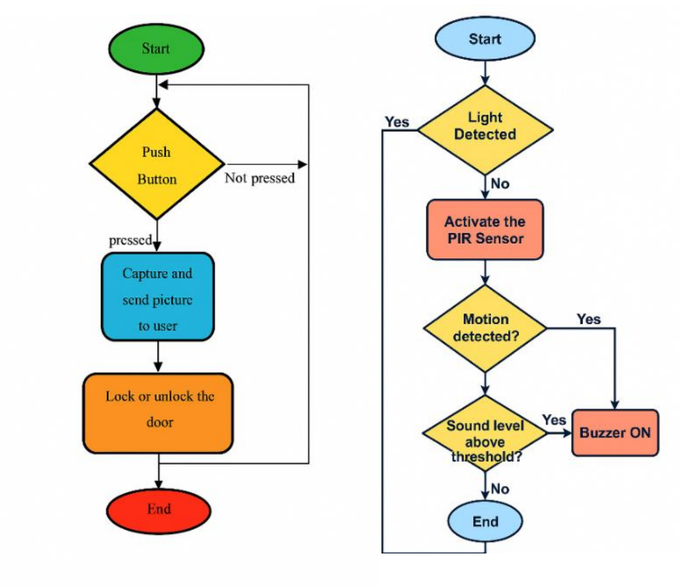
Component placement and wire routing were optimised to improve airflow and reduce electromagnetic interference. The complete system was tested in a mock-up door setting to verify reliable locking and unlocking mechanisms. These tests confirmed the effectiveness of the integrated hardware before moving on to full IoT software integration.

D. Software Development and IoT Integration

The smart door lock system operates on firmware developed using the Arduino IDE, programmed into the ESP32-CAM microcontroller. This firmware coordinates real-time input handling, network communication, camera control, and relay triggering, while integrating seamlessly with the Blynk IoT platform for user interaction. The software design is divided into two main operational branches: door lock control and intrusion alert detection, each illustrated in Figure 4(a) and Figure 4(b), respectively.

Figure 4(a) outlines the door lock control logic initiated when a push button is pressed at the door unit. Once triggered, the ESP32-CAM captures an image of the visitor and sends it to the user's mobile device through the Blynk application. The user can then decide to lock or unlock the door remotely via the app interface. If the button is not pressed, the system remains in standby mode, looping back to await input. This ensures efficient power usage and readiness for user action.

Figure 4(b) presents the alert system logic designed to detect suspicious activity in low-light conditions. The process begins with ambient light detection. If light levels fall below a certain threshold (e.g., nighttime), the PIR (Passive Infrared) sensor is activated to monitor motion. If motion is detected, or if the sound sensor picks up a noise above a preset threshold, the buzzer is triggered to alert nearby individuals. These decisions are managed locally by the ESP32 and can also be monitored remotely via the app.



(a) Door lock system flowchart (b) Alarm system flowchart

Figure 4. Software flowcharts of the smart door lock system

Both systems are integrated into the same firmware architecture, allowing the ESP32 to multitask without conflict. The code includes debounce logic, timed resets, and OTA (Over-the-Air) update capabilities for easy future upgrades. The mobile interface offers video streaming, button controls, and system feedback, making the experience intuitive and responsive. This dual-layer software model ensures the system provides both secure access control and responsive environmental awareness, making it highly effective for smart home security applications.

To prevent accidental activation due to network jitter, a debounce logic and timeout routine were implemented. A secondary safety feature was also embedded to re-lock the system automatically after a predefined delay, minimising user oversight. The Blynk mobile app was designed with a simple GUI consisting of button controls, video streaming widgets, and system status indicators. When the door camera is triggered, a real-time video stream is displayed on the app, allowing users to view the visitor before deciding to unlock. The system also includes OTA (Over-the-Air) firmware update capabilities, enabling remote code improvements without disassembly.

The program is also designed to allow future integration with voice assistants or additional biometric modules, making the system extensible. The seamless communication between hardware and cloud server was validated in multiple trials, ensuring low latency and consistent response. This software layer transforms the standalone hardware system into a connected smart home device, demonstrating the practical application of IoT in enhancing daily security and accessibility.

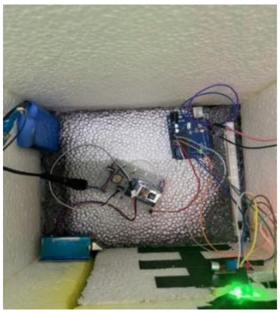
III. RESULTS AND ANALYSIS

This section presents the experimental results obtained during the implementation and testing of the IoT-based smart door lock and alarm system. The findings are discussed in relation to the project objectives, emphasising both the user-controlled lock mechanism and the automated environmental alert system. The project successfully demonstrates how a low-cost, Internet-enabled platform can deliver enhanced security, accessibility, and real-time feedback using ESP32-CAM and Telegram cloud services.

Figure 5 illustrates the physical distribution of the system components. External-facing components include the light sensor and motion detector, while the internal setup contains the ESP32 controller, buzzer, solenoid lock, and power management modules.

The system is fully interactive via Telegram Bot, as shown in Figure 6, where the user interface provides command options, system feedback, and photo evidence captured by the door camera. Upon pressing "Start" in the bot, a menu is sent, enabling the user to control the door remotely.

Figure 7 illustrates the process that takes place when a visitor presses the push button. The ESP32-CAM captures an image and instantly sends it to the authorised user via Telegram. The system detects the input, activates the camera, and delivers the picture with a time delay of less than 5 seconds, confirming its rapid response capability.



(a) Internal view

(b) External view

Figure 5. Smart Door Lock System

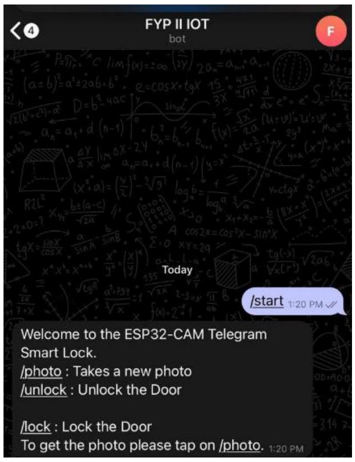


Figure 6. Telegram Bot User Interface

A. Lock System Test

As seen in Figure 8, the user is then provided with the option to unlock the door or request another photo for better identification. Once the door is unlocked, a confirmation message is sent. The same process is repeated when relocking the door.

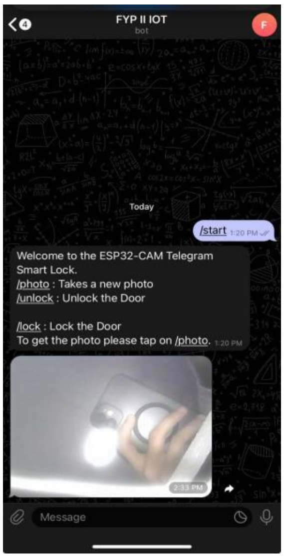


Fig. 7 Response to Push Button Activation

Figure 8. Lock/Unlock Control Options from User Interface

The system was tested under various Wi-Fi conditions. Notably, when the ESP32 was connected to the house Wi-Fi and the smartphone operated on mobile data, all functions still performed flawlessly. This validates that the system supports remote access without network constraints, provided an active Internet connection is present. The performance evaluation of the system revealed that the time between issuing a command and receiving a response was consistently within five seconds, demonstrating fast communication between the user and the device. The complete cycle for locking or unlocking the door, including photo capture and notification, was successfully completed in under one minute. Throughout all testing scenarios, the Telegram Bot maintained reliable synchronisation with the ESP32, regardless of the user's network or location.

B. Alarm System Test

The alarm system consists of an Arduino Uno controlling a light sensor (LDR), PIR sensor, sound sensor, and buzzer. As shown in Figure 9, the light sensor uses two LEDs: one to indicate power and another to reflect ambient light status. If the environment is dark (no light detected), the system activates the PIR sensor.

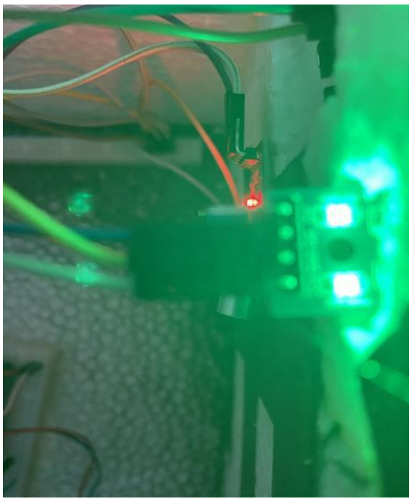


Figure 9. Light Sensor with Status Indicators

Upon detecting motion at night, the PIR sensor signals the Arduino Uno to trigger the buzzer. The buzzer's operational status is visually represented by a red LED, as seen in Figure 10.

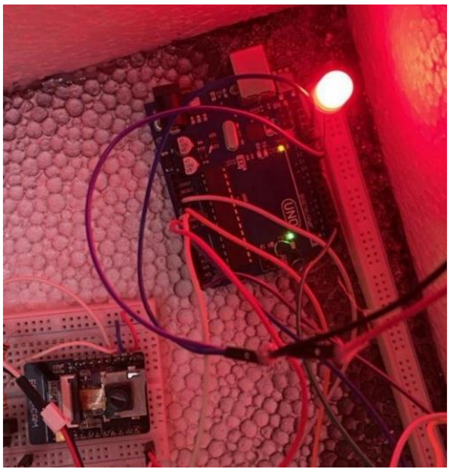


Figure 10. Motion-Based Buzzer Activation Using PIR Sensor

The sound sensor functions continuously and detects high-decibel events such as forced entry. As shown in Figure 11, a red LED indicates when a loud sound triggers the system. Thresholds were fine-tuned during testing to ignore background noise and respond only to significant disturbances.

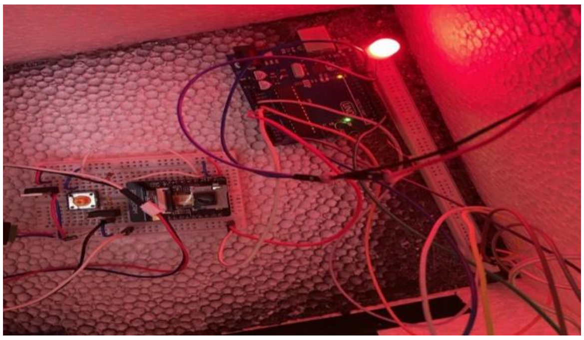


Figure 11. Sound Sensor Status with Visual Feedback

C. *Coding Implementation*

The software controlling the smart door lock and alert system was developed using the Arduino IDE. The code initialises essential hardware components, including the Telegram Bot, and defines relevant GPIO pin configurations. For the lock mechanism, pin 13 is assigned to the push button input, pin 12 controls the relay (connected to the solenoid lock), and pin 4 is used to activate the ESP32-CAM's onboard flash LED during photo capture. To maintain system security, the Telegram Bot is programmed to accept commands only from a predefined authorised user ID, preventing control by unauthorised users. Since Telegram supports multiple device access for the same account, users can control the door lock seamlessly from smartphones, tablets, or desktop clients. A simplified overview of the ESP32's initialisation and Telegram control code is presented in Figure 12, showing how the bot is configured and how hardware pins are declared and managed.

The system logic is event-driven, with Figure 13 illustrating the main loop that handles incoming Telegram commands such as "lock," "unlock," or "capture photo." Upon receiving a command, the ESP32-CAM executes the corresponding action and sends a confirmation or image back to the user through the Telegram bot. This ensures real-time, user-responsive control with minimal delay.

```
// Initialize Telegram BOT
String chatId = "968698922";
String BOTtoken = "7230704964:AAGEOaB81XGR5r-ERtCK2oMv-5_XuoXGNtU";

bool sendPhoto = false;

WiFiClientSecure clientTCP;

UniversalTelegramBot bot(BOTtoken, clientTCP);

// Define GPIOs
#define BUTTON 13
#define LOCK 12
#define FLASH_LED 4
```

Figure 12. Example: Initialisation Code for Telegram-Controlled Door Lock System

```
    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String fromName = bot.messages[i].from_name;
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("New photo request");
    }
    if (text == "/lock"){
      String r_msg = lockDoor();
      bot.sendMessage(chatId, r_msg, "");
    }
    if (text == "/unlock"){
      String r_msg = unlockDoor();
      bot.sendMessage(chatId, r_msg, "");
    }
    if (text == "/start"){
      String welcome = "Welcome to the ESP32-CAM Telegram Smart Lock.\n";
      welcome += "/photo : Takes a new photo\n";
      welcome += "/unlock : Unlock the Door\n\n";
      welcome += "/lock : Lock the Door\n";
      welcome += "To get the photo please tap on /photo.\n";
      bot.sendMessage(chatId, welcome, "Markdown");
    }
```

Figure 13. Example: Telegram Command Parsing and Execution Logic

The alarm system, implemented on a separate Arduino Uno, features logic that evaluates ambient light, motion, and sound thresholds sequentially. The buzzer is triggered only when low-light conditions are detected and followed by either motion or loud noise events. A representative code snippet of this logic is shown in Figure 14. The modular code structure supports easy updates and debugging. Threshold values for light and sound sensors were fine-tuned during testing to optimise performance and reduce false positives, demonstrating the flexibility of the Arduino environment in real-world security applications.

```
  bool isDark = ldrValue > ldrThreshold; // Check if it's dark
  bool isMotionDetected = digitalRead(pirPin) == HIGH; // Check if motion is detected
  bool isSoundDetected = soundValue >= soundThreshold; // Check if sound is detected

  // If it's dark and motion is detected or sound is detected, turn on the buzzer
  if ((isDark && isMotionDetected) || isSoundDetected) {
    digitalWrite(buzzerPin, HIGH); // Turn on the buzzer
  } else {
    digitalWrite(buzzerPin, LOW); // Turn off the buzzer
```

Figure 14. Example: Alarm System Logic for Light, Motion, and Sound Detection

The results confirm that the proposed smart door lock and alarm system meets its intended objectives. The lock system delivers rapid response, visual identification, and remote access using common mobile devices. The alarm system efficiently monitors environmental conditions and alerts occupants through a buzzer triggered by motion and sound, especially at night. The Telegram bot ensures user-friendly control, while the system architecture allows future enhancements, such as biometric integration or connection to emergency services.

IV. TECHNICAL DISCUSSION AND EVALUATION

The development and testing of the IoT-based smart door lock and alarm system revealed several key insights regarding system performance, practicality, and limitations. This section discusses the technical implications of the experimental results and evaluates the system against commercial standards.

The observed system response time of under five seconds from user command to device action is considered acceptable for residential applications. While not instantaneous, this delay provides a reasonable balance between functionality and security, especially in non-critical environments. Many commercially available smart locks, such as those from Yale, August, or Aqara, report average response times ranging from three to seven seconds, depending on network conditions, mobile app latency, and cloud service delays. The performance of the proposed system thus falls within industry norms, demonstrating its competitiveness despite being built with low-cost, open-source components. Furthermore, the complete operational cycle, including push button activation, image capture, remote command execution, and feedback notification—was completed in less than one minute. This confirms that the system is suitable for real-time home access control and visitor identification, even under varied network conditions.

In terms of usability, the Telegram Bot interface provided a lightweight yet effective communication platform for remote access. It ensured reliable synchronisation between the user's mobile device and the ESP32-CAM microcontroller, even when connected to different networks. However, this reliance on cloud-based communication also introduced vulnerabilities, such as potential latency during poor connectivity and dependency on the Telegram platform's stability.

Several challenges were encountered during the design and implementation process. Power management posed a significant issue due to the need for both 12V and 5V supplies for different components. The use of an LM7805 regulator helped resolve this, but additional thermal management was required to prevent overheating during continuous operation. Another challenge was the calibration of environmental sensors. The PIR and sound sensors initially produced false positives in noisy or humid conditions, which were later mitigated by adjusting threshold values and incorporating LED indicators for status debugging.

Additionally, the system required safeguards to prevent unintended activation due to signal noise or jitter. This was addressed through the implementation of debounce logic and a timeout routine in the firmware. From a security standpoint, the system was programmed to accept commands only from a predefined Telegram user ID, offering basic access control. While this method is effective for personal use, it does not offer the same level of encrypted multi-factor authentication found in proprietary commercial systems.

Despite these limitations, the proposed system presents a practical and cost-efficient alternative to existing smart locks. It is especially suitable for academic research, small-scale deployment, or users seeking a customisable solution. Its modular design and use of open-source tools make it highly scalable and adaptable for future integration with biometric authentication, local storage, or voice assistant compatibility. The ability to perform over-the-air firmware updates further enhances long-term usability and maintenance.

V. CONCLUSION

This paper presented the development and implementation of an IoT-based Smart Door Lock and Alarm System that integrates the ESP32-CAM module, cloud-based Telegram Bot interface, and multi-sensor alarm mechanisms to enhance home security and remote accessibility. The system successfully demonstrated the ability to perform core functionalities such as photo capture, lock/unlock control, and real-time alerts with minimal delay, achieving a response time of under 5 seconds and completing a full access cycle in less than a minute.

The smart door lock system was controlled remotely via the Telegram cloud bot, which served as a secure and intuitive user interface. The integration of motion detection, sound thresholding, and ambient light sensing provided an adaptive alarm subsystem that functions effectively in various environmental conditions, especially during nighttime or intrusion scenarios.

The modular design of both the hardware and software components ensures that the system is scalable and extensible, allowing future incorporation of biometric authentication, voice assistant integration, or additional IoT services. Security was enhanced by limiting access to authenticated Telegram user IDs and enabling multi-device synchronisation through the cloud.

This project demonstrates how low-cost hardware, when paired with cloud services and mobile integration, can yield powerful and reliable smart home security solutions. The system's performance under testing conditions confirms its viability for deployment in residential settings, particularly where affordability and remote access are key concerns. Moving forward, the system can be upgraded with machine learning-based intrusion prediction or cloud analytics, further aligning with the goals of smart city infrastructure and user-centric IoT ecosystems.

For future enhancement, the system can be expanded to include biometric verification such as fingerprint or facial recognition to increase security and reduce reliance on remote commands. Integration with cloud platforms like Firebase or AWS would enable real-time data logging, system health monitoring, and access history analytics. Additionally, incorporating voice assistant compatibility, such as Google Assistant or Alexa, could improve accessibility and allow hands-free operation. Enhancing the alarm module with a camera-based motion detection system and enabling offline functionality using local storage or GSM modules would also ensure continuous operation during internet outages, making the system more robust and suitable for deployment in diverse residential and commercial settings.

VI. ACKNOWLEDGEMENT

This research was supported by the Q436 TIER1 Grant from Universiti Tun Hussein Onn Malaysia. The authors also thank the Faculty of Electrical and Electronic Engineering and Faculty of Engineering Technology, UTHM, for providing the facilities and support throughout this study.

VII. CONFLICT OF INTEREST

Authors declare that there is no conflict of interest regarding the publication of the paper.

VIII. AUTHOR CONTRIBUTION

The authors confirm contribution to the paper as follows: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation: Mustafa Ali Mohammed Ahmed and Rosli Omar; review and validation of the final manuscript: Jumadi Abdul Sukor, Muhammad Zulhilmi and Jamaludin Jalani. All authors reviewed the results and approved the final version of the manuscript.

IX. REFERENCES

Ahmed, D, Ali, SZ, Khan, FB & Faruki, MH 2024, 'Design and implementation of an ESP32-based smart home automation system with environmental monitoring and automated controls', IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), vol. 19, no. 6, Ser. 1, pp. 23–28.

Al-Ali, AR & Al-Rousan, M 2004, 'Java-based home automation system', IEEE Transactions on Consumer Electronics, vol. 50, no. 2, pp. 498–504.

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, vol. 17, no. 4, pp. 2347–2376.

Anwar, S & Kishore, D 2016, 'IoT-based smart home security system with alert and door access control using smart phone', International Journal of Engineering Research & Technology (IJERT), vol. 5, no. 12, pp. 504–509.

Balaji, M, Padmaja, N, Swathi, N, Atheek, S, Manasa, M & Kumar, KC 2023, 'Biometric-based smart door locking system using biometric and OTP', in Proceedings of the 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, pp. 898–903.

Dey, S, Roy, A & Das, S 2016, 'Home automation using Internet of Thing', in Proceedings of the 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE, pp. 1–6.

Dzogovic, B, Santos, B, Noll, J, Do, VT, Feng, B & Do, TV 2019, 'Enabling smart home with 5G network slicing', in Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), IEEE, pp. 543–548.

Goyal, PK, Giri, M & Verma, S 2024, 'IoT-based smart door lock system with face recognition using ESP32 CAM and Android app', in Innovations in Electrical and Electronic Engineering, Lecture Notes in Electrical Engineering, Springer, pp. 365–376.

Kurniasyah, I & Rakhmawati, L 2024, 'Smart door lock innovation using integration of Bluetooth Low Energy and MQTT IoT protocol', INAJEEE: Indonesian Journal of Electrical and Electronics Engineering, vol. 8, no. 1, pp. 41–46.

Mahalingam, MS, Kumar, NS, Sivan, A, Kumar, S, Kumar, A & Anand, A 2024, 'Next-gen home security framework with facial recognition and IoT integration', in Proceedings of the 2024 International Conference on Sustainable Communication Networks and Application (ICSCNA), IEEE, pp. 103–107.

Mujeye, S 2024, 'Prototype design of securing the Internet of Things in smart homes', Online Journal of Applied Knowledge Management, vol. 12, no. 2, pp. 40–58.

Mutiarani, M & Ritonga, RR 2025, 'IoT LED control system implementation and optimization using Wi-Fi through a Telegram bot', Journal of Computer Science and Informatics Engineering, vol. 4, no. 1, pp. 10–20.

Nakandhrakumar, RS, Venkatasamy, R, Dhanraj, JA, Aravinth, S, Balachandar, K & Dhamodharan, N 2022, 'Design and development of IoT-based smart door lock system', in Proceedings of the 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), IEEE, pp. 1525–1528.

Paraschiv, E-A, Bica, O, Cristian, P & Ianculescu, M 2021, 'IoT-based ambient parameters monitoring device for in-home elderly care', in Proceedings of the 2021 23rd International Conference on Control Systems and Computer Science (CSCS), IEEE, pp. 495–500.

Patil, N, Nimbalkar, A, Pawar, G & Shardul, S 2024, 'Home automation system using ESP32 and Alexa', International Journal for Research in Applied Science and Engineering Technology, vol. 12, no. 4, pp. 2735–2739.

Rahman, M, Karim, AHMZ, Nyeem, S, Khan, F & Matin, G 2015, 'Microcontroller based home security and load controlling using GSM technology', International Journal of Computer Network and Information Security, vol. 7, no. 4, pp. 29–36.

Sahat, SJ, Tantya, J, Bandung, Y, Subekti, LB & Tanjung, D 2018, 'Internet of Things: Home monitoring and power management control based on Android', in Proceedings of the 2018 International Symposium on Electronics and Smart Devices (ISESD), IEEE, pp. 1–6.

Sarbishaei, G, Modarres, AMA, Jowshan, F, Khakzad, FZ & Mokhtari, H 2024, 'Smart home security: An efficient multi-factor authentication protocol', IEEE Access, vol. 12, pp. 114399–114416.

Seera, S & Behera, G 2025, 'The future of living smart home automation', International Journal of Scientific Research in Engineering and Management, vol. 9, no. 4, pp. 1–9.

Sharma, S, Mittal, P & Anuradha 2022, 'IoT-based smart security system for agriculture fields', in K Khanna, VV Estrela & JJPC Rodrigues (eds), Cyber Security and Digital Forensics, Lecture Notes on Data Engineering and Communications Technologies, vol. 73, Springer, pp. 143–151.

Shin, D, Yun, K, Kim, J, Astillo, PV, Kim, J-N & You, I 2019, 'A security protocol for route optimization in DMM-based smart home IoT networks', IEEE Access, vol. 7, pp. 142531–142550.

Simatupang, JW & Tambunan, RW 2022, 'Security door lock using multi-sensor system based on RFID, fingerprint, and keypad', in Proceedings of the 2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), IEEE, pp. 453–457.

Singh, AK, Laxmi, NH, Shamith, KM & Keshav, M 2024, 'Design and implementation of smart door lock system using IoT', in Proceedings of the 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), IEEE, pp. 1–4.

Skouby, KE & Lynggaard, P 2014, 'Smart home and smart city solutions enabled by 5G, IoT, AAI and CoT services', in Proceedings of the 2014 International Conference on

Contemporary Computing and Informatics (IC3I), IEEE, pp. 874–878.

Yan, Z, Zhang, P & Vasilakos, AV 2014, 'A survey on trust management for Internet of Things', Journal of Network and Computer Applications, vol. 42, pp. 120–134.

Zia, M, Siddiqua, M, Ouameur, M, Bagaa, M & Al-Turjman, F 2025, 'Securing the future: A survey on smart home security in IoT-integrated smart cities', Advances in Networks, vol. 12, no. 1, pp. 1–18.