



A Security Home System Feature as a Part of Home Assistant through IoT Platform

Indra Yasri*, Ahmad Al-Alif, Faril Pirwanhadi

Department of Electrical Engineering, Faculty of Engineering, Universitas Riau, Jl. HR Soebrantas KM. 12.5, Simpang Baru, Binawidya Pekanbaru, 28293 Riau, Indonesia

Article Info	Abstract
Article history: Received Nov 30 th , 2024 Revised Jan 25 th , 2025 Accepted Mar 12 th , 2025 Published Jun 30 th , 2025	The rapid development of Internet of Things (IoT) revolutionized the method to secure and monitor our homes. As smart home technology becomes increasingly accessible, the integration of robust, scalable, and cost-effective security systems is no longer a luxury but a necessity. This study proposed the creation and development of a home security system integrated with Home Assistant and Tailscale VPN using the ESP32-CAM WROVER microcontroller. By leveraging the IoT ecosystem, the system features real-time door detection, intrusion alarms, and remote monitoring capabilities. A reed switch sensor detects door or window activity, while the ESP32-CAM WROVER captures and transmits images of potential intrusions to a Linux-based Hass.io server. Notifications and images are sent to the user's smartphone for real-time updates, ensuring timely responses to security threats. The system is secured via a Tailscale VPN, allowing authorized users seamless access through MagicDNS. The research highlights the cost-effectiveness and efficiency of utilizing Linux-based virtual machines for smart home applications, as well as the performance advantages of the ESP32-CAM WROVER microcontroller. System testing performed by a single user with several activation attempts shows a success rate of ~93.33% for the alarm output, with a ~6.67% error rate attributed to rapid state changes or hardware noise. In contrast, the camera snapshot component demonstrated 100% reliability, with no errors in capturing or transmitting images. Comprehensive tests confirm the system's ability to enhance home security with its scalable and flexible architecture, making it a practical solution for modern smart homes.
Index Terms: IoT Home Assistant ESP32-CAM WROVER Tailscale	

This is an open access article under the [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) license.



*Corresponding Author: indra.yasri@eng.unri.ac.id

I. INTRODUCTION

In recent years, technological and network development has grown rapidly, particularly in the realm of the Internet of Things (IoT) [1]. IoT serves as a foundation for integrating sensor devices that connect directly to computers via the internet [2]. The rapid advancement of technology has penetrated into all aspects of life, contributing significantly to reducing crime rates and improving security systems [3]. Security systems are essential for preventing theft and other crimes. These systems are designed to curb the increasing number of theft incidents reported each year [4].

IoT connects physical devices to the internet, enabling automatic data exchange without the need for direct human intervention, creating a smarter and more efficient ecosystem [5]. IoT enables devices to be identified and interact with each other anywhere and anytime [6]. As wireless and microelectromechanical systems (MEMS) technologies continue to evolve, IoT adoption expands, with RFID and wireless sensors being commonly used communication

methods [7]. In this regard, security systems also benefit from IoT, as various components such as sensors, surveillance cameras, and alarms can interconnect and communicate with each other to detect, prevent, and respond to security threats more efficiently [8]. Home Assistant, an open-source platform for managing smart home devices, leverages this IoT ecosystem to integrate various devices from different manufacturers into a unified interface, providing users with full control [9]. The use of Home Assistant on a Linux-based server offers optimized performance and cost efficiency compared to hardware-based alternatives such as the Raspberry Pi, making it an appealing choice for many users [10].

Previous studies have explored IoT-based security systems. from Qasim et al. (2020) developed an Arduino and Raspberry Pi-based smart home security system capable of detecting humidity, gas, body temperature, and movement. Their Android-based prototypes collected sensor data and sent it to homeowners [11]. Similarly, research by Susanto and Nurcahyo (2020) introduced smart home security system designed to detect theft and fire hazards while enabling voice-

controlled electronic device management, beneficial for the elderly and people with disabilities. This system used Google Assistant on an Android application to control lights, detect movement with sensors that activate cameras, and detect fires using temperature and humidity sensors. The monitoring data was displayed on an LCD screen and sent to the Thingspeak IoT platform [12]. In addition, Omodunbi et al. (2024) developed a real-time monitoring system to monitor student behavior offline and online in the exam room, capable of identifying rule violations using video evidence and facial recognition. The system not only detected suspicious movements but also recorded video evidence to provide definitive proof of misconduct [13].

To overcome this problem, the authors designed a home security system integrated with Home Assistant and Tailscale using ESP32-CAM-WROVER. This system features automated detection and monitoring with flexible access through an encapsulated VPN connection, allowing authorized users to connect from anywhere. Upon detecting activity, the system automatically triggers an alarm mounted near the camera. The alarm is designed to distract the intruder while the camera captures multiple images of the event and transmits them to the web server. In addition, the system sends a notification to the homeowner's Android phone, alerting them of the attempted intrusion. This allows the homeowner to remotely monitor their property, and if necessary, report the incident to the authorities. The captured images, along with timestamps, are stored on the web server and can be downloaded as evidence when needed.

II. METHODOLOGY

This research evaluates and compares various microcontrollers, system architectures, flowcharts, hardware components, and their integration within a cohesive system. The hardware utilized in this project is selected to achieve an optimal balance of cost and performance, offering a user-friendly, low-power Wi-Fi-based system that is easily accessible. The system operates on a web-based platform, secured through a private VPN connection, and relies on a dedicated private server for its core functionalities and data storage.

A. Microcontroller Comparisons

This study focuses on identifying trends in microcontrollers commonly used in IoT applications, emphasizing devices with optimized performance and superior processing capabilities. The ESP32-CAM WROVER was selected for its advantages over other microcontrollers, such as the ESP8266 and Arduino Nano. The ESP32-CAM WROVER features a dual-core 240 MHz processor, significantly outperforming the ESP8266, which operates at 80 MHz or 160 MHz, and the Arduino Nano, limited to 16 MHz. Although similar to the ESP32-CAM in technical specifications, the ESP32-CAM WROVER offers superior reliability and cost-efficiency. Unlike the ESP32-CAM, which requires an additional development board, the ESP32-CAM WROVER includes a micro USB connector, simplifying its deployment. While a USB Type-C variant is available, the micro USB version is more cost-effective, with negligible differences in performance for most applications, aside from marginally faster compile times when using USB Type-C.

Additionally, the ESP32-CAM WROVER includes 4 MB of PSRAM (Pseudo-Static Random Access Memory), enabling it to handle complex tasks such as facial recognition and high-resolution image processing more effectively than its counterparts. However, these capabilities are contingent on a stable network connection to ensure optimal image resolution and system performance. Detailed specifications are presented in Table 1.

Table 1
Comparisons Between Various IoT Microcontroller

Feature	ESP32-CAM WROVER	ESP32-CAM	ESP8266	Arduino Nano
Processor	Xtensa LX6 (32-bit, 240 MHz)	Xtensa LX6 (32-bit, 240 MHz)	Tensilica L106 (32-bit, 160 MHz)	ATmega328P (8-bit, 16 MHz)
Clock Speed	240 MHz (dual-core)	240 MHz (dual-core)	80/160 MHz (single-core)	16 MHz
RAM	520 KB SRAM, 4 MB PSRAM	520 KB SRAM	80 KB SRAM	2 KB SRAM
Flash Memory	4 MB (external)	4 MB (external)	512 KB - 4 MB	32 KB
Wi-Fi	Yes (802.11 b/g/n)	Yes (802.11 b/g/n)	Yes (802.11 b/g/n)	No
Bluetooth	Yes (BLE and Classic)	Yes (BLE and Classic)	No	No
GPIO Pins	9 GPIO	9 GPIO	17 GPIO	14 Digital, 8 Analog
Camera Support	Yes (OV2640)	Yes (OV2640)	No	No
USB Support	Yes	No	Yes	Yes
Power Supply	3.3V	3.3V	3.3V	5V

B. System Integrations

The development of home security system based on the ESP32-CAM WROVER and integrated with Home Assistant begins with a comprehensive literature review to identify core concepts and analyze prior research on component specifications. This review facilitates the selection of optimal components and microcontrollers to address existing challenges. Hardware components such as reed switches and magnets are chosen to simplify the design, enhance system performance, and minimize costs. The ESP32-CAM WROVER was selected for its favorable price-to-performance ratio.

The system leverages the Home Assistant IoT platform [14], supported by the ESPHome framework, and operates on a Linux-based Hass.io server hosted on an Oracle VirtualBox virtual machine for cost-effective deployment. To ensure secure access, Tailscale VPN is implemented, allowing only authenticated users linked to Google accounts to access the system. Once the design is finalized, the system undergoes rigorous performance evaluation, with findings documented upon successful completion.

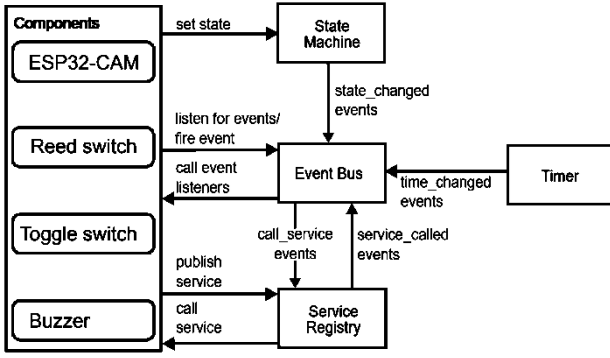


Figure 1. Home Assistant System Architecture

C. System Components

The system's components are categorized into hardware and software. Hardware components include the ESP32-CAM WROVER microcontroller and reed switch sensor, while the software components consist of the IoT platform (Home Assistant), server components (Hass.io), and Tailscale VPN for secure access. Together, these components form the backbone of the system, enabling seamless functionality and achieving the project's objectives.

1) ESP32-CAM WROVER

The ESP32-CAM WROVER is a microcontroller that operates with a 5V supply voltage, regulated to a stable 3.3V by an onboard voltage regulator [14]. This module is widely employed in IoT applications that require not only image and video capture but also the capability to process and wirelessly transmit data. Its dual communication capabilities via WiFi and Bluetooth make it particularly well-suited for scenarios requiring real-time data processing and remote connectivity.

2) Reed Switch

The reed switch is a simple, yet effective sensor composed of two closely positioned metallic plates. This device functions as a circuit activator that closes the circuit when exposed to a magnetic field within its activation range. When a sufficiently strong magnetic field is present, the metallic plates inside the switch come into contact, completing the circuit and enabling operation within the connected system [15].

3) Home Assistant

Home Assistant (HA) is a free, open-sourced client-server platform built to manage, monitor, and control smart devices in a single system, regardless of their manufacturer [16]. Developed collaboratively by a global community of skilled users and developers, HA provides a seamless integration of various automation devices within a home or building network [17]. Its versatility and user-driven development have made it a cornerstone for home automation projects.

4) Hass.io

Home Assistant OS, formerly known as Hass.io, is a Linux-based operating system tailored for running Home Assistant. It simplifies the installation and management of Home Assistant by supporting add-ons and delivering a robust platform for embedded devices. Built on HassOS, it offers features such as a lightweight design, a compressed root file system, secure update mechanisms, and low resource consumption. These capabilities allow for efficient containerized application deployment via Docker, ensuring

smooth integration of devices and services within a unified framework [18].

5) Tailscale

Tailscale is a VPN platform that enhances security, particularly when using public Wi-Fi networks. It facilitates seamless connectivity across devices and simplifies remote access to files and resources between work and home environments [19]. In this project, Tailscale is employed to deploy a secure private connection between user devices and the Hass.io server. It features a streamlined configuration process and access control limited to authorized accounts, ensuring reliable and secure communication for the system.

D. System Flowchart

Figure 1 illustrates the system flowchart for the home security system. The system begins by monitoring the status of doors and windows using reed switches. If no compromise is detected, the system continues monitoring. However, if a door or window is breached, the reed switch is triggered, initiating the next steps in the process. The ESP32-CAM captures images over a set period and these images are transmitted via Wi-Fi to the Home Assistant platform. Home Assistant stores the images on a virtual machine server and simultaneously sends a notification to the user, alerting them to a potential intruder. The user can then access the images to evaluate the situation and identify any intruders. If further action is deemed necessary, the owner reports the incident to the appropriate authorities or responsible party. If no further action is needed, the process concludes, and the system resumes monitoring.

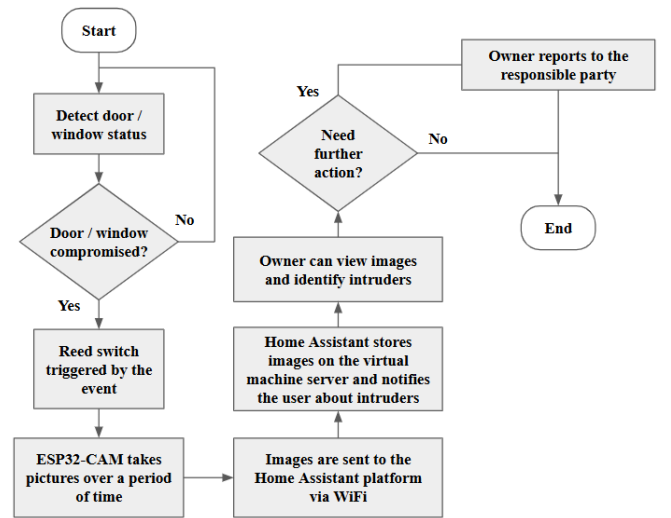


Figure 2. System Flowchart of the Security Home System feature as a part of Home Assistant through IoT platform

E. System Schematic

In this study, the ESP32-CAM-WROVER serves as the microcontroller and a camera module, integrated with a reed switch sensor, toggle switch, buzzer, and power bank. Figure 2 depicts the configuration of these components to create the system schematic.

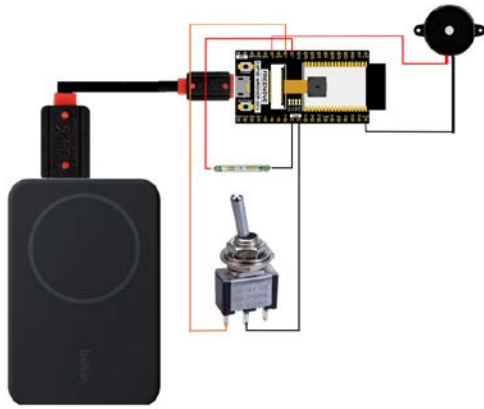


Figure 3. System Schematic in Research

F. Nomenclature Pin Configuration



Pin configuration nomenclature refers to the system used to label or assign specific names to the pins on electronic components, such as microcontrollers or sensors, to denote their functions or connectivity requirements. Each pin is assigned a unique role, and Table 2 outlines the nomenclature utilized in this study.



Table 2
Nomenclature Pin Configuration

GPIO Pin Configurations	Connection
GPIO 14	Reed Switch
GPIO 12	Toggle Switch
GPIO 13	Buzzer

The essential equipment features required for this research are presented in Table 3 [20-23]. These features are systematically organized to ensure that all critical components are adequately prepared to support the system's effective and efficient operation. By incorporating these features, the developed system is expected to align with the research objectives and achieve the desired quality standards.

Table 3
Required Equipment Set Features

Equipment	Key Features	Figure	Reference
ESP32-CAM WROVER	<ul style="list-style-type: none"> Input Voltage: 7-12 V Operating voltage: 3 ~ 3.6 V Analog Input Pins: 1 (GPIO36) Digital GPIO Pins: 9 Xtensa LX6 (32-bit, 240 MHz) Camera Module: OV2640 		[16]
Reed Switch	<ul style="list-style-type: none"> Switching Voltage (Max): 24V Switching Current (Max): 0.1A Operating Distance: 10-15 mm Response Time: 0.2-2 ms 		[17]

Toggle Switch	<ul style="list-style-type: none"> Input Voltage: Up to 250V AC or 30V DC Current Rating: 0.2A - 2A Switch Mode: On/Off 		[18]
Buzzer	<ul style="list-style-type: none"> Input Voltage: 3V Rated current:<30mA Height 9mm Resonant Frequency : 2300 ±300Hz Type: Active 		[19]

III. RESULT AND DISCUSSION

The research findings are organized into five distinct sections. The first section focuses on door detection using a reed switch, which plays a crucial role in identifying the status of the door (open or closed) as part of the security system. The second section details the development of an intruder alarm system, which leverages the capabilities of the ESP32-CAM-WROVER module to capture and monitor unauthorized entries. The third section highlights the integration of a Hass.io server, built on a Linux-based Virtual Machine, demonstrating its function as the central control hub for managing and coordinating system components. The fourth section discusses the activation of the camera and the transmission of data using the ESP32-CAM-WROVER, explaining how it processes and relays real-time visual information to the server for further analysis and response. Finally, the last section discusses VPN connectivity through Tailscale, demonstrating how the integration allows users to securely access their smart home systems from any location.

A. Door Detection Using Reed Switch

In this research, the door detection system using a Reed Switch was successfully implemented. The Reed Switch is mounted with a cable, while a magnet is simply attached or removed to the reed switch to illustrate the opening or closing of the door. Tests results indicate the following behavior: When the door is opened, the magnet approaches the Reed Switch, causing the contacts inside the Reed Switch to close and triggering a buzzer sound. Conversely, when the door is closed, the magnet distances away from Reed Switch, causing the metal contacts to disconnect and the buzzer to remain silent. These results confirm that the Reed Switch is capable of detecting changes in door position with high accuracy, as well as providing a clear indication of the door's status.

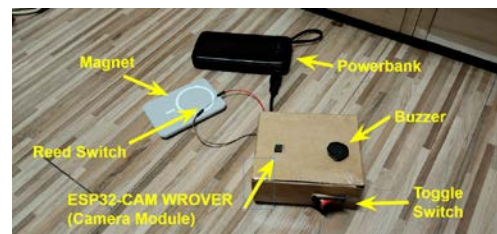


Figure 4. System Contraption of Final Result of Security Home System Feature as a Part of Home Assistant through IoT Platform

B. Intruder Alarm System with ESP32-CAM WROVER

The door detection system is integrated with an intruder alarm system, using the ESP32-CAM WROVER module which reads real-time sensor data from the Reed Switch and sends the information to other devices via a Wi-Fi network. When the sensor is triggered and the system is operational, the ESP32-CAM WROVER sends an alarm notification to

the user's device. The module activates the camera, capturing multiple images within a predetermined duration. Testing confirms that the notification is received by the user within a short time after the sensor is triggered, ensuring that the intruder alarm system works responsively. In addition, the images captured by the ESP32-CAM WROVER are sent via Wi-Fi connection to the Hass.io server for further processing and storage.

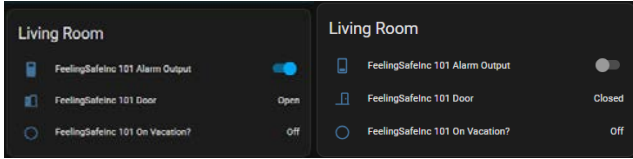


Figure 5. Home Assistant User Interface in reporting door conditions: open (left) and closed (right)

C. Role of Linux-based Hass.io Server in Virtual Machine

The Hass.io server, built on a Linux platform and operating within a virtual machine, serves the central hub for managing communication between the ESP32-CAM WROVER and the user's device. Testing demonstrates that the server efficiently receives data from the ESP32-CAM WROVER, processes the information, and delivers notifications and images to the user in a timely manner. This configuration establishes the Hass.io server as a highly effective control center, ensuring users are promptly alerted to any suspicious activity. The use of a virtual machine adds flexibility and scalability to the system, allowing seamless integration with additional devices and simplifying overall management. This design highlights the server's reliability and adaptability, making it an essential component of the security infrastructure.

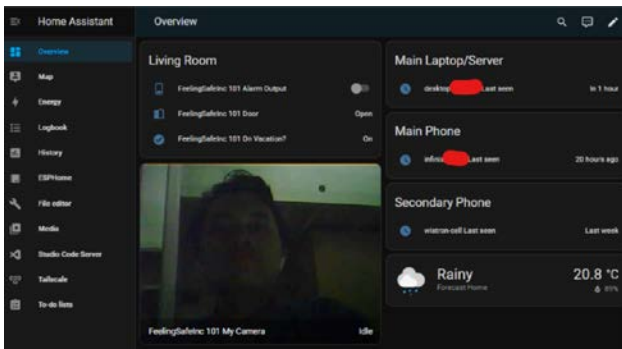


Figure 6. Hass.io Home Assistant Server User Interface

D. Camera Activation and Data Transmission with ESP32-CAM-WROVER

In addition to door detection, the system is designed to activate the ESP32-CAM-WROVER camera when the sensor detects an intrusion. This camera captures multiple images within a predetermined duration. However, full implementation of this image capture feature is still in progress. Once the images are captured, the ESP32-CAM-WROVER sends the data over a Wi-Fi connection to Hass.io. Upon receiving the images, Hass.io sends notification containing the captured to the user. This allows the user to view the captured images and identify the intruder for further action. Although the image capture functionality is still under development, the basic implementation of this system shows significant potential in improving home security by providing users with visual information in real time.

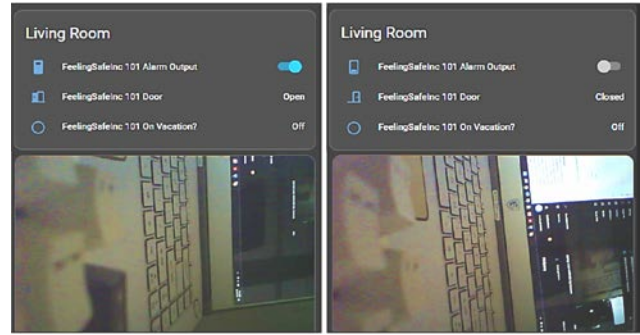


Figure 7. Home Assistant User Interface in reporting home surveillance of door condition: breached (left) and unbreached (right)

E. VPN Connectivity Through Tailscale on Any Devices

VPN connectivity through Tailscale enables seamless and secure remote access to the Home Assistant server from any device connected to the same Tailscale account, provided the device has an active internet connection. This setup leverages Tailscale's MagicDNS feature, which allows users to access the server using a custom DNS name, eliminating the need for complex network configurations. With this feature, users can interact with the Home Assistant server as if they were on the same local network, ensuring both convenience and security.

Figure 8 demonstrates screenshots from a smartphone accessing the Home Assistant server. These screenshots illustrate various functionalities including receiving notifications, viewing captured snapshots, monitoring live camera feeds, and checking the home's security status. The mobile interface mirrors the functionality of the desktop version, offering comprehensive access to the server, which runs in the background to ensure uninterrupted service. This level of integration ensures that users can effectively manage their smart home systems, regardless of location.

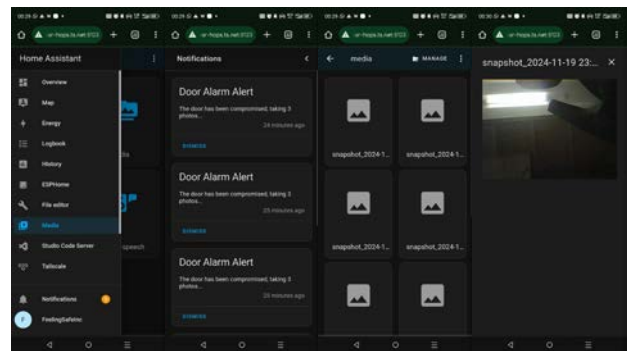


Figure 8. Screenshots from a smartphone demonstrating the access to the Home Assistant server

F. System Testing

The system testing data for the ESP32-CAM-WROVER Home Assistant home IoT security system evaluates the performance of both the alarm output and camera snapshot components. For the alarm output testing with the single user, the system achieved a success rate of approximately 93.33%, with two missed signals out of 30 activation attempts, resulting in an error rate of 6.67%, with failures occurring on the 14th and 22nd attempts. The errors are likely due to rapid state changes or hardware noise, which affected single detection. The camera snapshot testing demonstrated flawless performance, achieving a 100% success rate when the alarm

was activated. All three images were successfully captured and made accessible in the server media, yielding a 0% error rate. This reliable performance is attributed to the effective sequencing and delays integrated into the system's code, ensuring consistent operation.

IV. CONCLUSION

The study successfully developed an integrated home security system using IoT technologies, incorporating door detection, intruder alarms, camera activation, and remote access functionalities. The use of the Reed Switch for door detection proved highly effective in accurately monitoring door status, while the ESP32-CAM-WROVER module enhanced the system's capabilities by enabling real-time notifications and image capture upon detecting intrusions. The Hass.io server, operating through a Linux-based virtual machine, served as a reliable control hub, managing communication between the system components and delivering timely alerts to users. Furthermore, the integration of VPN connectivity via Tailscale ensured secure and convenient remote access, empowering users to monitor their home environment from any location.

System testing revealed that the alarm output function demonstrated a high success rate of 93.33%, while the camera snapshot feature performed flawlessly with a 100% success rate. These results confirm the system's robust performance and potential for enhancing home security. Despite some challenges with image capture implementation, the system shows great promise for future development, with potential improvements in its visual surveillance capabilities. Overall, this research demonstrates the viability and effectiveness of an IoT-based smart home security system, offering both convenience and reliability to users.

ACKNOWLEDGMENT

The author expresses heartfelt gratitude to the Universitas Riau, Faculty of Engineering, Department of Electrical Engineering, for support and research facilities provided during the preparation of this paper.

CONFLICT OF INTEREST

The authors stated there are no conflicts of interest related to the paper publication.

AUTHOR CONTRIBUTION

The authors declare their contributions to this paper as follows: design and concept were carried out by Indra Yasri and Ahmad Al-Alif; data collection was conducted by Ahmad Al-Alif and Faril Pirwanhadi; analysis and interpretation of the results were performed by Indra Yasri, Ahmad Al-Alif, and Faril Pirwanhadi; and the draft manuscript was prepared by Ahmad Al-Alif and Faril Pirwanhadi. All authors evaluated the results and acceded to the final version of the manuscript.

REFERENCES

- [1] Chandini Banapuram, Azmera Chandu Naik, Madhu Kumar Vanteru, V Sravan Kumar, Karthik Kumar Vaigandla, "A Comprehensive Survey on Internet of Things for Smart Cities : Applications, Communication Protocols, Network Types and Requirements", IJRITCC, vol. 11, no. 9, pp. 1773–1781, Nov. 2023.
- [2] G. G and D., Priscila, "IoT Evolution: Revolutionary Developments in Recent Years: IOT", International Journal of Information Technology, Research and Applications, vol. 3, no. 4, pp. 40–49, Dec. 2024.
- [3] A. U. Nabi, T. SNOUSSI, and Z. A. Abdalkareem, Trans., "A Review of IoT Convergence in Healthcare and Smart Cities: Challenges, Innovations, and Future Perspectives", BJIoT, vol. 2023, pp. 23–30, Apr. 2023.
- [4] L. Liu and A. R. Mishra, "Enabling technologies challenges of green Internet of Things (IoT) towards sustainable development in the era of Industry 4.0", TEDE, vol. 30, no. 2, pp. 344–375, Apr. 2024.
- [5] ZHANG, Tianyu, et al. A survey on industrial Internet of Things (IIoT) Testbeds for connectivity research. arXiv preprint arXiv:2404.17485, 2024.4, pp. 2233–2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.
- [6] I. Irvawansyah, U. Muhammad, M. Ihsan, A. Renanda, and K. Kurnia, "Prototype teknologi home assistant berbasis internet of things (IOT)," in Joule (Journal of Electrical Engineering), vol. 4, no. 1, pp. 16–21, Mar. 2023, doi:10.61141/joule.v4i1.376
- [7] ARYA, Shailesh D.; PATEL, Samir. Implementation of Google Assistant & Amazon Alexa on Raspberry Pi. arXiv preprint arXiv:2006.08220, 2020.
- [8] DOMÍNGUEZ-BOLAÑO, Tomás, et al. An overview of IoT architectures, technologies, and existing open-source projects. Internet of Things, 2022, 20: 100626.
- [9] M. N. Khan and M. Raisalat, "The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises," International Journal of Advanced Computer Science and Cloud Computing, vol. 6, no. 1, pp. 1–10, Jan.–Feb. 2024. [Online]. Available: <https://www.ijfmr.com/research-paper.php?id=22699>
- [10] Laghari, A.A., Li, H., Khan, A.A. et al. Internet of Things (IoT) applications security trends and challenges. Discov Internet Things 4, 36 (2024). <https://doi.org/10.1007/s43926-024-00090-5>.
- [11] H. H. Qasim, A. E. Hamza, H. H. Ibrahim, H. A. Saeed, and M. I. Hamzah, "Design and implementation home security system and monitoring by using wireless sensor networks WSN/internet of things IOT," in International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 3, p. 2617, Jun. 2020. doi:10.11591/ijece.v10i3.pp2617-2624
- [12] H. Susanto and A. Nurcahyo, "Design and implementation of a smart home security system using Voice Command and internet of things," in Khazanah Informatika : Jurnal Ilmu Komputer dan Informatika, vol. 6, no. 2, Aug. 2020. doi:10.23917/khif.v6i2.9320
- [13] R. A. Omodunbi, C. S. Odeyemi, O. M. Olaniyan, A. A. Soladoye and O. Awokoya, "Development Of An Internet Of Things (IoT) Based Surveillance System For Examination Supervision," in Ladoke Akintola University of Technology (LAUTECH) Journal of Engineering and Technology, vol. 18, no. 1, 2024, pp. 109–116.
- [14] Developers Home Assistant, "Core Architecture", Accessed: January 23, 2025, [Online]. Available: <https://developers.home-assistant.io/docs/architecture/core/>.
- [15] T. O. S. Yando, T. A. Wibowo, and D. A. Nurmantris, "Implementasi Sistem Keamanan Kendaraan Bermotor dengan Menggunakan Security Key dan Sensor Kecepatan," Proyek Akhir, Universitas Telkom, Bandung, Indonesia, 2014. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/pustaka/files/100466/jurnal_eproc/implementasi-sistem-keamanan-kendaraan-bermotor-dengan-menggunakan-security-key-dan-sensor-kecepatan.pdf.
- [16] L. Munteanu, M. C. Suvar, and G. D. Florea, "Residential security through the Home Assistant Platform," MATEC Web of Conferences, vol. 354, p. 00008, 2022. [Online]. Available: <https://doi.org/10.1051/mateconf/202235400008>.
- [17] "Automation Basics," Home Assistant Documentation, [Online]. Available: <https://www.home-assistant.io/docs/automation/basics>. [Accessed: Nov. 19, 2024].
- [18] Home Assistant Community, "New Hass.io images, based on HassOS," Home Assistant Blog, Jul. 11, 2018, Accessed: Nov. 19, 2024, [Online]. Available: <https://www.home-assistant.io/blog/2018/07/11/hassio-images/>.
- [19] D. F. Hrițcan and D. Balan, "Using Tailscale and PfSense for Security and Anonymity of IoT Environments," in 2024 17th International Conference on Development and Application Systems, DAS 2024 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 91–94. doi: 10.1109/DAS61944.2024.10541192.
- [20] Espressif Systems, "ESP32-WROVER-E & ESP32-WROVER-IE Datasheet", Accessed: Nov. 20, 2024, [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e_esp32-wrover-ie_datasheet_en.pdf.
- [21] Standex Electronics, "OKI Reed Switch ORD213 Datasheet", Accessed: Nov. 20, 2024, [Online]. Available:

- https://standelectronic.com/wp-content/uploads/OKI_Reed_Switch_ORD213.pdf.
- [22] TE Connectivity, "Switches Core Program Catalog", Accessed: Nov. 20, 2024, [Online]. Available: https://www.mouser.com/datasheet/2/418/3/Ng-Cs_1308111-1_Switches_Core_Program_Catalog_0308-1234899.pdf.
- [23] TDK Corporation, "Electromagnetic Buzzer Catalog", Accessed: Nov 20, 2024, [Online]. Available: https://product.tdk.com/en/system/files/dam/doc/product/sw_piezo/sw_piezo/em-buzzer/catalog/electromagnetic_buzzer_sd_en.pdf.