Smart Home Monitoring and Control with Human Detection and Telegram Messages using ESP32-Wrover CAM

Mochamad Fajar Wicaksono[1], Myrna Dwi Rahmatya [2]
and Aqsa Chalik Faturohman[3]

[1] Computer Technic, Universitas Komputer Indonesia, Bandung, Indonesia
[2] Informatic Management, Universitas Komputer Indonesia, Bandung, Indonesia
[3] Computer System, Universitas Komputer Indonesia, Bandung, Indonesia
mfajarw@email.unikom.ac.id

Abstract. This study aims to develop a smart home monitoring and control system using the ESP32-WROVER CAM module. Homeowners are able to control the system from their phone's Telegram app, and it can provide instant notifications if unauthorized entry, gas leakage, or fire is detected. The experimental method was used in this paper. The hardware design features the ESP32-WROVER CAM, a gas sensor, a flame sensor, a relay, a solenoid door lock, a buzzer, and a light. The software scripts utilize Arduino C, TensorFlow Lite library, and Telegram bot. The TensorFlow Lite model runs locally on the device to detect humans, and Telegram sends images, alerts, and commands to actuators. The tests showed that the system was able to detect humans, gas leakage, fire, and then act on commands from the homeowner with 100% percentage. The high performance of the system was due to the solidly designed hardware components and effective utilization of PSRAM memory on the ESP32-WROVER CAM for local inference with TensorFlow Lite, which allows the product to run reliably without internet access. In conclusion, the proposed system achieves its intended function by providing a design implementation for the four criteria of home monitoring and control.

Keywords: Smart Home, ESP32-Wrover CAM, Human Detection, Telegram Bot, Monitoring and Control.

1 Introduction

The advent of Internet of Things (IoT) technology has led to a considerable transformation in home security, especially with the utilization of smart sensors, image analysis, and real-time communication systems.

One of the most utilized technologies in developing IoT-based monitoring systems is the ESP32 microcontroller module. The modules available range from merely providing GPIO functions, wireless capabilities, and OLED display functions to modules with camera functions, such as the ESP32-CAM and ESP32-WROVER CAM.

L. Warlina and S. Luckyardi (eds.), *Proceedings of the 8th International Conference on Informatics, Engineering, Science & Technology (INCITEST 2025)*, Advances in Engineering Research 287,
https://doi.org/10.2991/978-94-6463-924-7_5

This study aims to develop a smart home monitoring and control system based on ESP32-WROVER CAM for enhancing home security, especially through visual monitoring of human presence, environmental conditions, and automated control of devices.

This study developed a smart home monitoring and control system with human detection and Telegram messages using ESP32-WROVER CAM. The proposed system is not only able to detect human presence in general through visual human detection but also sends real-time notifications via Telegram and provides automatic responses in the form of controlling household devices (such as turning on alarms or lights) to improve overall home security. By utilizing the ESP32-WROVER CAM, which has more GPIOs and a larger PSRAM capacity than the regular ESP32 CAM, this system is expected to be able to run a lightweight detection model locally (at the edge). These things are the advantages of this research. However, this system also has a drawback, namely, it still relies on a Wi-Fi connection for sending notifications, which can be an obstacle if there is a network disruption.

2 Literature Review

Various previous studies have attempted to implement home security systems using various types of ESP32 microcontroller modules with various approaches. A research conducted in 2022 successfully designed a home security system that combines an ESP32-CAM, a PIR sensor, a fire sensor, and a Telegram Bot [1]. While this system is adequate for baseline home monitoring scenarios, it has many limitations. One of the more evident limitations is that it relies on a PIR sensor for motion detection, and PIR sensors are not able to differentiate between humans or different objects that may generate motion, such as animals. Additionally, the system does not offer device control for things like automatic door locks, lights, and even environmental monitoring for leaks or lighting.

Another 2023 study created the Face Recognition Smart Door Lock system with ESP32 CAM that identifies faces to manage door access and transmits messages through Telegram [2]. This system can recognize registered people but will not detect humans it does not recognize, so the system is only capable of managing door access, not all-purpose security monitoring.

In 2023, additional research was performed utilizing an ESP32 CAM to recognize the faces of guests and match them against an internal database to allow or deny entry [3]. This study demonstrates the practicality of facial recognition systems in IoT-based setups, but again had similar shortcomings, where the system only responded to identified faces and did not anticipate threats from unknown humans..

In general, the three studies have major shortcomings in terms of the scope and intelligence of the detection system, which is still limited, such as not being able to distinguish the type of object detected, so that the potential for detection errors remains high, and some studies focus on facial recognition for access control but do not provide solutions to the presence of unknown individuals who can actually be a security threat.

To address these gaps, this study adopts several technologies. The MQ-2 gas sensor is employed due to its high sensitivity, low cost, and proven effectiveness in detecting LPG and other combustible gases [4]. Human detection is implemented using TensorFlow Lite on ESP32-WROVER CAM, which leverages on-device inference for distinguishing humans from non-human motion, offering greater accuracy than traditional PIR sensors. For communication, the Telegram Bot platform is chosen for its secure API, lightweight integration, and reliability in delivering real-time alerts and enabling remote user control [5, 6] . These components collectively enhance the robustness and intelligence of the proposed system compared to previous approaches.

3 Method

In this study, the experimental methodology was used to build the system [7, 8, 9, 10] . In accordance with the basic concepts that directed this research to accomplish the initial aims, a number of tests were conducted to put it into practice. This process consists of a study of literature, hardware and software design, simulation, implementation and testing to determine the system's dependability [11, 12]. These steps are illustrated in Fig. 1.

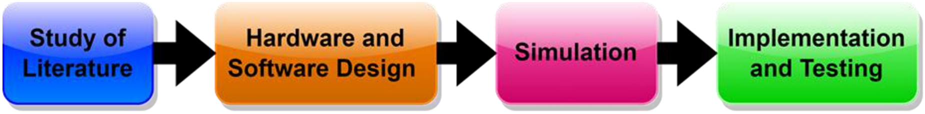

Fig. 1. The flow of the experimental method used.

3.1 Study of Literature

At this point, a review of prior research and theories about monitoring and control smart home was conducted. The proposed system in this study is functionally more complete and intelligent than the three previous studies [1, 2, 3]. Not only does it monitor the system, but it also reacts automatically to various types of threats (fire, gas, intrusion), and sends real-time notifications via Telegram.

3.2 Hardware and Software Design

In this section, hardware and software design are carried out to build smart home monitoring and control with human detection and telegram messages using ESP32-WROVER CAM. The algorithm program design approach is applied to the software design. Electronic circuits and system block diagrams for this learning medium are included on the hardware design side of things. Figure 2 displays the block diagram.

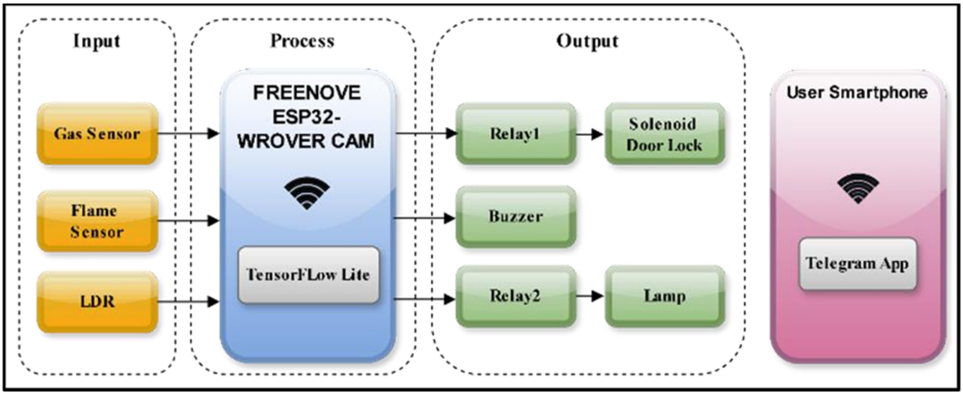

Fig. 2. Block diagram of the system.

Fig. 2 block diagram clearly demonstrates that the ESP32-WROVER CAM serves as the primary brain of this system. The ESP32-WROVER CAM is an ESP32-based microcontroller module equipped with 4–8 MB of PSRAM, enabling local image processing without relying on cloud computing. The use of PSRAM makes this module ideal for edge computing implementations, such as human detection and real-time image processing [13, 14]. The number of GPIOs on this module is 30. The number of usable GPIOs can be reduced if we use the camera feature. This microcontroller module is connected to the camera feature, 3 sensors (gas sensor, fire sensor, and LDR sensor), and 3 outputs, where two of them are connected to two relays and one of them to a buzzer.

On the input side, there are several sensors connected to the ESP32-WROVER CAM, specifically a gas sensor, fire sensor, and LDR sensor. The MQ-2 gas sensor module was used in this study. This sensor type detects the smell of LPG gas, which can be used to detect gas leaks [15, 16, 17, 18]. This sensor module operates on a voltage of 5V [8]. A digital value is the type of output read from this sensor module. In the current study, the MQ-2 sensor module was positioned in the kitchen. A fire sensor module is used to detect fire. In this study, the sensor module was placed inside the kitchen. The sensor module has a sensitivity range of 760 nm to 1100 nm, a maximum detection distance of 3 ft, and a detection angle of 60 degrees [19, 20, 21]. The result of this module is a numeric value in digital form. An LDR sensor was also utilized in this study to detect light. The placement of the sensor was on the porch of the house. The values read from this sensor are in an analog form. When the values read are outside the specified limits, the ESP32-WROVER CAM activates a relay which turns on the light. A relay is an electronic switch which is energized to operate once a current is passed through the coil [22]. Besides the sensor reading input, there exists a camera section. The camera captures an input image, which is analyzed by a special machine learning framework for the ESP32. The framework is known as TensorFlow Lite. TensorFlow Lite is a streamlined version of the TensorFlow framework. It was designed to run machine learning models on limited resources such as microcontrollers. TFLite Micro features a static interpreter with a small memory footprint that enables inference models to be run locally without dependence on a cloud connection. The representations used by this framework have been quantized or compressed to simple numeric data types (e.g., int8) to fit onto small processor architectures [23, 24]. In terms of technical application, the research utilizes a specific Arduino IDE library for the ESP32 and also employs a previously-trained model.

The output section includes two relays, a door lock solenoid, a buzzer, and a light. Relay 1 is connected to the door lock solenoid, while relay 2 is connected to the light. The digital output for the buzzer depends on the human detection value, gas sensor readings, and fire sensor readings. The output for relay 1 activates or deactivates the door lock solenoid based on user commands via the Telegram app. The final part of the output block is a light connected to relay 2. The value for relay 2 itself depends on the LDR sensor reading. Regarding messaging, this system uses a Telegram Bot to receive text or image messages and to issue commands to the system to perform desired actions [5, 6]. After the block diagram design process is complete, the next step is to design the electronic circuit for this system. The circuit design for this system is shown in the Fig. 3.

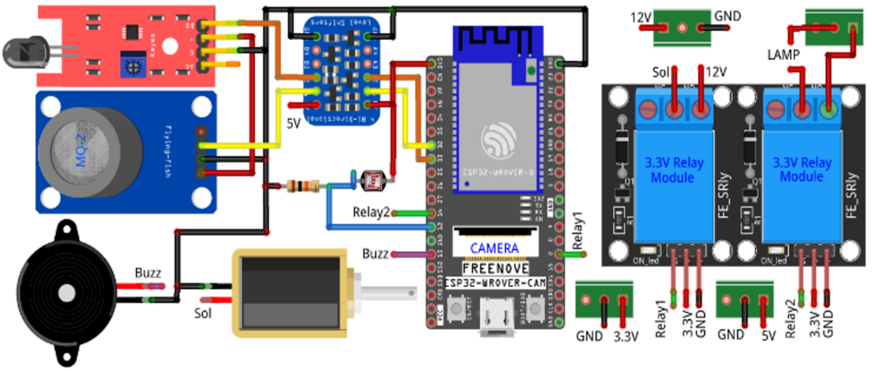

Fig. 3. Circuit design of the system.

Fig. 3 shows the circuit design of the system that has been created, which contains the ESP32-WROVER CAM, MQ-2 sensor, fire sensor, LDR sensor, two relay modules, solenoid, buzzer, and lamp. In addition, there is a bidirectional level converter module that is useful for changing the output level of the MQ-2 sensor and fire sensor, which is 5V to 3.3V. This is necessary because the highest voltage input limit for the ESP32 is 3.3V. After the circuit design process, the next step is to design the program algorithm according to the initial objective. The following is the program algorithm for this system, which is presented in the form of a text algorithm:

i. Start
ii. Initialize I/O and variables
iii. Initialize camera
iv. Initialize Wi-Fi connection
v. Initialize Telegram bot
vi. Detect human presence

a. If a person is detected:
- Capture image and send to Telegram
- Activate buzzer
- Proceed to step 7

b. If no person is detected:
- Proceed to step 7

vii. Read data from MQ-2 gas sensor

viii. Store value in variable gas and check:

a. If gas is LOW (gas leak detected):
- Activate buzzer
- Send warning message to Telegram
- Proceed to step 9

b. If gas is HIGH (no gas leak):
- Proceed to step 9

ix. Read data from flame sensor

x. Store value in variable flame and check:

a. If flame is LOW (flame detected):
- Activate buzzer
- Send warning message to Telegram
- Proceed to step 11

b. If flame is HIGH (no flame detected):
- Proceed to step 11

xi. Read data from LDR sensor

xii. Store value in variable light

xiii. Convert light value to digital and check:

xiv. Is the digital value ≤ 600?

a. If yes:
- Turn ON lamp via relay
- Proceed to step 15

b. If no:
- Turn OFF lamp via relay
- Proceed to step 15

xv. Check user commands from Telegram

a. Is there a command to capture an image?

1. If yes:
- Capture image and send to Telegram
- Proceed to step 16

2. If no:
- Proceed to step 16

b. Is there a command to activate solenoid door lock?

1. If yes:
- Activate door lock and send feedback to Telegram
- Proceed to step 16

2. If no:
- Proceed to step 16

c. Is there a command to deactivate solenoid door lock?
1. If yes:
- Deactivate door lock and send feedback to Telegram
- Proceed to step 16
2. If no:
- Proceed to step 16
d. Is there a command to activate buzzer?
1. If yes:
- Activate buzzer and send feedback to Telegram
- Proceed to step 16
2. If no:
- Proceed to step 16
e. Is there a command to deactivate buzzer?
1. If yes:
- Deactivate buzzer and send feedback to Telegram
- Proceed to step 16
2. If no:
- Proceed to step 16

xvi. Return to step 6

3.3 Simulation

To mimic the created program, a number of tests are conducted in this section. Both software and hardware program simulations are included in this simulation. Hardware program simulation is carried out by running the program and checking the output on the terminal. The simulations include I/O, sensor, actuator, and messaging to the Telegram.

3.4 Implementation and Testing

At this stage, all components are combined to implement smart home monitoring and control with human detection and telegram messages using ESP32-WROVER CAM. Testing is carried out based on the test scenario and the specified objective where the testing is done in a modular and overall manner.

4 Results and Discussion

The system created in this study allows users to monitor and receive alerts directly regarding the condition of their home. The system in this study is functionally more complete and intelligent, as evidenced by our system's 100% success rate in integrated testing, compared to the three previous studies that focused on a single functionality like motion or facial recognition [1, 2, 3]. Not only does it monitor, but it also reacts automatically to various types of threats (fire, gas, intrusion) and sends real-time notifications via Telegram. In addition, the use of ESP32-WROVER CAM allows better processing than the usual ESP32-CAM because of the additional PSRAM, so the system can handle human detection locally (edge computing). This conclusion is proven by the tests carried out, both modular testing and overall testing following the initial research objective scenario. Modular testing includes testing the flame sensor, testing the gas sensor, testing the LDR sensor, testing sending images to Telegram, testing sending string data to Telegram, and testing receiving commands from Telegram.

4.1 Gas and Flame Sensor Testing (Digital Output)

At this stage, data reading tests are carried out from the gas sensor and flame sensor. This test aims to ensure that the system can detect hazardous conditions such as gas leaks and the presence of fire. The gas sensor (MQ-2) and flame sensor produce digital values, with LOW indicating a hazard detection and HIGH indicating a normal condition. The results of this section of the test are shown in Table 1.

Table 1. Gas and flame sensor testing.

No	Gas Sensor	Flame Sensor	Buzzer	Status
1	HIGH	HIGH	Deactivated	Valid
2	HIGH	LOW	Activated	Valid
3	HIGH	HIGH	Deactivated	Valid
4	LOW	HIGH	Activated	Valid
5	LOW	HIGH	Activated	Valid
6	HIGH	LOW	Activated	Valid
7	HIGH	HIGH	Deactivated	Valid
8	HIGH	HIGH	Deactivated	Valid
9	LOW	HIGH	Activated	Valid
10	HIGH	HIGH	Deactivated	Valid

From ten tests, the results of which are shown in Table 1, the system successfully activates the buzzer automatically when gas (LOW) or fire (LOW) is detected. For example, in the 2nd and 6th tests, the flame sensor is LOW, and the system automatically activates the buzzer. This shows that the system's control logic against threats works well and consistently.

4.2 LDR Sensor Testing and Automatic Light Control

At this stage, testing is carried out on data reading from the LDR sensor. The LDR sensor in this system is used to measure the intensity of environmental light, where the results will be converted into digital values. A value of ≤600 indicates a dark environment and will turn on the lights, while a value of >600 indicates a bright environment and will turn off the lights. The results of this section of the test are shown in Table 2.

Table 2. LDR sensor testing and automatic light control.

No.	LDR Sensor	Relay2	Lamp	Status
1	≤600	Activated	On	Valid
2	>600	Deactivated	Off	Valid
3	>600	Deactivated	Off	Valid
4	>600	Deactivated	Off	Valid
5	>600	Deactivated	Off	Valid
6	≤600	Activated	On	Valid
7	≤600	Activated	On	Valid
8	>600	Deactivated	Off	Valid
9	≤600	Activated	On	Valid
10	>600	Deactivated	Off	Valid

The test results in Table 2 show that the system can automatically activate or deactivate the relay to control the lights. Test examples 1, 6, 7, and 9 show the system activates the lights when the light intensity is low, while in tests 2, 3, and 4, the system turns off the lights when the conditions are bright. This shows that the LDR-based automation function works accurately.

4.3 Image Capture and Image Send Testing

The testing in this section aims to ensure that the system can take pictures using the camera from the ESP32-CAM WROVER and send them to the Telegram bot. In this case, the image is taken automatically every 10 seconds. Table 3 shows the test results at this stage.

Table 3. Image capture and image send testing.

No.	Capture & Send Image	Telegram	Status
1	√	Received	Valid
2	√	Received	Valid
3	√	Received	Valid
4	√	Received	Valid
5	√	Received	Valid
6	√	Received	Valid
7	√	Received	Valid
8	√	Received	Valid
9	√	Received	Valid
10	√	Received	Valid

Based on 10 tests that have been carried out, the image was successfully sent to the user via the Telegram application. This indicates that the system integration process between the ESP32-WROVER CAM and the Telegram Bot is performing stably and responsively.

4.4 Testing Data Delivery to Telegram

This section involves the testing aspect, which includes sending data from gas and fire sensors, taking images, and sending text through Telegram. The purpose is to verify that the system can send sensor status data and send images to users. The results of testing this section are shown in Table 4.

Table 4. Testing data delivery to telegram.

No.	Data	Telegram	Status
1	Sent (Flame status)	Received	Valid
2	Sent (Gas Status)	Received	Valid
3	Sent (Photo and String)	Received	Valid
4	Sent (Photo)	Received	Valid
5	Sent (Photo and String)	Received	Valid
6	Sent (Flame status)	Received	Valid
7	Sent (Gas Status)	Received	Valid
8	Sent (Photo and String)	Received	Valid
9	Sent (Photo)	Received	Valid
10	Sent (Flame status)	Received	Valid

Based on the test results shown in Table 4, all messages were successfully sent and received by users with a valid status. This proves that the communication system runs smoothly and the Telegram API protocol can be accessed and used properly by ESP32-WROVER CAM.

4.5 Command Reception and Execution Testing

At this stage, testing is carried out to ensure that the system can receive commands from users via Telegram and carry out the requested actions, such as locking or unlocking the door (solenoid), activating or deactivating the buzzer, and taking photos. The results of this section of testing are shown in Table 5.

Based on the test results shown in Table 5, all commands were executed correctly, and the system sent feedback to Telegram as confirmation. This proves that the two-way communication between the system and the user is effective.

Table 5. Command reception and execution testing.

No.	Received Command	Action	Feedback	Status
1	/lock	Solenoid on	√	Valid
2	/unlock	Solenoid off	√	Valid
3	/photo	Photo captured	√	Valid
4	/buzzoff	Buzzer sound off	√	Valid
5	/photo	Photo captured	√	Valid
6	/lock	Solenoid on	√	Valid
7	/buzzon	Buzzer sound on	√	Valid
8	/unlock	Solenoid off	√	Valid
9	/buzzoff	Buzzer sound on	√	Valid
10	/photo	Photo captured	√	Valid

4.6 Overall Testing (System Integration)

After all modular tests related to the function of each part of the system are completed, the next step is to conduct an overall test that follows the initial objectives and program scenarios. Overall testing is carried out to ensure that the entire system functions in an integrated manner, such as human detection, gas and fire detection, light control based on LDR, device control, sending text and image data, and receiving Telegram commands.

The results of this test can also be viewed on the serial monitor. Everything related to the serial monitor is intended to ensure that the program on the ESP3-WROVER CAM runs according to the predetermined flow and to determine how long it takes from sending a command to executing the action.

Commands from the Telegram bot sent to the ESP32-WROVER CAM include opening and locking the door and activating and deactivating the alarm. Fig. 4 displays a snapshot of the results, which includes command reception, command response, and gas and fire detection results.

Fig. 4. Screenshot of the test results of 4 manual commands.

Fig. 4 shows four sections marked with blue boxes. These sections represent several commands successfully received and responded to by the ESP32-WROVER CAM.

The Telegram command messages used for door locks were "/lock" to lock and "/unlock" to unlock. Meanwhile, for alarms, the commands used were "/buzzon" to activate the alarm and "/buzzoff" to deactivate it. The time spent on the door lock activation and deactivation process, starting from the user sending the Telegram command to the ESP32-WROVER CAM receiving and executing the Telegram command, is 1 second each. Furthermore, the commands to activate and deactivate the alarm buzzer also take 1 second.

Furthermore, the section marked with a red box shows the results of the ongoing gas and fire detection process in the system. The results of testing the command to take a photo manually via the Telegram app and the image capture sent automatically by the ESP32-WROVER Cam when a human is detected can also be seen on the serial monitor shown in Fig. 5.

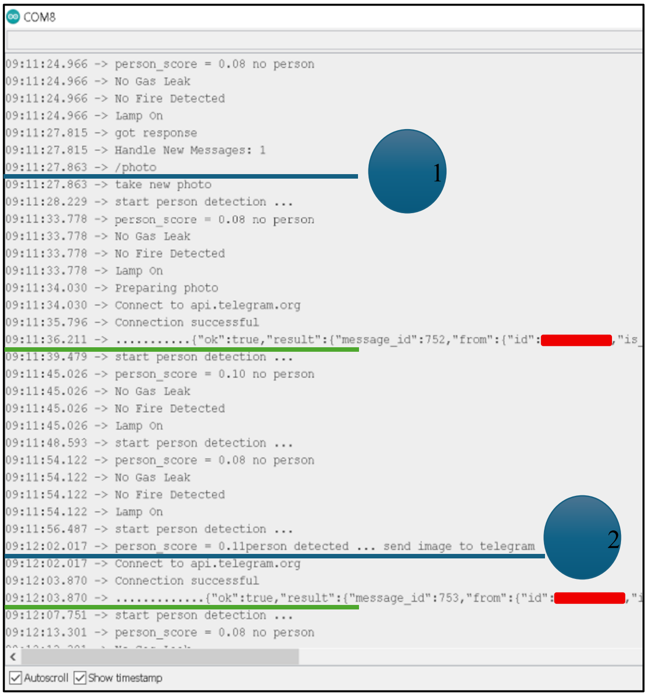

Fig. 5. Screenshot of the test results for taking and sending photos.

Fig. 5 shows the blue and green lines. The blue line indicates the initial command received, while the green line indicates the successful action, namely sending a photo to the Telegram bot. The section numbered 1 indicates that the manual photo-taking command has been received, and the section marked with a green line below it indicates that the image was successfully captured and sent to the Telegram bot. The image capture process took 9 seconds. Section number 2 shows the condition when the ESP32-WROVER CAM detects a human and takes the action of capturing and sending the image to the Telegram bot. The time required is 1 second. A screenshot showing one of the test results on the Telegram bot is shown in Fig. 6.

Fig. 6. Screenshot of the test results in telegram bot.

All the more complete test results related to the overall test results are presented in Table 6.

Table 6. Overall testing (system integration).

No	P	R	C	G	F	L	SD	B	L	T	S
1	x	x	x	x	x	≤600	on	off	on	-	Valid
2	√	x	√	x	x	>600	Prev state	on	off	√	Valid
3	x	Photo	√	x	x	>600	Prev state	off	off	√	Valid
4	x	Lock door	x	x	x	>600	on	off	off	√	Valid
5	√	x	√	x	x	≤600	Prev state	on	on	√	Valid
6	√	x	√	x	x	>600	Prev state	on	off	√	Valid
7	x	Un-lock door	x	√	x	≤600	off	on	on	√	Valid
8	x	x	x	x	√	≤600	Prev state	on	on	√	Valid
9	√	x	√	x	x	>600	Prev state	on	on	√	Valid
10	x	Buzzer on	x	x	x	>600	Prev state	on	off	√	

Description:
P: Person detected
R: Received Command
C: Capture Image
G: Gas leak
F: Flame detected
L: LDR
SD: Solenoid Door Lock
B: Buzzer
L: Lamp
T: Telegram
S: Status

Based on the tests that have been carried out, the system successfully detects and responds to various conditions simultaneously. For example, in the 5th and 6th tests, the system detects humans, sends photos, turns on the buzzer due to dangerous conditions, and turns on the lights due to low light intensity. This shows that the system is capable of monitoring and controlling the house automatically and in real-time, following the initial objectives of the study.

5 Conclusion

The ESP32-WROVER CAM system for monitoring and controlling smart homes with detection of humans and Telegram messages operates as desired with a 100% success rate as per the original objectives. The system can detect the presence of a human and send a telegram stating there is a human present, detect gas leaks, detect the presence of fire, and listen to commands from the homeowner through Telegram. Future work could also include the ability to control the system via a mobile app, log incidents in the cloud for data analytics, and make the system compatible with other smart devices via MQTT. These developments would enhance the system's overall scalability, interoperability, and functionality beyond just the ESP32-WROVER CAM. A significant limitation of this system is the need for a consistent and stable Wi-Fi connection to transmit notifications or alerts and to listen to commands. Due to Wi-Fi signal, the reliability of the system can decrease in areas with poor connectivity. Additionally, the system has only been tested in a controlled environment, so it is unclear how well it will work in a larger household or a household with more complexities. Future work could involve further improvement into mobile app control for the ESP32-WROVER CAM, logging data in the cloud for analytics, and extending support for additional smart devices via MQTT or other IoT smartphone protocols.

References

1. Cahyono, F.Y.A., Suharto, N., Mustafa, L.D.: Design and build a home security system based on an ESP32-CAM microcontroller with Telegram notification. Jurnal Jaringan Telekomunikasi **12**(2), 58–64 (2022).

2. Harun, N.B., Zainal, M.S.B.: Development of face recognition smart door lock system using ESP32-CAM and Telegram application as media control and monitoring. Progress in Engineering Application and Technology **4**(2), 035–048 (2023).
3. Khadafi, M., Pardede, A.M.H., Sembiring, H.: Design of a Guest Face Detection Tool Using ESP32-CAM Based on Internet of Things (IoT). Journal of Artificial Intelligence and Engineering Applications **4**(1), 124–130 (2024).
4. Daruwati, I., Hatika, R.G., Mardiansyah, D.: MQ-2 gas sensor using microcontroller Arduino Uno for LPG leakage with short message service as an information medium. In: Journal of Physics: Conference Series, **2049**(1), 012068 (2021).
5. Nobari, A.D., Sarraf, M.H.K.M., Neshati, M., Daneshvar, F.E.: Characteristics of viral messages on Telegram: the world's largest hybrid public and private messenger. Expert Systems with Applications **168**, 114303 (2021).
6. Jannati, N., Nakhaee, N., Yazdi-Feyzabadi, V., Tjondronegoro, D.: A cross-sectional online survey on patients' satisfaction using store-and-forward voice and text messaging teleconsultation service during the COVID-19 pandemic. International Journal of Medical Informatics, **151**, 104474 (2021).
7. Hidayat, H., Hasibuan, N.S., Wicaksono, F.: Design and implementation of electronic examination device for improving the blind students' comfort. J. Eng. Sci. Technol. **16**(1), 807–815 (2021).
8. Wicaksono, M.F., Rahmatya, M.D., Ilham, I.: IoT implementation for server room security monitoring using Telegram API. Int. J. Adv. Sci. Engineering and Information Technology. **12**(5), 1931–1937 (2022).
9. Wicaksono, M.F., Rahmatya, M.D.: Smart Cluster Housing Monitoring with ESP32, ESP-Mesh and Django.Journal of Engineering Research, 1–16 (2021).
10. Wicaksono, M.F., Rahmatya, M.D.: 3D Geometric Shape and Colors Interactive Learning Media using Raspberry Pi, OpenCV, and TensorFlow Lite. International Journal on Advanced Science, Engineering & Information Technology. **13**(5) (2023)
11. Wicaksono, M.F., Syahrul, S., Rahmatya, M.D.: Development of Laboratory Equipment Inventory System Using Radio Frequency and Internet of Things. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, **7**(2), 249–258 (2021)
12. Wicaksono, M.F.: IoT for Residential Monitoring Using ESP8266 and ESP-NOW. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, **8**(1), 93–106 (2022)
13. Jardian, J., Owen, M.: Smart Door Berbasis IoT dengan Deteksi Gerak dan Sistem Notifikasi melalui Aplikasi Blynk. Telcomatics **9**(2), 40–46 (2024)
14. Yudha, P.G.A.P., Kurniawan, E., Ratnaningsih, D.: Rancang Bangun Smart System Safety Doorlock pada Kamar Crew Kapal Menggunakan Face Recognition dengan ESP32 Wrover Cam. Ocean Engineering: Jurnal Ilmu Teknik dan Teknologi Maritim **3**(3), 49–66 (2024)
15. Aishwarya, K., Nirmala, R., Navamathavan, R.: Recent advancements in liquefied petroleum gas sensors: A topical review. Sensors International **2**, 100091 (2021)
16. Easterline, L.M., Putri, A.A.Z.R., Atmaja, P.S., Dewi, A.L., Prasetyo, A.: Smart Air Monitoring with IoT-based MQ-2, MQ-7, MQ-8, and MQ-135 Sensors using NodeMCU ESP32. Procedia Computer Science **245**, 815–824 (2024)
17. Prananda, M.I., Syahputra, S., Syari, M.A.: Design of an LPG Leak Detection System Using IoT-Based MQ-2 Sensor. Journal of Artificial Intelligence and Engineering Applications, **3**(1), 383–387 (2023)
18. Saini, J., Dutta, M., Marques, G.: Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review. International journal of environmental research and public health, **17**(14), 4942 (2020)

19. Hsu, W.L., Jhuang, J.Y., Huang, C.S., Liang, C.K., Shiau, Y.C.: Application of Internet of Things in a Kitchen Fire Prevention System.Applied Sciences, **9**(17), 3520 (2019)
20. Murad, M., Bayat, O., Marhoon, H.M.: Design and Implementation of a Smart Home System with Two Levels of Security Based on IoT Technology. (2021)
21. Sarwar, B., Bajwa, I.S., Jamil, N., Ramzan, S., Sarwar, N.: An Intelligent Fire Warning Application Using IoT and an Adaptive Neuro-Fuzzy Inference System. Sensors **19**(14), 3150 (2019).
22. Tjandi, Y., Kasim, S.: Electric Control Equipment Based on Arduino Relay. In: Journal of Physics: Conference Series, **1244**(1), 012028 (2019).
23. Chang, Y.H., Wu, F.C., Lin, H.W.: Design and Implementation of ESP32-Based Edge Computing for Object Detection. Sensors **25**(6), 1656 (2025).
24. David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., et al.: TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems. In: Proceedings of Machine Learning and Systems, 3, 800–811 (2021).