InComTech: Jurnal Telekomunikasi dan Komputer
vol.12, no.3, Desember 2022, 240-253
http://publikasi.mercubuana.ac.id/index.php/Incomtech
P-ISSN: 2085-4811 E-ISSN: 2579-6089

Design and Implementation of Image Capture for Cluster Housing Security System Based on IoT

Muhamad Arief Aryadi, Mia Galina*

Electrical Engineering, Faculty of Engineering, President University
Jababeka Education Park, 17530, Indonesia
*Corresponding Email: *miagalina@president.ac.id*

Abstract:
The performance of IoT platforms for security systems has been implemented by some researchers in various scopes, such as doors, garages, and house gates. An IoT platform is implemented at the residential gate cluster for entering and exiting the gate. Having an interactive system, sending an image of the visitor to the resident, and operating an automatic gate are the three main features developed in this work. Interconnections between the Arduino board and MATLAB and the Arduino board and Blynk are used to perform these three features. This work describes the entire creation process, from the hardware requirements to the system design and simulation test results from the running process. From the simulation test, the device can interact with the incoming visitor within 1.33 seconds on average, with the accuracy of the played voice being 100% correct and the image sent to the 100% proper corresponding resident within the time taken to respond to permission granted: 1.56 seconds, while the permission denied takes 1.39 seconds.

Key Words:
Security system;
IoT;
Image capture;
Residential security

Article history:
Received May 24th, 2022
Revised October 22th, 2022
Accepted November 9th, 2022
Published December 26th, 2022

DOI:
10.22441/incomtech.v12i3.15514

1. INTRODUCTION

As a developing country, Indonesia faces many obstacles such as crime, corruption, drugs, etc. According to the Indonesian National Police Headquarters (MABES POLRI), the number of robbery victims has the highest crime rate, at 85.41% in 2016 and 84.47% in 2017 [1]. To lower specific rates, we should focus more on contributors with higher rates; in this case, robbery is the highest paying contributor.

The Internet of Things (IoT) is a new paradigm that allows electrical gadgets and sensors to connect through the Internet, making our lives easier. Many researchers are now working on the Internet of Things (IoT). Despite such a thorough definition, academics define the IoT in various ways. By introducing the radio frequency identification of internet connections in 2009, Ashton was the first to apply IoT nomenclature to improve supply chain operations [2]. IoT is defined by Sundmaeker et al. [3] as a technology that connects people and things to anyone, anywhere, preferably across any network, path, and service. The IoT is defined by Miorandi et al. [4] as a global network of smart things linked by modern Internet technologies. According to Xia et al. [5], the IoT is made up of things that can communicate with one another to gather, process, store, and transfer data over an Internet connection. Aryal et al. defined the Internet of Things (IoT) as an intelligent system that allows data to be gathered, monitored, and controlled over the Internet [6]. Finally, Internet of Things (IoT) is described as a network of items linked by sensors, electronics, software, and communication [7]. Figure 1 depicts the evolution of the Internet and how it led to the Internet of Things [2].

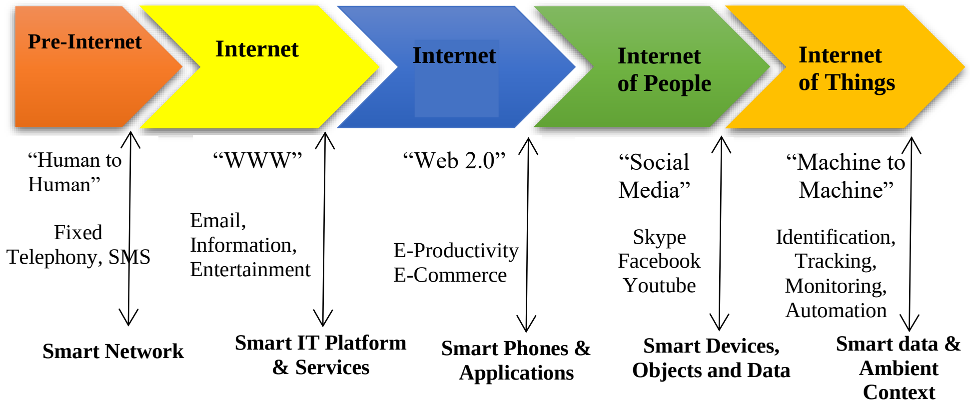


Figure 1. Evolution of the Internet from Pre-Internet to IoT [8]

Different researchers have different interpretations of specific interests and aspects of the IoT. IoT is an innovation that brings together various intelligent systems, frameworks, smart devices, and sensors, becoming an essential aspect of our lives. Many researchers have developed a security system to help in the elimination of robberies, such as by implementing a security system for smart homes using Global for Mobile Communications (GSM) technology [9], implementing a finger-based authentication system for unlocking doors [10, 11] or any security system by utilizing sensors, microcontrollers, and different mechanisms approaches to have the aimed results. By contrasting four types of research, the author identified the strengths and weaknesses of the existing automated gate/door technologies, as shown in Table 1, to improve the uniqueness and performance of the prototype. Researchers have different interpretations of specific interests and aspects of the IoT.

In terms of home security, we can integrate it with a security system in a single housing development for home security. Compared to a stand-alone home security system, this solution aims to provide better security guarantees. The goal behind this prototype was to allow the home's owners to approve visitors and monitor their visits starting from the first time they enter the town's gate. Additionally, this system can assist residential security in verifying who is allowed to enter tenants' houses, with direct confirmation from the residents of the house.

The main focus of this work is to use IoT technology to provide an unmanned security system solution for the cluster gate system. It aims to have a security system that can communicate interactively with the incoming visitor using mobile apps, capture and store the visitor's image, send the picture to the resident for authentication, and automatically operate the gate prototype.

Table 1. Related Project Comparison

Parameters	**Security Sys. With RFID** [12]	**Knocking Pattern Door** [13]	**Face Recognition Door** [14]	**Access Control and Security using smartphone** [15]	**Integrated Image Capture (this Work)**
Access	Physical	Physical	Physical	Bluetooth	Internet
Gate/Door controller	Automatic With solenoid	Automatic mechanic	Relay	Motor Driver and Servo	Motor Driver and DC Motor
Validation scheme	Reading by RFID receiver	Knocking pattern	Biometric face recognition	Signal given by mobile apps	Signal given by mobile apps
Time used	Fast	Slow	Fast	Faster	Faster
Area Implementation	Door	Door	Door	Home gate and garage door	Residence gate
Accessing Procedure	-	-	-	Mobile Apps	Email dan Mobile Apps
Notification	-	-	-	Buzzer	Email

2. RESEARCH METHOD

2.1 System Design

This section of the chapter describes the overall system design of the prototype. The whole system's processes are explained briefly by the block diagram and can be more fully detailed by the following flowchart of the corresponding methods. At a glance, the entire system is presented in Figure 2.

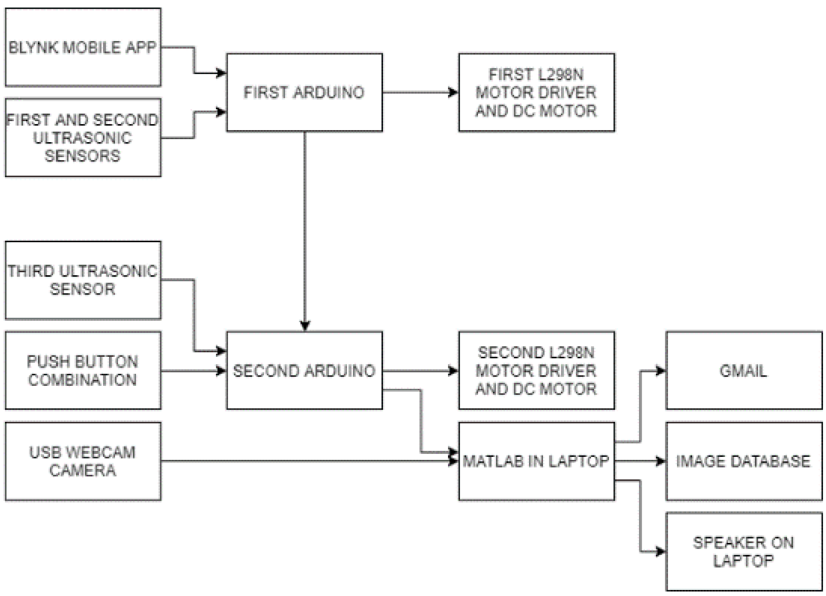


Figure 2. Block Diagram of the Cluster Security

Figure 2 shows a solid diagram of what the system consists of and which parts will be used as input and output for the system. Two Arduino microprocessors control the system. The classification of the process is divided into two characteristics: input and output. The inputs for the first Arduino are Blynk mobile apps and ultrasonic sensor reading. To complete the process, the first Arduino communicates with two ultrasonic sensors and gives the desired output to the DC motor on the entering gate controlled by the L298N motor driver. The first Arduino not only sends the signal to the DC motor, but it also sends a single-bit signal to the second Arduino to get the voice from MATLAB.

The second Arduino consists of input and output processes supported by several components besides the microcontroller. Push-button combination and ultrasonic sensor as the input process for the microcontroller to give the desired output through serial communication to MATLAB and the action of the DC motor on the exiting gate controlled by the L298N motor driver. MATLAB has inputs from Arduino and the security camera; the intended outputs are images captured by the security camera, saved in specified folders, and then sent to the corresponding email by the instructions from Arduino. The other important input read by the second Arduino is the signal sent by the first Arduino for a specific process. The system plot consists of two methods: the first is when a visitor (who can be resident or non-resident) comes to enter the cluster, and the second is when a visitor leaves the cluster. The corresponding processes are displayed in detail in the following flowcharts, as shown in Figure 3.

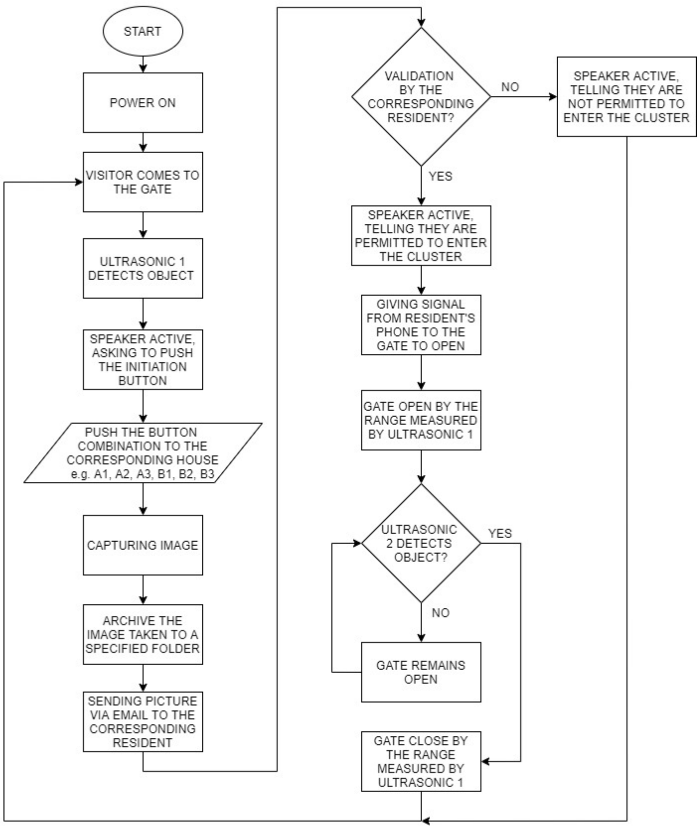


Figure 3. Flow Chart of the Entering Process

Figure 3 depicts the entry process in detail. Starting when the system is online, when ultrasonic sensor 1 placed on the front of the gate detects an object, the speaker activates right away to notify the visitor of what they should do, pushing the push button combination. After the combination of buttons is pushed, the Arduino sends a signal to MATLAB to take a picture and then sends it to the corresponding residents to ask for validation. If the confirmation is given, the gate begins to open to the extent that the ultrasonic sensor measured the range; if the validation is not given, the visitor is informed that they should not pass through the gate.After the gate is opened, the other ultrasonic sensor placed inside the cluster will activate the signal to Arduino to start closing the gate after the sensor does not detect any objects. Finally, when the ultrasonic does not detect any object, the gate will begin closing automatically and stop when the limit switch on the edge of the gate is hit.

The exiting process can be seen in the following flowchart, labeled as described in Figure 4. Unlike the entering process, the exiting process is very similar but less complicated than the entering process.This process occurs when the residents or visitors go outside the cluster. The first process is when the ultrasonic sensor detects an object and measures the range between the sensor and the vehicle. After that, the gate starts to open right away, just as the measurement from the ultrasonic sensor indicates. If the sensor detects an object, it will remain open for the next three seconds and do the same thing repeatedly until it does not detect any object. When it happens, the gate starts to close until the limit switch placed on the edge of the gate gets hit.

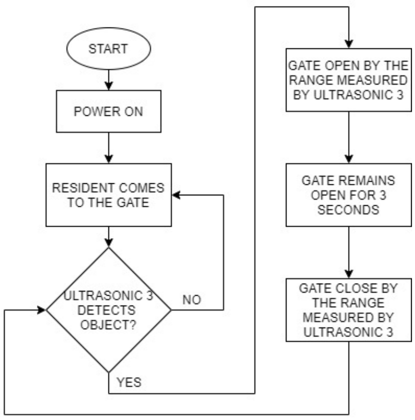


Figure 4. Flowchart of The Exiting Process

2.2 Hardware Implementation and the Wiring Diagram

This part of the chapter will be focused on how the cluster system is built. The presented pictures and wiring diagram will be explained in detail with the corresponding components. The overall components required to build the cluster security system are as follows:

- 2 Arduinos
- 1 Laptop as a server
- 3 HC-SR04 sensors
- 5 pushbuttons
- 2 DC motors
- 2 L298N motor driver
- 1 USB webcam

The Arduino UNO microcontroller used in this work serves as the brain in controlling the residence gate, sending notifications with an image inside to the residence via email, receiving data from the residence's mobile phone, and then providing output such as voice via the speaker provided on the front of the residence gate, as well as any activity done throughout the process.Push buttons in this final project have two different purposes. To begin, the five push buttons serve as an identifier for sending the signal to MATLAB, which then executes the sending images via email process. Those five push buttons are placed on the entry gate, which is accessible to the visitor. The ultrasonic sensor sends energy in the form of ultrasonic waves toward the target object and then reflects the waves back to the sensing head. In this work, the ultrasonic sensor detects an incoming visitor for signal input to the microcontroller.

The composition of the prototype consists of six residents' houses, three ultrasonic sensors (two on the entering gate and one on the exiting gate), two Arduinos as the controller (one for each gate), two L298N motor drivers, two DC motors (for each gate), and a green box with push buttons and a webcam camera inside. All the boxes are used as holders for the simulation phone. When the email is sent for one of the resident's emails, the notification can be seen clearly.

The complete wiring diagram consisted of two Arduinos, two L298N motor drivers, three ultrasonic sensors, eight pushbuttons, and two DC motors. The L298N motor driver and ultrasonic sensor are presented as Driver Bridge L298 and HC-SR04. All eight pushbuttons were labelled from 1 to 8 with one kΩ each. The DC motor is represented as M. The cluster security system can be implemented by running this wiring system model, as shown in Figure 5.

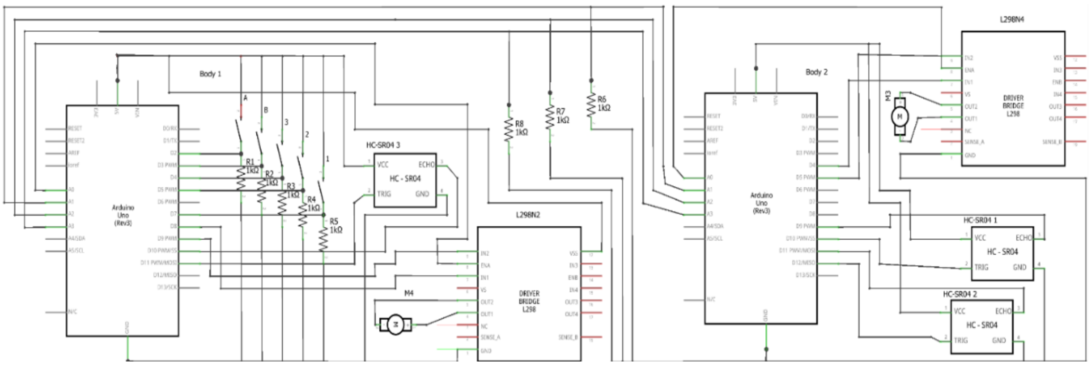


Figure 5. Complete wiring diagram of the system

2.3 Software Implementation

This research involved three software programs. The Arduino IDE and MATLAB are implemented in the prototype, while the Blynk mobile app is implemented on the residents' phones.

2.3.1 Arduino IDE

Arduino IDE (Integrated Development Environment) worked as a compiler for the Arduino microcontroller. Everything an Arduino microcontroller experiences is based on the code implemented by the Arduino IDE. There are two complete programs for the

Arduino IDE implementation: the first is for the entering gate process, and the second is for the exiting gate process. The first Arduino code-named entering primary gate function executes everything that is supposed to happen on the entering gate. The Arduino code is expected to handle two ultrasonic sensors, one DC motor, and the Blynk mobile app. The second Arduino dealt with the program to control MATLAB, ultrasonics, and a dc motor on the exit gate. This program code is also responsible for starting the email sending process by giving a specific value to MATLAB that is triggered by the push button combinations.

2.3.2 Blynk Mobile App

Blynk is an open-source platform that uses IOS and Android apps to control Arduino, Raspberry Pi, and other microcontrollers over the Internet. The mobile app is a digital dashboard that uses easy drag-and-drop widgets to create a graphic interface for a specific project. An authentication number is issued to each Blynk mobile app account used in the Arduino IDE code program. Blynk is used as the residents' interface to the system. As the only direct component for the residents, having a user-friendly and easy-to-use operation procedure is necessary. Blynk, in this work (Figure 6), is constructed with only two switches for two different expected outputs, allowing or disallowing the visitor to get into the cluster.

Figure 6. Blynk Mobile App for this prototype

2.3.3 MATLAB R2019a

The voice-playing process is programmed using MATLAB R2019a. Like Arduino IDE, MATLAB also has its own programming language called MATLAB. In MATLAB R2019a, changing languages is also possible. Two MATLAB codes have been built as shown in Figure 7. Inside the first code-named sending test, there are several functions: catching the signal sent by the second Arduino, playing saved voice as a response to the signal sent, and assigning the signal captured to one of the corresponding residents' emails. The second code is an extension code function that the first code can call, and this code is named the test email code. Its main parts are capturing images, creating the image database, and handling the email process, including email configuration and formatting.

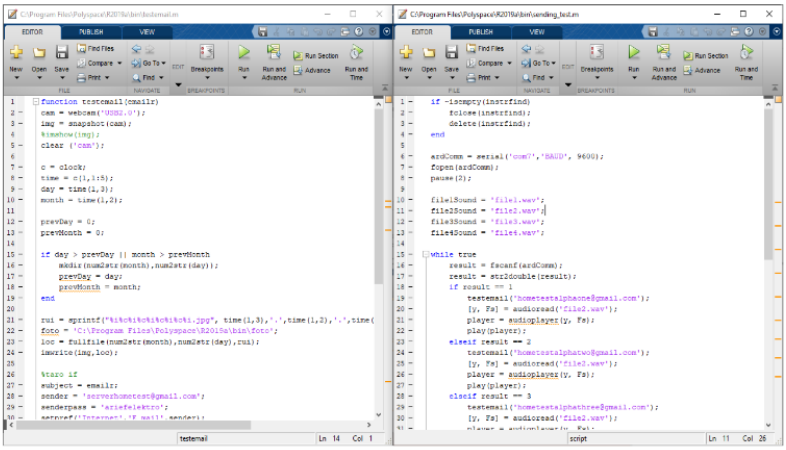


Figure 7. MATLAB Code Overview

In MATLAB, the saved voice files are File 1, File 2, File 3, and File 4. File 1 is a voice response when the visitor comes to the gate, telling the visitor what to do (press the pushbutton combination). File 2 is a voice response when the email is sent to the corresponding resident, while File 3 is a voice response when the visitor is permitted to enter the cluster, followed by the gate opening afterward. Finally, file 4 is a voice response when the visitor does not enter the cluster. For File 1, the input signal was sent by the first Arduino, triggered by the first ultrasonic sensor. The signal to play File 2 is obtained from the pushbutton combination connected to the second Arduino. After the combination button is pushed, this sends the test code, which selects the chosen resident's email and sends it to the test mail code. The last two Files, 3 and 4 signals, are sent from the first Arduino triggered by the mobile phones .

3. RESULT AND DISCUSSIONS

The prototype is designed to have a practical simulation of the system by covering six houses with six different corresponding emails as the residents' emails, as described in Figure 8. The systems are controlled by two Arduinos, which are the brains of the system.

Figure 8. The prototype of cluster housing security system based on IoT integrated with image capture function

The first Arduino controls the process of sending signals from the push buttons to residents' emails and is also responsible for the process happening in the exit gate, which includes its corresponding ultrasonic sensor and DC motor. The second Arduino is controlling the reading of the two ultrasonic sensors on the entering gate and its DC motor. Due to the demand for utilizing the voices generated by MATLAB, both Arduinos are connected with a simple connection to have practical communication. The experiments were done to complete the research objectives.

3.1 Entering Gate Test

The first ultrasonic connected to Arduino 1 is placed on the front of the entering gate. This ultrasonic has two functions to be tested. The first ultrasonic's function triggers the Arduino 1, instructing MATLAB to play an output voice from File 1. There are two indicators for this part of the test: the time taken by the ultrasonic sensor to read the object and the voice played. According to the flowchart of the system, the following process after File 1 is played is choosing a push-button combination and sending the captured image to the corresponding resident's email, as described in Figure 9.

Figure 9. An image while prototype testing for the entering gate test function

Data obtained from all the experiments for the first function on the first ultrasonic sensor gives a general result of good reading. The voices played are all in File 1. The tests for this part of the process were done, and the time taken for the system to respond is 1.33 seconds on average, as shown in Table 2.

Table 2. Entering Gate Initiation Test Result

Played Voice	Time Delayed (seconds)
File 1	1.41
File 1	0.86
File 1	1.06
File 1	1.11
File 1	1.17
File 1	1.74
File 1	1.28
File 1	1.84
File 1	1.24
File 1	1.54
Average	**1.33**

After File 1 is played, the following process involves choosing a push-button combination and sending the captured image to the corresponding resident's email. Tests for this part of the process were done, and the results can be seen in Figure 10.

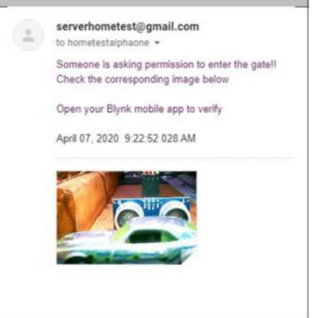
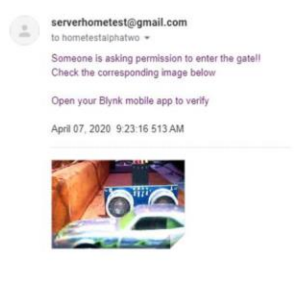
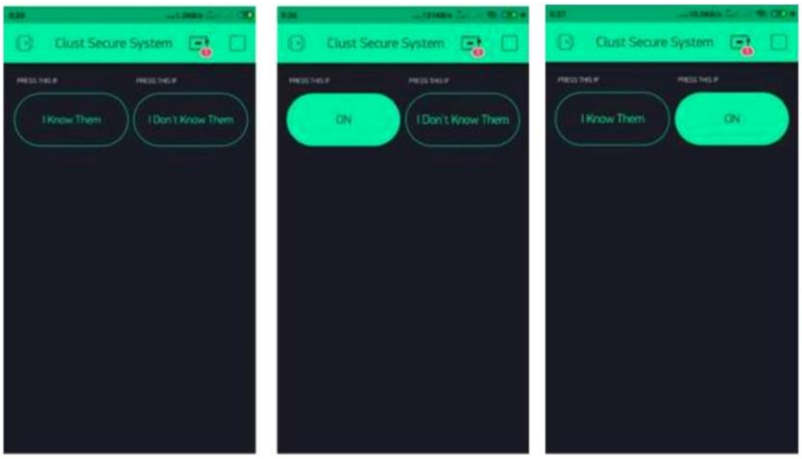
Button Pressed	Email sent	Email screen captured
A 1	hometestalphaone@gmail.com	
A 2	hometestalphatwo@gmail.com	

Figure 10. System Initiation Test Result

It shows that the tests were done successfully. All the button combinations were pressed, and the email was sent with the correct corresponding resident's email. The front seat of the car can be seen clearly in all six pictures taken by the webcam, and this is the information required for the residents to identify the visitor. File 2 will be played once the visitor pushes the push-button combination, indicating that the email was sent to the relevant resident.

The next step is the function, where residents come to play their roles in the process. The validation process started by receiving the email and checking whether they knew the visitor or not. They can respond immediately by opening the Blynk mobile apps on their mobile phones and pressing one of the two options provided on the apps, as shown in Figure 11.



(a) Online Condition (b) Responding when resident gives permission
(c) Responding when the resident does not give permission

Figure 11. Blynk UI for Resident

After the visitor presses one of the options provided as the response, the system will then play either File 3 or File 4 as the selected response. Table 3 shows the time taken from when the resident pressed the confirmation button until the system played either File 3 or File 4.

Table 3. The Confirmation System Test Result

Selected Respond	File Played	Time Delayed (Seconds)	Selected Respond	File Played	Time Delayed (Seconds)
Permitted	File 3	2.24	Not Permitted	File 4	0.70
Permitted	File 3	0.63	Not Permitted	File 4	1.83
Permitted	File 3	1.25	Not Permitted	File 4	1.00
Permitted	File 3	1.45	Not Permitted	File 4	2.18
Permitted	File 3	1.63	Not Permitted	File 4	0.97
Permitted	File 3	2.18	Not Permitted	File 4	1.63
Average		**1.56**		**Average**	**1.39**

The data in Table 3 provides a clear picture of the system's ability to read the same input and provide the appropriate responses. From 12 attempts, the system has successfully responded with 12 correct responses by playing the correct corresponding voice in either File 3 or File 4. The time taken for this part of the process was 1.47 seconds on average. Because this part of the process involves the use of an internet connection, the time required by the system is highly dependent on the speed of the internet connection. To summarize all the testing that had been done on the entering gate process, as presented in Table 4.

Table 4. Summary of The Entering Gate Test Result

Test Name	Testing condition	Parameters	Result	Additional information
Initiation Test	None	Time taken to respond	1.33 seconds in average	
		Voice played	100% success rate	10 times File 1 played on 10 tests
Sending Email Test	Using GSM	Button pressed function	100% success rate	All the button are working properly
		Voice played	100% success rate	6 times File 2 played on 6 tests
		Email sending accuracy	100% success rate	6 times sending to 6 different emails
Receiving signal from Blynk Test	Permission Granted	Time taken to respond	1.56 seconds in average	
		Voice played	100% success rate	5 times File 3 played on 5 tests
	Permission Denied	Time taken to respond	1.39 seconds in average	
		Voice played	100% success rate	5 times File 4 played on 5 tests

3.2 Existing Gate Test

This part of the process is simpler than opening the gate, which involves two components. First, the ultrasonic sensor was placed nearby the exiting gate, and second, the dc motor controlled the exiting gate activity, as described in Figure 12.

Figure 12. An image while prototype testing for the exiting gate test function

The tests were carried out by simulating the exit of a car from the cluster. The exit gate process is divided into three steps. First, the reading of the ultrasonic sensor near the gate changes, indicating an object is passing by. Second, the gate starts to open at the given range measured by the ultrasonic sensor and waits for three seconds. Third, the gate begins to close with the same range that it began to open with.

The final step is closing the exit gate after it has remained open for three seconds with the same range as when it was opened. When the car was going to get out, the cluster was on the edge of the road, meaning it was 0 cm away from the ultrasonic sensor. After the Arduino read the car's position, it asked the DC motor to start to open. The gate was open, not entirely as it should be, but only 1.5 cm of the gate was on the road. Table 5 recorded all the data from the tests.

Table 5. Summary of the existing gate test result

Car Position (cm)	Range of The gate Opened	Result
0	1.5	Gate opened
2.0	1.4	Gate opened
2.5	1.7	Gate opened
3.5	2.0	Gate opened
4.0	2.6	Gate opened
4.5	3.2	Gate opened
5	3.7	Gate opened

The measurement for this part of the process was the same as the opening process at the gate. The lack of precision of the gate model has a significant effect on this error measurement. The final step is closing the exit gate after it has remained open for three seconds with the same range as when it was opened.

4. CONCLUSION

The implementation of IoT-based integrated picture capture for cluster residential security systems has been successful. The result obtained for the first ultrasonic sensor function gives a general result of good reading with variation in time, and all voices were 100% successfully played, with an average of 1.33 seconds. All of the images taken by the USB Webcam camera were successfully saved in a specific folder. By combining the press button function appropriately, the sending process to each corresponding resident was completed with 100% accuracy. The system was successfully tested with time variation in terms of the response of a resident member to grant or deny permission to the visitor. The response time to permission granted is 1.56 seconds, while the response time to permission denied is 1.39 seconds. The voice used for both permission granted and denied has a 100% success rate. Several aspects can be improved in the future to achieve a better security system, such as using a higher resolution for the ultrasonic sensor to minimize error because the program heavily relies on the reading of the ultrasonic sensor for its input.

REFERENCES

[1] S. S. P. d. Keamanan, "Statistik Kriminal 2018," *Badan Pusat Statistik*, Jakarta, 2018.

[2] K. Lawal and H. N. Rafsanjani, "Trends, benefits, risks, and challenges of IoT implementation in residential and commercial buildings," *Energy and Built Environment*, vol. 3, no. 3, pp. 251-256, 2022, doi: 10.1016/j.enbenv.2021.01.009.

[3] H. Sundmaeker, P. Guillemin, P. Friess and S. Woelfflé, "Vision and challenges for realizing the internet of things, Brussels: Clust. Eur. Res. Proj.," *Internet Things Eur*, 2010, doi: 10.2759/26127.

[4] D. Miorandi, S. Sicari, F. De. Pellegrini, I. Chlamtac and S, "Internet of things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497-1516, 2012, doi: 10.1016/j.adhoc.2012.02.016.

[5] F. Xia, L. Yang, L. Wang and A. Vinel, "Internet of things," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1101-1102, 2012, doi: 10.1002/dac.2417.

[6] A. Aryal, F. Anselmo and B. B-Gerber, "Smart IoT desk for personalizing indoor environmental conditions," in *in Proceedings of the 8th International Conference on the Internet of Things (2019)*, Santa Barbara, CA, USA, 2018, doi: 10.1145/3277593.3277614.

[7] D. Mocrii, Y. Chen and P. Musilek, "IoT-based smart homes: a review of system architecture, software, communications, privacy and security," *Internet Things*, Vols. 1-2, pp. 81-98, 2018, doi: 10.1016/j.iot.2018.08.009.

[8] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture," *Computers and Electronics in Agriculture*, vol. 157, pp. 218-231, 2019, doi: 10.1016/j.compag.2018.12.039.

[9] K. Govinda, K. S. K. Prasad and S. R. Susheel, "Intrusion detection system for smart home using laser rays," *International Journal for Scientific Research & Development (IJSRD)*, vol. 2, no. 3, pp. 176-178, 2014.

[10] M. Sarma, A. Gogoi, R. Saikia and D. J. Bora, "Fingerprint Based Door Access System using Arduino," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 4, no. 8, pp. 1-5, 2020.

[11] D. Mahjabeen and M. R. Tarafder, "Unique Authentication for Door Lock system through Bio Scanning-Finger Print Security System," *Global Scientific Journals*, vol. 9, no. 10, pp. 1559-1565, 2021.

[12] A. Najib, R. Munadi and N. Kar, "Security system with RFID control using E-KTP and internet of things," *Bulletin of Electrical Engineering and Informatic*, vol. 10, no. 3, pp. 1436-1445, 2021, doi: 10.11591/eei.v10i3.2834.

[13] R. Reddy, P. V. Krishna, M. K. Chaitanya and M. Neeharika, "Security System Based on Knock-Pattern Using Arduino and GSM Communication," *International Journal of Engineering and Techniques*, vol. 4, no. 1, pp. 154-157, 2018.

[14] S. Yedulapuram, R. Arabelli, K. Mahender and C. Sidhardha, "Automatic Door Lock System by Face Recognition," *in International Conference on Recent Advancements in Engineering and Management (ICRAEM-2020)*, Warangal, India, 2020, doi: 10.1088/1757-899X/981/3/032036.

[15] H. Amiliansyah, M. Galina and J. W. Simatupang, "Access Control and Security System Via Bluetooth Application on Android Smartphone," *ELTIKOM*, vol. 6, no. 1, pp. 100-108, 2022, doi: 10.31961/eltikom.v6i1.475.