Smart Door Lock Innovation Using Integration of Bluetooth Low Energy and MQTT IoT Protocol

Ikmal Mughni Kurniasyah[1*], Lusia Rakhmawati[2]
[1,2]Electrical Engineering Department, Universitas Negeri Surabaya
[1]A5 Building Ketintang Campus, Surabaya 60231, Indonesia
[1]ikmalmughni.21002@mhs.unesa.ac.id
[2]lusiarakhmawati@unesa.ac.id

Abstract – This research develops a smart door lock system using ESP32 with Bluetooth Low Energy (BLE) and MQTT IoT protocol integration to ensure secure and efficient access control. The system supports RFID, fingerprint, and keypad authentication, with BLE utilized for device detection and proximity validation. Testing results show optimal performance, with MQTT QoS 0 achieving the highest throughput (5,052 kbps) and lowest delay (175 ms), while QoS 2 offers superior stability and minimal jitter (19 ms). RSSI analysis confirms the impact of distance and interference on BLE signal quality. The system reliably prevents unauthorized access and ensures real-time monitoring, making it suitable for modern security applications.

Keywords: Smart door lock, ESP32, Bluetooth Low Energy, MQTT IoT, RSSI

I. INTRODUCTION

The development of digital technology is growing more significant, especially in the Internet of Things (IoT). This development especially for the implementation in the smart home concept, which has a communication type called Machine to Machine (M2M).[1]

One of the parts in smart home concept is the security system for a house. especially in door security, which is commonly used the conventional key for locking the door.[2] By still using conventional door lock mechanism, a chance that the thief can be accessing a house. As perpetrators can forcibly open them rather than using the key owned by the homeowner.[3] this is reinforced by data from BPS on the thefts in East Java province from 2015 to 2017. The data said 14.213 cases, and 2.683 with violent thefts. Therefore, the new generation technology becomes a wise choice to consider.

For realizing implementation IoT in home security, several research aim to used ESP32 for smart home. Especially in monitoring and controlling lamp and plug. [4] second, the field of MQTT IoT gateway for bridging an interoperability from a device through internet.[5] However, the device must be connected to the internet for the system to function properly. third, the research about designed a presence detection system using the ESP32 for an automatic door lock system that utilities Bluetooth Low Energy (BLE) features.[6] However, this system is limited to hardware recognition, rather than sending data to an administrator who can monitor users accessing the house. On the other hand, the limited connection range is a disadvantage.

This study aims to overcome these limitations by designing a smart door lock system hat integrates BLE for user device recognition, MQTT IoT for monitoring and control by an administrator, and additional security sensors. The ESP32 microcontroller serves as the core component, leveraging its WiFi and BLE protocols and multiple GPIO pins for sensor connections. The solenoid door lock, an electromagnetic device powered by a 12-volt supply, acts as the locking mechanism, controlled by the ESP32 through a relay interface.[7] The system employs MQTT for efficient message transmission using a publish/subscribe model [8] and BLE for low-power communication within an extended range of up to 800 meters in version 5.0.[9]

With the continuous evolution of IoT in smart home systems, driven by the need for enhanced functionality, convenience, and security, integrating technologies like BLE and MQTT addresses the limitations of conventional locking mechanisms while offering a modern approach to security management. By combining real-time monitoring, multi-layered authentication, and remote accessibility, the development of smart door locks has the potential to redefine home security standards. In line with these advancements, this study seeks to develop a reliable and effective smart door lock system, ensuring enhanced security and ease in managing home door access while addressing the limitations of previous solutions.

II. METHODS

To design a smart door lock system on this research, it takes several stages to work on it. What steps are required can be seen in the following figure 1.

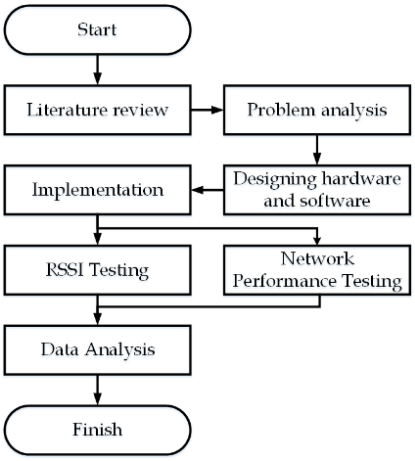

Figure 1. Research Stages

A. Hardware Design

The design of the system showed on picture:

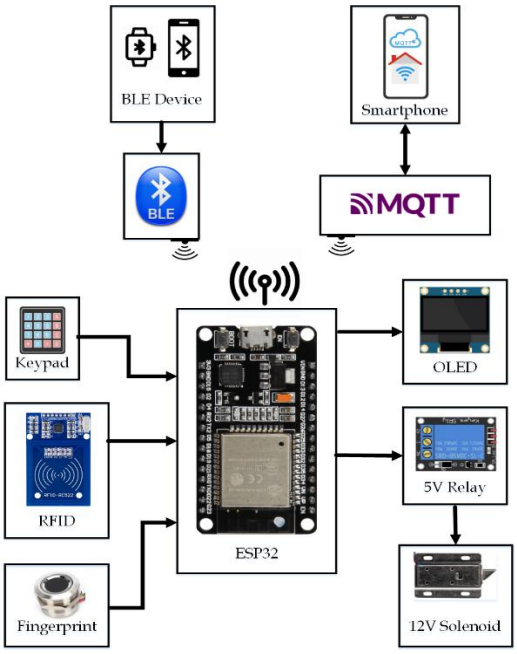

Figure 2. Hardware Design

Smart door lock system is built around the ESP32 microcontroller, which serves as the central processing unit for the entire system. It integrates multiple input devices, including a keypad, RFID module, fingerprint sensor, and Bluetooth Low Energy (BLE) module, providing diverse authentication methods to ensure robust security. These input devices capture user credentials and send the data to the ESP32 for processing and verification.

For output, the system includes an OLED display that provides visual feedback to the user, such as system status or error messages. A relay module acts as an electrical switch that controls the solenoid door lock, which physically secures or releases the door. The ESP32 is also equipped to communicate with a smartphone with a smartphone using the MQTT protocol, enabling remote monitoring and control of the system. This combination of components ensures a secure, versatile, and user-friendly smart lock solution.

B. Software Design

The design of software in this system showed on picture:

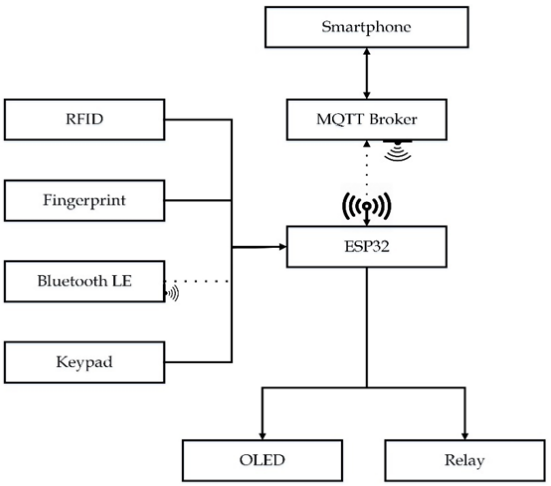

Figure 3. Software Design

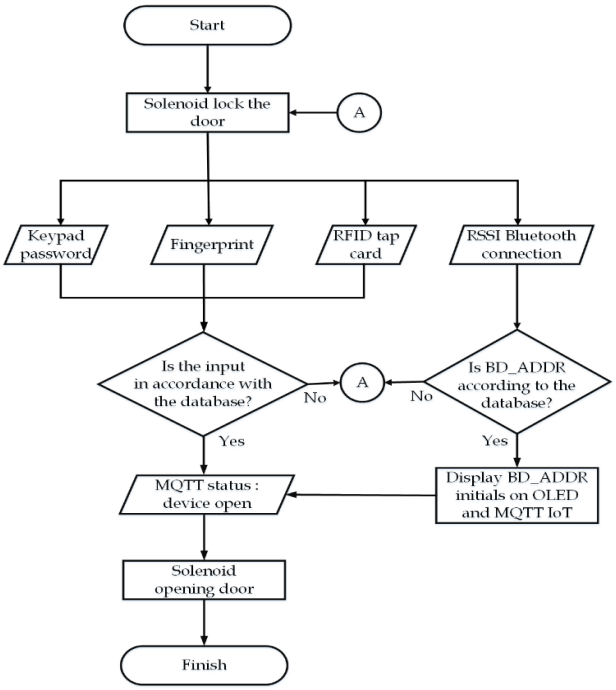

Figure 4. System design

Software is developed using the Arduino IDE, with a primary focus on efficient data processing and seamless device communication. The ESP32 microcontroller is programmed to validate user input-such as passwords, RFID tags, BLE, or fingerprints-against a predefined database stored in its memory. Upon successful authentication, the system triggers the relay to unlock the solenoid door lock and sends an

information to the administrator's smartphone.

In the case of invalid inputs, the system maintains the locked state of the door. Additionally, the development allowing the administrator to unlock the door and manage the system settings via an android. administrator communications between the ESP32 and the smartphone are conducted over the MQTT protocol. This design enables the smart door lock to deliver both convenience and high security for its users.

C. System Design

The system begins by connecting the ESP32 module to a WiFi network, allowing internet access. During this installation phase, the device sets up MQTT communication, BLE functionality, and security sensors. This includes configuring the necessary pinouts and libraries, along with entering the WiFi router's username and password. These steps prepare the system for real-time operation and communication.

Input data is gathered from BLE devices via their BD_ADDR or through security sensors such as a fingerprint scanner, RFID reader, or keypad. The ESP32 processes these input to verify their validity against the registered database. If the data matches, the solenoid relay triggers to unlock the door. MQTT IoT provides real-time updates, showing the door's lock/unlock status and, in the case of BLE, displaying the registered BD_ADDR initials. This ensures seamless and secure access controls.

D. Data Collection

This experimental procedure begins with hardware and software setup, including the installation of the ESP32 board, MQTT broker configuration, and sensor calibration. Inputs from BLE devices and security sensors are recorded in real-time. Specific steps include testing BLE BD_ADDR recognition, signal strength (RSSI), and different password combinations using fingerprint, RFID, and keypad inputs. Variations in distance, signal interference, and obstacles are introducted to test system reliability. Network performance metrics, such as RSSI values and packet data, are collected using a network protocol analyzer.

E. Data Analysis

Collected data is analyzed to evaluate system performance, focusing on key parameters like throughput, delay, jitter and packet loss for MQTT IoT communication. BLE security is tested by simulating attempts with both authorized and unathorized devices. The system's reliability is measured by calculating failure rates and determining whether the smart door lock functions as intended across varying conditions. This comprehensive analysis helps ensure the device meets user requirements for secure and efficient operations.

III. RESULT AND DISCUSSION

A. System Implementation

The research presents a device that functions as designed, accepting input from fingerprint, RFID, keypad, and Bluetooth Low Energy, and is controlled via an MQTT IoT Panel app. It measures 18cm x 11cm x 6cm, as shown in figure 5. The system was designed and tested, comprising a power supply, OLED display, fingerprint, RFID, keypad sensors, a microcontroller (ESP32), and a relay module. The power supply provided stablle voltages to all components, except for a slight undervoltage to the RFID module. The OLED display presented real-time system status. Sensors functioned as expected, with data processed by the ESP32. The system successfully connected to an MQTT broker, enabling administrator via a MQTT IoT Panel app. BLE functionality wast tested, and the relay module controlled the solenoid for door operation. Overall, the system demonstrated reliable performance.

Figure 5. Smart door lock Device

Table 1. RSSI without obstruction interference

Main device distance (m)	Signal interference	Interfering device distance (m)	RSSI (dBm)
1	no co-channel interference	-	-59.63
	co-channel interference	0	-60.75
3	no co-channel interference	-	-68.36
	co-channel interference	0	-69.50
		1	-69.12
		2	-68.85
7	no co-channel interference	-	-71.79
	co-channel interference	0	-73.40
		1	-73.25
		2	-73.00
		3	-72.70
		4	-72.35
		5	-72.40
		6	-72.10

B. Device Performance Analysis

Testing and analysis of the smart door lock performace with Bluetooth Low Energy and MQTT IoT protocol integration will be conducted. The performance matrix to be tested and analyzed includes two factors: RSSI BLE value and MQTT IoT throughput.

Table 2. RSSI with door obstruction

Main device distance (m)	Signal interference	Interfering device distance (m)	RSSI (dBm)
1	no co-channel interference	-	-63.01
	co-channel interference	0	-63.63
2	no co-channel interference	-	-68.02
	co-channel interference	0	-70.97
		1	-68.35
3	no co-channel interference	-	-74.1
	co-channel interference	0	-74.14
		1	-71.92
		2	-71.57

Table 3. RSSI with wall obstruction

Main device distance (m)	Signal interference	Interfering device distance (m)	RSSI (dBm)
2	no co-channel interference	-	-79.5
	co-channel interference	0	-79.89
		1	-79.72
3	no co-channel interference	-	-83.05
	co-channel interference	0	-83.91
		1	-83.63
		2	-83.28
4	no co-channel interference	-	-85.42
	co-channel interference	0	-86.43
		1	-86.02
		2	-85.85
		3	-85.61

1. RSSI BLE analysis

RSSI is measured based on distance, obstruction (e.g., wall or door), and co-channel interference. Data is collected 100 times and averaged. Results without obstruction interference are in table 1.

Testing with door obstruction interference is shown in table 2. Testing with wall obstruction is shown in table 3. The data indicates differences in RSSI results based on distance, co-channel interference, and obstruction interference. A two-way ANOVA test was conducted to analyze the effects of the independent variables like device distance, co-channel interference, and obstruction interference on RSSI. Data involved 100 samples under various conditions to calculate average RSSI values.

Hypoteses :

- H_0 (Null) : No significant differences in RSSI based on distance, co-channel interference, or obstruction interference
- H_1 (Alternative): Significant differences in RSSI exist based on one or more independent variables.

ANOVA results:

- Distance: p-value < 0.001, indicating a significant impact on RSSI.
- Obstruction interference: p-value < 0.001, showing significant effects of physical obstructions (none, door, wall)
- Co-channel interference: p-value = 0.034, indicating a smaller but significant effect on RSSI.

The ANOVA results confirm that distance, obstruction interference, and co-channel interference significantly influence RSSI. Controlling these variables is essential for optimal performance in Bluetooth Low Energy-based devices.

2. QoS analysis

The MQTT IoT network was analyzed and tested by measuring throughput, delay, jitter, and packet loss as the device's response using the MQTT IoT protocol with Wireshark software. Testing was conducted with three different QoS levels: 0, 1, and 2. The throughput result shown in table 4.

Table 4. Throughput test result

No.	Seconds	Throughput (kbps)		
		QoS 0	QoS 1	Qos 2
1.	1	5,082	3,373	2,298
2.	2	5,038	3,712	2,783
3.	3	4,892	3,604	2,668
4.	4	5,084	3,445	2,477
5.	5	5,088	3,742	2,697
6.	6	5,084	3,375	2,564
7.	7	5,094	3,635	2,473
Rata-rata		5,052	3,555	2,566

Throughput was highest at QoS 0 (5.052 kbps), followed by QoS 1 (3.555 kbps), and QoS 2 (2.566 kbps), indicating that higher QoS levels require more overhead, reducing data transfer efficiency. After throughput testing, delay testing

shown in table 5. QoS 0 showed the lowest delay (175 ms), while QoS 2

Table 5. Delay test result

No.	Seconds	Delay (ms)		
		QoS 0	QoS 1	QoS 2
1.	1	174	281	435
2.	2	172	282	431
3.	3	181	272	421
4.	4	176	267	418
5.	5	176	275	542
6.	6	175	262	421
7.	7	174	271	437
Rata-rata		175	273	444

Table 6. Jitter test result

No.	Seconds	Jitter (ms)		
		QoS 0	QoS 1	QoS 2
1.	1	24	32	34
2.	2	18	21	27
3.	3	22	30	18
4.	4	29	32	17
5.	5	25	47	3
6.	6	19	46	25
7.	7	29	39	8
Rata-rata		24	35	19

QoS 2 had the lowest jitter (19 ms), while QoS 1 had the highest (35 ms), with QoS 0 at 24 ms. QoS 2 is more stable in transmission time.

Table 7. Packet loss test result

No.	Seconds	Packet loss (%)		
		QoS 0	QoS 1	QoS 2
1.	1	0,00	0,00	0,00
2.	2	0,00	0,00	0,00
3.	3	0,00	0,00	0,00
4.	4	0,00	0,00	0,00
5.	5	0,00	0,00	0,00
6.	6	0,00	0,00	0,00
7.	7	0,00	0,00	0,00
Rata-rata		0,00	0,00	0,00

With any QoS services, no packet loss occurred (0%), ensuring stable data transmission.

C. Comparison of RSSI Readings Between Device and BLE Scanner

The comparison of RSSI readings from the device and the BLE Scanner application was conducted to evaluate the accuracy and quality of RSSI measurement from the device. Seven samples of RSSI readings were collected for both the device and the BLE Scanner application under conditions free of physical obstructions and signal interference. The error and percentage error were calculated using the equations (1) and (2). The comparison data is summarized in table 7.

$$Error = X_{prototype} - X_{Scanner} \quad (1)$$

$$Error = \frac{Error}{X_{BLE\ Scanner}} \times 100\% \quad (2)$$

To assess the difference in RSSI quality, a paired t-test was performed. The results showed that the RSSI readings from the device closely matched those of the BLE Scanner application, with an average error of 1.68 dBm and a standart deviation of 0.92 dBm, indicating minimal variation.

Table 8. RSSI comparison both device and BLE Scanner

Distance (m)	RSSI (Device) (dBm)	RSSI (BLE Scanner) (dBm)	Error (dB)	Error (%)
0	-32.47	-33.02	0.55	1.69%
1	-50.58	-51.34	0.76	1.50%
2	-61.93	-60.81	1.12	1.81%
3	-66.41	-65.03	1.38	2.08%
4	-70.22	-68.88	1.34	1.91%
5	-72.67	-70.96	1.71	2.35%
6	-73.82	-72.00	1.82	2.46%
7	-74.03	-71.70	2.33	3.15%

The t-test produced a t-value of -2.66 with 7 degrees of freedom and a p-value of 0.0324. Since the p-value is below 0.05, the difference in RSSI readings is statistically significant. However, the small error values suggest that the device has good accuracy and can reliably measure RSSI under the tested conditions.

D. Sensor Testing

The system was tested with RFID, fingerprint, and keypad sensors to evaluate their reliability in unlocking the door. Each sensor was tested five times, with three correct inputs followed by two incorrect inputs in table 9, 10, and 11.

Table 9. RFID Testing: The RFID sensor showed successful door unlocking with correct inputs

No	Keyword	Input	Door
1.	51,237,238 ,222	51 237 238 222	Open
2.		102 240 253 135	Closed
3.		51 237 238 222	Open
4.		102 17 253 135	Closed
5.		51 237 238 222	Open

Table 10. Fingerprint testing: The fingerprint sensor matched ID 1 for successful unlocking.

No.	Keyword	Input	Door
1.	ID : 1	Found match with ID : 1	Open
2.		Not found match	Closed
3.		Found match with ID : 1	Open
4.		Not found match	Closed
5.		Found match with ID : 1	Open

Table 11. Keypad testing: the keypad accepted the correct code and unlocked the door.

No	Keyword	Input	Door
1.	232323	232323	Open
2.		123456	Closed
3.		232323	Open
4.		654794	Closed
5.		232323	Open

Out of 15 trials, 9 successful correct inputs triggered the relay and solenoid, while 6 incorrect inputs kept the relay inactive. These results demonstrate the sensors' reliability and suitability for use as additional security layers in the smart door lock system

IV. CONCLUSION

Based on the research and testing that has been carried out, the following conclusions are obtained:

a. The smart door lock system was successfully implemented using ESP32, integrating Bluetooth Low Energy (BLE) for user detection and MQTT IoT for real-time monitoring and control.

b. RSSI signal performance is primarily influenced by distance and obstruction interference, with minimal effects from co-channel interference and signal variations.

c. MQTT QoS testing showed adequate network performance, with average throughput of 5.032 kbps (QoS 0), 3.555 kbps (QoS 1), and 2.566 kbps (QoS 2). Delay values were 175 ms (QoS 0), 2733 ms (QoS 1), and 444 ms (QoS 2), jitter values were below 25 ms, and packet loss was consistently under 1%.

d. Authentication testing using RFID, fingerprint, and keypad showed high accuracy and reliability, effectively opening the door for correct inputs.

Overall, the smart door lock system provides secure, reliable access control and is suitable for real world implementation

ACKNOWLEDGMENT

Thank you to Dr. Ir. Lusia Rakhmawati, S.T., M.T., as the supervisor who has provided valuable guidance, support, and constructive feedback throughput this research

REFERENCES

[1] A. R. Nimodiya and S. S. Ajankar, "A review on internet of things," *Int. J. Adv. Res. Sci. Commun. Technol., vol. 113, no. 1, pp. 135-144, 2022, doi: 10.48175/ijarsct-2251*.

[2] M. R. Asad, O. D. Nurhayati, and E. D. Widianto, "Sistem pengamanan pintu rumah otomatis via SMS berbasis mikrokontroller ATMega328P," J. Teknol. Sist. Komput., vol. 3, no. 1, pp. 1-7, 2015, doi : 10.14710/jtsiskom.3.1.2015.1-7.

[3] Septryanti and Fitriyanti, "Rancang bangun aplikasi kunci pintu otomatis berbasis mikrokontroller Arduino menggunakan smartphone Android," CESS J. Comput. Eng. Syst. Sci., vol. 2, no. 2, pp. 59, Jul. 2017.

[4] A. Restu Mukti, C. Mukmin, E. Randa Kasih, D. Palembang, S. I. Ulu, and S. Selatan, "Perancangan smart home menggunakan konsep Internet of Things (IoT) berbasis microcontroller," JUPITER J., vol. 14, no. 2, pp. 5166-522, 2022.

[5] B. M. Susanto, E. S. J. Atmadji, and W. L. Brenkman, "Implementasi MQTT protocol pada smart home security berbasis web," J. Informatika Polinema, vol. 4, no. 3, p. 201, 2018, doi: 10.33795/jip.v4i3.207.

[6] A. Prafanto, E. Budiman, P. P. Widagdo, G. M. Putra, and R. Wardhana, "Pendeteksi kehadiran menggunakan ESP32 untuk sistem pengunci pintu otomatis," J. Teknol. Terapan, vol. 7, no. 1, pp. 37-43, 2022.

[7] A. Jufri, "Rancang bangun dan implementasi kunci pintu elektronik menggunakan Arduino dan Android," *STT STIKMA Int*., vol. 7, no. 1, pp. 40-51, 2018.

[8] R. J. Cohn and R. J. Coppen, "MQTT Version 3.1.1 becomes an OASIS Standard," OASIS OPEN, Oct. 30, 2014.

[9] Bluetooth SIG, "Bluetooth specification version 4.0," 2010. [Online]. Available: https://www.bluetooth.com/specifications/specs/core-specification-4-0/