LM.UiTM.FKE(O).08/03

SCHOOL OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING
UNIVERSITI TEKNOLOGI MARA

FINAL YEAR PROJECT 2

SUBMISSION OF PROJECT TECHNICAL PAPER

STUDENT'S NAME	:	**SITI SARAH BINTI AZHAN**
STUDENT'S UiTM ID	:	**2021172537**
COURSE CODE	:	**EE242**
PROJECT TITLE		
DEVELOPMENT OF IOT CONTROLLER FOR SMART HOME		
This is to certify that the above student has submitted the project technical paper to the project supervisor (SV).		
SV's NAME	:	**MOHD SAUFY ROHMAD**
SV's SIGNATURE	:	
DATE OF SUBMISSION	:	**22ND JANUARY 2024**

1) ***Student needs to <u>fill in and submit this form together with the technical paper</u> to his/her SV for endorsement/approval.***
2) ***Student needs to return <u>this endorsed/approved form attached together with the technical paper</u> to the FYP2 Coordinator.***

Document Information

Analyzed document	draft_tp_sarah_latest.pdf (D184296495)
Submitted	2024-01-22 10:26:00 UTC+01:00
Submitted by	
Submitter email	2021172537@student.uitm.edu.my
Similarity	6%
Analysis address	saufy.UiTM@analysis.ouriginal.com

Sources included in the report

SA	UNIVERSITI TEKNOLOGI MARA (UiTM) / Soil Monitoring system using LoRa Technical Paper.pdf Document Soil Monitoring system using LoRa Technical Paper.pdf (D126956403) Submitted by: syed_farid@uitm.edu.my Receiver: syed_farid.UiTM@analysis.ouriginal.com	1
SA	UNIVERSITI TEKNOLOGI MARA (UiTM) / PROPOSAL_2021113573_CEEE241C7A_JULY2023.docx Document PROPOSAL_2021113573_CEEE241C7A_JULY2023.docx (D172420426) Submitted by: 2021113573@student.uitm.edu.my Receiver: saufy.UiTM@analysis.ouriginal.com	2
SA	Project_LIT_Review.pdf Document Project_LIT_Review.pdf (D179977311)	1
SA	Report.pdf Document Report.pdf (D139380881)	1
SA	research hrithik word.docx Document research hrithik word.docx (D171811862)	1

Entire Document

Development of IoT Controller for Smart Home Siti Sarah binti Azhan

85%	MATCHING BLOCK 1/6	SA	Soil Monitoring system using LoRa Technical Pa ... (D126956403)

School of Electrical Engineering College of Engineering University Teknologi MARA (UiTM) Shah Alam, Selangor, Malaysia 2021172537@student.uitm.edu.my Abstract—

Development of IoT Controller for Smart Home

Siti Sarah binti Azhan
School of Electrical Engineering
College of Engineering
University Teknologi MARA (UiTM)
Shah Alam, Selangor, Malaysia
2021172537@student.uitm.edu.my

***Abstract*—This paper focuses on Smart Home IoT Controller for sensor and actuator management. Three components make up the controller: ESP1 as a sensor, ESP2 central BLE processor, and ESP3 MQTT interface. The controller uses Bluetooth Low Energy (BLE), Hardware Serial connection, and MQTT integration. Sensors use BLE to transmit status changes to ESP2, which uses Hardware Serial to send data to ESP3. In order to provide external interfaces for remote device monitoring and control, ESP3 communicates with a MQTT broker. This smart home IoT controller's main goal is to provide a flexible, simple, user-friendly method of controlling sensors and appliances in a smart home setting. While Hardware Serial enables effective data interchange inside the controller, BLE guarantees dependable connectivity between sensors and the central controller. The increasing demand for responsive and centralised smart home ecosystems is met by the inclusion of MQTT. This paper delves further into the system architecture, communication protocols, and the critical role that MQTT plays in fostering a cohesive and adaptable smart home ecosystem. This research adds significantly to the continuing discussion on safe, effective, and useful solutions in the field of Smart Home IoT, in response to the increasing need for user-cantered, seamless connectivity and control.**

***Keywords*—ESP32; MQTT; BLE; Hardware Serial; controller**

I. INTRODUCTION

The use of Smart Home technology has transformed traditional homes into complex, networked ecosystems in the context of contemporary life. In order to maximise sensor management and control, this paper presents a cutting-edge Smart Home IoT Controller. The fundamental innovation consists of the smooth coordination of Hardware Serial communication, Bluetooth Low Energy (BLE), and MQTT, resulting in a unified system consisting of three essential parts: ESP1 (sensor nodes), ESP2 (central BLE processor), and ESP3 (MQTT interface).

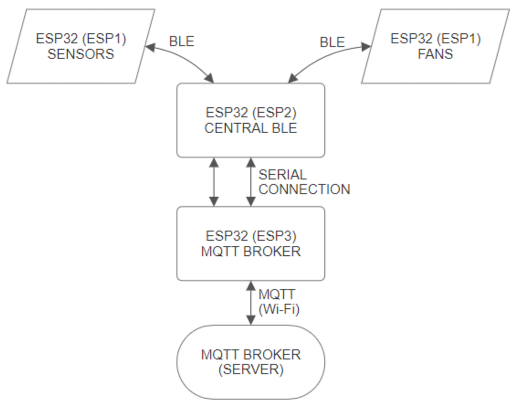

Fig. 1. Block diagram of the controller.

This controller's core component is Bluetooth Low Energy (BLE). The energy-efficient BLE technology plays a crucial role in enabling direct connection between sensors and the controller. It is essential to recognise, nonetheless, that bigger smart home setups may present issues due to BLE's intrinsically restricted range.[1] Therefore, a thorough solution is looked for to get over this range restriction and guarantee smooth communication across the living area.

In terms of improving the smart home, this includes components like fans, temperature sensors, motion detection sensors, and other networked sensors. The MQTT (Message Queuing Telemetry Transport) communication protocol has been selected in this particular situation. MQTT is a popular protocol that was created for automation and is distinguished by how well it transmits data.[2] It uses far less bandwidth and has a small footprint, ensuring smooth data flow. This feature is especially important for MQTT-enabled mobile applications since it extends battery life. As a result, in this particular scenario, MQTT has been deliberately chosen as the protocol foundation for the communication system.

Since MQTT makes use of the publish-subscribe (pub/sub) architecture, MQTT itself is lightweight, in other word, flexible.[3] To elaborate, MQTT's effective use of the publish-subscribe (pub/sub) architecture makes it especially useful for controlling smart home equipment. Within the previously discussed framework of the Smart Home IoT Controller, ESP3, which stands for the MQTT interface, plays a crucial function. Through the MQTT broker, ESP3 creates a dynamic channel to receive commands or messages by subscribing to particular topics. These messages provide crucial control information that determines how sensors and appliances operate, so making it easier to issue commands to switch them on or off in response to received instructions. Using the MQTT protocol, this sophisticated solution enables quick and easy control over the smart home environment.

Due to its low power consumption, short-range communication designed for the spatial dynamics of houses, and effective data transmission capabilities, Bluetooth Low Energy (BLE) stands out as the perfect communication technology for Smart Home applications. Bluetooth 5.0's integration of BLE mesh networking improves coverage and dependability, which is especially helpful for multi-device smart homes.[4] As BLE is widely supported in contemporary consumer electronics, it facilitates smooth communication between devices made by different manufacturers and adds to user friendliness. BLE's attractiveness is further reinforced by its affordable implementation, which makes it an appealing and comprehensive option for wireless communication in smart home settings.

For smart homes, integrating BLE with MQTT offers a very beneficial solution. BLE is an effective protocol for device interaction in a home setting because of its short-range communication, low power consumption, and mesh networking characteristics.[4] Nonetheless, the intrinsic range constraint [1] of BLE may be overcome by utilising MQTT, a resilient message system. Via ESP3, the MQTT interface, which subscribes to certain topics conveying instructions for sensor and appliance conditions, MQTT's publish-subscribe architecture enables smooth control of smart home appliance conditions. By providing a balance between MQTT's adaptability for wider control and BLE's efficiency for local device connection, this integration increases the system's overall flexibility and guarantees a safe and responsive smart home environment.

II. LITERATURE REVIEW

One of the most common issues in the field of smart home technology that has been brought up in the literature is the short range of Bluetooth Low Energy (BLE) connections, particularly for ESP32 devices. This range restriction is highlighted in the paper under consideration, which highlights the necessity for a workable method to get beyond the intrinsic communication distance restrictions of BLE-enabled ESP32 devices in the context of smart home applications.[1] Acknowledging this restriction as a possible barrier to seamless connectivity in smart home ecosystems, it becomes imperative to investigate and execute effective remedies. As a result, the literature recommends looking into potential fixes or implementing substitute technologies that can successfully increase the ESP32 devices' communication range in smart home settings.

When it comes to smart home technologies, the MQTT (Message Queuing Telemetry Transport) protocol—especially when used in conjunction with the Mosquitto broker—becomes essential for resolving communication issues. The research emphasises how important MQTT Mosquitto is for facilitating dependable and effective communication among networks of smart homes. With its lightweight design and publish-subscribe methodology, MQTT provides a useful way for devices in a smart home network to communicate with one another.[2] As a centralised hub, the Mosquitto broker makes it easier for different smart devices to communicate with one other. This architecture helps to improve energy efficiency, which is important for Internet of Things (IoT) applications, in addition to making smart home networks more flexible and scalable.

The study emphasises MQTT Mosquitto's value in streamlining smart home communication protocols and demonstrates how it may enhance and supplement the functionalities of ESP32 devices with BLE support. Therefore, integrating MQTT Mosquitto into smart home configurations offers a viable way to solve communication issues and improve the general operation of ESP32 devices in smart home settings.

III. METHODOLOGY

The development of a versatile smart home system is the primary goal of this project. In light of this, it is intended to be simple to add or remove any sensors as needed.

The overall block diagram for the controller is demonstrated in Fig. 1. The main part of this project would be the Central BLE and MQTT

Broker. Before that, the communications between each ESP32 will be explained in more detail.

The sensors are set to be in deep sleep mode when it first turned on (connected to power). In order to reduce power consumption, sensors go into deep sleep mode, which involves turning off unnecessary functions. In order to preserve battery life, the sensor lowers power consumption while deep sleep is occurring, momentarily stopping data collection and communication. The sensor resumes regular function after emerging from deep sleep, which enables it to balance energy efficiency.

The sensors only begin collecting data when they receive an indication from the MQTT Broker. One of the most important ways to find these indicators in ESP3 is through a subscription to essential topics. When ESP3 receives a message on one of the subscribed topics, it carefully examines the content to determine if it is an indication or some other type of communication. ESP3 uses an effective hardware serial connection to forward the indicator message to ESP2, the Central BLE processor, if it is an indicator. This methodology minimized energy consumption and overall system efficiency by guaranteeing that the sensors stay dormant until triggered by important indicators.

As the central BLE, ESP2 assumes a crucial role as the communication hub connecting sensors and the MQTT Broker. Its importance is found in the way it processes the information obtained from ESP3, an indicator used to control appliances (fans) or sensors. After processing, ESP2's duty is to notify the client using the data that has been processed. In addition, it serves as a channel through which information from the client is efficiently received and forwarded to the MQTT Broker. This dual functionality highlights ESP2's function as a central intelligence that synchronises data flow between clients, sensors, and the MQTT Broker for efficient control of smart homes. It also ensures smooth communication within the system.

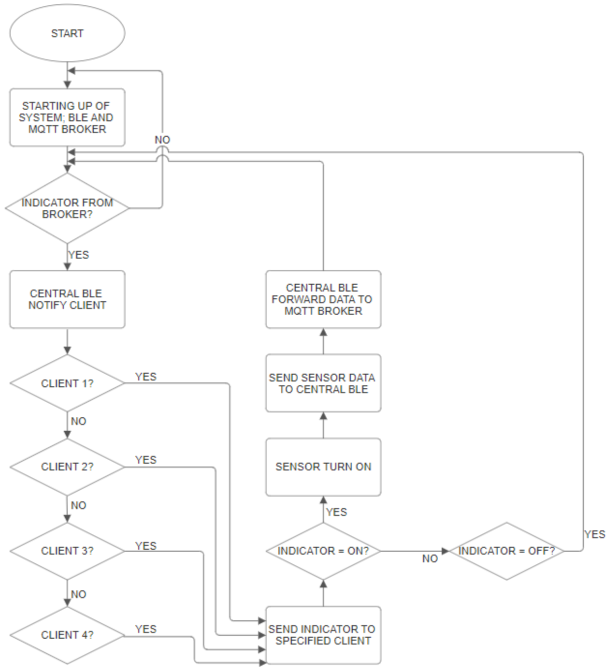

Fig. 2. Flowchart for overall system of controller.

Fig. 2. Shows the flowchart for smart home system controller. As the block diagram clarifies, for this system to operate, all necessary components must be connected, whether via hardware serial, BLE, or MQTT Broker. After these connections are made, ESP3 enters a waiting state until an indicator is published. As soon as this indicator is released, ESP3 starts sending the relevant information, therefore the cycle never ends. The system functions in a permanent mode, continually awaiting the publication of signs to assist actions such as turning sensors on or off. The dynamic and responsive nature of the smart home control system is shown by this ongoing activity.

A. *Hardware.*

1. ESP32.

Fig. 3. ESP32.

This figure shows the main microcontroller used for this system. The ESP32's capabilities and versatility make it a great choice for smart home systems. Due to its compatibility for MQTT and Bluetooth Low Energy (BLE), it is a major player in the communication of smart home devices. BLE guarantees low-power, effective communication between devices, making it perfect for sensors that run on batteries. Furthermore, the ESP32 can easily connect and communicate with the larger smart home ecosystem thanks to the incorporation of MQTT, which makes reliable data transmission and control possible. The ESP32's ability to handle both BLE and MQTT helps to create smart home systems that are interoperable, responsive, and energy-efficient.

B. Software.

1. Arduino IDE.

Fig. 4. Arduino IDE.

The Integrated Development Environment, or Arduino IDE, is a user-friendly platform that may be used to programme ESP32, Arduino microcontrollers, and many other devices. The Arduino IDE, which is well-known for its simplicity and usability, offers a clear interface for creating, assembling, and uploading code to microcontroller boards. Its intuitive editor and built-in libraries make coding easier for novices while maintaining versatility for more experienced users. The IDE is adaptable for a variety of projects because it supports a large selection of boards and shields. The Arduino IDE is a well-liked option for creating embedded systems and Internet of Things (IoT) applications by both experts and enthusiasts because to its robustness and effectiveness despite its simplicity.

2. MQTT Mosquitto.

Fig. 4. MQTT Mosquitto.

As an open-source MQTT broker, MQTT Mosquitto is an essential part of the smart home system's communication network. Its MQTT compliance and lightweight architecture make it a great option for enabling dependable and effective device-to-device messaging. Since Mosquitto is open-source and simple, it makes using it easy for developers to integrate and modify the broker to suit the unique requirements of their smart home applications. As it supports the publish-subscribe format, it improves communication efficiency, which is important for reducing energy consumption in smart home devices with limited resources.

IV. RESULT AND DISCUSSION.

A. MQTT and BLE Integration.

```
Invalid Indicator. Ignoring.
Connected to client
Received Value from device c0:49:ef:f1:43:ca: Connected to Receiver
Received Value from device c0:49:ef:f1:43:ca: Connected to Receiver
```

Fig. 5. Central BLE connected to client.

The illustration in Fig. 5 shows how the Central BLE (ESP2) connects to a client (ESP1). The Central BLE scans for clients that are available for connection as soon as the system is turned on. The Serial Monitor presents a display of important data as soon as the Central BLE connects to a client, as illustrated in Fig. 5. This data consists of the client's MAC address, which is used to confirm the connection, and the transmission of any messages that the connected client sends. This procedure improves the overall comprehension and interaction inside the system by verifying the successful connection and enabling the monitoring of incoming messages from the client.

```
Connecting to Hann's iPhone
E (3407) wifi:Association refused temporarily, comeback time 200 mSec
E (3612) wifi:Association refused temporarily, comeback time 200 mSec
E (3837) wifi:Association refused temporarily, comeback time 200 mSec
E (4043) wifi:Association refused temporarily, comeback time 200 mSec
E (3452) wifi:Association refused temporarily, comeback time 200 mSec
E (3658) wifi:Association refused temporarily, comeback time 200 mSec
E (3863) wifi:Association refused temporarily, comeback time 200 mSec
E (4068) wifi:Association refused temporarily, comeback time 200 mSec

WiFi connected
IP address:
172.20.10.3
Attempting MQTT connection...connected
```

Fig. 6. Controller connected to the Wi-Fi and MQTT Broker.

In Fig. 6, ESP3, which is identified as the MQTT client, connects to the broker and Wi-Fi as soon as it is turned on. The Wi-Fi SSID and password are pre-configured within the code to expedite this networking process and make it easier to find the network. The Serial Monitor shows the allocated IP address of the ESP3 after a successful Wi-Fi connection has been made, allowing for effective control and monitoring.

Then, ESP3 tries to establish a connection with the MQTT Broker. In the case that the broker is temporarily unavailable, ESP3 continues to try until a successful connection with the broker is established. Due to this strong connectivity plan, which guarantees system resilience, ESP3 may easily interact with the MQTT Broker and add to the network's overall dependability.

```
Message received on topic: /Amir/sensor1
Received message: a
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: Connected to Receiver
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: a
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: a
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: a
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: a
Message received on topic: /Amir/sensor1
Received message: b
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: b
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: b
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: b
Invalid data received: Sensor Data from c0:49:ef:f1:43:ca: b
```

Fig. 7. Sent indicator from MQTT broker.

One essential aspect of ESP3 is subscribing to different topics related to ESP1. Among the subscribed topics shown in Fig. 6 is "/Amir/sensor1." As soon as the broker releases important indications, ESP3 immediately sends the associated data via the hardware serial port to ESP2 (Central BLE). Following that, ESP2 is in charge of sending this received data to ESP1 via a BLE connection, handling it as an important message. The sensor is then activated or deactivated depending on the information received by this indication, which in turn causes actions on ESP1. The system's collaborative and linked nature is highlighted by this complex communication flow, in which ESP3 is crucial in transferring important data between ESP1 and ESP2.

```
Hardware serial is connected..
Received from ESP3: a
a
Broker sent an indicator..
Data sent to peripheral c0:49:ef:f1:43:ca
Received Value from device c0:49:ef:f1:43:ca: a
Received Value from device c0:49:ef:f1:43:ca: a
Received Value from device c0:49:ef:f1:43:ca: a
Received Value from device c0:49:ef:f1:43:ca: a
Hardware serial is connected..
Received from ESP3: b
b
Broker sent an indicator..
Data sent to peripheral c0:49:ef:f1:43:ca
Received Value from device c0:49:ef:f1:43:ca: b
Received Value from device c0:49:ef:f1:43:ca: b
Received Value from device c0:49:ef:f1:43:ca: b
Received Value from device c0:49:ef:f1:43:ca: b
```

Fig. 8. Central BLE Serial Monitoring.

When an indicator is sent from the MQTT Broker, the Serial Monitoring interface for ESP2 records the operation, as seen in Fig. 8. After ESP2 receives this indication, it relays it to ESP1 via the BLE connection that has been established between the MQTT Broker and ESP2. ESP1's next course of action depends on the type of indicator. If it indicates that the sensor has been activated, it will immediately relay sensor data back to Central BLE. On the other hand, ESP1 stops transmitting sensor data if the indicator requires that the sensor be turned off. The bidirectional communication flow that ESP2 orchestrates inside the system is highlighted by this complex interchange of indicators and sensor data, highlighting its critical role in coordinating, and enabling the real-time interaction between ESP1 and Central BLE.

```
a
Sensor Data from c0:49:ef:f1:43:ca: Connected to Receiver
Sensor Data from c0:49:ef:f1:43:ca: Connected to Receiver
Sensor Data from c0:49:ef:f1:43:ca: a
Sensor Data from c0:49:ef:f1:43:ca: a
Sensor Data from c0:49:ef:f1:43:ca: a
Sensor Data from c0:49:ef:f1:43:ca: a
b
Sensor Data from c0:49:ef:f1:43:ca: b
Sensor Data from c0:49:ef:f1:43:ca: b
Sensor Data from c0:49:ef:f1:43:ca: b
Sensor Data from c0:49:ef:f1:43:ca: b
```

Fig 9. MQTT Broker output.

The MQTT Broker's output interface, which acts as the primary hub for publishing sensor data and indicators, is shown in Fig. 9. ESP3 actively keeps an eye on the messages posted to the topics to which it has subscribed inside this domain. When ESP3 finds the important message related to the subscribed subject, it uses the hardware serial interface to send this important data to ESP2. This complex procedure highlights the Broker's critical function as a source of important data, while ESP3 plays the role of a perceptive middleman, relaying only certain messages to support the smooth system communication flow.

B. Problem Encountered.

A significant issue with the Central BLE (Bluetooth Low Energy) module's connectivity surfaced during the Smart Home IoT Controller's testing phase. The goal of the Central BLE's planned functionality was to connect to a number of clients in the smart home network, including different appliances and sensors. Nevertheless, a problem was found where the Central BLE module seems to run into restrictions, enabling it to connect to a single ESP32 at a time. A "load prohibited" error was returned when an attempt was made to connect to more than one ESP at once, which caused the Central BLE ESP to reset.

The system's intended scalability and multi-device communication capabilities are seriously hampered by this problem. Comprehending and resolving this constraint is essential to guaranteeing the smooth incorporation of the Central BLE with multiple sensors and appliances, an essential prerequisite for the all-encompassing and effective functioning of the Smart Home IoT Controller. In order to move the system closer to its intended capacity to connect and handle several devices simultaneously, more research into the root reasons of this connectivity bottleneck and potential remedies will be essential.

V. CONCLUSION AND RECOMMENDATION.

In conclusion, the Smart Home IoT Controller is a notable development in the field of smart home systems due to its combination of MQTT and Bluetooth Low Energy (BLE). The smooth integration of BLE and MQTT has addressed key issues like maximising energy efficiency, increasing communication range, and strengthening security procedures. The ability of BLE to handle low-power, short-range communication has been perfectly utilised in the connection between sensors and the central BLE. Concurrently, adding MQTT has greatly improved the system's adaptability, dependability, and security, providing a strong basis for responsive and user-focused smart home experiences. Since the publish-subscribe architecture of MQTT, the continuous operation model makes sure that the smart home environment is always alert and flexible.

However, testing has shown a serious problem in the connectivity capabilities of the Central BLE module, even though the introduction of the Smart Home IoT Controller integrating Bluetooth Low Energy (BLE) and MQTT technologies represents a significant advancement in smart home systems. The system's intended scalability and multi-device communication are seriously hampered by this connectivity barrier.

For the Smart Home IoT Controller to advance, it will be essential to recognise this difficulty, deal with the underlying causes, and put effective solutions in place. In order to fully utilise the system's potential for handling several sensors and appliances at the same time and promote a responsive and all-encompassing smart home experience, multi-client connectivity constraints must be resolved. In order to guarantee the smooth integration of the Central BLE with the intended network of devices, future iterations and improvements must give priority to fixing this connectivity constraint. This emphasises the need for constant evolution and improvement in the ever-changing field of smart home technology.

In order to increase its adaptability and usefulness, the Smart Home IoT Controller may in the future investigate integrating a wider range of sensors and gadgets. Specialising on improving security features possibly by applying sophisticated encryption techniques would strengthen the system's defences against new cyberthreats in the dynamic field of smart home technologies. Maintaining interoperability with an ever-expanding range of smart home devices and communication protocols requires constant testing and optimisation. Together, these suggestions provide a path for the Smart Home IoT Controller's continued development and improvement, putting it at the forefront of safe and intelligent home automation.

ACKNOWLEDGMENT

The author expresses gratitude to the College of Engineering, School of Electrical Engineering, Universiti Teknologi MARA (UiTM) for their unwavering support of this project through to its completion.

REFERENCES

[1] K. Agarwal, A. Agarwal and G. Misra, "Review and Performance Analysis on Wireless Smart Home and Home Automation using IoT," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 629-633, doi: 10.1109/I-SMAC47947.2019.9032629.

[2] J. Prabaharan, A. Swamy, A. Sharma, K. N. Bharath, P. R. Mundra and K. J. Mohammed, "Wireless home automation and security system using MQTT protocol," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2017, pp. 2043-2045, doi: 10.1109/RTEICT.2017.8256958.

[3] A. Cornel - Cristian, T. Gabriel, M. Arhip-Calin and A. Zamfirescu, "Smart home automation with MQTT," 2019 54th International Universities Power Engineering Conference (UPEC), Bucharest, Romania, 2019, pp. 1-5, doi: 10.1109/UPEC.2019.8893617.

[4] Q. T. Tran, D. D. Tran, D. Doan and M. S. Nguyen, "An Approach of BLE Mesh Network For Smart Home Application," 2020 International Conference on Advanced Computing and Applications (ACOMP), Quy Nhon, Vietnam, 2020, pp. 170-174, doi: 10.1109/ACOMP50827.2020.00034.

[5] A. Raza et al., "A Home Automation Through Android Mobile App By Using Arduino UNO," 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 2020, pp. 1-6, doi: 10.1109/INMIC50486.2020.9318084.

[6] B. V. Albons, K. H. Yusof, N. H. Mahamarowi, A. S. Ahmad and A. S. M. Azlan, "Designation of a Home Automation System using Nodemcu with Home Wireless Control Appliances in Traditional Malay House," 2022 Engineering and Technology for Sustainable Architectural and Interior Design Environments (ETSAIDE), Manama, Bahrain, 2022, pp. 1-3, doi: 10.1109/ETSAIDE53569.2022.9906385.

[7] A. S. Hamzah and L. A. Abdul-Rahaim, "Smart Homes Automation System Using Cloud Computing Based Enhancement Security," 2022 5th International Conference on Engineering Technology and its Applications (IICETA), Al-Najaf, Iraq, 2022, pp. 164-169, doi: 10.1109/IICETA54559.2022.9888307.
[8] A. Chakraborti, A. Jain, S. Menon and K. Samdani, "A Review of Security Challenges in Home Automation Systems," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, pp. 1-6, doi: 10.1109/ICSCAN.2019.8878722.
[9] C. S. Siddhartha, G. P. Kalyan, T. V. Naga Dinesh, D. Vadlamudi and S. Thatavarti, "Enhancing Home Security: user Authentication Techniques for Home Automation," 2023 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 2023, pp. 1166-1171, doi: 10.1109/ICICT57646.2023.10133997.
[10] T. Akilan, R. Srivastava, Chandraprabha, B. Ahmad, H. Islam and M. Saif, "Arduino Based Smart Security For Home Automation," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021, pp. 804-808, doi: 10.1109/ICAC3N53548.2021.9725436.
[11] A. Arunachalam, R. Raghuraman, P. Obed Paul and J. Vishnupriyan, "A System for Energy Management and Home Automation," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 2021, pp. 1-3, doi: 10.1109/ICSCAN53069.2021.9526526.
[12] M. S. K. Rafi, T. K. Dia, Z. F. Chowdhury, R. Shelim and M. Al Khaledunnabi, "Conserve and Reserve: A New Approach to Home Automation and Bill Management System," 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA), Subang Jaya, Malaysia, 2018, pp. 1-6, doi: 10.1109/ICACCAF.2018.8776738.
[13] M. E. Haque, M. R. Islam, M. T. Fazle Rabbi and J. I. Rafiq, "IoT Based Home Automation System with Customizable GUI and Low Cost Embedded System," 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2019, pp. 1-5, doi: 10.1109/STI47673.2019.9068035.
[14] S. K. Vishwakarma, P. Upadhyaya, B. Kumari and A. K. Mishra, "Smart Energy Efficient Home Automation System Using IoT," 2019 4th International Conference on IoT: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-4, doi: 10.1109/IoT-SIU.2019.8777607.
[15] R. K. Kodali and S. Yerroju, "Energy Efficient Home Automation Using IoT," 2018 International Conference on Communication, Computing and IoT (IC3IoT), Chennai, India, 2018, pp. 151-154, doi: 10.1109/IC3IoT.2018.8668155.
[16] K. Khanchuea and R. Siripokarpirom, "A Multi-Protocol IoT Gateway and WiFi/BLE Sensor Nodes for Smart Home and Building Automation: Design and Implementation," 2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), Bangkok, Thailand, 2019, pp. 1-6, doi: 10.1109/ICTEmSys.2019.8695968.