Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
Volume7-Issue2, January 2024
Paper: 15

IoT IR Remote Control System

Dharanika S[1], Dhaniya Lakshmi G[2], Divya K[3], Priya Darsini V M.E[4],

[1,2,3] Students,and [4] Faculty
Dept. of Information Technology,
Panimalar Engineering College,
Chennai, India.
dharanika27112003@gmail.com

Abstract: *The project aims to create a home automation system using an ESP32C3 microcontroller,IR (Infrared) communication, and the MQTT protocol. This system enables users to remotely control devices (such as TVs, air conditioners, or lights) using IR signals sent via MQTT messages in an application. The ESP32C3 serves as the central hub, connecting to an MQTT broker and translating incoming MQTT messages into IR signals to control various devices. This project enhances user convenience and facilitates seamless device management within a smart home environment. This project showcases the integration of an ESP32C3 microcontroller with an MQTT broker, allowing the device to communicate with other devices or applications in an IoT ecosystem. It demonstrates the use of MQTT for message exchange, IR communication for device control, and sensor readings for environmental monitoring. The project's modular architecture allows for flexibility and scalability, making it suitable for a wide range of IoT applications.*

Keywords: *Internet of Things(IoT), Sensors, Infrared Rays(IR), Room temperature, Microcontroller, Hardware devices, MQTT protocols, MQTT brokers, Room temperature, LDR.*

INTRODUCTION:

In the modern era, the Internet of Things (IOT) has revolutionized the way we interact with technology, transforming everyday devices into intelligent, interconnected systems. Home automation, a prominent facet of IOT, has redefined living spaces, making them more efficient, convenient, and responsive to our needs. One significant challenge in this landscape has been the integration of Infrared (IR) devices – the ubiquitous controllers of TVs, air conditioners, and other appliances into the broader IOT ecosystem. This challenge is the focal point of our research. Ananya Roy et al.[8] 2017 created a monitoring system using IOT that can use various access methods which includes Wi-Fi, GPRS, Ethernet, and so on, and the data obtained may also be saved. It employs an ATmega 328 Microcontroller. The real-time detection of the temperature and humidity of the storage rooms may be improved with the help of this monitoring system, and the longevity of the products can be ensured.

In an era where smart homes are becoming increasingly prevalent, the ability to control various devices remotely and seamlessly is crucial. Utsav Gada et al.[10] 2021 presented a temperature monitoring system developed on the hardware side with a Raspberry Pi, camera, and sensors, and Python operating in the backend. Temperature monitoring systems are real-time systems that save data on a cloud-based real-time database called Firebase. This project presents an innovative solution that combines the power of an ESP32-C3 microcontroller, Infrared (IR)

communication, and the MQTT (Message Queuing Telemetry Transport) protocol to control the devices. Our research endeavors to bridge this gap by developing a sophisticated IOT framework that harmoniously unites IR devices with the broader IOT ecosystem. The primary objective is to empower users with effortless, unified control over all their appliances. This control extends beyond the confines of their living spaces, allowing for remote operation and intelligent automation. Additionally, we aim to create a system that is adaptive, capable of understanding various IR protocols, and flexible enough to accommodate new devices seamlessly.

ARCHITECTURE AND ITS COMPONENTS:

The architecture of an IoT Infrared (IR) Remote Control System includes a sophisticated framework that combines infrared technology and the Internet of Things (IoT) to create a faultless and intelligent control ecosystem with the help of the MQTT protocol.

ESP32C3 Microcontroller:

The ESP32C3 acts as the central processing unit. It is responsible for handling Wi-Fi connectivity, MQTT communication, IR signal generation, and device control logic.

IR Transmitter and Receiver: The ESP32 is equipped with an IR transmitter to send IR signals to control devices and an IR receiver to capture and decode incoming IR signals from remote controls.

MQTT Protocol: MQTT serves as the communication protocol between the ESP32 and other devices or applications. The ESP32 subscribes to specific MQTT topics and listens for incoming messages. When it receives MQTT messages indicating device control commands, it translates these commands into corresponding IR signals for transmission.

Infrared (IR) Blaster: The IR blaster is a crucial component that translates digital MQTT commands into IR signals. These signals mimic those from traditional remote controls, allowing the ESP32 to communicate with IR-enabled devices.

Device Control Logic: This layer interprets incoming MQTT messages and maps them to specific IR codes corresponding to various devices. It contains logic to encode the commands and send appropriate IR signals through the IR transmitter.

Wi-Fi Connectivity: The ESP32 establishes a Wi-Fi connection to the local network, allowing it to communicate with the MQTT broker and other devices within the network.

3. LAYERS AND FRAMEWORKS

The implemented system consists of the various layers that are described below.

3.1 Hardware Layer:

ESP32 Development Board: ESP32C3 microcontroller serves as the hardware foundation for the

project.

IR LEDs and Receivers: Components for transmitting and receiving IR signals.

3.2 Communication Layer:

MQTT Protocol: Used for sending and receiving messages between the ESP32C3 and other devices or applications in the network.

Wi-Fi Connectivity: Establishes a wireless connection to the local network, enabling MQTT communication.

3.3 Device Control Layer:

IR Signal Generation: Logic for encoding device-specific commands into IR signals.

MQTT Message Interpretation: Interprets incoming MQTT messages and converts them into actionable device control commands.

User Interface: Mobile App or Web Interface: Provides users with a graphical interface for sending control commands via MQTT messages. Users can interact with devices, view device status, and receive feedback.

3.4 Security Layer :

Authentication and Encryption: Implements security measures to ensure secure communication between the ESP32C3 and the MQTT broker. This layer may include username/password authentication and encryption protocols to protect sensitive data.

By integrating these layers and frameworks, the project creates a robust and flexible system for remotely controlling devices within a smart home environment. The use of MQTT protocol ensures efficient and reliable communication, while the IR functionality allows seamless integration with existing IR-controlled appliances and devices.

4. OUTLINE OF ESP32C3:

ESP32-C3 is an open-source RISC-V, a microcontroller SoC with a single core that supports Bluetooth 5 (LE) and Wi-Fi. It provides the most affordable solution for linked devices by achieving the ideal balance between power, I/O capabilities, and security. The presence of both Bluetooth 5 (LE) and Wi-Fi connection simplifies the device's setting process and opens up a range of dual access use cases.ESP32 is a system on a chip that integrates the following characteristics:

1. Wi-Fi (2.4 GHz band)
2. Bluetooth
3. Dual high performance Xtensa® 32-bit LX6 CPU cores
4. Ultra Low Power co-processor
5. Multiple peripherals

Powered by 40 nm technology, ESP32 provides a robust, highly integrated platform, which helps meet the continuous demands for efficient power usage, compact design, security, high performance, and reliability and the other specifications are mentioned in ***Table 1.***Below ***Figure***

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
Volume7-Issue2, January 2024
Paper: 15

2 shows the ESP Devkit with corresponding GPIO pins.

Fig.2.ESP32 DevKit VI-DOIT

4.1 Features of ESP32C3:

The various features of ESP32-C3 are described below.

4.1.1 RISC-V at the Core: The 32-bit core RISC-V microprocessor in the ESP32-C3 has the highest clock speed of 160 MHz. It supports a broad range of use cases involving linked devices because of its 22 GPIOs that are flexible 400 KB of internal RAM, and low-power mode support. Several changes to the MCU exist with both built-in and external flash. Its high-temperature support is perfect for industrial and lighting use cases.

4.1.2 Rich Connectivity: Rich Connectivity: Creating devices with high coverage and raised value is made possible by Bluetooth 5 (LE) and Wi-Fi, both of which offer long-range (LR). Espressif Wi-Fi Mesh and Bluetooth LE SIG Mesh are still available by the ESP32-C3. Moreover, ESP32-C3 maintains its strong RF performance at a greater working temperature.

4.1.3 Security: In maintaining the availability of technologies like flash encryption based on AES-128/256-XTS and secure boot based on RSA-3072, ESP32-C3 ensures that connected devices may be built safely. Applications can benefit from a secure device identity thanks to the HMAC and the cutting-edge digital signature devices. Effectiveness for safe communication in a local network and with the cloud is ensured by hardware acceleration support for cryptographic algorithms.

4.1.4 Software Availability: The open-source ESP-IDF currently powers millions of devices in use and includes ESP32-C3. This ensures developers will have simple access to a stable SDK and tools and a path for switching applications. ESP-AT and ESp-hosted solutions can also be utilized with an external host MCU and ESP32-C3.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
Volume7-Issue2, January 2024
Paper: 15

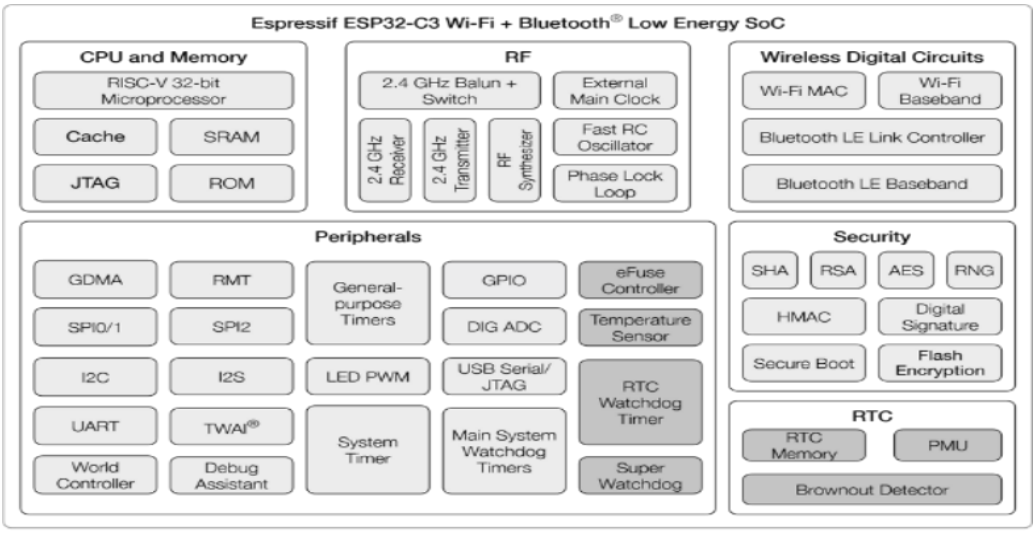


Fig.3 Function Block diagram of ESP32.

Microcontroller	ESP32C3
Operating voltage	3.3V
Input voltage (recommended)	3.3V
Input voltage (limit)	1.8V-3.6V
Digital I/O pins	22 pins
Analog input pins	18 pins
DC current per I/O pin	12mA
DC voltage for 3.3V pin	100mA
Flash memory	384KB
SRAM	400KB
Clock speed	160MHz(default)

Table1: Specifications of ESP32C3

5. WORKING OF MQTT WITH ESP32-C3:

The MQTT (Message Queuing Telemetry Transport) protocol serves as a backbone in numerous IoT applications, enabling efficient communication between devices. When integrated into an IR (Infrared) blaster system powered by ESP32-C3, MQTT enhances the control and interoperability of IR devices. Below, we explore how MQTT is utilized in an IR blaster context,

particularly focusing on the ESP32-C3 microcontroller, to control various devices seamlessly.The below ***Figure 4.*** Shows the basic architecture and working of MQTT protocol.

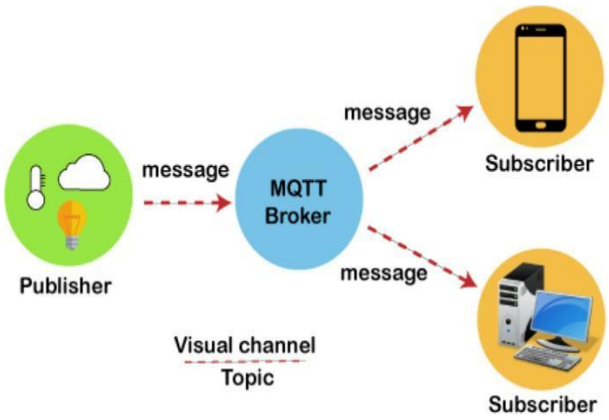


Fig.4.Architecture of MQTT

5.1 Publishing and Subscribing to Topics:

Publishing IR Commands: The ESP32-C3, acting as a publisher, sends IR commands as messages to specific MQTT topics. For example, sending a message to the topic "/devices/TV" could indicate a command to turn on the television.

Subscribing to Commands: IR blaster devices, acting as subscribers, subscribe to relevant topics. For instance, an AC unit IR blaster may subscribe to topics like "/devices/AC" to receive commands specific to air conditioning control.

5.2 Device Control Commands:

IR Code Encapsulation: IR commands, representing functions like power on, volume control, or mode selection, are encapsulated within MQTT messages. These messages are then published to respective topics.

Interpreting MQTT Messages: Upon receiving MQTT messages, the ESP32-C3 extracts the IR commands, translates them into IR signals, and emits them using the IR blaster module. This process enables the ESP32-C3 to control devices that accept IR commands.

5.3 Bi-Directional Communication:

Status Feedback: Devices controlled by IR blasters can provide status feedback. For instance, an MQTT message sent by a TV to the topic "/devices/TV/status" might indicate whether the TV is on or off. Subscribed clients, such as a mobile app, can receive this status information, enabling real-time feedback and synchronization.

5.4 Integration with Home Automation Systems:

Scene and Automation Triggers: MQTT messages can be integrated with home automation systems. For example, a "Movie Night" scene could trigger various IR commands to turn on the

TV, dim the lights, and adjust the AC, all initiated by a single MQTT message.

5.5 Security and Authentication:

Secure Communication: MQTT protocol supports secure communication through mechanisms like TLS/SSL encryption. This ensures that IR commands and status updates are transmitted securely between devices and brokers.

Authentication: ESP32-C3 devices can authenticate themselves with the MQTT broker, preventing unauthorized access and ensuring that only authenticated devices can publish or subscribe to specific topics.

By leveraging MQTT, an IR blaster system powered by ESP32-C3 becomes a versatile, efficient, and secure solution for controlling a wide array of devices, ranging from entertainment systems to climate control appliances, within the IoT landscape. Its ability to handle real-time communication and accommodate various control scenarios makes it an invaluable tool in modern smart homes and industrial automation setups.

6. IMPLEMENTATION:

The implementation techniques that are used in this research are described below.

6.1 MQTT Protocol Integration:

Initialization: The code initializes an MQTT client with the specified broker URL and client ID. The client is then started and connected to the MQTT broker.

Event Handling: The application defines event handler functions for different MQTT events like connection, disconnection, subscription, publishing, and receiving data. These handlers manage the MQTT communication flow.

6.2 Event-Driven Programming:

Event-Driven Architecture:The code follows an event-driven architecture, where specific functions are triggered in response to MQTT events. For example, when the device is connected to the MQTT broker(MQTT_EVENT_CONNECTED), it subscribes to a specific topic. These events drive the application's behavior.

6.3 MQTT Message Handling:

Publishing Messages: The code publishes MQTT messages to specific topics using the esp_mqtt_client_publish function. Messages are formulated as JSON objects, containing relevant data such as IR commands or status information.

Receiving Messages: Incoming MQTT messages are processed in the MQTT_EVENT_DATA event handler. The code extracts data from received messages and performs actions based on the message content, such as triggering IR commands or reading sensor data.

6.4 Error Handling and Debugging:

Error Logging: The code includes error handling mechanisms, logging detailed error messages when MQTT-related errors occur. Error information includes ESP-TLS errors and transport

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
Volume7-Issue2, January 2024
Paper: 15

socket errors, aiding in debugging and troubleshooting.

Debugging Statements: Debugging statements (ESP_LOGI and print) are strategically placed throughout the code to provide real-time information about the application's state, MQTT events, and received data.

6.5 Integration with External Components:

IR Functionality: The code integrates with an IR blaster module, allowing the device to send and receive IR commands. IR commands are encapsulated in MQTT messages, enabling the control of IR devices remotely.

Temperature and LDR Readings: The application reads temperature and light-dependent resistor (LDR) values. These readings can be triggered remotely using MQTT messages, providing additional functionality beyond IR control.

6.6 Resource Management and Optimization:

Memory Management: The code includes statements to check and display free heap memory, allowing developers to monitor memory usage and optimize their applications. ***Task Delay:*** The code utilizes task delays (vTaskDelay) to manage the timing of certain operations, ensuring efficient resource utilization and preventing unnecessary CPU cycles. Below ***Figure 5*** represents the development of applications using ESP32-C3.

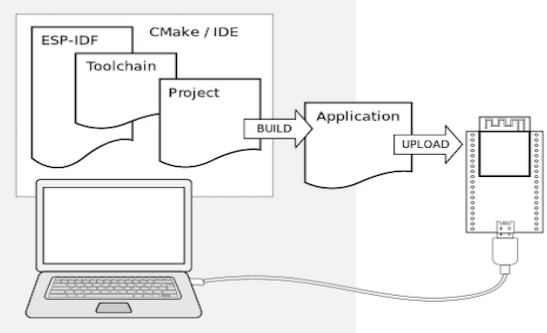
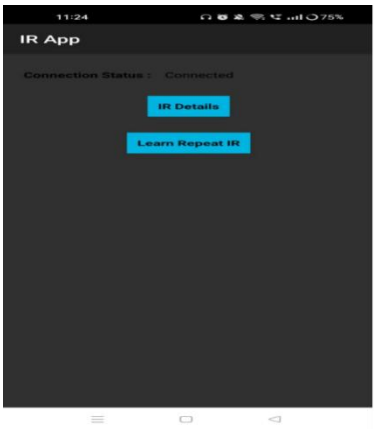


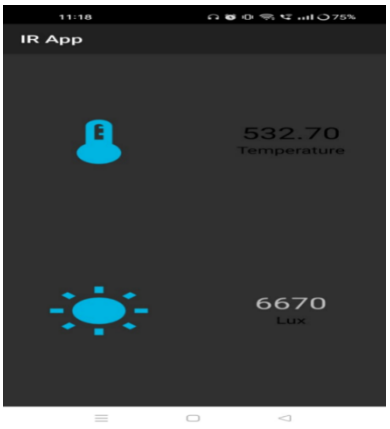
Fig.5. Development of applications for ESP32C3

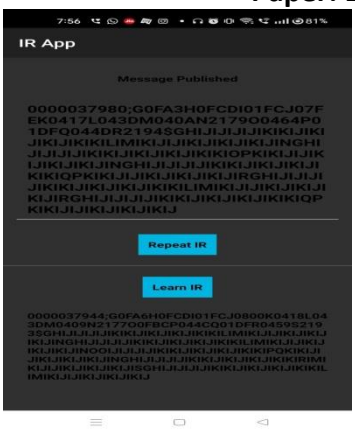
7. SAMPLE OUTPUT:

Front View	IR details -> Displaying Temperature & LUX	Learn IR Repeater IR: -> Message published

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
Volume7-Issue2, January 2024
Paper: 15







8 . CONCLUSION:

In this research, we have explored the implementation of an ESP32-based IoT device utilizing the MQTT protocol for seamless communication with a broker. The implementation demonstrates a robust event-driven architecture, enabling the device to send and receive MQTT messages effectively. By integrating with external components such as IR blasters and sensors, the device showcases real-world applications, such as remote control of IR devices and environmental monitoring. The adaptability is highlighted through dynamic configuration options, allowing users to input the MQTT broker URL interactively. Additionally, comprehensive error handling and debugging mechanisms ensure the reliability of the system, making it suitable for practical IoT deployments.

9. Future Enhancements*:*

In subsequent years, this research may have further improvements that are mentioned below.

Voice Control: Integrating voice recognition technology would enable users to control IR devices using voice commands, enhancing user convenience.

Device Discovery: Implement mechanisms for automatic discovery of devices within the MQTT network. This simplifies the configuration process for users and improves overall user experience.

Advanced Sensor Integration: Integrate additional sensors such as motion sensors, gas sensors, or sound sensors. This broadens the device's capabilities, allowing it to monitor a wider range of environmental parameters.

Energy Efficiency: Implement power-saving techniques to optimize the device's energy consumption. This is crucial for IoT devices powered by batteries, ensuring longer operational life spans. By addressing these future enhancements, the IoT system can be elevated to a higher level of functionality, security, and user experience, making it well-suited for various IoT applications in smart homes, industrial automation, and environmental monitoring.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
Volume7-Issue2, January 2024
Paper: 15

REFERENCE

[1]. Ibtihaj A. Abdulrazzak," Humidity and temperature monitoring",2018 International Journal of Engineering & Technology (IJET).

[2]. Karishma Bonia," Controlling Home Appliances by IR Remote Control using Arduino Uno",2021 ADBU Journal of Electrical and Electronics Engineering (AJEEE) Volume 4, Issue 2.

[3]. Rajarajeswari S," home automation through smart lighting, smart security and other appliances",2021 ITM Web of Conferences 37, 01024 .

[4]. Ravi Kishore Kodali," IoT Based Smart Security and Home Automation System",2016 International Conference on Computing, Communication and Automation (ICCCA2016).

[5]. Lalbihari Barik," IoT based Temperature and Humidity Controlling using Arduino and Raspberry Pi", 2019 International Journal of Advanced Computer Science and Applications(IJACSA).

[6]. Zheng Shi," Outage Performance and AoI Minimization of HARQ-IR-RIS Aided IoT Networks",2023 Institute of Electrical and Engineers Conference (IEEE).