Course Name: Internet of Things (IO4041)
Semester: BCS-9A(9th)

Project Title: Home Automation using MQTT and ESP8266

Group Members:

Muhammad Junaid……19L-2250
Armash Javed…….19L-2278

Submission Date: 5th December 2023

Armash Javed: Smart Bulb
Muhammad Junaid: Temperature Sensor

Armash Javed:

Armash Javed created the connection between mqtt broker and the mobile phone. He has implemented the LED control aspect of the project using MQTT protocols. He utilized the NodeMCU ESP8266 module to effectively manage the LED's functionality. Armash utilized MQTT protocols to establish communication between the LED and the MQTT broker. Using MQTT's capabilities, he successfully developed the functionality allowing remote control to turn the LED on or off.

Muhammad Junaid:

Muhammad Junaid created the mqtt server(broker) which is mosquitto. He contributed to the project by focusing on integrating the temperature sensor into the MQTT-based system. He utilized the NodeMCU ESP8266 module to interface with the temperature sensor effectively. Using MQTT protocols, Muhammad established a mechanism that allowed the temperature sensor to publish real-time temperature data to the MQTT broker. Moreover, he configured MQTT clients to subscribe to this data stream, enabling remote monitoring of temperature readings.

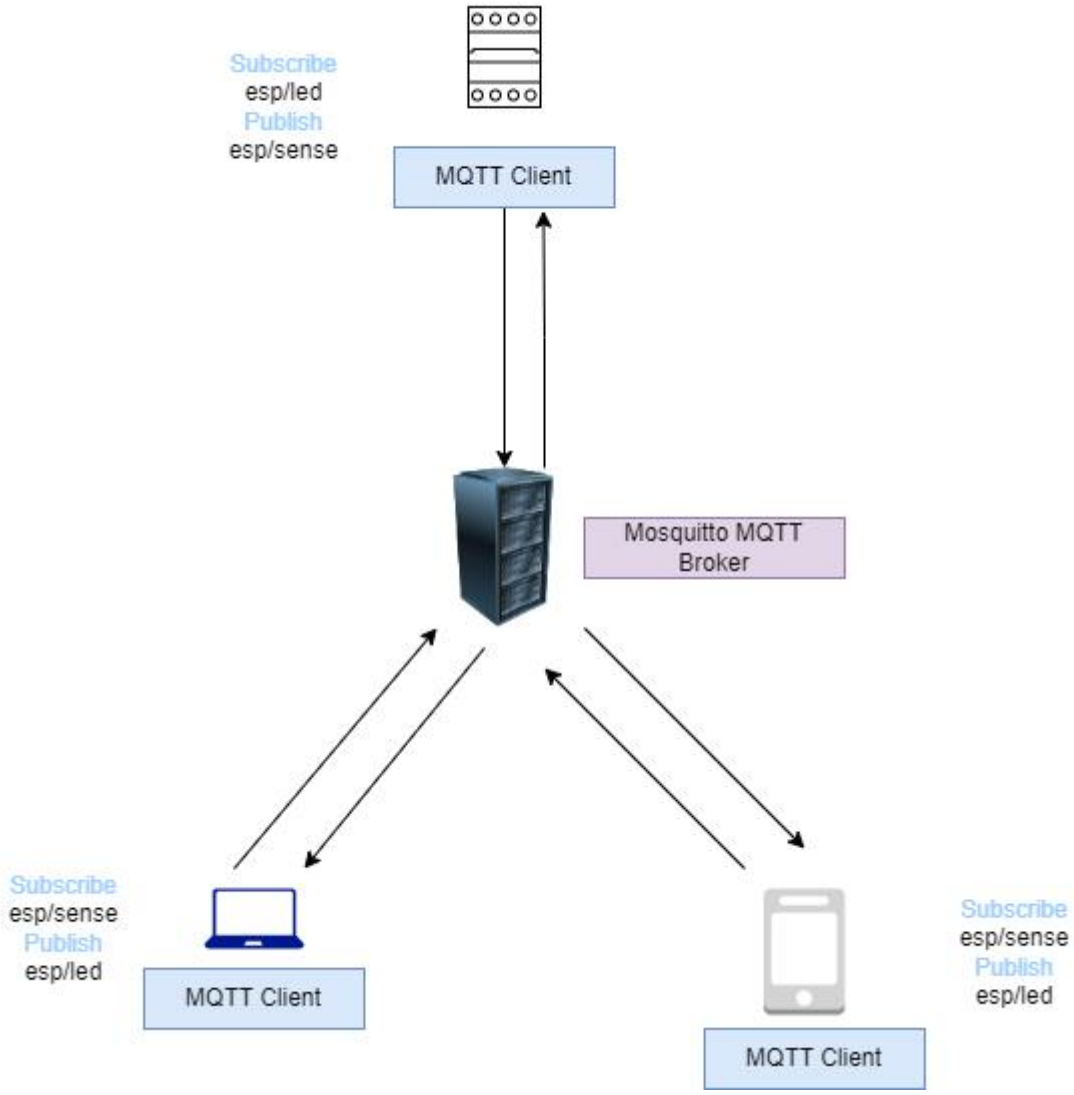
Overview:

This project focuses on enhancing homes by introducing smart technology that enables remote monitoring and control of various appliances and sensors within the house. By utilizing a specific type of messaging system and a small but powerful computer chip, it aims to establish connectivity between different devices at home, such as lights, sensors, and other gadgets. This connectivity allows users to manage and supervise these devices even when they're away from home. Ultimately, the objective is to enhance energy efficiency, and overall convenience. In our project, we are specifically targeting two main categories one is led and the other one is temperature sensing.

Problem Statement:

Traditional home setups face several challenges. For instance, smaller devices often struggle to communicate effectively without draining a lot of power. Ensuring that the system is secure, user-friendly, and cost-effective can also pose difficulties. This project seeks to address these issues by developing a smart home system that overcomes these challenges. It intends to create a solution that is easy to use, reliable, secure, and power-efficient, making it suitable for regular households without being overly expensive.

High Level Block Diagram:



MQTT Client:

It can be any device that is subscribing to a topic like if a smart phone is subscribing to a topic it is a mqtt client or if a LED is subscribing to a led topic for actuation it is also a mqtt client.

MQTT Broker:

It is a mqtt server which is helping to communicate mqtt clients by either publishing or subscribing to a topic. In our project we have used the mosquito mqtt broker as it is an open source.

Methodology/Mechanism

This section highlights the approach taken to implement the smart home system using MQTT and ESP8266.

1. Network Setup:

This project utilizes ESP8266, a development board connected to various sensors and actuators. The board processes data from sensors and controls the actuators. It serves as a gateway for transmitting data through WiFi. MQTT protocol is used for communication, allowing ESP8266 to publish sensor data under a specific topic and subscribe to receive commands for actuation.

2. Software Setup:

The Arduino IDE is employed to program the ESP8266 module as an MQTT client. A Mosquitto broker, an open-source MQTT broker, is set up on a Windows PC, managing MQTT clients. MQTTLens, a Google Chrome-based application, and an Android app, serve as MQTT clients for message transmission and reception.

3. MQTT Protocol Operations:

The MQTT protocol is structured around specific operations:

- Connection Establishment: Client can establish a connection to the MQTT broker through the connect() method of the PubSubClient library.
- Publishing Data: Client publishes temperature data retrieved from an sensor to the MQTT broker using the publish() method of the PubSubClient library
- Subscribing/Unsubscribing: Client subscribes to the "device/led" topic to receive messages that control the built-in LED. When a message is received on this topic, the callback() function is triggered, allowing the device to respond accordingly by turning the LED on or off.

This section outlines how ESP8266 is configured, the software tools utilized, and the operational aspects of MQTT, offering a comprehensive understanding of the technical process involved in building the home automation system.

IoT Aspects

The IoT (Internet of Things) aspects embedded within this MQTT and ESP8266-based home automation system are fundamental to its functionality and capabilities which are listed below:

1. Device Interconnection:

IoT involves interconnecting various devices (sensors, actuators, and the ESP8266 module in this case) to form a network. The ESP8266, acting as a central node or gateway, connects these devices to enable communication and data exchange.

2. Sensor Data Collection:

The system incorporates sensors (e.g., temperature sensors) to collect real-time data about the home environment. These sensors capture information crucial for decision-making and control within the automation system.

3. Actuator Control:

Actuators (such as LEDs) are integrated into the setup, allowing for remote control based on received commands. These can be turned on or off through MQTT messages, enabling remote interaction with home appliances.

4. MQTT Protocol for Efficient Communication:

MQTT, a lightweight and efficient protocol, is a key IoT component here. It enables seamless communication between devices while optimizing network bandwidth usage. MQTT's publish/subscribe mechanism allows devices to exchange data without complex setups, fitting well within IoT scenarios with limited resources.

5. Wireless Connectivity:

Utilizing WiFi connectivity via ESP8266 aligns with IoT principles, providing a wireless communication platform within the home network. This wireless connection simplifies deployment and connectivity across devices without extensive cabling.

6. Low Power Consumption:

An important IoT consideration is power efficiency. ESP8266's low power consumption aligns with IoT device requirements, ensuring efficient operation without consuming excessive energy.

Results associated with the Project

Here's the breakdown of the results associated with the MQTT and ESP8266-based home automation project:

1. MQTT Server Setup:

The project successfully implemented a Mosquitto MQTT broker. This allowed the establishment of a server capable of handling MQTT communication. Upon initiation, the server facilitated the publishing and subscribing of application messages between devices.

2. Data Transmission and Reception:

The ESP8266 module was effectively configured to publish sensor data to the MQTT broker. This sensor data, specifically the temperature measured by the LMH sensor, was transmitted over the MQTT protocol to the broker under the topic 'esp/sense.' Subsequently, any subscribed MQTT client could access and view these sensor readings.

3. Actuator Control:

Actuators (LED) connected to the ESP8266 were successfully controlled remotely through MQTT commands. The ESP8266 subscribed to topics 'esp/led' and 'esp/temp' to receive instructions, enabling the switching on or off of LEDs and the buzzer.

4. Message Exchange Verification:

The communication between MQTT clients and the broker, including the exchange of MQTT messages, was validated and monitored through tools like MQTTLens and an android app. These applications served as MQTT clients, effectively subscribing to sensor data and publishing commands, demonstrating the functional interaction between devices.

These results collectively demonstrate the successful implementation of a home automation system leveraging MQTT and ESP8266. The project accomplished remote data monitoring, actuator control, efficient message exchange, and compliance with IoT principles of connectivity, efficiency, and remote accessibility. The demonstrated functionalities lay the groundwork for creating smart and energy-efficient home environments.

Conclusion and Future works/directions

Here's the breakdown of the conclusion and future works/directions associated with the MQTT and ESP8266-based home automation project:

Conclusion:

1. Achieved Objectives:

The project successfully showcased the effectiveness of using MQTT as a lightweight protocol and ESP8266 as a reliable IoT device for home automation. It demonstrated remote monitoring of sensor data and actuator control through a common home gateway.

2. Efficiency and Accessibility:

Using the MQTT protocol ensured low bandwidth consumption and power-efficient communication. This led to a smart home infrastructure that promoted energy efficiency, accessibility, and seamless control of devices.

3. Enhanced Living Conditions:

The implemented system significantly improved living conditions by providing intelligent and comfortable automation solutions. It enabled users, including elderly or differently-abled individuals, to control home appliances conveniently and efficiently.

Future Works/Directions:

1. Cloud Integration:

Integration with cloud platforms can extend the system's capabilities, allowing data aggregation, analysis, and visualization.

2. Customized User Interface:

Developing a tailored graphical user interface (GUI) could offer users an easy to use and user-friendly platform for remote monitoring and control. A user-centric interface could simplify interaction and enhance user experience.

3. Security Enhancements:

Implementing robust security measures, such as encryption and authentication protocols, is crucial for safeguarding IoT devices and data against potential cyber threats. Enhancing security would ensure a more reliable and protected system.