IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101263

HOME AUTOMATION SOLUTION USING NODE-RED AND MQTT

Shubham Gupta[1], Saurav[2], Vidit Patira[3], Nishant Jain[4], Dr. Vijay Kumar[5]

Department of Computer Science and Engineering FET- Jain University Bangalore, Karnataka, India[1-5]

Abstract: Internet of Things (IoT) advancements have enabled smart home and industrial automation improvements, allowing gadgets in homes to be monitored and controlled remotely. Because appliances are monitored and controlled by small, resource restricted embedded devices, such systems have resulted in energy efficiency and cost savings. Using Message Queuing Telemetry Transport (MQTT) and Node-RED, the article offers a concept for an ESP8266 NodeMCU smart home solution. On the Raspberry Pi 4B, a single board computer development board, a MQTT mosquito broker is used in the smart home system design. A MQ2 sensor is connected to an ESP8266 microcontroller to collect the sensor data, with the Raspberry Pi acting as a MQTT broker to transfer sensor data to a Node-RED dashboard.

***Key-words*:** Node-RED, MQTT, Home Automation

I.INTRODUCTION

Internet of Things (IoT) is that the foremost evolving technology nowadays. This advancing technology has several applications resembling home automation, information Monitoring. attributable to home automation, we have a tendency to are accomplishing solace in our everyday lives. Home automation additional completely depicts homes inside that nearly everything: appliances, electrical outlets, heating associated cooling systems are snared to a remotely tractable framework. Home automation needs mutual communication of an oversize number of IoT devices. With the considerable rise within the amount of gadgets on the cloud platforms, there's a demand for refreshing microcode fairly often. It includes eliminating already introduced gadgets, creating necessary changes inside the code and flashing the altered code all over again. To beat these issues, process of knowledge ought to be presumably done elsewhere. Node-RED, that may be a visual wiring tool that helps in associating gadgets simply transfer regarding quick and easy affiliation setups. Gadgets are connected along to ESP8266 and a Mosquitto primarily based MQTT broker victimisation Node-RED and a connection is prepared up for remote watching and control.

Home Automation has been available for a few decades as a result of lighting and basic appliance administration, but it is only recently that technology has caught up with the idea of an interconnected world, enabling full control of the home from any location, becoming a reality and thus establish a smart home. The term "smart home" relates to a home in which all of the devices are connected to the internet. It can be viewed of as a framework which uses smartphones and computers to manage or communicate with household appliances. Today, the Internet of Things (IoT) has developed into a useful platform for a wide range of applications, including industrial automation, home automation, and medical applications, by connecting different sensors that monitor various device activities and analysing those data from anywhere.

II.LITERATURE SURVEY

Home automation is the automatic control of electronic devices in your home. These devices are connected to the Internet, which allows them to be controlled remotely. The scope not only includes automation but can be extended to security of the house also. This paper is presented with a simple MQTT implementation in a smart home environment using ESP8266 based development boards, Node MCU and Node-RED platform. MQTT is a simple protocol that employs a publish-subscribe (pub/sub) architecture. Messages are transmitted from a client to a server, which then transmits them to another client in the pub/sub architecture. The client receiving the message must be subscribed to the topic where the message was published in order for the message to be appropriately sent by the server. Because most smart home devices are geared for low power usage and have limited computing capability, MQTT is a natural choice. As a result, the MQTT pub/sub architecture allows smart devices, or the user directly, to collect data from the sensors and give commands in actual or close to real time in line with the collected signal. Node-RED enables users to fasten together Web services and gadgets by replacing common coding tasks and this should be possible with a visual drag-drop interface. Various components in Node-RED are connected together to create a flow in the Node-RED editor. We use Node-RED that acts as a subscriber to receive data from MQTT broker. With Node-RED, we design flows to manage and handle the received data to display them in the form of gauge, text and chart on the dashboard.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101263

III.PROPOSED SYSTEM ARCHITECTURE

The main requirements of the proposed system is to acquire the data of the sensors and send them by using MQTT protocol to the user whenever the user wishes to check the working or the environment data of its home or other place. The Raspberry Pi serves as a home gateway server that is linked to the home network, functioning as a tiny system with control over the sensors that provide information about the home environment, like temperature and humidity, and also device status. Likewise, here we have used MQ2 Gas Sensor for our project. Figure 1. display's the proposed architectural system for the NodeMCU ESP8266 smart home with MQTT and Node-RED system. The smart home automation application would be running on the Raspberry Pi MQTT broker controlling various devices in real time. This architecture presents merits of being low-cost, affordable, low resource needs and open source. The proposed system design of the ESP8266 smart home using MQTT and Node-RED was implemented with hardware not limited to NodeMCU ESP8266 microcontroller, Smoke and Gas sensor MQ2, Raspberry Pi 4b, Led and Buzzer.

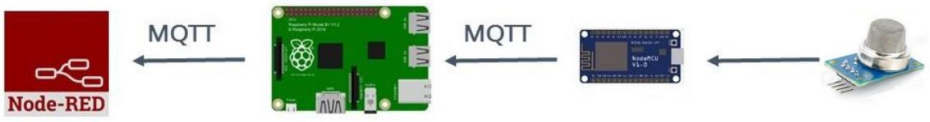

Fig. 1. Overview of a System Architecture

i.RASPBERRY PI

The Raspberry Pi is a small, low-cost computer. It works with a variety of programming languages, including Scratch, Python, Java, and others. Some of the product's primary features include a 64-bit quad-core processor with RAM memory varying from 2 GB to 8 GB, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE functionality. The dual-band wireless LAN and Bluetooth compliance certification on the board is modular, allowing it to be integrated into end devices with significantly less compliance testing, lowering both cost and time to market. The raspberry pi in this system is running the mosquitto MQTT broker and the deployment of the Node-RED.

ii.NODEMCU ESP8266

NodeMCU ESP8266 is an open-source Espressif microcontroller board based on Lua firmware. It can be used for different IoT applications based on a few script lines. The firmware runs on the ESP8266 Wi-Fi SoC from Espressif System. NodeMCU ESP8266 is a low-cost Wi-Fi module which has a TCP/IP. ESP8266 has ESP-12E module containing ESP8266 chip with Tensilica Xtensa 32-bit LX106 RISC microprocessor. The board has 128 KB of RAM and 4MB of Flash memory for data storage of data and programs. It has high processing power which makes it ideal crosscutting open edge IoT projects.

iii.MQ2 GAS SENSOR:

The MQ2 gas sensor is one of the most widely utilised in the MQ sensor series. It's a Metal Oxide Semiconductor (MOS) type Gas Sensor, also referred as Chemiresistors, because of detection being based on a modification in the sensing material's resistance when the gas makes contact with it. Gas concentrations can also be detected using just a simple voltage divider network. The MQ2 Gas Sensor runs on 5V DC and consumes about 800mW. It has a detection range of 200 to 10000ppm for LPG, Smoke, Alcohol, Propane, Hydrogen, Methane, and Carbon Monoxide.

iv.NODE-RED

Node-RED is an IBM-developed flow-based development tool for integrating hardware devices, APIs, and web applications as part of the Internet of Things. Node-RED provides a flow editor that can be used to build JavaScript functions in the browser. Application elements can be stored or distributed for re-use. Node.js is used to create the runtime. JSON is often used to store the flows made in Node-RED. MQTT nodes can now establish properly configured TLS connections since version 0.14. In 2016, IBM released Node-RED towards the JS Foundation as an open source platform.

v.MQTT

MQTT is a publish-subscribe protocol that runs over the TCP/IP network. It's a lightweight protocol that's meant to maximise network traffic while also providing the dependability of messaging applications created for these

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101263

environments fast and easily. Mobile applications, as well as telemetry and sensor device to device applications, can all gain from MQTT. A message broker is required for the public-subscribe messaging pattern. As a result, in MQTT, the MQTT broker acts as a middleman, transferring signals received from publishers to subscribers based on the message's topic.

vi.MOSQUITTO BROKER

Eclipse Mosquitto is a free software message broker (EPL/EDL authorized) that implements MQTT versions 5.0, 3.1.1, and 3.1. Mosquitto is a compact operating system that can be used on a broad variety of devices, from low-power single-board computers to full-fledged servers. The MQTT protocol uses a publish/subscribe model to provide a compact method of messaging. This makes it ideal for Internet of Things (IoT) communications, such as with low-power sensing or portable devices like phones, embedded computers, and microcontrollers. The Mosquitto project comprises of a C library for developing MQTT clients, and also the most widely used command-line MQTT clients mosquitto pub and mosquitto sub.

IV.METHODOLOGY

The circuit schematic diagram display's step by step how to interface the MQ2 with the NodeMCU ESP8266 microcontroller. The data pin of the MQ2 is connected to A0 of the NodeMCU ESP8266. The VCC and GND of the MQ2 are connected to the 3V3 supply voltage and GND respectively. The buzzer is connected to GPIO D2 and GND and finally, the Arduino sketch is uploaded into the NodeMCU ESP8266 microcontroller using the Arduino IDE. Figure 2. shows the circuit schematic diagram which was used in the implementation of the prototype.

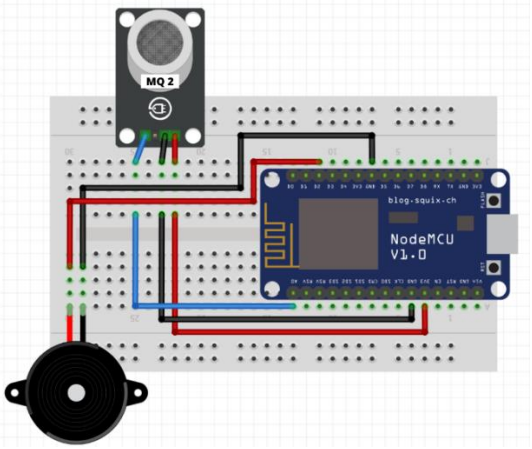

Fig. 2. Circuit Schematic Diagram

Figure 3. display's the implemented prototype of the ESP8266 smart Home of MQTT and Node-RED where MQ2 Gas sensor, Buzzer are interfaced with the NodeMCU ESP8266 to get the desired experimental result.

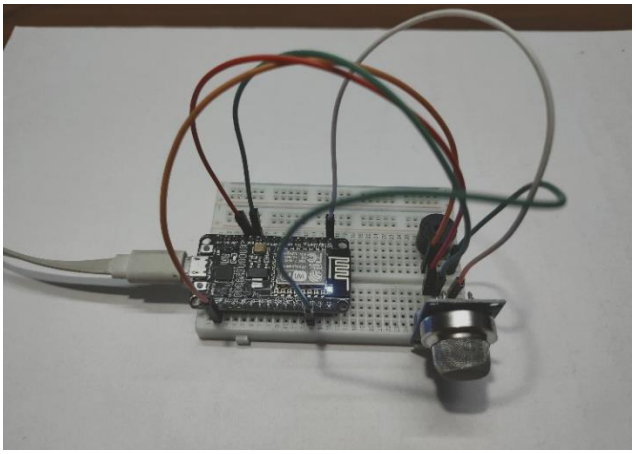

Fig. 3. Implemented Prototype

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101263

V.RESULT

Figure 4. show's the Node-RED flows wiring which enables the dashboard visualization and the control of outputs as well as publishing the sensor data to the web dashboard.

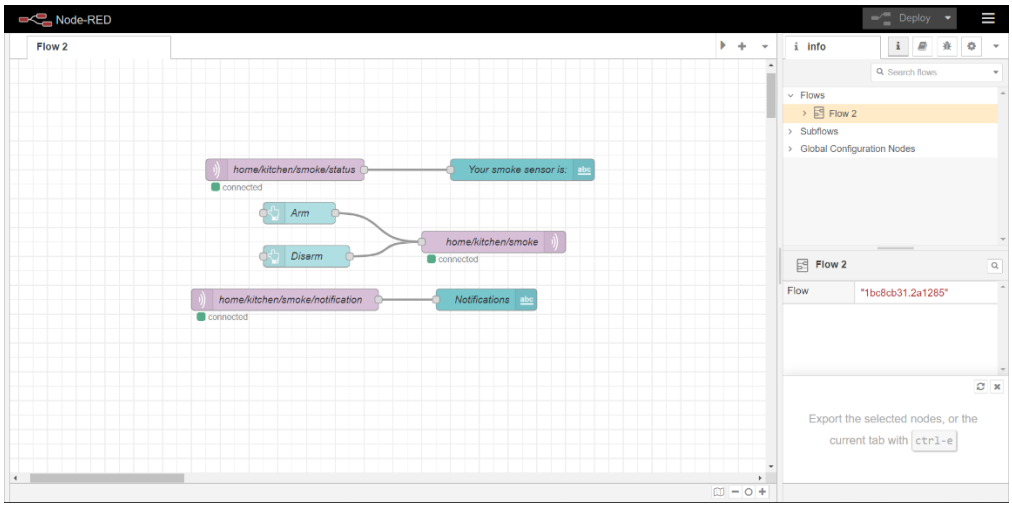

Fig. 4. Node-RED flows

The NodeMCU ESP8266 is publishing the current status of the gas sensor on the home/kitchen/smoke/notification and home/kitchen/smoke/status topic. The Node-RED dashboard that supports MQTT can be used to receive this sensor data if it subscribes to the above topics. It has also control options which can turn the sensor on and off through the topic home/kitchen/smoke. Figure 5. shows the web dashboard of Node-RED displaying sensor status and control inputs.

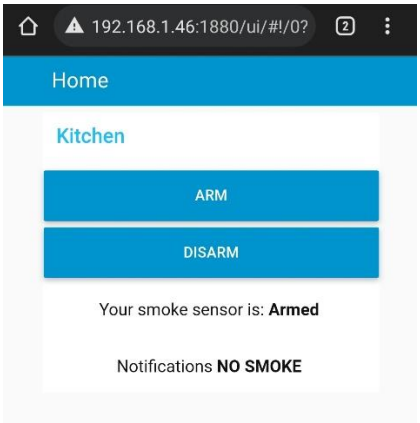

Fig. 5. Web Dashboard

VI.FUTURE SCOPE AND CONCLUSION

In our setup we presented a simple MQTT implementation example in a smart home environment using ESP8266 based development boards, Node-RED platform. The ESP8266 offers the ability to control devices like sensors or relay shields via wireless internet network. This means that devices can be used anywhere on our premises as long they are in the Wi-Fi router coverage area. The configuration used can be extended by adding sensor for AC voltage and current measurements for all the power sockets where the daily used appliances are plugged in. In addition, there can be added automation controls for powering on the air conditioning system when the temperature exceeds a certain setpoint value or to automate lighting control in house based on the ambient light. Furthermore, implementing a cloud solution will allow you to access data from everywhere, not only your personal network A system like this will evolve with the further development of technologies like ESP8266, openHAB platform and MQTT.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101263

VII.REFERENCES

[1] MQTT protocol [Online]: http://mqtt.org/faq
[2] NODE-RED (online): https://nodered.org/
[3] Mosquitto broker (online): https://mosquitto.org/
[4] MQTT pub/sub architecture [Online]: https://1sheeld.com/mqtt-protocol/
[5] Available online: https://randomnerdtutorials.com/esp8266-and-nodered-with-mqtt/
[6] Perry Lea, "Internet of Things for Architects", Packt Publishing, 2018
[7] John C. Hastings, David M. Laverty, "Modernizing wide-area grid communications for distributed energy resource applications using MQTT publish subscribe protocol", IEEE Power & Energy Society General Meeting, 16-20 July 2017
[8] Wi-Fi-based Hierarchical Wireless Networked Control Systems, Esraa A. Makled, Hassan H. Halawa, Ramez M. Daoud, Hassanein H. Amer Electronics and Communications Engineering Department American University in Cairo (AUC) Cairo, Egypt, IEEE 2015

