

IoT Based HOME AUTOMATION Using Node-RED

Ravi Kishore Kodali

Department of E.C.E,

National Institute of Technology, Warangal

WARANGAL, 506004, India

Email: ravikkodali@gmail.com

Arshiya Anjum

Department of E.C.E.

National Institute of Technology, Tadepalligudem

Tadepalligudem, 534101, Andhra Pradesh INDIA

Email: arshiyaanjum1398@gmail.com

Abstract—Internet of Things (IoT) is the most evolving technology nowadays. This advancing technology has many applications such as Home automation, Data Monitoring. Due to home automation, we are accomplishing solace in our everyday lives. Home automation more absolutely depicts homes in which nearly everything: appliances, electrical outlets, heating and cooling systems are snared to a remotely tractable framework. Home automation requires mutual communication of a large number of IoT devices. With the appreciable rise in the number of gadgets on the cloud platforms, there is a requirement for refreshing firmware very often. It includes taking out already introduced gadgets, making necessary changes in the code and flashing the altered code once again. To conquer these issues, processing of information should be possibly done elsewhere. Node-RED, which is a visual wiring tool that helps in associating gadgets easily bringing about fast and effortless connection setups. Gadgets are linked together to ESP8266 and a Mosquitto based MQTT broker using Node-RED and a connection is set up for remote monitoring and control.

Keywords— NodeMCU, Node-RED, IoT, Home Automation, MQTT

I. INTRODUCTION

Home Automation has been around for numerous decades as far as lighting and basic appliances control is concerned and only recently has technology caught up for the idea of the interconnected world, permitting full control of home from any place, to become a reality and thus build a smart home. Smart Home generally alludes to a home where the gadgets are associated with cloud. It can be viewed as a framework which utilizes smartphones, PCs to control or connect with the gadgets of home.

With advancements in home automation, you can direct and coordinate how a gadget ought to respond, when it must respond, and why it needs to respond. Your duty is to set the schedule and the rest is automated and based off of your desired penchants hence giving control and overall a smart home. Attributable to the universal accessibility of WiFi all the appliances within a home can be linked together through a common gateway and devices are controlled using MQTT protocol implemented on ESP8266 using Node-RED.

The rest of the paper is organised as follows: section II gives the related work, section III contains a brief overview

of Node-RED, section IV provides an outline of the MQTT protocol, in and section V discusses implementation details and the working. Finally, section VI concludes the work.

II. RELATED WORK

In [1] the authors have discussed various applications of Node-RED. In [2] the authors discussed regarding home automation system using MQTT and NodeRED. In [3] the authors discussed the novel prototype of home automation. It also focuses on the security alerting system which ensures safe and adaptable home automation system. It also gave an immense space to client-server interaction with user notifications. The research work done in [4] mentions automating home using a personal assistant. The work done in [6] demonstrates that MQTT is superior to HTTP for the nodes with limited assets. It has been demonstrated that information transmission through MQTT devours just around 0.05 percent of battery for every hour by utilizing wireless services like 3G. In [8] the authors mentioned home automation using Bluetooth and mobile app.

III. NODE-RED

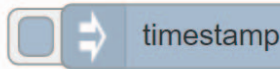
The Internet of Things (IoT) now does not just mean 'diverse things', but has developed into 'smart things' which have onboard calculation and system associations. In particular, they have the capacity to sense the environment around us and accordingly, act shrewdly. Node-RED is a programming tool for wiring together hardware devices, online services and APIs. Node-RED is open source and created by IBM Emerging Technology. Basically, it is a visual programming tool intended for the Internet of Things, but it can also be used for different applications to very rapidly assemble flows of different services. It is based on Node.js (a server-side javascripting platform). 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. Node-RED flows are stored using JSON, which can be effortlessly imported and exported for sharing with others. Nodes are gathered together in a flow keeping to achieve a predetermined objective. Likewise, flows can be coherently

assembled together to achieve higher-order objectives. Node-RED can run on a local browser at <http://localhost:1880>. Node-RED also provides a user interface module that provides a set of nodes in Node-RED to quickly create a live data dashboard. The user interface is available using local browser at <http://localhost:1880/ui>.

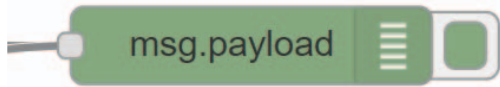
A. Create a flow

Drag and drop the nodes from node palette to add them to the flow.

1) *Add an Inject node:* The Inject node enables you to infuse messages into a stream, either by tapping the button on the hub or setting a period interim between infuses.



2) *Add a Debug node:* The Debug node causes any message to be displayed in the Debug sidebar. By default, it just displays the payload of the message, but it is possible to display the whole message object.



3) *Add a function node:* The Function node allows you to pass each message through a JavaScript function. Wire the Function node in between the Input and Output nodes

4) *Wire them together:* Connect the Inject and Debug nodes together by dragging between the yield port of one to the information port of the other.

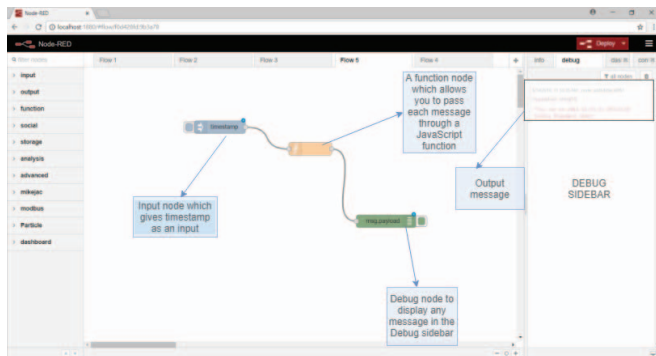


Fig. 1. Node-Red Flow 1:Time and date

B. Deploy

Now, the nodes just exist in the browser-based editor and must be deployed to the server. Click the Deploy button. With

the Debug sidebar tab chosen, click the Inject button. And the output will be displayed in the debug sidebar.

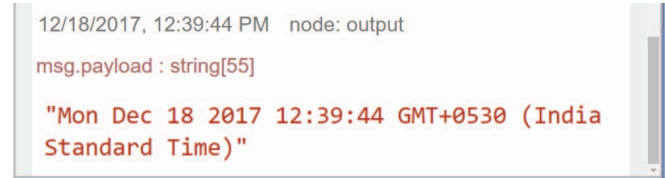


Fig. 2. Node-Red Flow 1:Output

IV. MESSAGE QUEUING TELEMETRY TRANSPORT (MQTT)

MQTT provides a lightweight messaging protocol which uses a publish/subscribe model because of which it is worthy to be used on all devices from low power boards to servers. A client can publish or subscribe to a topic or do both. Whereas as a broker receives all the messages, filters them and send them to the subscribed client. It is based on top of TCP/IP stack. So, both client and broker ought to possess a TCP/IP stack. The publish/subscribe pattern of MQTT broker allows messages to be pushed to the client devices not needing the gadgets requiring to constantly poll the server. MQTT over WebSockets can be secured with SSL.

A. Initiating, Maintaining and Terminating the connection

The MQTT connection itself is always between one client and the broker, a client is not connected to any other client directly. A client initiates the connection by sending a CONNECT packet to the broker and in response, the broker sends a CONNACK packet and a status code. Now the established connection has may terminate after some time-out. To avoid this the client passes a PINGREQ packet to which the broker reacts by sending a PINGRESP packet and keeps the connection alive. Once the connection is setup, the broker will keep it open till the client doesn't send a disconnect command or it loses the connection. And to quit the connection the client just needs to send a DISCONNECT packet to the broker. Even if the client does not send a packet that it is alive the connection will be terminated.

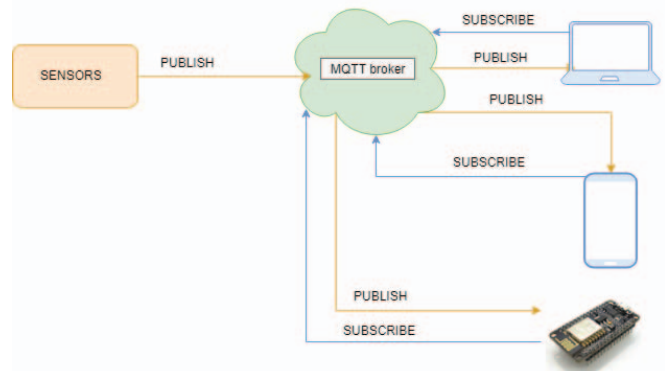


Fig. 3. Transmission of message using MQTT

B. Publish - Subscribe pattern

The Publish/Subscribe pattern is entirely different from the conventional client-server model, where the client has a direct connection with an endpoint. This means the publisher (who sends a message) and the subscriber (who receives a message) doesn't know each other and only the broker knows both the publisher and the subscriber. If in case the client wants to act as a publisher, it transmits a PUBLISH packet to the broker with all the details regarding QoS level of transmission, topic, message payload, etc. Basically, 3 levels of QoS are supported by the MQTT. If the client transmits the message with QoS 0, it will not receive any acknowledgement from the broker. And if it transmits with QoS 1, the broker acknowledges with PUBACK along with the packet identifier. Whereas in QoS 2 a total of four packets are exchanged. The broker acknowledges the client by sending a PUBREC packet on receiving the PUBLISH packet. The client then sends a PUBREL packet and followed by it the broker sends a PUBCOMP packet.

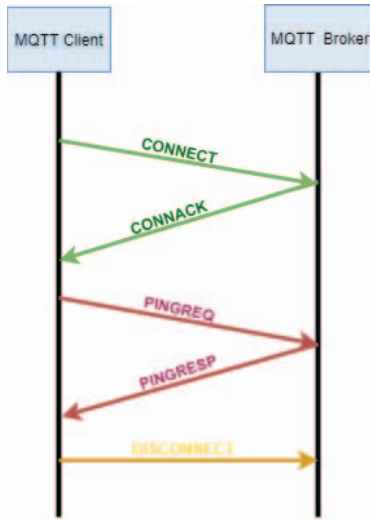


Fig. 4. Initiating, Maintaining and Terminating MQTT connection



Fig. 5. ESP8266 WiFi Module

V. IMPLEMENTATION DETAILS

A. ESP8266

ESP8266 development board, a system-on-chip (SoC) gets access to WiFi network with integrated TCP/IP protocol stack.

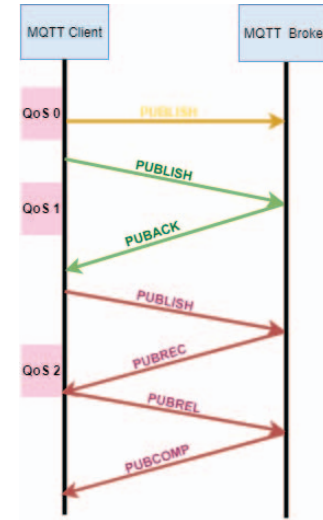


Fig. 6. Publishing messages in MQTT

It comprises a Tensilica microcontroller(32 bit) and digital peripheral interfaces and 10-bit ADC. It supports 2.4 GHz Wi-Fi (802.11 b/g/n). It has 16 GPIO, Inter-Integrated Circuit (I2C), SPI, I2S and UART interfaces.

B. Software Setup

ESP8266 must be programmed to act as MQTT client. To program the development board Arduino IDE programming software is used. In order to program the NodeMcu ESP8266 add-on in the Arduino IDE and a PubSubClient library is required which enable ESP8266 to publish/subscribe messages from the MQTT broker. The SSID and password must be included in the Arduino code for the module to connect to a router. Since, Node-red used to create flows easily by wiring hardware devices, online services..., here we use it to create a flow. MQTT nodes in Node-RED platform allow us to stay connected with the MQTT server. In this flow, two mqtt nodes are included. The mqtt input node node-red-node-mqttin connects to the mqtt broker and subscribes the messages from a specified topic and the mqtt output node node-red-node-mqttout connects to the mqtt broker and publishes messages. Node-red UI package must also be installed to control the LED. Here the button nodes (UI nodes) used for sending messages for switching off the led and switching on the led Node-red enables us to access the devices which are connected from anywhere in the world.

C. Network Setup

In this setup, we use "mosquitto" broker. It is an open source message broker. Mosquitto, an MQTT broker is lightweight and is suitable for use on all gadgets from low power boards to full servers that use MQTT protocol to publish and subscribe messages. The publish and subscribe pattern occurs as discussed in section IV.

Mosquitto is installed and runs on the same laptop on which node red is running on the port 1883. The communication

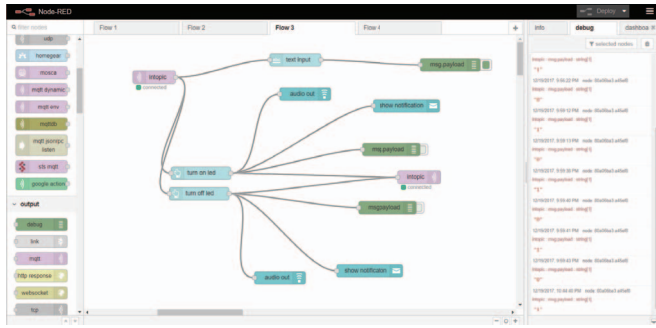


Fig. 7. Node-Red:MQTT connection with ESP8266

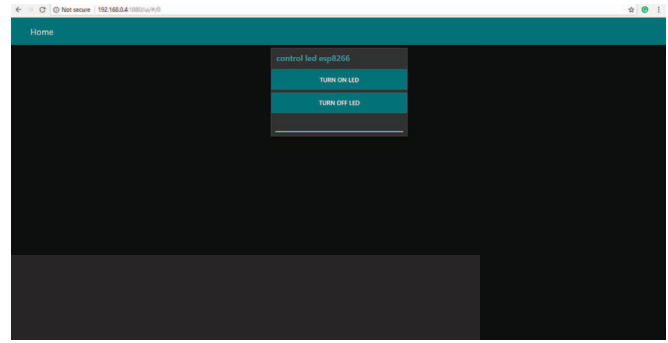


Fig. 9. Node-Red:User Interface

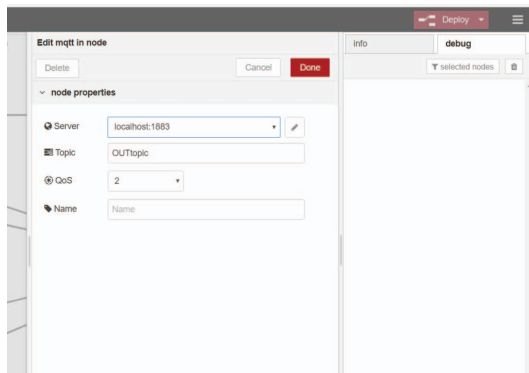


Fig. 8. Node-Red:MQTT node

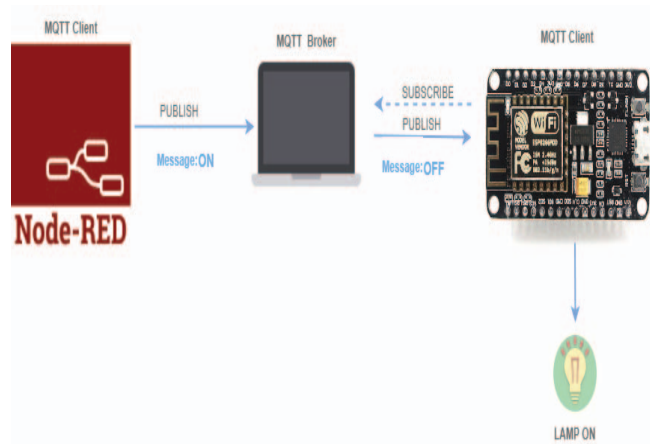


Fig. 10. Network setup

VI. CONCLUSIONS

Nowadays smart gadgets are found everywhere. Having a control on them individually is simple, however connecting them with each other is a troublesome errand. All these connected gadgets together create a smart home.

Node-Red is thus an efficient platform to link a number of IoT gadgets and can be controlled from any part of the world. In this paper, an efficacious home automation system using low-cost Wi-Fi development boards is proposed. As of now, Node-Red runs on the nearby local PC but it can also run on Raspberry Pi, Beagle Bone Black or a Linux machine. Node-Red can also be brought into a cloud platform using IBM Bluemix platform. We can also use other platforms like Intel Edison, AWS. The Internet of things will interface different gadgets in homes, industries in which the clients can control. These gadgets can likewise control mutually. This will be the future of innovation.

REFERENCES

- [1] Anoja Rajalakshmi, Hamid Shahnasser "Internet of Things using Node-Red and Alexa." International Symposium on Communications and Information Technologies (ISCIT), 2017.
- [2] Ravi Kishore Kodali, SreeRamya Soratkal "MQTT based home automation system using ESP8266", 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 2016.

between node red and ESP8266 development board is done using the MQTT protocol by publishing and subscribing messages.

MQTT nodes in the Node-RED flow publish a message to a topic and the ESP8266 development board is already subscribed to the same topic. So, the ESP8266 module receives a message from the server. After receiving the message from the server perform the required operation. MQTT performs the function of a gateway for transmitting data to the load (fan/light) through WiFi.

ESP8266 module subscribes to the server to get commands to control LED such as switching it on or switching it off. Similarly, any gadget or machine in the home can be controlled by subscribing to the MQTT server.

- [3] Brundha S.M.,Lakshmi P. and Santhanalakshmi S."Home Automation in Client-Server Approach with User Notification along with Efficient Security Alerting system" International Conference On Smart Technology for Smart Nation,2017.
- [4] Dr.V.Chayapathy,Dr.Anitha G S,Sharath B "IOT BASED HOME AUTOMATION BY USING PERSONAL ASSISTANT." International Conference On Smart Technology for Smart Nation,2017.
- [5] Hivemq. [Online]. Available: <http://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt>
- [6] A. Kumar and S. Johari, "Push notification as a business enhancement technique for e-commerce," in 2015 Third International Conference on Image Information Processing (ICIIP), Dec 2015, pp. 450–454.
- [7] Nodemcu – an open-source firmware based on esp8266 wifi-soc. [Online]. Available: http://nodemcu.com/index_en.html/
- [8] Kumar Mandula, Ramu Parupalli, CH.A.S.Murty, E.Magesh, Rutul Lunagariya "Mobile based Home Automation using Internet of Things(IoT) " International Conference on Control,Instrumentation, Communication and Computational Technologies (ICCICCT),2015.
- [9] Eclipse. Mosquitto an open source mqtt v3.1/v3.1.1 broker. [Online]. Available: <http://mosquitto.org/>
- [10] ESP8266 and Node-RED with MQTT. [online] Available: <https://randomnerdtutorials.com/esp8266-and-node-red-with-mqtt/>
- [11] Node-Red :Connect to an MQTT Broker. [online]. Available: <https://cookbook.nodered.org/mqtt/connect-to-broker>