IoT based Smart Home Applications using Cloud based MQTT Protocol

Pradeep Kumar Sayannagari[1], Mr. S.P. Manikanta[2]
[1]PG Scholar: Dept. of ECE, St. Martin's Engineering College, Hyderabad
[2]Associate Professor: Dept. of ECE, St. Martin's Engineering College, Hyderabad

Abstract- MQTT is a standardized publish/subscribe messaging protocol. It was designed in 1999 for use on satellites and as such is very light-weight with low bandwidth requirements making it ideal for M2M or IoT applications. As such, it has become one of the most common protocols for those situations. When it comes to devices communicating over TCP/IP, there is no lack of protocols. The key is choosing the right one. Understanding the protocol is part of choosing it for your application. You could probably use another protocol for message brokering, but most IoT devices use MQTT as it is very light-weight protocol.

In this paper smart home automation system is being implemented by using a very lightweight protocol that is Message Queue Telemetry Transport (MQTT) which send and receive the messages in very high speed irrespective of the speed of the of the internet (IEEE 802.11 standard). Here Raspberry pi a single board computer with 1.2 GHz of processing speed and 1GB of RAM which will have inbuilt Wi-Fi and BLE. Three loads will be connected to the raspberry pi using relay driver circuit, which will be controlled from the MQTT APP in a Wide Area Network (WAN) with high security using authentication. We willable to control the appliances from any area using lightweight MQTT protocol.

Keywords - MQTT, TCP, IOT

I. INTRODUCTION

Internet of things [6] today is a famous research topic. To enhance the convenience of life, connecting most sensors and appliances can be a good solution. By using the central home server, people can use the Wi-Fi or Bluetooth connection to control the home appliances. Suppose that all the home appliances are connected to the network and already ondemand identified by the central home server, all the states of the appliances can be monitored remotely. However, not all current home appliances can be connected to the network. Most of the appliances are turn on/off based on the mechanical switch. In addition, different home appliances provide different functions and services. Hence, how to connect these different home appliances to the network for remote control becomes an important issue. Currently, the extension cord with manual switches (or sockets) is popular and generally used. In addition to the mechanical switch of individual home appliance, the manual mechanical switch can be used to enable the specific socket for home appliance using. In other words, there are two phases for appliance controlling: 1) the switch of extension power cord for power providing, 2) the switch for function activation of the appliance. If the home appliance is controlled and monitored, it means that the appliance is powered on and already connected to the network. In opposition, to save the power and reduce the cost, the appliance should be turn off if it is not used. Since most of current appliances today are not equipped the intelligent power module, to directly turn on or power on the appliance via using wire or wireless signal is too difficult. In addition to the power, different appliances provide different services. In the other hand, the corresponding function commands will be needed for each appliance. Considering the current appliances used, to query the service functions from these appliances is almost impossible. These appliances cannot reply the queries to the central home server automatically. In other words, for the home central server, to dynamically identify each home appliance for executing the specific function or service is not possible. Hence, how to identify different appliances automatically becomes an important issue. Today, Internet of Things (IOT) is proposed to make all the things connected by network. Suppose that the devices equip the Internet connection module for information exchanging based on network. All the powered devices will be treated as the network devices and exchange the data between device and controller. In addition, based on IOT concept, the control server or the management system can automatically identify each individual device. However, until now, most home appliances are not the "Home IOT" type devices. Therefore, how to establish a home IOT environment for the existed home appliances should be considered.

II. EXISTING SYSTEM

In the existing system home automation is done by sending messages through mobile and devices were controlled by the user commands.

a. PROPOSED SYSTEM

In our proposed system designing a basic home automation application on Raspberry Pi using IOT with using of MQQT protocol.

III. BLOCK DIAGRAM

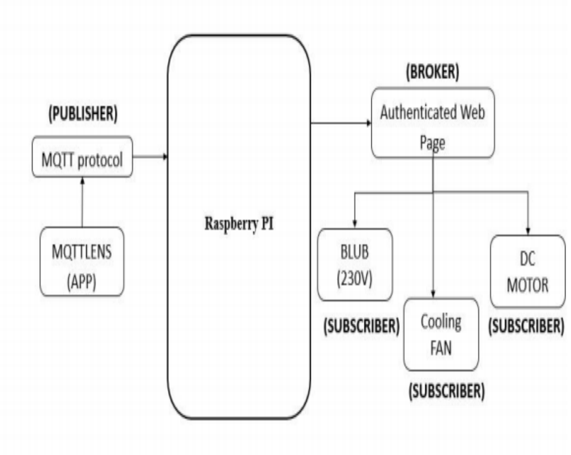


Fig.1: Block Diagram of proposed project using Raspberry pi

The above block diagram explains the hardware with input and output of the total project, Raspberry pi 3 is used which has a processing speed of 1.2 GHZ with ARMv8 BCM2837 SoC processor including in built Wi-Fi and BLE (Bluetooth Low Energy) with 1GB of RAM is used to which is connected to the BULB (230 V), Colling FAN and DC MOTOR. Now a MQTT APP will be used to which the user has to authenticate and then publish the message to subscriber using the Authenticated cloud which acts as subscriber.

a. FLOW CHART

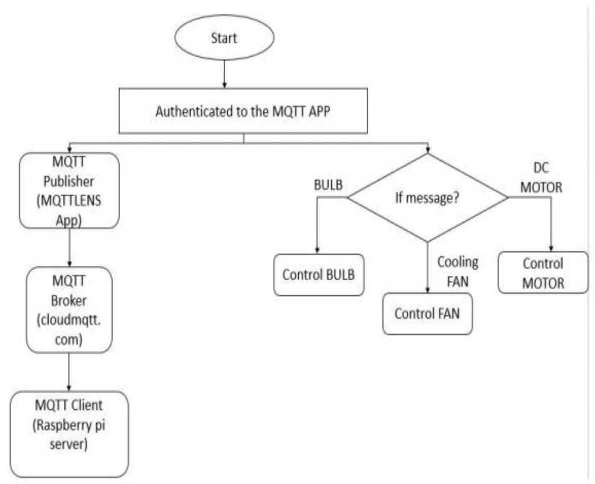

Start
Authenticated to the MQTT APP
MQTT Publisher (MQTTLENS App)
MQTT Broker (cloudmqtt.com)
MQTT Client (Raspberry pi server)
BULB
If message?
DC MOTOR
Cooling FAN
Control BULB
Control FAN
Control MOTOR

Fig.2: Flow chart of proposed project using Raspberry pi

The above flow chart represents the flow of the total program which is written in High-Level programming language i.e. PYTHON. As the program starts all the user will login to the MQTT APP and publish the message. In MQTT we have three major thing PUBLISHER, BROKER and CLIENT. In this the user who is sending the messages will act as PUBLISHER which will keep on sending the messages to the BROKER i.e. cloudmqtt.com which can be only authenticated by us using the Authentication SID and Token given and inserted in the program, now the for CLIENT raspberry pi server where all appliances are connected which can be automatically controlled using lightweight protocol MQTT.

IV. HARDWARE TOOLS

a. RASPBERRY PI ZERO W

The Raspberry Pi foundation is working on yet another model of the popular Raspberry Pi boards, as the Raspberry Pi 3 model B board. The new board looks very similar to Raspberry Pi 2 model-B , but adds on-board Wi-Fi 802.11 b/g/n (2.4GHz only) and Bluetooth 4.0. Let's play "spot the difference" with Raspberry Pi 2 at the top and Raspberry Pi 3 under. We'll 9 find the WiFi/BT chip antenna on the top left corner, and two through holes on the right of the 40-pin connectors, likely the RUN header for reset that can be found on the RPi2 where the chip antenna is now placed on RPi 3. So, the through holes are not new, they've just moved it. All connectors have the exact same placement between the two versions. Let's check out the other side of the board. The wireless module (likely Broadcom based) can be found just above the micro SD slot, and J5 connector is soldered. J5 is the JTAG connector, so it will probably not be soldered with the version that ships. The picture is not very clear, but it looks like they've used the same Elpida B8132B4PB-8D-F RAM chip (1GB) as on Raspberry Pi 2. So, although we can't be 100% certain right now, the RAM appears to be the same, and the processor is still connected to a similar USB to Ethernet chip, so they've probably kept the same architecture, expect possibly for the CPU core. So the only major changes on Raspberry Pi 3 appears to be built-in Wi-Fi and Bluetooth, and 64-bit ARM cores (likely Cortex A53).

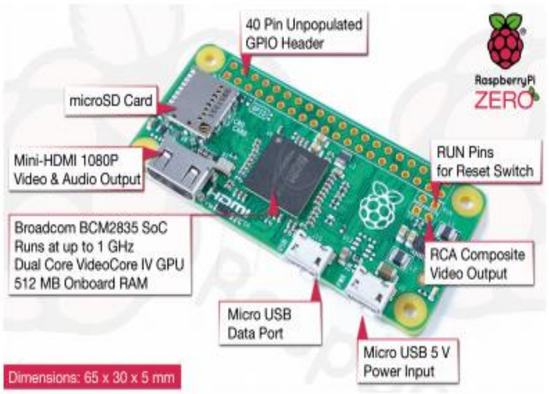

40 Pin Unpopulated GPIO Header
RaspberryPi ZERO
microSD Card
Mini-HDMI 1080P Video & Audio Output
RUN Pins for Reset Switch
Broadcom BCM2835 SoC Runs at up to 1 GHz Dual Core VideoCore IV GPU 512 MB Onboard RAM
RCA Composite Video Output
Micro USB Data Port
Micro USB 5 V Power Input
Dimensions: 65 x 30 x 5 mm

Fig.3: Raspberry pi zero w

b. DC MOTOR

Fig.4: - DC Motor

DC engines are arranged in numerous sorts and sizes, including brush less, servo, and apparatus engine composes. An engine comprises of a rotor and a changeless attractive field stator. The attractive field is kept up utilizing either changeless magnets or electromagnetic windings. DC engines are most regularly utilized in factor speed and torque. Movement and controls cover an extensive variety of parts that somehow are utilized to produce as well as control movement. Regions inside this class incorporate direction and bushings, grips and brakes, controls and drives, drive parts, encoders and resolves, Integrated movement control, restrict switches, straight actuators, straight and rotating movement segments, straight position detecting, motors(both AC and DC engines), introduction position detecting, pneumatics and pneumatic segments, situating stages, slides and aides, control transmission(mechanical),seals, slip rings, solenoids, springs.

V. SOFTWARE TOOLS

a. Linux

Linux is a free open source working framework and it has a place with the Unix working frameworks. In reality Linux implies the piece itself which is the core of the working framework and handles the correspondence between the client and equipment. Regularly Linux is utilized to allude to the entire Linux dispersion.

Linux appropriation is a gathering of programming in view of the Linux Kernel. It comprises of the GNU-task's parts and applications. Since Linux is an open source venture, anybody can alter and circulate it.

b. Raspbian Wheezy

Raspbian Wheezy is a free working framework in view of Debian appropriation. It is made by a little group of designers who are enthusiasts of Raspberry Pi. Raspbian is improved for the Raspberry Pi's equipment and it accompanies more than 35000 packagesand pre-incorporated programming. Raspbian is still under dynamic advancement and it intends to enhance the solidness and execution of the Debian bundles

c. Python

Python is a multi-worldview programming dialect: protest arranged programming and organized writing computer programs are completely upheld, and there are various dialect highlights which bolster practical programming and viewpoint situated programming (counting by meta programming and by enchantment strategies). Numerous different standards are bolstered utilizing expansions, including configuration by contract and rationale programming.

d. MQTT PROTOCOL

An MQTT system consists of clients communicating with a server, often called a "broker". A client may be either a publisher of information or a subscriber. Each client can connect to the broker.

Information is organized in a hierarchy of topics. When a publisher has a new item of data to distribute, it sends a control message with the data to the connected broker. The broker then distributes the information to any clients that have subscribed to that topic. The publisher does not need to have any data on the number or locations of subscribers, and subscribers in turn do not have to be configured with any data about the publishers.

If a broker receives a topic for which there are no current subscribers, it will discard the topic unless the publisher indicates that the topic is to be retained. This allows new subscribers to a topic to receive the most current value rather than waiting for the next update from a publisher.

When a publishing client first connects to the broker, it can set up a default message to be sent to subscribers if the broker detects that the publishing client has unexpectedly disconnected from the broker.

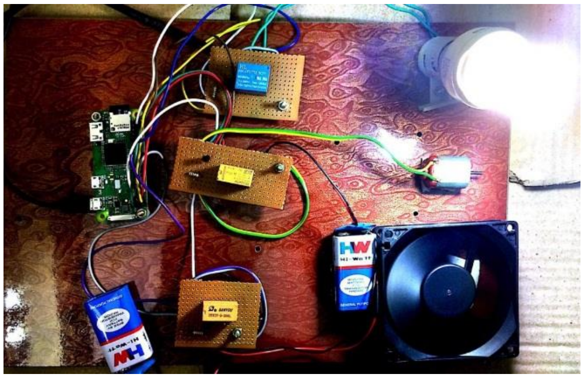
Clients only interact with a broker, but a system may contain several broker servers that exchange data based on their current subscribers' topics.

A minimal MQTT control message can be as little as two bytes of data. A control message can carry nearly 256 megabytes of data if needed. There are fourteen defined message types used to connect and disconnect a client from a broker, to publish data, to acknowledge receipt of data, and to supervise the connection between client and server.

MQTT relies on the TCP protocol for data transmission. A variant, MQTT-SN, is used over other transports such as UDP or Bluetooth.

MQTT sends connection credentials in plain text format and does not include any measures for security or authentication. This can be provided by the underlying TCP transport using measures to protect the integrity of transferred information from interception or duplication.

VI. RESULT



VII. CONCLUSION

In this paper we have implemented a smart cloud switch using light weight protocol i.e. Message Queue Telemetry Transport (MQTT) MQTT for smart home automation system that is which send and receive the messages in very high speed irrespective of the speed of the of the internet (IEEE 802.11 standard). Here Raspberry pi a single board computer with 1.2 GHz of processing speed and 1GB of RAM which will have inbuilt Wi-Fi and BLE. Three loads will be connected to the raspberry pi using relay driver circuit, which will be controlled from the MQTT APP in a Wide Area Network (WAN) with high security using authentication. We will able to control the appliances from any area using lightweight MQTT protocol.

VIII. REFERENCES

[1]. S, J., K. K., “Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security,” Recommendations of the National Institute of Standards and Technology, 2006

[2]. M, S. A, “Secure security model implementation for security services and related attacks base on end-to-end, application layer and data link layer security,” Proceedings of the 7th International Conference, 2013

[3]. S.A et al., "A New Cellular Architecture for Information Retrieval from Sensor Networks through Embedded Service and Security Protocols." Sensors 16, no. 6 (2016): 821.

[4]. SCADA system, https://en.wikipedia.org/wiki/SCADA

[5]. J.G., J.L, “SCADA communication and security issues. Security and Communication Networks”, 2013

[6]. Internet of Things, https://en.wikipedia.org/wiki/IoT

[7]. IoT and SCADA: Complimentary technologies for Industry 4.0.

[8]. Lopez Research LLC, “Building Smarter Manufacturing with The IoT,”2014

[9]. R., P., J. L, "Securing the Internet of Things, in Computer, vol. 44, no.9, pp. 51-58, Sept. 2011.

[10]. ICON LABS, “Secure the Internet of Things”,http://www.iconlabs.com/prod/internetsecure-things

[11]. R.M et al., "Security in the Industrial Internet of Things," 2016

Author Profile

Mr.Pradeep Kumar Sayannagari was born on January 10 1994,completed his graduation in Electronics and Communications Engineering from Institute of Aeronautical Engineering. Currently he is pursuing his M.Tech in Embedded systems from St. Martins Engineering College. His areas of interest include Home automation, Wireless Communications and Embedded Systems.

Prof.Mr.S.P.Manikanta working as Associate Professor in the Dept. of Electronics and Communication Engineering at St.Martin's Engineering College. He received his Bachelor's degree from Campus College of Engineering, Andhra University. He received his Master's degree from Acharya Nagarjuna University. He has about 12 years of teaching experience. He is a co-author of many International Conference and Journal Publications.

.