International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. **06** Issue **02** | **2022**

Home Automation Using NodeMCU

Asst. Prof. P.V.Sarode[1], Rukhtaj Naushad Ali Ansari [2], Payal Bhimrao Kale [3], Shubham DadaraoAdhau [4], Anand Raju Kale[5]

[1] Asst.Prof.,Department of Electrical Engineering, Manav School of Engineering, Maharashtra, India[2,3,4,5] Student, Department of Electrical Engineering, Manav School of Engineering, Maharashtra, India

ABSTRACT

This project aims at a disparate approach to tackle the notion of home automation. The concept of Internet of Things has been embraced in order to tackle automation problem. The goal is to device a method to control the day-to-day household appliances in a way that is simplistic, comfortable, and easy to practice and understand by generic individuals. The availability of low-cost microprocessors has been instrumental in developing this project. The salient feature of this project is granting the users complete dominion over their household appliances from anywhere in the world over the Internet. The state of the appliances can be monitoredand changed according to the users necessity.

Keyword: - Internet of Thing(IOT), microprocessor, NodeMCU

1. INTRODUCTION

1.1 Defination:

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to- computer interaction.

The Internet of Things (IoT) is the network of physical objects—devices, vehicles, buildings and other items—embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data.

1.2 NodeMCU:

Development Board: NodeMCU is an Open Source Firmware Development Board that helps to mainly build IoT projects with the help of few LUA Scripts. Usually the development board consists of a pre-loaded LUA scripting language, but it is also compatible with Arduino IDE and can be programmed in a fashion similar to that of Arduino.It is analogous to Arduino Hardware with a built in Input / Output pins, it possesses a chip with built in Wi-Fi that enables direct connectivity to internet, thereby facilitating easy control of things from online. It accelerates the Internet of Things projects by providing simplistic and easy development environment for the programmers. The Development Board is based on ESP8266 Chip, integrated GPIO (General Purpose Input Output), PWM (Pulse with Modulation), IIC (Interconnected Integrated Circuit), and ADC (Analog to Digital Converter). It supports a wide variety of libraries, including those for HTTP, UDP, MQIT, and normal GPIO controls.

ESP8266: The ESP8266 chip is the primary component of the development board, it works on 3V logic. It is the chip that is chiefly responsible for the board's ability to connect to Wi-Fi. The early models possess a self-contained System on Chip, and a full TCP/IP stack thereby providing any microcontroller connected to it, the ability to access the Wi-Fi networks. Owing to the technological developments, the recent models have microcontroller ability integrated with the Wi-Fi chip. It has a powerful on-board processing unit and the storage capability is sound enoughto allow integration with sensors and other applications via the GPIO pins. Its **features** are:

- SDIO 2.0, SPI, UART
- 32-pin QFN package
- Integrated RF switch, 24dBm PA, DCXO, and PMU
- Integrated RISC processor, on-chip memory and external memory interfaces
- Integrated MAC/baseband processors
- Quality of Service management
- I2S interface for high fidelity audio applications
- Integrated WEP, TKIP, AES, and WAPI

Specifications:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), SoftAP
- Integrated TCP/IP protocol stack
- Integrated T/R switch, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1/2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- Wake up and transmit packets in < 2ms 12 Standby power consumption of < 1.0mW
- **MQTT:** The reduction in the size of the device has placed immense constraints on the power consumption and memory availability. The yester year protocols were too heavy and complex, providing minimal optimization for the presented hardware. Therefore a new protocol with light and simple architecture was long overdue and so MQTT was framed. MQTT (Message Queuing Telemetry Transport) is a message oriented machine-to-machine (M2M) connectivity protocol.

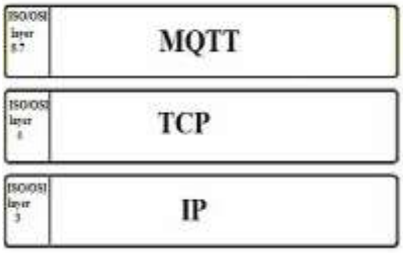

Fig -1: MQTT,TCP,IP

2. EXISTING SYSTEM:

The system that is most widely prevalent around the world is empowering the user to control the appliancesin and around him by using IR remote. Yet another variant is controlling the appliance by sending a short message from your phone by availing the SMS service provided by GSM. In some places, for controlling the appliances over a short range Bluetooth technology has been employed thereby providing access to the appliances via any smart phone equipped with such facility. Though these methods have their own distinct advantages, they are all unified by their common setback, namely their inability to access appliances over long distance. Each of the methods stated above are constrained by the range of the components and devices used in crafting the system.

3. PROPOSED SYSTEM:

3.1 Overview-

Though the system in existence is used in many places, they are limited in functionality and the range of connectivity by form of communication employed. The aim of this project is to make an automation system specifically catering to control the house hold appliances, without comprising the currently existing system's drawbacks. The concept of Internet of Things is exercised in order to create an automation system that is not bound or limited by the physical range of the device or the communication type employed.

3.2 Working-

The user accesses the Web page that is served by a typical HTTP Web Server. Depending on the need the user might toggle an appliance ON or OFF. This data is transmitted to the MQTT broker. Both the HTTP server and MQTT broker are hosted from Raspberry Pi which is in turn connected to the Internet either through wired Ethernet cable or over Wi-Fi. Since there is not a direct channel or methodology to transfer data from a Web page working in HTTP protocol, to a broker working on a different protocol namely MQTT, we transfer the data by means of a Web Socket. The clients here refer to programmable microcontroller devices with capability to connect to Wi-Fi. They are connected to the home Wi-Fi network and IP addresses are provided to them on successful connection to Wi-Fi. The clients subscribed to particular topics receive messages from the same; on receiving a specific set of messages the client is programmed to trigger certain GPIO pins high. By connecting relays to the above stated devices we can control the triggering of any appliance. In short, we are providing IP addresses to the appliances indirectly by channeling it through the client devices. The clients are programmed to work on MQTT protocol and can be programmed to inter- communicate amongst themselves. In order for the clients to receive message from the broker, they need to unsubscribe from the specific topics namely the appliances. The clients are programmed with QoS.

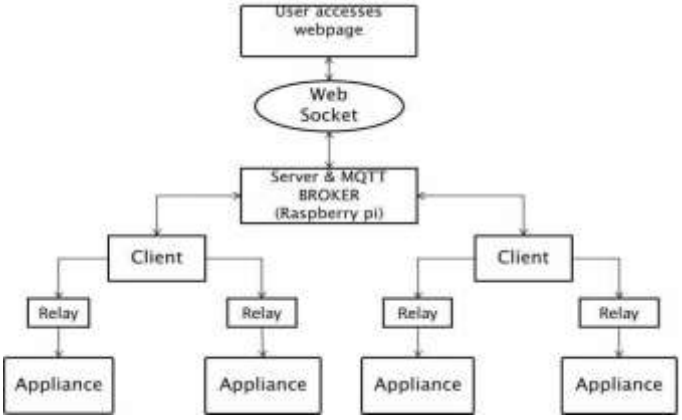

Fig -1 Block Diagram

3.3 Hardware-

Raspberry Pi: Raspberry Pi provides all the core functionality of a desktop computer at meager power consumption. The HTTP web server and the MQTT broker both are hosted using the Raspberry Pi booted with an OS that Linux based namely Raspbian. The device must be connected with the Internet in order for the server and broker to go online. Any Linux based system can be programmed to act and serve a Web page and host a broker by using proper libraries. The main advantage of choosing Raspberry Pi is that its efficiency, easy to handle and consumes low power.

NODMCU: The clients shown in the block diagram refer to the devices called as NodeMCU. It is a developmental board integrated with ESP8266 chip. Owing to the facilities of the chip, the developmental board has the ability to connect to a Wi-Fi network and is loaded with microcontroller capabilities. They can be programmed to perform a specific dedicate purpose and the program is stored in a flash memory seen on- board the chip. The board is in turn connected to the relay through one of its GPIO pins; controlling the relay's triggering mechanism.

Relay(Working)- A relay is analogous to an electronic lever that operates both electrically and mechanically; it triggers with a small current and in effect it triggers another, much larger current carrying circuit. They are used widely owing to the fact that many sensors produce only tiny electric current and it is impossible to bridge the gap incurrent to drive bigger machinery. Electromagnet is at the main component of a relay and it performs the switching mechanism. There are two basic type of relay switching namely,

- Normally Open Contact (NO) – Also called make contact. It closes the circuit when the relay is activated. It disconnects the circuit when the relay is inactive.
- Normally Closed Contact (NC) –Also called as break contact. When the relay is activated, the circuit disconnects. When the relay is deactivated, the circuit connects.

The working of a five pin relay is quite simple; when current flows through the control input terminal, the inductor coil gets energized and the electromagnet attached to the movable slider is attracted towards the coil and closes the switching circuit.

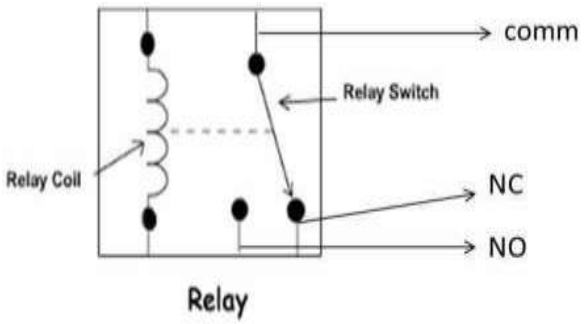

Fig 2 Basic Relay StructureProtocol:

MQTT-Clients and Servers: MQTT has a client/server model, where every sensor is a client and connects to a server, known as a broker, over TCP. Being a message oriented protocol, every message is transmitted as discrete chunks of data, and also ensuring it is opaque to the broker. Messages are published to an address, also known as a topic. Clients are able to receive the messages by subscribing to these addresses or topics, furthermore client may subscribe to more than a single topic. Every client subscribed to a topic receives every message published to that specific topic.

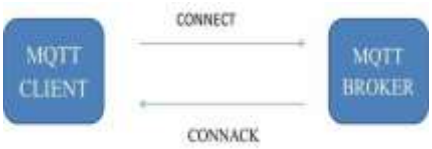

Fig 3 MQIT Cliebnt and broker link

MQTT broker: The MQTT broker forms a bridge between the various clients that are using the MQTT protocol. Every client that is publishing the data must be connected to the broker in order to establish the channel for communication. The data is transmitted amongst the clients because of the broker is able to analyze and direct the data to the respective and intended user by having track of which topic is subscribed by the individual clients. These topics basically act as the address of the clients and provide detailed information for the broker on which of the clients are the intended recipient for the transmitted message. There are several brokers available for the implementation of the MQTT protocol such as mosquito, HiveMQ, rabbitMQ, ActiveMQ, etc.

Quality of Service: Quality of service determines how each MQTT message is delivered and must be specified for every message sent through MQTT. It is essential to choose an optimal value for QoS as it is vital in determining how the server and the client communicate and transfer message amongst themselves. Three QoS for message delivery could be achieved using MQTT:

- QoS 0 (At most once) - where messages are delivered according to the best efforts of the operating environment. Message loss can occur.
- QoS 1 (At least once) - where messages are assured to arrive but duplicates can occour.
- QoS 2 (Exactly once) - where message are assured to arrive exactly once.There is a simple rule when considering performance impact of QoS. It is **"The higher the QoS, the lower the performance"**. MQTT provides flexibility to the IoT devices, to choose appropriate QoS They would need for their functional and environment requirements.

3.4 MQTT ARCHITECTURE: There are two dominant data exchange protocol architectures-

3.1 Broker based- In this architecture, Broker controls the distribution of information. It stores, forwards, filters andprioritizes public requests from the publisher client to the subscriber clients. Clients switch between publisher role and subscriber roles depending on the functionalities desired.Examples: AMQP,CoAP,MQTT,etc.

3.2 Bus based- In this architecture, clients publish messages for a specific topic which are directly delivered to the subscribers of that topic. There will not be any centralized broker or any broker based services here. Examples: DDS,REST,XMPP,ETC.

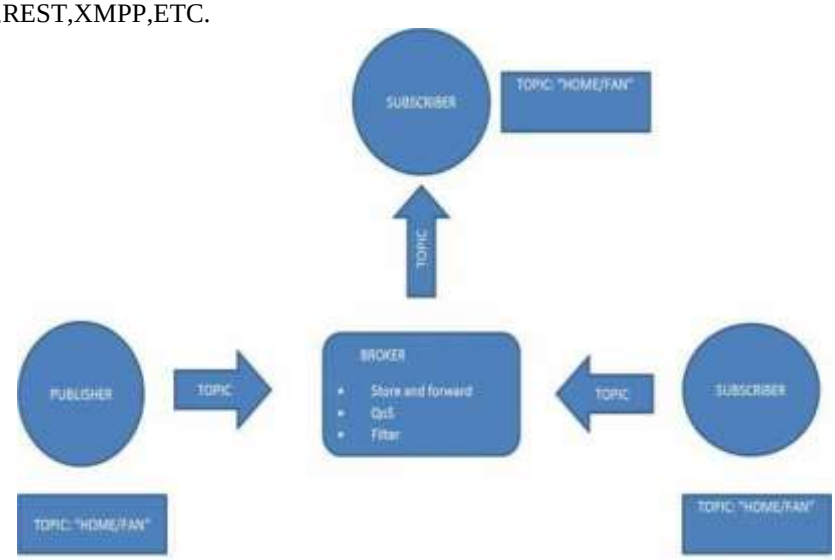

Fig 4 MQTT Architecture

3.3 MQTT MESSAGE FORMAT- MQTT messages contain a mandatory fixed-length header of 2 bytes and an optional variable length header specific for messages and message payload. In general, Optional fields complicate protocol processing. However, MQTT is optimized for bandwidth constrained and unreliable networks; typically wireless networks; so optional fields are used to reduce data transmissions as much as possible. MQTT uses networkbyte and bit ordering.

3.4WebSocket- The WebSocket protocol makes more interaction between a browser and a website possible,

facilitating the real-time data transfer from and to the server. This is made possible by providing a standardized way for the server to send content to the browser without being sought by the client, and allowing messages to

be passed back and forth while keeping the connection open. In this way a bi-directional simultaneous conversation can take place between a browser and the server. The communications are done over TCP port number 80, to avoidunnecessary blocking of the connection by the firewall.

3.5 Web browsers- WebSocket requires web applications on the server to support it and the protocol is currently supported in most major browsers including Google Chrome, Safari, Firefox. Additionally, WebSocket enables streams of messages on top of TCP. TCP alone deals with streams of bytes with no inherent concept of a message. WebSocket protocol enables two way communications without compromising security assumptions of the web. The WebSocket protocol specification defines ws and wss as two new uniform resource identifier (URI) schemes that areused for unencrypted and encrypted connections, respectively. Apart from the scheme name and fragment, the restof the URI components are defined to use URI generic syntax.

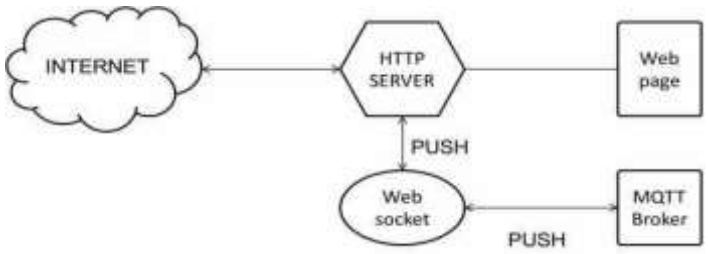

Fig 5 WebSocket interface with HTTP server

3.6 Initiating WebSocket- To establish a WebSocket connection, the client in this case the MQTT broker sends a WebSocket handshake request, for which the server returns a WebSocket handshake response. The handshake resembles HTTP so that servers can handle HTTP connections as well as WebSocket connections on the same port. Once the connection is established, communication switches to a bidirectional binary protocol that does not conform to the HTTP protocol. Before WebSocket, all communication between web clients and servers relied on HTTP. With the advent of WebSocket, dynamic data can flow freely over WebSocket connections that are persistent, full duplex and fast.

4. HTTP

4.1 What is HTTP?

The web server that is hosting the web page is a generic web server following HTTP protocol. It is an application-level stateless protocol for distributed and collaborative information systems. In this protocol, the browser initiates an HTTP request and after a request is made, the browser disconnects from the server and waits fora response. The server processes the request and re-establishes the connection with the client to send a response back; hence termed as connectionless. It is media independent and can transfer any data format as long as client and server can understand and process the data.

4.2 Software: (ARDUINO IDE)-

Introduction- The open-source Arduino Software (IDE) is used to ease the process of writing code and upload it to any developmental board compatible with this software. It can run on Windows, Mac OS X, and Linux. The Integrated Development Environment is written in Java and the execution is based on Processing projects. It contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the developmental board hardware in order to upload programs and communicate with them.

Scripting- The programs that are written in the IDE are generally referred to as sketches; they are written in the text editor region provided in the IDE; like any text editor they support all the common and basic functionalities. The scripts are generally saved with an extension of ".ino". The IDE provides provision for compiling and verifying the code before uploading the code to the hardware of the board that is used. Sketchbook is the concept used by the Arduino IDE. It is a standard place wherein the user can review his most recent scripts and access the same.

Uploading- While uploading the code to the hardware it is mandatory to ensure that proper port is selected and baudrate is accepted by the hardware. The data is generally transmitted in a serial fashion. There is a provision for visualizing the transferring and displaying the data given to the hardware by using Serial Monitor. Depending on theboard's hardware proper baud rate must be selected.

Language- Most of the boards are supported by the arduio IDE owing to the inputs from the vast community. Each board has its own specific sets of libraries that can be downloaded directly from the IDE. The language that is usedin programming the IDE is normal C/C++. Most of the standard C/C++ libraries will function in the IDE with exception of a few; this depends on the hardware specifications of the board connected and the RAM capability.

4.3 MQTT Broker – MOSQUITTO:

Introduction- The broker is the most important part of the MQTT protocol; it forms the bridge between all the clients and it is responsible for storing the data regarding the clients like topic, QoS, etc. There many message brokers that are available to be employed as broker for MQTT, mosquitto is once among them. It is available in the repository and can be downloaded directly or can be built after downloading the repository by using cmake.in caseof any Linux based system; it is also available for Windows and Mac OS X.Configuration.

4.4 Port Forwarding:

Introduction- Port forwarding is more of an executive step to be performed rather than describing it as software. Port forwarding is the process of redirecting a communication request from a specific address and port number combination to another while the packets are traversing a network gateway, such as router. This is due to the advent of Network address Translation. All the IP address are assigned dynamically using the DHCP protocol, basically the IP address of your device changes repeatedly while using Internet. The web Server that we host is a local network namely LAN, so in order for it to be accessed from anywhere in the world we need to employ port forwarding.

Implementing port forwarding- The IP address of the device hosting the web server in the LAN (also called as private IP) is bound to its IP address used for accessing the Internet. Therefore when a HTTP request is performed tothe public IP address of the server it is port forwarded by the router to the private IP address of the server, thereby forming a channel to the Internet. The same process is executed for hosting the msoquitto broker over Internet.

5. CONCLUSION:

The appliances are triggered and controlled by the user via the web page. The status of each and every oneof those interconnected appliances is monitored with ease. The transmission of data is swift, instantaneous, and mainly transmitted to every client connected to the Internet. Depending on the environment in which the client is deployed, the quality of service of the MQTT protocol can be varied to obtain the desired result. MQTT being a light weight protocol and also it works as an overhead to the pre-existing TCP/IP protocol, so there is no need to ponder over the question of compatibility as most of the Internet connections are based on TCP/IP protocol.

6. FUTURE SCOPE:

Though this methodology is unique and fairly new in origin, there is always scope for further improvement. Some of them arE,

- The power consumed by the appliances can be calculated by keeping track of the time duration the appliance is ON and knowing its power rating.
- A database can be developed on when the appliances switch ON and OFF with details like date, time, which user accessed the device, etc
- Instead of having a webpage that is accessed by button clicks, we can deploy a voice recognition system and use the output of that system to trigger appliances.

There are endless possibilities for developing a system with better functionality and making it more robust,efficient and secure.

7. REFERENCES

[1]. Xue Li ; Sch. of Comput. Sci. & Technol., Harbin Inst. of Technol., Harbin, China ; Lanshun Nie ; Shuo Chen ; Dechen Zhan, An IoT Service Framework for Smart Home: Case Study on HEM,Mobile Services (MS), IEEE International Conference, June 27 2015 Pg. 438 – 445.

[2]. Asghar, M.H. ; Sch. of Comput. Sci. & Inf. Technol., Univ. of Hyderabad, Hyderabad, India; Mohammadzadeh, N., Design and simulation of energy efficiency in node based on MQTT protocol in Internet of Things , Green Computing and Internet of Things (ICGCIoT), 2015 International Conference, 8-10 Oct. 2015: 1413
– 1417.

[3]. Seong-Min Kim ; Dept. of Multimedia Eng., Hanbat Nat. Univ., Daejeon, South Korea; Hoan-Suk Choi ; Woo- Seop Rhee, IoT home gateway for auto- configuration and management of MQTT devices, Wireless Sensors (ICWiSe), 2015 IEEE Conference, 24-26 Aug. 2015: 12 – 17.

[4]. Singh, M. ; TCS Innovation Labs., Bangalore, India ; Rajan, M.A. ; Shivraj, V.L. ; Balamuralidhar, P., Secure MQTT for Internet of Things (IoT), Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference, 4-6 April 2015 : 746 – 751.