Voice Controlled Smart Home

Indu Dokare [1], Raghav Potdar [2], Rahul Sohandani [3], Soham Phutane [4], Alish Wadhwani [5]

Department of Computer Engineering, Vivekanand Education Society's Institute of Technology, Mumbai, India [1,2,3,4,5]

Abstract

Over the past few years, the home automation industry has seen huge expansion due to the fact that home automation has become comparatively highly accessible due to ever increasing dependency on smartphones and smart devices. In Spite of this, problems in the interoperability of devices arise because of the absence of uniformly accepted protocol inside the application layer which prompts the user to be able to access and monitor multiple smart applications and appliances. During the course of this study, a voice-controlled smart home automation system was developed and implemented using the OpenHab framework that provides a platform that is focused on mixing various smart devices and technologies at the back-end. Since much of the smart devices possess differing communication protocols, the aim of this project was on the process of development of a sensible home automation solution that is modular and versatile. In addition, this system uses Google Voice Assistant for voice control which is integrated into the system via OpenHab cloud connector. RESTful protocols are used to integrate different devices with different communication protocols and used MQTT protocol to add sensors to NodeMCU that keeps a watch on the ambient conditions of the room.

Keywords: *IoT, OpenHab, MQTT Protocol, NodeMCU, Home Automation, Voice Control, Firebase, Thunkable*

1. Introduction

The development of technologies makes it possible to introduce automation in every aspect of life. We like to control the devices from a single place whether it may be in our home or out of the home in order to ensure safety, less power usage and for our comfort too. There has also been an increase in demand for controlling electronic devices through voice. This project will be achieving this with the invention of new smart home devices and home automation systems.

A Fully functioning home automation system is user friendly, covers home security, can monitor the home appliances and allows remote access through an interface.

2. Literature Review

The idea of home automation system is present from the late 1970s. New technology change has changed to face of home automation system. Internet of Things (IoT) got evolved in 2008 has changed the human-to-machine and machine-to-machine interaction.

Some existing home automation systems:

1. Bluetooth based home automation system controlled using mobile phones
2. ZigBee based home automation system using mobile application
3. GSM based home automation system using mobile phones
4. Home automation using RF modules[1]
5. Wi-Fi-based home automation system using mobile application
6. Home automation system based on Cloud
7. Home automation using Android SDK
8. Home automation system using Z-Wave based sensors

Lack of Interoperability between devices is a big concern in each of these systems. Smart Bulbs which are seen today in the market are based on ZigBee Technology (Similar to Wi-Fi). If the home automation system is Bluetooth based then the complete system has to be changed or to use a different application (vendor-specific) to use a smart bulb.

Also if the vender is different for two smart bulbs then also the complexity of a Home automation system increases.

1. Voice assistant systems
2. Google Assistant
3. Apple Siri
4. Amazon Alexa
5. Samsung Bixby
6. Microsoft Cortana

A voice assistant is a digital assistant that uses voice recognition, speech synthesis, and natural language processing (NLP) to provide a service through a particular application.[7] Google Assistant's routines are an easy way to automate your home. It is a popular voice assistant system, especially for home automation.

3. System Model

The core of this system is a central controller (server), which is the central part of an IoT system. If the central server is down the entirety of the system is down. Different devices are connected to the server. Some devices may be able to communicate with each other considering that they have the capability to do so, but it is always a better approach to avoid it and use the server as a central controller so that there is centralized control and troubleshooting in case of occurrence of an error. The working of the Voice Controlled Smart Home can be understood by Figure 1.

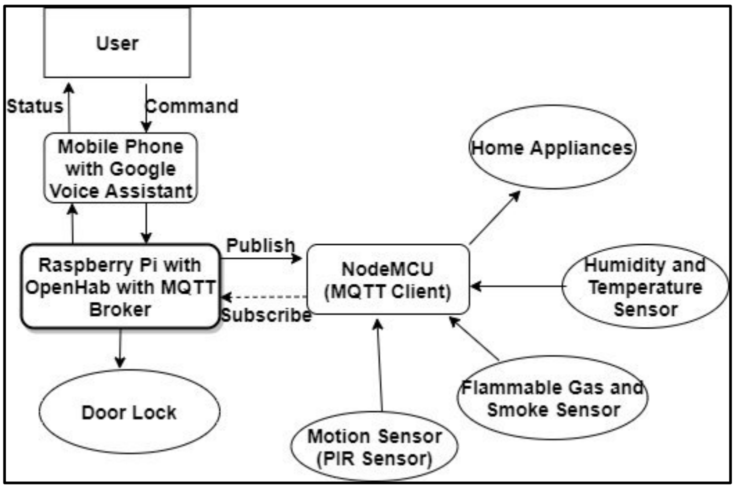

User
Status
Command
Mobile Phone with Google Voice Assistant
Home Appliances
Publish
Raspberry Pi with OpenHab with MQTT Broker
NodeMCU (MQTT Client)
Subscribe
Humidity and Temperature Sensor
Flammable Gas and Smoke Sensor
Door Lock
Motion Sensor (PIR Sensor)

Figure 1. Working of Voice Controlled Smart Home

To communicate with all other devices and integrate them with the central server, OpenHab helps us with a big deal of facilities. The OpenHab means open Home Automation Bus is a central home automation platform for the all the smart home devices at home.

4. Theoretical Background

Theoretical section is divided into six sections. The first two sections discuss OpenHab, the third section discusses Google Voice Assistant, fourth and fifth discuss RESTful protocols and MQTT protocol respectively and the sixth section discusses Thunkable-the application building tool.

4.1. OpenHab

OpenHab is based on an Eclipse SmartHome (ESH) architecture. It is a free and open source solution for home automation system. It is a flexible framework for a smart home. Eclipse SmartHome is not bound to use a single communication protocol. Instead, ESH provides support for many communication protocols. The framework is designed to run on embedded devices, such as a Raspberry Pi, a BeagleBone Black or an Intel Edison.[6] It requires a Java 8 compliant JVM and an OSGi (4.2+) framework, such as Eclipse Equinox™ [6].

4.2. Components of OpenHab framework

In the OpenHab smart home framework, things, channels, and items, bindings are the important components. OpenHab generated representation of devices are referred to as things. things provides functionalities of the devices using one or more channels. Channels can be controlled by items. An item is considered as state of a channel. Persistence can store data from OpenHab which can be retrieved at a later time.

Figure 2. Shows the components of OpenHab system. Let's consider an example of a Smart Bulb for OpenHab integration. Here, Smart Bulb is a thing having different channels. A channel is a physical functionally of a thing. Channels of a Smart Bulb are Color Channel, Color Temperature Channel (brightness). An item can be ON state, OFF state, Color wheel, Color temperature gauge, etc.

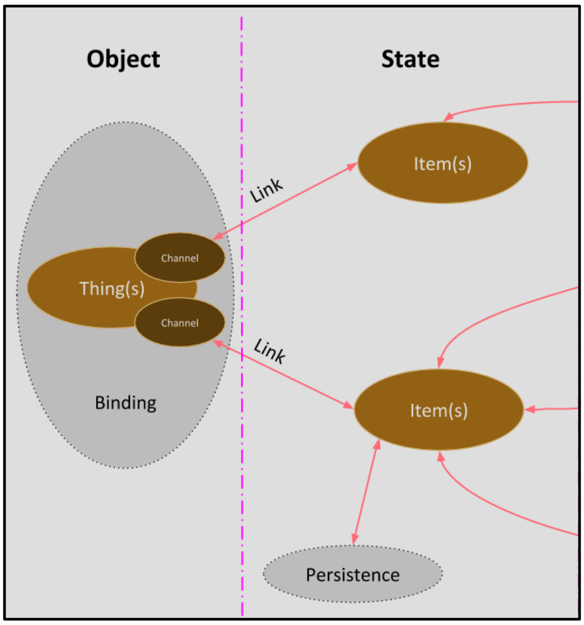


Figure 2 [4]. Components of the OpenHab framework

4.3. Google Voice Assistant

Google Assistant is an artificial intelligence-powered virtual assistant developed by Google that is primarily available on mobile and smart home devices.[5] Google assistant is capable of providing two-way conversation very effectively.

Users can interact with Google assistant via natural language. User can also provide commands via keyboard input. The Google Assistant gives permission to the users to activate and make modification to the vocal shortcut commands for performing actions on their device or configuring it as a hub for home automation. The feature of speech recognition is primarily available in English, among other languages.

4.4. REST API

REST and API are two different things. API is an Application Programming Interface. It is used to communicate between two software components. API contains set of instructions for data transmission from one software product to another. REST (Representational State Transfer) is used for creating web services by defining some set of constraints. It is a set of rules that developers follow when they create their API.[8]

REST API is used to get a piece of data called as a response from a specific URL (link) called as a request. Request consist of four important things i.e. endpoint, method, headers, data (body). Body can be send request and also body can accept data from response.

REST API can work with four method of a web application service which includes GET, POST, PUT and DELETE. Many IoT appliances uses GET and POST HTTP requests to change their functionalities. REST API can also be used to authenticate the client devices.

4.5. Message Query Telemetry Transport Protocol (MQTT)

MQTT (Message Query Telemetry Protocol) is a lightweight, publish-subscribe network protocol that transport messages between devices.[3] There are two important entities in MQTT i.e. MQTT broker and some clients. Each client sends message to broker and broker routes message to appropriate client. In MQTT, data point is considered as topic. Client has to subscribe specific topic for message conversation. There is no direct client to client communication. MQTT broker can be setup on local machine or there are some public broker like hiveMQ for testing purposes.

The use of MQTT is very ideal for IoT application because of its power efficiency.[2] Using this protocol, the devices are opened to connection, maintains its status using little power, and delivers messages with small transport overhead.[2] MQTT protocol is very useful for IoT devices which runs on quite low power.

4.6. Thunkable

Thunkable is a platform to create android apps. Here anyone can build native mobile apps quite easily. This platform allows user to drag and drop visual objects from inventory to create application. Thunkable also provide a live test feature. Recently Thunkable classic got updated to ThunkableX which can create apps for android as well as IOS.

5. Methodology

This section is divided into six sections: (1) list of hardware and software components used, (2) Implementation of smart home system, (3) smart home system architecture, and (4) implementation of MQTT based sensors. (5) Implementation of Google Voice Assistant. (6) Development of Mobile application using Thunkable.

5.1. List of Hardware and Software tools utilized

The Raspberry Pi for the central server as OpenHab will be installed on it. Also MQTT broker will be added via OpenHab bindings. Central server uses Raspberry Pi 3 Model B+. It’s very important that Raspberry Pi should always get a stable and uninterrupted power supply. Also it should be always connected to the home network.

Lights, Fans, Temperature and Humidity Sensors, MQ sensors for fire safety are integrated with OpenHab using MQTT protocol. Also vendor provided smart devices can also be integrated with OpenHab via their in-built binding e.g. Philips Hue Bulb, Google Chromecast for TV, Smart AC's, etc.

5.2. Implementation of Smart Home System

To set up the OpenHab home automation system, first the openhabian image provided officially by OpenHab should be installed into Raspberry Pi which contains the OpenHab server. BalenaEtcher is used to flash openhabian image to the SD card of Raspberry Pi.

After turning ON the Raspberry Pi, it takes around 10-20 min to get set up. Then Putty or WinSCP can be used to access the OpenHab server remotely. Firstly, OpenHab needs to be updated. After the initial configurations were done, the configuration files that can be created and modified using Visual Studio Code Extension for OpenHab. Code for the components of OpenHab such as things, channels, and items have been defined and changed for all smart devices which have to be integrated. With the communication between the smart devices and the server has been established, things can be controlled or monitored from PaperUI. This interface is developed and provided by OpenHab known as HABPanel. HABPanel is a user-friendly interface, especially a good touchscreen experience.

5.3. Smart Home System Architecture

System architecture contains three parts: OpenHab Server, MQTT appliances, Google Voice assistant binding provided by OpenHab.

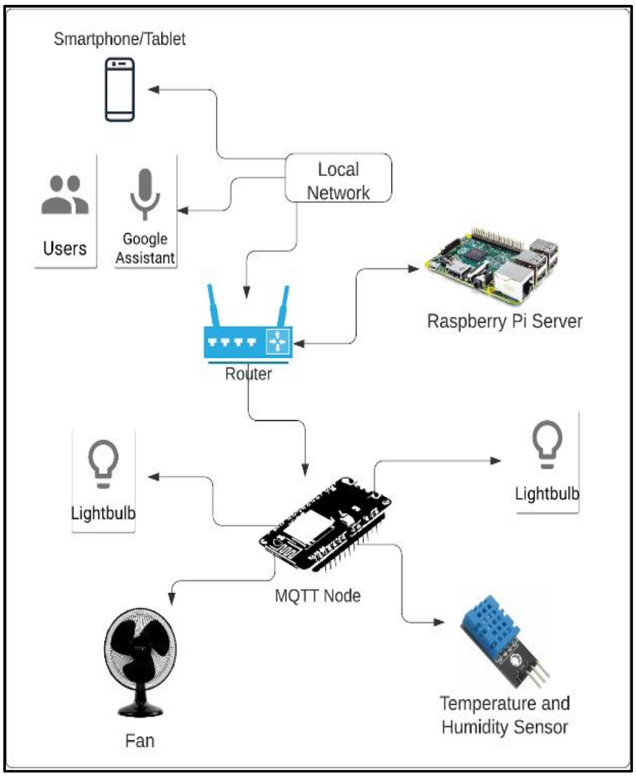


Figure 3. Smart Home System Architecture

The first part is OpenHab server on Raspberry Pi which allows the user to provide input and monitor the connected devices through a smartphone application and through Google assistant. The second part is MQTT. All the MQTT appliances has to be integrated with OpenHab server. The cloud connector in OpenHab system allows the user to access and monitor devices in real-time over a remote network. The third part is Google Voice Assistant (GVA). To use GVA, first myopenhab.org account should be setup. Add GVA keywords for OpenHab in item file of OpenHab. Then link myopenhab account with google home to control appliances through voice.

5.4. Implementation of MQTT

MQTT broker can be installed and configured in OpenHab server using in-built binding for MQTT broker. NodeMCU is a MQTT client. Temperature and Humidity sensor values are published to their topics via MQTT broker. MQTT broker send these value to OpenHab server and OpenHab displays it on UI. Light and Fan topics are subscribed to their topics. OpenHab Publishes ON/OFF values to these topics to control these appliances.

All the smart devices are based on MQTT protocol like Light, Fan and also it is used for monitoring Temperature and Humidity. Their topics are subscribed to Raspberry Pi based broker.

5.5. Implementation of Google Voice assistant

The OpenHab Cloud Connector allows connecting the local OpenHab runtime to a remote OpenHab Cloud instance, such as myopenHAB.org, which is an instance of the OpenHab Cloud service hosted by the OpenHab Foundation.[4] The OpenHab Cloud service (and thus the connector to it) is useful for different use cases[4]:

- It allows remote access to local OpenHab instances without having to expose ports to the Internet or to require a complex VPN setup.[4]
- This cloud instance can also be used for services like IFTTT, Smart locks like August, etc.

Google assistant is a virtual personal assistant provided by Google. Before that Google Now was used as assistant.

To control home appliances through Google assistant. Firstly, Home appliances features declared in items file have to be exposed via metadata. Currently some supported metadata values are Switch / Dimmer / Color { ga="Light" }, Rollershutter { ga="Curtain" }, etc. Then Link OpenHab with Google Home to control devices by logging in myopenhab.org. Then allot devices according to their specific rooms.

5.6. Development of Mobile Application using Thunkable

Mobile Application design through Thunkable consists of two parts: 1. Design section 2. Block Section.

Design section covers the front end or the User Interface of the application. It contains options to add visible (components like textbox, buttons, labels, switches, etc.) and invisible (components like APIs) to the application. It provides multiple options and customizations to make the User interface more appealing.

Block Section covers the backend or control section of the application. It provides various options to define operations like control statements, logical statements, events, objects and functions at the backend of the mobile application.

User Database management has been integrated through the Google Firebase. Google Firebase also provides real time database facility.

6. Results and Discussion

This section discusses the important features of the voice-controlled Smart Home system. To be specific, these include the following parts:

(1) Communication protocol interpretation

(2) User Interface of the system

(3) Voice Control through the Google Assistant

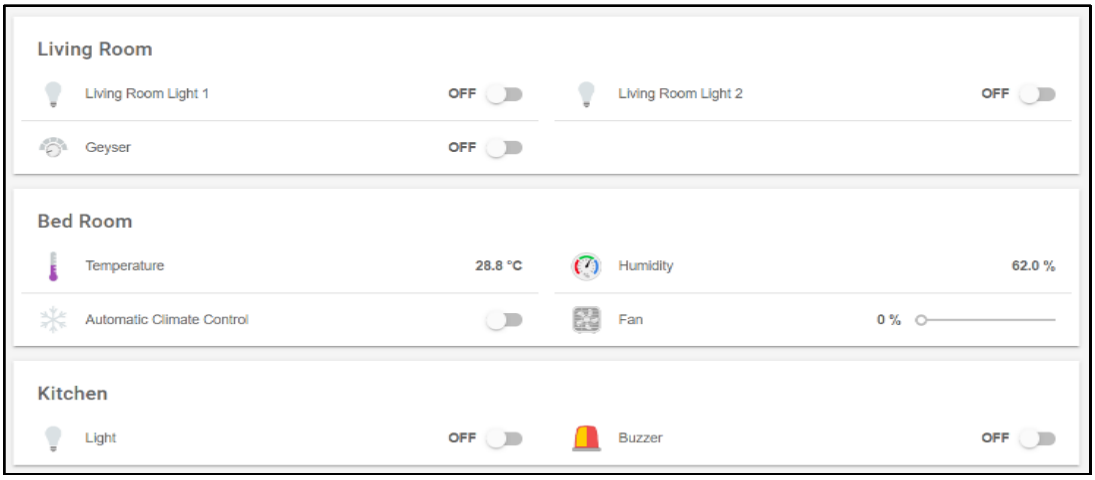
6.1. Communication Protocol Interpretation

The central server architecture implemented using the OpenHab framework. The architecture consists of physical and the virtual layer which enables the translation and integration of hardware and software components. Appliances in the system are connected to OpenHab server using MQTT protocol. Sensors uses SPI communication protocol between embedded board and sensors.

6.2. User Interface of the system

There are two parts to the user interface of smart home system: (1) OpenHab basic UI (2) Mobile Application available to the user.

The user interface developed using Basic UI is shown in the below Figure 4.



Living Room

Living Room Light 1 OFF

Living Room Light 2 OFF

Geyser OFF

Bed Room

Temperature 28.8 °C

Humidity 62.0 %

Automatic Climate Control

Fan 0 %

Kitchen

Light OFF

Buzzer OFF

Figure 4. User Interface using OpenHab BasicUI

The user interface developed in the mobile application which is available to the end-user is shown in the figures 5, 6.

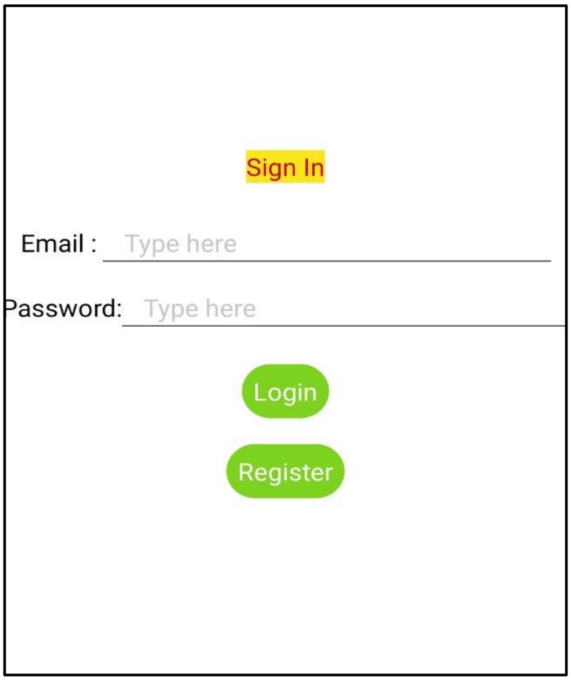
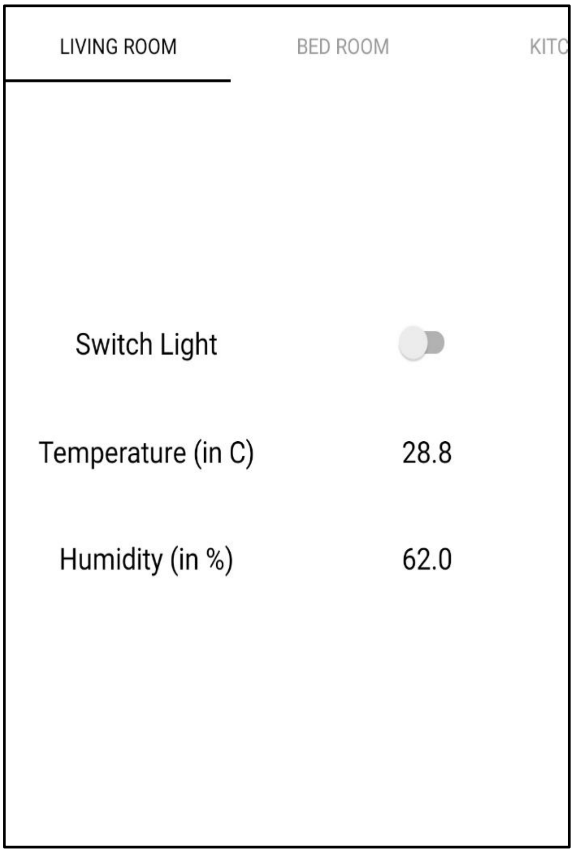


Figure 5. User Interface - Login in Mobile Application



Switch Light

Temperature (in C) 28.8

Humidity (in %) 62.0

Figure 6. User Interface Mobile Application example

The above screenshots are general examples considered for demonstration.

ISSN: 2233-7857 IJFGCN

6.3. Voice control through Google Assistant

Voice Commands, integrated through OpenHab cloud connector, now can be used to operate the appliances from different rooms.

7. Conclusion

The study was successfully able to build and deploy a voice-controlled smart home system that is dependent on the OpenHab Framework with the following main features: The system can be deployed in houses. The system has the ability to effectively use the wireless network system that is already set up in place within the house. Its architecture is suitable to make changes to provide facilities for multiple smart home appliances. The system has a central controlling device or a central server that is responsible control and monitor all the smart home appliances. The system does the integration of all home appliances into a single user interface to be able to monitor and control appliances with ease.Additionally Google Voice Assistant allows this system to operate the appliances through human speech.

Smart Devices are integrated Message Queuing Telemetry Transfer Protocol for communication between NodeMCU and the central server.

This prototype is currently working on low powered appliances. It is open to further development for the usage of Smart appliances like Smart TV and Smart AC. Further enhancements in security like cameras can be integrated into this system.

References

8.1. Journal Article

[1] Noorul Hamitha B, Manojkumar R, Jeya Prakash S, Yogesh D, Bala Murugan VJ “Wireless Home Automation System Using OpenHAB” International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 04 **(2018)**

8.2. Conference Proceedings

[2] Raymark C. Parocha, Erees Queen B. Macabebe. "Implementation of Home Automation System Using OpenHAB Framework for Heterogeneous IoT Devices", IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), **(2019)**

8.3. Internet Sources

[3] MQTT from wikipedia. https://en.wikipedia.org/wiki/MQTT

[4] Openhab https://www.openhab.org/docs/

[5] Google Assistant's routines are an easy way to automate your home https://www.engadget.com/2018-04-23-google-routines-assistant-smart-home-automation.html

[6] Eclipse SmartHome https://www.eclipse.org/smarthome/

[7] Voice Assistants https://www.smartsheet.com/voice-assistants-artificial-intelligence

[8] Understanding And Using REST APIs https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/