iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VIII Month of publication: August 2021

DOI: https://doi.org/10.22214/ijraset.2021.37411

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Voice Enabled Home Automation System

Akshad Patel
Department of Electronics and Communication Engineering, VIT University, Vellore, India

Abstract: In this era of automation innovation, life is getting less demanding, simpler and the world is getting smarter. IoT gives us the power of internet, data processing and analytics to handle or control the real world. All the day to day entities can take instructions or data from the internet to function without human intervention. Home automation is the automatic control of the electronic home appliances i.e. these appliances along with being controlled manually by switches, can also be controlled through mobiles, laptops and human voice. To accomplish this, AWS IoT core has been used that allows connection of multiple devices on internet without the need of servers. Along with AWS, ESP8266 module is used which helps connect all appliances to the internet along with all the other sensors. Python programming language is used to enable voice control.

Keywords: IoT, esp8266, AWS, AWS IoT core, Python, Home automation, voice control

I. INTRODUCTION

Home automation systems using IoT make use of internet protocols and cloud services. These systems usually require a physical or a remote server which helps one manage the flow of data, processing of data and device connections in the network. This job of servers can sometimes be handled by IoT platforms like AWS IoT core, Firebase, ThingWorx, which help in connection of multiple clients or ‘things’.

A home automation system involves switching on and off of appliances like fans, light or even air-conditioners when the temperature limit has reached. Along with this, a home automation system can sometimes involve intrusion detection i.e. burglar alert. It can be designed in a way to alert either the owner of the house or an emergency contact as soon as a trespasser is detected. Instead of calling it as a home automation system, it can even be called a smart home automation system. The plants can be properly watered, water tank can be checked with just one command along with a lot of other functionalities.

Home automation is not considered to be a luxury these days, in fact, it is considered to be a need. A large number of homes, public complexes, industries or even small scale plants across the globe are adopting IoT based machines and automated systems. It becomes easier for someone to control everything by just speaking out a command instead of a large manpower to control each node in the system.

The designed system consists of a hub that controls the flow of commands and the functioning of the sensors and the devices connected to it. This hub interacts with the servers (remote or physical) or the IoT platform through the internet.

Communication through the internet needs a communication protocol i.e. a set of rules that define how the hub sends and receives data from the cloud or the server. Communication protocol consists of 4 layers – (i) Application layer (HTTPS, MQTT, CoAP, etc.); (ii) Transport layer (TCP, UDP); (iii) Network layer (IPv4, IPv6); (iv) Link layer (Ethernet, Wi-Fi, LR-WPAN, etc.) The hub connects and interacts to the cloud or an IoT platform through these protocols. Most commonly used Application protocol is HTTPS.

The applications or software to help the user control the appliances, can be programmed using multiple languages. A lot of SDKs (Software development kits) are available for Python, Java, NodeJS, for app development languages like android, IOS, dart, etc. With the help of these SDKs, the applications can then connect to the hub of the system and then send or receive data accordingly.

II. DESIGN AND SPECIFICATIONS

This project contains a lot of technical components put together to establish the automation functionality. This technological stack is described or divided into multiple layers similar to any IoT model.

1) *Physical Layer:* This layer consists of all the hardware components connected within the walls of the house. This layer needs to have a hub as mentioned above which connects all the other sensors or the appliances in the system. This hub enables communication among the devices and also with the user through a network which is internet in this case. In the proposed system, ESP8266 has been used as a hub. The sensors used include DHT11 (Temperature sensor), Moisture sensor, and 5V power relay modules.

2) *Communication Layer:* This layer of an IoT system handles the communication part of the system. A particular communication protocol is selected according to the factors like security and speed. In the proposed system, Message Queue Telemetry Transport (MQTT) protocol is used. It is light-weight messaging protocol that uses publish – subscribe model. MQTT uses a client-server architecture where the client (an IoT device) connects to the server (also called an MQTT broker) as shown in fig 1. This broker can be any physical server or any application or platform and publishes a message to the client subscribed topics. This protocol is well suited for systems with limited processing ability and where the network is unreliable.

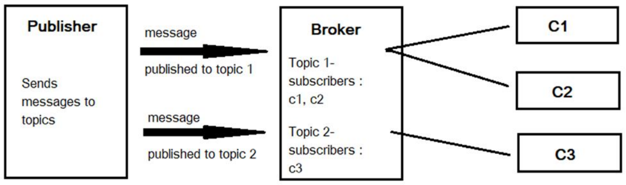


Fig. 1: MQTT communication model

3) *Application Layer:* This layer provides an interface that a user can use to interact with the physical layer. There are various other aspects such as live status, sensor readings and verifications of the system that need to be monitored from time to time. This can be done using the applications. In the proposed system, Amazon Web Services (AWS) IoT Core has been used which helps in controlling the hub as well as all the other components connected to it without a server. Python programming language has been used to enable the voice control feature. A software coded in python will act as the user interface for the proposed system.

4) *Management Layer:* This layer provides functions to monitor the system. This is an important layer if the system involves continuous analysis of the data transferred or received. This layer also involves data management i.e. using a database which can be an SQL or a Non-SQL to make the system efficient. In the proposed system, DynamoDB of AWS is used for management of the statuses of the appliances.

III. PROPOSED SYSTEM

This project aims on developing not just a home automation system, but also a smart home automation i.e. one that can sense its surroundings. To interact with the environment, specific sensors are used. The architecture of the project is shown in fig 2.

A. *Hardware Components*

The sensors and modules as mentioned earlier, are controlled by ESP8266. The hardware components used the in the system include -

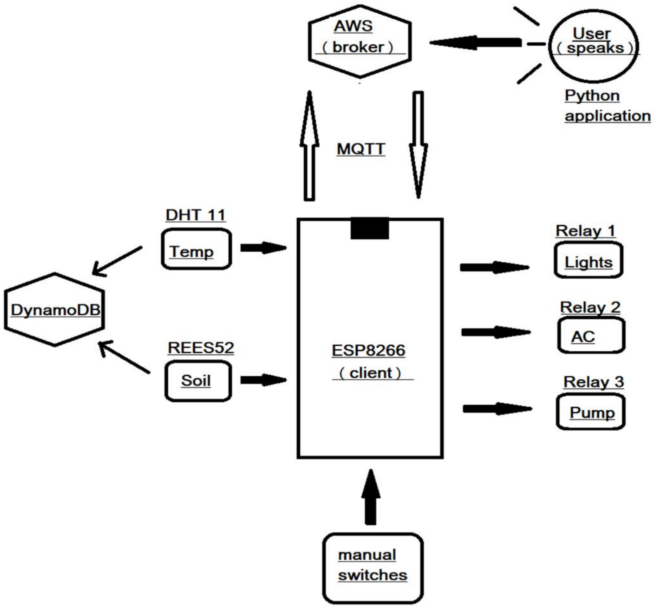


Fig.2. Proposed system-block diagram

1) *ESP8266:* The ESP8266 is an IoT board or a System on a Chip (SoC), manufactured by the Chinese company Espressif. It consists of the Tensilica L106 32-bit micro controller unit (MCU) embedded along with a Wi-Fi transceiver. It has 11 GPIO pins (General Purpose I/O pins), and also an analog input. It can be programmed in a way similar to that of Arduino. Arduino IDE is used here to write and upload the code onto ESP8266.

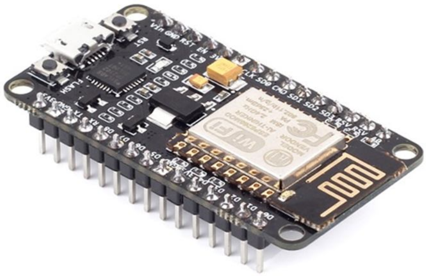


Fig.3. ESP8266 module

2) *DHT11:* The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity module and a thermistor to measure the surrounding environment, and gives out a digital signal on the data pin. This data pin is a digital pin and not an analog pin. The only real drawback of this sensor is you can only get new data from it once every 2 seconds. KY013 module can be used instead of DHT11 if time and speed is an important factor and if these quantities are changing constantly.

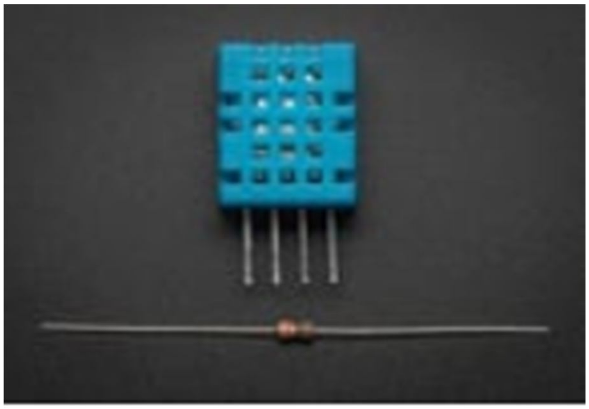


Fig.4. DHT11 module

3) *REES52 – Soil moisture Sensor:* This is a digital soil moisture sensor. It gives out a digital output of 5V when the moisture level is high and 0V when the moisture level is low in the soil. The sensing element conjointly includes a potentiometer to line the required moisture threshold. This sensor can also be connected to an analog pin to get an analog value of the moisture. This value when mapped according to the required unit using basic calculations, is very helpful in determining the precise value of moisture in the soil.

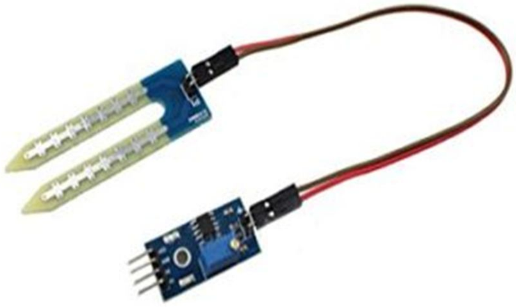


Fig.5. Soil moisture sensor

4) *5V Power Relay Module:* Relay is an electromechanical module that makes use of an electric current to open or close the contacts of a switch. It can be controlled directly by a number of microcontrollers such as Arduino, AVR, PIC and ARM. It uses a low level triggered control signal usually from 3.3-5VDC to control the relay. In this project, 5V relay is used i.e. it needs 5V to operate (as a switch).

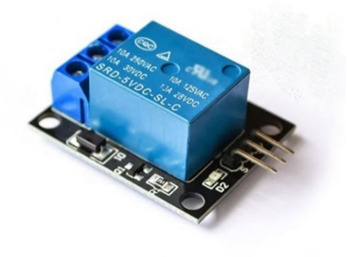


Fig.6. 5V power relay module

B. *Working*

The three different relay modules need to be activated not only according to the commands given by the user, but also from the readings of the sensors used. There are 3 processes going on in the system - lighting control, water pump for plants control and Air conditioner control.

1) *Lighting Control:* The lights' relay is solely controlled by the user. But the user can either control it by speaking out the command or through a manual switch provided in the room. The first scenario is where the user speaks out a command either to switch on or off the lights. This is then captured by the python application at the user's end. According to the kind of command, the application sends 0 (LOW) or 1 (HIGH) value to a topic called 'lights' in AWS IoT core.

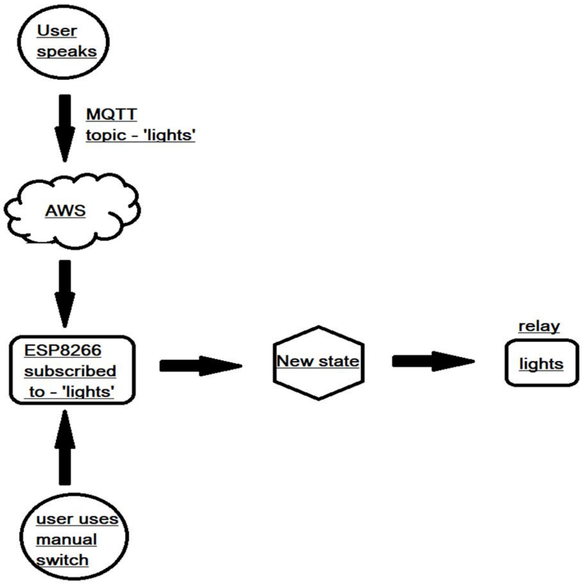


Fig.7. Flow of lighting control

This topic is subscribed by the ESP8266 module and thus it receives the command to do the same for relay (which is basically the switch) for the lights. The second scenario is where the user uses the manual switch that is connected directly to the hub i.e. the ESP8266.

Now, for both the scenarios to work together, a current state variable is used. This current state variable determines the status of the lights at any instant. Now when a new command is received by the ESP8266, the new state (if different) overrides the old state. Thus, the new command is processed and the state changes to the new state.

2) *Air Conditioning Control:* To control the AC relay, there are 2 ways for the user and along with that there is an automated control. This automated control is driven by the readings of DHT11. If the data from DHT11 reads greater than a limit (25^{o}C), then 1 (HIGH) is sent to the current state variable. These values of temperature are also sent to the non-SQL DynamoDB of AWS. These values can later be processed and analysed for more efficient control. The second scenario is when the user uses his voice commands. These commands work similar to that of the lighting control. The topic involved in the communication in this case is ‘AC’. The current state value gets over-ridden by the new state sent through voice. Along with that the user can also use the manual switches to control the same. The manual switches and the DHT11 control is a local process and does not involve AWS.

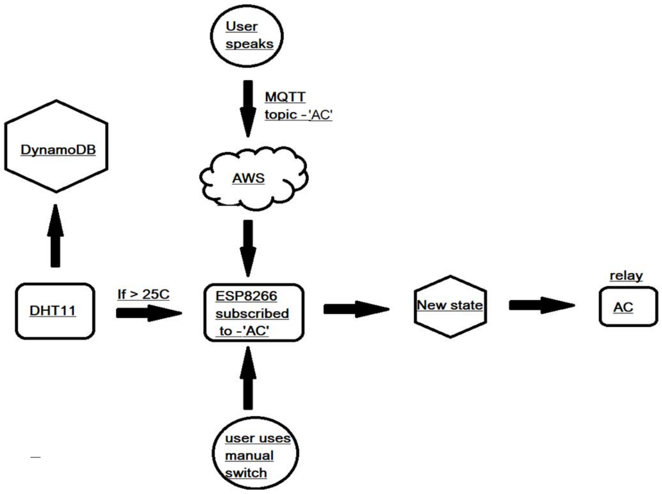


Fig.8. Flow Air conditioning control

3) *Water Pump for Plants Control:* The control of water pump to water the plants is somewhat similar to AC control. There are three ways to control this relay out of which two involve the user. The topic here involved is ‘pump’. The moisture sensor reading drives the automated control of the pump. If the moisture level drops below 10%, the current state is switched to 1 (HIGH). The moisture sensor readings are also sent to the DynamoDB of AWS for analysis and processing, using which the system performance can be enhanced. The user command is processed by the application similar to the above cases and then accordingly the changes are made in the current state through the cloud through the topic ‘pump’. The user can also control the pump using a manual switch. The current state variable gets overridden every time a new command is given to the relay. Again, the automated moisture sensor control and manual switches are localised processes and don’t involve the cloud.

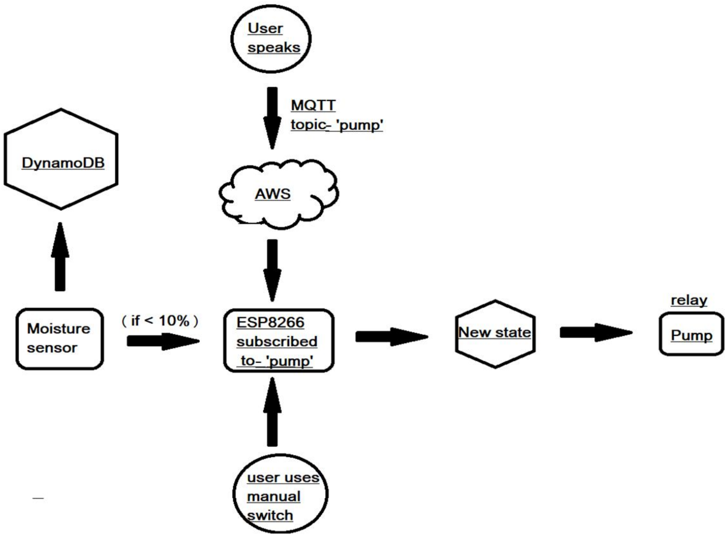
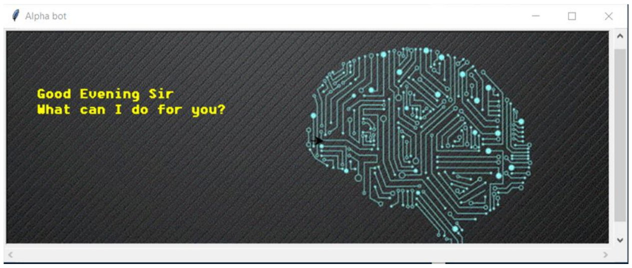


Fig.9. Water pump control

C. User Interface

The user interface is made smarter by enabling voice control. The application is programmed using python. The application makes use of the inbuilt read-out voices provided by the Windows OS. It is on an infinite loop of activating microphone and trying to recognise it. The 'ALPHA' assistant greets the user when switched o. after that a few commands – 'Switch on/off the lights', 'water the plants' and 'switch on the AC' are recognised by the application and accordingly, it sends the data needed to the respective topics on AWS IoT core.



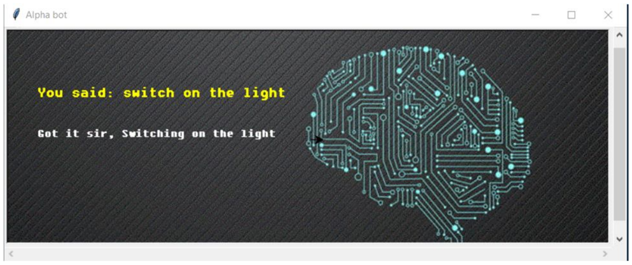
Good Evening Sir
What can I do for you?

Fig.10. Designed voice assistant – ALPHA

The following system can also be controlled from the console provided by the AWS IoT core.

IV. RESULTS AND DISCUSSION

The voice commands are visible on the python application which are then recognised and the necessary tasks are carried out. These voice command's interface can be seen in the fig. 11.



You said: switch on the light

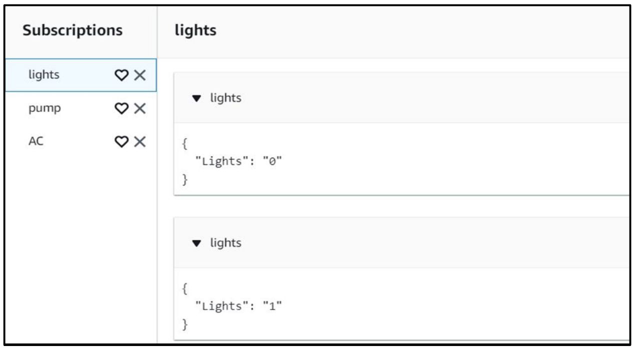
Got it sir, Switching on the light

Fig.11. Designed voice assistant – ALPHA

The manual control given to the system comprised of push buttons. Once you press a push button, the state changed. Three push buttons were connected for three different relays.

The relay wasn't connected directly to the ESP8266 hub. An NPN transistor diode is used to provide fast DC switch on-off.

The AWS console can be seen in Fig 13. The different topics involved in the project with the corresponding HIGH and LOW values. These values are then sent to the ESP8266, and then accordingly, the relays are operated. Each row of these console (for a particular topic) are the new status values i.e. the new state variable will get updated every time a new message is received.



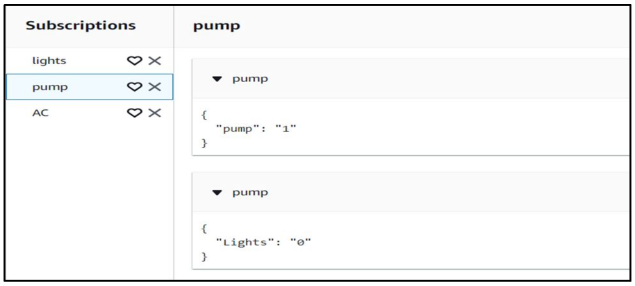
lights

```
{
  "Lights": "0"
}
```

lights

```
{
  "Lights": "1"
}
```

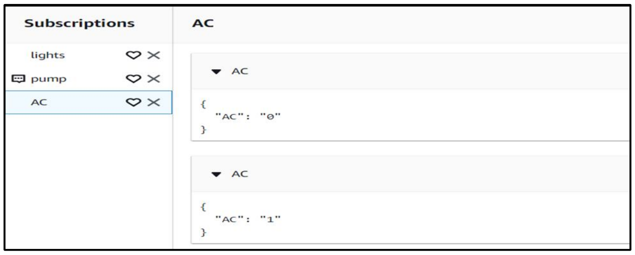
Fig.13.a Topic – lights on AWS console



```
pump
{
  "pump": "1"
}

pump
{
  "Lights": "0"
}
```

Fig.13.b Topic – pump on AWS console

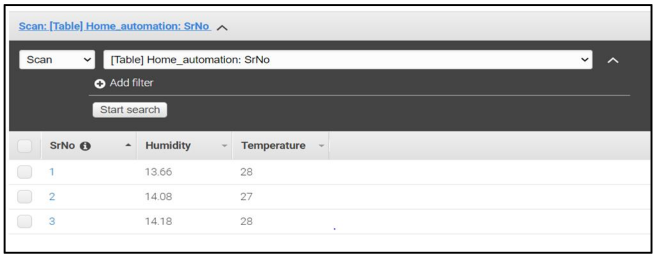


```
AC
{
  "AC": "0"
}

AC
{
  "AC": "1"
}
```

Fig.13.c Topic – AC on AWS console

The readings of DHT11 and moisture sensor are also sent to the DynamoDB. This will be extremely useful to carry out analysis and data processing as mentioned earlier. The Fig.14 shows sample data added into the DynamoDB.


Scan: [Table] Home_automation: SrNo
Scan
[Table] Home_automation: SrNo
Add filter
Start search

SrNo	Humidity	Temperature
1	13.66	28
2	14.08	27
3	14.18	28

Fig.14 DynamoDB entries

V. CONCLUSION

The automation industry still has a lot of challenges to be solved. The modules and the sensors used in the project, worked perfectly as expected and the results obtained were perfect too. To use a bigger system, one with a lot of relays and lot of sensors, ESP-WROOM-32 or RaspberryPi can be used. Further on, AWS also provides the feature of using Alexa to communicate with AWS which can be used instead of the python application. I believe that this is definitely a step towards cost-effective smart homes or industries. IoT has a bright future and it is the key towards a smart world.

REFERENCES

[1] Lohokare, Jay, et al. "An IoT ecosystem for the implementation of scalable wireless home automation systems at smart city level." TENCON 2017-2017 IEEE Region 10 Conference. IEEE, 2017.

[2] Kodali, Ravi Kishore, and SreeRamya Soratkal. "MQTT based home automation system using ESP8266." 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, 2016.

[3] Minchev, Dimitar, and Atanas Dimitrov. "Home automation system based on ESP8266." 2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA). IEEE, 2018.

[4] Kodali, Ravi Kishore, and Kopulwar Shishir Mahesh. "A low cost implementation of MQTT using ESP8266." 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I). IEEE, 2016.

[5] Ara, Anjum, and Shivkumar Jawaligi. "NodeMCU (ESP8266) Control Home Automation using Google Assistant." International Research Journal of Engineering and Technology 3644 (2019).

[6] Kodali, Ravi Kishore, et al. "IoT based smart security and home automation system." 2016 international conference on computing, communication and automation (ICCCA). IEEE, 2016.

[7] ElShafee, Ahmed, and Karim Alaa Hamed. "Design and implementation of a WIFI based home automation system." World academy of science, engineering and technology 68.2012 (2012): 2177-2183.