

# **VOICE BASED HOME AUTOMATION USING IBM WATSON**

Submitted by

P.RAJESH	-	18481A04F5
P.DIVYA	-	18481A04F6
P.LIKITHA	-	18481A04F7
P.MANIKANTA	-	18481A04F8
P.MANVITHA	-	18481A04F9

(ELECTRONICS AND COMMUNICATION ENGINEERING)  
GUDLAVALLERU ENGINEERING COLLEGE

## **CONTENTS :**

### 1.Introduction

#### 1.1 Overview

#### 1.2 Purpose

### 2.Literature

#### 2.1 Existing problem

#### 2.2 Proposed solution

### 3.Theoretical analysis

#### 3.1 Block diagram

#### 3.2 Hardware/software designing

### 4.Experimental investigations

### 5.Flowchart

### 6.Result

### 7.Advantages & Disadvantages

### 8.Applications

### 9.Conclusion

### 10.Future scope

### 11.Bibilography

### 12.Appendix

#### 12.1 Source code

#### 12.2 UI Output screenshot

# VOICE BASED HOME AUTOMATION USING IBM WATSON

## 1. INTRODUCTION :

### 1.1 Overview :

Automation is the use of control systems and information technology to control equipment, reducing the need for human intervention. In the scope of industrialization, automation is step beyond mechanization. Automation greatly reduces the need for human sensory and mental requirements as well.

This voice based home automation system acts according to the person's commands. It helps the user in controlling appliances in the home easily by giving voice commands. Home automation refers to use of computer and information technology to control home appliances and features. Home automation is about and there is no end to its application. Automation plays an increasingly important role in global economy and in daily experiences.

Our project voice based home automation helps to control the electrical appliances based on the input signal. The device receives this input signal using IBM Watson. This system solves the issue as now the user just has to give voice commands to turn on or off the appliances. Here we use lights, fridge to demonstrate. All these appliances can be individually turned ON/OFF or all loads. This system solves the issue by interfacing a unit with home appliances that switches based on the input given by the device.

### 1.2 Purpose :

Automation plays a key role in human life. Home automation allows us to control household electrical appliances like light, door, fan, AC etc. It also provides home security and emergency system to be activated. Home automation not only refers to reduce human efforts but also energy efficiency and time saving.

We would develop an authentication to the system for authorized person to access home appliances. The device with low cost and scalable to less modification to the core is much important. It presents the design and implementation of automation system that can monitor and control home appliances.

The main aim of our system is to build a perfect companion for someone to be at home. Generally, home automation research targeted many needs like applications that provide the luxury and smart requirements. Our system is a computer based system that can accept voice to direct commands and process them. The system provides us switching any device ON/OFF.

## 2.LITERATURE :

### 2.1 Existing problem :

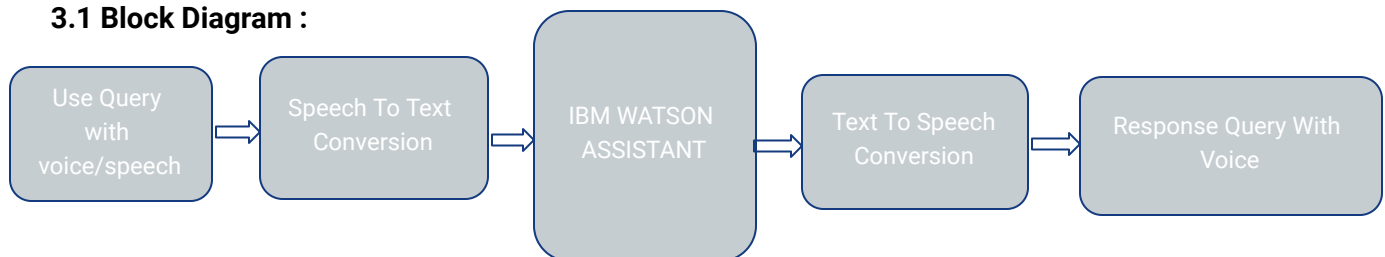
As the modern software system, home automation control system should achieve any requirements from users by its intelligence and generalization. It is impossible for any users to understand the instructions or algorithm of programming. Fortunately, we have a lot of browsing sites that support us with the information. But sometimes, this might not be the case.

### 2.2 proposed solution :

To overcome this, our project voice-based home automation is very helpful for the users. Voice is the most direct and simple way for people to convey their requirement. We propose one solution to enable voice control, and the solution integrates the available voice recognition within the existing home automation system. First, we describe the proposed architecture and system components. Then we verify the correctness of the implementation. Also, we confirm that the response time of the proposed solution is acceptable.

## 3. THEORETICAL ANALYSIS :

### 3.1 Block Diagram :



### 3.2 Hardware/Software Designing :

Software Designed :

- Installation of Python Idle
- Installation of Node Red
- Installation of Watson Assistant
- Installation of Text to Speech(TTS)
- Installation of Speech to Text(STT)

Along with their App URL and their respective API Keys.

## 4.EXPERIMENTAL INVESTIGATION :

For the experimental set up, initially we need to make the appropriate connects in the **Node Red app** provided with their service credentials and deploy the app. This yields the **TTS** and **STT** conversions. Later we need to design a chatbot in such a way that it takes the user input and results relevant output assigned by the developer. An assistant node is to be taken and supplied with the **Watson** credentials so that voice-enabled chatbot.

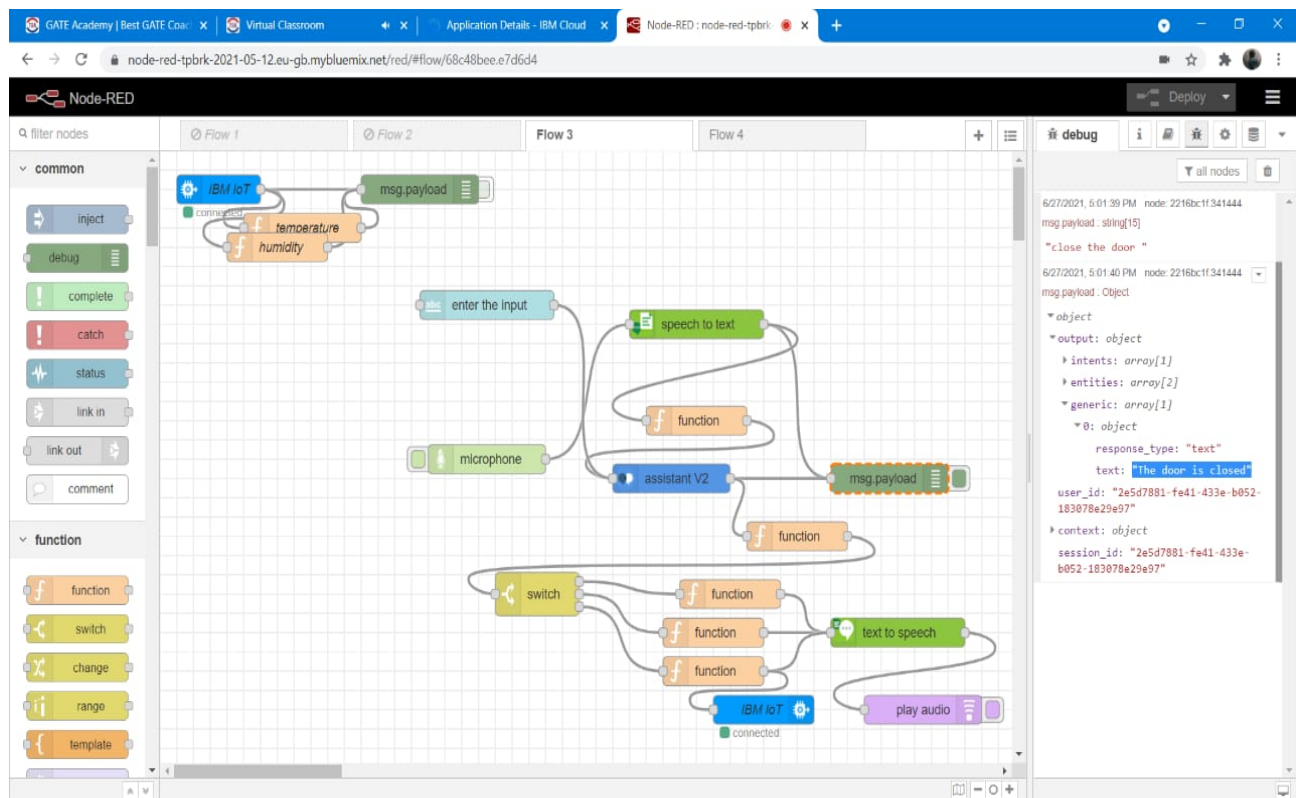
Firstly we need to connect the node-red sensor to the IBM IoT platform, from this we get the temperature and humidity ranges with msg.payload. Next, here the input is given in the form of speech voice message using a microphone installed from the manage pallets and the output is text.

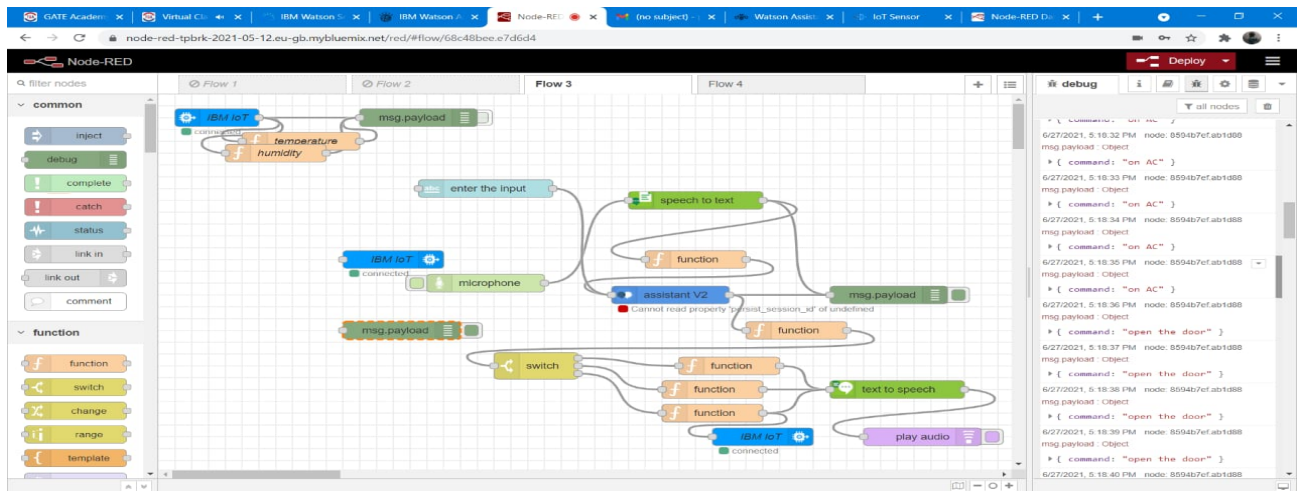
form which is displayed using a debug node. The assistant v2 node is supplied with its service credentials from the watson assistant and it is taken from the output of speech to text node. The output of assistant node is given to a function node rather than directly supplying to the speech to text node there by avoiding bad requests.

Necessary **function** is written inside the function node using the **split** method and the Node Red app is thus deployed to save the changes. and the node red is thus deployed to save the changes. The output of the function node is connected to switch node with 3 function nodes as temperature, humidity and for all appliances. The Text to Speech node is used to convert the given text input to speech output. We give 3 function nodes at the input end whereas play audio node at the output end.

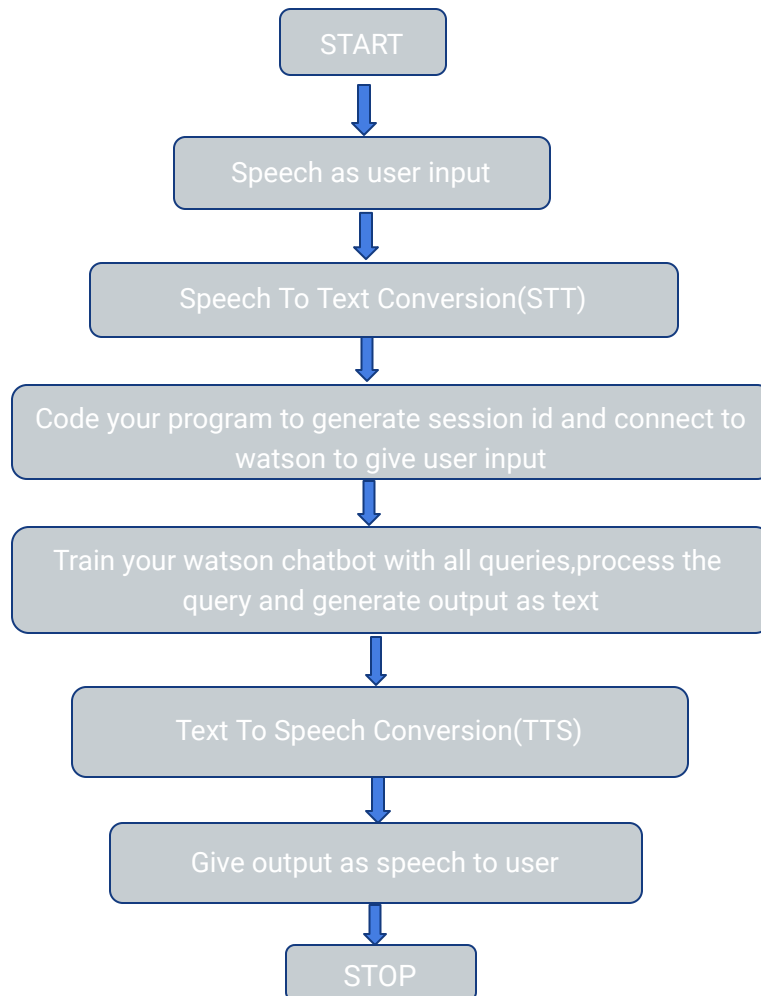
The ibm iot is connected to the function node, and ibm iot is preliminary trained with few queries that the user might put to the bot and the answers were also saved to the bot. When the user requests for his required information using his microphone, the smart assistant gathers the information from the Watson Assistant using the assistant node and yields speech output.

Ensure that the API keys and the URL's are properly located in their respective block of nodes. If the service credentials are misplaced from their respective nodes we may encounter a few errors and if the assistant credentials are given mistake, then it displays missing property. In this manner we can find the flawless nodes and connect them.





## 5.FLOWCHART :



## **6.RESULT:**

This voice based system makes the particular work easier regarding home appliances, which saves the time of a user who are struggling to do it physically in particular time.

## **7.ADVANTAGES & DISADVANTAGES :**

### **Advantages :**

- Managing all of your home devices from one place.
- Flexibility of new devices and appliances.
- Maximizing home security.
- Remote control of home functions.
- Increased energy efficiency.
- Improved appliance functionality.
- Home management insights.

### • **Disadvantages :**

- Voice based systems are not completely hands free.
- Voice recognition Messes up.
- Potential Violation of privacy.
- Limited to Basic Commands.

## **8.APPLICATIONS :**

The concept of Home Automation is gaining popularity as it helps in reducing human effort and errors and thus increasing the efficiency. With the help of Home Automation system, we can control different appliances like lights, fans, TV, AC etc. Additionally, a home automation system can also provide other features like security, alarms, emergency systems etc. can be integrated.

- The Voice Based Home Automation system will help us control different loads (electrical appliances) with simple voice commands.
- This kind of system is very useful for people with disabilities.
- Further, the project can be expanded by adding different sensors (light, smoke, etc.).

## **9.CONCLUSION :**

Voice based home automation is successfully implemented. Establishing the system with little modifications could bring some change in the way we interact with our home appliances.

## **10.FUTURE SCOPE :**

Future scope for the home automation systems involves making homes even smarter. Homes can be interfaced with sensors including motion sensors, light sensors and temperature sensors and provide automated toggling of devices based on conditions. More energy can be conserved by ensuring occupation of the house before turning on devices and checking brightness and turning off lights if not necessary.

The system can be integrated closely with home security solutions to allow greater control and safety for home owners. The next step would be to extend this system to automate a large scale environment, such as offices and factories. Home Automation offers a global standard for

interoperable products. Standardization enables smart homes that can control appliances, lighting, environment, energy management and security as well as the expandability to connect with other networks.

## 11.BIBLIOGRAPHY :

<https://youtu.be/RQ-fiBbesa4>

<https://youtu.be/BZV019HFpPM>

<https://youtu.be/hhhsy1l0N9g>

## 12.APPENDIX:

### 12.1 Source code :

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import json

#Provide your IBM Watson Device Credentials
organization = "3enokg"
deviceType = "iotdevice"
deviceId = "1001"
authMethod = "token"
authToken = "1234567890"
# Initialize the device client.
T=0
H=0
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    if cmd.data['command']=='lighton':
        print("LIGHT ON IS RECEIVED")
    elif cmd.data['command']=='lightoff':
        print("LIGHT OFF IS RECEIVED")
    if cmd.data['command']=='off fan':
        print("FAN OFF IS RECEIVED")
    elif cmd.data['command']=='on fan':
        print("FAN ON IS RECEIVED")
    if cmd.data['command']=='off AC':
        print("AC OFF IS RECEIVED")
    elif cmd.data['command']=='on AC':
        print("LIGHT OFF IS RECEIVED")
```

```

    if cmd.data['command']=='off fridge':
        print("LIGHT OFF IS RECEIVED")
    elif cmd.data['command']=='on fridge':
        print("LIGHT OFF IS RECEIVED")
    if cmd.data['command']=='close the door':
        print("LIGHT OFF IS RECEIVED")
    elif cmd.data['command']=='open the door':
        print("LIGHT OFF IS RECEIVED")
    if cmd.command == "setInterval":
        if 'interval' not in cmd.data:
            print("Error - command is missing required information: 'interval'")
        else:
            interval = cmd.data['interval']
    elif cmd.command == "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information: 'message'")
        else:
            print(cmd.data['message'])
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod,
"auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10
times
deviceCli.connect()
while True:
    T=23
    H=45
    #Send Temperature & Humidity to IBM Watson
    data = {"d":{"temperature": T, 'humidity': H }}
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % T, "Humidity = %s %" % H, "to IBM Watson")
    success = deviceCli.publishEvent("Data", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
    time.sleep(1)
    deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()

```



```

Command Prompt - python fp.py
Command received: light on
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Command received: light on
Command received: light on
Command received: light on
Command received: light on
Command received: light on
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Command received: light on
Command received: light on
Command received: light on
Command received: light on
2021-06-27 17:46:59,139 ibmiotf.device.Client ERROR Unexpected disconnect from the IBM Watson IoT Platform: 1
Not connected to IoT
Not connected to IoT
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:47:01,879 ibmiotf.device.Client INFO Connected successfully: d:3enokg:iotdevice:1001
Command received: light on
Command received: light on
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Command received: light on
Command received: light on
Command received: light on
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:47:06,942 ibmiotf.device.Client ERROR Unexpected disconnect from the IBM Watson IoT Platform: 1

```

```

Command Prompt - python fp.py
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:33:53,991 ibmiotf.device.Client INFO Connected successfully: d:3enokg:iotdevice:1001
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:33:56,093 ibmiotf.device.Client ERROR Unexpected disconnect from the IBM Watson IoT Platform: 1
Not connected to IoT
Not connected to IoT
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:33:58,954 ibmiotf.device.Client INFO Connected successfully: d:3enokg:iotdevice:1001
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:34:00,948 ibmiotf.device.Client ERROR Unexpected disconnect from the IBM Watson IoT Platform: 1
Not connected to IoT
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:34:03,798 ibmiotf.device.Client INFO Connected successfully: d:3enokg:iotdevice:1001
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:34:06,772 ibmiotf.device.Client ERROR Unexpected disconnect from the IBM Watson IoT Platform: 1
Not connected to IoT
2021-06-27 17:34:08,520 ibmiotf.device.Client INFO Connected successfully: d:3enokg:iotdevice:1001
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
2021-06-27 17:34:11,459 ibmiotf.device.Client ERROR Unexpected disconnect from the IBM Watson IoT Platform: 1
Not connected to IoT
Not connected to IoT

```

## 12.2 UI Output Screenshot

