

# Expanding Voice Commands for Visually Impaired Interfacing Google Home with Arduino using Webhooks

Sameer Tuteja

UG Student,

Department of Mechanical Engineering, Acropolis Technical Campus

Affiliated to R.G.P.V., Bhopal

Email: sameer.t008@gmail.com

**Abstract-** There are challenges these days to find remotes to control our home appliances be it our television or setup box or even AC. Smartphones do help us but not completely. "Internet of Things" is a term growing very rapidly, it includes creating a network of physical objects around us and connecting them to the world of Internet. We have designed an **IOT based installable smart assistive low-cost voice-controlled device** as a solution. This can also be beneficial for visually impaired people. The solution aims to perform all functions which can be controlled by a remote like turning off/on the lights or appliances, controlling TV channels, opening doors, maintaining temperature, surveillance etc. through voice commands and is easily installable. The difference is that our project reduces the integration time of ESP8266 with Google Assistant. The latency rate of response between devices reduces as command has to cover only through a single layer of data transfer.

**Index Terms-** IOT, Google Assistant, Webhooks, Arduino, ESP8266-01

## 1. INTRODUCTION

IoT provides the ability to connect a large number of devices using networking. These devices have their own unique identifiers and protocol, create a smart environment through which other devices can be connected over internet [1]. These are incorporated with microcontrollers which controls the working of the device. Microcontroller is connected to sensors which acts as their source of information for the physical changes around the device. It activates the response with the help of actuators corresponding to that specific input signal. Wireless modules like ESP8266-01 by Ai-Thinker is used to connect to the internet which helps in monitoring and controlling these devices using web services. This works on a simple protocol, Google Home (Google Assistant) and Arduino (Arduino UNO) are connected using ESP8266 through 'Webhooks' a service by IFTTT (If This Then That) [2]. Webhooks help to send direct web requests to a publicly accessible web page or URL (Uniform Resource Locator). As there is no direct method to integrate commands through Google Assistant to a DIY smart device made using ESP, Webhooks acts as a simple channel.

Some integrations have already been conducted using various services available on internet which are either free or require a nominal charge to be paid. But have large latency issues. ESP8266 is a system-on-chip which is integrated with full TCP/IP stack used to connect to Wi-Fi network and make simple TCP/IP connections. ESP works on the factory firmware which operates on AT commands developed by Dennis Hayes in 1981 [3]. To establish a web server using ESP8266 module, this firmware with AT commands is used which is surprisingly straightforward and requires very simple set of codes.

## 2. METHODOLOGY

This project works on the follows line of action:

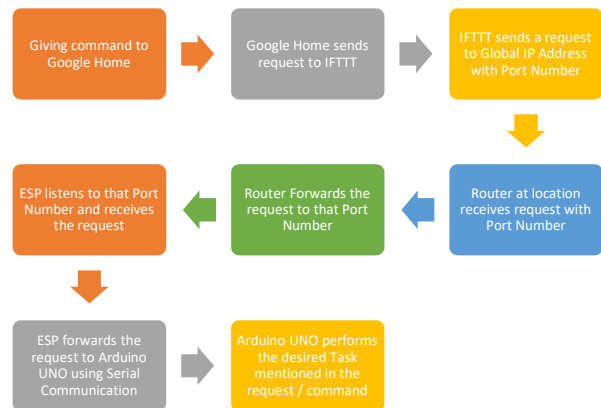


Figure 1 Process Flow Chart

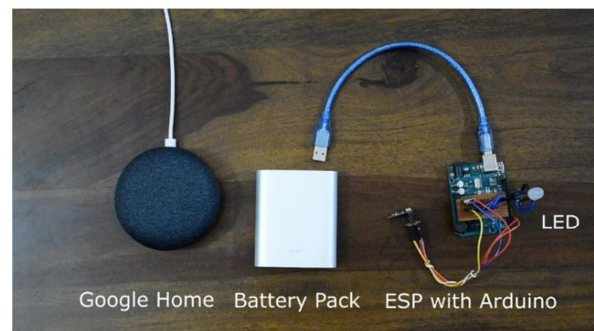


Figure 2 Overall Setup

## 3. RESOURCES USED

### 3.1. Hardware Used

- Google Home Mini / Assistant

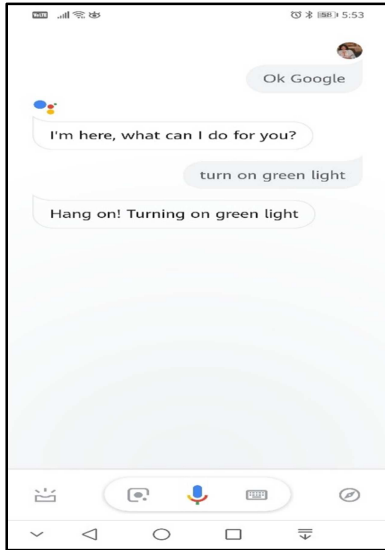


Figure 3 Google Assistant Interface

- Wi-Fi Router / Network Router
- Arduino UNO
- Ai-Thinker ESP8266-01
- RGB Led
- Battery Pack / Power Supply

### 3.1.1. Google Home Mini / Assistant

Google Assistant is a virtual assistant developed by Google. It creates a personalized voice model so that it can identify its user. Google Home is smart speaker which incorporates assistant service that also supports

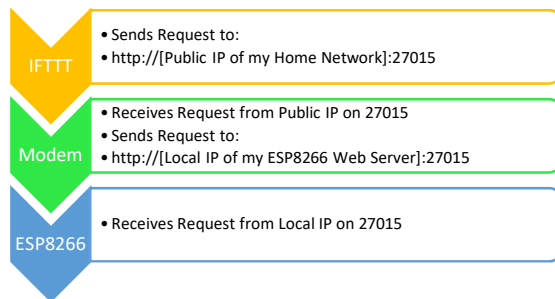


Figure 4 Data Flow Path

home automation. A preprogrammed command “OK Google” is used to wake up the device.

### 3.1.2 Wi-Fi Router

It is used for connecting the Google Home Mini and ESP8266 to internet. Static IP address and a port

number is provided to ESP8266. Port Numbers are specific logical address for each application or service connected to a network which helps to identify the sender and the receiver of the message. In this case IFTTT sends information to the public IP and the port number which ESP8266 is listening to. 27015 is selected as the port number used for both internal and external routing [4].

### 3.1.3 Arduino UNO

Microcontroller controls the sensors and actuators used in the project. All devices to be controlled are attached to this board [5].

### 3.1.4 ESP8266-01

Low cost Wi-Fi module which allows creation of local server with dedicated port number to support HTTP requests.

### 3.1.5 RGB Led

This is attached to Arduino UNO which helps to reflect the output using voice commands.

### 3.1.6 Battery Pack / Power Supply

Power supply of 5V is required for Arduino UNO. ESP8266 requires 3.3V supplied by UNO. Here a Xiaomi 10,000mAh is used.

## 3.2. Software Used

- IFTTT Webhooks
- Arduino IDE

### 3.2.1 IFTTT Webhooks

Webhooks service by IFTTT for using GET and POST



Figure 5 Webhooks Logo

methods on HTTP web page. Here the web page is directly created by ESP8266 [6].

GET command in http is used to fetch data from specific source while POST command is used to send data to specific server to add or update the value.

### 3.2.2 Arduino IDE

Coding for Arduino UNO is done using this IDE. The code for UNO contains AT commands which are communicated using serial communication done between UNO and ESP8266. This uses TX and RX pins on both devices.

### 3.3. Connections

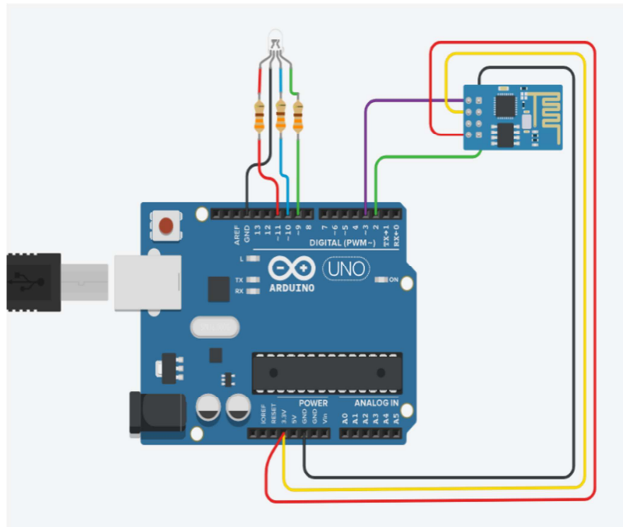


Figure 6 Connections with Arduino UNO

## 4. RESULTS AND ANALYSIS

The ESP8266 connects to the Wi-Fi router and by saying the hot word “OK Google” with the commands:

- Turn on/off Red Light
- Turn on/off Blue Light
- Turn on/off Green Light
- Turn on/off All Lights

We get the desired output with change in state of LED. Complete working of the project can be seen on: <https://youtu.be/9BSXK59POIE>

The latency rate between acceptance of command by Google Home Mini and execution of command in form of LED toggle is represented in the graph below. This shows an average latency of 5 seconds. The data in this graph can be verified by the following video:

<https://youtu.be/oZqfekhXqcM>

An implementation of this project to control TV using voice commands like ‘switch to Channel 251’ or any other number of that form. A small demonstration can be seen in this video: [https://youtu.be/aDX-r-yB\\_ek](https://youtu.be/aDX-r-yB_ek)



Figure 7 ESP8266 Pinouts [8]

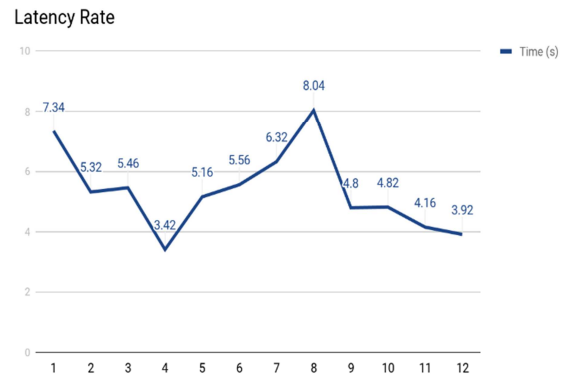


Figure 8 Graph for Latency Rate

## 5. FUTURE IMPLEMENTATIONS

This project can be expanded to control various devices which are already present in market. Attaching relay modules to this project enhances its working capabilities.

The use of IFTTT webhooks can be further utilized with multiple applets and devices. Integration with Amazon's Alexa.

## 6. CONCLUSION

Technology must be used as a tool to unlock learning and expand the horizons of its end-users. It is a great equalizer. It should also be used to improve the functional capabilities of blind and visually impaired which forms the locus of this project [9].

This project being cost efficient and user friendly will help students and enthusiasts to build their own IoT devices which can be controlled using Google Assistant on their smartphones.

This project shows that Google Home / Assistant can be used to control DIY smart devices using phrases or commands. The results are obtained in real time with minimum latency. This project can be modified according to the requirement and is beneficial for both beginners and experts.

The data is never lost and if the device gets overloaded, data is stored in queue. ESP8266 is provided with a timeout value for each command which helps in improving its efficiency to take new commands.

## ACKNOWLEDGMENT

I would like to express my sincere gratitude to my Professor Mr. Narendra Sharma, Department of Mechanical Engineering and Professor Nilesh Dubey, Department of Electronics and Communication,

Acropolis Technical Campus, Indore for their guidance and encouragement in this project.

I would also like to thank my sister for her continuous support throughout this project.

#### **REFERENCES**

- [1] "What is the Internet of Things? | IoT Technology - SAP.com." <https://www.sap.com/trends/internet-of-things.html>.
- [2] "IFTTT.com." <https://ifttt.com/>.
- [3] "ESP8266 AT command examples – Espressif Systems." [https://www.espressif.com/sites/default/files/documentation/4besp8266\\_at\\_command\\_examples\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/4besp8266_at_command_examples_en.pdf).
- [4] "Setting Up Port Forwarding on Your Router – Parsec." <https://support.parsecgaming.com/hc/enus/articles/115002770371-Setting-Up-Port-Forwarding-On-Your-Router>
- [5] "Arduino ArduinoUno." <https://www.arduino.cc/en/Guide/ArduinoUno>.
- [6] "Do more with Webhooks IFTTT." [https://ifttt.com/maker\\_webhooks](https://ifttt.com/maker_webhooks).
- [7] "Taking MQTT and NodeMcu to IOT: Communication in Internet of Things." [https://www.researchgate.net/publication/325657453\\_Taking\\_MQTT\\_and\\_NodeMcu\\_to\\_IOT\\_Communication\\_in\\_Internet\\_of\\_Things](https://www.researchgate.net/publication/325657453_Taking_MQTT_and_NodeMcu_to_IOT_Communication_in_Internet_of_Things).
- [8] "ESP-01/07/12 Series Modules User's Manual." [http://wiki.aithinker.com/\\_media/esp8266/a000um01a3.pdf](http://wiki.aithinker.com/_media/esp8266/a000um01a3.pdf).
- [9] "An overview of assistive technology for students Paths to Literacy." <http://www.pathstoliteracy.org/overview-assistive-technology>.