IOT Home Automation via Google Assistance

Ram Shankar Singh¹, V. Naveen Venkat², Mr. A. V. Sudhakara Reddy³ and Dr. D. Raja Reddy⁴

1 Final Year B.Tech, Department of Electrical and Electronics Engineering, Malla Reddy Engineering College (A), India.
²Associate Professor, Department of Electrical and Electronics Engineering, C.B.I.T, Proddatur, Andhra Pradesh, India.
³Associate Professor, Department of Electrical and Electronics Engineering, Malla Reddy Engineering College (A). India.
⁴E-Mail: sreddy4svu@gmail.com

Abstract— This project presents a proposal for home automation using voice via Google Assistant. Home automation or domestics a term for home automation coined by Jim Hill has been evolving drastically. We saw many home automation technologies introduced over these years from Zigbee automation to Amazon Echo, Google Home and Home from Apple. It has become a craze these days. Google Home price is around 150$ (USD) with an additional cost of the devices to be connected to, the total cost of the system reaches over 250$ (USD). Apple Home Kit too is pretty more expensive, over 100$ (USD) more than the Google Home just for a basic setup. Philips Hue, a smart light which is controlled by the Google Assistant, Amazon Echo and Siri, voice assistant by Apple is priced around 145$ (USD). Similarly, Belkin’s Wemo light is priced around 44$ (USD) per unit and this can be controlled both by Siri and Google Assistant. So, overall we can see here that to make our home smart we need to invest quite a lot, let’s say some 250$ (USD) for a basic setup. What if we can automate our house within (cost of the Smartphone is not included as it is assumed to be owned by every individual these days) 10$ (USD) and can control up to 8 appliances using Google Assistant? Well, this project describes the implementation of such a system. The system is implemented using ordinary household appliances Natural language voice commands are given to the Google Assistant and with the help of IFTTT (If This Then That) application and the Adafruit application the commands are decoded and then sent to the microcontroller, the microcontroller in turn controls the relays connected to it as required, turning the device connected to the respective relay On or OFF as per the users request to the Google Assistant. The microcontroller used is NodeMCU (ESP8266) and the communication between the microcontroller and the application is established via Wi-Fi (Internet).

Keywords— home automation, Google Home, Amazon Echo, Google Assistant, micro controller, Wi-Fi

I. INTRODUCTION

Home, it is the place where one fancies or desires to be after a long tiring day. People come home exhausted after a long hard working day. Some are way too tired that they find it hard to move once they land on their couch, sofa or bed. So any small device/technology that would help them switch theirs lights on or off, or play their favorite music etc. on a go with their voice with the aid of their smart phones would make their home more comfortable. The hardware also called the Control Unit comprises of the Node MCU microcontroller and the Relay board. Node MCU’s digital output pins are connected to the Relay pins of the Relay board. Finally, each Relay is connected to an appliance.

Google assistant is AI (Artificial Intelligence) based voice command service. Using voice, we can interact with google assistant and it can search on the internet, schedule events, set alarms, control appliances, etc. This service is available on smartphones and Google Home devices. We can control smart home devices including lights, switches, fans and thermostats using our Google Assistant. We will build an application which can control home appliances. Here, we will control 220V using Google Assistant service.
II. ELECTRICAL AND ELECTRONICS COMPONENTS

A. Hardware
   (a) Esp8266 WiFi Module
   (b) Electromagnetic Relay 4-Channel 12v
   (c) 5v Adapter
   (d) 220 AC Power Supply
   (e) 220v AC Loads

B. Software
   (a) ARDUINO Software
   (b) ADAFRUIT/ADAFRUIT MQTT WEBHOOK
   (c) IFTTT
   (d) Android Device (Google Assistance)

C. System Overview

The CP2102 is a highly-integrated USB-to-UART Bridge Controller providing a simple solution for updating RS-232 designs to USB using a minimum of components and PCB space. The CP2102 includes a USB 2.0 full-speed function controller, USB transceiver, oscillator, EEPROM, and asynchronous serial data bus (UART) with full modem control signals in a compact 5 x 5 mm MLP-28 package. No other external USB components are required. The on-chip EEPROM may be used to customize the USB Vendor ID, Product ID, Product Description String, Power Descriptor, Device Release Number, and Device Serial Number as desired for OEM applications. The EEPROM is programmed on-board via the USB allowing the programming step to be easily integrated into the product manufacturing and testing process. Royalty-free Virtual COM Port (VCP) device drivers provided by Silicon Laboratories allow a CP2102-based product to appear as a COM port to PC applications. The CP2102
UART interface implements all RS-232 signals, including control and handshaking signals, so existing system firmware does not need to be modified. In many existing RS-232 designs, all that is required to update the design from RS-232 to USB is to replace the RS-232 level-translator with the CP2102. An evaluation kit for the CP2102 (Part Number: CP2102EK) is available. It includes a CP2102-based USB to UART/RS-232 evaluation board, a complete set of VCP device drivers, USB and RS-232 cables, and full documentation. Contact a Silicon Labs’ sales representatives or go to www.silabs.com to order the CP2102 Evaluation Kit.

D. Specification
- Voltage: 3.3V. Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA–170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80–160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n. • Maximum concurrent TCP connections:

E. Configuration
The most basic way to use the ESP8266 module is to use serial commands, as the chip is basically a WiFi/Serial transceiver. However, this is not convenient. What we recommend is using the very cool Arduino ESP8266, which is a modified version of the Arduino IDE that you need to install on your computer. This makes it very convenient to use the ESP8266 chip as we will be using the well-known Arduino IDE. Following the below step to install ESP8266 library to work in Arduino IDE environment.

F. Main Features
- Switching capacity available by 10A in spite of small size design for high density P.C. board mounting technique.

Special Issue:
Department of Electrical and Electronics Engineering, Malla Reddy Engineering College (Autonomous)
• UL,CUL,TUV recognized.
• Selection of plastic material for high temperature and better chemical solution performance.
• Sealed types available.
• Simple relay magnetic circuit to meet low cost of mass production.

III. CONSTRUCTION

The system design is broken down into two main categories.

A. The Hardware
It has the capability to connect to the router. It would also be able to turn on/off specified devices, such as lights and fans. It is called the ‘Control Unit’.

B. The Software
The Adafruit website, the IFTTT app and the Google Assistant constitute the software of the design and these applications would be integrated in the Android device. The Control Unit comprises of the microcontrollerNodeMCU and the 4/8 Channel Relay board. Relay board uses ULN 2803 IC to control the relays. The Adafruit website on an Android device communicates with the microcontroller and sends the desired signal via the internet. Figure 1 below shows the basic system design architecture.

IV. HARDWARE

A. Connection between NodeMCU and RELAY

• NodeMCU digital pins D1, D2, D3, D4 is connected to the RELAY pins IN1, IN2, IN3, IN4.
• Give 3.3V supply to the NodeMCU.
• Give 5V to the relay.

Fig. 3.1: NodeMCU and RELAY Connection
B. Steps of Construction

Fig. 3.2 Hardware Connection between Load and RELAY

Steps of Construction

Main supply is connected to NO (Normally OPENED) of RELAY 1 and then connected in series to RELAY 2 C(COMMON) connector.

Fig. 3.3 RELAY to Load Connection

Special Issue:
Department of Electrical and Electronics Engineering, Malla Reddy Engineering College (Autonomous)
RELAY3 C (COMMON) connector,
RELAY4 C (COMMON) connector.

The phase part of the loads are connected to
RELAY 2 NO (Normally Opened),
RELAY 3 NO (Normally Opened),
RELAY NO Normally Opened).

V. SOFTWARE

A. Adafruit

First, created account at www.Adafruit.io

B. SIGN UP

Now, create dashboard at Adafruit. This dashboard is a user interface to control things remotely.
After following above steps, provide name to the dashboard and save it. We can see our dashboard as follows

Now, create feed (user interface) to control light On-Off. To create it, just click on ‘+’ symbol

Special Issue:
Department of Electrical and Electronics Engineering, Malla Reddy Engineering College (Autonomous)
Step 5: Hardware Output Execution

Step 6: Command Execution via Google Assistance

Here, IFTTT application is used to bridge the gap between the Google Assistant commands and the Adafruit app. Setting up the IFTTT application first requires logging in after which we need to create an applet and then “This”, i.e. the trigger, here we select Google Assistant and then we will type in the commands to which the Google Assistant should respond and to this command it should control the appliance/relay associated with it. The response command from the Google Assistant can also be typed in as desired. After configuring the trigger, i.e. “This” of the application we need to configure the “That”. What should be done once the Google Assistant hears the command which we just configured? This is decided by setting “That” of the app. We click “That” and then select webhooks and click connect. Webhooks will allow us to send commands to the Adafruit Server. Now, in the URL we type the IP address of the Adafruit server followed by the Authentication token sent by the Adafruit and then the pin number of the microcontroller to which the device to be controlled is connected. The URL should be in the following format:

```plaintext
<IP address>/api/devices/<device_id>/commands/<command>
```
http://188.166.206.43/AuthToken/pin/CorrespondingDigit alPinNo Then in the method we select ‘PUT’ and the content type is ‘Application/JSON’ [8] and in the body we write ‘{"1"}’ to turn ON and ‘{"0"}’ to turn OFF. This creates the action for the trigger i.e. the Google Assistant command. The action taken by it is simply sending a message to the Adafruit website to either turn ON or OFF the concerned connected device. Finally, the microcontroller is programmed with the actions it needs to do once it receives the signal from the Adafruit website. Before that, the Adafruit and the microcontroller should communicate and the communication is done via the internet and since the microcontroller, NodeMCU comes with inbuilt Wi-Fi module, it is programmed to connect to the desired network once plugged in. ‘C’ language is used to program the microcontroller and is programmed in the Arduino IDE.

VI. CONCLUSION

The aim of this project was to propose a cost effective voice controlled (Google Assistant) home automation controlling general appliances found in one’s home. The approach discussed in the project was successful as GACHA’s (Google Assistant Controlled Home Automation) design was successfully implemented. This system is highly reliable and efficient for the aged people and differently abled person on a wheel chair who cannot reach the switch for the switching ON/OFF the device and are dependent on others. The future scope for GACHA can be huge. There are many factors to improve on to make GACHA more powerful, intelligent, scalable, and to become better overall for home automation. For example, controlling the speed of the fan, more number of devices can be integrated, like a coffee machine, air conditioner etc. To make the system respond more faster own private Adafruit server can be made. Well, no system is ever perfect. It always has a scope for improvement. One just needs to put on a thinking cap and try and make the system better.

REFERENCES


Special Issue:
Department of Electrical and Electronics Engineering, Malla Reddy Engineering College (Autonomous)
© Malla Reddy Engineering College (Autonomous), Vol. 5, Issue 1, March 2019. 52


[13] https://docs.blynk.cc/