

GOOGLE ASSISTANT BASED VOICE CONTROL HOME AUTOMATION

¹Mansi Solanki, ²Ruchika Doda

¹Student, ²Assistant Professor

Department of Electronics and Communication Engineering
MVSIT, Sonipat, India

Abstract Now a days comfort is becoming everyone's first priority. The proposed project is based on home automation using NodeMCU, blynk and google assistant. Basically, this paper describes how to configure IFTTT to control the home appliances like bulb over the internet using google assistant. NodeMCU cannot be directly connected with Google Assistant, an intermediate app i.e. blynk app is required. Blynk app can be directly connected to NodeMCU and send data to it. If we send the commands interpreted by google assistant directly to the Blynk application, the Blynk app forward those command to the NodeMCU. But the google assistant cannot directly understand the commands like "turn on the light" or "turn off the light" on its own. To handle this situation, another intermediate tool called IFTTT is used which will cover the gap between google assistant and the Blynk application. when we say a voice command to google assistant, IFTTT will interpret that command and send the request to the Blynk application via Blynk server and Blynk app sends the command to the NodeMCU and then to the home appliances like bulb etc.

Keywords: NodeMCU (ESP8266), IFTTT (If This Than That) Application, Blynk Application, Internet of Things (IoT), Google Assistant, Voice Control, Home Automation.

1. INTRODUCTION

Today in the headway of Automation innovation, life is getting simpler and less demanding in all spheres. Home automation is a modern technology that modifies your home to perform different sets of task automatically. No wonders, home automation in India is already the buzz word, especially as the wave of second generation home owners grows, they want more than shelter, water, and electricity. There is no shortage of possibilities for smart home IoT devices as home automation seems to be the wave of the future. The requirement for Office and Home automation arises due to the advent of IoT, in a big way in homes and office space. These smart automation devices happen to have an interface with IoT. IT automation will be the key to bridging the gap between human limitations and technology's capabilities. The modern homes are automated through the internet and the home appliances are controlled. The user commands over the internet will be obtained by the Wi-Fi modems. The Microcontroller has an interface with this modem. The system status is displayed through the LCD

display, along with the system data. This is a typical IoT based Home Automation system, for controlling all your home appliances.

This paper proposes such system that uses the Google Assistant, the IFTTT (if this than that) application, the Blynk application and the NodeMCU which the main microcontroller and used as major components along with a relay module. Human voice is used to give commands to the Google Assistant. All these components are connected over the internet using Wi-Fi which puts this system under the Internet of Things (IOT).

2. PROPOSED SYSTEM

In this project, nodeMCU ESP8266 is put in with a blynk application account with the IFTTT application then added to google assistant for voice command. voice command control the home appliances. The proposed system eliminates the complication of wiring in case of wired automation. System's operating range is more than the Bluetooth. The existing system doesn't allow the monitoring and controlling of appliances via remote but in the proposed system that using the Wi-Fi based home automation system it allows to monitor and control the appliances. The block diagram of google assistant based home automation shown below:

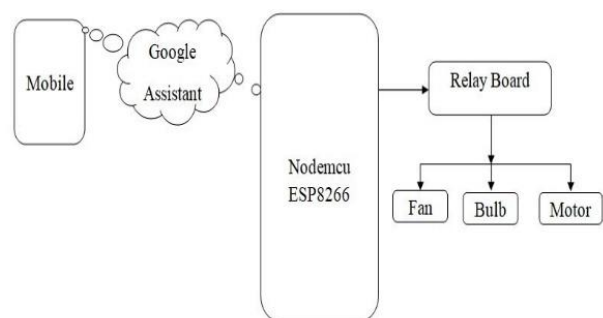


Fig 1: Block diagram of Proposed System

2.1 Explanation of block diagram is as follow

In the block diagram shown in Fig:1, NodeMCU is linked with the Blynk application account with the IFTTT(if this than that) application which is connected to the Google assistance cloud. Home appliances bulb, fan, motor etc. are connected to the nodeMCU.

The system consists of two main categories:

1. **Hardware:** It is the main control unit which has the ability

to control the home appliances like bulb fan etc.

2. Software: It contains IFTTT (if this then that), Blynk application, Google Assistant.

2.1.1 Hardware component description

1. NodeMCU: NodeMCU is an open source lua based firmware and development board specially targeted for IOT based applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The NodeMCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

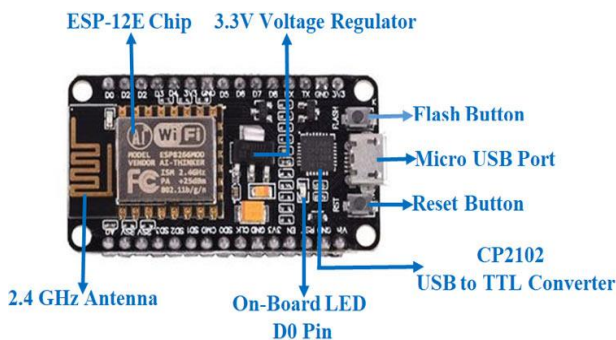


Fig 2: NodeMCU (ESP8266) Board

The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use. Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All we need is the Arduino IDE, a USB cable and the NodeMCU board itself.

2. Relay: The relay is the device that opens or closes the contacts to cause the operation of the other electric control. It detects the undesirable condition with an assigned area and gives the commands to the circuit breaker to disconnect the affected area through ON or OFF.

Every electromechanical relay consists of

1. Electromagnet
3. Mechanically movable contact
3. Switching points and
4. Spring

There are 3 pins in the relay module, these are as follow:

COM: common pin

NO: normally open – there is any contact between the common pin and the normally open pin. So, when you trigger the relay, it connects to the COM pin and power is provided to the load.

NC: Normally closed – there is contact between the common pin and the normally closed pin. There is always connection between the COM and NC pins, even when the relay is turned off. When you trigger the relay, the circuit is opened and there is no supply provided to the load.

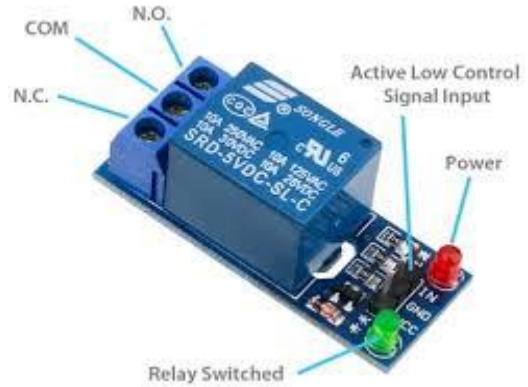


Fig 3: one channel relay module

It works on the principle of an electromagnetic attraction. When the circuit of the relay senses the fault current, it energizes the electromagnetic field which produces the temporary magnetic field. This magnetic field moves the relay armature for opening or closing the connections. The small power relay has only one contact, and the high power relay has two contacts for opening the switch.

2.1.2 Software components description

1. BLYNK: Blynk is a new platform that allows you to quickly build interfaces for controlling and monitoring your hardware projects from your iOS and Android device. After downloading the Blynk app, you can create a project dashboard and arrange buttons, sliders, graphs, and other widgets onto the screen. Using the widgets, you can turn pins on and off or display data from sensors.

Blynk started as a Kickstarter campaign, raising just over \$49k. Currently, Blynk supports most Arduino boards, Raspberry Pi models, the ESP8266, Particle Core, and a handful of other common microcontrollers and single-board computers, and more are being added over time. Arduino Wi-Fi and Ethernet shields are supported, though you can also control devices plugged into a computer's USB port as well.

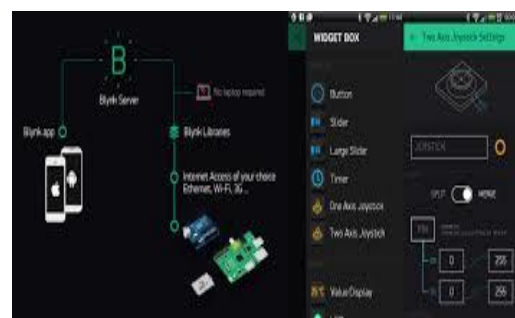


Fig 4: Blynk Application

2. IFTTT (if this then that): IFTTT derives its name from the programming conditional statement “if this, then that.” It provide software platform that connects apps, devices and services from different developers in order to trigger one or more automations involving those apps, devices and services. IFTTT is simple to use. It can be downloading on the mobile app and create a free account and running with automations in minutes. There is a bewildering array of applets available, so IFTTT helpfully provides automation recommendations for new users to try. Its Collections groups together applets for different platforms – such as iOS, Android and voice assistants – and showcases everything from applets for news and weather services to home automation. It is also possible to search for individual applets, or browse under categories such as business tools, connected car or health and fitness. The My Applets screen lets users manage which applets are currently turned on, and provides a history of those that have been used previously. Users can create their own applets by combining various app "services" and setting trigger parameters.

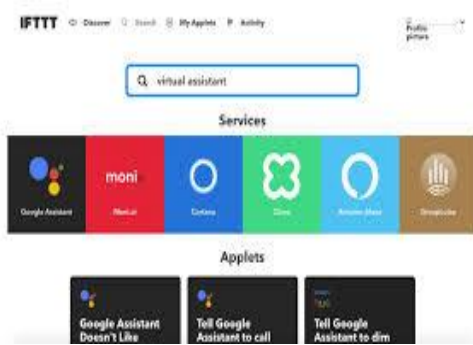


Fig 5: IFTTT (if this then that) application

There are several IFTTT (if this then that) application payment plans designed for either individual or business customers. Individuals have access to a limited free version or to the new Pro tier that launched in September 2020 – if they're willing to pay for additional features and access.

With the introduction of the Pro subscriptions comes a significant change: IFTTT's free tier no longer allows for unlimited applet creation; the maximum is set at three (free users retain unlimited access to community applets, however).users that has created more than three applets will need to either pay for the Pro version or delete some applets.

3. Arduino IDE: Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. The Arduino software is easy-to-

use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

1. Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50.

2. Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

3. Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.

4. Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

3. WORKING

There are mainly four steps involving in the working of the project.

- Step 1. Install and setup the blynk application.
- Step 2. Setting up Arduino ide and blynk library.
- Step 3. Hardware connections
- Step 4. Configure IFTTT

3.1 install and setup the blynk application

1. Create a Blynk Account: After downloading the Blynk App, there is need to create a New Blynk account. This account is separate from the accounts used for the Blynk Forums, in case one has already have one account

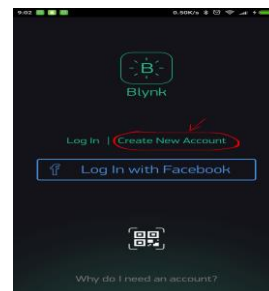


Fig 6: Create Blynk account

An account is needed to save the projects and have access to them from multiple devices from anywhere. It's also a security measure

2. Create a New Project: After successfully logged into the account, start by creating a new project.

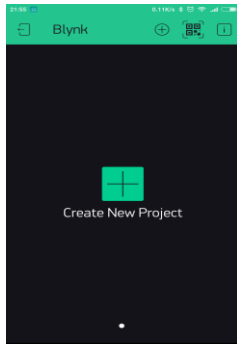


Fig 7: Create new project

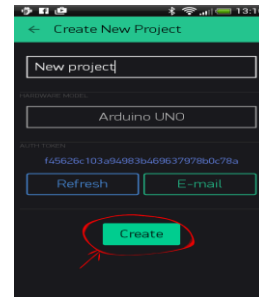
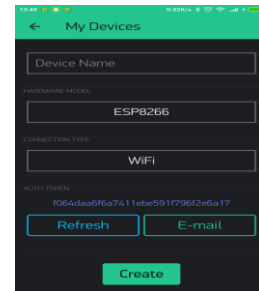


Fig 9: Auth token

3. Choose Your Hardware: Select the hardware model as NodeMCU.

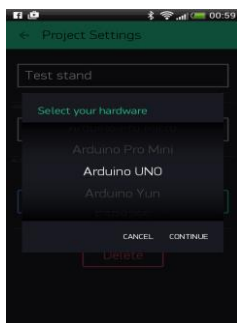
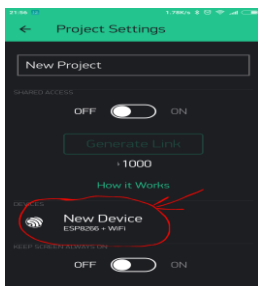


Fig 8: Select hardware

4. Auth Token: Auth Token is a unique identifier which is needed to connect the hardware to the smartphone. Every new project that has been create will have its own Auth Token. The Auth Token will automatically send on the email after project creation. It can also be copied manually. Click on devices section and selected required device:



5. Add a Widget: Add a button to control the LED. Tap anywhere on the canvas to open the widget box. All the available widgets are located here. Now pick a button.

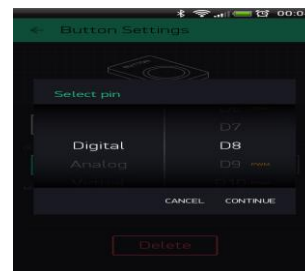
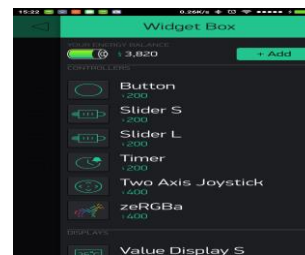
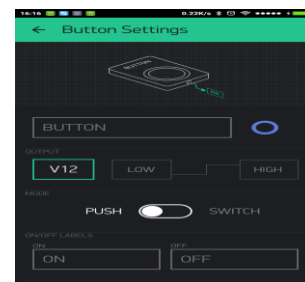


Fig 10: Widgets

6. Run the Project: When all settings are done- press the PLAY button. This will switch the user from EDIT mode to PLAY mode where user can interact with the hardware. While in PLAY mode, user won't be able to drag or set up new widgets, press STOP and get back to EDIT mode.

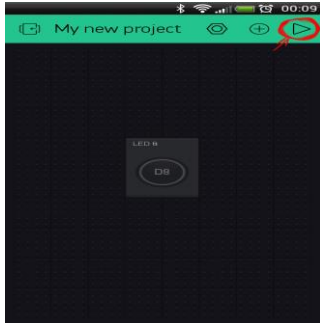


Fig 11: Run the project

3.2 Setting up Arduino ide and blynk library

Go to Arduino website or windows store and install the ide >next step is downloading the blynk library from github >after this,connect the nodemcu to the Pc for uploading the code to it.

Open Arduino ide>go to tools> select the board as nodeMCU 1.0 then select the port.

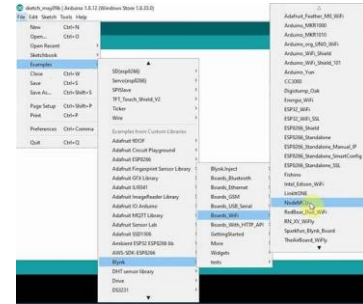
The next step is writing the code.for this go to files>example>selectblynk>selectboard>wifi>choose ESP8266_standalone.

Next step is changing few lines in the code.

Copy the auth token of blynk app in the auth token field>write the name of Wi.Fi n/w and password to connect to nodemcu.

Sketch:

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] =
"CUbRTA4TAMStlE_89NfcuUXNyaer34Sn";
char ssid[] = "Kartik'sHouse";
char pass[] = "Elite0007";
void setup()
{
// Debug console
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
}
void loop()
{
Blynk.run();
}
```



```
ESP8266_Standalone | Arduino 1.6.7
ESP8266_Standalone

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
// Debug console
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
}

void loop()
{
Blynk.run();
}
```

Fig 12 : setting up Arduino ide

Next step is to verify and upload the code to the nodemcu board,

When nodemcu gets power,it will automaticaaly connet to the wi-fi.

3.3 hardware connections

Hardwares req: nodemcu,one channel relay module(active low),1 bulb, Jumper wires,breadboard,Arduino board.

Connections:

1. Attach the nodemcu to the breadboard then connect digital Pin D2 of nodemcu to the input pin of relay board using the Jumper wires, ground pin of nodemuc to the gnd pin of relay.
2. Connect the one end of bulb to the open ends of relay using wire and other end of bulb to the socket to supply power to the bulb.
3. Use data cable to power the nodemcu via PC and to power the relay using Arduino board (can use battery also), connect vcc of relay board to the 3.5v of Arduino board.
4. Attach the bulb to the wire and in the end plug the light bulb to the power socket and turn this on.
5. Tap the play button in the blynk app of any smartphone to test the project, but to control the home appliances using voice commands, this will be done using the platform called IFTTT.

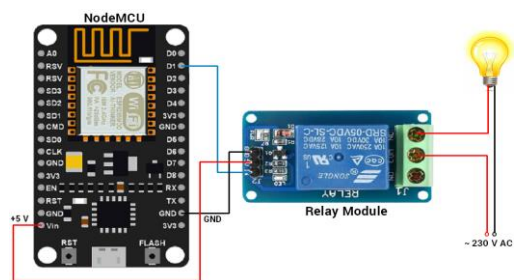


Fig 13: connections

4.4 configure ifttt(is this then that)

1. Open the ifttt.com and sign up using google account>click on my applet>new applet>click on THIS>search google assistant and select it>connect.
- 2.Ifittt will ask the permission to use google account to add commands for assistant>allow>in choose trigger,click on first one which says “say a simple phrase”.fill in the commands that we say to the assistant such as turn on the bulb> finally click on create trigger.this means we are sending request to blynk app to turn on the bulb

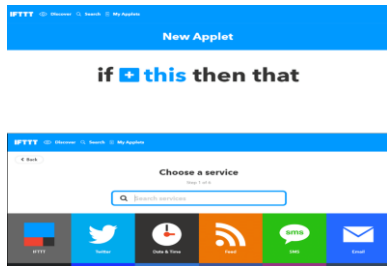


Fig 14: configure IFTTT (is this then that)application

3. Next to program this.click on THAT>search for webhook>connect>click on make a web request>for url field type http://188.166.206.43/auth/token/update/pin(use gpio pin of d2 i.e GPIO4,use D4)>select method as PUT>content type as application/json>body as [{"0"}](0 means to turn on). Similarly make another applet to turn off the bulb.for turning off,use body as [{"1"}]. This is finally done, after this if we say “okay google, turn on the bulb or turn off the bulb”, the bulb will be turn on or off using the commands we will give to the assistant.

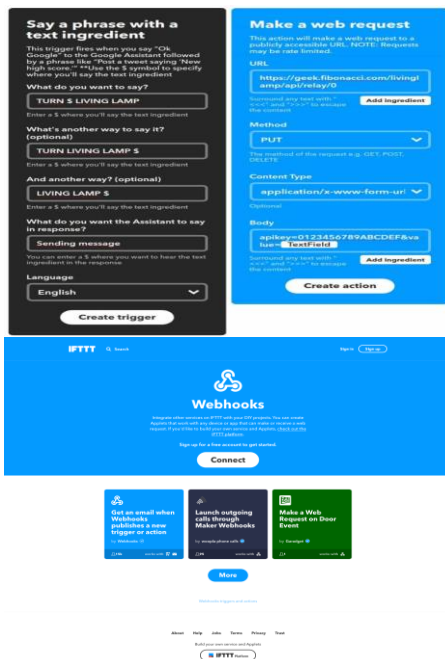


Fig 15: applets

4. RESULT

The result of the project was positive and the system responded well. The demo shown below is the complete implementation of the proposed system.



Fig 16: bulb turn on and off by blynk app

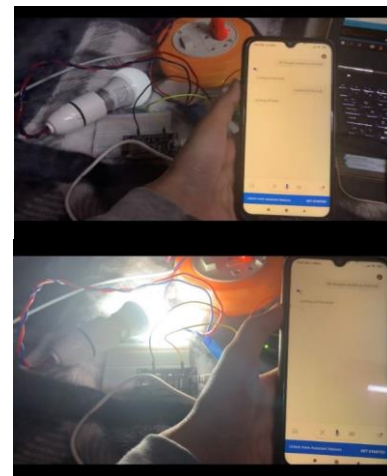


Fig 17: bulb turn on and off by Google assistant

APPLICATIONS

1. 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.
2. Lighting control system: a "smart" network that incorporates communication between various lighting system inputs and outputs, using one or more central computing devices.
3. Home automation for the elderly and disabled.
4. Smart Kitchen and Connected Cooking. Using voice

control devices like Google Home to kitchen appliances.

5. CONCLUSION

The aim of the proposed system was to implement a low cost, reliable and scalable google assistant based voice controlled home automation system that can be used to switch on/off any home appliances. The purpose of this project is to make homes simpler, better, or more accessible. This system is highly reliable and efficient for the elderly folks in the house and differently abled person on a wheel chair who are dependent on others and cannot reach the switch for the turning on/off.

FUTURE SCOPE

The future scope for google assistant based home automation can be huge. There are many factors that make the project more reliable, 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, AC etc. Well, no system is ever perfect. It always has a scope for improvement. One just needs to put efforts and make the system better.

REFERENCES

1. Q. F. Hassan, "Implementing the Internet of Things for Renewable Energy", *Internet of Things A to Z: Technologies and Applications* IEEE, 2018.
2. Q. Wang and Y. G. Wang, "Research on Power Internet of Things Architecture for Smart Grid Demand", 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), pp. 1-9, 2018.
3. R. Sinha, S. Patil, L. Gomes and V. Vyatkin, "A survey of static formal methods for building dependable industrial automation systems", *IEEE Trans. Ind. Informat.*, vol. 15, pp. 3772-3783, Jul. 2019.
4. R. K. Kodali and S. Soratkal, "MQTT based home automation system using ESP8266", *Proc. IEEE Region 10 Humanitarian Technology. Conf. (R10-HTC)*, pp. 1-5, Dec. 2016.
5. Tan, Lee and Soh – "Internet based Monitoring of Distributed Control Systems", - *Energy and power Engineering*. Publisher: IEEE Transactions on Education, Place: New Jersey, Country: USA, Year: 2002, Vol: 45, Iss. No. 2., pp. 128-134.
6. Gill, K., Yang, S.-H., Yao, F., Xin, L.: A zigbee-based home automation system. *IEEE Trans. Consum. Electron.* 55(2), 422–430 (2009)
7. Gill, K., Yang, S.-H., Yao, F., Xin, L.: A zigbee-based home automation system. *IEEE Trans. Consum. Electron.* 55(2), 422–430 (2009)
8. Kang, B., Park, S., Lee, T., Park, S.: IOT based monitoring

system using tri-level context making model for smart home services. In: *IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, pp. 198–199 (2015)

9. <https://ieeexplore.ieee.org/document/8846205>

10. <https://ieeexplore.ieee.org/xpl/conhome/8767165/proceeding>

11. *Proceedings of First International Conference on Computing, Communications, and Cyber-Security*