

# IoT based Smart Home

Naresh Poloju

Research Scholar, Department of Information and Communication Engineering, Anna University, Chennai, India

\*\*\*

**Abstract**--Home automation can be used for maintaining hassle free living conditions within a home. Because of its ability to ensure security and ease in access, home automation is gaining more popularity day by day. This paper proposes IOT based Smart Home application used to remotely control the home appliances. In the system NodeMCU will be interfaced with Relay Module and SMART HOME android application installed on the smartphone.

**Index Terms:** NodeMCU, ESP8266, Home Automation, Smart Home, Internet of Things

## I. INTRODUCTION

Internet of Things (IoT) is one of the most upcoming technologies which can be used for managing and controlling any object by connecting it to the internet [1]. IoT can be used in various applications of automation where automation is the process of operating or controlling various applications or equipment with less or no human intervention [2].

Automation can be categorized depending on their applications such as home automation, industrial automation, building automation, etc. The complexity of life has significantly reduced with the evolution in automation technology. Everyday, manual systems are being replaced by automatic systems [3]. With the ever increasing internet users over the past few years, it has become a part of everyone's life [3]. IoT is the latest emerging internet technology. This project is about home automation using Internet of Things.

Smart Home application can turn on or turn off the home appliances remotely through the internet or by setting local WiFi server of ESP8266.

## 2. NodeMCU

NodeMCU is an open-source firmware based on LUA developed with the ESP8266 WiFi chip. NodeMCU hardware design is open for edit/modify/build. The NodeMCU board is shown in the Fig 1.



Figure 1: NodeMCU Development Board (Version2)

NodeMCU Development board is loaded with digital pins, analog pins, WiFi, and serial communication protocols. We can code NodeMCU with ESPlorer IDE using Lua scripts or we can develop applications on NodeMCU using the Arduino development environment.

## 3. RELAY MODULE

High voltage electronic devices are operated using relays. A Relay is a switch which is electrically operated by an electromagnet. The electromagnet is activated with a low voltage, for the NodeMCU 3.3V current is used to operate Relay Module. The Fig.2 shows the 1-channel Relay module.



Figure 2: 1-channel Relay Module

The VCC, GND, IN pins are connected to the NodeMCU. On the other side of Relay we have ports for connecting appliances.

## 4. PROPOSED SYSTEM ARCHITECTURE

The design for the proposed IOT Based Smart Home system is given in Fig. 3.

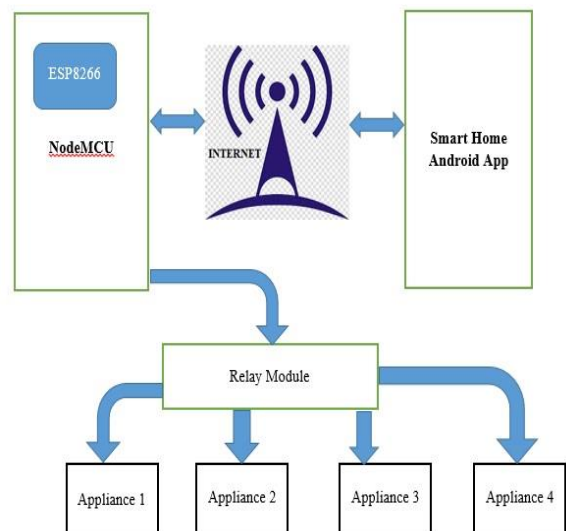


Fig 3: System Architecture

As shown in the figure, the NodeMCU has an integrated wifi module called ESP8266. Using this wifi module webserver and client can be created. Using this server NodeMCU will communicate with android application. From the Smart Home Android application appliances on/off request will be forwarded to the NodeMCU from which relay module will be controlled. When the user is outside the home then internet can be used as medium of communication. If the user is inside the home, then NodeMCU could be acted as server, to which smartphone can be connected through WiFi.

Connection from NodeMCU to the Relay Module is shown in the Fig. 4

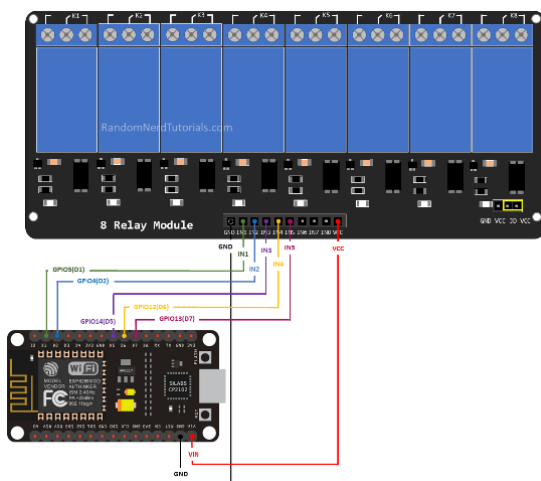


Figure 4: NodeMCU to Relay Module Connection

## 5. IMPLEMENTATION

Initially NodeMCU will be equipped with the code for setting up WiFi client/server and the code for managing the Relay module. In the system 8 channel Relay Module was used, but for testing purpose, only 4 channels are utilized. Arduino IDE was used to write this code.

### 5.1 Code For NodeMCU

```
#include <ESP8266WiFi.h>
#ifndef STASSID
#define STASSID "honor"
#define STAPSK "nani@4800"
#endif
void send_data(String x, WiFiClient client){
  client.print(F("HTTP/1.1 200 OK\r\nContent-
Type: text/html\r\n\r\n\r\n\r\n\r\n\r\n"));
  client.print(x);
  client.print(F(""));
}
const char* ssid = STASSID;
const char* password = STAPSK;
// Create an instance of the server and specify
the port to listen on as an argument
WiFiServer server(80);
void setup() {
  Serial.begin(115200);
```

```
// prepare LED
pinMode(LED_BUILTIN, OUTPUT);
pinMode(16, OUTPUT);
pinMode(5, OUTPUT);
pinMode(4, OUTPUT);
pinMode(0, OUTPUT);
digitalWrite(16, LOW);
digitalWrite(5, LOW);
digitalWrite(4, LOW);
digitalWrite(0, LOW);
digitalWrite(LED_BUILTIN, 0);

// Connect to WiFi network
Serial.println();
Serial.println();
Serial.print(F("Connecting to "));
Serial.println(ssid);

WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  digitalWrite(LED_BUILTIN, HIGH); // LED on
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // LED off
  delay(1000);
  Serial.print(F("."));
}
Serial.println();
Serial.println(F("WiFi connected"));

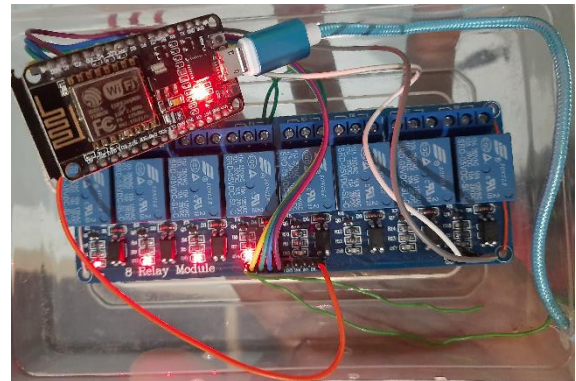
// Start the server
server.begin();
Serial.println(F("Server started"));
digitalWrite(LED_BUILTIN, LOW);
// Print the IP address
Serial.println(WiFi.localIP());
}
void loop() {
  // Check whether a client is connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  while(!client.available()){ delay(1); }
  Serial.println(F("new client"));
  client.setTimeout(5000); // default is 1000
  // Read the request first line
  String req = client.readStringUntil('\r');
  Serial.println(F("request: "));
  Serial.println(req);
  // Match the request
  String val;
  if (req.indexOf(F("/1ON")) != -1) {
    val = "RELAY1 ON";
    digitalWrite(16, LOW);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(400);
```

```

    digitalWrite(LED_BUILTIN, LOW);
    send_data(val,client);
}
else if (req.indexOf(F("/1OFF")) != -1) {
    val = "RELAY1 OFF";
    digitalWrite(16,LOW);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(400);
    digitalWrite(LED_BUILTIN, LOW);
    send_data(val,client);
}
else if (req.indexOf(F("/2ON")) != -1) {
    val = "RELAY2 ON";
    digitalWrite(5,LOW);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(400);
    digitalWrite(LED_BUILTIN, LOW);
    send_data(val,client);
}
else if (req.indexOf(F("/2OFF")) != -1) {
    val = "RELAY2 OFF";
    digitalWrite(5,HIGH);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(400);
    digitalWrite(LED_BUILTIN, LOW);
    send_data(val,client); }
else if (req.indexOf(F("/3ON")) != -1) {
    val = "RELAY3 ON";
    digitalWrite(4,LOW);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(400);
    digitalWrite(LED_BUILTIN, LOW);
    send_data(val,client);
}
else if (req.indexOf(F("/3OFF")) != -1) {
    val = "RELAY3 OFF";
    digitalWrite(4,HIGH);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(400);
    digitalWrite(LED_BUILTIN, LOW);
    send_data(val,client); }
else if (req.indexOf(F("/4ON")) != -1) {
    val = "RELAY4 ON";
    digitalWrite(0,LOW);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(400);
    digitalWrite(LED_BUILTIN, LOW);
    send_data(val,client); }
else if (req.indexOf(F("/4OFF")) != -1) {
    val = "RELAY4 OFF";
    digitalWrite(0,HIGH);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(400);
    digitalWrite(LED_BUILTIN, LOW);
    send_data(val,client); }
Serial.println(F("Disconnecting from client"));
}

```

Once this code is loaded into the NodeMCU kit, we can connect Relay Module with NodeMCU as shown in Fig 5. Each Relay of the Relay module will be connected to one appliance.



**Figure 5: Working setup of the system**

After setting up the system Smart Home Android app is used to communicate with the NodeMCU for managing the appliances. In order to communicate with the intended NodeMCU, the IP address of the NodeMCU was embedded in the Smart Home application. Smart Home application is provided with the convenience of changing and resetting the IP address. Change IP address is used to update the old IP address. Reset IP is used to set the IP address to the initial IP address. The Smart Home application screen shots are given below.



## 6. CONCLUSION

In this paper smart home system was designed and implemented using NodeMCU as microcontroller and WiFi as method of communication with the appliances which enables the remote accessing to the user. This will help in minimizing the hassles in operating the appliances and saves lot of human time

## 7. FUTURE SCOPE

This work can be enhanced by attaching smoke sensors to detect the gas leakage in the home and motion sensors to detect the intruders when the user is not in the home.

## REFERENCES

- [1] Silviu Folea, Daniela Bordenca, Casiana Hotea, and Honoriu Valean, "Smart home automation system using Wi-Fi low power devices,"
- [2] Wireless Home Automation [Online], Available : <https://www.elprocus.com/wireless-home-automation-using-internet-of-things/>
- [3] Vinay sagar K N and Kusuma S M, "Home automation using Internet of Things," in International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 03 | June-2015, e-ISSN: 2395-0056, p-ISSN: 2395-0072
- [4] Internet Of Things with ES8266 by Marco Schwartz, PAKT Publishing, 2016
- [5] Android Programming: The Big Nerd Ranch Guide by Bill Phillips, Chris Stewart, Kristin Marsicano
- [6] Head First Android Development 2e: A Brain-Friendly by Dawn Griffiths , David Griffiths, Orielly, 2017
- [7] Read about Arduino mega 2560 [Online], Available : <https://www.arduino.cc/en/Main/arduinoBoardMega2560>
- [8] Wireless Home Automation [Online], Available : <https://www.elprocus.com/wireless-home-automation-using-internet-of-things/>
- [9] Read about relay [online], Available : <https://www.seeedstudio.com/Grove-Relay-p-769.html>
- [10] Arduino Software IDE [Online], Available : <https://www.arduino.cc/en/Main/Software>