

Electronics Engineering
Project
Home Automation using NodeMCU
4th Semester



Submitted to: Engr. Ali Hassan

Section: C

Name	CMS	Report	Performance	Viva
Zuhaib Chaudhary	373070			
Adil Saleem	372066			
Muhammad Ahmad	367558			
Ifra Irshad	393718			

School of Mechanical and Manufacturing Engineering

ABSTRACT:

This project explores the advancements in consumer electronics and embedded systems, specifically focusing on the control of electrical appliances in homes through a mobile application. The objective is to design an automated system that enables remote control and automation using a NodeMCU microcontroller board and a Wi-Fi module. The system utilizes the NodeMCU, based on the ESP8266 Wi-Fi module, to establish wireless communication between the mobile application and the control unit. By eliminating the need for physical cables, users can conveniently control their home appliances using their smartphones. The design emphasizes cost-effectiveness, user-friendliness, and easy installation. The NodeMCU interfaces with the Wi-Fi module and controls the switching of electrical appliances via relays. The mobile application provides an intuitive interface for users to interact with and manage their home appliances. Overall, this project showcases the potential of integrating NodeMCU and Wi-Fi technology with a mobile application to create a connected home environment. The wireless capabilities of Wi-Fi enable efficient remote control and automation of electrical appliances, offering convenience and flexibility to users

Introduction:

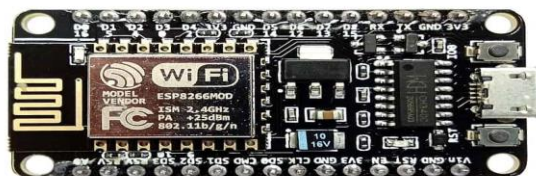
Automation is the most frequently spelled term in the field of electronics. The hunger for automation brought many revolutions in the existing technologies. These had greater importance than any other technologies due to their user-friendly nature. These can be used as a replacement for the existing ones. Switches in home which produces sparks and results in fire accidents in few situations. The concept of home automation has gained immense popularity in recent years, enabling homeowners to control and automate various aspects of their homes for enhanced convenience, energy efficiency, and security. One of the key components of a home automation system is the integration of microcontrollers, sensors, and actuators to create an intelligent network that can be controlled remotely. In this introduction, we will explore the fundamental components of a home automation system, including:

- NodeMCU ESP8266
- Relay 5V
- Regulator IC
- Diode 1N4007
- NPN Transistor
- Terminal Block

NodeMCU ESP8266:

The Node MCU ESP8266 is a highly versatile and cost-effective Wi-Fi-enabled microcontroller board that serves as the brain of the home automation system. It is based on the ESP8266 chip and provides a robust platform for building IoT (Internet of Things) applications.

The Node MCU ESP8266 offers built-in Wi-Fi connectivity, allowing seamless integration with other smart devices and internet-based services.



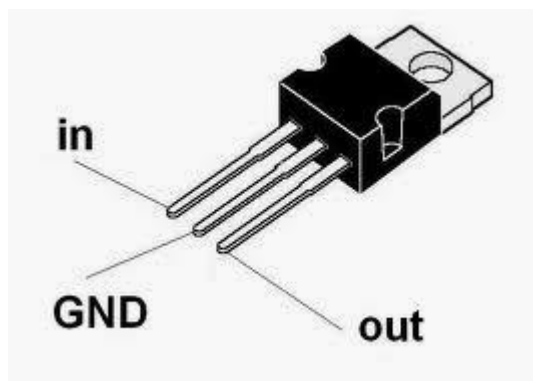
Relay 5V:

The relay 5V is an electromechanical switch that is commonly used in home automation systems to control high-power devices such as lights, fans, and appliances. It acts as a bridge between the low-voltage control signals from the microcontroller and the higher voltage requirements of the connected devices. By utilizing relays, the microcontroller can easily turn these devices on or off remotely, offering enhanced control and automation capabilities.



Regulator IC:

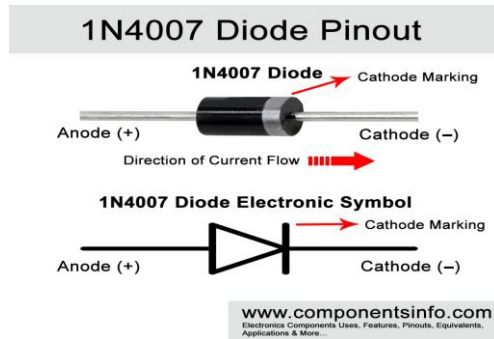
The 7805-voltage regulator IC is a popular component used in electronic circuits to regulate and stabilize the supply voltage. In a home automation system, it ensures a consistent and reliable power supply to the microcontroller and other components, protecting them from voltage fluctuations or surges. The 7805 IC specifically regulates the input voltage to a stable 5 volts, which is typically required by microcontrollers such as the NodeMCU ESP8266.



Diode 1N4007:

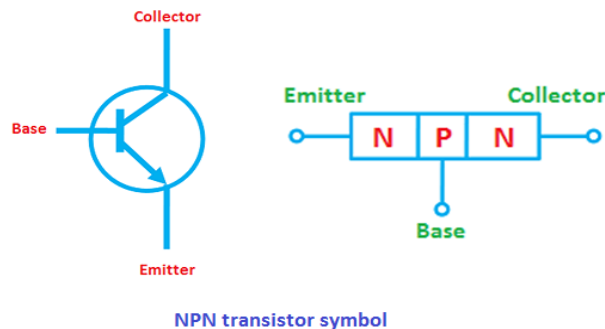
The diode 1N4007 is a general-purpose rectifier diode that is frequently used in electrical circuits to allow current flow in one direction while blocking it in the

opposite direction. In a home automation system, the diode protects the components from reverse current or voltage spikes, safeguarding them against potential damage. It is often connected in parallel with inductive loads like relays or motors to prevent back EMF (electromotive force) from affecting the other components.



NPN Transistor:

The NPN transistor BC547 is a widely used bipolar junction transistor (BJT) that serves as a switch or amplifier in electronic circuits. In a home automation system, the BC547 transistor can be utilized to control the flow of current through the relay, allowing the microcontroller to turn the connected devices on or off. By using the transistor as a switching element, the microcontroller can handle higher current loads without directly drawing power from its pins.



Lastly, the terminal block is a connection interface that facilitates the easy and secure connection of wires and cables in a home automation system. It provides a convenient way to connect the various components, such as the microcontroller, relay, and power supply, ensuring proper electrical connections and minimizing the risk of loose or faulty connections.

Literature Review:

- **Wi-Fi-based Home Automation System:**
- The Wi-Fi-based home automation system described in the review employs an Arduino

board and a Wi-Fi module for wireless communication between home appliances and a mobile phone. The Arduino program, written in a high-level interactive language like C, enables the control and status monitoring of devices through the established Wi-Fi connection. The system also incorporates password protection to ensure authorized access to the appliances. The feedback from the mobile phone indicates the status of the devices, allowing users to remotely monitor and control them.

- b. GSM-based Home Automation System using Cell Phones:
- The GSM-based home automation system utilizes mobile phones and Global System for Mobile Communication (GSM) technology for communication. Various options such as SMS-based, GPRS-based, and DTMF-based communication methods are considered. The system architecture involves home sensors and devices connected to the home network, communicating through GSM and SIM (subscriber identity module). Transducers are used to convert machine functions into electrical signals, which are then processed by a microcontroller. The sensors in the system convert physical qualities such as sound, temperature, and humidity into voltage signals. The microcontroller analyzes these signals and sends commands to the GSM module for communication and control purposes.

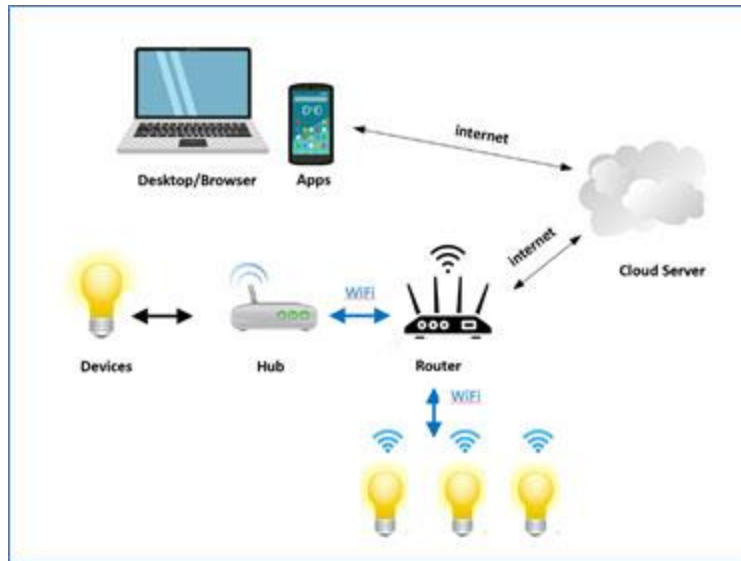
Home automation wifi through common wifi.

Wi-Fi based home automation system mainly consist of three modules, the server, the hardwareInterface module, and the software package.

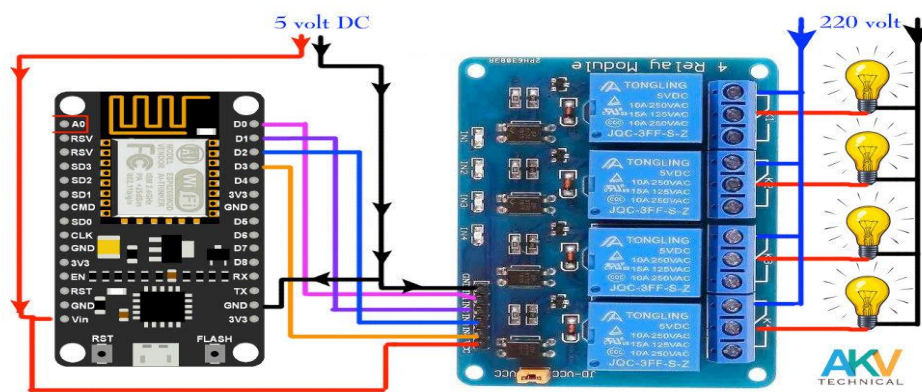
The figure below shows the system model layout. Wi-Fi technology is used by server, and Hardware Interface module to communicate with each other. The same technology is used to login to the server web-based application. The server is connected to the internet, so remote users can access server web-based application through the internet using compatible web browser. Software of the latest home automation system is split to server application software, and Microcontroller (Arduino) firmware. The Arduino software, built using C language, using IDE comes with the microcontroller itself.

Arduino software is culpable for gathering events from connected sensors, then applies action preprogramed in the server. Another job is to report and record the history in the server database.

The server application software package for the proposed home automation system, is a web based application built using asp.net. The server application software can be accessed from internal network or from internet if the server has real IP on the internet using any internet navigator supports asp.net technology. Server application software is culpable of, maintain the whole home automation system, setup, configuration.



Circuit Diagram:



Arduino IDE Code:

```
#include <ESP8266WiFi.h>

const char* ssid = "Ahmad";    //enter your wi-fi name

const char* password = "Vinrock099";    //enter the wifi password

unsigned char status_led=0;

WiFiServer server(80);

#define Relay1 12 //D6
int value1;
```

```
#define Relay2 14 //D2
int value2;

#define Relay3 4 //D1
int value3;

#define Relay4 5 //D5
int value4;

void setup() {
    Serial.begin(115200);
    pinMode(Relay1,OUTPUT);
    pinMode(Relay2,OUTPUT);
    pinMode(Relay3,OUTPUT);
    pinMode(Relay4,OUTPUT);

    digitalWrite(Relay1,HIGH);
    digitalWrite(Relay2,HIGH);
    digitalWrite(Relay3,HIGH);
    digitalWrite(Relay4,HIGH);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }
    server.begin();
    Serial.println(WiFi.localIP());
}

void loop() {

    WiFiClient client = server.available();

    if (!client) {

        return;

    }

    while (! client.available())

    {

        delay (1);
```



```
}
String req = client.readStringUntil('\r');
client.flush();
if (req.indexOf("/Relay1OFF") != -1) {

digitalWrite(Relay1,LOW);
    Serial.println("Relay 1 OFF");

}

else if(req.indexOf("/Relay1ON") != -1)

{

digitalWrite(Relay1,HIGH);
    Serial.println("Relay 1 ON");

}
    if (req.indexOf("/Relay1OFF") != -1) {

digitalWrite(Relay1,LOW);
    Serial.println("Relay 1 OFF");

}

else if(req.indexOf("/Relay1ON") != -1)

{

digitalWrite(Relay1,HIGH);
    Serial.println("Relay 1 ON");

}
    if (req.indexOf("/Relay2OFF") != -1) {

digitalWrite(Relay2,LOW);
    Serial.println("Relay 2 OFF");

}

else if(req.indexOf("/Relay2ON") != -1)
```

```
{  
  
    digitalWrite(Relay2,HIGH);  
        Serial.println("Relay 2 ON");  
  
}  
if (req.indexOf("/Relay3OFF") != -1)    {  
  
digitalWrite(Relay3,LOW);  
        Serial.println("Relay 3 OFF");  
  
}  
  
else if(req.indexOf("/Relay3ON") != -1)  
  
{  
  
    digitalWrite(Relay3,HIGH);  
        Serial.println("Relay 3 ON");  
  
}  
  
    if (req.indexOf("/Relay4OFF") != -1) {  
digitalWrite(Relay4,LOW);  
        Serial.println("Relay 4 OFF");  
  
}  
  
else if(req.indexOf("/Relay4ON") != -1)  
  
{  
  
    digitalWrite(Relay4,HIGH);  
        Serial.println("Relay 4 ON");  
  
}  
  
client.println("HTTP/1.1 200 OK");  
  
client.println("Content-Type: text/html");  
  
client.println("Connection: close");
```

```
client.println("");

client.println("<!DOCTYPE HTML>");

client.println("<HTML>");

client.println("<H1> HOME AUTOMATION </H1>");

client.println("<br />");


client.println("<a href=\"Relay1ON\\\"> <button style='FONT-SIZE: 40px; color: red; HEIGHT: 200px; align: center; WIDTH: 200px; 126px; Z-INDEX: 0; TOP: 200px;'> LED 1 OFF </button> </a>");

client.println("<a href=\"Relay1OFF\\\"> <button style='FONT-SIZE: 40px; color: green; HEIGHT: 200px; align: center; HEIGHT: 200px; WIDTH: 300px; 126px; Z-INDEX: 0; TOP: 200px;'> LED 1 ON </button> </a><br>");

client.println("<a href=\"Relay2ON\\\"> <button style='FONT-SIZE: 40px; color: red; HEIGHT: 200px; align: center; WIDTH: 200px; 126px; Z-INDEX: 0; TOP: 200px;'> LED 2 OFF </button> </a>");

client.println("<a href=\"Relay2OFF\\\"> <button style='FONT-SIZE: 40px; color: green; HEIGHT: 200px; align: center; HEIGHT: 200px; WIDTH: 300px; 126px; Z-INDEX: 0; TOP: 200px;'> LED 2 ON </button> </a><br>");

client.println("<a href=\"Relay3ON\\\"> <button style='FONT-SIZE: 40px; color: red; HEIGHT: 200px; align: center; WIDTH: 200px; 126px; Z-INDEX: 0; TOP: 200px;'> FAN OFF </button> </a>");

client.println("<a href=\"Relay3OFF\\\"> <button style='FONT-SIZE: 40px; color: green; HEIGHT: 200px; align: center; HEIGHT: 200px; WIDTH: 300px; 126px; Z-INDEX: 0; TOP: 200px;'> FAN ON </button> </a><br>");

client.println("<a href=\"Relay4ON\\\"> <button style='FONT-SIZE: 40px; color: red; HEIGHT: 200px; align: center; WIDTH: 200px; 126px; Z-INDEX: 0; TOP: 200px;'> DOOR CLOSE </button> </a>");

client.println("<a href=\"Relay4OFF\\\"> <button style='FONT-SIZE: 40px; color: green; HEIGHT: 200px; align: center; HEIGHT: 200px; WIDTH: 300px; 126px; Z-INDEX: 0; TOP: 200px;'> DOOR OPEN </button> </a>");
```

```
client.println("</html>");  
delay(1);  
}
```

Conclusion:

Nowadays, Android stands as a powerful open-source operating system, offering incredible flexibility to incorporate various functionalities according to our needs. By utilizing Android devices, we can easily create a remarkable home automation system at an affordable cost, improving both home security and convenience. It is a user-friendly solution that enhances the overall comfort of our homes.

Consequently, the idea of developing a comprehensive home automation system with remote control and monitoring capabilities has finally come to fruition. The outcome of this project is an impressive array of home appliances that can be conveniently controlled through a smartphone connected via a Bluetooth module. The successful completion of this project is attributed to the dedicated efforts of all project members and the invaluable assistance received from others.

Engaging in this project has been an enriching experience, providing us with valuable opportunities to learn and experiment. We have gained firsthand knowledge about the intricate process involved in designing and developing home automation systems. We are immensely proud to have explored this fascinating topic as our major project, effectively creating our own version of a home automation system. It has allowed us to closely engage with a rapidly advancing field of study and research that has the potential to revolutionize people's lifestyles in the near future.

References:

<https://www.hackster.io/Sanjus-Robotics-Studio/home-automation-using-google-firebase-4dcddf>

<https://www.youtube.com/watch?v=Z3YdtMsGmjE>

<https://www.youtube.com/watch?v=tOKTNex2wwA>

<https://thingspeak.com>

<https://how2electronics.com/home-automation-using-google-firebase-nodemcu-esp8266/>