

DIY IoT Hardware Project

IoT Based Smart Home Automation

Introduction

Anand Trehan

180108006

IIT Guwahati

Table of Contents

Introduction	3
Main Objectives	4
Implemented Attributes	5
Hardware Requirements	6
Schematic.....	7
Sample Outputs and Demo video	8-10
Codes.....	11-25

DIY IoT Hardware Project
IoT Based Smart Home Automation
Introduction

Smart homes are a natural extension of current information, electronic, and communication technologies. During the past years, its concept mainly refers to comfort, leisure and healthcare. Through these smart home systems, remote meters reading can be achieved. That is to say, the data associated with home water, power, gas, and telecommunications can be transmitted to the corresponding utility company automatically in order to improve the work efficiency. Furthermore, by virtue of smart home systems, the home ventilation, doors and windows, air conditioning, and lighting can be manipulated by remote control. And each electrical device like refrigerators, cooking devices and washing machines can be controlled by programs or remote platforms. Entertainment equipments such as televisions and radios can be connected to remote common channels. In addition, home healthcare and security are also important aspects of smart homes. For example, health assistance devices can help an elder individual to transmit request or alarm to a professional medical center or a family member. Therefore, a number of smart homes have been developed [1-6] in the past years. In each case the house and its various electrical appliances have been fitted with sensors, actuators. And the home devices operate in a local network, which is sometimes connected to a remote management platform for data collection and processing.

Main Objective

IOT is used in order to control home appliances, thus automating modern homes through the internet. Home Automation system (HAS) using IoT is a system that uses computers or mobile devices to control basic home functions and features automatically through internet from anywhere around the world. It is meant to save the electric power and human energy.

In this project we aim to achieve a few simple objectives which are:-

- Building a System that uses 3 loads which represents house lighting and fan
- Using a proximity sensor to automate lighting
- Using a temperature and humidity sensor to automate the working of a fan
- Controlling the system using a Web Application
- Displaying System Status on an LCD display

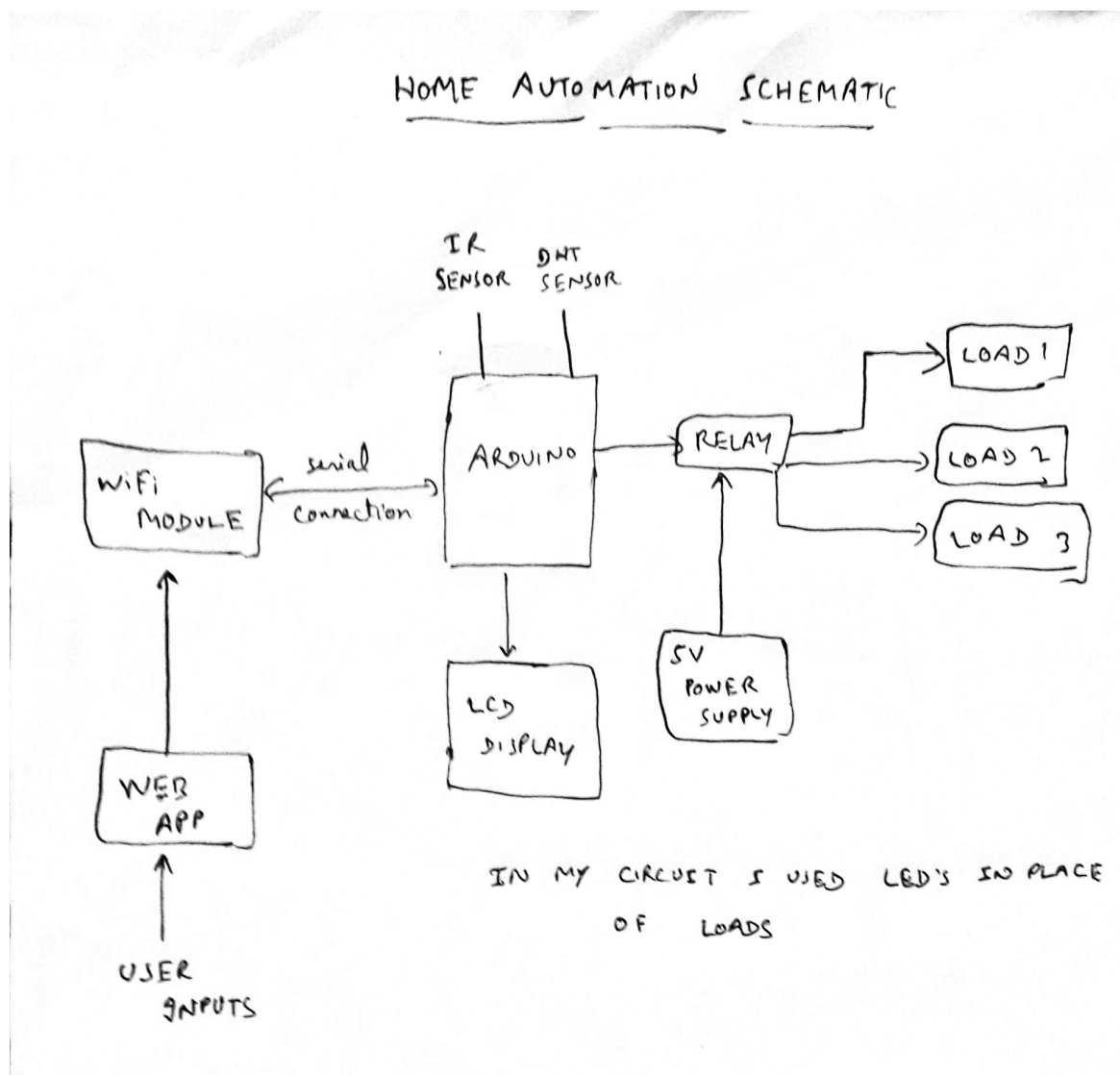
Implemented Attributes

- Controlling the system using a Web Application via NodeMCU wifi module.
- Serial communication between Arduino UNO microcontroller and NodeMCU to control the system.
- Automated Lighting Control through a proximity sensor.
- Automating a Fan using a temperature and humidity Sensor
- Displaying system status on an external LCD display
- Using a Relay to switch Loads
- Added a Feature where a user can switch on or off the automation features

Hardware Requirements

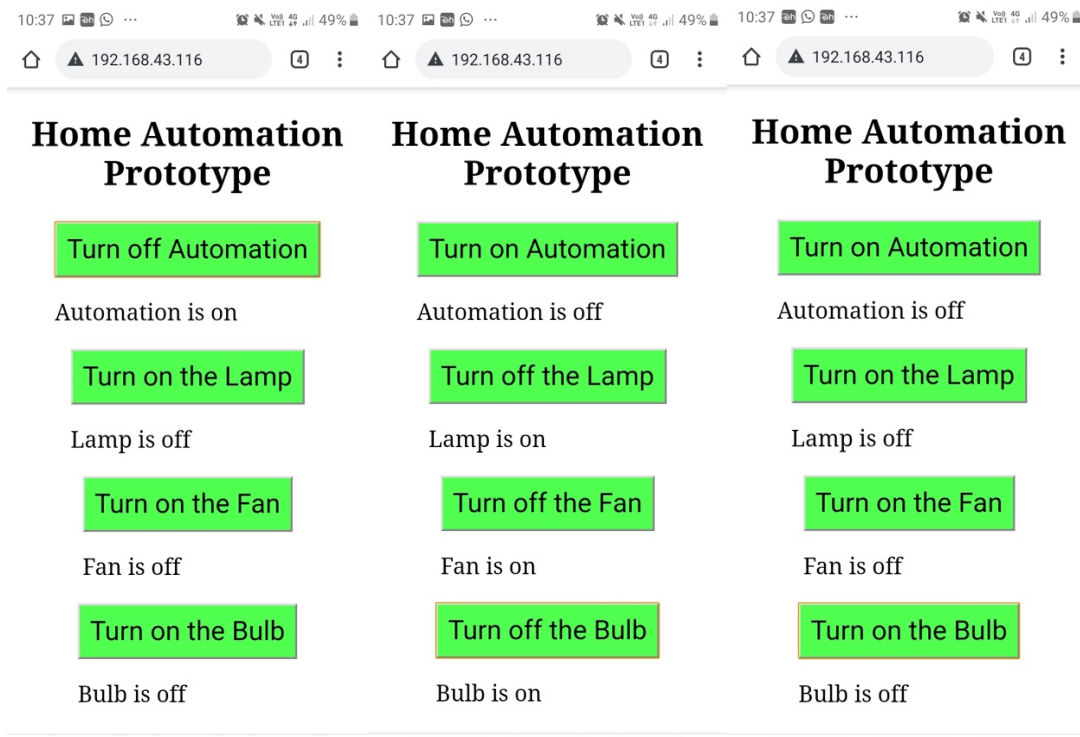
- DHT11 Temperature and Humidity Sensor
- IR Proximity Sensor
- ESP8266 based WiFi Module
- Arduino UNO microcontroller
- 20x04 LCD display
- 4 channel 5V relay module
- LEDs
- Jumper Wires
- Breadboards
- Power Supply

Circuit Schematic

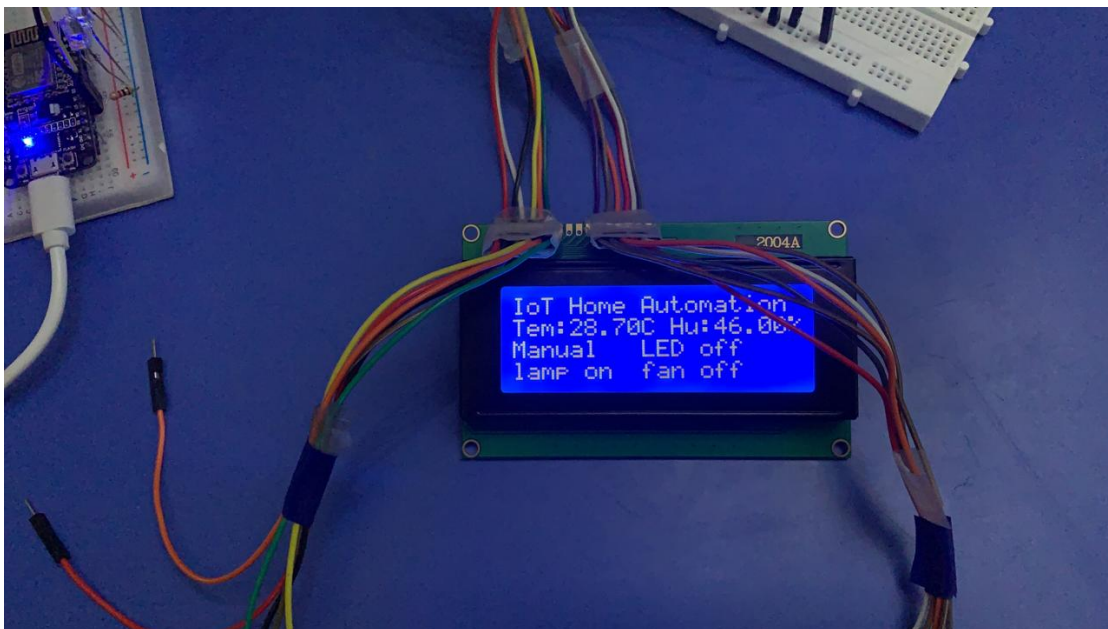


Sample Outputs

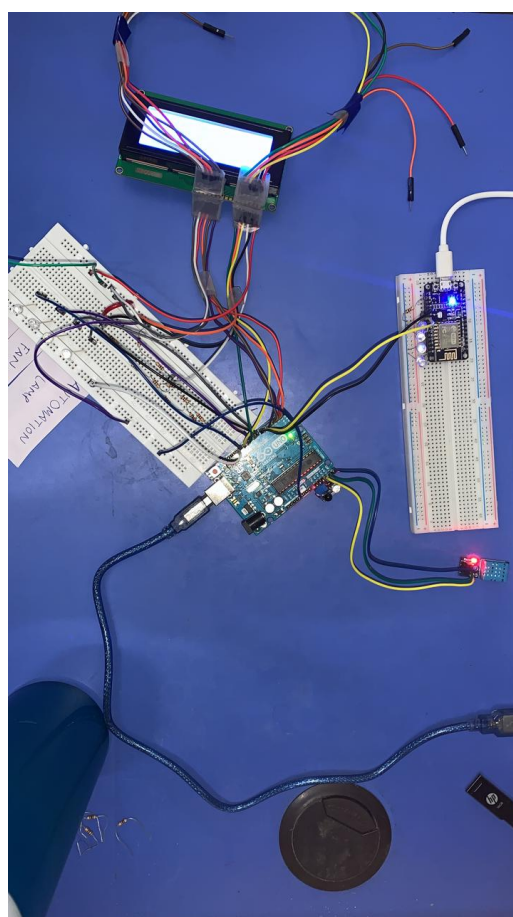
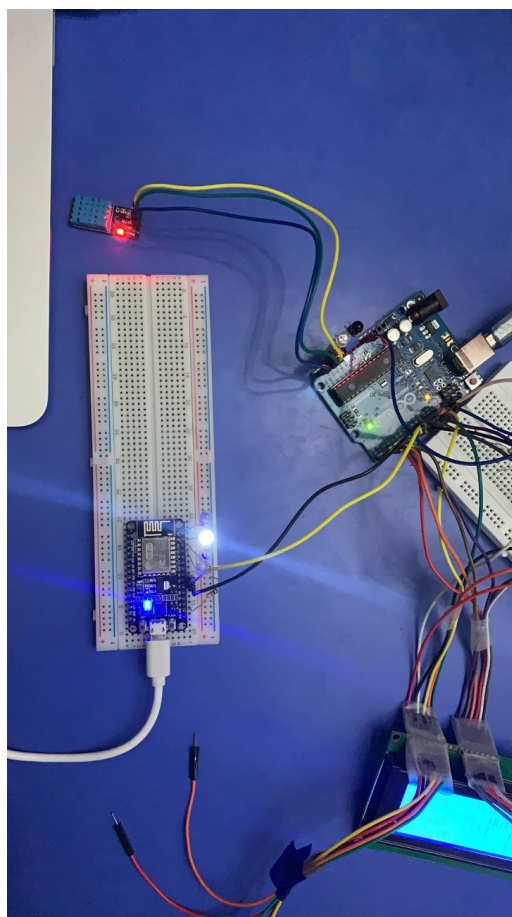
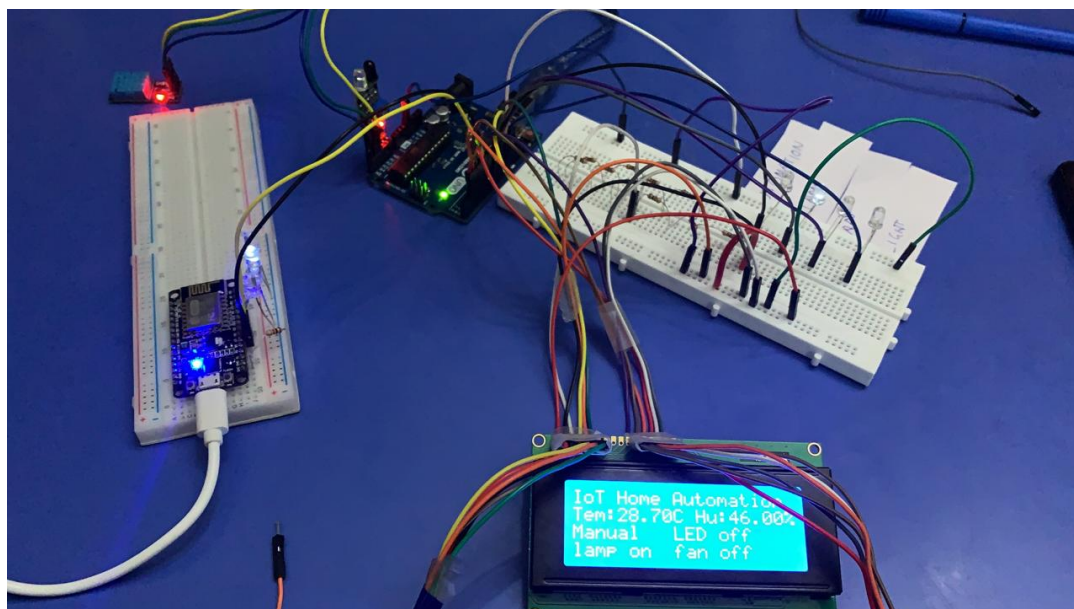
Web Application:



LCD Display:



Circuit:



Codes

Code to be uploaded to Arduino:

```
#include <ArduinoJson.h>
#include <SoftwareSerial.h>

#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN A4
#define DHTTYPE DHT11
// #define fan_out 11
// #define lamp_out 10
DHT dht(DHTPIN,DHTTYPE);
SoftwareSerial s(7,6);
bool status = false;
unsigned long read_ir;
unsigned long read_dht;
// unsigned long send_data;

float h;
float t;
float hic;
int Auto; //8
int lamp; //9
int fan; //10
int light; //13
// int data3;

#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {

  read_ir=0;
  read_dht=0;
  // send_data=0;

  pinMode(A2,INPUT);
  pinMode(A0,OUTPUT);
  pinMode(A1,OUTPUT);
  pinMode(A3,OUTPUT);
```

```

pinMode(A5,OUTPUT);
// pinMode(10,OUTPUT);
digitalWrite(A0,HIGH);
digitalWrite(A1,LOW);
digitalWrite(A3,HIGH);
digitalWrite(A5,LOW);

s.begin(9600);
Serial.begin(9600);
lcd.begin(20, 4);

// Print a message to the LCD.
lcd.print("IoT Home Automation");

dht.begin();
}

void loop() {

  unsigned long curr_millis=millis();

  receive_json();

  if(curr_millis-read_ir>=500){
    perform_ir();
    read_ir=curr_millis;
  }

  if(curr_millis-read_dht>=2000){
    perform_dht();
    read_dht=curr_millis;
  }

}

void perform_ir() {
  if(analogRead(A2)<250){
    status= !status;
  }

  if(Auto==1){

```

```
digitalWrite(8,HIGH);
lcd.setCursor(0,2);
lcd.print("Auto ");

if(status==false){
    digitalWrite(9,LOW);
    lcd.setCursor(0,3);
    lcd.print("lamp off");
}
else{
    digitalWrite(9,HIGH);
    lcd.setCursor(0,3);
    lcd.print("lamp on ");
}

if(light==1){
    digitalWrite(13,HIGH);
    lcd.setCursor(9,2);
    lcd.print("Led on ");
}
else{
    digitalWrite(13,LOW);
    lcd.setCursor(9,2);
    lcd.print("Led off");
}
}

else{
    digitalWrite(8,LOW);
    lcd.setCursor(0,2);
    lcd.print("Manual");
    if(lamp==1){
        digitalWrite(9,HIGH);
        lcd.setCursor(0,3);
        lcd.print("lamp on ");
    }
    else{
        digitalWrite(9,LOW);
        lcd.setCursor(0,3);
        lcd.print("lamp off");
    }
}

if(fan==1){
    digitalWrite(10,HIGH);
    lcd.setCursor(9,3);
    lcd.print("fan on ");
```

```

    }
    else{
        digitalWrite(10,LOW);
        lcd.setCursor(9,3);
        lcd.print("fan off");
    }

    if(light==1){
        digitalWrite(13,HIGH);
        lcd.setCursor(9,2);
        lcd.print("LED on ");
    }
    else{
        digitalWrite(13,LOW);
        lcd.setCursor(9,2);
        lcd.print("LED off");
    }

}
}

void perform_dht(){
    h=dht.readHumidity();
    t=dht.readTemperature();

    if (isnan(h) || isnan(t) ) {
        return;
    }

    hic = dht.computeHeatIndex(t, h, false);

    if(Auto==1){

        digitalWrite(8,HIGH);
        lcd.setCursor(0,2);
        lcd.print("Auto ");
        if(t>=29){
            digitalWrite(10,HIGH);
            lcd.setCursor(9,3);
            lcd.print("fan on ");
        }
        else{
            digitalWrite(10,LOW);
            lcd.setCursor(9,3);
            lcd.print("fan off");
        }
    }
}

```

```

if(light==1){
    digitalWrite(13,HIGH);
    lcd.setCursor(9,2);
    lcd.print("LED on ");
}
else{
    digitalWrite(13,LOW);
    lcd.setCursor(9,2);
    lcd.print("LED off");
}
}

```

```

else{
    digitalWrite(8,LOW);
    lcd.setCursor(0,2);
    lcd.print("Manual");
    if(lamp==1){
        digitalWrite(9,HIGH);
        lcd.setCursor(0,3);
        lcd.print("lamp on ");
    }
    else{
        digitalWrite(9,LOW);
        lcd.setCursor(0,3);
        lcd.print("lamp off");
    }
}

```

```

if(fan==1){
    digitalWrite(10,HIGH);
    lcd.setCursor(9,3);
    lcd.print("fan on ");
}
else{
    digitalWrite(10,LOW);
    lcd.setCursor(9,3);
    lcd.print("fan off");
}

```

```

if(light==1){
    digitalWrite(13,HIGH);
    lcd.setCursor(9,2);
    lcd.print("LED on ");
}
else{

```

```

        digitalWrite(13,LOW);
        lcd.setCursor(9,2);
        lcd.print("LED off");
    }

}

lcd.setCursor(0, 1);
lcd.print("Tem:");
lcd.print(t);
lcd.print("C ");
lcd.print("Hu:");
lcd.print(h);
lcd.print("%");

Serial.print(F("Humidity: "));
Serial.print(h);
Serial.print(F("% Temperature: "));
Serial.print(t);
Serial.print(F("°C Heat index: "));
Serial.print(hic);
Serial.print(F("°C "));
Serial.print("\n");

}

void receive_json(){
    StaticJsonBuffer<1000> jsonBuffer;
    JsonObject& root = jsonBuffer.parseObject(s);

    if (root == JsonObject::invalid())
    {
        return;
    }

    Serial.println("JSON received and parsed");
    root.prettyPrintTo(Serial);
    Serial.println("");
    Serial.print("Automate: ");
    Auto=root["LED1"];
    Serial.println(Auto);
    Serial.print("Lamp: ");
    lamp=root["LED2"];
    Serial.println(lamp);

```



```

Serial.print("Fan: ");
fan=root["LED3"];
Serial.println(fan);
Serial.print("light ");
light=root["LED4"];
Serial.println(light);
Serial.println("-----xxxxx-----");
Serial.println("");
}

```

Code to be uploaded to NodeMCU:

```

String header = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";

String html_1 = R"=====(
<!DOCTYPE html>
<html>
<head>
<meta name='viewport' content='width=device-width, initial-scale=1.0'/>
<meta charset='utf-8'>
<style>
  body {font-size:140%;}
  h2 {text-align:center; }
  #main1 {display: table; margin: auto; padding: 0 10px 0 10px; }
  #LED_button1 { padding:10px 10px 10px 10px; width:100%; background-color:
#50FF50; font-size: 120%;}

  #main2 {display: table; margin: auto; padding: 0 10px 0 10px; }
  #LED_button2 { padding:10px 10px 10px 10px; width:100%; background-color:
#50FF50; font-size: 120%;}

  #main3 {display: table; margin: auto; padding: 0 10px 0 10px; }
  #LED_button3 { padding:10px 10px 10px 10px; width:100%; background-color:
#50FF50; font-size: 120%;}

  #main4 {display: table; margin: auto; padding: 0 10px 0 10px; }
  #LED_button4 { padding:10px 10px 10px 10px; width:100%; background-color:
#50FF50; font-size: 120%;}

</style>

<script>
  function switchLED1()
  {

```

```

    var button_text = document.getElementById("LED_button1").value;
    if (button_text=="Turn on Automation")
    {
        document.getElementById("LED_button1").value = "Turning on Automation";
        ajaxLoad('AutoON');
    }
    else
    {
        document.getElementById("LED_button1").value = "Turning on Automation";
        ajaxLoad('AutoOFF');
    }
}

```

```

function switchLED2()
{
    var button_text = document.getElementById("LED_button2").value;
    if (button_text=="Turn on the Lamp")
    {
        document.getElementById("LED_button2").value = "Turning on the Lamp";
        ajaxLoad('LampON');
    }
    else
    {
        document.getElementById("LED_button2").value = "Turning off the Lamp";
        ajaxLoad('LampOFF');
    }
}

```

```

function switchLED3()
{
    var button_text = document.getElementById("LED_button3").value;
    if (button_text=="Turn on the Fan")
    {
        document.getElementById("LED_button3").value = "Turning on the Fan";
        ajaxLoad('FanON');
    }
    else
    {
        document.getElementById("LED_button3").value = "Turning off the Fan";
        ajaxLoad('FanOFF');
    }
}

```

```

function switchLED4()
{
    var button_text = document.getElementById("LED_button4").value;

```

```

    if (button_text=="Turn on the Bulb")
    {
        document.getElementById("LED_button4").value = "Turning on the Bulb";
        ajaxLoad('BulbON');
    }
    else
    {
        document.getElementById("LED_button4").value = "Turning off the Bulb";
        ajaxLoad('BulbOFF');
    }
}

var ajaxRequest = null;
if (window.XMLHttpRequest) { ajaxRequest =new XMLHttpRequest(); }
else
    { ajaxRequest =new ActiveXObject("Microsoft.XMLHTTP"); }

function ajaxLoad(ajaxURL)
{
    if(!ajaxRequest){ alert("AJAX is not supported."); return; }

    ajaxRequest.open("GET",ajaxURL,true);
    ajaxRequest.onreadystatechange = function()
    {
        if(ajaxRequest.readyState == 4 && ajaxRequest.status==200)
        {
            var ajaxResult = ajaxRequest.responseText;

            if ( ajaxResult == "Automation is on" )
            { document.getElementById("LED_button1").value = "Turn off Automation";
              document.getElementById("reply1").innerHTML = ajaxResult; }
            else if ( ajaxResult == "Automation is off" )
            { document.getElementById("LED_button1").value = "Turn on Automation";
              document.getElementById("reply1").innerHTML = ajaxResult; }

            else if ( ajaxResult == "Lamp is on" )
            { document.getElementById("LED_button2").value = "Turn off the Lamp";
              document.getElementById("reply2").innerHTML = ajaxResult; }

            else if ( ajaxResult == "Lamp is off" )
            { document.getElementById("LED_button2").value = "Turn on the Lamp";
              document.getElementById("reply2").innerHTML = ajaxResult; }

            else if ( ajaxResult == "Fan is on" )
            { document.getElementById("LED_button3").value = "Turn off the Fan";
              document.getElementById("reply3").innerHTML = ajaxResult; }

```

```

        else if ( ajaxResult == "Fan is off" )
        { document.getElementById("LED_button3").value = "Turn on the Fan";
document.getElementById("reply3").innerHTML = ajaxResult; }

        else if ( ajaxResult == "Bulb is on" )
        { document.getElementById("LED_button4").value = "Turn off the Bulb";
document.getElementById("reply4").innerHTML = ajaxResult; }

        else if ( ajaxResult == "Bulb is off" )
        { document.getElementById("LED_button4").value = "Turn on the Bulb";
document.getElementById("reply4").innerHTML = ajaxResult; }

    }
    }
    ajaxRequest.send();
}

```

```

</script>

<title>Home Automation Prototype</title>
</head>
<body>
    <h2>Home Automation Prototype</h2>

    <div id='main1'>
        <input type="button" id = "LED_button1" onclick="switchLED1()" value="Turn on
Automation" />
        <p id = "reply1">Reply appears here</p>
    </div>

    <div id='main2'>
        <input type="button" id = "LED_button2" onclick="switchLED2()" value="Turn on the
Lamp" />
        <p id = "reply2">Reply appears here</p>
    </div>

    <div id='main3'>
        <input type="button" id = "LED_button3" onclick="switchLED3()" value="Turn on the
Fan" />
        <p id = "reply3">Reply appears here</p>
    </div>

    <div id='main4'>

```

```

        <input type="button" id = "LED_button4" onclick="switchLED4()" value="Turn on the
Bulb" />

```

```

        <p id = "reply4">Reply appears here</p>
    </div>

```

```

</body>
</html>

```

```

)=====";

```

```

#include <ESP8266WiFi.h>

```

```

#include <SoftwareSerial.h>

```

```

#include <ArduinoJson.h>

```

```

SoftwareSerial s(D6,D7);
int data1;
int data2;
int data3;
int data4;

```

```

StaticJsonBuffer<1000> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();

```

```

// change these values to match your network
char ssid[] = "Node_mcu";    // your network SSID (name)
char pass[] = "12345678";    // your network password

```

```

WiFiServer server(80);

```

```

String request = "";
int LED_Pin1 = D0;
int LED_Pin2 = D1;
int LED_Pin3 = D2;
int LED_Pin4 = D3;

```

```

void setup()
{
    pinMode(LED_Pin1, OUTPUT);
    pinMode(LED_Pin2, OUTPUT);
    pinMode(LED_Pin3, OUTPUT);

```

```

pinMode(LED_Pin4, OUTPUT);

Serial.begin(9600);
Serial.println();
Serial.println("Serial started at 9600");
Serial.println("Home Automation");
Serial.println();

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

while (WiFi.status() != WL_CONNECTED)
{
    Serial.print(".");
    delay(500);
}

Serial.println("");
Serial.println(F("[CONNECTED]"));
Serial.print("[IP ");
Serial.print(WiFi.localIP());
Serial.println("]");

// start a server
server.begin();
Serial.println("Server started");
s.begin(9600);
} // void setup()

void loop()
{
    // Check if a client has connected
    WiFiClient client = server.available();
    if (!client) { return; }

    // Read the first line of the request
    request = client.readStringUntil('\r');

    Serial.print("request: "); Serial.println(request);

    if ( request.indexOf("AutoON") > 0 )
    {
        digitalWrite(LED_Pin1, HIGH);
    }
}

```

```

        data1=1;
        root["LED1"]=data1;
        root.printTo(s);

        client.print( header );
        client.print( "Automation is on" );
    }
    else if ( request.indexOf("AutoOFF") > 0 )
    {
        digitalWrite(LED_Pin1, LOW);
        data1=0;
        root["LED1"]=data1;
        root.printTo(s);

        client.print( header );
        client.print( "Automation is off" );
    }
    /*else
    {
        client.flush();
        client.print( header );
        client.print( html_1 );
        delay(5);
    }*/
    //////////
    else if ( request.indexOf("LampON") > 0 )
    {
        digitalWrite(LED_Pin2, HIGH);
        data2=1;
        root["LED2"]=data2;
        root.printTo(s);

        client.print( header );
        client.print( "Lamp is on" );
    }
    else if ( request.indexOf("LampOFF") > 0 )
    {
        digitalWrite(LED_Pin2, LOW);
        data2=0;
        root["LED2"]=data2;
        root.printTo(s);

        client.print( header );
        client.print( "Lamp is off" );
    }
    /* else

```

```

{
  client.flush();
  client.print( header );
  client.print( html_1 );
  delay(5);
}*/
//////////
else if ( request.indexOf("FanON") > 0 )
{
  digitalWrite(LED_Pin3, HIGH);
  data3=1;
  root["LED3"]=data3;
  root.printTo(s);

  client.print( header );
  client.print( "Fan is on" );
}
else if ( request.indexOf("FanOFF") > 0 )
{
  digitalWrite(LED_Pin3, LOW);
  data3=0;
  root["LED3"]=data3;
  root.printTo(s);

  client.print( header );
  client.print( "Fan is off" );
}
/*else
{
  client.flush();
  client.print( header );
  client.print( html_1 );
  delay(5);
}*/
//////////

else if ( request.indexOf("BulbON") > 0 )
{
  digitalWrite(LED_Pin4, HIGH);
  data4=1;
  root["LED4"]=data4;
  root.printTo(s);

  client.print( header );
  client.print( "Bulb is on" );
}

```



```

else if ( request.indexOf("BulbOFF") > 0 )
{
    digitalWrite(LED_Pin4, LOW);
    data4=0;
    root["LED4"]=data4;
    root.printTo(s);

    client.print( header );
    client.print( "Bulb is off" );
}
else
{
    client.flush();
    client.print( header );
    client.print( html_1 );
    delay(5);
}
//////////

// The client will actually be disconnected when the function returns and 'client' object is
detroyed
} // void loop()

```