
# Computer Engineering Department
# CSE 493 – Senior Design Project II

# IOT Smart Home Automation Project

**Made by:** Batuhan Özel & Hamza Bora Kocaoğlu
**Student Number:** 20170808013 & 20180808006
**Instructor:** Yusuf Sinan Hanay
**Date:** June 6, 2022

# Contents

**Introduction……...1**

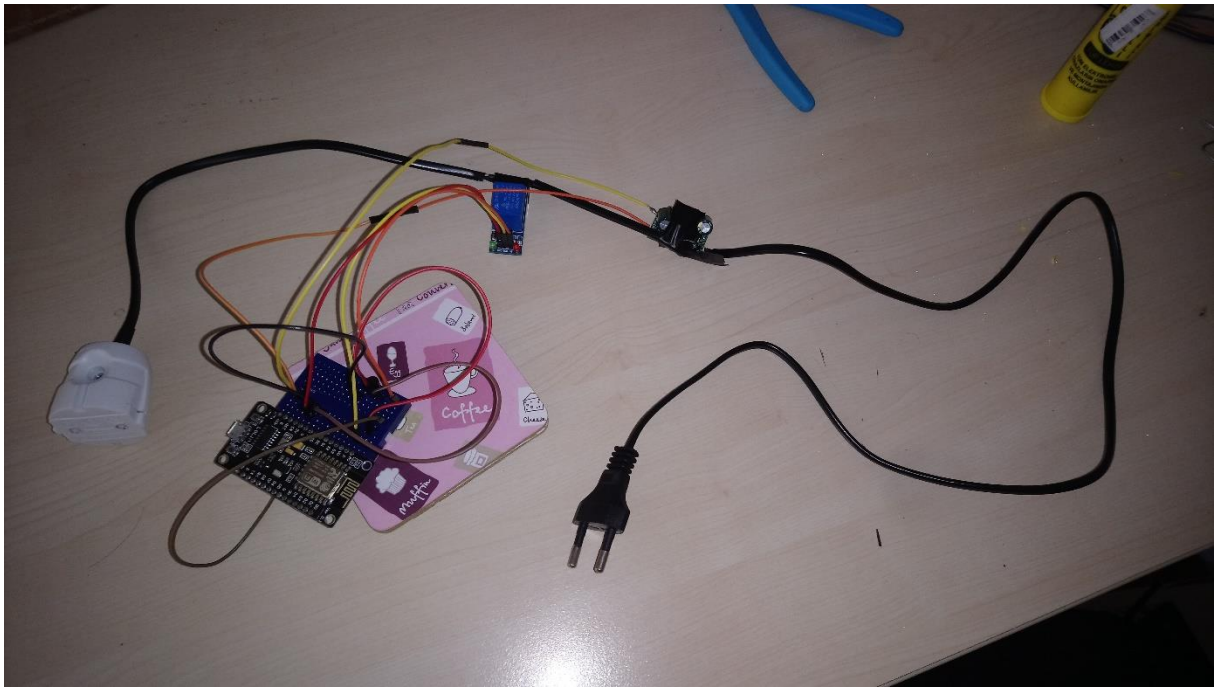
**Logic of Our Circuit…….2**

**Codes of NodeMCU……..3-4**

**Control Center of Plug……5**

# Introduction

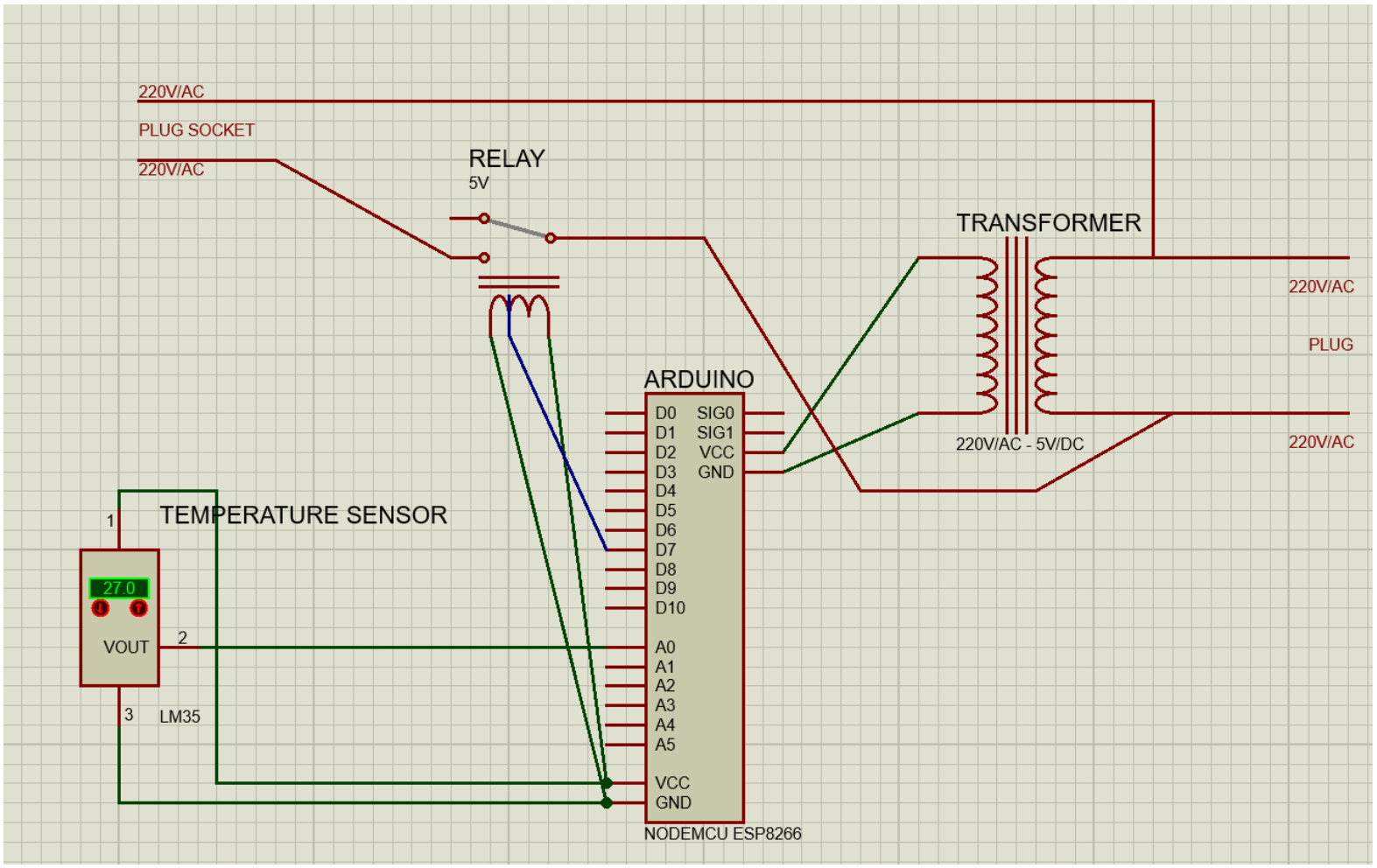
We have made an example of a smart plug to use in smart homes. Here is our circuit;



We have used these components;

1) **A Plug** to take energy from **220V** Power Grid.
2) **A NodeMCU ESP8266** as an **Arduino device.** It works as our **client.**
3) **A Transformer** to transform **220V/AC** to **5V/DC.**
4) **A Relay module** to **Open & Close** the circuit for plug socket.
5) **A Plug Socket** to **plug** users' devices in.
6) **A Mini Breadboard**
7) **An LM35 Temperature Sensor** to read temperature data.

# Logic of Our Circuit

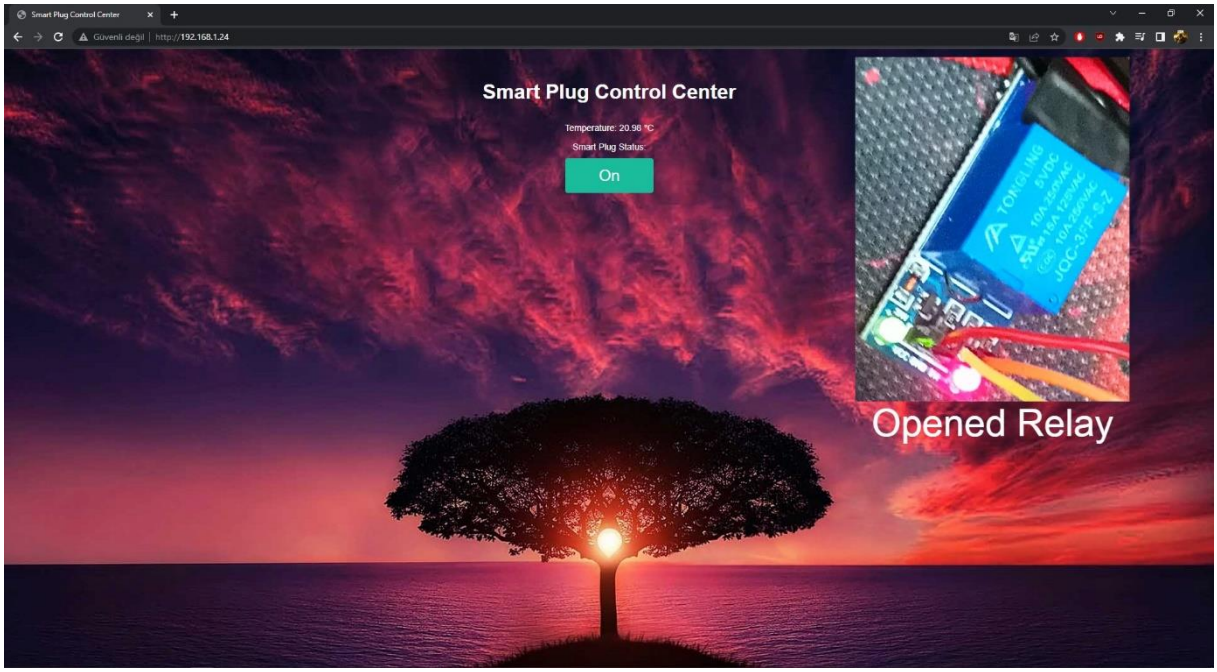


# Codes of NodeMCU

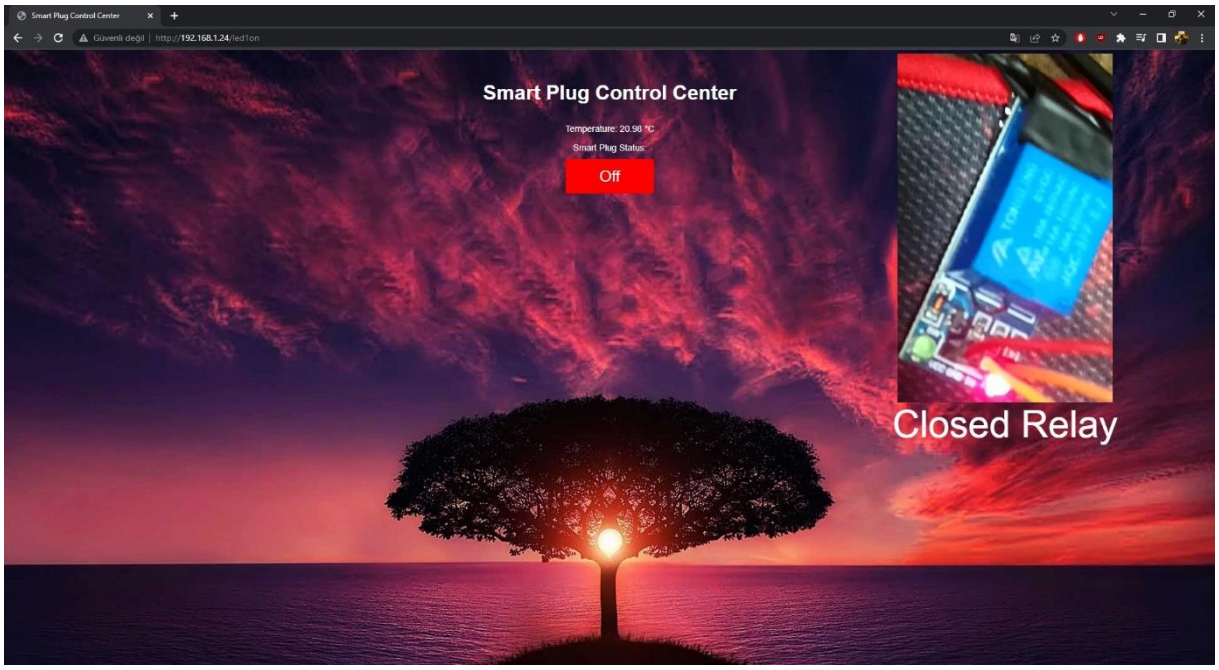
```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

const char* ssid = "EddaNetwork23";  // Enter targeted WIFI name
const char* password = "Zeynepbora2";  //Enter targeted WIFI password

ESP8266WebServer server(80); //We created a webserver object by using port 80

uint8_t LED1pin = D7; //Selecting the pins
bool LED1status = LOW;

uint8_t LED2pin = D6;
bool LED2status = LOW;

void setup() {
  Serial.begin(115200);
  delay(100);
  pinMode(LED1pin, OUTPUT);//Setting the pins as output pins
  pinMode(LED2pin, OUTPUT);


  Serial.println(ssid);
  Serial.println(" Ağına Bağlanılıyor");


  WiFi.begin(ssid, password);//We access to wifi that has been specified earlier

  //We check if the connection is established
  while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.print("."); // Until the connection is established, we print dots to serial port
  }
  Serial.println("");
  Serial.println("Ağ Bağlantısı Sağlandı..!");
  Serial.print("IP Adresiniz: ");  Serial.println(WiFi.localIP()); //We learn the local IP from serial port and then we use the IP for browser access

   server.on("/", handle_OnConnect);
   server.on("/led1on", handle_led1on);
   server.on("/led1off", handle_led1off);
   server.on("/led2on", handle_led2on);
   server.on("/led2off", handle_led2off);
   server.onNotFound(handle_NotFound);

   server.begin(); //Start the server
   Serial.println("HTTP Sunucusu Başlatıldı");
}
float temp_val;
void loop() {
  server.handleClient();
  float temp_adc_val;
  temp_adc_val = analogRead(A0);  /* Read Temperature */
  temp_val = (temp_adc_val / 1023.0)*5000; /* Convert adc value to equivalent voltage */
  temp_val = (temp_val/10); /* LM35 gives output of 10mv/°C */
  Serial.print("Temperature = ");
  Serial.print(temp_val);
  Serial.print(" Degree Celsius\n");
  delay(1000);
  //Change the pin voltages according to relevant variables(HIGH or LOW)
  if(LED1status)
  {digitalWrite(LED1pin, HIGH);}
  else
  {digitalWrite(LED1pin, LOW);}

  if(LED2status)
  {digitalWrite(LED2pin, HIGH);}
  else
  {digitalWrite(LED2pin, LOW);}
}
```

```
void handle_OnConnect() {//Deafult entries
  LED1status = LOW;
  LED2status = LOW;
  Serial.println("GPIO7 Durumu: OFF | GPIO6 Durumu: OFF");
  server.send(200, "text/html", SendHTML(LED1status,LED2status)); //Updates the html codes
}

void handle_led1on() {//Updates the variables after button event on browser
  LED1status = HIGH;
  Serial.println("GPIO7 Durumu: ON");
  server.send(200, "text/html", SendHTML(true,LED2status)); //Updates the html codes
}

void handle_led1off() {//Updates the variables after button event on browser
  LED1status = LOW;
  Serial.println("GPIO7 Durumu: OFF");
  server.send(200, "text/html", SendHTML(false,LED2status)); //Updates the html codes
}

void handle_led2on() {//Updates the variables after button event on browser
  LED2status = HIGH;
  Serial.println("GPIO6 Durumu: ON");
  server.send(200, "text/html", SendHTML(LED1status,true)); //Updates the html codes
}

void handle_led2off() {//Updates the variables after button event on browser
  LED2status = LOW;
  Serial.println("GPIO6 Durumu: OFF");
  server.send(200, "text/html", SendHTML(LED1status,false)); //Updates the html codes
}

void handle_NotFound(){//404 not found case
  server.send(404, "text/plain", "Sayfa Bulunamadı");
}
String SendHTML(uint8_t led1stat,uint8_t led2stat){ // Updating the html codes
  String ptr = "<!DOCTYPE html> <html>\n";
  ptr+="<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\"><meta http-equiv=\"Content-Type
  ptr+="<h1 style=\"color:rgb(255, 255, 255);\">Smart Plug Control Center</h1>";
  ptr+="<p style=\"color:rgb(255, 255, 255);\">" "Temperature: " + String(temp_val) +" °C" + "</p>";
  if(led1stat){
   ptr+="<p style=\"color:rgb(255, 255, 255);\">Smart Plug Status:</p> <a class=\"button button-off\" href=\"/led1off\">Off</a>\n";
  }
  else{ptr +="<p style=\"color:rgb(255, 255, 255);\">Smart Plug Status:</p><a class=\"button button-on\" href=\"/led1on\">On</a>\n";}

  /*if(led2stat){
  ptr +="<p style=\"color:rgb(255, 255, 255);\">LED2 Durum: On</p><a class=\"button button-off\" href=\"/led2off\">Off</a>\n";
  }
  else{
  ptr +="<p>LED2 Durum: Off</p><a class=\"button button-on\" href=\"/led2on\">On</a>\n";
  }*/
  ptr +="</body>\n";
  ptr +="</html>\n";
  return ptr;
}
```

# Control Center of Plug




When button in control center is on, relay opens too and plug is working.




When button in control center is off, relay is closed and plug is not working