



# Asansol Engineering College

## Innovative Hardware Model Making Competition-2023

Our Project Title : **Wifi Controlled Home Automation using NodeMCU & Android**

Team Name:- “THE SIERRAS”

Team Number:-13

Team Members:-

- (1) Pallab Chatterjee (Team Leader)
- (2) Ujala Naaz
- (3) Hrithik Gupta
- (4) Ankit Kumar Singh

## Components Used:

- (1) Breadboard
- (2) NodeMCU
- (3) Power Supply
- (4) Jumper Wire
- (5) Bulb Holder
- (6) Relay
- (7) Socket
- (8) Bulb



Breadboard



Bulb Holder



Relay



NodeMCU



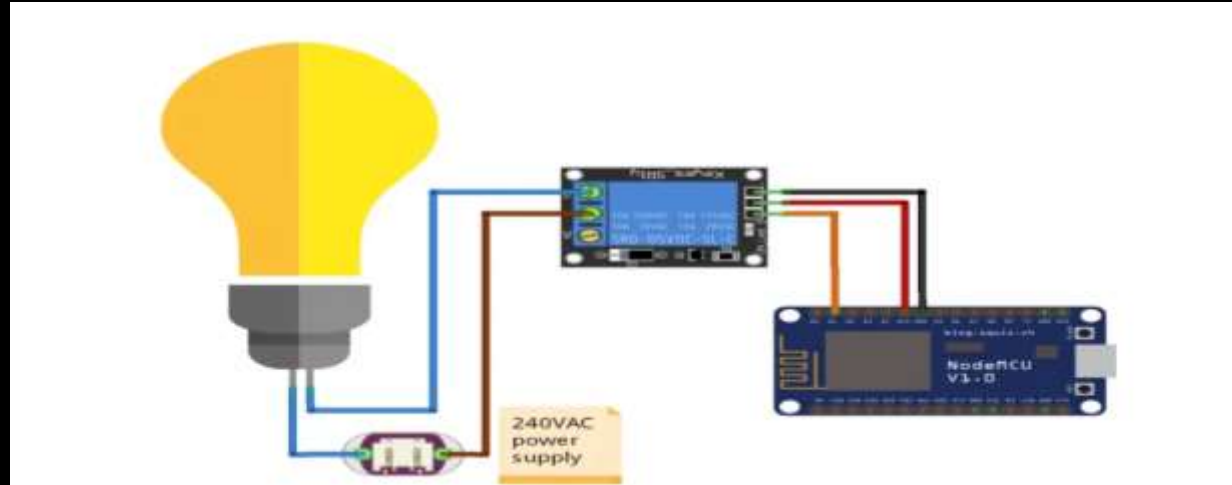
Bulb

## Wifi Controlled Home Automation with ESP8266 & Android

This presentation is all about Wifi Controlled Home Automation Using NodeMCU & Android. Here we will do the same, but instead of using Alexa, we will develop our own App on an Android smartphone to control our locally (using buttons or voice), our home devices.

In this project, we will be interfacing 2 relay with NodeMCU ESP8266 12E Wifi Module and control one socket and a bulb holder. These are some blueprint of home appliances. We will use an Android app having 5+5 ON+OFF Buttons to send a signal to NodeMCU in order to control relay output. Every NodeMCU has a particular IP Address. We will assign this IP Address to Android App for access control. Similarly, NodeMCU is needed to be connected to local Wifi.

## Block Diagram:

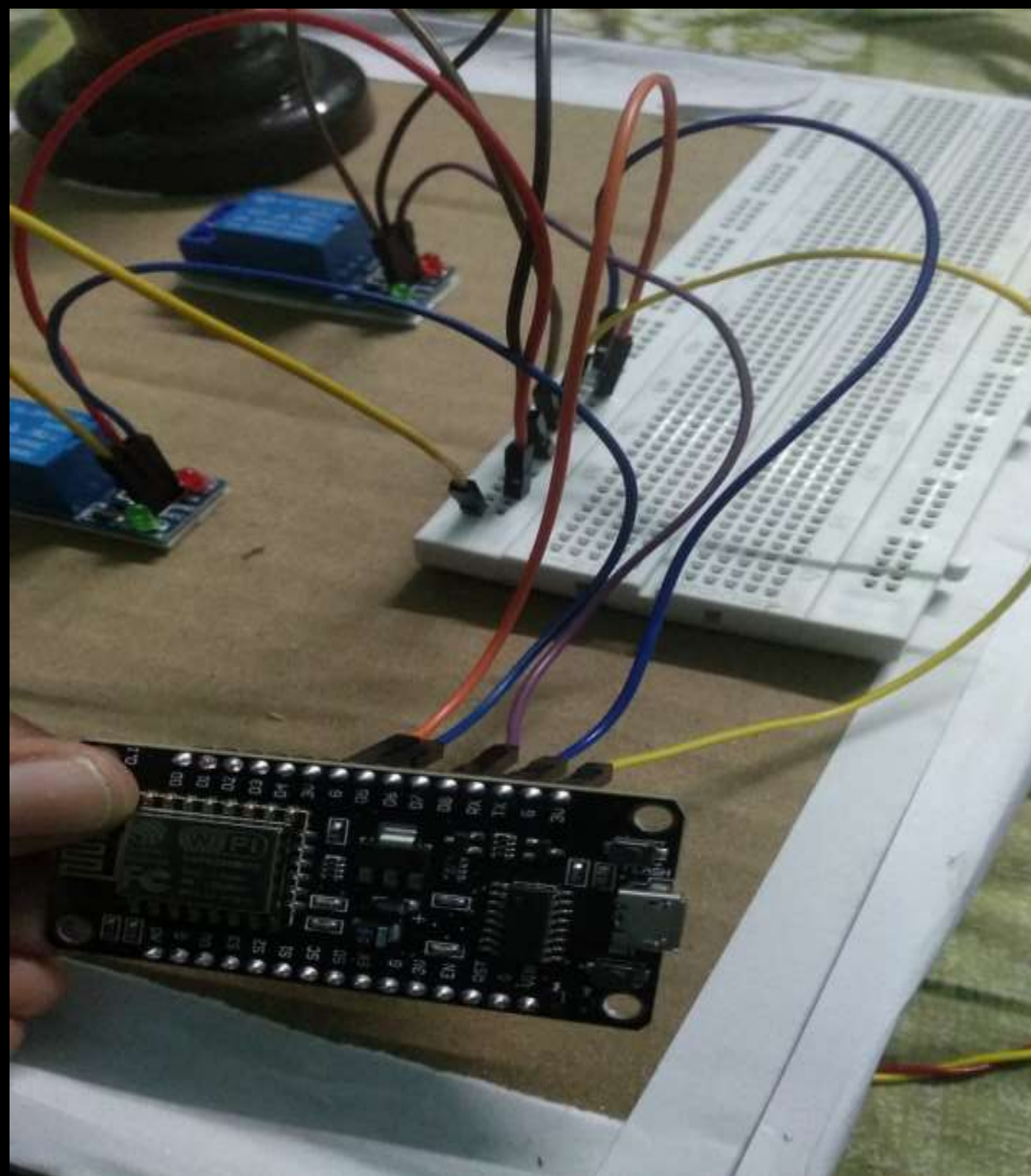
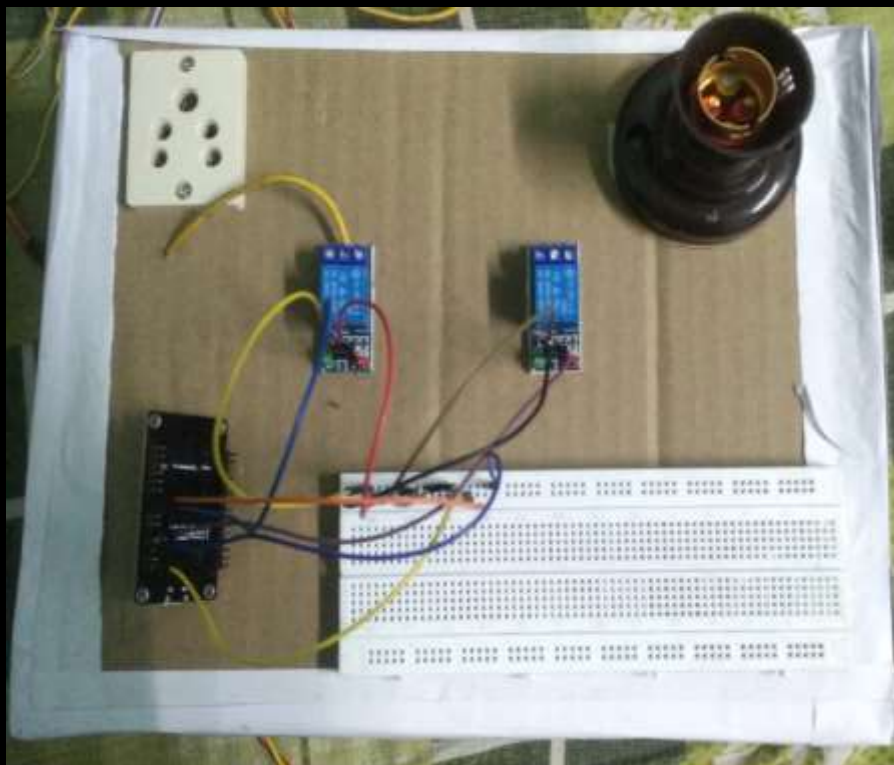


NodeMCU is interfaced with Relay using GPIO Pins of NodeMCU. The output pin of the NodeMCU is connected to different sources. The NodeMCU is connected to Local Wifi and is capable of receiving a signal via the Internet.

The Android Device has a “**Home Automation Control**” Android app installed on it. To control the NodeMCU input/output, the IP address of NodeMCU is entered on the Android App IP box.

Once this setup is done, you can control the Android App staying in any part of the world. Your Android Phone acts as a remote and NodeMCU as a receiver and signal are transferred via the Internet.

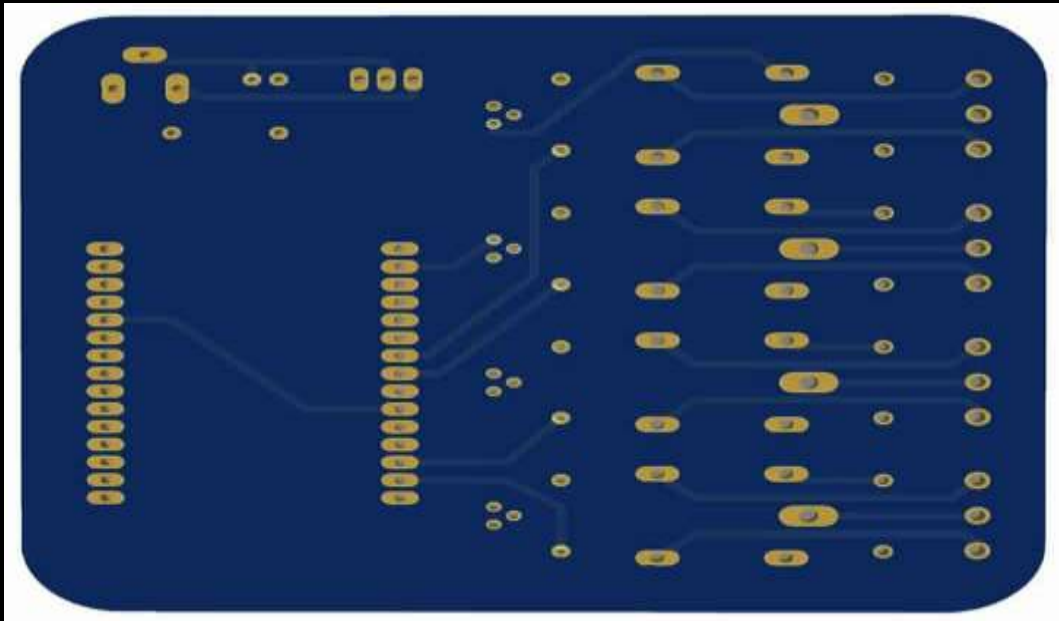
Circuit:



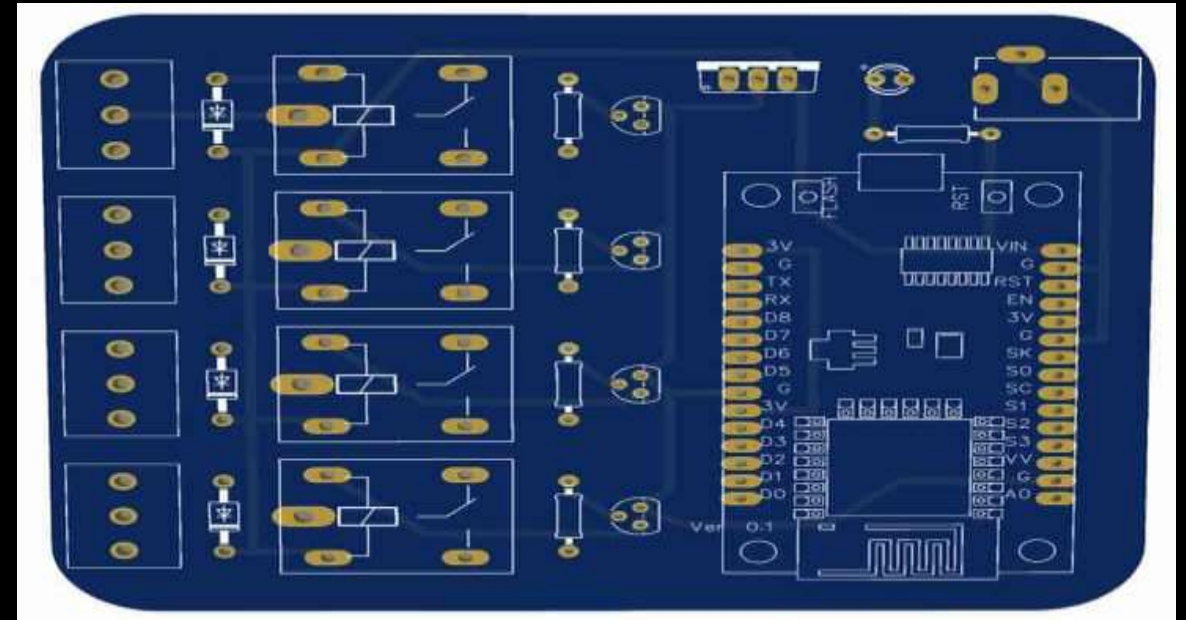


# Home Automation PCB & Gerber File

If you don't want to assemble the circuit on a breadboard and you want PCB for the project, then here is the PCB for you. The PCB Board for the Home Automation Project is designed using [EasyEDA](#) online Circuit Schematics & PCB designing tool. The front side and back side of the PCB is given below.



## PCB Back View



## PCB Front View

## The Android App for Wifi Controlling:



The Android App will send a string that must be interpreted by the code to activate each one of the relays defined as follows:

### **Relay1:**

Turn-On: "r1on";

Turn-Off: "r1off";

### **Relay2:**

Turn-On: "r2on";

Turn-Off: "r2off";

If the Android App sends as a command: "r1on", Relay1 must be turn on. We have also defined "group commands" to turn-on ("all on") and turn-off ("all off") simultaneity all devices. Similarly, the voice input image is also given which when clicked gives a pop up for Google Assistant to Accept Voice Command.

## Creating The Android App Using MIT App Inventor:

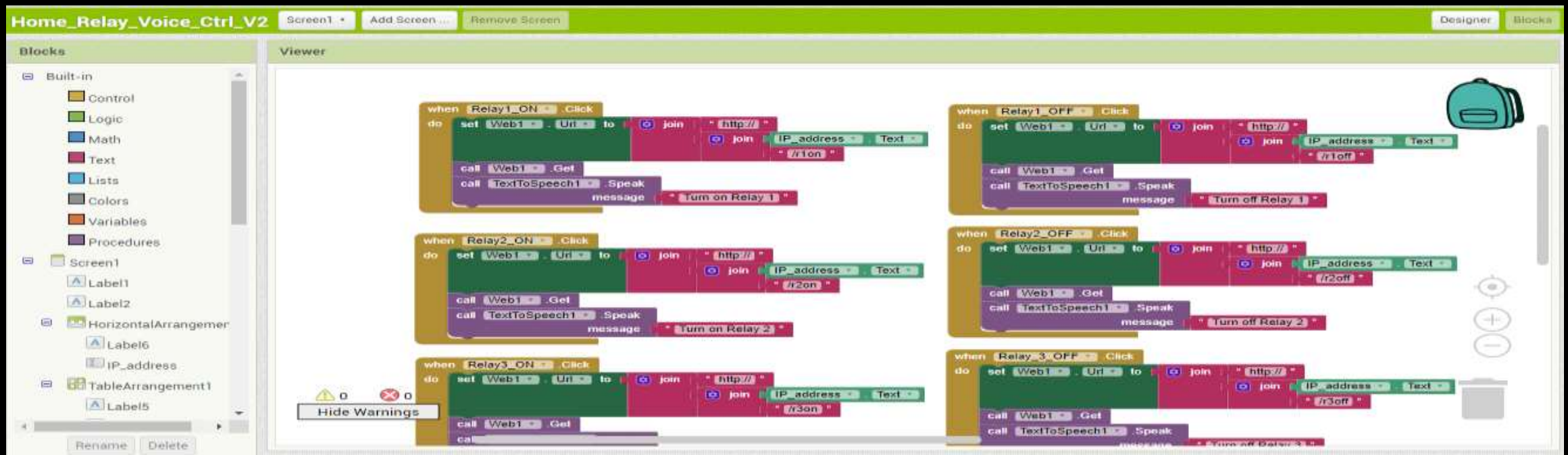
This step is optional. I have given the link below to download the Android app directly. I have also added a .aia link that can directly be downloaded and modified according to your need.





Main Components on Screen are these as shown in the figure above:

- User Interface:
  - Input of IP Address (TextBox named "IP\_Address")
  - 8 ON/OFF Buttons, one for each Relay:
    - Relay\_1\_ON
    - Relay\_2\_OFF
    - etc
  - 2 ON/OFF Buttons for All Devices:
    - All\_Devices\_ON
    - All\_Devices\_OFF
  - Voice Input Button
    - Voice\_Input
- Non Visible Components:
  - Web1
  - SpeechRecognizer1
  - TextToSpeech1
- Other:
  - Text Box:
    - Speech\_To\_Text
  - Label:
    - Comm\_Status



## Code for Finding the Device IP Address:

```
#include <ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
const char* ssid = "how2electronics";
const char* password = "12345678";
void setup()
{
  Serial.begin(115200);
  connectWiFi();
  server.begin();
}
void loop()
{
}
/* connecting WiFi */
void connectWiFi()
{
  Serial.println("Connecting to WIFI");
  WiFi.begin(ssid, password);
  while (!(WiFi.status() == WL_CONNECTED))
  {
    delay(300);
    Serial.print("..");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("NodeMCU Local IP is : ");
  Serial.print(WiFi.localIP());
}
```

Source Code:



Thank You

Team Sierras