



CONTROL A RELAY USING ESP8266 AND ANDROID APP INVENTOR

K. R. Biradar¹, Shivraj Jeure², Vraj Patel³

Assistant Professor¹, Student², Student³

^{1,2,3}Department of Electronics & Telecommunication Engineering, W.I.T., Solapur, PAH Solapur University, Maharashtra, India.

ABSTRACT

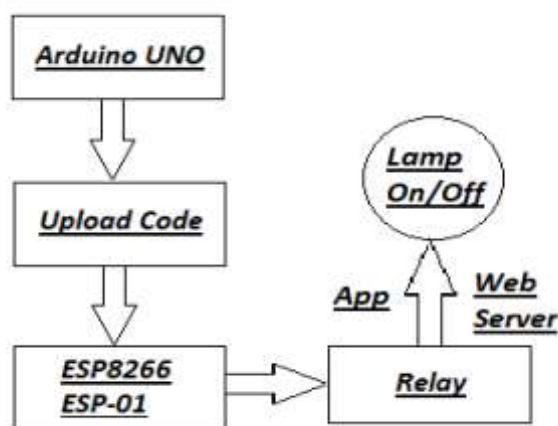
In home automation, wireless home automation (using Wi-Fi or internet connection to connect electrical appliances in your home) is becoming increasingly popular. Our project focuses on using ESP8266 (a low-cost Wi-Fi microchip) to control a Relay. ESP8266 Wi-Fi Module has been a main figure in the DIY IoT market. There are several flavors of ESP8266 like ESP-01 by AiThinker, Node MCU and so forth but the intention and the job are the same. The code is uploaded to ESP8266 Wi-Fi chip by Arduino. There are several IoT Projects based on ESP8266 Wi-Fi Module but Home Automation i.e. controlling different electrical appliances through Wi-Fi (or Internet) has always been a trending and in demand project.

INTRODUCTION

ESP8266 (ESP-01) is a small module that allows microcontrollers to connect to a Wi-Fi network and make simple IP (Internet Protocol) connections. It can be programmed by using Arduino, Node MCU IDE or ESP8266 SDK. Several other modules like ESP-02, ESP-07 were also released. All these are essentially based on ESP8266, the only

difference is the number of GPIO pins. In this Project, we will connect a simple Relay Module to the ESP8266 Wi-Fi Module and how to control a Relay using ESP8266 and also create an Android App using which you can control the relay. This app can be installed on your android device and then by connecting it to ESP8266, you can control a Relay through the same Wi-Fi connection your phone uses.

Block Diagram



The ESP8266 ESP-01 module had 8 pins. These are VCC, GND, TX, RX, CH_PD, GPIO0 and GPIO2. Connect the VCC and GND to 3.3V (the ESP-01 module is not compatible with 5V so 3.3V is preferred). The rest of the connections are made accordingly (GND to GND, TX pin to TX of Arduino). The only difference is the GPIO2 pin is connected to the INPUT of Relay. First, make all the

necessary connections as per the circuit diagram and upload the program to the ESP8266 Wi-Fi Module. After the program is uploaded, you will get a confirmation message regarding Wi-Fi connection and Static IP Address. Now open the Android App which we developed using App Inventor 2 and installed it on your Android Phone.



Advantages

- Reduced Installation cost.
- Internet Connectivity.
- Scalable and Expandable.

Disadvantages

- The operation of the circuit depends on the working of internet connection. If the working internet connection is not available it will not run.
- Overall maintenance cost maybe higher.

- You can program ESP8266 like any other microcontroller and use a Wi-Fi connection to control it.

Applications

- ESP8266 is capable of hosting applications and is trending in the IoT.
- You can connect to the Internet, host a web server, and connect your smartphone to it.

CONCLUSION

- We are using ESP8266 Wi-Fi chip which has the ability to talk to the internet. It understands AT commands only. We are using embedded code to input AT commands for the particular functions.
- For the controlling and switching operation, we are developing an Android application through App Inventor. The application connects to the server and the server connects to the ESP8266 Wi-Fi chip. Here we are connecting lamp and a relay to the Wi-Fi chip through the server and control the electrical devices through the mobile application.

REFERENCE

1. <https://www.arduino.cc/en/main/arduinoBoardUno>
2. https://ieeeprojectsmadurai.com/IEEE%202019%20IOT%20BASEPAPERS/36_IOT%20BASED%20HOME%20AUTOMATION.pdf
3. https://www.dline.info/tmd/fulltext/v6n2/tmdv6n2_2.pdf