Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Fall, Year:2021), B.Sc. in CSE (Day)

Course Title: Microprocessor and Microcontroller Lab
Course Code: CSE 304 Section: 201 DD

Lab Project Name: Home Automation Using ESP8266 Microcontroller & Android App.

Student Details

	Name	ID
1.	Md. Obaydullah Khan	201002093
2.	Rukaiya Afroze	201002180

Submission Date : December 29, 2021
Course Teacher's Name : Mr. Md Parvez Hossain

[For Teachers use only: Don't Write Anything inside this box]

Lab Project Status

Marks: …… **Signature: .....**

Comments: ..... **Date: .....**

Table of Contents

Chapter 1

Introduction

1.1 Introduction

This project presents a design and prototype implementation of home automation system that uses WiFi [1] technology as a network infrastructure connecting its parts. The proposed system consists of two main components; the first part is the server (web server), which presents system core that manages, controls, and monitors users' home.
Users and system administrator can locally (LAN) or remotely (internet) manage and control system code. Second part is hardware interface module, which provides appropriate interface to sensors and actuator of home automation system.

Unlike most of available home automation system in the market the proposed system is scalable that one server can manage many hardware interface as long as it exists on WiFi network coverage. System supports a wide range of home automation devices like power management components, and security components. The proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

1.2 Design Goals & Objective

Some days before we discussed about automation[2] and remote control via internet. We are talking about controlling our computer from remote place or anywhere in this world, there are many software for controlling computer from remote place but there is a problem too. Those software can only work when the computer is turned on. But always on Computer is miss use of electricity too. Beside our electricity is not stable. We face loadshading without any notice. So when our power will cut off then we can not control our computer again. After that we are thinking to make something that can turn on any electronic device from remote place and we know about home automation.

- The goal of this project is to develop a home automation system that gives the user complete control over all remotely controllable aspects of his or her home.

- The automation system will have the ability to be controlled from a central host PC, the Internet, and also remotely accessed via a Android Mobile based application.

Chapter 2

Implementation of the Project

2.1 Device and Components For This Project

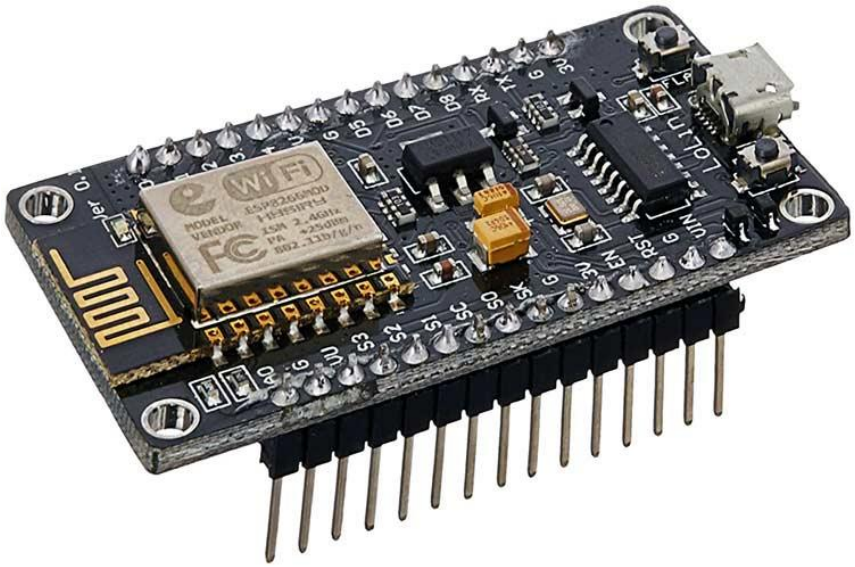
In this section we describe all device and components for this project.

2.1.1 Device list

- ESP8266 [3]
- Relay Devices [4]
- Android Phone
- Electronic Plug and Light Holder
- Bulb
- Jumper Cable (Male-Female)
- Connecting Cable
- USB Cable

2.1.2 Devices Details and Why We Use This Device

- **ESP8266**



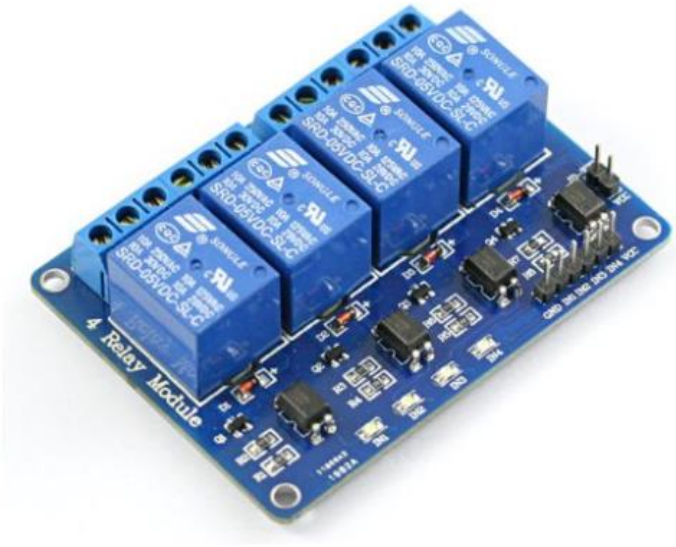
Img : ESP 8266

Description: The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts. There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution!

Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

Relay module:



Img : Relay Module (4 channel)

A relay is an electrical device which is generally used to control high voltages using very low voltage as an 35 Input. This consists of a coil wrapped around a pole and a two small metal flaps(nodes) that are used to close the circuit. One of the node is fixed and other is movable. Whenever an electricity is passed through the coil, it creates a magnetic field and attracts the moving node towards the static node and the circuit gets completed. So, just by applying small voltage to power up the coil we can actually complete the circuit for the high voltage to travel. Also, as the static node is not physically connected to the coil there is very less chance that the Microcontroller powering the coil gets damaged if something goes wrong.

Features:-

- Datasheet - here
- Power led: Yes
- Relay leds: YesHigh quality
- 4 SPDT Relay channels - selectable by user: 36
- PCB parameters: FR4 / 1.5mm / two layers / metalized holes / HAL / white
- stamp / solder mask / extra PCB openings for better voltage isolation /
- doubled high voltage tracks
- Power supply: from USB port
- Current consumption: 400 mA
- Chip: FT245RL
- Size: 77mm x 56mm x 17mm

Advantages:-

- High quality
- Low cost
- No extra power supply
- Software with many functions
- Control electrical devices according weekday/date/time
- Create timers or pulses with our software

Applications:-

- Home automation
- Robotics
- Alarms
- Timers
- Open doors and windows via PC

- Aquariums applications

2.2 Implementation

2.2.1 Hardware part

The connection of all components shown below by drawing and real life picture:

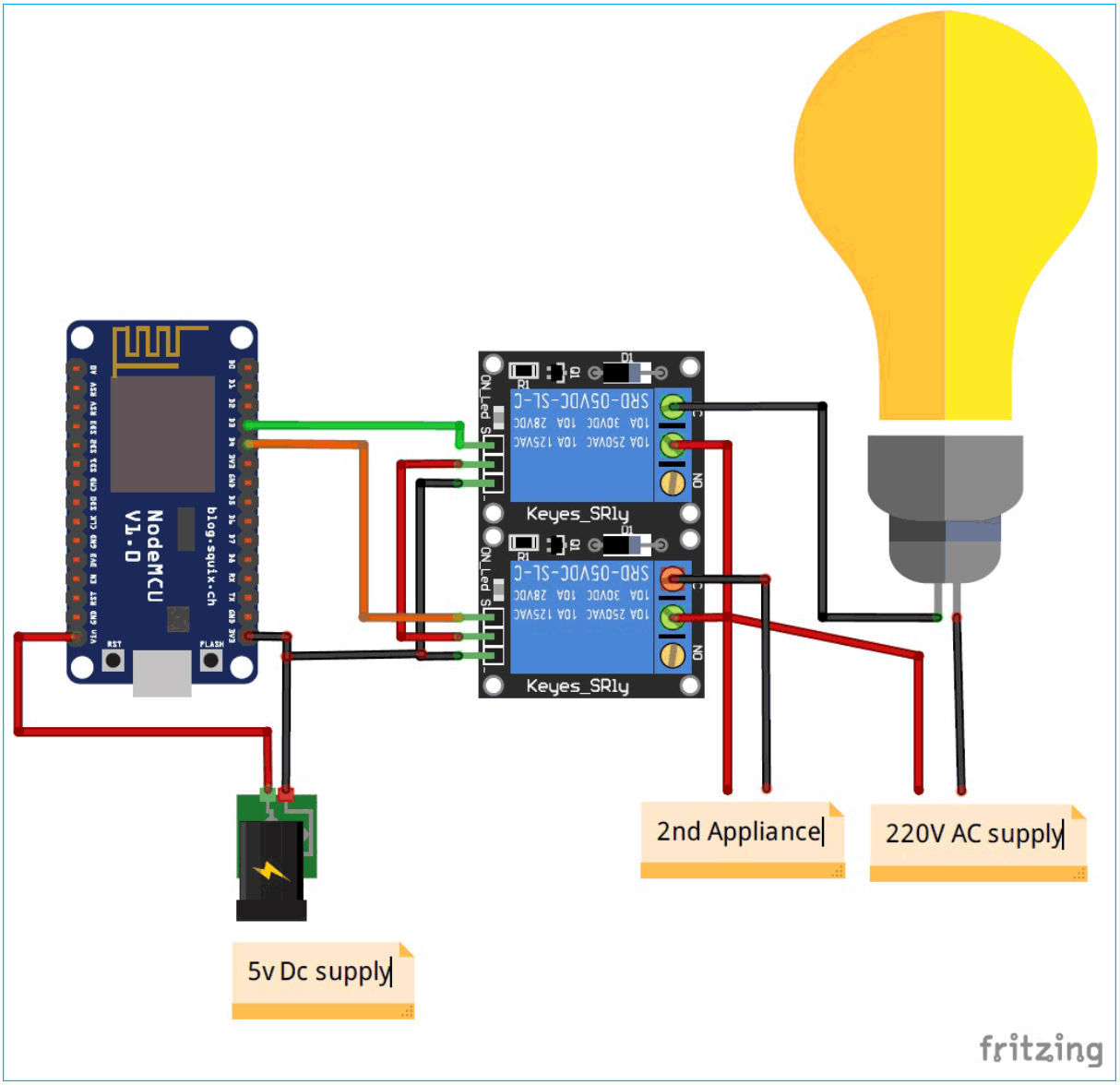
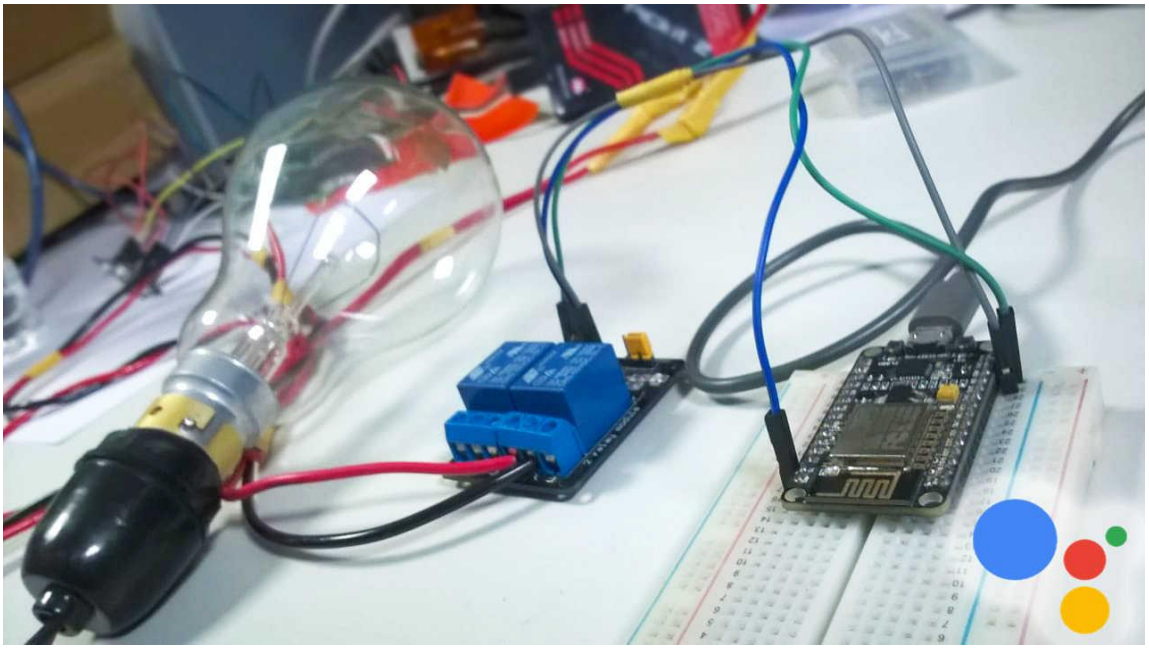
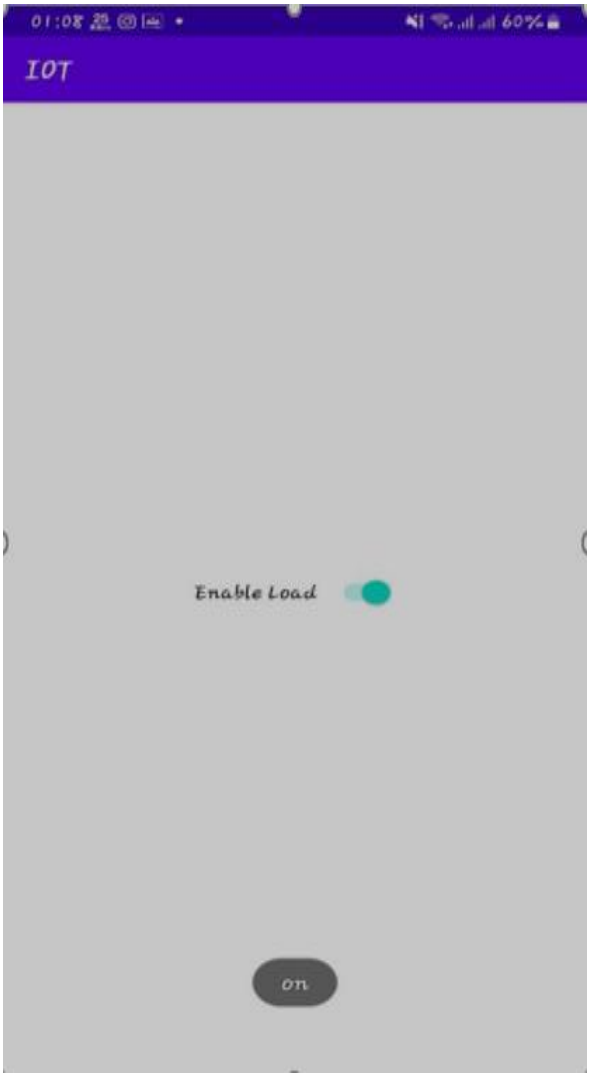


Fig : Connections of all components in Simulation



Img: Real life Implementation;

2.2.2 Software part



Img : 1 Switch Android app for controlling automation

We will control any electronic device by using this app from any where in the world.

Template code :

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">



    <Switch
        android:id="@+id/simpleSwitch1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:checked="false"
        android:text="switch 1"
        android:textOff="Off"
        android:textOn="On" />

    <Switch
        android:id="@+id/simpleSwitch2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"
        android:checked="true"
        android:text="switch 2"
        android:textOff="Off"
        android:textOn="On" />

    <Button
        android:id="@+id/submitButton"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:background="#009284"
        android:padding="10dp"
        android:text="Submit"
        android:textColor="#fff"
        android:textStyle="bold" />
</LinearLayout>
package example.abhiandriod.switchexample;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Switch;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    Switch simpleSwitch1, simpleSwitch2;
    Button submit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // initiate view's
        simpleSwitch1 = (Switch) findViewById(R.id.simpleSwitch1);
        simpleSwitch2 = (Switch) findViewById(R.id.simpleSwitch2);
        submit = (Button) findViewById(R.id.submitButton);
        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String statusSwitch1, statusSwitch2;
                if (simpleSwitch1.isChecked())
                    statusSwitch1 = simpleSwitch1.getTextOn().toString();
                else
                    statusSwitch1 = simpleSwitch1.getTextOff().toString();
```

```
                if (simpleSwitch2.isChecked())
                    statusSwitch2 = simpleSwitch2.getTextOn().toString();
                else
                    statusSwitch2 = simpleSwitch2.getTextOff().toString();
                Toast.makeText(getApplicationContext(), "Switch1 :" + statusSwitch1 + "\n" + "S
witch2 :" + statusSwitch2, Toast.LENGTH_LONG).show(); // display the current state for switch's
            }
        });


    }
}
```

We collect this template code from github and make android app for this automation microcontroller.This application have only one switch but we can add more switch as we want.

Chapter 3

Performance Evaluation

3.1 Simulation Procedure

To run This project at first we have to connect all component as shown as Implementation part. Then we have to connect the plug with electricity now all connection part is complete. After that we
Have to powerup ESP8266.
When ESP 8266 will powered up then we can see a wifi connection named ESP8266. We have to connect any communication device like Android mobile or Laptop and then we have to go its user panel by opening web browser. The hosts Address will be 192.168.4.1. Then we can see connect network button and then we will connect our ESP8266 with our local network.

After that Our total connection procedure both hardware-software & software-network connection have done.

Now we have to go in our Android Application that we made, when we turned on switch the relay

module will get a signal and after 1 second the microcontrollers signal LED will turned on and It means The microcontroller is working successfully and send 3 volt signal in relay module.
The relay module have 2 port, one is when It get 3 volt it will cut off the circuit and total circuit will break and the light will turned off and other is when It get 3 volt signal The circuit will be closed and the light or any other device will get electricity for turning on.

3.2 Results and Discussions

3.2.1 Results

Img: Final result of Home Automation using ESP 8266 and Android app

3.2.2 Analysis and Outcome

This is a useful project for home , office , school , university or any others place. The mobile application based home automation is very easy to control an also setting up is easy. If anyone want to Implemet this in his home he must be have a relay module with more than 4 channel (here 4 channel mean he can control four device) , he also must have stable Internet connection for controlling Automation.

Chapter 4

Conclusion

4.1 Introduction

At the time of global village or time of internet. This time everyone want to be connect with everyone and everyone like automation or remote working. This project will fully allow user to control electronic device remotely.

4.1 Practical Implications

This project is very useful project for home, office , school , college , university or any other place. Every place in this world is now under coverage of electricity and internet, and there are many place where we miss use electricity by not turning electronic device off in right time. If we use home automation , we can timely turn of our un wanted electronic device and can save more electricity .

Beside everyone want automation for their classy life , and this device can make life more easier.

We can use more channel relay device to cover more device under this automation process.

4.2 Scope of Future Work

Day by day, the field of automation is blooming and these systems are having great impact on human beings. The project which is to be implemented is a home automation using Easy IOT Webserver and WIFI and has very good future development. In the current system webserver is installed on a windows PC so the home appliances can be controlled using only by using the device on which webserver is installed.This can be further developed installing webserver on cloud . Advantage of installing webserver on the cloud is that home can be controlled by using any device which has WIFI 802.1 and a web browser. By visiting the IP address of the cloud the control actions can be taken.

References

[1] Garber, Megan (23 June 2014). "'Why-Fi' or 'Wiffy'? How Americans Pronounce Common Tech Terms". The Atlantic. Archived from the original on 15 June 2018.A. Rezi and M. Allam, ”Techniques in array processing by means of transforma- tions, ” in Control and Dynamic Systems, Vol. 69, Multidemsional Systems, C. T.Leondes, Ed. San Diego: Academic Press, 1995, pp. 133-180.

[2] Groover, Mikell (2014). Fundamentals of Modern Manufacturing: Materials, Processes, and Systems.K. Schwalbe, Information Technology Project Management, 3rd ed. Boston: Course Technology, 2004.

[3] "ESP8266 Overview". Espressif Systems. Retrieved 2017-10-02.

[4] https://mysite.du.edu/~jcalvert/tel/morse/morse.htm#H1