Project report
On
Internet Controlled Switch

Course Title: Physic Lab
Course Code : PHY-104

Submitted to:

Afsana Mimi Raka
Lecturer
Department of EEE
Green University of Bangladesh

Submitted By:

Muhammad Sajjad Amin (211002081)
Manik Shekh (211002106)

Department of CSE
Green University of Bangladesh.

Date of Submission: 09 August 2020

Acknowledgment

We are really grateful because we managed to complete our project Internet Controlled Switch within the time given by our lecturer Afsana Mimi Raka. This assignment wouldn't have been completed without the effort and co-operation of our group members Md. Manik shekh and Muhammad Sajjad Amin. We also sincerely thank our lecturer, Afsana Ma'am for the guidance and encouragement in finishing this project.

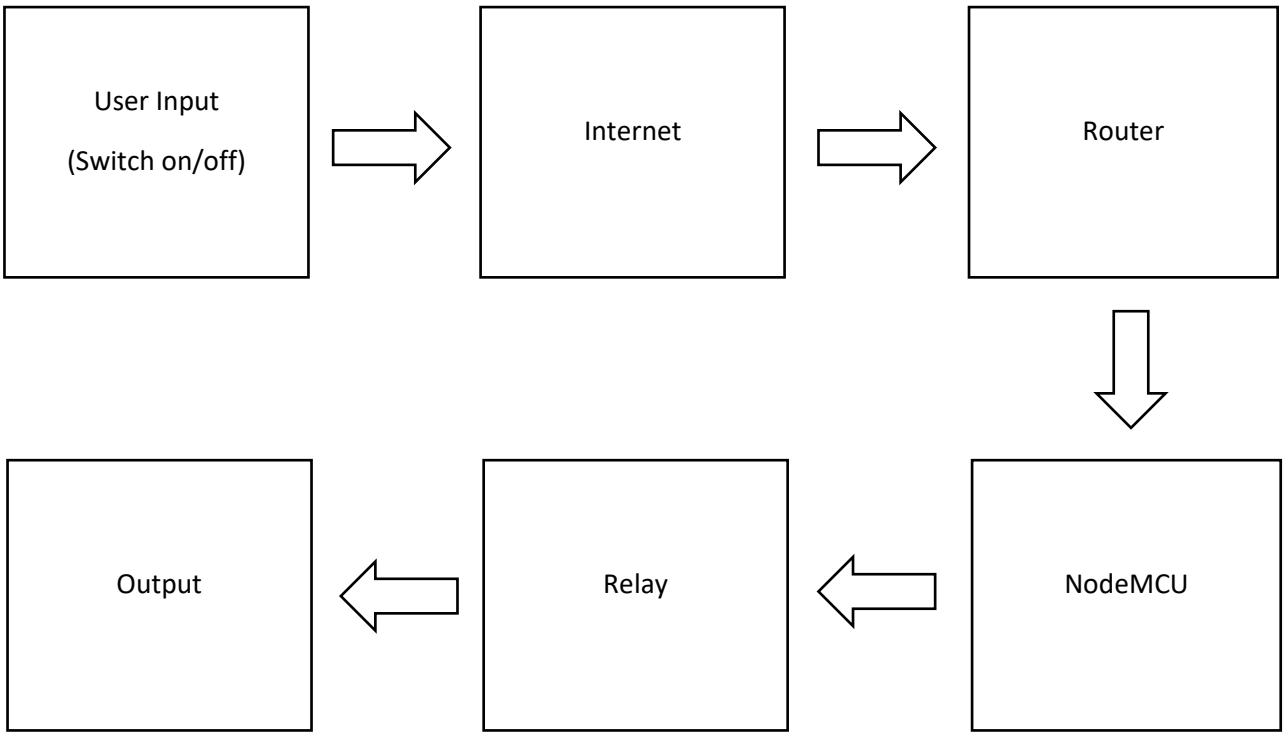
Internet Controlled Switch

Theory:

The internet-controlled switch is a digital switching system where a microcontroller circuit is connected to a local network by Wi-Fi, LAN, or others connectivity mediums. But, a Wi-Fi network is commonly used for it. This local network is connected to the public network with a public IP address to access the switch through the internet. The microcontroller unit acts as a server. The end-user sends the switching request using a web browser or mobile application and it receives the request and switching the associated relay module by passing the high and low voltage by its digital pin. We will use the NodeMCU microcontroller circuit in this project because it has a built-in Wi-Fi module esp8266. So we can connect our microcontroller to the Wi-Fi network.

Block Diagram:

The block diagram of the internet-controlled switch project is shown in the figure.



Here NodeMCU is a server that maintains HTTP protocol. A user turns on/off the switch from any browser/mobile/computer etc. this command transfers via internet to the microcontroller. The microcontroller reads this command and makes a decision on what to do. Then it controls relay by user's demand.

List of Equipment:

- NodeMCU ESP8266
- Relay module
- Wi-Fi router
- LED 2 piece
- Some jumper wires
- Breadboard (if necessary)
- 5 volt DC power adapter

Circuit Diagram:

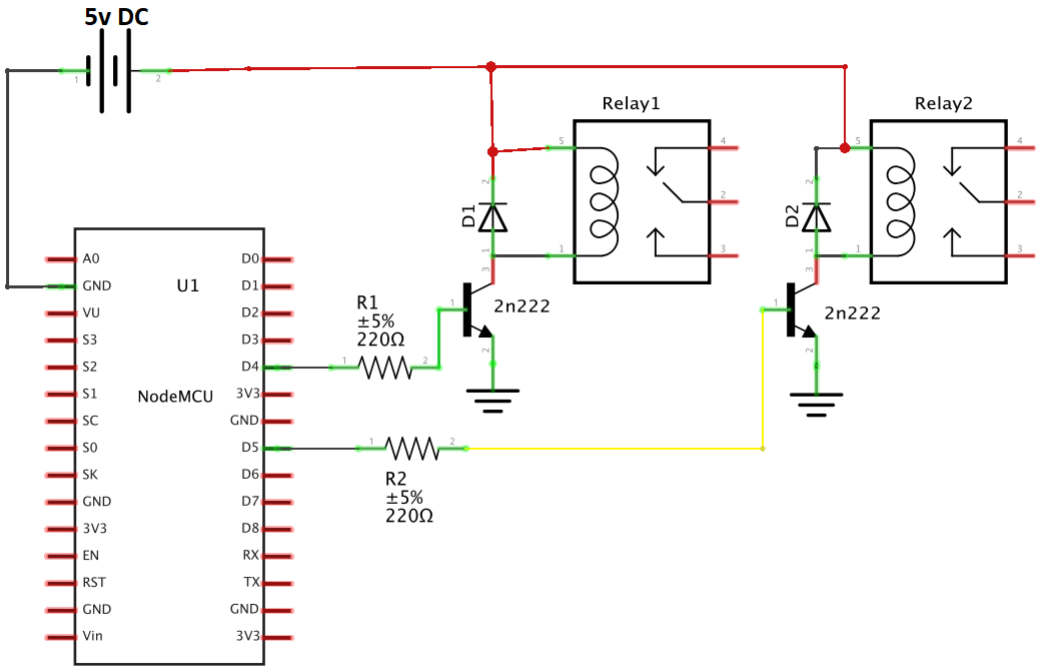
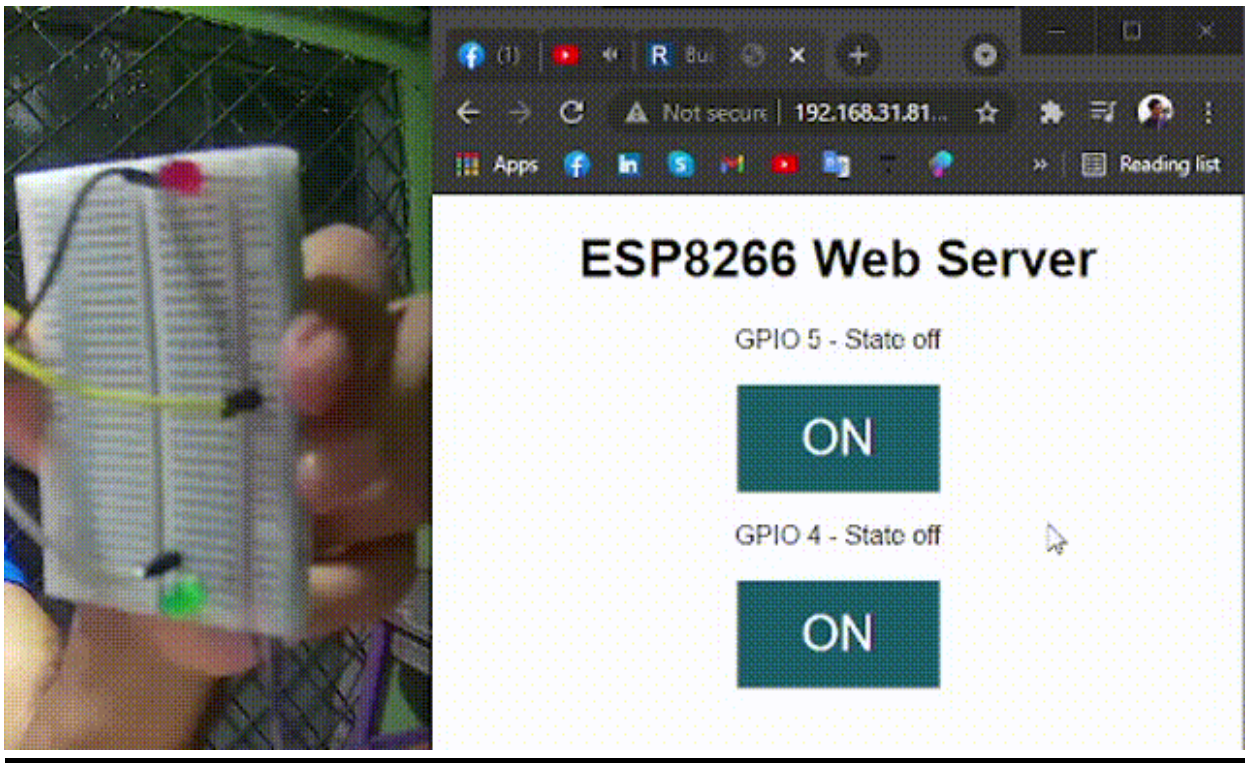
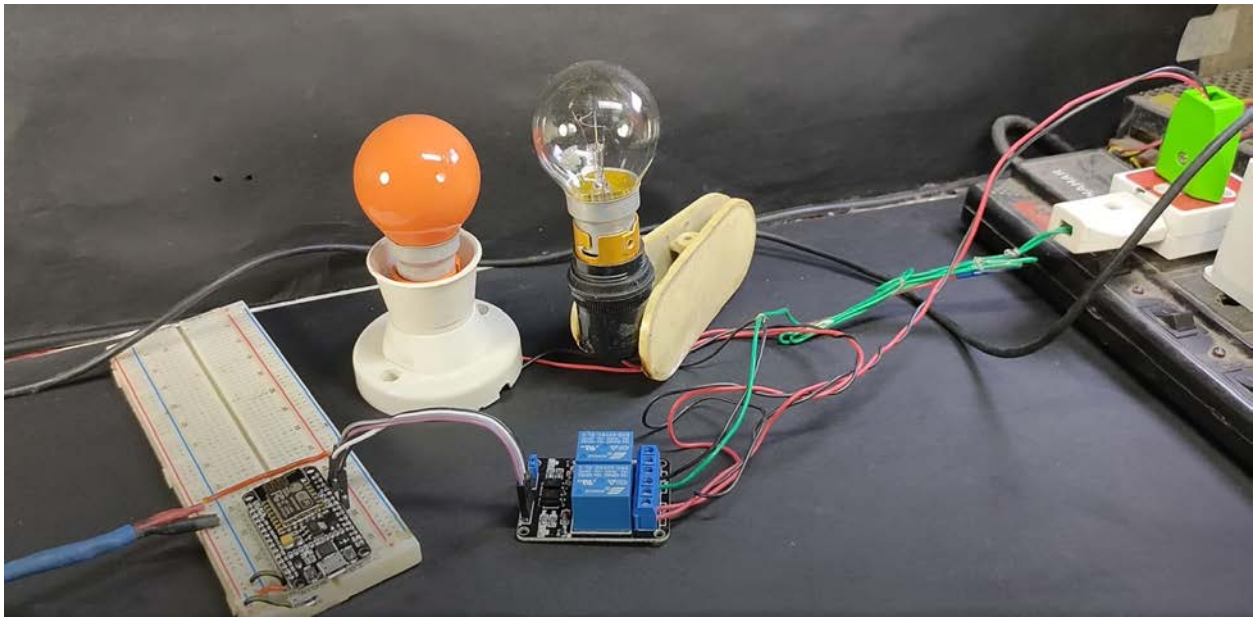


FIG: Circuit diagram of internet controlled switch

Project Picture:





Code:

```
#include <ESP8266WiFi.h>
WiFiServer server(80);
String header;
String output5State = "off";
String output4State = "off";
const int output5 = 5;
const int output4 = 4;
unsigned long currentTime = millis();
unsigned long previousTime = 0;
void setup() {
  pinMode(output5, OUTPUT);
  pinMode(output4, OUTPUT);
  digitalWrite(output5, LOW);
  digitalWrite(output4, LOW);
  WiFi.begin("Sayem", "sajjadaminsayem");
  server.begin();
}
void loop(){
  WiFiClient client = server.available();
  if (client) {
    currentTime = millis();
    previousTime = currentTime;
    String currentLine = "";
    while (client.connected() && currentTime - previousTime <=
2000) {
      currentTime = millis();
      if (client.available()) {
        char c = client.read();
        header += c;
        if (c == '\n') {
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
```

```
            client.println("Connection: close");
            client.println();
            if (header.indexOf("GET /5/on") >= 0) {
              output5State = "on";
              digitalWrite(output5, HIGH);
            } else if (header.indexOf("GET /5/off") >= 0) {
              output5State = "off";
              digitalWrite(output5, LOW);
            } else if (header.indexOf("GET /4/on") >= 0) {
              output4State = "on";
              digitalWrite(output4, HIGH);
            } else if (header.indexOf("GET /4/off") >= 0) {
              output4State = "off";
              digitalWrite(output4, LOW);
            }
            client.println("<!DOCTYPE html><html>");
            client.println("<head><meta name=\"viewport\" conte
nt=\"width=device-width, initial-scale=1\">");
            client.println("<link rel=\"icon\" href=\"data:,\">
");
            client.println("<style>html { font-
family: Helvetica; display: inline-
block; margin: 0px auto; text-align: center;}");
            client.println(".button { background-
color: #195B6A; border: none; color: white; padding: 16px 40px;
");
            client.println("text-decoration: none; font-
size: 30px; margin: 2px; cursor: pointer;}");
            client.println(".button2 {background-
color: #77878A;}</style></head>");
            client.println("<body><h1>ESP8266 Web Server</h1>")
;
            client.println("<p>GPIO 5 - State " + output5State
+ "</p>");
            if (output5State=="off") {
```

```
            client.println("<p><a href=\"/5/on\"><button clas
s=\"button\">ON</button></a></p>");
            } else {
              client.println("<p><a href=\"/5/off\"><button cla
ss=\"button button2\">OFF</button></a></p>");
            }
            client.println("<p>GPIO 4 - State " + output4State
+ "</p>");
            if (output4State=="off") {
              client.println("<p><a href=\"/4/on\"><button clas
s=\"button\">ON</button></a></p>");
            } else {
              client.println("<p><a href=\"/4/off\"><button cla
ss=\"button button2\">OFF</button></a></p>");
            }
            client.println("</body></html>");
            client.println();
            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }
      }
    }
    header = "";
    client.stop();
  }
}
```

Procedure & Working:

In this project, we have used the NodeMCU ESP8266 microcontroller. We have created an HTTP server on NodeMCU. It can handle any kind of HTTP request. When we power up the NodeMCU it connects to a wifi network that we provided in the program. Our Wi-Fi router gives it an IP address. We can visit the webpage that we have created in the NodeMCU server using this IP address. On this webpage, there are two buttons. One is for pin 4 another for pin 5 of the NodeMCU. We have define the 4 and 5 number digital pin of the NodeMCU for our output. When the user presses a switch on the webpage, an HTTP request sends to the server. The server reads the request and finds which pin should be active. After that, the corresponding pin voltage will be high or low.

If the output pin voltage remains high, it will flow in the relay and the relay will be active and the associated component of the relay will turn on.

On the other hand, if the output pin voltage remains low, the relay will be inactive and the associated component of the relay will turn off.

Result:

The main purpose of the internet-controlled switch is to control the electric component of the house from a remote location. We have created the system in our project successfully.

Discussion:

From this project, We have learned that how to control electric components over a network. We have also acquired knowledge about HTTP protocol and HTTP server.

References:

https://www.electronics-lab.com/project/home-automation-using-nodemcu-esp8266-board/

https://randomnerdtutorials.com/esp8266-web-server/

https://www.youtube.com/watch?v=HFGP1YqUPy0