NodeMCU AS A WEB-SERVER

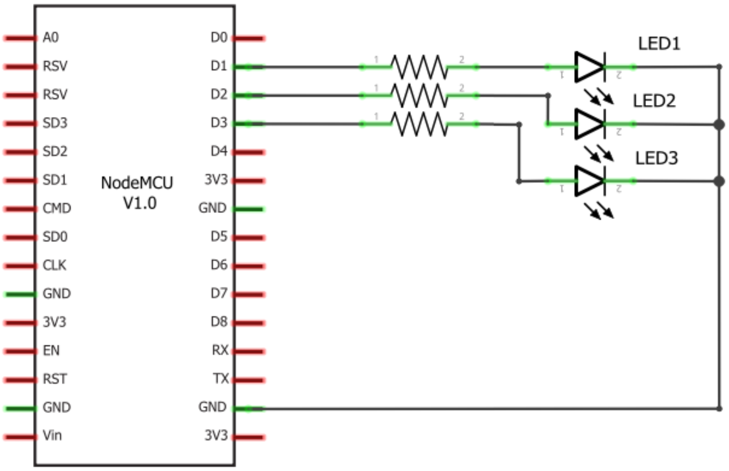
Description:

In this we present NodeMCU to create a simple Web server. Using the ESP8266WiFi library, the device will be able to answer a HTTP request with the Wi-Fi. After opening a browser and navigating to Wi-Fi IP address, NodeMCU will respond with just enough HTML for a browser to display the webpage. This web server is used to serve up web pages that can be accessed from a web browser running on any computer connected to the same network as the Wi-Fi. And also control led's through web application.

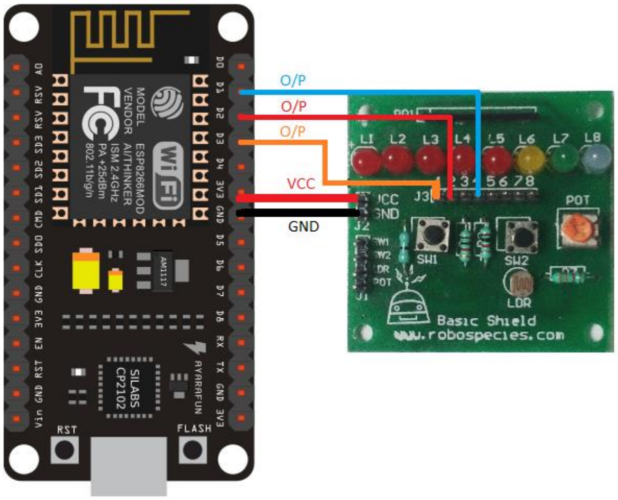
Components Required:

- NodeMCU
- Basic Shield
- Embedded C
- Html

Circuit Diagram:



Connection Diagram:



Library Required:

#include <ESP8266WiFi.h>

Specifications(Nodemcu):

- ✓ Memory - 20kB
- ✓ Storage - 4MB
- ✓ Operating Voltage - 3v

Code:

```
#include <ESP8266WiFi.h>

int LIGHT = D1;

int FAN   = D2;

int AC    = D3;

String readString;

const char* ssid     = "";

const char* password = "";

int Temp;
```

```
// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);
void setup()
{
Serial.begin(9600);
delay(10);
  pinMode(D2, OUTPUT);
  digitalWrite(D2, 1);
   pinMode(D3, OUTPUT);
  digitalWrite(D3, 1);
     pinMode(D4, OUTPUT);
  digitalWrite(D4, 1);

// Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

   while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  // Start the server
  server.begin();
  Serial.println("Server started");
  // Print the IP address
  Serial.println(WiFi.localIP());
}
void loop()
{

// Check if a client has connected
  WiFiClient client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();

        //read char by char HTTP request
        if (readString.length() < 100) {
          //store characters to string
          readString += c;
          //Serial.print(c);
         }

         //if HTTP request has ended
        if (c == '\n') {
```

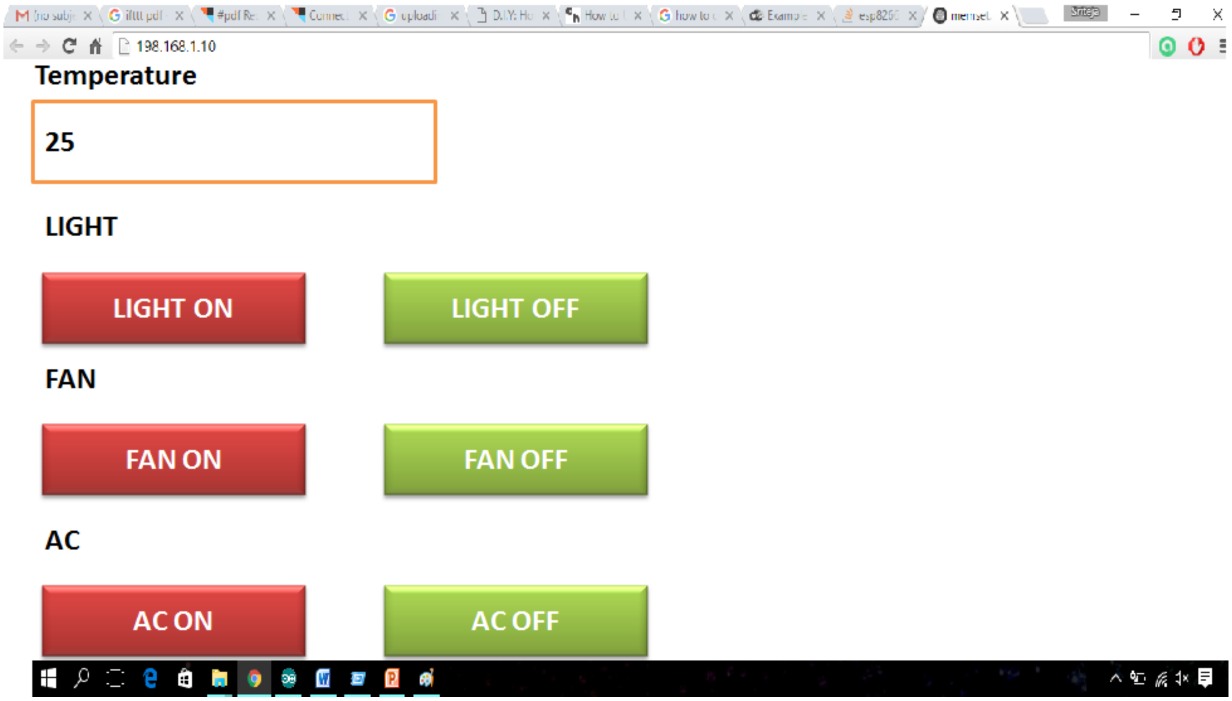
```
        Serial.println(readString); //print to serial monitor for debuging
        Temp = analogRead(A0);
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: close");  // the connection will be closed after completion of the response
        client.println("Refresh: 5");  // refresh the page automatically every 5 sec
        client.println();
        client.println("<!DOCTYPE HTML>");
        client.println("<html>");
        client.println("<head>");
        client.println("<meta charset=\"utf-8\">");
        client.println("<meta http-equiv=\"X-UA-Compatible\" content=\"IE=edge\">");
        client.println("<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
        client.println("<script src=\"https://code.jquery.com/jquery-2.1.3.min.js\"></script>");
        client.println("<link rel=\"stylesheet\" href=\"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css\">");
        client.println("</head><div class=\"container\">");

        client.println("<h1>SMART HOME AUTOMATION HUB</h1><br><br>");
        client.println("<div class=\"row\">");
        client.println("<div class=\"col-sm-4\">");
        client.println("<h2>Temperature</h2>");
        client.println("<input type= \"text\" value= "+String(Temp, DEC)+" class= \"form-control\">");
        client.println("</div >");
        client.println("</div >");
```

```
    client.println("<h2>Light</h2>");
    client.println("<div class=\"row\">");
    client.println("<div class=\"col-md-2\"><a href=\"?pin=LIGHTON\" class=\"btn btn-block btn-lg btn-success\" role=\"button\">LIGHT ON</a></div>");
    client.println("<div class=\"col-md-2\"><a href=\"?pin=LIGHTOFF\" class=\"btn btn-block btn-lg btn-danger\" role=\"button\">LIGHT OFF</a></div>");
    client.println("</div>");
    client.println("<h2>Fan</h2>");
    client.println("<div class=\"row\">");
    client.println("<div class=\"col-md-2\"><a href=\"?pin=FANON\" class=\"btn btn-block btn-lg btn-primary\" role=\"button\">FAN ON</a></div>");
    client.println("<div class=\"col-md-2\"><a href=\"?pin=FANOFF\" class=\"btn btn-block btn-lg btn-warning\" role=\"button\">FAN OFF</a></div>");
    client.println("</div>");
    client.println("<h2>AC</h2>");
    client.println("<div class=\"row\">");
    client.println("<div class=\"col-md-2\"><a href=\"?pin=ACON\" class=\"btn btn-block btn-lg btn-primary\" role=\"button\">AC ON</a></div>");
    client.println("<div class=\"col-md-2\"><a href=\"?pin=ACOFF\" class=\"btn btn-block btn-lg btn-warning\" role=\"button\">AC OFF</a></div>");
    client.println("</div>");
    client.println("</html>");
      delay(1);
      //stopping client
      client.stop();
      //controls the Arduino if you press the buttons
     if (readString.indexOf("?pin=LIGHTON") >0){
         digitalWrite(LIGHT, LOW);
      }
```

```
            else if (readString.indexOf("?pin=LIGHTOFF") >0){
               digitalWrite(LIGHT, HIGH);
           }
           if (readString.indexOf("?pin=FANON") >0){
               digitalWrite(FAN, LOW);
           }
            else if (readString.indexOf("?pin=FANOFF") >0){
               digitalWrite(FAN, HIGH);
           }
             if (readString.indexOf("?pin=ACON") >0){
               digitalWrite(AC, LOW);
           }
           else if (readString.indexOf("?pin=ACOFF") >0){
               digitalWrite(AC, HIGH);
           }
            //clearing string for next read
            readString="";
          }
        }
     }
  }
}
```

Web Page:



198.168.1.10

Temperature

25

LIGHT

FAN

AC