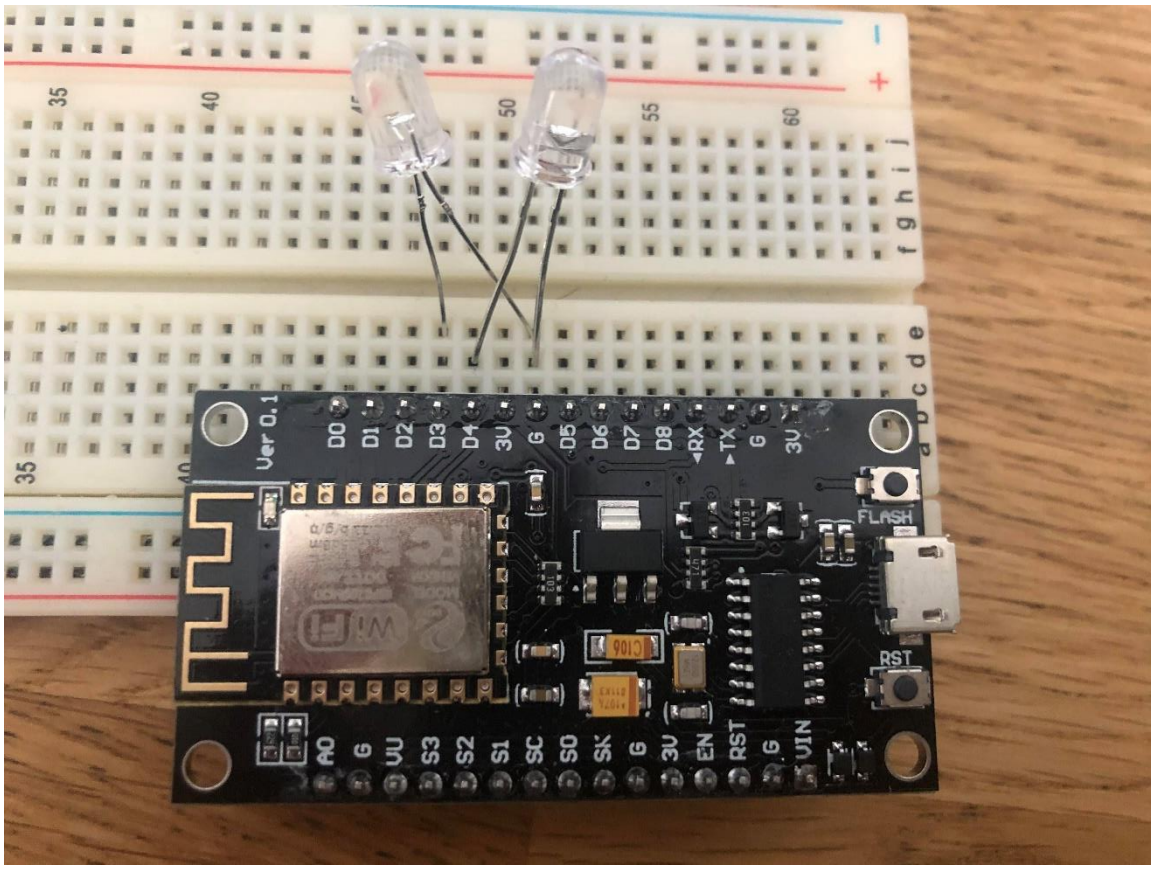
1. Opis

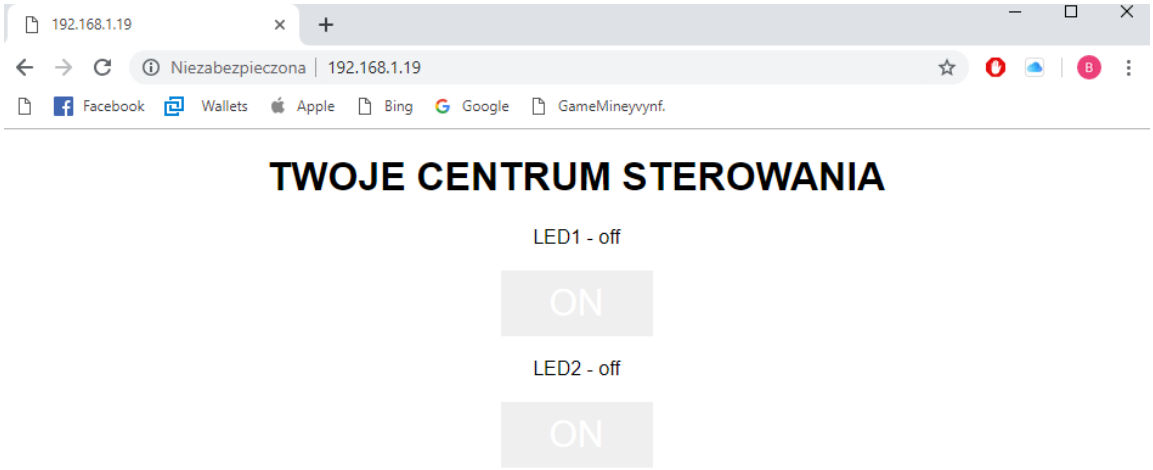
Umożliwienie sterowania zasilaniem różnych urządzeń w sieci domowej.

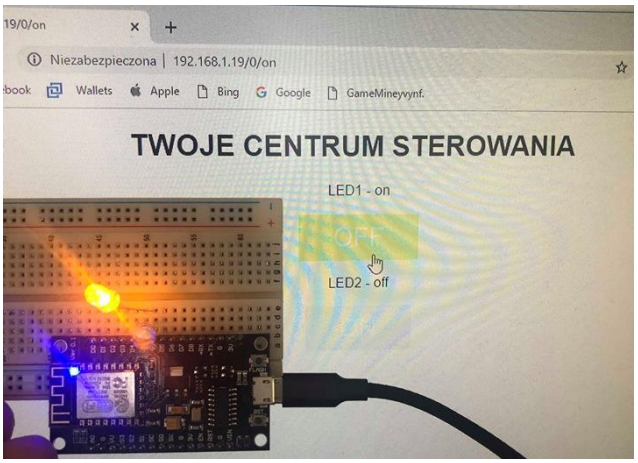
2. Układ

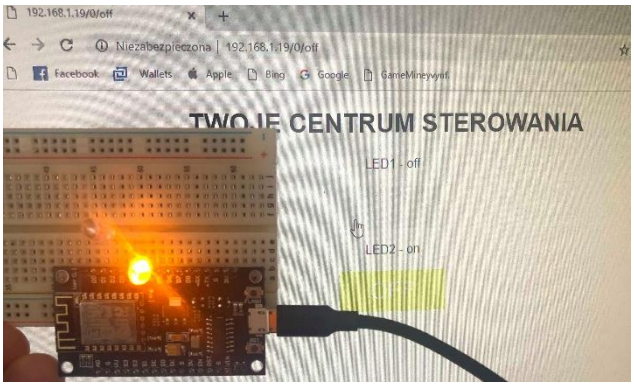


Do budowy układu zastosowałem kontroler NodeMCU, do symulowania przekaźników sterujących zasilaniem wykorzystałem diody LED.3.

3.Wygląd strony







Kod Programu.

```
#include <ESP8266WiFi.h>
```

```
const char* ssid     = "Al2G";
const char* password = "123qweASD";

WiFiServer server(80);

String header;

String output0State = "off"; //poczatkowe stany 2 wyjsc uzywanych ustawione na 0
String output1State = "off";

const int output0 = 0;
const int output1 = 2;

void setup() {
  Serial.begin(115200);

  //ustawianie stanow poczatkowych

  pinMode(output0, OUTPUT);
  pinMode(output1, OUTPUT);

  digitalWrite(output0, LOW);
  digitalWrite(output1, LOW);


  //laczenie sie z wifi (kominikaty wyswietlane na porcie szeregowym)
  Serial.print("Laczenie sie z siecia: ");
  Serial.println(ssid);
```

```
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("Polaczono");
  Serial.println("Adres IP: ");
  Serial.println(WiFi.localIP());
  Serial.println("(wpisz w przegladarce)");
  server.begin();
}

void loop(){
  WiFiClient client = server.available();

//polaczenie sie nowego klienta z "serwerem"
  if (client) {
    Serial.println("New Client.");
    String currentLine = "";
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        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();

            //ustawianie adresow do danej operacji
            if (header.indexOf("GET /0/on") >= 0) {
              Serial.println("LED1 on");
              output0State = "on";
              digitalWrite(output0, HIGH);
            } else if (header.indexOf("GET /0/off") >= 0) {
              Serial.println("LED1 off");
              output0State = "off";
              digitalWrite(output0, LOW);
            } else if (header.indexOf("GET /1/on") >= 0) {
              Serial.println("LED2 on");
              output1State = "on";
              digitalWrite(output1, HIGH);
            } else if (header.indexOf("GET /1/off") >= 0) {
              Serial.println("LED2 off");
              output1State = "off";
              digitalWrite(output1, LOW);
            }

            // Wysweitlanei strony html
            client.println("<!DOCTYPE html><html>");
            client.println("<head><meta name=\"viewport\" content=\"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;}");

//widok przyciskow za pomoca CSS-a

client.println(".button { background-color: #efefef;   border: none;   color: white; padding: 10px 40px;");


client.println("text-decoration: none;    font-size: 30px;    margin: 2px;   cursor: pointer;}");


client.println(".button2 {background-color: #e3f774; }</style></head>");




client.println("<body><h1>TWOJE CENTRUM STEROWANIA</h1>");


client.println("<p>LED1 - " + output0State + "</p>");


if (output0State=="off") {

  client.println("<p><a href=\"/0/on\"><button class=\"button\">ON</button></a></p>");

} else {

  client.println("<p><a href=\"/0/off\"><button class=\"button button2\">OFF</button></a></p>");

}


client.println("<p>LED2 - " + output1State + "</p>");

// Zmiana wyswietlanego napisu w zaleznosci od stanu na pinie

if (output1State=="off") {

  client.println("<p><a href=\"/1/on\"><button class=\"button\">ON</button></a></p>");

} else {

  client.println("<p><a href=\"/1/off\"><button class=\"button button2\">OFF</button></a></p>");
```

```
            }

            client.println("</body></html>");

            client.println();

            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }
      }
    }
    // Clear the header variable
    header = "";
    // Close the connection
    client.stop();
    Serial.println("Client disconnected.");
    Serial.println("");
  }
}
```