

# Sisteme cu microprocesoare

~ Proiect ~

## Controlarea intensitatii unor LED-uri folosind Wi – Fi

Student : Narcis – Alfred Szene

Grupa : 332AC

### Cuprins

Descrierea proiectului.....	2
Componente hardware.....	2
Implementarea hardware. ....	3
Implementarea Software. ....	4
Demonstrarea functionalitatii.....	11
Referinte.....	12

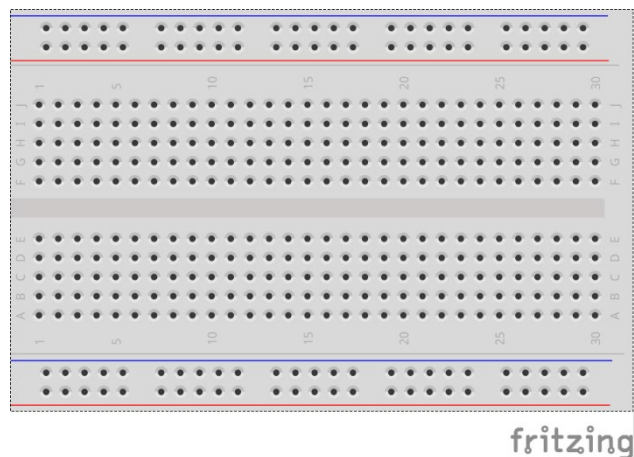
## Descrierea proiectului.

Proiectul consta in controlarea intensitatii a 3 dispozitive LED de culoare **albastru**, **galben** si **rosu**. Controlarea intensitatii se face folosind tehnica **PWM**, iar conectarea la Wi-Fi se realizeaza folosind un modul **ESP8266**.

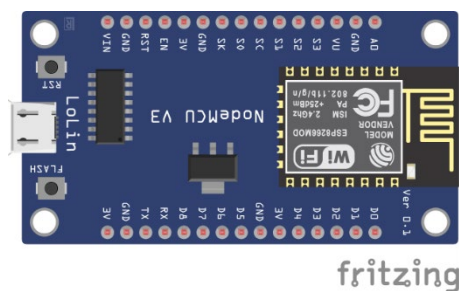
## Componente hardware.

Pentru realizarea acestui proiect s-au folosit urmatoarele componente hardware :

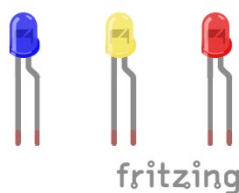
- Breadboard 400 puncte.



- Modul Wi-Fi NodeMCU ESP8266 CP2102



- 3 LED-uri.



- Rezistentă.
- Fire jumper.

## Implementarea hardware.

În schema de mai jos este reprezentată aproximativ schema de mai jos ( Fig. 1 ). Conectarea s-a făcut în acest fel datorită spațiului restrâns de pe breadboard-ul de 400 de puncte.

LED-urile s-au conectat la pinii digitali **D5**, **D6** și **D7**, corespunzător cu **GPIO14**, **GPIO12**, **GPIO13** ( Fig. 2 ).

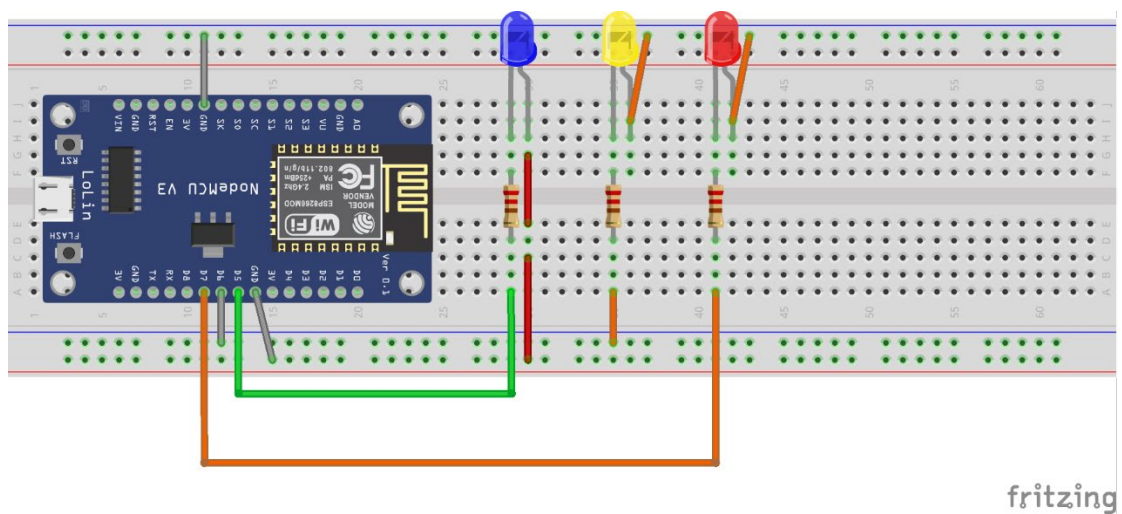


Fig. 1

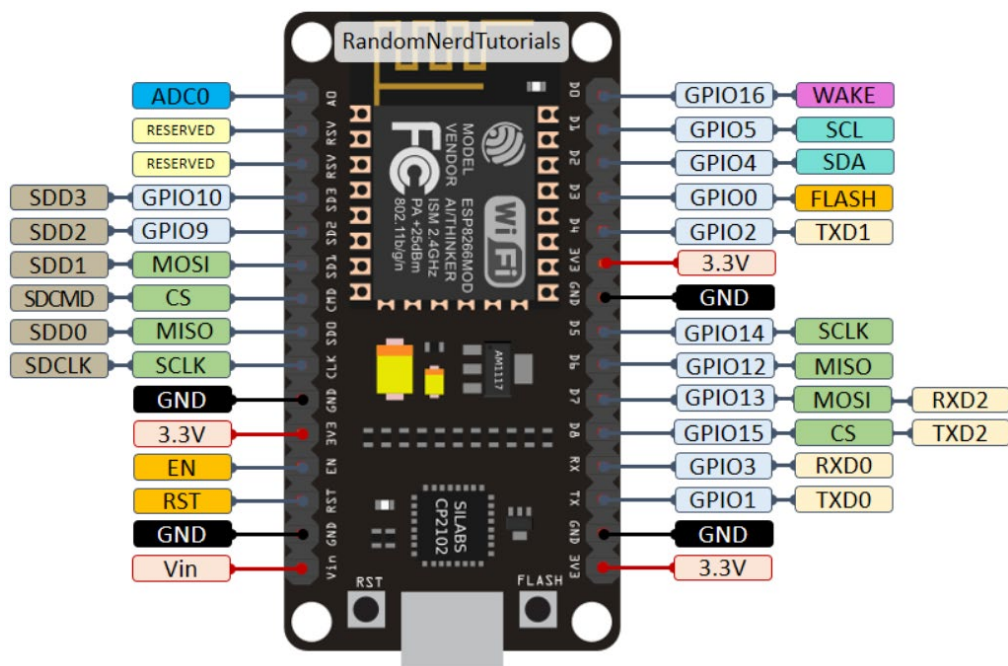


Fig. 2

## Implementarea Software.

Partea Software a proiectului a fost implementata folosind **Arduino IDE**, cu bibliotecile specifice pentru modulul Wi-Fi si pentru sincronizarea cu serverul Web.

Codul aplicatiei poate fi gasit mai jos :

```
//bibliotecile necesare

#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>


//credentiale retea
const char* ssid = "Network Name";
const char* password = "Network Password";


//variabila secventa start
int variable = 0;


//led albastru D5
const int led_pin1 = 14; //GPIO
//led galben D6
const int led_pin2 = 12; //GPIO
//led rosu D7
const int led_pin3 = 13; //GPIO


//valori pentru slider.
String slider_value1 = "0";
String slider_value2 = "0";
String slider_value3 = "0";
//caractere pentru a identifica led-ul corespunzator
const char* input_parameter1 = "value";
const char* input_parameter2 = "value2";
const char* input_parameter3 = "value3";
```

```

//crearea serverului
AsyncWebServer server(80);

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>

<head>

//scalarea paginii pentru orice tip de dispozitiv
    <meta name="viewport" content="width=device-width, initial-scale=1">

//titlul paginii
    <title>Controlarea intensitatii unui LED prin Wi-fi Server</title>

    <style>

//modelarea paginii web folosind stil CSS
        html {font-family: Times New Roman; display: inline-block; text-align:
center;}

        h {font-size: 2.3rem;}
        p {font-size: 2.0rem;}

//slider1
        body {max-width: 400px; margin:0px auto; padding-bottom: 25px;}

        .slider { -webkit-appearance: none; margin: 14px; width: 360px; height:
25px; background: #1500ff;

            outline: none; -webkit-transition: .2s; transition: opacity .2s;}

        .slider::-webkit-slider-thumb {-webkit-appearance: none; appearance:
none; width: 35px; height: 35px; background:#01070a; cursor: pointer;}

        .slider::-moz-range-thumb { width: 35px; height: 35px; background:
#01070a; cursor: pointer; }

//slider2
        h2 {font-size: 2.3rem;}
        p2 {font-size: 2.0rem;}

//slider2
        body {max-width: 400px; margin:0px auto; padding-bottom: 25px;}

        .slider2 { -webkit-appearance: none; margin: 14px; width: 360px;
height: 25px; background: #f2ff00;

            outline: none; -webkit-transition: .2s; transition: opacity .2s;}

        .slider2::-webkit-slider-thumb {-webkit-appearance: none; appearance:
none; width: 35px; height: 35px; background:#01070a; cursor: pointer;}

        .slider2::-moz-range-thumb { width: 35px; height: 35px; background:
#01070a; cursor: pointer; }

```

```

        //slider3

        h3 {font-size: 2.3rem;}

        p3 {font-size: 2.0rem;}

        body {max-width: 400px; margin:0px auto; padding-bottom: 25px;}

        .slider3 { -webkit-appearance: none; margin: 14px; width: 360px;
height: 25px; background: #ff0000;

        outline: none; -webkit-transition: .2s; transition: opacity .2s;}

        .slider3::-webkit-slider-thumb {-webkit-appearance: none; appearance:
none; width: 35px; height: 35px; background:#01070a; cursor: pointer;}

        .slider3::-moz-range-thumb { width: 35px; height: 35px; background:
#01070a; cursor: pointer; }

    </style>
</head>

//definirea body-ului HTML
<body>

    <h>Proiect SMP - Controlarea intensitatii unui LED</h>

    //parametrii pentru slider1

    <p><span id="textslider_value1">%SLIDERVALUE1%</span></p>

    <p><input type="range" onchange="updateSliderPWM(this)" id="pwmSlider1"
min="0" max="255" value="%SLIDERVALUE1%" step="1" class="slider"></p>

    <h2></h2>

    //parametrii pentru slider2

    <p2><span id="textslider_value2">%SLIDERVALUE2%</span></p2>

    <p2><input type="range" onchange="updateSliderPWM2(this)" id="pwmSlider2"
min="0" max="255" value="%SLIDERVALUE2%" step="1" class="slider2"></p2>

    <h3></h3>

    //parametrii pentru slider3

    <p3><span id="textslider_value3">%SLIDERVALUE3%</span></p3>

    <p3><input type="range" onchange="updateSliderPWM3(this)" id="pwmSlider3"
min="0" max="255" value="%SLIDERVALUE3%" step="1" class="slider3"></p3>

<script>

//functie slider 1

function updateSliderPWM(element)

{

    var slider_value1 = document.getElementById("pwmSlider1").value;

    document.getElementById("textslider_value1").innerHTML = slider_value1;

    console.log(slider_value1);

```

```

//HTTP GET request
var xhr = new XMLHttpRequest();
xhr.open("GET", "/slider?value="+slider_value1, true);
xhr.send();
}

//functie slider2
function updateSliderPWM2(element)
{
var slider_value2 = document.getElementById("pwmSlider2").value;
document.getElementById("textslider_value2").innerHTML = slider_value2;
console.log(slider_value2);
//HTTP GET request
var xhr = new XMLHttpRequest();
xhr.open("GET", "/slider?value2="+slider_value2, true);
xhr.send();
}

//functie slider3
function updateSliderPWM3(element)
{
var slider_value3 = document.getElementById("pwmSlider3").value;
document.getElementById("textslider_value3").innerHTML = slider_value3;
console.log(slider_value3);
//HTTP GET request
var xhr = new XMLHttpRequest();
xhr.open("GET", "/slider?value3="+slider_value3, true);
xhr.send();
}
</script>
</body>
</html>
)rawliteral";
//returnarea valorii din slider

```

```

String processor(const String& var)
{
    if (var == "SLIDERVALUE1")
    {
        return slider_value1;
    }
    else if (var == "SLIDERVALUE2")
    {
        return slider_value2;
    }
    else return slider_value3;
    return String();
}

void setup()
{
    //setarea pinilor pentru secventa de start
    pinMode(14,OUTPUT);
    pinMode(12,OUTPUT);
    pinMode(13,OUTPUT);
    Serial.begin(115200);
    WiFi.begin(ssid, password);
    //conectarea la internet a modulului.
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.println("Connecting...");
    }
    Serial.println(WiFi.localIP());
    while(variable==0)
    {
        //secv 1
        digitalWrite(14,HIGH);
        delay(250);
        digitalWrite(14,LOW);
    }
}

```



```
digitalWrite(12,HIGH);  
delay(250);  
digitalWrite(12,LOW);  
digitalWrite(13,HIGH);  
delay(250);  
digitalWrite(13,LOW);  
delay(250);
```

```
//secv2  
digitalWrite(13,HIGH);  
delay(250);  
digitalWrite(13,LOW);  
digitalWrite(12,HIGH);  
delay(250);  
digitalWrite(12,LOW);  
digitalWrite(14,HIGH);  
delay(250);  
digitalWrite(14,LOW);
```

```
//secv3  
digitalWrite(14,HIGH);  
digitalWrite(12,HIGH);  
digitalWrite(13,HIGH);  
delay(300);  
digitalWrite(14,LOW);  
digitalWrite(12,LOW);  
digitalWrite(13,LOW);  
delay(300);  
digitalWrite(14,HIGH);  
digitalWrite(12,HIGH);  
digitalWrite(13,HIGH);  
delay(300);  
digitalWrite(14,LOW);  
digitalWrite(12,LOW);
```

```

    digitalWrite(13,LOW);
    delay(300);
    digitalWrite(14,HIGH);
    digitalWrite(12,HIGH);
    digitalWrite(13,HIGH);
    delay(300);
    digitalWrite(14,LOW);
    digitalWrite(12,LOW);
    digitalWrite(13,LOW);
    variable=1;
}

//GET request pentru /root URL
server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request)
{
    request->send_P(200, "text/html", index_html, processor);
});

//GET request pentru /slider URL
server.on("/slider", HTTP_GET, [] (AsyncWebServerRequest *request)
{
    String message;
//request slider1
    if (request->hasParam(input_parameter1))
    {
        message = request->getParam(input_parameter1)->value();
        slider_value1 = message;
        analogWrite(led_pin1,slider_value1.toInt());
    }
//request slider2
    if (request->hasParam(input_parameter2))
    {
        message = request->getParam(input_parameter2)->value();
        slider_value2 = message;
        analogWrite(led_pin2,slider_value2.toInt());
    }
}

```

```

    }

    //request slider3

    if (request->hasParam(input_parameter3))
    {
        message = request->getParam(input_parameter3)->value();
        slider_value3 = message;
        analogWrite(led_pin3,slider_value3.toInt());
    }

    });

    //pornirea serverului

    server.begin();

}

void loop()

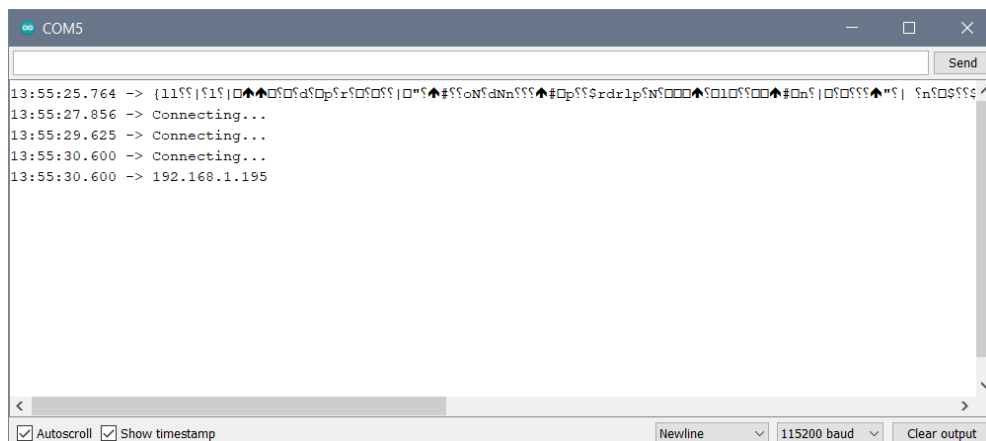
{

}

```

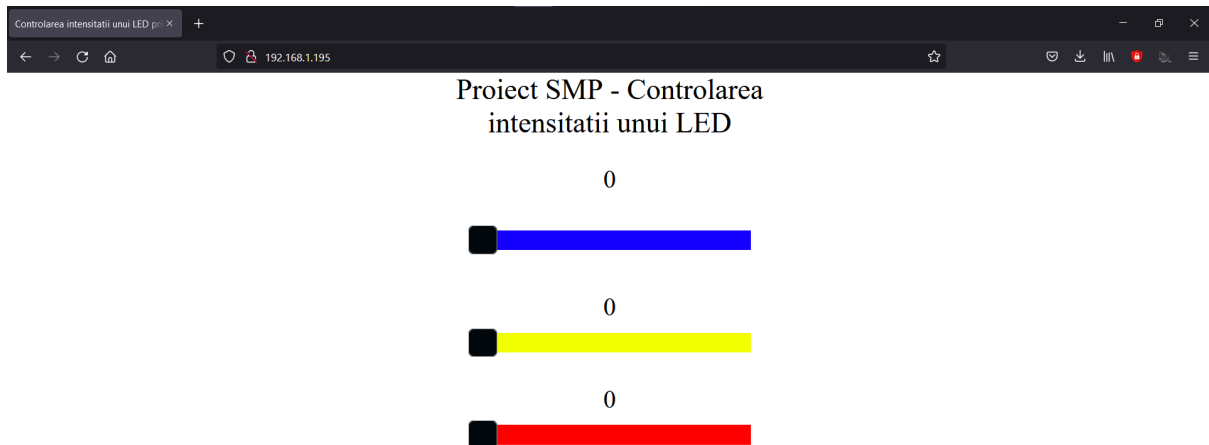
## Demonstrarea functionalitatii.

Dupa ce am alimentat placa la un dispozitiv, putem deschide Arduino IDE. Dupa apasarea butonului **RST**, modulul va incerca sa stabileasca o conexiune cu retea. In momentul in care conexiunea a reusit, pe monitorul serial ( Serial Monitor ) va aparea adresa IP a serverului ( Fig. 3 ), iar LED-urile vor incepe o secventa de start, in care se aprind si se sting.



*Fig. 3*

In acest moment ne putem conecta la adresa IP indicata. Pagina Web va arata asa :



*Fig. 4*

De aici putem controla intensitatea fiecarui LED in parte. Valorile merg de la 0 la 255.

## Referinte.

- <https://microcontrollerslab.com>
- <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>