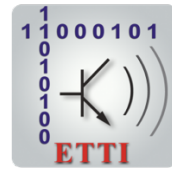


Universitatea “POLITEHNICA” din București



Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



Temă finală – Proiect informatică aplicată –

Student : Ștergărel Elisabeta-Elena

Grupa: 411C

Cod PIA Hunt: 510

Profesor coordonator: Prof. Univ. Dr. Ing. Stanciu Mihai

Data: 25.05.2023

An universitar: 2022-2023

Cod PIA Hunt: 510

Specificațiile temei:

Parametrul de afișat pe 7 segmente:

uv_index_max ; fiind 7 valori, se vor afișa pe rând pe afișajul dual cu 7 segmente, cu durată de 2.5 secunde pentru o valoare. Întrucât sunt valori pe mai mult de 2 cifre, de forma xy.z (1 zecimală), se vor afișa

- mai întâi partea întreagă, timp de 1s,
- apoi partea zecimală, cu punctul zecimal aprins în față, timp de 1s,
- apoi afișajul stins pt. 0.5s

Parametru de afișat pe bargraph:

apinderea a câte unui singur LED pe bargraph: LED-ul 1 când pe afișaj e valoarea pt. ziua 1, LED-ul 2 pt ziua 2, etc

Valorile de afișat pe serială:

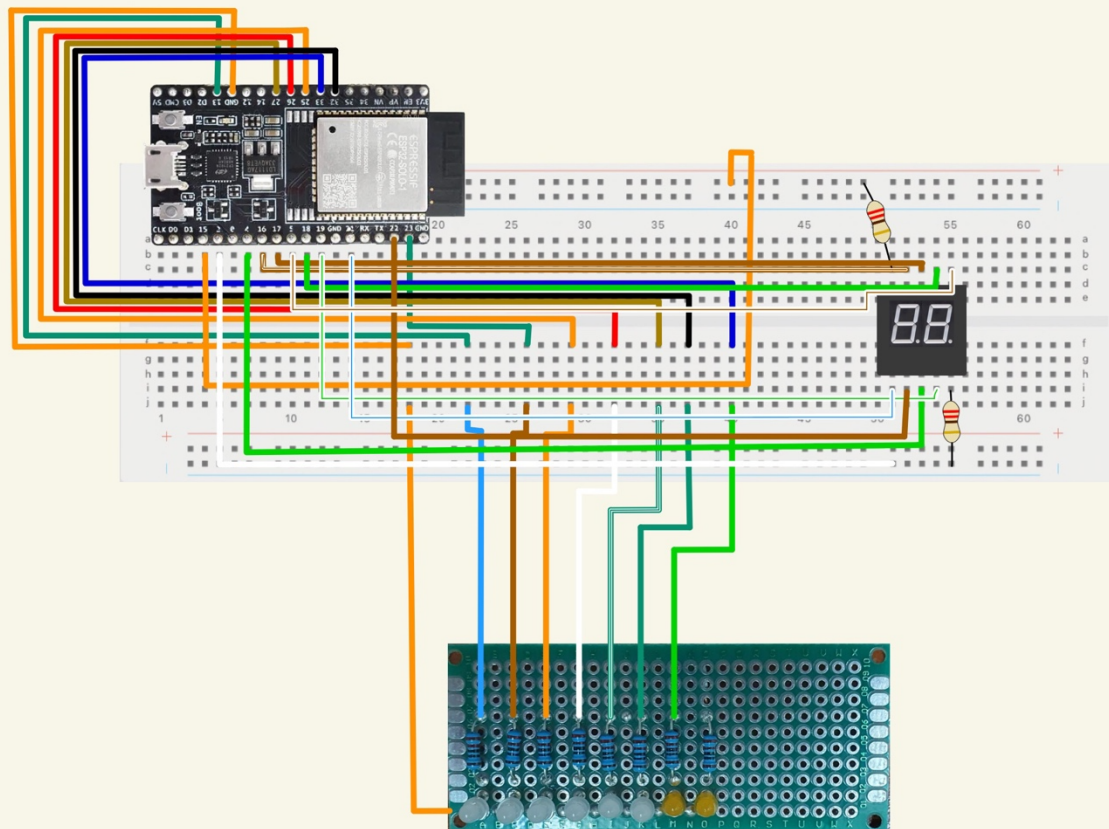
La început, să se scrie "PIA - Echipa 510", numele rețelei WiFi, RSSI-ul, să apară cele 7 valori separate prin virgulă, pe același rând, așa cum sunt citite din API, de exemplu: 05:50,05:52,05:53,...

Pe măsură ce apar pe afișajul cu 7 segmente, valorile să apară și pe serială, câte 2 pe un rând, precedate de nr. zilei și 2 două puncte, de exemplu:

1: xxx 2: yyy (xxx este valoarea parametrului pt prima zi)
3: zzz 4: ttt

...

Schemă circuit:



Cod Proiect:

```
#include <ArduinoJson.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <String.h>

const char* ssid = "B100";
const char* password = "";
float values[8];
const int A = 4;
const int B = 16;
const int C = 17;
const int D = 5;
const int E = 18;
const int F = 19;
const int G = 21;
const int comma = 22;

const int digit1 = 2;
const int digit2 = 15;

const int bits[96] = {
    1, 1, 1, 1, 1, 1, 0, 0,
    0, 1, 1, 0, 0, 0, 0, 0,
    1, 1, 0, 1, 1, 0, 1, 0,
    1, 1, 1, 1, 0, 0, 1, 0,
    0, 1, 1, 0, 0, 1, 1, 0,
    1, 0, 1, 1, 0, 1, 1, 0,
    1, 0, 1, 1, 1, 1, 1, 0,
    1, 1, 1, 0, 0, 0, 0, 0,
    1, 1, 1, 1, 1, 1, 1, 0,
    1, 1, 1, 1, 0, 1, 1, 0,
};

void printSerial(int i)
{
    if(i%2==0)
    {
        Serial.print(i+1);
        Serial.print(": ");
        Serial.print(values[i]);
        Serial.print(" ");
    }
    else
    {
        Serial.print(i+1);
        Serial.print(": ");
        Serial.print(values[i]);
        Serial.println(" ");
    }
}

void afisareStinsa()
{
    digitalWrite(digit1, HIGH);
    digitalWrite(digit2, LOW);
}
```

```

        digitalWrite(A, 0);
        digitalWrite(B, 0);
        digitalWrite(C, 0);
        digitalWrite(D, 0);
        digitalWrite(E, 0);
        digitalWrite(F, 0);
        digitalWrite(G, 0);
        digitalWrite(comma, 0);
    delay(5);
    digitalWrite(digit1, LOW);
    digitalWrite(digit2, HIGH);
    digitalWrite(A, 0);
    digitalWrite(B, 0);
    digitalWrite(C, 0);
    digitalWrite(D, 0);
    digitalWrite(E, 0);
    digitalWrite(F, 0);
    digitalWrite(G, 0);
    digitalWrite(comma, 0);
    delay(5);

    digitalWrite(digit2, LOW);
}

void afisareNumarFractional(String a) {
    char digit1R = a[0];

    digitalWrite(digit1, HIGH);
    digitalWrite(digit2, LOW);
    digitalWrite(A, 0);
    digitalWrite(B, 0);
    digitalWrite(C, 0);
    digitalWrite(D, 0);
    digitalWrite(E, 0);
    digitalWrite(F, 0);
    digitalWrite(G, 0);
    digitalWrite(comma, 1);
    delay(5);
    digitalWrite(digit1, LOW);
    digitalWrite(digit2, HIGH);
    digitalWrite(A, bits[String(digit1R).toInt()*8+0]);
    digitalWrite(B, bits[String(digit1R).toInt()*8+1]);
    digitalWrite(C, bits[String(digit1R).toInt()*8+2]);
    digitalWrite(D, bits[String(digit1R).toInt()*8+3]);
    digitalWrite(E, bits[String(digit1R).toInt()*8+4]);
    digitalWrite(F, bits[String(digit1R).toInt()*8+5]);
    digitalWrite(G, bits[String(digit1R).toInt()*8+6]);
    digitalWrite(comma, bits[String(digit1R).toInt()*8+7]);
    delay(5);
    digitalWrite(digit2, LOW);
}

void afisareNumarReal(String a) {
    char nr1 = a.charAt(0);
    char nr2 = a.charAt(1);
    digitalWrite(digit1, HIGH);
    digitalWrite(digit2, LOW);

    if(nr1>='0' && nr1<='9')
    {
        digitalWrite(A, bits[String(nr1).toInt()*8+0]);
        digitalWrite(B, bits[String(nr1).toInt()*8+1]);
        digitalWrite(C, bits[String(nr1).toInt()*8+2]);
    }
}

```

```

digitalWrite(D, bits[String(nr1).toInt()*8+3]);
digitalWrite(E, bits[String(nr1).toInt()*8+4]);
digitalWrite(F, bits[String(nr1).toInt()*8+5]);
digitalWrite(G, bits[String(nr1).toInt()*8+6]);
digitalWrite(comma, bits[String(nr1).toInt()*8+7]);
}
else if(nr1=='?')
{
    digitalWrite(A, 0);
    digitalWrite(B, 0);
    digitalWrite(C, 0);
    digitalWrite(D, 0);
    digitalWrite(E, 0);
    digitalWrite(F, 0);
    digitalWrite(G, 0);
    digitalWrite(comma, 0);
}
delay(5);
digitalWrite(digit1, LOW);
digitalWrite(digit2, HIGH);

digitalWrite(A, bits[String(nr2).toInt()*8+0]);
digitalWrite(B, bits[String(nr2).toInt()*8+1]);
digitalWrite(C, bits[String(nr2).toInt()*8+2]);
digitalWrite(D, bits[String(nr2).toInt()*8+3]);
digitalWrite(E, bits[String(nr2).toInt()*8+4]);
digitalWrite(F, bits[String(nr2).toInt()*8+5]);
digitalWrite(G, bits[String(nr2).toInt()*8+6]);
digitalWrite(comma, bits[String(nr2).toInt()*8+7]);
delay(5);

digitalWrite(digit2, LOW);
}
void afisareNumar(float nr)
{
    int a,b;
    String strA,strB;
    a = static_cast<int>(nr);
    float ok = static_cast<int>(nr * 10) % 10;
    b = int(ok);
    if(a<=9 && a>=0)
        strA = "?" + String(a);
    else
        strA = String(a);

    strB = String(b);

    unsigned long timpInceput = millis();
    while (millis() - timpInceput < 1000)
    {
        afisareNumarReal(strA);
    }
    timpInceput = millis();
    while (millis() - timpInceput < 1000)
    {
        afisareNumarFractional(strB);
    }
    timpInceput = millis();

    while (millis() - timpInceput < 500)
    {

```

```

        afisareStinsa();
    }

}

void ledStins()
{
    digitalWrite(13, LOW);
    digitalWrite(23, LOW);
    digitalWrite(25, LOW);
    digitalWrite(26, LOW);
    digitalWrite(27, LOW);
    digitalWrite(32, LOW);
    digitalWrite(33, LOW);
}

void regulaLed(int i)
{
    ledStins();
    switch (i)
    {
        case 0:
            digitalWrite(13, HIGH);
            break;
        case 1:
            digitalWrite(23, HIGH);
            break;
        case 2:
            digitalWrite(25, HIGH);
            break;
        case 3:
            digitalWrite(26, HIGH);
            break;
        case 4:
            digitalWrite(27, HIGH);
            break;
        case 5:
            digitalWrite(32, HIGH);
            break;
        case 6:
            digitalWrite(33, HIGH);
            break;
        default:
            break;
    }
}

void setup() {
    pinMode(A, OUTPUT);
    pinMode(B, OUTPUT);
    pinMode(C, OUTPUT);
    pinMode(D, OUTPUT);
    pinMode(E, OUTPUT);
    pinMode(F, OUTPUT);
    pinMode(G, OUTPUT);
    pinMode(comma, OUTPUT);
    pinMode(digit1, OUTPUT);
    pinMode(digit2, OUTPUT);
    pinMode(13, OUTPUT);
    pinMode(23, OUTPUT);
    pinMode(25, OUTPUT);
    pinMode(26, OUTPUT);
}

```

```

pinMode(27, OUTPUT);
pinMode(32, OUTPUT);
pinMode(33, OUTPUT);
Serial.begin(115200);
Serial.println("PIA - Echipa 510");
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
    delay(1000);
}
Serial.println("");
Serial.print("numele retelei WiFi: ");
Serial.println(WiFi.SSID());
Serial.print("RSSI: ");
Serial.println(WiFi.RSSI());

if (WiFi.status() == WL_CONNECTED)
{
    HTTPClient http;

    String url = "https://api.open-
meteo.com/v1/forecast?latitude=44.43&longitude=26.14&timezone=auto&current_
weather=true&daily=uv_index_max";

    http.begin(url);

    int httpResponseCode = http.GET();
    if (httpResponseCode == 200)
    {
        String payload = http.getString();

        DynamicJsonDocument doc(1024);
        DeserializationError error = deserializeJson(doc, payload);

        if (error)
        {
            Serial.println("Eroare (JSON 1)");
        }
        else
        {
            JsonArray vectorUV = doc["daily"]["uv_index_max"];
            for (size_t i = 0; i < vectorUV.size(); i++)
            {
                float vectorValori = vectorUV[i].as<float>();
                values[i]=vectorValori;
                if(i<vectorUV.size()-1)
                {
                    Serial.print(values[i]);
                    Serial.print(",");
                }
                else
                {
                    Serial.print(values[i]);
                }
            }
        }
    }
    http.end();
}
Serial.println(" ");

```



```
for(int i=0;i<7;i++)
{
    printSerial(i);
    regulaLed(i);
    afisareNumar(values[i]);

}
ledStins();

}
void loop() {

}
```