

Haine inteligente

Miron Cezar Andrei

Pentru realizarea proiectului, am folosit următoarele componente :

- Arduino Nano
- Senzor de temperatură TMP102
- Senzor analogic pentru monitorizarea razelor UV
- Senzor temperatură și umiditate – DHT11
- Buzzer piezoelectric
- Display Nokia 5110
- Breadboard + fire de interconectare

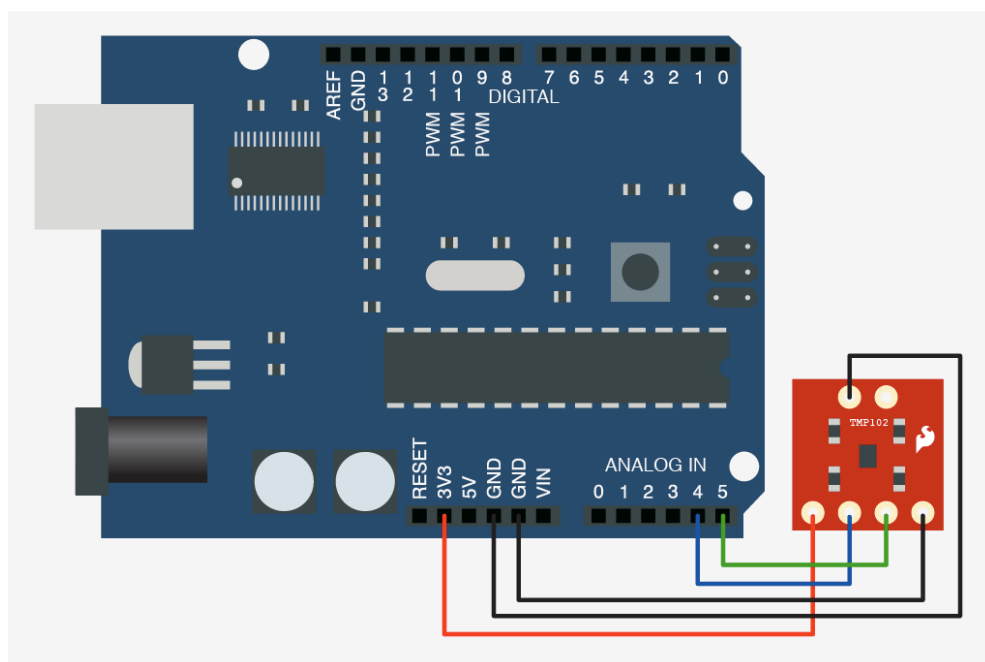
Etape de proiectare :

1. Realizarea comunicației între senzorul de temperatură și Arduino

Pentru a realiza aceasă comunicare, am ales să folosesc protocolul de comunicație I2C, protocol suportat și de către senzorul ales de mine. Acest sensor este este unul digital, precizia valorilor înregistrate fiind mult mai bună decât în cazul senzorilor analogici. (ce depind de ADC-ul cu care sunt interpretați)

Conexiunea cu Arduino :

- VCC → 3.3V
- GND → GND
- SDA → SDA/A4
- SCL → SCL/A5



*Conexiunea cu Arduino Nano este identică

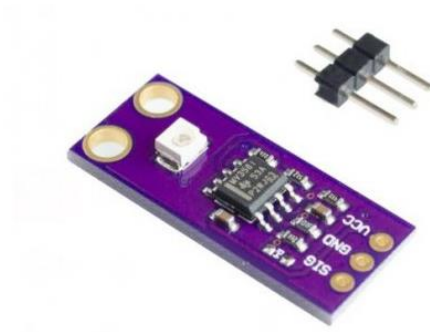
Pentru a citi datele oferite de senzor, m-am folosit de biblioteca SparkFunTMP102. După ce am inițializat obiectul, funcția readTempC() îmi returnează periodic valoarea temperaturii citite.

2. Conexiunea senzorului UV cu Arduino

Fiind un senzor analogic, citesc periodic valoarea lui, folosind un pin din Nano. Această valoare este interpretată în cod și oferă un indice UV.

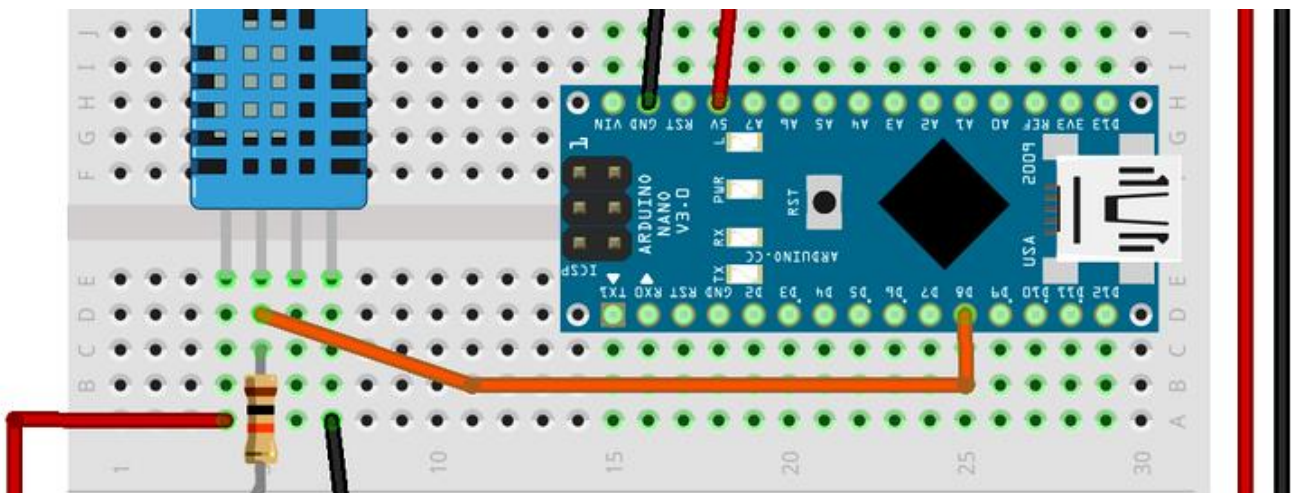
Conexiunea cu Arduino :

- VCC – 5V
- GND – GND
- SIG – A1



3. Conexiunea senzorului DHT11 cu Arduino

- VCC – 5V
- GND – GND
- SIG – pin 7

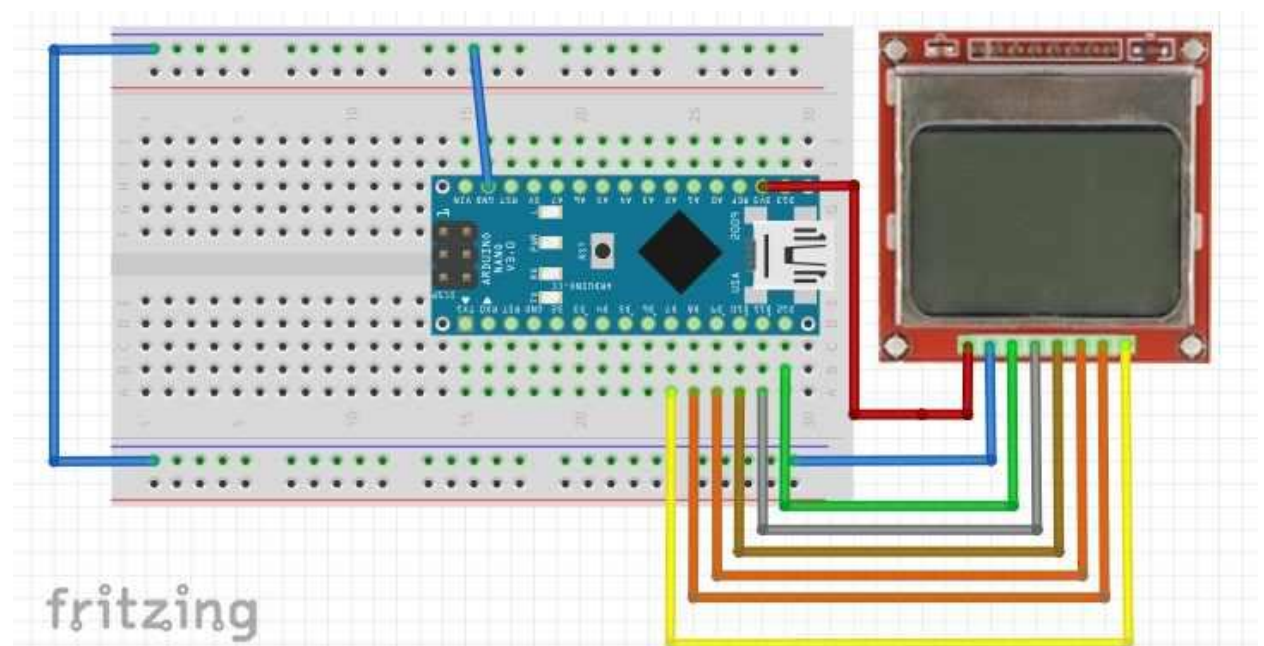


4. Conectarea afisajului cu Arduino

Display-ul ales este un display folosit în telefoanele Nokia 5110. Acesta comunică prin protocolul SPI cu NANO. Pentru a popula display-ul cu date utile, m-am folosit de biblioteca Nokia5110.

Conexiunea cu Arduino :

- LCD serial clock (SCLK) - pin 13
- MOSI (DIN) - pin 11
- Data/Command (DC) - pin 5
- Chip Select (CS) - pin 4
- Reset (RST) - pin 3



Rezultatul final

Scopul proiectului este acela de a atenționa o persoană atunci când temperatura resimțită la nivelul corpului trece de xx grade C și razele UV ating un prag superior. Atenționarea se face atât vizual (prin display) cât și auditiv, folosind un buzzer. Totodata, pe display se poate observa și umiditatea relativă.


```

B00000111,B11111111,B11111111,B11111111,B11111111,B
00000000,
B00000000,B00000001,B11111111,B11111111,B11111111,B
00000000,
B00000000,B00000001,B11111111,B11111111,B11111110,B
00000000,
B00000000,B00000000,B11111111,B11111111,B11111110,B
00000000,
B00000000,B00000000,B00000000,B00000000,B00000000,B
00000000,
B00000000,B00000000,B00000000,B00000000,B00000000,B
00000000,
B00000000,B00000000,B00000000,B00000000,B00000000,B
00000000,
B00000000,B00000000,B00000000,B00000000,B00000000,B
00000000,
};

```

```

static const unsigned char PROGMEM soare_bmp[] = {
    B10000010,
    B01000100,
    B00011000,
    B11111111,
    B00011000,
    B01000100,
    B10000001
};

```

```

static const unsigned char PROGMEM picatura_bmp[] = {
    B00100000,
    B00100000,
    B01110000,
    B11111000,
    B11111000,
    B11111000,
    B01111000
};

```

```

static const unsigned char PROGMEM baterie_bmp[] = {
    B00111100,
    B00111100,
    B11111111,
    B11100111,
    B11100111,
    B11100111,
    B11100111,
    B11100111,
    B11100111,
    B11100111,
    B11111111
};

```

```

void setup() {
    Serial.begin(9600);
    sensor0.begin();
    Wire.begin();
    display.begin();

```

```

    display.setContrast (60);
    display.display();
    delay(2000);

```

```

    display.clearDisplay();
    display.display();

```

```

}

```

```

void loop() {

```

```

    float temperatureS = sensor0.readTempC();
    //senzor uv - sensorValue = analogRead(A1) unde a1 este
    pinul analogic pt uv
    // indexUv = check(sensorValue);
    indexUv = 3;
    byte temperature = 0;
    byte humidity = 0;
    int err = SimpleDHTerrSuccess;
    if ((err = dht11.read(pinDHT11, &temperature, &humidity,
    NULL)) != SimpleDHTerrSuccess) {
        Serial.print("Read DHT11 failed, err=");
        Serial.println(err);delay(1000);
        return;
    }

```

```

    Serial.print("Val adc ");
    Serial.print(sensorValue);
    Serial.print(" Indice UV : ");
    Serial.println(indexUv);
    Serial.print("Spk tmp ");
    Serial.println(temperatureS);

```

```

/*
    Serial.println("TEST BUZZER");
    tone(BUZZER_PIN, 3000 , 500);
    delay(1000);
    noTone(BUZZER_PIN);
*/
    Serial.println("TEST LCD");

```

```

    display.clearDisplay();
    display.setCursor (30,0);
    display.setTextColor (BLACK);
    display.setTextSize(1);
    display.setTextColor (BLACK);
    display.setCursor (0,10);
    display.setTextSize(1);
    display.print(temperatureS,1);
    display.setCursor (25,10);
    display.print("o");
    display.setCursor (32,10);
    display.setTextSize(1);
    display.print("C");
    display.setCursor (36,10);
    display.print(" ");

```

```

display.setTextColor (BLACK);
display.setCursor (0,20);
display.setTextSize(1);
display.print("H :");
display.print((int)humidity,1);
display.setCursor (26,20);
display.print(" ");
display.setCursor (55,20);
display.setTextSize(1);
display.print("!.");
display.print(indexUv);

if((int)humidity > 29)
    display.drawBitmap(6,40,picatura_bmp,5,8,BLACK);
if(temperatureS > 27)
{
    display.setCursor (40,30);
    display.print("Alarma!");
    tone(BUZZER_PIN, 3000 , 500);
    delay(1000);
    noTone(BUZZER_PIN);
}

//if(1 == 1)
// display.drawBitmap(75,0,baterie_bmp,8,11,BLACK);

display.display();

delay(1000);
}

byte check(int index){

if(index<=55)
{
    return 0;
}

else if(index > 55 && index <= 227)
{
    return 1;
}

else if(index > 227 && index <= 318)
{
    return 2;
}

else if(index > 318 && index <= 408)

```

```

{
    return 3;
}

else if(index > 408 && index <= 503)
{
    return 4;
}

else if(index > 503 && index <= 606)
{
    return 5;
}

else if(index > 606 && index <= 696)
{
    return 6;
}

else if(index > 696 && index <= 795)
{
    return 7;
}

else if(index > 795 && index <= 881)
{
    return 8;
}

else if(index > 881 && index <= 976)
{
    return 9;
}

else if(index > 976 && index <= 1079)
{
    return 10;
}

else if (index > 1079 && index <= 1170)
{
    return 11;
}
}

```