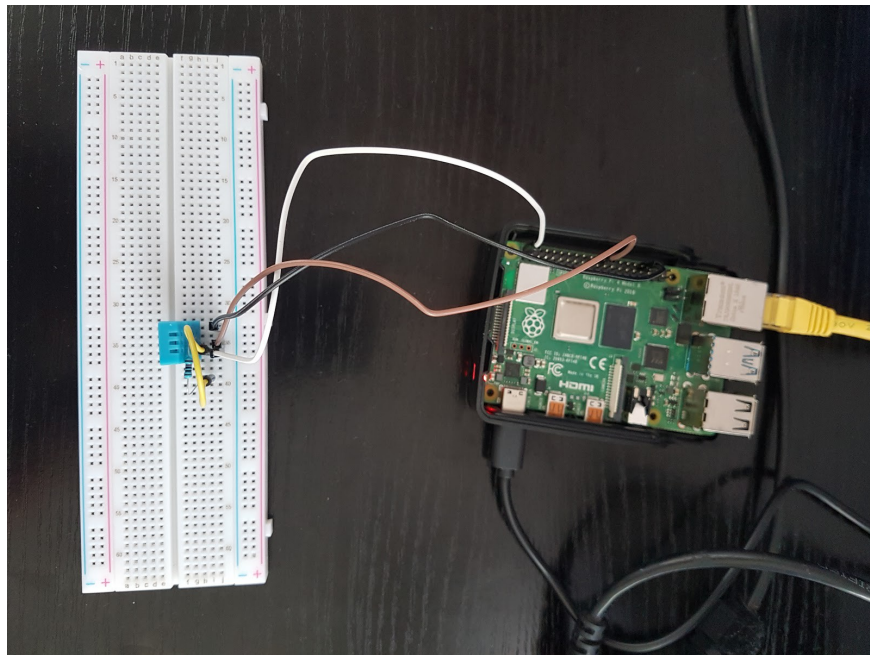


Corban Sergiu Ștefan



Server web cu afișare de temperatură și umiditate



Descriere și folosire:

Aplicație ce afișează temperatura și umiditatea camerei în browser. Pentru realizarea ei se folosește senzorul DHT11 împreună cu un Raspberry Pi.

Pentru a putea vedea datele prelucrate de aplicație de la senzorul DHT11, se pornește serverul python și apoi se deschide browserul la adresa și portul afișate în consola serverului. Datele se vor actualiza la un interval de o secundă fără a fi nevoie de a reîmprospăta pagina browser-ului.

Video demonstrativ:

<https://youtu.be/kAm9VVch0no>

Componente și aplicații:

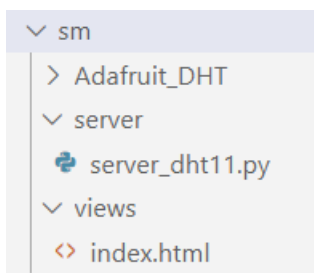
Componente software folosite:

- Raspberry Pi OS
- Python 3, HTML, JavaScript, jQuery
- BottlePy
- [Adafruit_DHT](#)

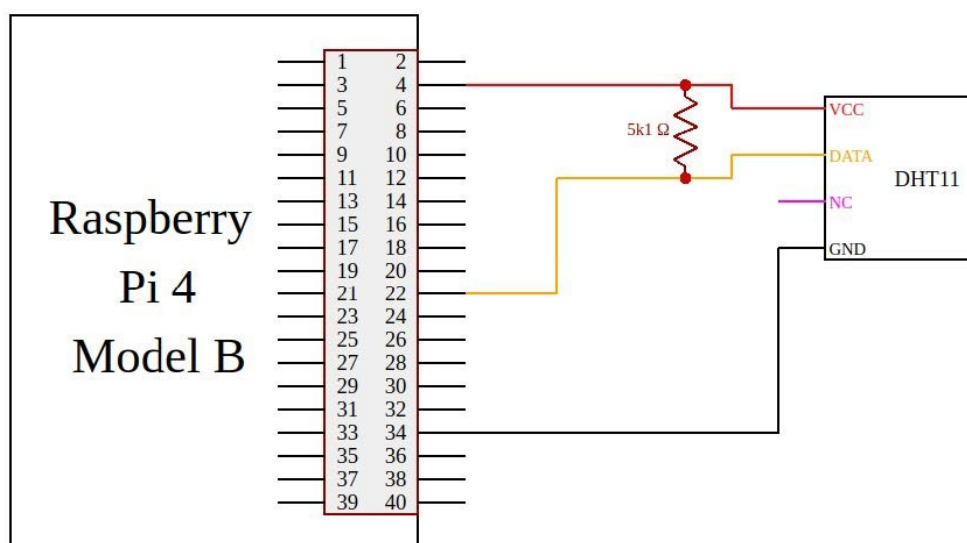
Componente hardware folosite:

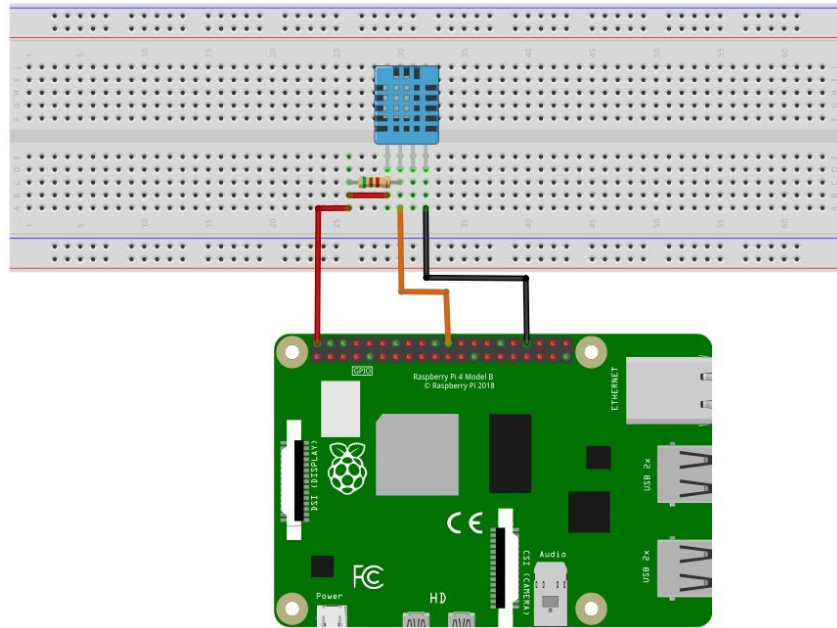
- Raspberry Pi 4 Model B;
- Breadboard 830p;
- Senzor de temperatura și umiditate DHT11;
- 1x rezistor de 10000 Ω ;
- 1x fire jumper male-to-male
- 3x fire jumper male-to-female

Structura fișierelor:



Schematici:





Cod:

```
server/server_dht11.py
```

```
"""
* Proiect Sisteme cu Microprocesoare
* Titlu: Server web cu afișare de temperatura și umiditate
* Autor: Corban Sergiu Ștefan
* Grupa: 1309B
* Conținut fișier: Logica serverului python și logica de prelucrare a
datelor a senzorului DHT11
"""
```

```
#importam librăriile necesare
from bottle import route, run, template, request
import RPi.GPIO as GPIO
import Adafruit_DHT

GPIO.setmode(GPIO.BCM)

#declarăm tipul de senzor folosit (în cazul nostru DHT11)
type = Adafruit_DHT.DHT11
```

```
#declarăm pinul folosit de senzor în GPIO
dht11 = 25

#setam senzorul ca INPUT
GPIO.setup(dht11, GPIO.IN)

#definim ruta catre pagina principala
@route('/')
def index():
    #la ruta '/' returnam pagina index.html
    #template-ul se afla in folderul 'views'
    return template('index.html')
@route('/refresh')
#definim o funcție pentru a trimite datele către client
def refresh():
    #facem citirea
    humidity, temperature = Adafruit_DHT.read(type, dht11)
    global buffer1
    global buffer2
    buffer1 = temperature
    buffer2 = humidity
    #trimitem noile valori dacă ele nu sunt nule
    #altfel vom trimite valorile precedente
    if humidity is not None and temperature is not None:
        #returnam valorile catre client
        return {'temperature':temperature, 'humidity':humidity}
    else:
        #returnam valorile catre client
        return {'temperature':buffer1, 'humidity':buffer2}

run(host = '0.0.0.0', port = '5000')
```

```
GPIO.cleanup()
```

views/index.html

```
<!-- Proiect Sisteme cu Microprocesoare -->
<!-- Titlu: Server web cu afișare de temperatura și umiditate -->
<!-- Autor: Corban Sergiu Ștefan -->
<!-- Grupa: 1309B -->
<!-- Conținut fișier: Codul html care va apărea în pagina și javascript-ul
care va prelua datele de la server -->

<html lang="en">
<head>
  <meta charset="utf-8">
  <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
  <title>Proiect SM</title>
</head>
<body>
  <h2>Proiect SM - Corban Sergiu Ștefan - 1309B</h2>
  <table style="font-size:20px">
    <tr>
      <td>
        <div>Temperatură: </div>
      </td>
      <td>
        <div id="Temperature"> </div>
      </td>
    </tr>
    <tr>
      <td>
        <div>Umiditate: </div>
```

```
</td>

<td>

  <div id="Humidity"> </div>

</td>

</tr>

</table>

<script>

$(document).ready(function () {

  setInterval(refreshFunction, 1000);

});

function refreshFunction() {

$.getJSON('/refresh', function (e) {

  $('#Temperature').html(e.temperature);

  $('#Humidity').html(e.humidity);

});

}

</script>

</body>

</html>
```