



2023

## 8. Temperatuurstation (Celsius, Fahrenheit)

Projectnummer: 2021-1-FR01-KA220-SCH-000031617



**Co-funded by  
the European Union**

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

SCRAPY Partnerschap

31/05/2023



## Inhoud

Experiment 8: Temperatuurstation (Celsius, Fahrenheit) .....	2
Korte Beschrijving .....	2
Uitgebreide Beschrijving.....	2
Doelstellingen .....	2
Te gebruiken materialen.....	3
Te volgen stappen.....	3
Schakelschema.....	3
Code .....	4

## Experiment 8: Temperatuurstation (Celsius, Fahrenheit)

### Korte Beschrijving

Creëer een temperatuurstation dat een rode LED aanzet als het warm is en een blauwe LED als het koud is, met Raspberry Pi Pico en DHT11 digitale temperatuur- en vochtigheidssensor.

### Uitgebreide Beschrijving

Het Raspberry Pi Pico-temperatuurstation is een kleinschalig project dat is ontworpen om de temperatuur te bewaken met behulp van de DHT11 digitale temperatuur- en vochtigheidssensor. Dit project maakt gebruik van het Raspberry Pi Pico-microcontrollerbord om temperatuurgegevens van de sensor te lezen en de verlichting van LED's te regelen op basis van de temperatuurmetingen.

Door de integratie van twee LED's, een rode LED voor het aangeven van warme temperaturen en een witte LED voor koude temperaturen, geeft dit temperatuurstation een visuele weergave van de huidige temperatuur. De LED's worden aangestuurd door de Raspberry Pi Pico, die de temperatuurmetingen van de DHT11-sensor verwerkt en bepaalt welke LED dienovereenkomstig moet oplichten.

Dit project heeft tot doel te demonstreren hoe je een DHT11-sensor kunt koppelen aan de Raspberry Pi Pico en deze kunt gebruiken om een eenvoudig temperatuurbewakingssysteem te creëren.

### Doelstellingen

Door deze activiteit zal de gebruiker experimenteren met het bouwen van een temperatuursysteem met behulp van het Raspberry Pi Pico-bord en de DHT11 digitale temperatuur- en vochtigheidssensor. De gebruiker verwerft kennis over:

- Het vermogen van een DHT11 digitale temperatuur- en vochtigheidssensor om temperatuur en vochtigheid te detecteren.
- De basisprincipes van programmeren in Python en hoe je code schrijft om het Raspberry Pi Pico-bord te besturen.
- De principes van circuitontwerp en hoe componenten met elkaar kunnen worden verbonden op een rapid prototyping-bord om een functioneel omkeerradarsysteem te creëren.

Door dit project te voltooien, zal de gebruiker elektronica, engineering en programmeren beter begrijpen. Ze zullen ook een praktisch en nuttig apparaat hebben dat ze kunnen gebruiken om de temperaturomstandigheden te controleren en te melden wanneer het warm of koud is.

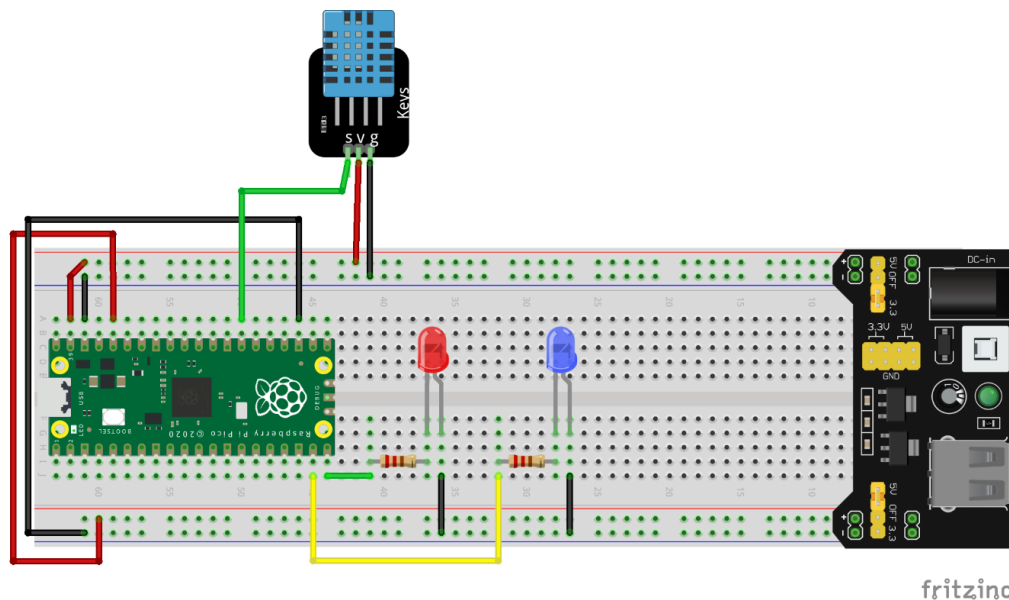
## Te gebruiken materialen

- 1 x Raspberry Pi Pico
- 1 x Pico-broodplankset
- 1 x broodplank van volledige grootte
- 1 x DHT11 digitale temperatuur- en vochtigheidssensor
- 2 x 220 Ohm weerstand
- 1 x blauwe LED
- 1 x rode LED
- Doorverbindingsdraden

## Te volgen stappen

1. Sluit de VCC-pin van de DHT11-sensor aan op de 3,3V-pin op de Raspberry Pi Pico.
2. Sluit de GND-pin van de DHT11-sensor aan op de GND-pin op de Raspberry Pi Pico.
3. Sluit de DATA-pin van de DHT11-sensor aan op een willekeurige GPIO-pin op de Raspberry Pi Pico. Laten we voor dit voorbeeld GPIO-pin 21 gebruiken.
4. Sluit de kathode (kortere poot) van de rode LED aan op GPIO-pin 15 op de Raspberry Pi Pico met behulp van een weerstand van 220 ohm.
5. Sluit de kathode (kortere poot) van de witte LED aan op GPIO-pin 14 op de Raspberry Pi Pico met behulp van een weerstand van 220 ohm.
6. Sluit de anode (langere poten) van beide LED's aan op de 3,3V-pin op de Raspberry Pi Pico.

## Schakelschema





## Code

```
from machine import Pin
from time import sleep
import dht

sensor = dht.DHT11(Pin(21))

led_red = Pin(14, Pin.OUT)
led_white = Pin(15, Pin.OUT)

while True:
    sensor.measure()
    temp = sensor.temperature()
    fahr = temp * (9/5) + 32
    hum = sensor.humidity()
    print("Room temperature: ", temp, "Celsius")
    print("Room temperature: ", fahr, "Fahrenheit")
    print("Room humidity:", hum, "%")

    if temp > 40:
        led_red.on()
        led_white.off()
    else:
        led_red.off()
        led_white.on()

    sleep(2)
```