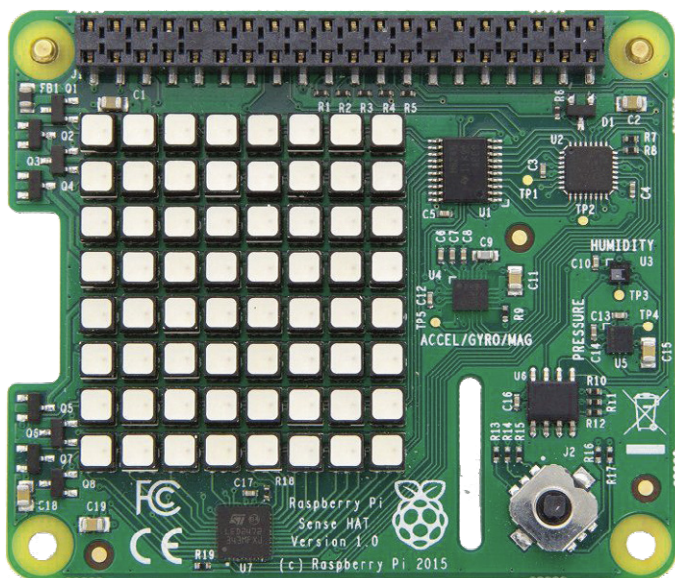


Eenvoudige aan/uit temperatuurregeling met Raspberry Pi HAT



Dogan Ibrahim (Groot-Brittannië)

De Raspberry Pi Sense HAT wordt normaal gesproken in de 40-polige connector van de Raspberry Pi geprikt. Om naast het Sense HAT-board ook nog externe componenten op de Raspberry Pi te kunnen aansluiten, moeten we de Sense HAT aan de Raspberry Pi koppelen met behulp van een stuk een flatcable of jumperdraden zodat andere pinnen van de Raspberry Pi toegankelijk zijn. Daarom moeten we weten welke pinnen van het Sense HAT-board door Raspberry Pi worden gebruikt en welke pinnen van Raspberry Pi voor andere doeleinden beschikbaar zijn. Het Sense HAT-board bestaat uit zeven hoofdcomponenten en een LED-matrix. De componenten op het board worden aangestuurd via een I²C-businterface. Dit betreft:

Component	I ² C-busadres	Functie
HTS221	0x5F	vochtigheidssensor
LPS254H	0x5C	druk-/temperatuursensor
LSM9DS1	0x1C, 0x6A	magnetometer + versnellingsmeter
SKRHABE010	–	joystick
LED2472G	0x46	LED-matrix controller
LED matrix	–	–
ATTINY88	–	Atmel microcontroller

Naast de I²C-besturingslijnen kan de ATTINY88-microcontroller op het bord worden geprogrammeerd via de SPI-stuurlijnen (MOSI, MISO, SCK, CE0) op het board.

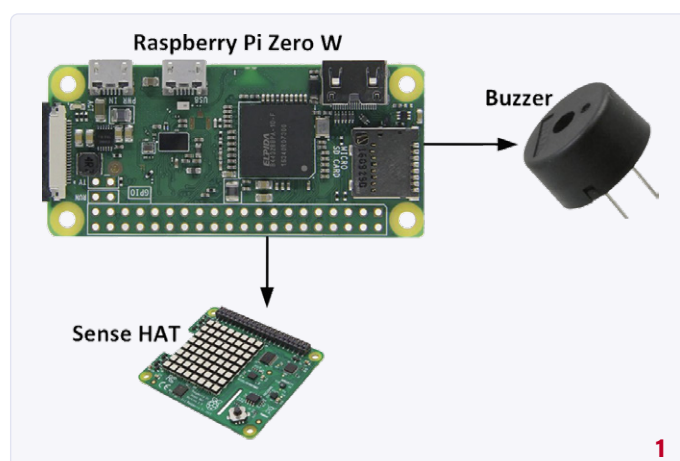
Het Sense HAT-board kan op de Raspberry Pi worden aangesloten met alleen de volgende 9 pinnen van de 40-polige connector:

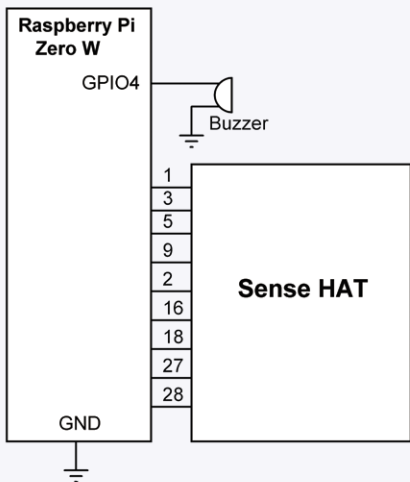
Sense HAT pin	Raspberry Pi Pin	Functie
3	3 (GPIO2)	SDA (I ² C)
5	5 (GPIO3)	SCL (I ² C)
1	1 (+3,3V)	voeding
9	9 (GND)	voedingsmassa
2	2 (+5V)	voeding
16	16 (GPIO23)	joystick
18	18 (GPIO24)	joystick
27	27 (ID_SD)	EEPROM
28	28 (ID_SC)	EEPROM

Zoals u nu ziet, blijft er een groot aantal Raspberry Pi-pinnen vrij voor gebruik met externe apparaten. Het 'project' waarmee we u aan de gang willen krijgen is een aan/uit temperatuurregeling waarbij de Sense HAT is aangesloten op de Raspberry Pi Zero W om de omgevingstemperatuur te meten. Bovendien is een kleine zoemer aangesloten op een van de Raspberry Pi-poorten. De ingestelde temperatuurwaarde is 'hard' in het programma gecodeerd. Als de omgevingstemperatuur lager is dan de ingestelde waarde, wordt de zoemer geactiveerd en geeft de LED-matrix de omgevingstemperatuur in rood weer. Als de omgevingstemperatuur daarentegen hoger is dan de ingestelde temperatuurwaarde, wordt de zoemer uitgeschakeld en wordt de omgevingstemperatuur in blauw weergegeven. De zoemer in dit project kan eenvoudig worden vervangen door een relais waarmee een verwarmingselement kan worden aangestuurd. De verwarming wordt dan ingeschakeld als de omgevingstemperatuur lager is dan de ingestelde waarde.

Figuur 1 toont het 'blokschema' van het project en **figuur 2** het 'uitgewerkte schema'. De zoemer is aangesloten op poortpin GPIO4 van de Raspberry Pi. Zowel de zoemer als het Sense HAT-bord zijn via jumperdraden met de Raspberry Pi verbonden.

De code die moet worden gebruikt voor het controleprogramma (*tempcont.py*) staat in **listing 1**, maar kan ook gratis worden gedownload van [1]. Aan het begin van het programma worden de benodigde





2

modules geïmporteerd. De zoemer is toegewezen aan nummer 4, corresponderend met GPIO4. De ingestelde temperatuurwaarde wordt opgeslagen in de variabele `SetTemperature` en heeft in dit voorbeeld de 'hardgecodeerde' waarde 24. De zoemer wordt aan het begin van het programma uitgeschakeld. De rest van het programma loopt in een eindeloze lus. Binnen deze lus wordt de omgevingstemperatuur uitgelezen; deze gemeten waarde wordt vergeleken met de ingestelde waarde. Als de omgevingstemperatuur lager is dan de ingestelde waarde, wordt de zoemer ingeschakeld en wordt de omgevingstemperatuur (stationair) rood weergegeven. Als de omgevings-temperatuur daarentegen de ingestelde waarde overschrijdt, wordt de zoemer uitgeschakeld en wordt de omgevingstemperatuur blauw weergegeven.

De zoemer die in dit project wordt gebruikt, kan eenvoudig worden vervangen door een relais op de contacten waarvan een verwarmingselement is aangesloten. Dit moet op een elektrisch veilige en verantwoorde manier gebeuren, zeker wanneer het de netspanning betreft! De kamertemperatuur wordt dan door het programma geregeld. 

200191-04

Bron: *Raspberry Pi Sense Hat*, door Dogan Ibrahim.

Verschijnt binnenkort.



IN DE STORE

- > **Raspberry Pi Sense HAT**
www.elektor.nl/raspberry-pi-sense-hat
- > **Engelstalig boek: Raspberry Pi Sense HAT.** Verschijnt binnenkort. Kijk op www.elektor.nl/books

Listing 1. tempcontr.py

```
#-----
#  ON-OFF TEMPERATURE CONTROLLER
#  -----
#
# This is an ON-OFF temperature control project. In this project
# a buzzer is connected to port pin GPIO4 of the Raspberry Pi in
# addition to the Sense HAT. The Sense HAT is connected using
# jumper wires. The buzzer is turned ON if the ambient temperature
# is below the setpoint temperature. At the same time, the ambient
# temperature is displayed in red colour. If on the other hand the
# ambient temperature is higher than the setpoint value then the
# buzzer is turned OFF and the display is in blue colour.
#
# The buzzer in this program can be replaced with a relay for
# example to control a heater
#
# Author: Dogan Ibrahim
# Date  : March 2020
# File  : tempcont.py
#-----

from display import Disp          # import Disp
from sense_hat import SenseHat    # import Sense HAT
sense=SenseHat()
import time                       # import time
import RPi.GPIO as GPIO          # import GPIO

GPIO.setwarnings(False)          # disable warnings
GPIO.setmode(GPIO.BCM)           # set GPIO mode

Buzzer = 4                        # Buzzer at GPIO4
SetTemperature = 24               # setpoint temp
red = (255, 0 ,0)                # red colour
blue = (0, 0, 255)               # blue colour

GPIO.setup(Buzzer, GPIO.OUT)      # Buzzer is output
GPIO.output(Buzzer, 0)            # Buzzer OFF

while True:
    T = int(sense.get_temperature_from_humidity())
                                     # get temperature

    if(T < SetTemperature):          # T < setpoint?
        Disp(T, red, 0)             # display in red
        GPIO.output(Buzzer, 1)      # Buzzer ON
    else:
        Disp(T, blue, 0)            # display in blue
        GPIO.output(Buzzer, 0)      # Buzzer OFF

    time.sleep(5)                  # wait 5 secs
```

WEBLINK

[1] tempcontrol.py download: www.elektormagazine.nl/200191-04