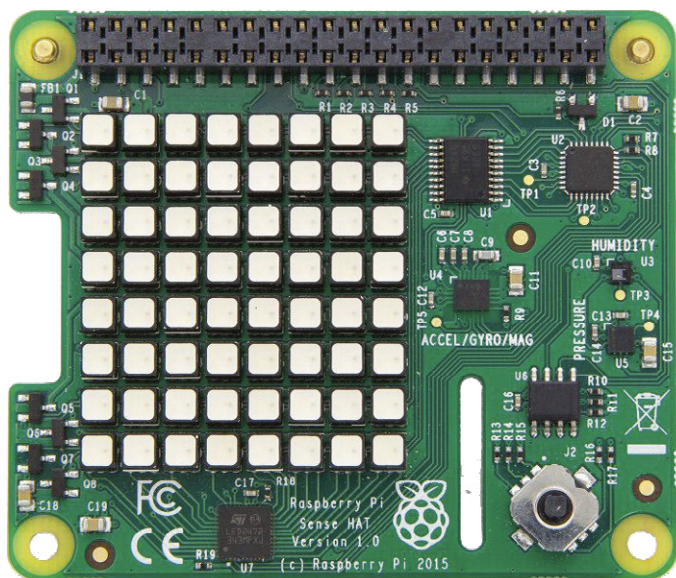


thermostat simple avec Raspberry Pi



Dogan Ibrahim (Royaume Uni)

Normalement l'extension *Sense* est branchée comme HAT sur le connecteur à 40 voies du Raspberry Pi. Pour pouvoir, en plus de cette carte *Sense*, interfacier d'autres composants externes avec le Raspberry Pi, nous devons connecter le HAT au Raspberry Pi en utilisant soit un câble en nappe, soit des fils de connexion, de sorte que les autres broches du Raspberry Pi restent accessibles. Il faut donc savoir quelles broches de la carte *Sense* sont utilisées par le RPi, et quelles broches du RPi sont libres.

Le HAT *Sense* compte sept composants principaux et une matrice de LED. Ces composants, commandés via l'interface I²C, sont :

composant	adresse I ² C	fonction
HTS221	0x5F	capteur d'humidité
LPS254H	0x5C	capteur de pression/température
LSM9DS1	0x1C, 0x6A	accéléromètre + magnétomètre
SKRHABE010	–	manche
LED2472G	0x46	commande de matrice de LED
matrice de LED	–	–
ATTINY88	–	µC Atmel

En plus des lignes de commande par I²C, le µC ATTINY88 de la carte peut être programmé via les lignes de commande du bus SPI (MOSI, MISO, SCK, CE0) présent sur cette carte.

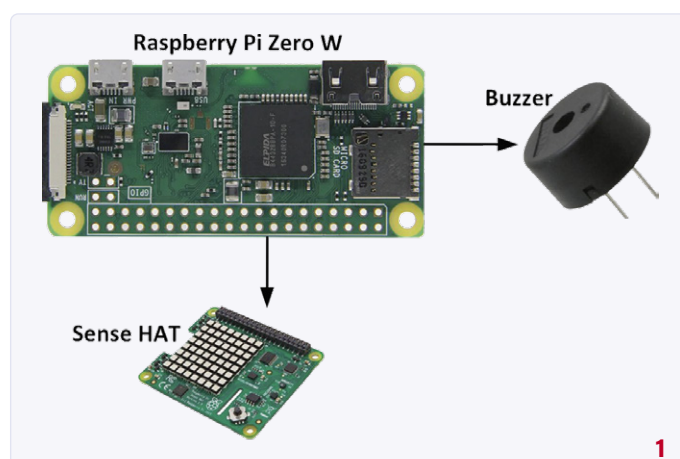
La carte *Sense* peut être connectée au RPi par les 9 broches suivantes du connecteur à 40 voies :

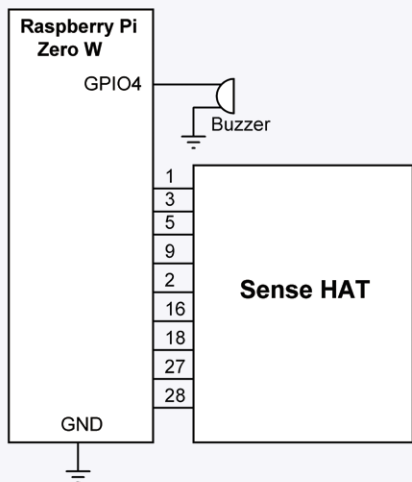
HAT	Raspberry Pi	fonction
3	3 (GPIO2)	SDA (I ² C)
5	5 (GPIO3)	SCL (I ² C)
1	1 (+3,3V)	puissance
9	9 (GND)	masse
2	2 (+5V)	alimentation
16	16 (GPIO23)	manche
18	18 (GPIO24)	manche
27	27 (ID_SD)	EEPROM
28	28 (ID_SC)	EEPROM

Ce qui laisse libres un grand nombre de broches RPi pour des appareils externes. Notre projet est un régulateur de température marche-arrêt avec le HAT *Sense* connecté au Raspberry Pi Zero W pour mesurer la température ambiante. Un buzzer est connecté à l'un des ports du RPi. La température de consigne est codée dans le programme. Si la température ambiante est inférieure à la consigne, le buzzer est activé et la matrice de LED affiche la température ambiante en rouge. Si la température ambiante dépasse la consigne, le buzzer est désactivé et la température ambiante affichée en bleu. Le buzzer peut facilement être remplacé par un relais, p. ex. pour commander le chauffage. Celui-ci se met alors en marche dès que la température ambiante est inférieure à la consigne.

La **figure 1** montre le diagramme du projet, la **figure 2** le schéma. Le buzzer est connecté à GPIO4 du Raspberry Pi. Le buzzer et la carte *Sense* sont tous deux connectés au RPi par des fils.

Le code pour le programme de commande *tempcont.py* est dans le **listage 1**, téléchargeable gratuitement [1]. Au début du programme, les modules requis sont importés. Le buzzer se voit attribuer le numéro 4 qui correspondra à GPIO4. La température de consigne





2

est codée dans la variable *SetTemperature* : 24 dans notre exemple. Le buzzer est désactivé au début du programme. Le reste du programme est une boucle au cours de laquelle la température ambiante est lue par le HAT et comparée à la température de consigne. Si la température ambiante est inférieure à la consigne, le buzzer est activé et la température ambiante est affichée en rouge et sans défilement. Quand la température ambiante dépasse la consigne, le buzzer est désactivé et la température ambiante affichée en bleu.

Le buzzer peut être remplacé par un relais aux contacts duquel est connecté un chauffage. Attention ! Cette opération, si elle porte sur la tension alternative du réseau, requiert des compétences particulières pour **éviter tout risque de choc électrique**. La température ambiante sera alors contrôlée par le programme.



200191-03

Source: Livre: *Raspberry Pi Sense Hat*, par Dogan Ibrahim.

Publication en préparation.

Listage 1: tempcontr.py

```
#-----
#           ON-OFF TEMPERATURE CONTROLLER
#           -----
#
# le buzzer peut être remplacé par un relais
# pour commander un chauffage
#
# Auteur: Dogan Ibrahim
# Date  : mars 2020
# File  : tempcont.py
#-----

from display import Disp          # importer Disp
from sense_hat import SenseHat    # importer Sense HAT
sense=SenseHat()
import time                       # importer l'heure
import RPi.GPIO as GPIO          # importer GPIO
GPIO.setwarnings(False)          # désactiver les notifications
GPIO.setmode(GPIO.BCM)          # activer le mode GPIO
Buzzer = 4                       # buzzer sur GPIO4
SetTemperature = 24              # température de consigne
red = (255, 0, 0)                # couleur rouge
blue = (0, 0, 255)              # couleur bleue
GPIO.setup(Buzzer, GPIO.OUT)     # buzzer est une sortie
GPIO.output(Buzzer, 0)           # buzzer éteint
while True:
    T = int(sense.get_temperature_from_humidity())
    # quelle est la température?

    if(T < SetTemperature):      # T < consigne ?
        Disp(T, red, 0)         # afficher en rouge
        GPIO.output(Buzzer, 1)  # activer le buzzer
    else:
        Disp(T, blue, 0)        # afficher en bleu
        GPIO.output(Buzzer, 0)  # désactiver le buzzer
    time.sleep(5)                # attendre 5 s
```



@ WWW.ELEKTOR.FR

> **Raspberry Pi Sense HAT**
www.elektor.fr/raspberry-pi-sense-hat

> **Livre : Raspberry Pi Sense HAT. Publication en préparation.**
www.elektor.com/books

LIEN

[1] **téléchargement de tempcontrol.py** : www.elektormagazine.fr/200191-03