

Enregistreur de données Sense HAT

Créer un appareil d'enregistrement de données
avec le matériel Sense HAT et Python



Etape 1 Ce que vous allez réaliser

Dans cette activité, vous allez apprendre à utiliser le matériel Sense HAT pour réaliser un appareil d'enregistrement de données capable de collecter tout un éventail d'informations sur son environnement immédiat.

Lorsque vous aurez créé votre enregistreur de données, vous pouvez l'utiliser pour conduire vos propres expériences et enregistrer des données. Cet enregistreur a subi de nombreux tests de résistance. Il a notamment été :

- jeté du haut d'un immeuble de quatre étages ;
- placé dans un réfrigérateur afin d'observer l'évolution de la température ;
- envoyé à la limite de l'espace avec un ballon à l'hélium.

Ce que vous allez apprendre

En créant un enregistreur de données Sense HAT avec votre Raspberry Pi, vous apprendrez :

- comment collecter les données de plusieurs capteur et les ajouter à une liste structurée ;
- à écrire et à ajouter des données dans un fichier de texte dans le cadre d'un programme Python ;
- à capturer une entrée du joystick Sense HAT et à y répondre ;
- à utiliser des threads pour permettre l'exécution simultanée de plusieurs parties d'un programme.

Cette ressource couvre des éléments des axes suivants du Raspberry Pi Digital Making Curriculum (<https://www.raspberrypi.org/curriculum/>):

- Mettre en œuvre l'abstraction et la décomposition pour résoudre des problèmes plus complexes_ (<https://www.raspberrypi.org/curriculum/programming/developer>)
- Traiter des données d'entrée pour surveiller l'environnement ou y réagir_ (<https://www.raspberrypi.org/curriculum/physical-computing/developer>)

Étape 2 Ce dont vous avez besoin

Matériel

- Un ordinateur Raspberry Pi
- Un Sense HAT

Si vous n'avez pas de Sense HAT, vous pouvez utiliser le programme d'émulation disponible sur votre logiciel

Raspberry Pi

Vous aurez besoin de la dernière version de Raspbian (<https://www.raspberrypi.org/downloads/>), qui inclut déjà les paquets logiciels suivants :

- Python 3
- Sense HAT pour Python 3

Si, pour une raison quelconque, vous devez installer un paquet manuellement, suivez

ces instructions : Saisissez cette commande dans le terminal pour installer le package

Sense HAT :

```
sudo apt-get install sense-hat
```

Étape 3 Obtenir les données du Sense HAT

Avec le Sense HAT, vous pouvez collecter les données des capteurs suivants :

- Capteur de température
- Capteur d'humidité
- Capteur de pression
- Capteur d'orientation
- Capteur d'accélération
- Gyroscope
- Capteur de champ magnétique

Tout d'abord, écrivez un script court pour obtenir les relevés des capteurs Sense HAT et les afficher à l'écran.

- Fixez votre Sense HAT sur votre Raspberry Pi.
- Une fois votre Sense HAT fixé, démarrez votre Pi.
- Ouvrez IDLE et créez un nouveau fichier dans lequel travailler.
- Pour commencer ce script, vous devez importer les modules Python pour commander votre Sense HAT et pour récupérer les données et l'heure du Raspberry Pi. Commencez en ajoutant ces trois lignes de code :

```
from sense_hat import SenseHat from
datetime import datetime

sense = SenseHat()
```

- Vous allez à présent créer une fonction qui récupèrera toutes les données de capteur et les retournera dans une liste. Commencez par définir votre fonction et créer une liste vide.

```
def get_sense_data():
    sense_data = []
```

- La section ci-dessous vous montre comment collecter les données des différents capteurs. Dans chaque cas, vous ajoutez les données à la liste `sense_data`. Terminez la fonction en revenant à la liste `sense_data`.
- Les seules autres données dont vous avez besoin sont la date et l'heure. Pour les trouver, vous pouvez utiliser le code suivant :

```
datetime.now()
```

- Obtenez les relevés de capteur restants ainsi que la date et l'heure en ajoutant les lignes suivantes à votre fonction :

```
```python
mag = sense.get_compass_raw()
sense_data.append(mag["x"])
sense_data.append(mag["y"])
sense_data.append(mag["z"])
```

```
acc = sense.get_accelerometer_raw()
sense_data.append(acc["x"]) sense_data.append(acc["y"])
sense_data.append(acc["z"])

gyro = sense.get_gyroscope_raw()
sense_data.append(gyro["x"])
sense_data.append(gyro["y"])
sense_data.append(gyro["z"])

sense_data.append(datetime.now())
` ``
```

- Pour terminer, vous pouvez consulter les données en imprimant la liste dans une boucle infinie. Ajoutez la fin suivante à votre script, puis enregistrez-le et exécutez le code.

```
while True:
 print(get_sense_data())
```

- Vous devez observer un flux continu de données dans l'interface de commande, chaque ligne ressemblant à quelque chose comme ça :

```
[26,7224178314209, 25,068750381469727, 53,77205276489258, 1014,18017578125, 3,8002126669234286, 306,1720338870328,
0.3019065275890227, 71.13333892822266, 59.19926834106445, 39.75812911987305, 0.9896639585494995, 0.12468399852514267,
-0.004147999919950962, -0.0013064055237919092, -0.0006561130285263062, -0.0011542239226400852, datetime.datetime(2015,
9, 23, 11, 53, 9, 267584)]
```

## Étape 4 Écrire les données dans un fichier

---

Le programme que vous avez produit jusqu'à présent peut vérifier en continu les capteurs Sense HAT et écrire ces données à l'écran. Cependant, à moins d'être un lecteur ultra rapide, ce n'est pas très utile.

Il serait beaucoup plus utile d'écrire ces données dans un fichier CSV (valeurs séparées par des virgules), que vous pourrez analyser une fois le programme d'enregistrement terminé. Pour créer ce fichier, vous devez procéder comme suit :

- Spécifiez le nom de ce fichier.
- Ajoutez une ligne d'en-tête au début du fichier.
- Écrivez régulièrement un lot de données dans le fichier.

Commencez par apprendre comment écrire les données de la liste dans un fichier CSV en Python, vous vous occuperez de l'en-tête plus tard.

- À présent, vous pouvez modifier votre code pour écrire en continu les données de votre fonction `get_sense_data()` dans un fichier CSV. Voici une manière de procéder :
  - Avant que votre boucle ne démarre, ouvrez un fichier csv et créez votre scripteur.
  - Dans votre boucle, écrivez les données retournées de la fonction vers le fichier.

À présent, écrivez simplement ces données dans le fichier.

```
with open('data.csv', 'w', newline='') as f:
 data_writer = writer(f)

 while True:
 data = get_sense_data()
 data_writer.writerow(data)
```

## Étape 5 Ajouter un en-tête au fichier CSV

---

Vous collectez de nombreux types de données différents dans le fichier CSV. Pour savoir quel type de données chaque colonne contient, il serait utile d'ajouter une ligne d'en-tête au fichier CSV.

Pour cela, vous pouvez simplement écrire une ligne supplémentaire dans le fichier CSV avant de démarrer la boucle infinie.

- Ajoutez cette ligne après avoir créé votre objet `writer` et avant que la boucle `while True` ne démarre :

```
data_writer.writerow(['temp', 'pres', 'hum',
 'yaw', 'pitch', 'roll',
 'mag_x', 'mag_y', 'mag_z',
 'acc_x', 'acc_y', 'acc_z',
 'gyro_x', 'gyro_y', 'gyro_z', 'datetime'])
```

- Assurez-vous que les en-têtes sont dans le même ordre que les données produites par votre fonction `get_sense_data()`

## Étape 6 Enregistrer à intervalles spécifiques

---

Votre script enregistre alors les données aussi rapidement que possible. Cela s'avère très utile pour certaines expériences, mais vous pouvez choisir de n'enregistrer les données qu'une fois par seconde, voire encore moins fréquemment.

Dans ce genre de situation, on utilise normalement une fonction `sleep()` pour mettre le script en pause. Toutefois, cela peut se solder par des relevés inexacts de certains des capteurs d'orientation du Sense HAT, qui doivent être consultés régulièrement.

Pour contourner le problème, vous pouvez utiliser `timeDelta` pour vérifier l'écart de temps entre deux relevés.

Pour utiliser cette approche de collecte de données, procédez comme suit :

- Au début de votre script, créer une variable `timestamp` définie sur `datetime.now()`
- Décidez de la durée de l'intervalle entre les enregistrements des données (en secondes) et créez une variable appelée `delay` pour stocker ce nombre.
- Dans votre boucle infinie, si l'écart entre la variable `timestamp` et la durée du relevé retourné par votre fonction `get_sense_data()` est supérieur à la variable `delay`, écrivez les données et réinitialisez `timestamp`

- Essayez d'ajouter un délai d'une seconde à votre intervalle d'écriture des données et utilisez les conseils ci-dessous si vous êtes coincé.

Si cette variable `dt` est supérieure à la variable `delay` que vous avez spécifiée, les données peuvent être écrites et la variable `timestamp` peut être réinitialisée.

```
while True:
 data = get_sense_data() dt
 = data[-1] - timestamp if
 dt.seconds > delay:
 data_writer.writerow(data) timestamp
 = datetime.now()
```

- Essayez avec différentes valeurs pour la variable `delay` chaque fois que vous exécutez votre script.
- Pouvez-vous définir un délai d'une microseconde ?

## Étape 7 Lancer votre enregistreur de données au démarrage

---

Cette étape est totalement facultative, mais vous pouvez souhaiter exécuter votre script dès que le Raspberry Pi démarre. Pour cela, vous pouvez utiliser un job cron. Consultez la section ci-dessous pour découvrir comment modifier votre crontab afin d'exécuter des scripts au démarrage.



## Étape 8 Défi : sélectionner les données à enregistrer

---

Vous ne souhaitez peut-être pas enregistrer toutes les données des capteurs. Vous pouvez alors commenter les lignes dont vous n'avez pas besoin dans votre fonction `get_sense_data()`.

Une autre solution consiste à mettre en œuvre une sélection conditionnelle.

- Pouvez-vous configurer votre script afin de ne mettre en œuvre dans votre fonction `get_sense_data` que les capteurs que vous souhaitez utiliser, ceux-ci étant alors les seuls enregistrés ? N'oubliez pas d'ajouter une méthode pour modifier aussi la ligne d'en-tête de votre fichier CSV.

---

Publié par Raspberry Pi Foundation (<https://www.raspberrypi.org>) sous licence Creative Commons (<https://creativecommons.org/licenses/by-sa/4.0/>).

Consulter les projets et les licences sur GitHub (<https://github.com/RaspberryPiLearning/sense-hat-data-logger>)