



Démarrer avec le Sense HAT

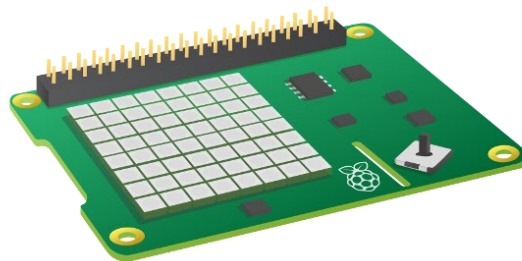
Découvrez la platine du capteur Sense HAT et sa bibliothèque Python



Étape 1 Introduction

Ce que vous allez réaliser

Le Sense HAT, qui est un élément fondamental de la mission [Astro Pi \(https://esero.fr/projets/astro-pi/\)](https://esero.fr/projets/astro-pi/), permet à votre Raspberry Pi de sentir le monde autour de lui.



Durant ce projet, vous allez apprendre comment commander la matrice à LED du Sense HAT et collecter les données du capteur, et vous associerez ces notions dans de petits projets.

Ce que vous allez apprendre

En suivant cette ressource avec votre Raspberry Pi et votre Sense HAT, vous apprendrez comment :

- communiquer avec le Sense HAT à l'aide de Python ;
- accéder aux sorties du Sense HAT ;
- programmer les entrées du Sense HAT ;
- utiliser la bibliothèque Sense HAT pour afficher des messages et des images ;
- utiliser des variables pour stocker les données du capteur ;
- utiliser des boucles pour répéter les comportements.

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

- Utilisation de la programmation de base pour créer des programmes simples (<https://www.raspberrypi.org/curriculum/programming/creator>)
- Traitement 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

- Raspberry Pi
- Sense HAT

Logiciels

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

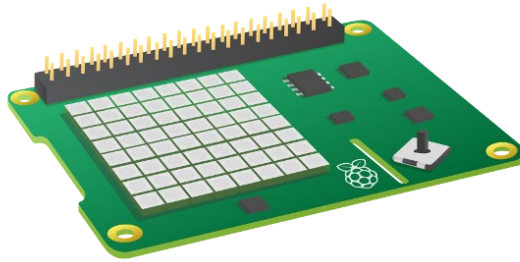
- Python 3
- Sense HAT pour Python 3

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 Qu'est-ce qu'un Sense HAT ?

Le Sense HAT est une carte d'extension pour le Raspberry Pi spécialement conçue pour le concours [Astro Pi](https://esero.fr/projets/astro-pi/) (<https://esero.fr/projets/astro-pi/>). Cette carte vous permet de réaliser des mesures de la température, de l'humidité, de la pression et de l'orientation et de fournir des informations avec sa matrice à LED intégrée.



Si vous n'avez pas accès à un véritable Sense HAT, vous pouvez utiliser un émulateur.

Étape 4 Afficher du texte

- Affichez le texte « Astro Pi est super » sur l'afficheur à LED de votre Sense HAT.

Nous pouvons à présent modifier l'affichage du message en ajoutant des paramètres `show_message` supplémentaires à la commande.

`scroll_speed` : affecte la vitesse de défilement du texte. La valeur par défaut est `0.1`. Plus la valeur est élevée, plus la vitesse est lente.

`text_colour` : modifie la couleur du texte et varie en fonction de trois valeurs spécifiant le rouge, le vert et le bleu. Celles-ci sont également appelées valeurs RGB.

Consultez les sections ci-dessous pour en savoir plus sur les valeurs RGB.

`back_colour` : modifie la couleur du fond et fonctionne de la même manière que `text_colour`.

- Ajoutez une ligne de code avant votre message pour définir une variable appelée `blue` avec la valeur `(0, 0, 255)`.
- Ajoutez une autre ligne de code pour définir une variable appelée `yellow` avec la valeur `(255, 255, 0)`.
- Ajoutez les paramètres à la commande `show_message` pour afficher le texte en jaune sur fond bleu.

Étape 5 Afficher un seul caractère

- Ajoutez un autre paramètre appelé `scroll_speed` à la commande `show_message` et définissez une vitesse de `0.05` pour accélérer la vitesse de défilement de votre message.
- Définissez une boucle `while` pour votre message défilant afin qu'il se répète.
- Affichez la lettre « A » sur l'afficheur à LED de votre Sense HAT.

Nous pouvons modifier l'affichage de la lettre deux des paramètres utilisés pour la commande `show_message` : `text_colour` et `back_colour`. Les lettres ne défilent pas, donc le `scroll_speed` est absent.

- Affichez la lettre « J » en rouge sur fond blanc.
- Utilisez la fonction `sleep` pour afficher les lettres de votre nom l'une après l'autre, chacune dans une couleur différente avec une pause d'une seconde entre chacune.
- Générez une couleur de manière aléatoire `randint` en utilisant valeurs RGB pour choisir un nombre entre 0 et 255 pour chacune des 3 composantes couleur
- Utilisez `sense.clear()` à la fin de votre code pour effacer la matrice à LED.

Étape 6 Paramétrer l'orientation

Vous pouvez remplir la totalité de la matrice à LED avec une couleur unique en utilisant la méthode `clear` avec la couleur choisie.

Paramétrer des pixels uniques

La matrice à LED peut afficher bien plus que du texte ! Il est possible de contrôler chacune des LED individuellement pour créer une image.

- Allumez les pixels (LED) aux quatre coins de la matrice dans la couleur de votre choix.

Pouvez-vous deviner ce que ce code génère ?

- Modifiez le code pour produire une image différente.

Paramétrer des pixels multiples

Paramétrer des pixels individuellement fonctionne parfaitement, mais cela se complique quand vous voulez définir des pixels multiples. Pour modifier tous les pixels simultanément, il existe la commande `set_pixels`

- Utilisez la méthode `set_pixels` pour afficher une image sur la matrice à LED.

Jusqu'à présent, tous nos textes et images apparaissent dans le même sens, avec le port HDMI en bas. Cependant, ce n'est pas toujours la véritable orientation du Sense HAT (en particulier dans l'espace), et vous pouvez être amenés à modifier l'orientation de la matrice à LED.

- Modifiez l'orientation de l'image de l'étape précédente en ajoutant du code pour faire pivoter l'affichage sous le code que vous avez écrit pour connecter le Sense HAT :

Vous pouvez créer une animation simple en retournant une image de manière répétée :

Étape 7 Détection de l'environnement

Le Sense HAT est doté d'un ensemble de capteurs environnementaux pour détecter les conditions environnantes. Il peut mesurer la pression, la température et l'humidité.

- Créez un texte défilant qui informe les personnes présentes des relevés actuels de pression, de température et d'humidité. Vous pouvez utiliser le code d'affichage d'un texte défilant que vous avez écrit à l'étape « Affichage d'un texte » pour vous aider.

Selon la documentation en ligne (<http://wsn.spaceflight.esa.int/docs/Factsheets/30%20ECLSS%20LR.pdf>), la Station spatiale internationale maintient ces conditions aux niveaux suivants :

Température : 18,3-26,7 degrés Celsius

Pression : 979-1 027 millibars

Humidité : environ 60 %

- Définissez des variables pour les couleurs verte (0, 255, 0) et rouge (255, 0, 0).
- Utilisez une instruction IF dans votre code pour vérifier que la température est comprise entre 18,3 et 26,7 degrés Celsius.
- Si la température est comprise dans la plage normale, affichez le message déroulant sur fond vert. Dans le cas contraire, affichez-le sur fond rouge.
- Ajoutez d'autres instructions IF pour tester aussi les conditions de pression et d'humidité.

Étape 8 Détection de mouvements

Le Sense HAT est doté d'une puce UMI (unité de mesure inertielle) qui comprend un ensemble de capteurs détectant le mouvement :

- un gyroscope (pour détecter dans quel sens se trouve la carte)
- un accéléromètre (pour détecter les mouvements)
- un magnétomètre (pour détecter les champs magnétiques)

Écrivez un programme pour détecter le tangage, le roulis et le lacet. Exécutez le programme et déplacez le Sense HAT. Observez comment les valeurs évoluent avec le mouvement du Sense HAT.

Remarque : lorsque vous utilisez des capteurs de mouvement, il est important de réaliser des relevés fréquents. Si vos relevés sont trop éloignés, par exemple si vous incluez dans votre boucle, vous obtiendrez des résultats bizarres. Cela est dû au fait que le code a besoin de beaucoup de mesures pour combiner correctement les données issues du gyroscope, de l'accéléromètre et du magnétomètre. `time.sleep(0.5)`

Comment distinguer le haut du bas ?

La méthode `sense.get_accelerometer_raw()` vous indique la quantité de force g à l'œuvre sur chaque axe (x, y, z). Si un axe affiche $\pm 1g$, alors vous savez que cet axe pointe vers le bas.

Dans cet exemple, la quantité d'accélération gravitationnelle de chaque axe est extraite puis arrondie au nombre entier le plus proche :

- Faites pivoter le Sense HAT. Vous devez voir les valeurs de x et y s'invertir `-1` et `1`. Si vous posez le Pi à plat entre ou que vous le retournez, la valeur de l'axe z passe de `1` à `-1`.

Utilisez ces informations pour définir l'orientation de la matrice à LED.

- En partant du code ci-dessus, ajoutez du code avant la boucle while pour afficher la lettre « J » sur la matrice à LED. Utilisez la méthode `show_letter` que vous avez déjà vue.
- Après le code affichant les valeurs de force g pour les axes x, y et z, ajoutez une instruction IF pour vérifier dans quelle direction pointe le Sense HAT. Mettez à jour l'orientation de l'affichage en utilisant `set_rotation` que vous la méthode avez vue précédemment. Voici du pseudocode pour vous aider à démarrer :

```
If the x axis has -1 G, rotate 90 degrees
Else if the y axis has 1G, rotate 0 degrees
Else if the y axis has -1G, rotate 180 degrees
Else rotate 270 degrees
```

Secouez la carte

Si la carte pivote uniquement, elle connaît simplement une accélération de 1g dans n'importe quelle direction ; si elle était secouée, le capteur enregistrerait plus de 1g. Il serait alors possible de détecter ce mouvement rapide et d'y répondre.

Pour ce programme, nous allons introduire la fonction `abs()`, qui n'est pas propre à la bibliothèque Sense HAT mais au langage Python standard. `abs()` fournit une valeur absolue et ignore si cette valeur est positive ou négative ; par exemple, `abs(1)` et `abs(-1)` retournent `1`. Cette fonction est utile car nous ne nous

intéressons pas
au sens dans lequel le capteur est secoué, mais simplement au fait qu'il est secoué.

```
from sense_hat import SenseHat sense

s = SenseHat()

red = (255, 0, 0)

while True:
    acceleration = sense.get_accelerometer_raw()
    x = acceleration['x']
    y = acceleration['y']
    z = acceleration['z']

    x = abs(x)
    y = abs(y)
    z = abs(z)

    if x > 1 or y > 1 or z > 1:
        sense.show_letter("!", red)
    else:
        sense.clear()
```

L'émulation étant un peu délicate, vous devriez essayer ce programme si possible avec un véritable Sense HAT. Si vous trouvez que le programme est trop sensible (c'est-à-dire s'il estime que le Sense HAT est secoué en permanence), essayez de modifier la valeur 1 pour une valeur supérieure afin de relever le seuil de ce qui est considéré comme une « secousse ».

Étape 9 Utilisation du joystick

Vous pouvez détecter à quel moment le joystick du Sense HAT est actionné, maintenu et relâché dans cinq directions différentes : haut, bas, gauche, droite et milieu.

- Selon la direction dans laquelle le joystick a été actionné, affichez l'une des lettres H, B, G, D ou M sur la matrice à LED.

Vous pouvez aussi appeler une fonction lorsque le joystick du Sense HAT est actionné dans une direction donnée.

- Créez des fonctions pour remplir la matrice à LED avec quatre couleurs différentes. Ajoutez des déclencheurs pour appeler une fonction pour chaque direction possible dans laquelle le joystick peut être actionné.

Étape 10 Défi : tout rassembler

Maintenant que vous avez exploré la plupart des fonctionnalités du Sense HAT, vous pouvez les combiner pour créer un projet. Voici un exemple de jeu de réaction qui peut être utilisé par les astronautes pour tester leurs réflexes :

Pivotez la carte pour que la flèche pointe vers le haut. Si vous y parvenez dans le délai imparti, la flèche devient verte et votre score augmente ; dans le cas contraire, la flèche devient rouge et le jeu se termine. Le jeu continue d'afficher des flèches dans de nouvelles directions jusqu'à ce que vous perdiez, et chaque tour est plus rapide.

Cette idée associe :

- Affichage de messages et d'images sur la matrice à LED
- Paramétrage et détection de l'orientation
- Utilisation de variables, randomisation, itération et sélection

Ce jeu étant beaucoup plus complexe que les programmes précédents dans cette ressource, il est utile de planifier les étapes impliquées en pseudocode :

Import les bibliothèques requises (`sense_hat`, `time`, `random`) Créer un objet sensehat

Déterminer les variables pour les couleurs requises (blanc, vert, rouge, vide) Créer trois flèches différentes (blanche, verte, rouge)

Définir une variable `pause` sur 3 (durée initiale entre les tours)

Définir les variables `score`, `angle` sur 0

Définir une variable `play` sur `True` (elle servira à arrêter le jeu plus tard)

```
while play == random
```

Choisir un nouvel angle

Afficher la flèche blanche

`Sleep` pour la durée de la pause

If l'orientation et la flèche
coïncident...

Ajouter un point et afficher la flèche en vert

Sinon, définir le jeu sur `False` et afficher la flèche en rouge

Réduire légèrement la durée de la pause

Pause avant la flèche suivante

Lorsque le joueur quitte la boucle, afficher un message avec le score

Étape 11 Défi : plus d'idées

Maintenant que vous avez exploré les bases du Sense HAT, vous pouvez étudier d'autres choses à faire avec :

- Racontez une blague sur l'écran à LED.
- Si votre Sense HAT est connecté à Internet, vous pouvez utiliser une bibliothèque d'API Twitter pour afficher des tweets entrants.
- Créez vos propres images à afficher sur la matrice à LED.
- Pouvez-vous afficher alternativement des images pour créer une animation ? Regardez cette vidéo de Geek Gurl Diaries (<https://www.youtube.com/watch?v=b84EywkQ3HI>) pour trouver l'inspiration.
- Créez un dé électronique comme celui-ci (<https://www.youtube.com/watch?v=UfP-R6ArMSk>). Secouez le Pi pour faire rouler le dé.
- Créez un simple thermomètre graphique qui affiche différentes couleurs ou différents motifs selon la température.
- Écrivez un programme qui affiche une flèche (ou un autre symbole) à l'écran ; ce symbole peut servir à indiquer le bas. Ainsi, les astronautes en situation de faible gravité sauront toujours où se trouve la Terre.
- Utilisez l'accéléromètre pour détecter les petits mouvements — dans le cadre d'un jeu, d'un système d'alarme ou même d'un détecteur de séismes.
- Utilisez le capteur d'humidité pour détecter la respiration et afficher une couleur en fonction de l'humidité.

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 le site de GitHub (<https://github.com/RaspberryPiLearning/getting-started-with-the-sense-hat>)