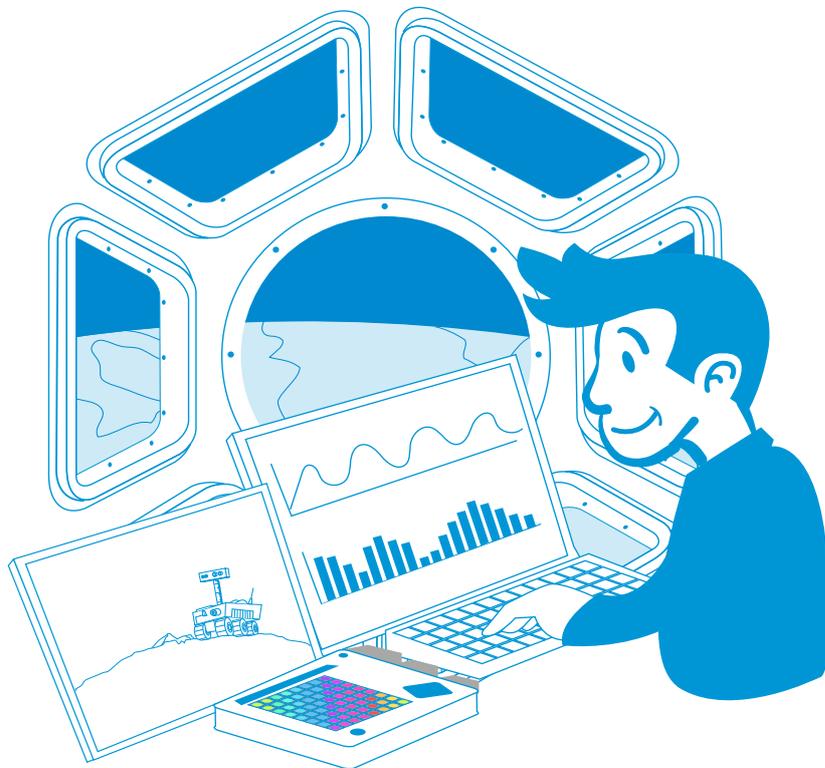
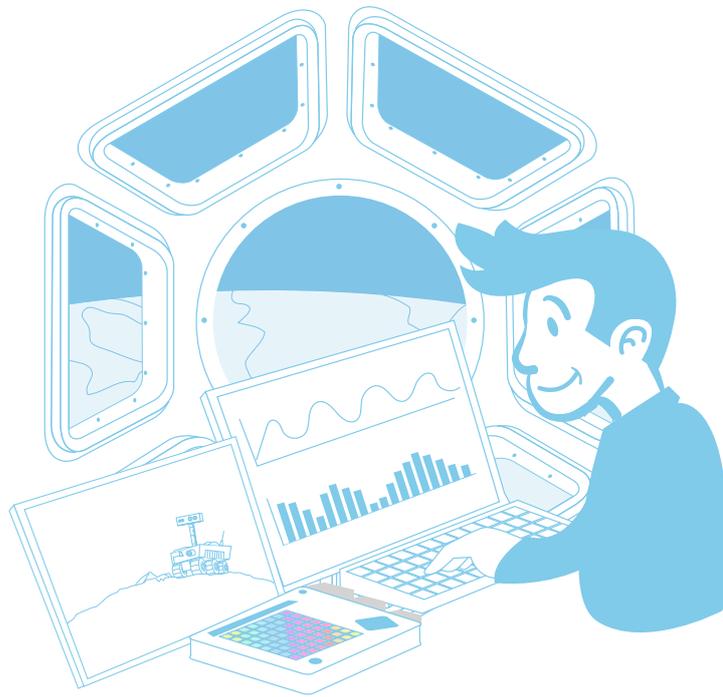


teach with space

→ COMMENT COLLECTER LES DONNÉES DE L'ASTRO PI

Utilisation des capteurs du Sense HAT pour collecter des données environnementales





Challenge Astro Pi européen

page 3

Activité 1 - Rester frais sur l'ISS

page 4

Activité 2 – La régulation de l'humidité à l'intérieur de l'ISS

page 8

Activité 2.3 Affichage de données

page 15

→ COMMENT COLLECTER LES DONNÉES DE L'ASTRO PI

Utilisation des capteurs du Sense HAT pour collecter des données environnementales

Pendant le Challenge Astro Pi européen, l'Astro Pi Ed à bord de la Station Spatiale Internationale (ISS) collectera tout un éventail de données en utilisant ses capteurs.

Dans cet ensemble d'activités, vous allez découvrir les conditions de vie à bord de l'ISS et les comparerez à celles sur Terre en utilisant les capteurs du Sense HAT pour comprendre l'espace qui vous entoure.



Figure A1

[↑ Astro Pi logo](#)

Équipement

- Kit Astro Pi
- Moniteur
- Clavier USB
- Souris USB

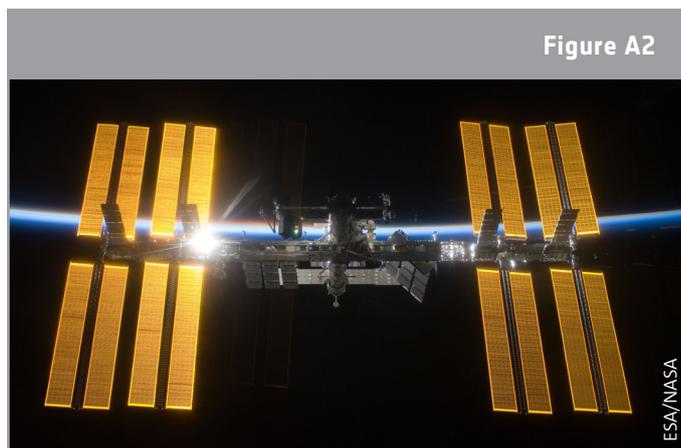
→ ACTIVITÉ 1 - RESTER FRAIS SUR L'ISS

Il est important pour les astronautes que la température dans l'ISS reste aux alentours de 24°C, mais ce n'est pas chose aisée. Du côté ensoleillé de l'ISS, les températures peuvent grimper à 121°C, tandis que sur la face à l'ombre elles peuvent dégringoler jusqu'à -157°C ! Vous allez utiliser les capteurs du Sense HAT pour mesurer la température dans votre salle de classe et la comparer à celle à bord du module Columbus de l'ISS.

Exercice

1. Pourquoi pensez-vous qu'il est important que la température à bord de l'ISS reste aussi proche que possible de 24°C ?

2. Ouvrez Python 3 en cliquant sur le logo Raspberry dans le haut de l'écran. Le menu s'ouvre. Sélectionnez Programming > Python 3 (Programmation > Python 3). Une fenêtre de shell de Python s'affiche. Sélectionnez File > New File (Fichier > Nouveau fichier) et saisissez le code ci-dessus dans la nouvelle fenêtre.



↑ La Station spatiale internationale (ESA/NASA)

Remarque : les phrases qui commencent par un # sont de simples commentaires. Le programme ne les exécutera pas et il est inutile que vous les copiez.

```
File Edit Format Run Options Windows Help
#to allow the program to use the Sense HAT hardware
from sense_hat import SenseHat

#to create a sense object which represents the Sense HAT
sense = SenseHat()

#to collect temperature and store it as temp
temp = sense.get_temperature()

#to round the value of the temperature with two decimal points
temp = round(temp, 2)

print (temp)
```

3. Sélectionnez File > Save (Fichier > Enregistrer) et choisissez un nom de fichier pour votre programme. Sélectionnez Run > Run module (Exécuter > Exécuter le module). Prenez note de la température enregistrée.

4. La température dans le module Columbus indiquée dans le Tableau 1 ci-dessous a été relevée par un des Astro Pi à bord de l'ISS.

Tableau A1			
Données expérimentales du module Columbus		Données expérimentales de votre salle de classe	
Température (en °C)	Date et heure	Température (en °C)	Date et heure
27,53	16/2/16 10:45		
27,52	16/2/16 10:45		
27,54	16/2/16 10:45		
27,55	16/2/16 10:45		
27,53	16/2/16 10:45		
27,55	16/2/16 10:45		
27,54	16/2/16 10:46		
27,54	16/2/16 10:46		
27,53	16/2/16 10:46		
27,52	16/2/16 10:46		
27,53	16/2/16 10:46		
27,53	16/2/16 10:46		

↑ Une comparaison entre la température de l'ISS et la température de la classe.

- a) Ajoutez une boucle « while True » (tant que la condition est vraie) à votre code. Cela vous permettra de collecter en continu les données du capteur. Votre code devrait ressembler maintenant à celui du cadre ci-dessous. sensor.

```
File Edit Format Run Options Windows Help
|from sense_hat import SenseHat
#to allow the program to use the time module
import time
sense = SenseHat()
#to repeat the code until a condition is met
while True:
    temp = sense.get_temperature()
    temp = round(temp, 2)
    #
    time.sleep(10)
#the strftime function is used to print the date and time
print (time.strftime('%x %X'),temp)
```

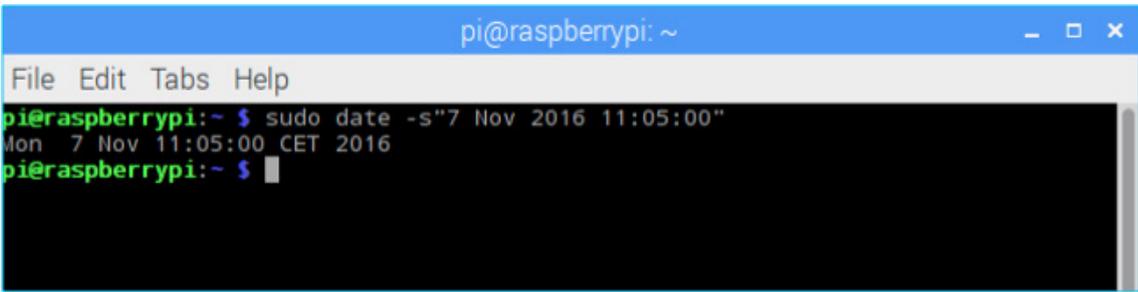
b) Écrivez un commentaire dans votre code pour expliquer la commande `time.sleep(10)` (temps. sommeil(10)).

c) Le Raspberry Pi n'a pas d'horloge en temps réel. Pour pouvoir voir l'heure exacte, vous devez manipuler directement le système. Ouvrez une fenêtre de terminal en cliquant sur l'icône du terminal dans le haut de l'écran. Vous devriez voir une nouvelle fenêtre avec l'invite suivante :

```
pi@raspberrypi ~ $
```

Saisissez la commande suivante (reportez-vous à l'exemple de la capture d'écran ci-dessous) et appuyez sur Entrée :

```
sudo date -s "Jour Mois Année hh:mm:ss"
```



```
pi@raspberrypi:~ $ sudo date -s"7 Nov 2016 11:05:00"
Mon 7 Nov 11:05:00 CET 2016
pi@raspberrypi:~ $
```

Remarque : il est possible que vous deviez fermer la fenêtre de terminal pour voir l'heure mise à jour.

d) Revenez à votre fenêtre d'éditeur et exécutez votre code. Complétez le Tableau 1 avec les données imprimées. Pour arrêter d'exécuter le programme, appuyez sur Ctrl + C.

e) En utilisant les données du Tableau 1, calculez la température moyenne dans le module Columbus et la température moyenne dans votre salle de classe. Sont-elles proches de 24°C ? Qu'est-ce qui, à votre avis, explique cette conclusion ?

Extension

L'espace est le milieu des extrêmes. Comment pensez-vous qu'il soit possible de réguler la température à l'intérieur de l'ISS ? Planifiez une recherche pour déterminer quels sont les matériaux que nous pourrions utiliser pour maintenir une température adéquate pour les astronautes. Vous pouvez suivre le modèle ci-dessous.

Planifiez votre modèle d'enquête

1. Formulez une question pour votre recherche. Émettez ensuite une hypothèse à vérifier.

Question de la recherche :

Hypothèse :

2. Concevez un plan pour tester votre hypothèse en utilisant votre Astro Pi. De quels autres matériaux avez-vous besoin ?

3. Collectez et analysez vos données.

4. Tirez-en vos conclusions et essayez de répondre à la question de la recherche.

→ ACTIVITÉ 2 – LA RÉGULATION DE L'HUMIDITÉ À L'INTÉRIEUR DE L'ISS

Bien que vous ne puissiez pas la voir, il y a de l'eau dans l'air qui nous entoure. L'humidité est la mesure de la quantité de vapeur d'eau présente dans l'air. Elle est normalement exprimée en pourcentage d'humidité relative. 100 % d'humidité relative, à une température donnée, signifie que l'air contient le maximum de vapeur d'eau possible.

Dans cette activité, vous allez utiliser l'Astro Pi pour simuler le système de contrôle d'humidité utilisé sur l'ISS et apprendre à partager les données collectées par les capteurs de l'Astro Pi.

Activité 2.1 – Mesurer l'humidité

Le niveau d'humidité à bord de l'ISS est normalement maintenu aux alentours de 60 %. Ce n'est pas un aspect facile à réguler. Les activités de tous les jours augmentent constamment le taux d'humidité sur l'ISS. Sans compter que chaque astronaute consomme environ 2,7 litres d'eau par jour pour boire et manger, et qu'une partie de cette eau s'échappe de son corps sous forme de vapeur (par les pores ou par la respiration).

Si le niveau d'humidité est trop élevé, le système de survie de l'ISS fait en sorte de collecter la vapeur d'eau en surplus dans l'air. Pour cela, l'humidité est enregistrée en continu par des capteurs très précis dans l'ISS. Dans cet exercice, vous allez utiliser le capteur d'humidité de l'Astro Pi pour mesurer l'humidité dans votre salle de classe, exactement comme sur l'ISS.

Figure A3

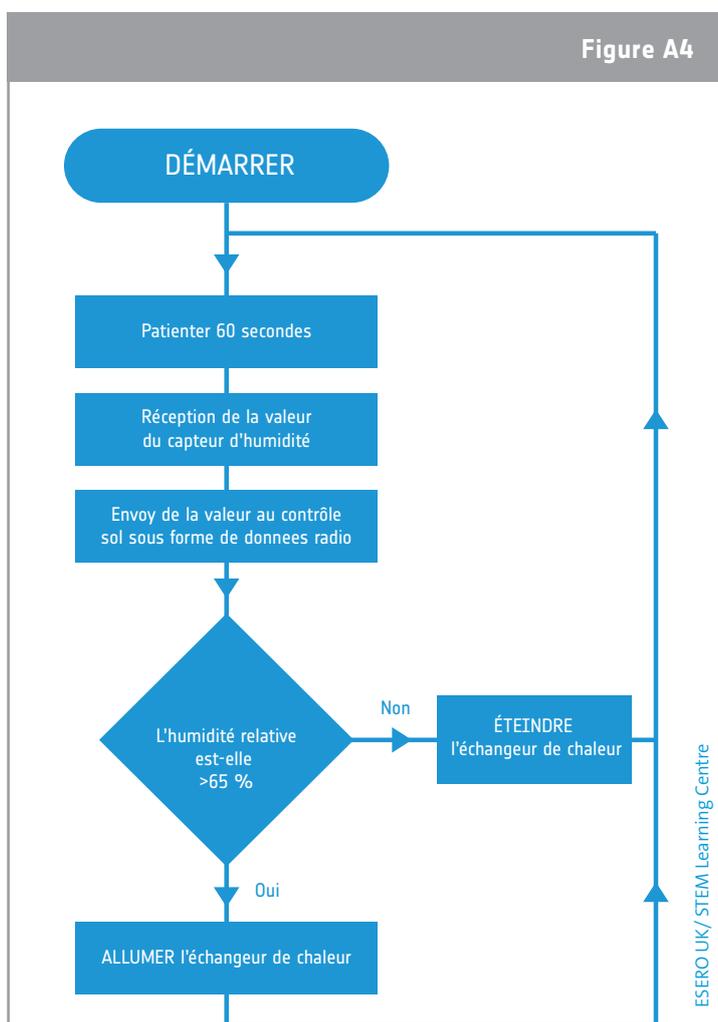


↑ L'astronaute de l'ESA Alexander Gerst sur le vélo d'appartement de la Station spatiale. Les activités de tous les jours, comme l'exercice physique, augmentent constamment le taux d'humidité sur l'ISS. (ESA/NASA)

Exercice

1. Pourquoi pensez-vous qu'il soit important de réguler l'humidité à l'intérieur de l'ISS ? Discutez avec vos camarades et donnez deux raisons justifiant l'utilisation d'un système de régulation d'humidité.

2. La Figure A4 contient un organigramme qui représente le système de régulation d'humidité de l'ISS. Dans le cadre ci-dessus, planifiez/écrivez un court programme en Python qui exécutera les trois premières étapes de l'organigramme.



↑ Système de régulation d'humidité. Si l'humidité relative est supérieure à 65 %, une série de ventilateurs et d'échangeurs thermiques (similaires à ceux qu'il y a dans un frigidaire) refroidissent et sèchent l'air pour diminuer la vapeur d'eau. (Centre de formation ESERO UK/ STEM)

3. Ouvrez une nouvelle fenêtre Python 3 et écrivez votre code. Sélectionnez File > Save (Fichier > Enregistrer) et choisissez un nom de fichier pour votre programme. Sélectionnez ensuite Run > Run module (Exécuter > Exécuter le module). Prenez note de l'humidité relevée.

Activité 2. 2 Collecte et envoi de données

Un réseau mondial de centres de contrôle au sol aide les astronautes qui vivent et travaillent dans l'ISS, notamment en mesurant leurs conditions de vie à bord de la Station spatiale. Pour cela, il est très important de partager les données collectées. Vous allez recréer l'étape 4 de l'organigramme de la Figure A4 et envoyer les données collectées sous la forme d'un fichier CSV (dont les valeurs sont séparées par des virgules) que vous pouvez partager et analyser.

Exercice

1. Les données collectées sur l'ISS sont envoyées au sol sous forme de données radio. Mettez par écrit ce que veut dire pour vous « données radio ».

2. Les données fournies par l'ISS sont au format CSV. C'est un format qui permet d'enregistrer les données sous une forme structurée tabulaire, plus facile à étudier et appliquer pour les scientifiques. Ouvrez une nouvelle fenêtre de shell Python et saisissez le code ci-dessous :

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
import time
sense = SenseHat()

#to open a file named Datafile in which the program will add the collected data
file = open("Datafile.csv", "a")

#to write in the file the name of the table columns. \n is used to break and to create a new line
file.write("Time, Humidity \n")

print ("Time, Humidity")

for n in range(20):
#repeatedly gets data from the humidity sensor until adding 20 lines to the file table
    humidity = sense.get_humidity()
    humidity = round(humidity, 2)

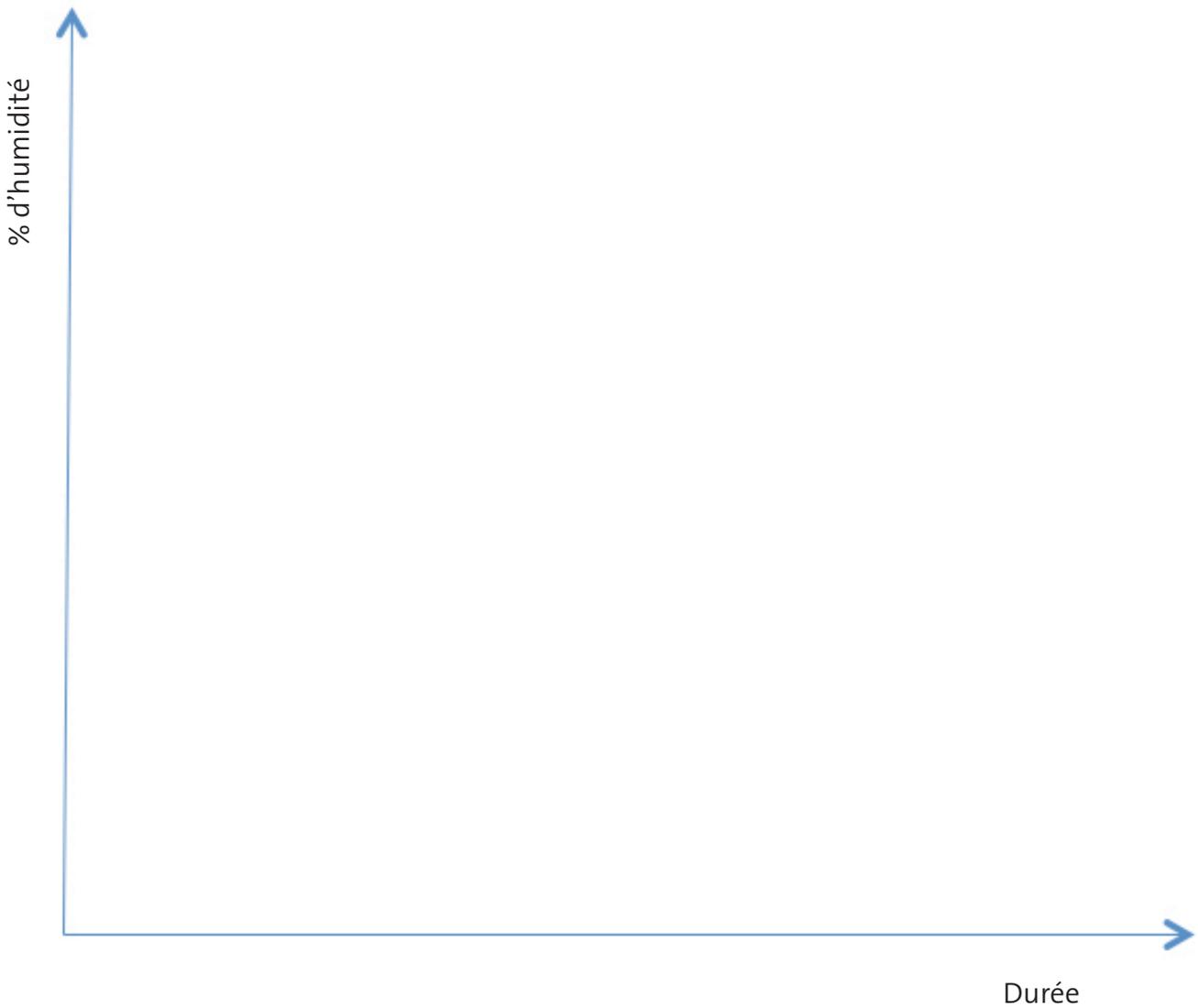
#to write in the file the data collected
    file.write(time.strftime('%X'))
    file.write(",")
    file.write(str(humidity))
    file.write("\n")

    print (time.strftime('%X'),humidity)

    time.sleep(1)

file.close()
```

3. Une fois vos données dans un format pratique, vous serez en mesure de les étudier de nombreuses manières. Les données collectées par le capteur d'humidité ont été enregistrées dans le gestionnaire de fichiers (vous pouvez y accéder dans le haut du bureau). Ouvrez le fichier et, en utilisant les données, tracez la courbe de l'humidité en fonction du temps sur le graphe ci-dessous.



4. Exécutez de nouveau votre code, mais cette fois soufflez doucement sur le capteur. Ouvrez de nouveau le fichier de données. Les nouvelles données collectées y ont été ajoutées. En utilisant ces nouvelles données, tracez vos nouveaux points d'humidité en fonction du temps sur le graphe ci-dessus. Que pouvez-vous conclure de la comparaison des deux courbes ?

Activité 2.3 Affichage de données

Sur l'ISS, les astronautes sont chargés de contrôler si l'équipement fonctionne correctement et de signaler les conditions anormales au contrôle sol. Vous allez recréer l'étape 5 de l'organigramme de la Figure A4 en créant une alarme visuelle pour signaler aux astronautes quand l'humidité dépasse 65 %.

Exercice

1. Examinez le bout de code suivant. Que pensez-vous qu'il fasse ? Écrivez votre réponse en modifiant les commentaires dans le cadre ci-dessous.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
import time

sense = SenseHat()

while True:
    humidity = sense.get_humidity()
    humidity = round(humidity, 2)

# _____
    if humidity < 65:
        bg = [0, 100, 0]

# _____
    else:
        bg = [100, 0, 0]

    sense.show_message(humidity, scroll_speed=0.05, back_colour=bg)
```

2. Copiez maintenant le code dans une nouvelle fenêtre Python. Sélectionnez File > Save (Fichier > Enregistrer) et choisissez un nom de fichier pour votre programme, sélectionnez ensuite Run > Run Module (Exécuter > Exécuter le module). Que se passe-t-il quand vous exécutez le code ?

Remarque : pour effacer la matrice, saisissez la commande `sense.clear()` (`sense.effacer()`) dans le shell de Python.

3. Le Sense HAT peut uniquement afficher les données qu'il reconnaît comme des chaînes (c'est-à-dire composées des caractères du clavier), mais la variable d'humidité est à virgule flottante (c'est un nombre avec des chiffres après la virgule). Pour pouvoir afficher la valeur d'humidité sur le Sense HAT, vous devez donc convertir l'humidité en une chaîne. Pour cela, ajoutez la ligne suivante juste devant la dernière ligne du code ci-dessus :

```
humidity = str(humidity)
```

Exécutez de nouveau votre code. A-t-il fait ce que vous aviez prévu à l'exercice 1 ?

4. Expirez lentement sur les capteurs jusqu'à ce que atteigniez une valeur d'humidité supérieure à 65 %. Qu'arrive-t-il aux données affichées sur la matrice ?
-

5. Une meilleure manière de visualiser l'humidité peut être de dessiner un diagramme en barres simple. Essayez le code ci-dessous dans un nouveau fichier.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
sense = SenseHat()
white = (255,255,255)
for height in range (4):
    sense.set_pixel(3,height, white)
    sense.set_pixel(4,height,white)
```

- a) Que se passe-t-il si vous changez le nombre dans range(4) (plage(4)) ?
-

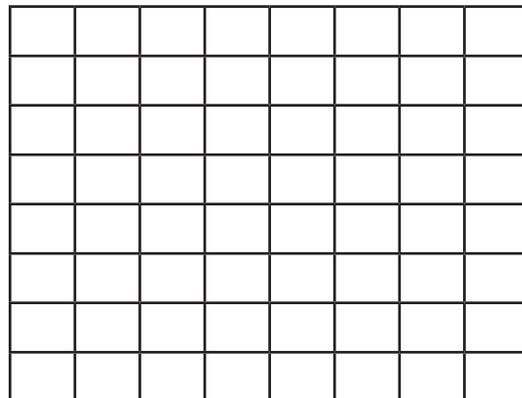
- b) Quel est le plus grand nombre que vous pouvez mettre dans range() sans obtenir d'erreur ? Expliquez pourquoi.
-

6. L'humidité peut varier de 0 à 100. Si cette valeur pouvait être réduite à entre 0 et 8, elle pourrait être utilisée dans range() pour générer un diagramme de l'humidité dans l'air. Diviser l'humidité par environ 12,5 devrait faire l'affaire.

- a) Pouvez-vous expliquer pourquoi nous avons choisi 12,5 ?
-
-

b) Sur la grille ci-dessous, représentez ce que vous pensez qui s’affichera sur la matrice de LED si vous saisissez le code ci-dessous (supposez que l’humidité soit la même que celle affichée à l’exercice 3). Expliquez votre représentation.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
sense = SenseHat()
white = (255,255,255)
humidity = sense.get_humidity()
# int is used to convert data to an integer number
humidity = int(humidity/12.5)
for height in range (humidity):
    sense.set_pixel(3,height, white)
    sense.set_pixel(4,height,white)
```



c) Pour vérifier votre réponse, copiez votre code dans un nouveau fichier Python. Sélectionnez File > Save As (Fichier > Enregistrer sous) et choisissez un nom de fichier pour votre programme, sélectionnez ensuite Run > Run module (Exécuter > Exécuter le module).

d) Ajoutez une boucle « while True » (tant que la condition est vraie) à votre code et regardez s’il mesure bien constamment l’humidité et l’affiche sous forme de diagramme en barres. Copiez votre code dans le cadre ci-dessous.

Extension

Essayez de créer une mini-station environnementale régulée dans votre salle de classe et contrôlez la température, l’humidité et la pression en utilisant votre Astro Pi. Vous pouvez commencer votre projet en enquêtant sur les conditions environnementales idéales dans une salle de classe. Vous pourrez ensuite créer un système d’alarme qui vous informera de si les conditions sont normales ou non.

→ ACTIVITÉ 3 – DE QUEL CÔTÉ DESCEND-ON ?

Sur Terre tout est soumis à la pesanteur. C'est la force qui, lorsque vous sautez, vous ramène continuellement au sol. C'est pour cela que, sur Terre, il est facile de savoir où est le bas. Le « bas » correspond au sens dans lequel la pesanteur vous attire et le « haut » à l'inverse.

Dans la Station spatiale internationale, il n'y a ni haut ni bas. Il n'y a aucune différence entre le sol et le plafond. Cette désorientation peut donner mal au cœur aux astronautes (c'est ce que l'on appelle le « mal de l'espace ») tant qu'ils ne s'habituent pas à cet étrange environnement.

Dans cette activité, vous allez apprendre à utiliser le capteur accéléromètre pour détecter l'orientation.

Exercice

1. La Station spatiale internationale tourne autour de la Terre sur une orbite circulaire à une altitude de 400 km. Étant donné que la pesanteur s'affaiblit avec la distance, la pesanteur à cette altitude est d'environ 90 % de ce qu'elle est à la surface de la Terre. Mais si vous regardez les astronautes sur la Station spatiale internationale, vous avez l'impression qu'ils flottent sans avoir la notion du haut et du bas. Pouvez-vous expliquer l'origine de ce phénomène ?

Figure A5

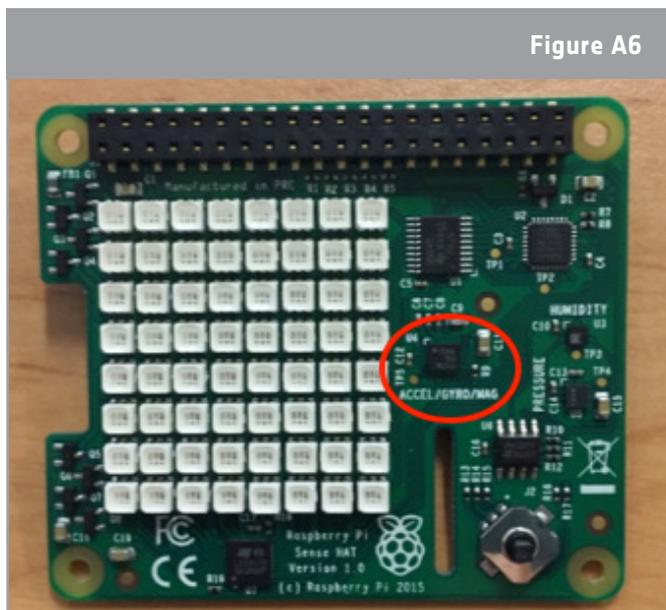


↑ Il n'y a ni haut ni bas dans l'ISS (ESA/NASA)

2. Le Sense HAT a un capteur de mouvement appelé un IMU (Inertial Measurement Unit, unité de mesure inertielle), qui rassemble en fait trois capteurs en un :
- un gyroscope (qui mesure la rotation et la quantité de mouvement) ;
 - un accéléromètre (utilisé pour trouver le sens de la pesanteur quand un objet est au repos) ;
 - un magnétomètre (qui mesure le champ magnétique de la Terre, un peu comme une boussole).

Les accéléromètres mesurent en mètres par seconde carrée (m/s^2) ou en forces g (g). g est le symbole correspondant à l'accélération moyenne ($9,8 m/s^2$) produite par la pesanteur à la surface de la Terre (au niveau de la mer). Une force g sur la planète Terre équivaut à $9.8 m/s^2$.

Ouvrez une nouvelle fenêtre Python et saisissez le code ci-dessous :



↑ IMU sensor in the Sense HAT

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
sense = SenseHat()
acceleration = sense.get_accelerometer_raw()
print(acceleration)
```

Open a new Python window and type the code below:

3. Exécutez votre code et notez les mesures de l'accéléromètre.

4. L'accéléromètre du Sense HAT collecte les données de 3 axes (soit en trois dimensions). Ce n'est pas très utile de lire les données dans le format que vous venez d'imprimer. Pour imprimer les données dans un format plus pratique, copiez et exécutez le code suivant.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
sense = SenseHat()

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

x=round(x, 0)
y=round(y, 0)
z=round(z, 0)

print("x {0} y {1} z {2}".format(x,y,z))
```

5. Notez les résultats. Quel est selon vous le sens de la force de pesanteur ? Discutez de votre réponse avec vos camarades et votre professeur.

6. Faites pivoter votre Astro Pi de 90 degrés. Notez les résultats et expliquez les différences par rapport aux résultats précédents.

7. Complétez le schéma ci-dessous pour indiquer le sens des axes X, Y et Z depuis votre Astro Pi. Si vous le jugez nécessaire, vous pouvez faire pivoter encore une fois votre Astro Pi.

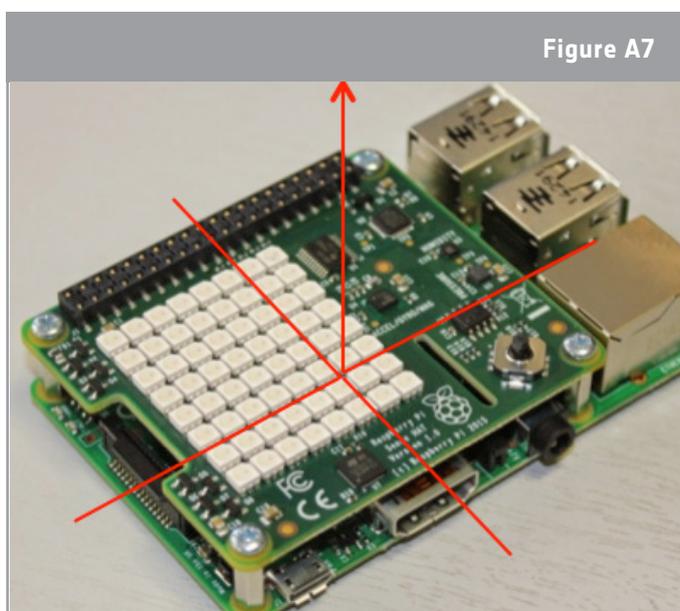


Figure A7

8. Le Tableau 2 montre les relevés des axes X, Y et Z de l'accéléromètre de l'Astro Pi sur l'ISS une fois par jour de travail. Pourquoi pensez-vous que la lecture de l'accéléromètre est proche de zéro ? Pensez-vous que cela puisse être une anomalie ? Essayez de répondre à cette question en comparant les données collectées sur l'ISS avec les données que vous avez déjà recueillies. Examinez votre réponse à la Question 1 de cette activité aussi. Discutez de votre réponse avec vos camarades et votre professeur.

Tableau A2 - Relevés de l'accéléromètre de l'Astro Pi de l'ISS:

accel_x	accel_y	accel_z	time_stamp
-0,00057	0,019359	0,014357	10:45:00 AM
-0,00044	0,019405	0,014425	11:45:00 AM
-0,00056	0,019531	0,014597	12:45:00 AM
-0,00056	0,019506	0,014432	01:45:00 PM
-0,00058	0,019464	0,014569	02:45:01 PM
-0,00056	0,01939	0,014578	03:45:00 PM
-0,00053	0,019384	0,014389	04:45:00 PM
-0,00046	0,01926	0,01444	05:45:00 PM
-0,00053	0,019266	0,014568	06:45:01 PM

Extension

1. L'ISS perd de 50 à 100 mètres d'altitude par jour. Si cela n'était pas contrôlé, l'orbite s'abaisserait, sa hauteur diminuerait et l'ISS se retrouverait en grand danger. Ce phénomène est dû au fait que, même à une altitude de 400 km, il y a encore un peu d'atmosphère. Cet air crée une traînée atmosphérique qui entraîne un lent abaissement de l'orbite de l'ISS dans le temps. Pour éviter cela, des rehaussements orbitaux sont effectués par des moteurs-fusées. Ils sont programmés à raison de 3 à 4 par mois. Quelles données peuvent être collectées par l'Astro Pi pour enregistrer ces événements ?

2. Téléchargez [ici](#) les données collectées dans le module Columbus du 16/02/2016 au 29/02/2016. Pouvez-vous deviner s'il y a eu des rehaussements de l'ISS pendant cette période et combien de temps ils ont duré ? Consultez [ici](#) le dernier graphique d'altitude ; vous pourrez peut-être établir une relation avec les données du fichier. Notez vos conclusions.

teach with space – comment collecter les données de l'astro pi | T05.3b
www.esa.int/education

An ESA Education production

in collaboration with Raspberry Pi Foundation, UK Space Agency, ESERO Poland and ESERO UK

Copyright 2017 © European Space Agency