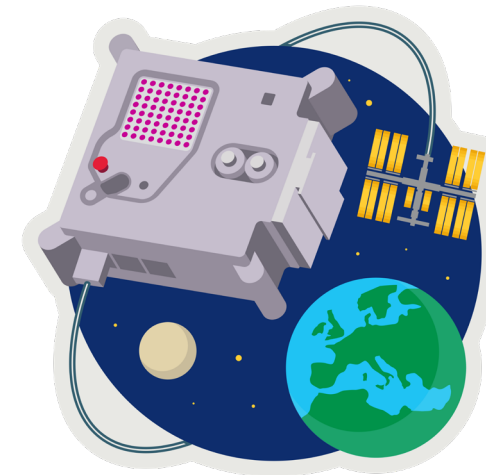


De Sense HAT programmeren

Met de Sense HAT kun je allerlei omstandigheden in het ISS waarnemen en opslaan. Ook kun je ze laten zien als tekst of plaatje op het ledscherm. In deze reeks activiteiten maak je kennis met de Pythoncommando's van de Sense HAT.

Wat heb je nodig?

geprepareerde Astro Pi met monitor, toetsenbord en muis of een pc met de Sense HAT-emulator. Ga naar https://missions.astro-pi.org/nl-NL/mz/code_submissions/new.



ASTRO PI

Starten

De Astro Pi bestaat uit drie onderdelen

- Raspberry Pi-computer
- Sense HAT met ledscherm en sensoren
- Camera (infrarood of zichtbaar licht)

Als je een Astro Pi hebt, dan kun je rechtstreeks op de Astro Pi programmeren. Start de Astro Pi op. Klik op het Raspberry Pi-logo linksboven. Selecteer Programming > Thonny Python IDE. Er verschijnt een venster met een Python-shell. Selecteer File > New File. Er start een nieuw venster waar je je programmacode in kunt schrijven.

Als je geen Astro Pi hebt, dan kun je toch een programma schrijven en testen. Hiervoor gebruik je de Sense HAT-emulator. Open je webbrowser en ga naar https://missions.astro-pi.org/nl-NL/mz/code_submissions/new. Verwijder de code die in het linkervak staat en je kunt starten met programmeren.

Meer weten

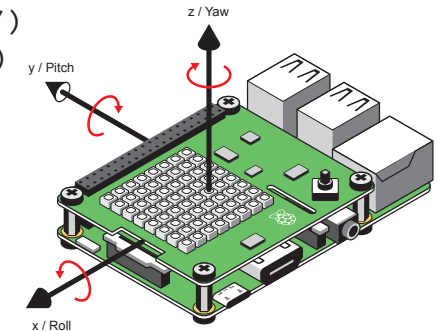
Je kunt de drie waarden uit de sensoren ook opslaan als variabelen:

```
orientatie = sense.get_orientation()
```

```
pitch = orientatie.get('pitch')
```

```
roll = orientatie.get('pitch')
```

```
yaw = orientatie.get('yaw')
```



De bewegingssensor bestaat eigenlijk uit drie losse sensoren:

- de gyroscoop: meet of de sense HAT draait (in radialen per seconde)
- de accelerometer: meet of er kracht op de Sense HAT staat (in g)
- de magnetometer: meet het magneetveld (in microtesla)

Deze drie sensoren werken op een slimme manier samen om te bepalen hoe de Sense HAT in de ruimte staat: rechtop, op zijn kop, op zijn zij, enzovoorts. Elke sensor geeft drie waarden: een voor elke as. Van elke sensor kun je de ruwe data uitlezen:

```
sense.get_gyroscope_raw()
```

```
sense.get_accelerometer_raw()
```

```
sense.get_compass_raw()
```

Bij de gyroscoop lees je pitch, roll en yaw uit, bij de kompas en de accelerometer lees je x, y en z uit.

In het ISS geeft de bewegingssensor andere resultaten dan op aarde. Vooral de accelerometer werkt anders. Probeer de accelerometer uit te lezen en bereken wat er anders zou zijn in het ISS.

De bewegingssensor

De Sense HAT heeft ook een set sensoren waarmee je beweging kunt meten. Zo kun je meten of de Sense HAT verplaatst of gedraaid wordt.

Schrijf onderstaand programma en voer het uit.

```
from sense_hat import SenseHat
sense = SenseHat()

orientation = sense.get_orientation()

#een makkelijke manier om de resultaten te laten zien:
print(sense.get_orientation())

#een ingewikkelde manier om de resultaten te laten zien:
print("p: {pitch}\nr: {roll}\ny: {yaw}\n\n".format(**orientation))
```

De waarden verschijnen niet op de Sense HAT, maar op een extra scherm. In de Sense HAT-emulator vind je dit scherm onder de Sense HAT. Achter op dit werkblad kun je lezen hoe de bewegingssensor werkt.

Kijk nog eens goed naar de code en naar het resultaat en denk na over de volgende dingen:

- Bij de opdracht `print`: welke tekens verschijnen letterlijk op het scherm en welke tekens niet?
- Wat doet `\n`?
- Waarvoor zijn de accolades `{}`? En wat betekenen de woorden die ertussen staan?

Je eerste computerprogramma

Schrijf in de Sense HAT-emulator de volgende code:

```
#importeer de functies van de Sense Hat
from sense_hat import SenseHat

#maak een variabele waarmee je functies van de Sense Hat
eenvoudig kunt oproepen
#dit is niet per se nodig, maar anders moet je overal waar
sense staat SenseHat() invullen
sense = SenseHat()

#laat de volgende tekst op het display zien
sense.show_message("Groetjes van de aarde!")
```

`sense.show_message` laat tekst verschijnen op het ledscherm. Je kunt zelf bepalen wat voor tekst je invoert. Je kunt ook de scrolsnelheid veranderen. Probeer maar eens het volgende commando in te voeren:

```
sense.show_message("Groetjes van de aarde!", scroll_
speed=0.05)
```

Meer experimenteren

Probeer ook eens het volgende programma uit:

```
from sense_hat import SenseHat
sense = SenseHat()

#importeer functies uit 'time'
#nodig voor time.sleep
import time

sense.show_letter("h")
time.sleep(0.3)
sense.show_letter("o")
time.sleep(0.3)
sense.show_letter("i")
```

Stel jezelf de volgende vragen:

- Wat zijn de verschillen tussen show_message en show_letter?
- Wat doet sleep?

Meer experimenteren

Luchtvochtigheid is een percentage, waarbij 0% 'niets' is en 100% 'alles'. Dit kun je mooi weergeven door het scherm te vullen met pixels. Alle pixels uit betekent een luchtvochtigheid van 0%; de helft van de pixels aan betekent 50% en alle pixels aan betekent 100%. Met de volgende code kun je het scherm de opdracht geven om een bepaald aantal pixels te laten branden.

```
while True:

#meet de luchtvochtigheid en zet dit getal om in een
aantal pixels
#int maakt van de uitkomst een geheel getal
lv = sense.get_humidity()
pixels = int(lv/100*64)

x=0
y=0
sense.clear()

#volg deze instructie zolang n tussen 0 en pixels zit
#na elke lus wordt n met 1 verhoogd
#op die manier worden alle pixels een voor een aangezet
for n in range (0,pixels):
    sense.set_pixel(x,y,0,0,255)
```

Deze code is niet compleet. Je moet nog regels voor x en y opstellen om zo de juiste lampjes aan te zetten. Gebruik hiervoor de volgende commando's:

```
x+=1 #tel 1 bij x op
y+=1 #tel 1 bij y op
```

De luchtvochtigheidssensor

De Sense HAT heeft een luchtvochtigheidssensor waarmee je de relatieve luchtvochtigheid kunt meten. De relatieve luchtvochtigheid is de hoeveelheid waterdamp als percentage. Een waarde van 0% betekent dat er geen waterdamp in de lucht zit. Een waarde van 100% betekent dat de lucht is verzadigd met waterdamp.

Schrijf onderstaand programma en voer het uit.

```
from sense_hat import SenseHat
sense = SenseHat()

#lees de luchtvochtigheid van de sensor en sla op als lv
lv = sense.get_humidity()

#rond lv af op twee decimalen
lv = round(lv,2)

#zet het nummer om in een stuk tekst
lv = str(lv)

sense.show_message(lv)
```

De luchtvochtigheid meten werkt hetzelfde als de temperatuur meten. Je gebruikt alleen een ander commando.

Voer deze regel in om voor de persoon die het scherm leest duidelijk te maken dat het om luchtvochtigheid gaat:

```
sense.show_message("luchtvochtigheid is " + lv + "%")
```

De kleuren van het scherm

Het scherm van de Sense HAT bevat 64 leds die verschillende kleuren kunnen weergeven. In elk van de 64 leds bevinden zich drie kleinere leds: een rode, een groene en een blauwe, net als in het scherm van een tv of smartphone. Je kunt alle drie deze kleuren apart aansturen.

Schrijf het volgende programma:

```
from sense_hat import SenseHat
sense = SenseHat()

#de drie basiskleuren van de ledlampjes
#elke kleur kan een waarde 0 tot 255 hebben
rood = (255,0,0)
groen = (0,255,0)
blauw = (0,0,255)

sense.show_message("rood", text_colour=rood)
sense.show_message("groen", text_colour=groen)
sense.show_message("blauw", text_colour=blauw)
```

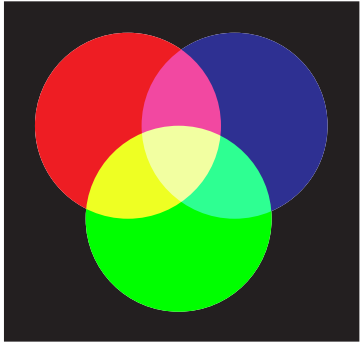
Je kunt ook kleuren mengen. Ook kun je het getal dat hoort bij een kleur, rechtstreeks invoeren in `sense.show_message`. Hieronder een voorbeeld.

```
sense.show_message("welke kleuren zijn dit?", text_
colour=(255,255,0), back_colour=(255,0,255))

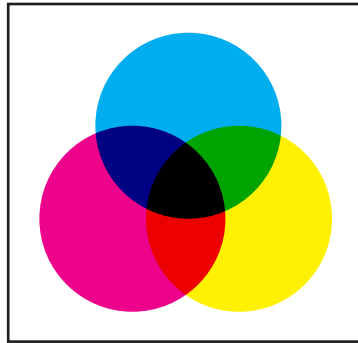
#maak het scherm weer zwart
sense.clear()
```

Meer weten

Er bestaan oneindig veel kleuren, maar jouw ogen kunnen er maar drie zien: rood, blauw en groen. Elke andere kleur die jij ziet, is een mengsel van deze drie kleuren. Zo is geel voor ons hetzelfde als rood plus groen. Wit is voor ons hetzelfde als alle drie de kleuren samen.



Kleuren mengen met licht



Kleuren mengen met verf

Let op: kleuren mengen met licht werkt anders dan kleuren mengen met verf. De drie basiskleuren van verf zijn magenta (rozerood), cyaan (soort blauw) en geel. Dit komt doordat je met licht kleuren bij elkaar optelt en met verf juist het licht filtert.

Meer experimenteren

Probeer ook eens de volgende regel uit:

```
sense.show_message(str(round(sense.get_temperature())))
```

Op deze manier voer je alle opdrachten in één regel uit. Vind jij dit handiger of juist niet?

Schrijf nu een programma waarbij je elke seconde de temperatuur op het display laat zien. Maak gebruik van het `while True:-`commando.

Hieronder een voorbeeld dat nog niet helemaal af is. Vul dit aan. Verander de temperatuurschuif in de Sense HAT-emulator en controleer of de temperatuur verandert.

```
while True:
    #let op, er ontbreekt hier nog iets!

    sense.show_message(temp, scroll_speed=0.03)
    time.sleep(2)
```

Let op! Na `while True:` staat een dubbele punt en de regels eronder zijn ingesprongen.

Je kunt ook een plaatje maken dat reageert op de temperatuur.

Bijvoorbeeld een smiley als de temperatuur boven de 20 graden is en een uitroepteken als die onder de 20 graden is. Bedenk zelf hoe je dat kunt doen en probeer het uit. Hint: maak gebruik van het commando `if (temp<20):`. Vergeet ook hier de dubbele punt en de ingesprongen regels niet.

De temperatuursensor

De Sense HAT heeft een temperatuursensor waarmee je de temperatuur kunt meten. Schrijf onderstaande code en voer uit.

```
from sense_hat import SenseHat
sense = SenseHat()

#lees de temperatuur van de sensor en sla op als temp
temp = sense.get_temperature()

#rond de temperatuur af op twee decimalen
temp = round(temp,2)

#zet het nummer om in een stuk tekst
temp = str(temp)

sense.show_message("temp")
sense.show_message(temp)
```

Stel jezelf de volgende vragen:

- Wat gebeurt er als je de regel met round weglaat?
- Wat doet de functie str?
- Wat is het verschil tussen "temp" en temp?

De ledlampjes besturen

Het scherm bestaat uit kleine puntjes, pixels. Op het ledscherm kun je een afbeelding van 64 pixels maken. Ontdek hier hoe je elke pixel apart kunt aansturen.

Schrijf het volgende programma:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()

#zet de x- en de y-coördinaat op 0
x=0
y=0

#zet de kleuren op maximale sterkte
r=255
g=255
b=255

sense.set_pixel(x,y,r,g,b)
```

Verander x in waarden 0 t/m 7. Wat gebeurt er op het scherm?

Doe hetzelfde voor y.

Waarom kun je voor x of y geen 8 of hoger invullen?

Probeer ook eens andere kleuren uit door r, g en b te veranderen.

Een pixel laten bewegen

Een bewegende animatie bestaat eigenlijk uit een heleboel plaatjes die je snel achter elkaar ziet. Dit kun je ook op een slimme manier doen met het ledscherm.

Schrijf het volgende programma:

```
from sense_hat import SenseHat
import time
sense = SenseHat()

x=0
y=0

#herhaal zolang x kleiner dan 8 is
while (x<8):
    sense.clear()
    sense.set_pixel(x,y,255,255,255)
    time.sleep(0.1)
    x=x+1
```

Het `while`-commando noemen we ook wel een lus of loop (spreek uit als loop). Op die manier kun je een opdracht een aantal keren herhalen, maar hoeft je niet meerdere keren dezelfde code te schrijven. Let op! Vergeet de dubbele punt en de ingesprongen regels niet.

Probeer de pixel ook eens naar boven te laten bewegen, of schuin omhoog.

Meer experimenteren

```
p = (240, 230, 140) #Khaki
q = (255, 255, 0) #Geel
r = (184, 134, 11) #Donkergeel
s = (139, 69, 19) #Bruin
t = (255, 140, 0) #Oranje
u = (178, 34, 34) #Donkerrood
v = (255, 0, 0) #Rood
w = (255, 192, 203) #Roze
y = (255, 20, 147) #Donkerroze
z = (153, 50, 204) #Paars
```


Meer experimenteren

Je kan het scherm van de Astro Pi een kleur laten weergeven na je meting. Voeg twee regels toe waarbij het scherm een kleur laat zien na je meting. Maak gebruik van de verschillende waarden die je rood, groen en blauw kan geven. Het commando `sense.clear()` maakt het scherm leeg en laat een gekozen kleur zien. Als je geen kleur invult, wordt het scherm zwart.

```
#kies je eigen waarden voor rood, groen en blauw tussen
0 - 255
x = (178, 34, 34)
sense.clear(x)
```

Je kunt ook je eigen plaatje maken.

Kies de kleuren uit die je wilt gebruiken en voeg deze toe aan je code. Om je plaatje te maken, verander je de letters tussen de blokhaken.

Kan je een slang maken? En een bloem?

```
a = (255, 255, 255) #Wit
b = (105, 105, 105) #Grijs
c = (0, 0, 0) #Zwart
d = (100, 149, 237) #Lichtblauw
e = (0, 0, 205) #Blauw
f = (25, 25, 112) #Donkerblauw
g = (0, 191, 255) #Middenblauw
h = (0, 255, 255) #Cyaan
j = (143, 188, 143) #Donker zeegroen
k = (46, 139, 87) #Zeegroen
l = (0, 255, 127) #Lichtgroen
m = (34, 139, 34) #Donkergroen
n = (154, 205, 50) #Geelgroen
o = (128, 128, 0) #Olijf
p = (240, 230, 140) #Khaki
```

Het hele scherm besturen

Via het commando `set_pixels` bestuur je het hele scherm ineens. Zo hoef je niet voor elke pixel apart een commando te schrijven.

Schrijf onderstaand programma en voer uit.

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()

w=(255,255,255)
z=(0,0,0)

smiley = [
    Z,Z,W,W,W,W,Z,Z,
    Z,W,Z,Z,Z,W,Z,
    W,Z,W,Z,Z,W,Z,W,
    W,Z,Z,Z,Z,Z,W,
    W,W,Z,Z,Z,W,W,
    W,Z,W,W,W,W,Z,W,
    Z,W,Z,Z,Z,W,Z,
    Z,Z,W,W,W,W,Z,Z]

sense.set_pixels(smiley)
```

Verander de afbeelding door de kleuren te veranderen. Wat kun je nog meer tekenen?

Meer experimenteren

Met de commando's `set_pixel`, `set_pixels` en `clear()` kun je eindeloos variëren in wat je op het scherm laat zien. Probeer eens de volgende dingen:

- Laat een oog knipperen van de smiley. Welke pixel moet je dan veranderen?
- Probeer zelf een plaatje te maken van iets wat je leuk vindt. Welke kleuren heb je nodig?
- Hoe kun je een plaatje laten bewegen? Welke berekeningen moet je dan uitvoeren?

De kleurensensor

De Sense HAT heeft een kleurensensor waar mee je de kleuren uit de omgeving kunt meten. Vul je code aan met onderstaande code en voer uit.

```
from sense_hat import SenseHat
from time import sleep
sense = SenseHat()
sense.clear()

sense.set_rotation(270)

w=(255,255,255)
z=(0,0,0)

for i in range(28):
    rgb = sense.colour
    z = (rgb.red,rgb.green,rgb.blue)

    smiley = [
        w,w,z,z,z,z,w,w,
        w,z,w,w,w,w,z,w,
        z,w,z,w,w,z,w,z,
        z,w,w,w,w,w,z,
        z,z,w,w,w,w,z,z,
        z,w,z,z,z,z,w,z,
        w,z,w,w,w,w,z,w,
        w,w,z,z,z,z,w,w]

    sense.set_pixels(smiley)
    sleep(1)
```

Stel jezelf de volgende vragen:

- Wat doet `sleep`?
- Wat gebeurt er na 29 seconden?
- Hoe kan je dit veranderen?