

De Sense HAT programmeren

De Sense HAT is een uitbreiding voor de Raspberry Pi die speciaal voor de Astro Pi-wedstrijd is gemaakt. Met dit bord is het mogelijk om allerlei informatie waar te nemen en weer te geven op een led-scherm. In deze reeks activiteiten maak je kennis met de Python-commando's van de Sense HAT.

Wat je nodig hebt

- Astro Pi kit met monitor, toetsenbord en muis of
- Een PC met de Sense HAT emulator (www.trinket.io/sense-hat)



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 sh staat SenseHat() invullen

sh = SenseHat()

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

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

```
sh.show_message("Groetjes van de aarde!", scroll
_speed=0.05, )
```

Meer experimenteren

Probeer ook eens het volgende programma uit

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

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

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

Stel jezelf de volgende vragen:

- wat zijn de verschillen tussen show_message en show_letter?
- wat doet time.sleep?

Starten

De Astro Pi bestaat uit drie onderdelen

- Raspberry Pi computer
- Sense HAT met LED scherm 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 linksbovenin. Selecteer *Programming* > *Python 3*. Selecteer *File* > *New File* en je kunt starten met je programma schrijven in het nieuwe venster.

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 www.trinket.io/sense-hat en je kunt starten met programmeren!

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
sh = SenseHat()

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

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

Je kunt ook kleuren mengen. Ook kun je de getallen die horen bij een kleur rechtstreeks invoeren in `sh.show_message`. Hieronder een voorbeeld.

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

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

De LED lampjes apart besturen

Het scherm kan meer dan alleen tekst weergeven. Je kunt ook een afbeelding maken. Een afbeelding bestaat uit kleine puntjes, pixels. Op het LED scherm kun je een afbeelding van 64 pixels maken. Ontdek hier hoe je de pixels apart kunt aansturen.

Schrijf het volgende programma.

```
from sense_hat import SenseHat
sh = SenseHat()
sh.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

sh.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 is eigenlijk een heleboel plaatjes die je snel achter elkaar ziet. Dit kun je ook op een slimme manier doen met het LED scherm.

Schrijf het volgende programma

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

x=0
y=0

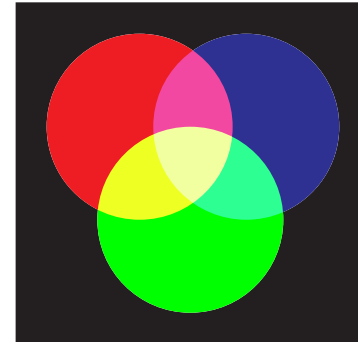
#Herhaal zolang x kleiner dan 8 is
while (x<8):
    sh.clear()
    sh.set_pixel(x,y,255,255,255)
    time.sleep(0.1)
    x=x+1
```

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, zodat je niet meerdere keren dezelfde code hoeft te schrijven.

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

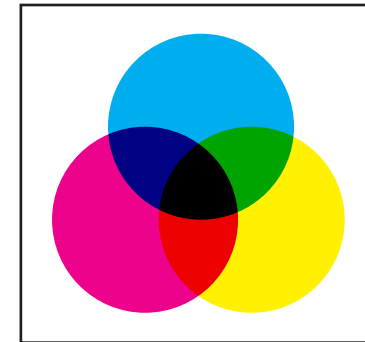
Meer weten

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



Additief:

kleuren bij elkaar op tellen



Subtractief:

kleuren van elkaar af trekken

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' (add) en bij verf juist het licht filtert (van elkaar af trekt, subtract).

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
sh = SenseHat()
sh.clear()

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

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

sh.set_pixels(smiley)
```

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

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
sh = SenseHat()

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

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

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

sh.show_message("temp")
sh.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`?

Meer experimenteren

Probeer ook eens de volgende regel uit:

```
sh.show_message(str(round(sh.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!

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

Je kunt ook een plaatje maken die reageert op de temperatuur. Bijvoorbeeld een smiley als de temperatuur boven de 20 graden is en een uitroepteken als het 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 de dubbele punt niet).

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 dat je leuk vindt. Welke kleuren heb je nodig?
- Hoe zou je een plaatje kunnen laten bewegen? Welke berekeningen moet je dan uitvoeren?

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 verzadigd met waterdamp is.

Schrijf onderstaand programma en voer het uit:

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

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

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

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

sh.show_message(lv)
```

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

Om voor de persoon die het scherm leest duidelijk te maken dat het om luchtvochtigheid gaat, voer deze regel in

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

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()
print("p: {pitch}\nr: {roll}\ny: {yaw}\n\n".format(**orientation))

zwaartekracht = sense.get_accelerometer_raw()
print("x: {x}\ny: {y}\nz: {z}\n".format(**zwaartekracht))
```

De waarden verschijnen niet op de Sense HAT, maar op het beeldscherm. Achterop 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?

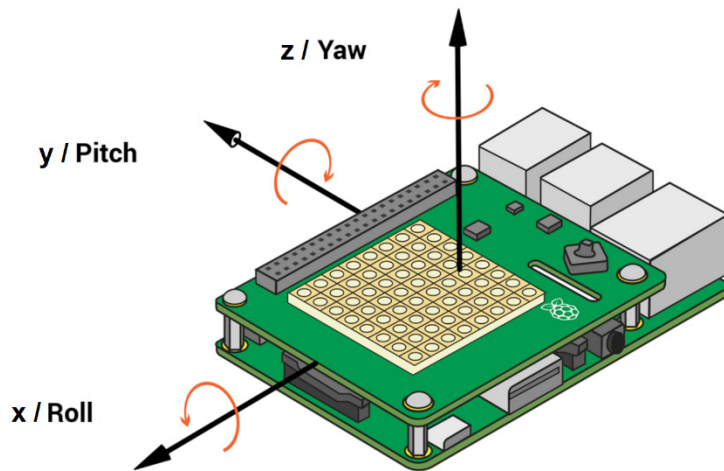
Met `get_accelerometer_raw()` kun je krachten meten in de eenheid G.

De zwaartekracht op aarde is 1G. Kun jij zien in welke richting de zwaartekracht staat in de Sense HAT emulator?

Meer weten

De bewegingssensor bestaat eigenlijk uit drie losse sensoren:

- De gyroscoop: meet of de Sense HAT draait
- De accelerometer: meet of er kracht (versnelling) op de Sense HAT staat
- De magnetometer: meet het magneetveld (de richting van de noordpool)



Deze drie sensoren werken op een slimme manier samen om zo te bepalen of de Sense HAT beweegt. Bewegingssensoren worden vaak gebruikt in mobiele telefoons, bijvoorbeeld om je scherm mee te draaien als je hem kantelt.

In het ISS zal hij heel anders werken dan op aarde. Hoe zou dat komen?

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.

```
#zet pixels om in een geheel getal
pixels = int(pixels)

#beginwaarden van x en y (de coördinaten van het
scherm)
x = int(0)
y = int(0)

#in deze lus worden de lampjes een voor een aangezet
for n in range (0,pixels):

    #bij 0 procent mogen er geen lampjes branden
    if pixels == 0:
        sh.clear()

    # x en y worden bij elke lus aangepast voor een
    extra pixel
    if x==8:
        x=0
        y+=1
    sh.set_pixel(x,y,0,0,255)
    x+=1
```

Deze code is nog niet compleet! We hebben namelijk nog niet uitgerekend hoeveel pixels er aan moeten. Voeg op de juiste plek een extra regel toe om pixels te berekenen.