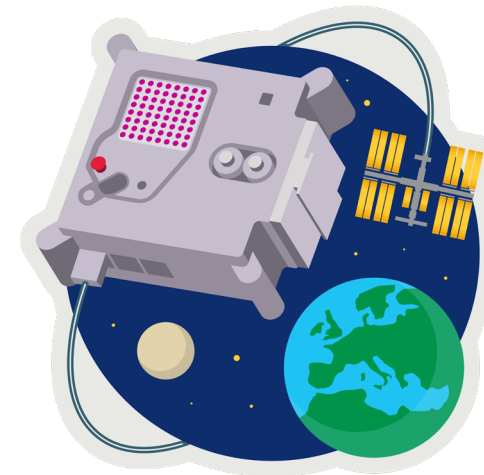


De Camera programmeren

In je Raspberry Pi Kit vind je een HQ camera en een 6mm lens. Deze camera heeft een infrarood filter.

Als je hebt aangegeven dat je camera infraroodlicht moet opvangen, heb je ook een rood optisch filter en een inbussleutel gekregen. Hiermee kan je van je HQ camera een NoIR-camera maken (NoIR staat voor No InfraRed filter).

De camera's zijn bijna identiek en zijn op dezelfde manier te programmeren. Bij sommige werkbladen gaan we wat dieper in op de verschillen.



ASTRO PI

De camera gebruiksklaar maken

Als het goed is, heb je de camera al gemonteerd. Zo niet, kijk dan op werkblad 1 *de Astro Pi gebruiksklaar maken* hoe dat moet. Vergeet niet de camera aan te zetten!

Voor deze werkbladen heb je een echte Astro Pi met camera nodig. De camera kan niet geprogrammeerd worden in de Sense HAT-emulator.

Wil je met je camera infrarood licht zien? Ga naar dan <https://rpf.io/ap-msl-guide> om te zien hoe je van je HQ camera een NoIR-camera kunt maken. Hiervoor heb je een rood optisch filter en een inbussleutel nodig.

Dit werkblad is voor leerlingen met enige programmeerervaring. Heb je nog niet eerder geprogrammeerd? Doe dan eerst werkblad 2, *de Sense HAT programmeren*.

Meer weten

NDVI staat voor Normalized Difference Vegetation Index. Dit is een maat voor de gezondheid van een plant. De NDVI waarden van een NoIR camera bereken je als volgt:

$$\text{NDVI} = \frac{\text{NIR} - \text{ZL}}{\text{NIR} + \text{ZL}}$$

NIR staat voor de hoeveelheid nabij-infrarood en ZL voor de hoeveelheid zichtbaar licht. De NDVI waarde bereken je over het licht dat van een plant wordt weerkaatst. Een plant absorbeert het zichtbare licht, maar weerkaatst NIR. Een voorwerp dat veel NIR weerkaatst en veel ZL absorbeert, zal een waarde van rond de 1 hebben. Een voorwerp dat weinig NIR weerkaatst, maar veel ZL zal een waarde van rond de -1 hebben.

Elke pixel heeft een NDVI waarde die in de foto is weergegeven als kleur. Rood is hoog en blauw is laag.

Het beeld op het scherm laten zien

Schrijf in het programmeervenster het volgende computerprogramma.

```
# importeer commando's voor de camera
from picamera import PiCamera
cam = PiCamera()

# importeer het commando 'sleep' uit 'time'
from time import sleep

cam.start_preview()
sleep(10)
cam.stop_preview()
```

Let op! Eindig je programma ALTIJD met `camera.stop_preview()`. Doe je dat niet, dan blijft je scherm de preview weergeven en kun je het scherm niet meer gebruiken voor andere doeleinden.

- Ben je toch de laatste regel vergeten? Druk dan op CTRL+C. Typ daarna (blind) `camera.stop_preview()` gevolgd door ENTER.
- Om toch nog het scherm te gebruiken, kun je ook deze regel gebruiken: `cam.start_preview(alpha=200)`.

Meer weten



Bij de NoIR-camera zijn de kleuren iets anders dan bij de gewone camera. Dit komt doordat de NoIR ook infrarood licht waarneemt en laat zien als rood licht. Het beeld is dus roder dan in het echt.

Er zit ook meer blauw in het beeld. Dit komt door de automatische witbalans. Witbalans is een kleurcorrectie om het beeld realistischer te maken. Bij zonsondergang is de kleur rood bijvoorbeeld dominant. Onze hersenen corrigeren hiervoor en maken het beeld kleurrijker. De camera doet dat met witbalans: is er te veel rood in het beeld, dan wordt er extra blauw toegevoegd.

Deze twee effecten samen zorgen ervoor dat infrarood er op het beeld uitziet als een paarsachtige kleur. Zoals je ziet, stralen bladeren veel infrarood uit.

In de Astro Pi-kit zit een rood filtertje dat je voor de camera kunt houden. Deze houdt het groene en het blauwe licht tegen. Je houdt dan alleen rood en infrarood licht over. De camera zal opnieuw de witbalans corrigeren, ditmaal door meer groen en blauw toe te voegen. Hierdoor wordt het infrarood licht extra fel ten opzichte van het overige licht.

Draai het programma nog eens en doe afwisselend het rode filter wel en niet voor de lens. Je ziet dat het beeld gecorrigeerd wordt.

Extra: de gezondheid van planten

Wist je dat je met de NoIR camera kunt meten of planten gezond zijn? Dat kan door van je foto een NDVI afbeelding te maken. Ga naar <https://projects.raspberrypi.org/en/projects/astropi-ndvi> en volg de stappen.

- Maak hierna een NoIR foto van een plant met het rode filter.
- Maak een NDVI afbeelding van de foto volgens de beschreven stappen.
- Bekijk de NDVI afbeelding. Wat valt je op aan de plant? En wat betekenen de kleuren?

Meer weten

Een video opnemen is ongeveer hetzelfde als heel snel foto's achter elkaar maken. Een video bestaat uit 25 tot 60 plaatjes (frames) per seconde. Elke opgenomen pixel wordt razendsnel omgezet in videodata. Als de video een hoge resolutie (veel pixels) en een hoge framerate (veel plaatjes per seconde) heeft, kost het de Astro Pi veel moeite om de video te maken. Hierdoor kunnen foutmeldingen ontstaan. Als dat gebeurt, sluit dan andere programma's of verlaag de resolutie.

De PiCamera neemt op in het H.264-formaat. Dit zijn ruwe data die bestaan uit een serie plaatjes, zonder geluid en zonder informatie over hoe deze moeten worden afgespeeld. Als je dit bestand afspeelt met Omxplayer, dan zal hij te snel afspelen. Om de video correct af te spelen, moet je deze omzetten in MP4-formaat. Open de terminal en typ achtereenvolgens de volgende commando's:

<code>sudo apt-get update</code>	Het besturingssysteem wordt geüpdatet.
<code>sudo apt-get install gpac</code>	Er wordt een programma geïnstalleerd waarmee je H.264 kunt omzetten in MP4.
<code>MP4Box -add voorbeeld.h264 voorbeeld.mp4</code>	De H.264 video worden omgezet in een MP4-video.

De draaiing en resolutie van het beeld

Je kunt het beeld van de camera ook draaien. Verander de rotatie van de camera met de volgende code:

```
from picamera import PiCamera
cam = PiCamera()

from time import sleep

#stel de rotatie van de camera in
cam.rotation = 90

cam.start_preview()
sleep(10)
cam.stop_preview()
```

Stel jezelf de volgende vragen:

- In welke richting draait het beeld? Met de klok mee, of tegen de klok in?
- Probeer voor `cam.rotation` een aantal waarden uit tussen de -360 en 360 graden. Welke waarden kun je invoeren? Wat gebeurt er met waarden die niet deelbaar door 90 zijn?

Vervang nu `cam.rotation` door `cam.resolution = (64,64)`. Wat is resolutie? En wat betekent het getal 64?

Meer experimenteren

Rotatie

Schrijf een code waarmee de rotatie van de camera elke 3 seconden met 90 graden verandert. Gebruik een for-loop. Dat is de code hieronder.

```
for x in [0, 90, 180, 270]:  
    cam.rotation = x  
    sleep(3)
```

Bij een loop (lus in het Nederlands) wordt een opdracht een aantal keren herhaald. De waarde van x is elke keer anders.

Een strikvraag: hoeveel graden moet je het beeld in het ISS draaien voor een correct beeld?

Resolutie

De PiCamera maakt vanuit het ISS een foto van het aardoppervlak. De camera heeft een resolutie van 4056 x 3040 pixels. De maximale beeldhoek is 63 graden. Het ISS draait op een hoogte van 408 km. Hoeveel meter aarde zie je op 1 pixel van de camera?

Video's opnemen

Typ de volgende code in het venster:

```
from picamera import PiCamera  
cam = PiCamera()  
from time import sleep  
  
#de camera kan beter opnemen in een lage resolutie, zie  
#achterzijde werkblad  
cam.resolution = (640, 480)  
cam.start_preview()  
cam.start_recording('/home/pi/video.h264')  
  
#stelt de opnameduur in  
#deze werkt hetzelfde als time.sleep, maar werkt beter  
#als er fouten plaatsvinden tijdens het opnemen  
cam.wait_recording(10)  
  
cam.stop_recording()  
cam.stop_preview()
```

Om de video af te spelen, moet je een terminalvenster openen en deze opdracht schrijven:

```
omxplayer [naam video]
```

De video wordt te snel afgespeeld. Lees achterop dit werkblad hoe dat komt en hoe je dat oplost.

Foto's maken

Probeer eens een programma te schrijven waarbij je een film maakt waarbij de helderheid binnen 10 seconden van 1 naar 100 gaat.

Schrijf een code waarmee alle effecten tegelijk in één programma worden gebruikt. Hieronder staat de variabele effects. Dit is een array, een lijst met namen die je apart kunt aanroepen. Neem nu een film op waarbij het effect elke 2 seconden verandert. Gebruik weer een for-loop.

```
effects = [ 'none', 'negative', 'solarize', 'sketch',  
'denoise', 'emboss', 'oilpaint', 'hatch', 'gpen', 'pastel',  
'watercolor', 'film', 'blur', 'saturation', 'colorswap',  
'washedout', 'posterise', 'colorpoint', 'colorbalance',  
'cartoon', 'deinterlace1', 'deinterlace2']
```

```
for x in effects:
```

```
....
```

Met het programma hieronder laat je de camera een foto maken.

```
from picamera import PiCamera  
cam = PiCamera()  
from time import sleep  
  
#zet eerst de camera aan zodat je hem kunt richten  
cam.start_preview()  
sleep(5)  
  
#maak een foto en sla deze op als voorbeeld.jpg  
cam.capture('/home/pi/Desktop/voorbeeld.jpg')  
cam.stop_preview()
```

Jouw foto staat nu op het bureaublad.

Met onderstaande regels kun je tekst toevoegen aan jouw foto:

```
cam.annotate_text = 'Groeten uit de ruimte'
```

Als je de grootte van de tekst wilt veranderen, gebruik dan deze instructie. Probeer meerdere waarden uit en kijk welke jij het beste bij de foto vindt passen.

```
cam.annotate_text_size = 30
```

Meer weten en experimenteren

Let op! Foto's die vanuit het ISS worden teruggestuurd, krijgen een andere naam. Dan weet ESA welke foto van welk team is. Als jij informatie in de naam zet, gaat die mogelijk verloren. Je kunt de informatie dus beter als tekst in de foto weergeven.

Probeer eens een time lapse te maken. Bij een time lapse maak je een aantal foto's achter elkaar met vaste tussenpozen.

Maak een reeks van 10 foto's met een for-loop. Laat i van 0 tot 10 lopen.

```
for i in range (10):  
    cam.capture('/home/pi/Desktop/image%s.jpg' % i)  
    sleep(0.7)
```

Bekijk de eerste foto door erop te dubbelklikken. Met de pijltoetsen kun je dan naar de volgende foto. Als je dat snel doet, heb je een time lapsefilm!

Je kunt je time lapsefoto's aanvullen met tekst waarin je de datum laat zien. Handig als je wilt weten hoeveel tijd er tussen de foto's zit.

```
#deze regel moet aan het begin van je programma  
import datetime as dt  
  
#variabele met daarin de datum en tijd als tekst  
datumentijd = dt.datetime.now().strftime(, %d-%m-%Y %H:%M:%S')
```

Effecten

Met de volgende commando's kun je het beeld van de camera bewerken. Voeg ze toe aan een bestaand programma en kijk wat er gebeurt:

```
cam.brightness = 30  
cam.contrast = 50
```

Wat betekenen brightness en contrast? En welke waarden kun je invullen? Welke waarden geven het mooiste beeld?

Probeer ook de volgende regels uit:

```
cam.image_effect = 'negative'  
cam.image_effect = 'sketch'  
cam.image_effect = 'none'
```