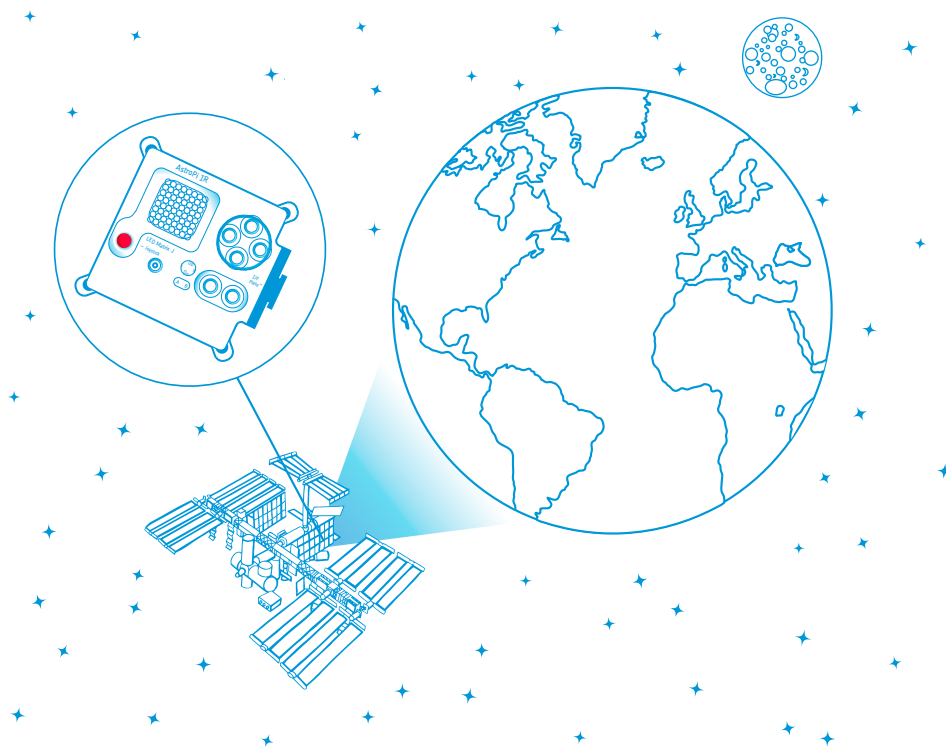


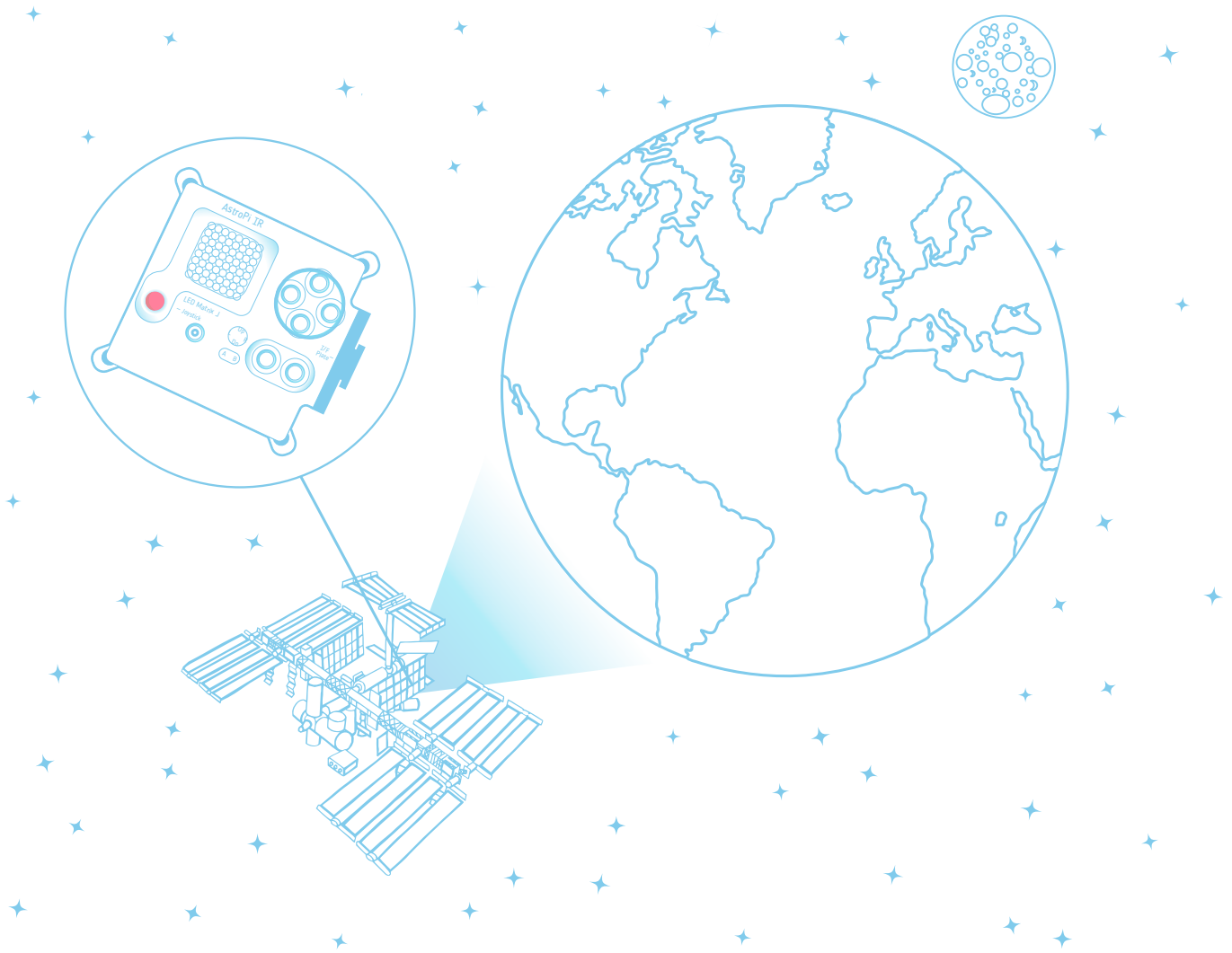
technology | T05.4

teach with space

→ UMGANG MIT DEN ASTRO-PI-KAMERAS

Mit einem Raspberry Pi Bilder machen
und in das nahe Infrarot hineinsehen





Aufgabe 0: Zusammenbau der Pi-Kamera Seite 4

Aufgabe 1: Verwendung der Pi-Kamera Seite 5

Aufgabe 2: Standbilder und Aufnahme
von Videos Seite 10

Aufgabe 3: In das nahe Infrarot hineinsehen Seite 13

Weiterführende Aufgabe – Verwendung
der NoIR-Kamera zur Überwachung des
Pflanzenwachstums Seite 15

Umgang mit den Astro-Pi-Kameras

Zwei besondere Astro Pis mit den Namen Ed und Izzy wurden für den Flug ins Weltall ausgesucht und befinden sich an Bord der Internationalen Weltraumstation (ISS). Ed und Izzy sind identisch bis auf ein Detail: Astro Pi Ed verfügt über eine Standardkamera für das sichtbare Spektrum, und Astro Pi Izzy über eine Infrarot- (bzw. NoIR-)Kamera. Beide Kameras decken das sichtbare Spektrum ab; die NoIR-Kamera reicht jedoch in das nahe Infrarot hinein und bietet viele zusätzliche Optionen für Experimente. Mit den Kameras könnt ihr Standfotos machen sowie Videos und Zeitrafferaufnahmen aufnehmen.

In diesem Aufgabenteil erkundet ihr die Astro-Pi-Kameras und lernt die wichtige Codiersprache, die ihr zum Programmieren der Kameras braucht.

Versuchsmaterial

- Raspberry Pi
- Stromversorgung
- Pi-Kamera (sichtbares Spektrum und NoIR)
- Monitor
- USB-Tastatur und -Maus

Die wichtigsten Fakten

Altersgruppe: 14-19

Art: Schüleraufgaben **Schwierigkeitsgrad:** leicht

Zeitbedarf: 2 Stunden

Erforderliche Materialien: Astro-Pi-Bausatz, Tastatur, Maus, Monitor, TV-Fernbedienung

Kurzfassung

Dieses Lernmittel bietet eine Einführung zur Verwendung der in dem Astro-Pi-Bausatz enthaltenen Kameras. Es enthält Anweisungen zum Aufbau und zur Verwendung der Kamera zur Aufnahme von Standbildern und Videos sowie Vorschläge für etwaige Anwendungen der Kamera. Das Ziel besteht darin, Anregungen zu vermitteln und die Schüler dabei zu unterstützen, die Möglichkeiten selber zu erkunden.

Lernziele

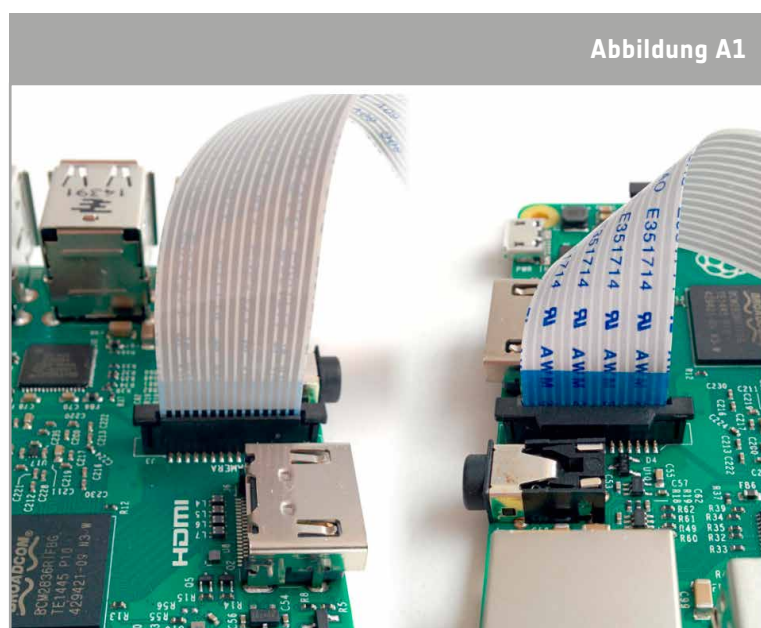
- Die Astro-Pi-Kameras installieren und aufbauen.
- Mit den Kameras Standbilder und Videos aufnehmen.
- Die Kameras mit Filtern versehen.
- Die Grundlagen des elektromagnetischen Spektrums, insbesondere den Infrarot-Bereich verstehen.
- Die Überwachung der Pflanzengesundheit unter Verwendung einer Kamera für nahes Infrarot erörtern.

→ AUFGABE 0: ZUSAMMENBAU DER PI-KAMERA

In dieser Aufgabe lernt ihr, die Pi-Kamera zusammenzubauen, anzuschließen und auf dem Raspberry Pi zu aktivieren.

Übung:

1. Schaltet zuerst euren Raspberry Pi aus und trennt ihn vom Stromnetz. Dann könnt ihr eure Kamera anschließen. Ihr könnt euch für beide Kameras entscheiden, denn der Installationsvorgang ist für beide Kameras derselbe. Sucht den Kamera-Port, zieht die Verriegelung hoch und steckt das Flachbandkabel hinein. Es sollte sich zwischen den Ethernet- und HDMI-Ports befinden, und die Silberstecker zu dem HDMI-Port zeigen. Der blaue Teil des Kabels sollte zum Ethernet-Port zeigen.



Anmerkung: Um technische Probleme zu vermeiden, installiert eure Kamera am Astro Pi nur bei ausgeschaltetem Computer.

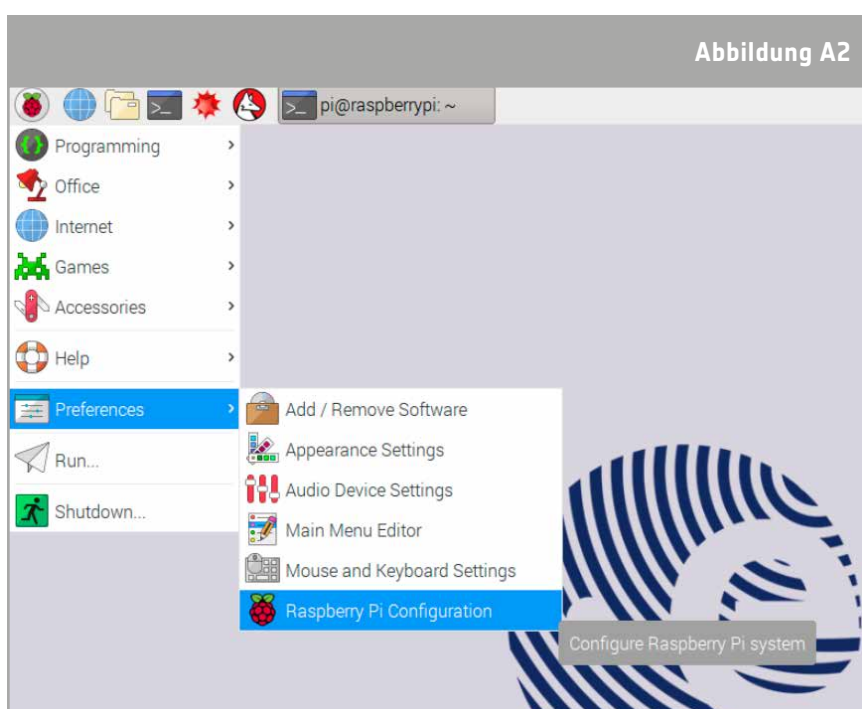
2. Schaltet den Raspberry Pi ein.
3. Wählt Main menu [Hauptmenü] → Preferences [Vorzugseinstellungen] → Raspberry Pi Configuration [Raspberry-Pi-Konfiguration]

↑ Die Abbildung zeigt den mit dem Flachbandkabel der Kamera verbundenen Raspberry Pi

4. Dann achtet im Interface-[Schnittstellen-]Bereich darauf, dass eure Kamera für den Computer aktiviert ist. Falls dies nicht der Fall ist, aktiviert sie und startet den Raspberry Pi neu.

Anmerkung: Zur Verwendung der Picamera-Bibliothek benötigt ihr eine aktualisierte Fassung von Rasbian. Wenn ihr sie nicht habt, müsst ihr sie in einem Terminalfenster installieren, zu dem ihr durch Anklicken des blau umrandeten Bildsymbols in Abbildung A2 gelangt. Der Befehl hierfür lautet: `sudo apt-get install python3-picamera`

Denkt daran, dass ihr hierfür eine Internet-Verbindung benötigt.



↑ Zugang zum Konfigurationsmenü des Raspberry Pi

→ AUFGABE 1: VERWENDUNG DER PI CAMERA

Die Astro-Pi-Kameras auf der ISS sind beides 5-Megapixel-Kameras, die Standbilder hoher Qualität liefern und viele Video-Aufnahmemodi unterstützen – ähnlich wie bei einem normalen Mobiltelefon.

Bei dieser Aufgabe erkundet ihr zunächst die grundlegenden Merkmale der Astro-Pi-Kameras.

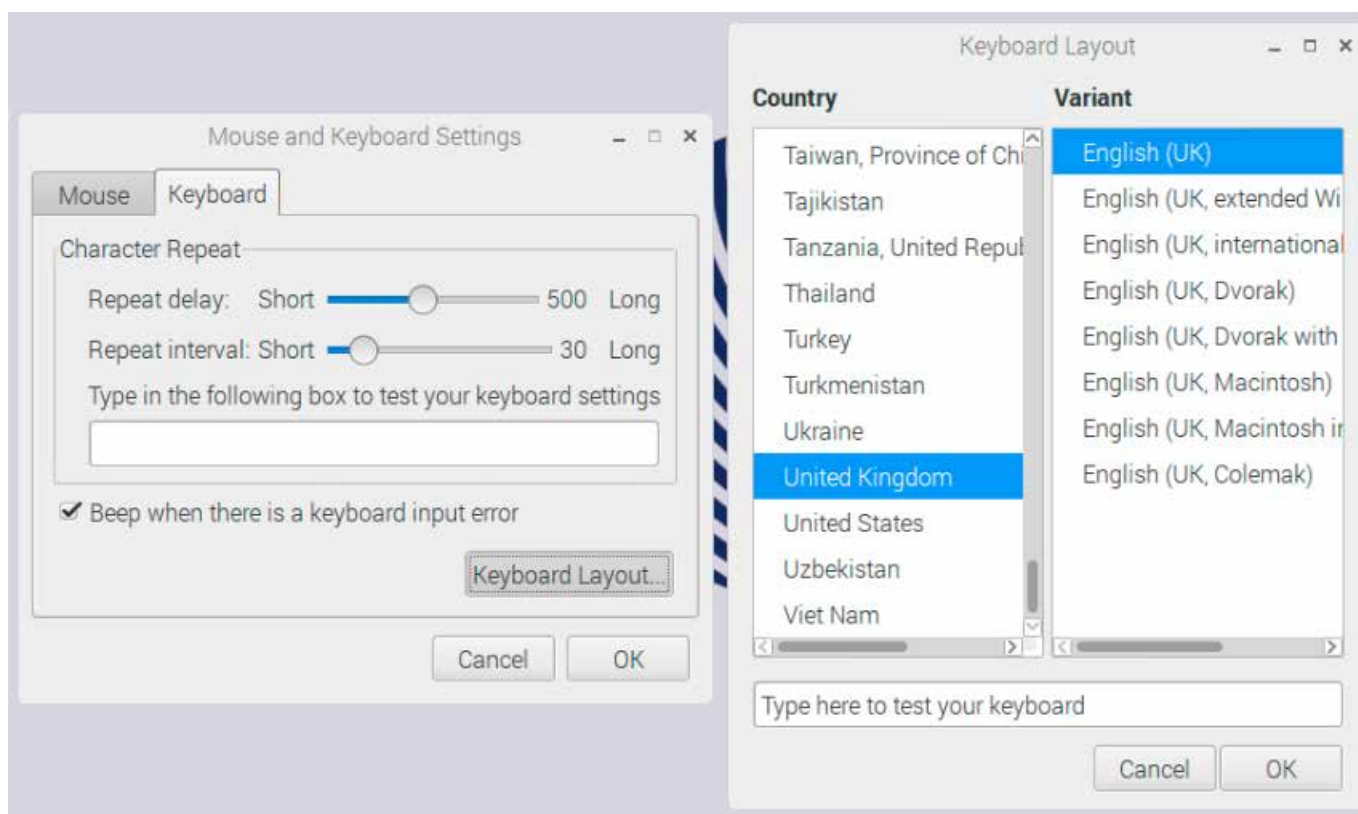
Aufgabe 1.1 – Bitte lächeln!

Macht zu Beginn dieser Aufgabe ein Bild von euch selbst.

Übung

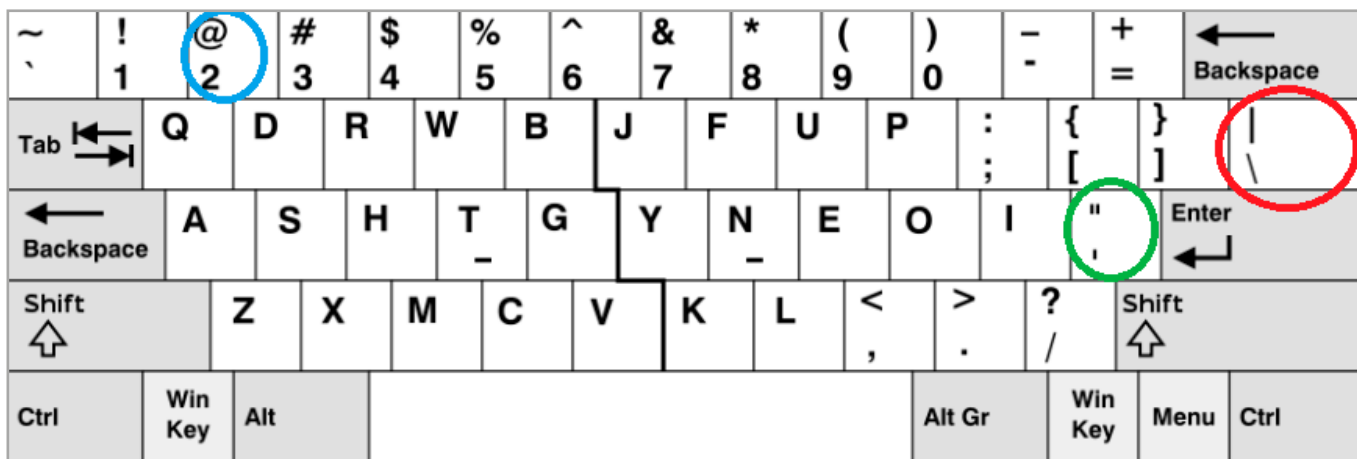
1. Öffnet Python 3 durch Anklicken des Raspberry-Logos am oberen Bildschirm. Dadurch öffnet sich das Menü. Wählt Programming [Programmieren] > Python 3 aus. Dadurch öffnet sich ein Python-Shell-Fenster.

Anmerkung: Möglicherweise müsst ihr auf dem Raspberry Pi die Tastatureinstellungen ändern, um wichtige Tasten wie ‘, “ und # darzustellen. Der Zugang zu den Tastatureinstellungen erfolgt über dasselbe Vorzugseinstellungen-Menü wie bei Aufgabe 0.



Die Anordnung der oben erwähnten Tasten richtet sich nach dem gewählten Layout. Beim Layout UK-English sind die wichtigen Tasten wie folgt zu finden:

- # die Taste für den umgekehrten Schrägstrich (\)
- “ Zifferntaste 2 mit Umschalten drücken
- , befindet sich neben Enter (Eingabe)



Daran denken: Bei Python wird das Zeichen # zur Darstellung von Kommentaren verwendet; alles hinter dem Zeichen # auf der Zeile wird als ein Teil des Codes ausgeführt. Sinnvollerweise verwendet ihr Kommentare zur Darstellung, was auf jeder Zeile passiert. So versteht ihr es besser, wenn ihr es wieder durchlest.

2. Wählt File > New File [Datei > Neue Datei] aus, und gebt den unten stehenden Code in das neue Fenster ein.

```
File Edit Format Run Options Windows Help
from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.start_preview()
sleep(10)
camera.stop_preview()
```

3. Sichert euren Code und führt ihn aus. Was passiert?

Anmerkung: Nennt eure Datei **nicht** 'picamera.py'. Wenn eure Kameravorschau nicht aufhört, könnt ihr die Alt+F4-Tastenkombination zum Schließen des Fensters benutzen.

4. Versucht die Kamera zu bewegen, und beobachtet, was auf dem Monitor passiert. Könnt ihr erläutern, was in jeder Zeile des Codes passiert?

```

File Edit Format Run Options Windows Help
from picamera import PiCamera
from time import sleep #Imports the camera and time libraries (1)

camera = PiCamera() #Defines the Pi Camera as a variable (2)
camera.start_preview() #Starts the camera preview (3)
sleep(10) #10 second delay (4)
camera.stop_preview() #Stops the camera preview (5)

```

↑ Die Kommentare (mit # gekennzeichnet) in diesem Code erklären, was in jeder Phase passiert, und sind nur für Menschen sinnvoll. Sie werden vom Computer ignoriert.

(1) #Importiert die Bibliotheken für Kamera und Zeit, (2) #Definiert die Pi-Kamera als eine Variable, (3) #Startet die Kameravorschau, (4) #Verzögerung von 10 Sekunden, (5) #Beendet die Kamera-Seitenansicht

5. Beide Kameras (Pi-Kamera und Pi-NoIR-Kamera) können Standbilder mit der Auflösung 2592 x 1944 and Videoaufnahmen mit 1920 x 1080 aufzeichnen, was hochauflösende Bilder und Videos ergibt. Zur Änderung der Auflösung können wir den Befehl `camera.resolution [Kameraauflösung]` benutzen. Um die Auflösung zu 1024 x 768 zu ändern, würde der Code so aussehen:

```

File Edit Format Run Options Windows Help
from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.resolution = (1024,768)
camera.start_preview()
sleep(5)
camera.stop_preview()

```

Versucht, die Mindestauflösung (64 x 64) einzustellen. Könnt ihr beide Bilder vergleichen? Gehen in dem Bild mit der geringeren Auflösung wichtige Einzelheiten verloren?

6. Versucht jetzt, die Auflösung an die Auflösung eures Smartphones anzupassen. Funktioniert das? Tipp: Im Internet könnt ihr die Auflösung eures Smartphones finden.

Daran denken: Der Astro-Pi-Bausatz enthält die Pi-Kamera V2 für das sichtbare Spektrum und die NoIR-Pi-Kamera V2, während Astro Pi Ed und Izzy mit V1 ausgerüstet sind! Ganz wichtig: Dies bedeutet, dass die die höchste Auflösung, die sie unterstützen können, unterschiedlich ausfällt. Um die Unterschiede herauszufinden, geht auf den nachstehenden Link:

www.raspberrypi.org/documentation/hardware/camera

Aufgabe 1.2 – Änderung der Kameraausrichtung

Die Ausrichtung der Astro-Pi-Kamera ist nach dem Einschalten möglicherweise noch nicht so wie erwartet. Das ist aber kein Problem, da ihr die Bilder und Videos speichern und anschließend drehen könnt. Allerdings könnte es zweckmäßiger sein, das Bild vor dem Speichern zu drehen. Dadurch entspricht die Seitenansicht, die ihr seht, genau der endgültigen Datei, die ihr speichert.

Übung

1. Versuchen wir, die Ausrichtung der Kamera zu ändern. Dreht die Kamera anhand des nachstehenden Codes:

```
File Edit Format Run Options Windows Help
from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.rotation = 180
camera.start_preview()
sleep(10)
camera.stop_preview()
```

2. Könnt ihr einen Code schreiben, durch den sich die Drehung der Kamera alle 3 Sekunden um 90° ändert?

Schreibt euren Code hier auf und probiert ihn auf der Astro-Pi-Kamera aus:

Es gibt viele Möglichkeiten, um das Problem zu lösen; eine Lösung steht unten:

```
File Edit Format Run Options Windows Help
from picamera import PiCamera
from time import sleep

x=0

camera = PiCamera()
camera.start_preview()
for x in [0,90,180,270]:
    camera.rotation = x
    sleep(3)
camera.stop_preview()
```


Aufgabe 1.3 – Bilder mit Text versehen

Die Hinzufügung von Text ist sinnvoll, um eure Bilder mit zusätzlichen Informationen zu versehen. Das könnte einfach nur das Datum oder die Uhrzeit sein, aber auch die Koordinaten der Astro-Pi-Kamera bei der Aufnahme des Fotos oder Videos beinhalten.

Übung

1. Text lässt sich der Seitenansicht mit nachstehender Zeile hinzufügen:

```
camera.annotate_text = 'Grüße von der Erde'
```

```
File Edit Format Run Options Windows Help
from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.start_preview()
camera.annotate_text = 'Greetings from Earth'
sleep(5)
camera.stop_preview()
```

(1) 'Grüße von der Erde'

Anmerkung: Zur Änderung der Textgröße könnt ihr den Befehl benutzen:

```
'camera.annotate_text_size = '
```

→ AUFGABE 2: STANDBILDER UND AUFNAHME VON VIDEOS

Jetzt wisst ihr zwar schon, wie ihr die Kamera verwenden müsst, um ein Bild zu zeigen, aber für die Astro-Pi-Mission müsst ihr Daten erfassen. Diese können Bilder, Protokolldateien oder CSV-Dateien mit einer Tabelle eurer Sensorwerte sein. In dieser Aufgabe lernt ihr, wie ihr ein Standbild und ein Video aufzeichnet.

Aufgabe 2.1 Macht ein Selfie!

Macht zunächst ein Bild von euch selbst.

Übung

1) Öffnet ein neues Python-3-Fenster und schreibt den unten stehenden Code.

```
File Edit Format Run Options Window Help
from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.start_preview() #Starts the preview
sleep(5)                #Runs for five seconds
camera.capture('/home/pi/Desktop/image.jpg') #Saves an image to the desktop named image.jpg
camera.stop_preview()
```

(1) #Startet die Seitenansicht, (2) #Läuft fünf Sekunden lang, (3) #Speichert ein Bild auf dem Desktop mit der Bezeichnung Bild.jpg

2) Wählt File > Save As [Datei > Speichern unter] aus und wählt eine Dateibezeichnung für euer Programm. Wählt dann Run > Run Module [Ausführen > Modul ausführen] aus. Ihr habt euer erstes Astro-Pi-Bild für euer Astro-Pi-Album gespeichert.

3) Jetzt wollen wir eine Zeitraffer-Aufnahme erstellen. Hierzu werden Standbilder über einen längeren Zeitraum aufgezeichnet, wobei zwischen jeder Bildaufzeichnung eine Verzögerung stattfindet. Zum Ausprobieren dieses Merkmals nehmen wir eine Reihe von fünf Bildern unter Verwendung eines For-Schleife-Befehls auf.

Ihr müsst eure Aufzeichnungen anhand eines Iterators benennen. Das bedeutet, dass jede Aufnahme, die ihr macht, unter einem anderen Namen gespeichert wird und wir die vorherigen Bilder nicht verlieren! Hierzu brauchen wir eine Codezeile wie die nachstehende:

```
camera.capture('/home/pi/Desktop/image%s.jpg' % i),
```

wo *i* ein Iterator ist. Dies bedeutet, dass jedes Mal während der Schleife sich die als *i* gespeicherte Zahl um 1 erhöht und sich die Zahl in der Dateibezeichnung auch um 1 erhöht.

```
File Edit Format Run Options Windows Help
from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.start_preview()
for i in range(5): #Begins the iteration
    sleep(5)
    camera.capture('/home/pi/Desktop/image%s.jpg' % i) #Saves a picture every five seconds
    #as image0.jpg, image1.jpg... etc.
camera.stop_preview()
```

(1) #Beginnt die Iteration, (2) #Speichert alle fünf Sekunden ein Bild, (3) #als Bild0.jpg, Bild1.jpg,... usw.

Jetzt könnt ihr manuell euren ersten Zeitraffer in der Astro Pi mit der Darstellung des ersten Fotos ansehen und durch Drücken der rechten Pfeiltaste auf eurer Tastatur die anderen Bilder als Diaschau ansehen.

Aufgabe 2.2 Achtung Aufnahme!

Die Aufnahme von Bildern ist eine hervorragende Möglichkeit, um die Kamera zur Erfassung von Daten zu verwenden. Eine andere Möglichkeit ist die Aufzeichnung eines Videos. Schauen wir uns jetzt an, wie wir die Kamera zur Erstellung eines Videos nutzen können.

Übung

1) Die unten stehende Zeile ist der Befehl zum Starten der Aufnahme:

```
camera.start_recording('path')
```

wo 'path' [Pfad] der Ort ist, an dem ihr das Video speichern wollt.

Versucht jetzt mit dem unten stehenden Code ein einfaches Video zu erzeugen:

Kümmert euch nicht um '.h264'; dies ist ein Hinweis auf die Dateiarart, unter der wir unsere Videos speichern werden, genau wie '.jpg' in unserem Standbild-Code!

```
File Edit Format Run Options Windows Help
from picamera import PiCamera
from time import sleep

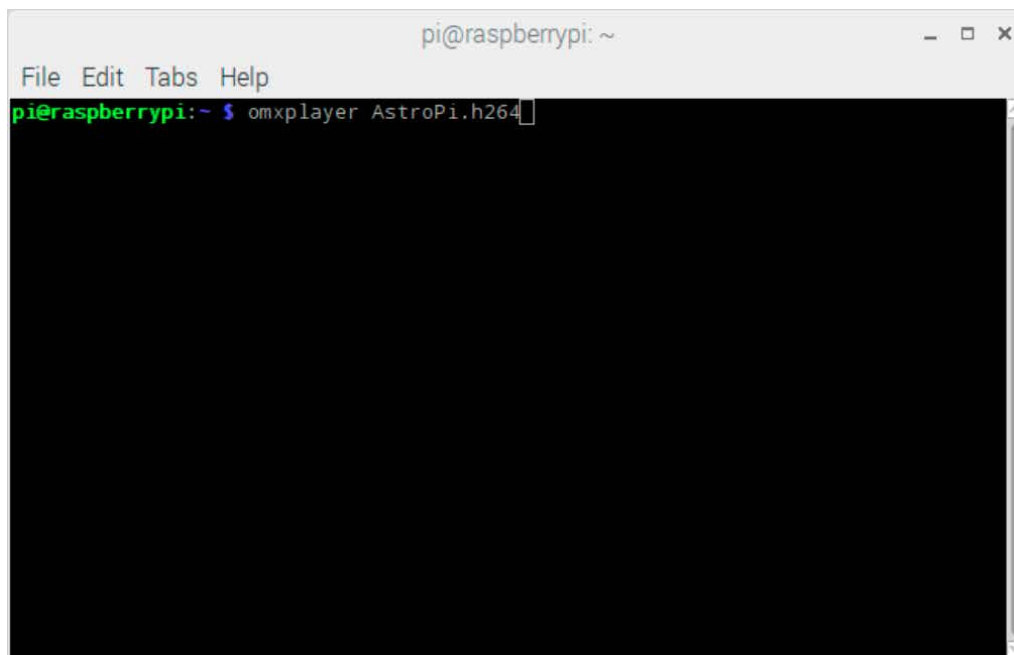
camera = PiCamera()
camera.start_preview()
camera.start_recording('/home/pi/Desktop/AstroPi.h264') #Begins recording a video named 'AstroPi' to the desktop (1)
camera.wait_recording(10) #Records for 10 seconds (2)
camera.stop_recording() #Stops recording the video (3)
camera.stop_preview()
```

(1) #Beginnt mit der Aufnahme eines Videos mit der Bezeichnung 'AstroPi' auf das Desktop, (2) #Nimmt 10 Sekunden lang auf, (3) #Beendet die Aufnahme des Videos

Um euer Video abzuspielen, müsst ihr ein Bildschirmfenster öffnen und den nachstehenden Befehl schreiben:

```
omxplayer AstroPi.h264
```

Der Bildschirm sieht dann wie folgt aus:



Das ESA-Astro-Pi-Team würde sich über die von euch erstellten Videos freuen! Vielleicht könntet ihr einen Gruß an einen Astronauten formulieren und uns zusenden?

Sendet eure Videos an astropi@esa.int !

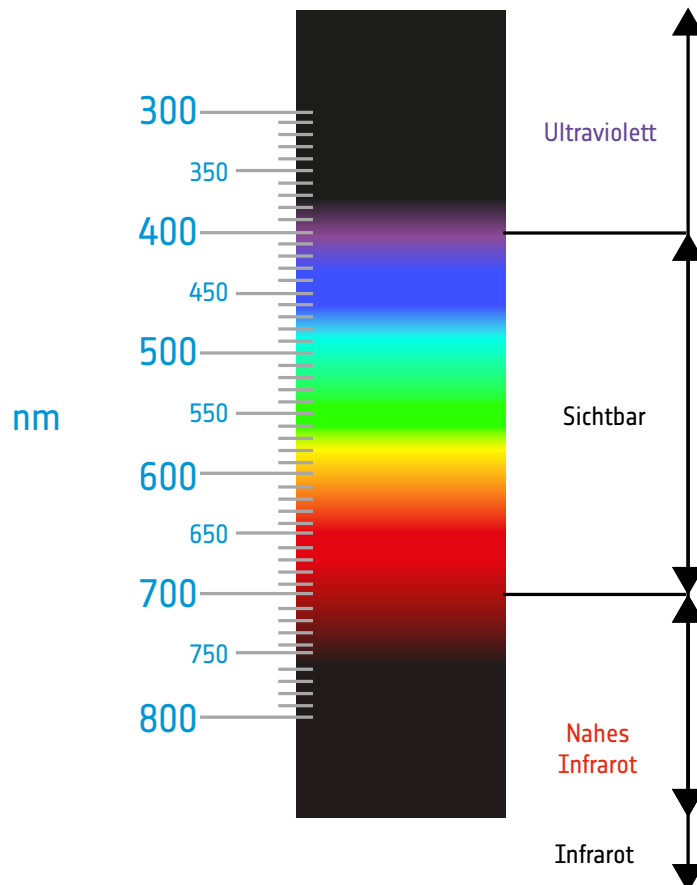
2) Diskutiert die Vor- und Nachteile von Standbildern und Video-Aufnahmen. Wenn ihr die Kamera von der ISS auf die Erde richten würdet, was wären die Vorteile von Stand- bzw. Videobildern?

Geht auf die unten stehenden Links um euch einige Fotos und Zeitrafferaufnahmen anzuschauen, die der ESA-Astronaut Paolo Nespoli mit seiner Profikamera an Bord der ISS von der Erde machte!

[\[Fotos des ESA-Astronauten Paolo Nespoli anlässlich der VITA-Mission auf der Internationalen Weltraumstation VITA-Mission Ausgabe 'Timelapse a Day' \(Zeitraffer eines Tages\) – von Kalifornien bis Mexiko\]](#)

→ AUFGABE 3: IN NAHES INFRAROT HINEINSEHEN

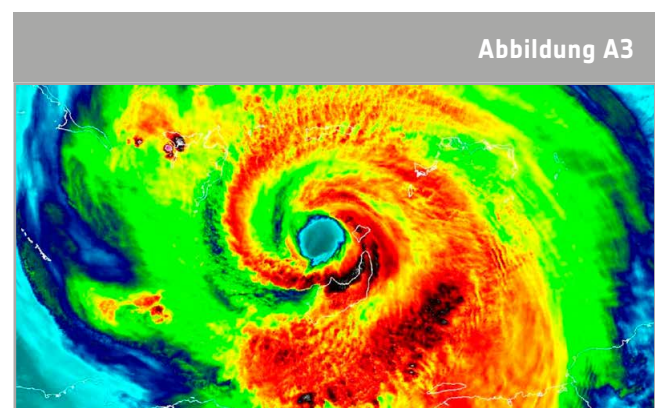
Nahes Infrarot ist ein Teil des elektromagnetischen Spektrums. Das elektromagnetische Spektrum verwenden wir zur Klassifizierung der Wellen und Phänomene, die uns in unserem Alltag umgeben. Der Teil des elektromagnetischen Spektrums, mit dem ihr am besten vertraut seid, ist das sichtbare Spektrum. Das ist das Licht und die Farben des Regenbogens. Nahes Infrarot ist eine andere, für Menschen unsichtbare Wellenart, genauso wie Mikrowellen oder Röntgenstrahlen. Eine TV-Fernbedienung blitzt in nahem Infrarot, wenn ein Knopf gedrückt wird, ohne dass wir es sehen können – doch die NoIR-Kamera kann es sehen!



Viele Kameras sind in der Lage, das Licht des nahen Infrarots zu sehen. Um jedoch eine Farbverfälschung zu vermeiden, verfügen Mobiltelefone und andere kleine Kameras normalerweise über ein Infrarotfilter, durch das das sichtbare Licht in die Kamera einfallen kann. Bei der NoIR-Kamera wurde dieses Filter entfernt, so dass sie für diesen Teil des IR-Spektrums, das sich in der Nähe des sichtbaren roten Lichts befindet, empfindlich ist. Wir wollen das ausprobieren.

Was eine Kamera für nahes Infrarot nicht sehen kann

Wichtig ist, zwischen einer Kamera für nahes Infrarot und einer solchen für thermisches Infrarot zu unterscheiden. Ihr seid wahrscheinlich mit den wunderbaren farbigen Bildern vertraut, die Temperaturunterschiede zeigen, wenn ihr an eine Infrarotkamera denkt. Leider sieht eine Kamera für nahes Infrarot den thermischen Teil des Spektrums nicht, so dass ihr keine Bilder wie dieses NASA-Bild erzeugen könnt, mit dessen Hilfe die Hotspots des Hurrikans nachverfolgt wurden.



↑ Hurrikan Irma.

Übung

1) Drückt die Knöpfe einer TV-Fernbedienung – könnt ihr etwas beim Emitter am oberen Teil der Fernbedienung feststellen?

2) Schreibt ein Programm in Python, das 15 Sekunden lang eine Seitenansicht zeigt. Schaltet das Zimmerlicht aus und richtet die TV-Fernbedienung auf die Kamera und drückt irgendeinen Knopf auf der Fernbedienung. Was passiert?

3) Kann man das bei Tageslicht sehen? Wodurch könnten die Unterschiede bedingt sein, die ihr feststellt?

4) Könnt ihr die Objekte am Nachthimmel ermitteln, die Quellen für das Licht des nahen Infrarots sind? Welche Probleme könnten sich bei dem Versuch ergeben, diese Objekte mit einer NoIR-Kamera anzusehen?

Wusstet ihr schon...



Astronomen müssen sich viel Mühe geben, um die Bilder zu erfassen, die sie für ihre Forschung benötigen! Viele Observatorien, wie dieses auf Mauna Kea, befinden sich auf hohen Berggipfeln. Die Strahlung des nahen Infrarots wird durch Wasserdampf absorbiert, so dass Teleskope für nahes Infrarot über den Wolken aufgestellt sind, um ein klares Bild zu erhalten.

Weiterführende Aufgabe – Verwendung der NoIR-Kamera zur Überwachung des Pflanzenwachstums

Die an Satelliten mitgeführten Sensoren können auch andere Teile des elektromagnetischen Spektrums ‚sehen‘. Weltraumwissenschaftler untersuchen die Strahlung des nahen Infrarots, um die Vegetation der Erde aus dem Weltraum zu überwachen. Pflanzen reflektieren stark das Licht des nahen Infrarots, und gesunde Pflanzen reflektieren stärker als geschwächte Pflanzen. Gesunde Pflanzen reflektieren einen Großteil dieses Lichts, da sie es nicht für die Fotosynthese benötigen. Genau so wie wir atmen, um zu überleben, benutzen die Pflanzen die Fotosynthese, um am Leben zu bleiben. Sie verarbeiten das Licht zu Sauerstoff und Kohlenhydraten. Bei dieser Reaktion ist das Chlorophyll sehr wichtig. Es absorbiert langwelliges (rotes) und kurzwelliges (blaues) Licht. Grünes Licht wird nicht absorbiert, so dass uns die Blätter einer Pflanze grün erscheinen.

Wusstet ihr schon?



Sentinel-3 ist eine Meer- und Land-Mission, die aus drei vielseitigen Satelliten besteht (Sentinel-3A, Sentinel-3B und Sentinel-3C). Das Meer- und Land-Farbinstrument von Sentinel-3A kann das Chlorophyll in der Vegetation ‚sehen‘. Die Menge des Chlorophylls in den Pflanzen spielt eine wichtige Rolle bei der Bestimmung von deren Gesundheitszustand. Die Überwachung des Chlorophylls aus dem Weltraum bietet ein wertvolles Werkzeug zur Unterstützung von Vegetationsmanagement, Land- und Forstwirtschaft.

Eure Aufgabe besteht darin herauszufinden, wie ihr die NoIR-Kamera zur Untersuchung der Gesundheit von Pflanzen verwenden könnt.

Hierfür könnt ihr das letzte Stück aus dem Versuchsmaterial des ESA-Astro-Pi-Bausatzes verwenden: das kleine blaue Quadrat. Es ist ein blaues Filter (deshalb so genannt, weil es unseren Augen blau vorkommt); es sperrt grünes und rotes Licht und lässt blaues und nahes Infrarot-Licht durch. Das Datenblatt zu dem Blaufilter findet ihr [hier](#).

Lasst euch bei der Untersuchung von nachstehenden Fragen leiten!

1) Welcher Teil des Blattes ist hauptsächlich für die Farbe der Pflanzen verantwortlich?

2) Wie unterscheidet sich dies zwischen gesunden und kranken Pflanzen?

3) Erläutert die festgestellten Unterschiede zwischen gesunden und kranken Pflanzen unter Verwendung der NoIR-Kamera und des Blaufilters.

→ LINKS

Astro-Pi-Ausbildungs-Webinar:

astro-pi.org/updates/mission-space-lab-training-webinar

Astro Pi GitHub:

www.raspberrypi.org/learning/astro-pi-guide

Von STEM UK erstellte Astro-Pi-Ressourcen:

www.stem.org.uk/elibrary/collection/4204

Informationen zu den Astro-Pi-Kameras:

www.raspberrypi.org/documentation/hardware/camera

Von Paolo Nespoli aufgenommene Fotos:

www.flickr.com/photos/astro_paolo

Informationen zu dem Blaufilter:

www.raspberrypi.org/blog/whats-that-blue-thing-doing-here

Von der Raspberry Pi Foundation erstellte Astro-Pi-Ressourcen

projects.raspberrypi.org/en/projects/getting-started-with-picamera

teach with space – Umgang mit den Astro-Pi-Kameras | T05.4
www.esa.int/education

**Das ESA Education Office (ESA-Bildungsbüro) freut sich über Rückmeldungen
und Kommentare**
astropi@esa.int

**Eine Produktion von ESA Education
in Zusammenarbeit mit ESERO Polen und ESERO Niederlande**

Copyright 2018 © European Space Agency