



Lernziele:

- Können einen Sensor in Betrieb nehmen.
- Können Programme, welche Sensoren einsetzen, analysieren und nachbauen.
- Können einen Programtablaufplan interpretieren.

Nun habt ihr schon die ersten Programme für den Raspberry Pi geschrieben und die Kamera gesteuert. Jetzt wollen den ersten Sensor einsetzen (Englisch «sense» heisst spüren). Wir verbinden eine Taste mit der Kamera.

Ihr werdet mit Hilfe des Buttons Fotos machen und Videoaufnahmen starten, beziehungsweise stoppen.

Bis jetzt haben wir ein Programm gestartet, das hat ein Foto oder ein Video gemacht. Danach war das Programm fertig und hat gestoppt.

Wenn wir jetzt Sensoren und später Aktoren einsetzen, darf das Programm nicht nach einmal durchlaufen stoppen. Sehr wahrscheinlich wurde dann noch gar kein Sensor betätigt.

Wenn wir mit Sensoren & Aktoren arbeiten, darf das Programm nicht stoppen es muss «endlos» testen, ob ein Sensor aktiviert wird. Eigentlich kennt ihr das schon von unseren Computern oder Handys. Die Betriebssysteme laufen in einer «Endlosschleife» und warten bis ein Benutzer oder ein Sensor etwas tut.

Aufgabe 1 (Foto machen wenn ein Knopf gedrückt wird)

Analysiert zusammen folgenden Code. Nach der Analyse gebt ihr den Code ein und testet ihn. Die Kommentare, das ist der Text hinter dem «#», müsst ihr nicht eingeben. Das sind nur Erklärungen für euch.

```
1 #!/usr/bin/python3 # mit dieser Anweisung kann das Programm mit einem Doppelklick gestartet werden
2
3 # An dieser Stelle werden die nötigen Programmteile geladen
4 # picamera – damit Python mit der Kamera das Raspberry Pi Kontakt aufnehmen kann
5 # grovepi – damit hat Python zugriff auf den Grovepi+
6 from picamera import PiCamera
7 import grovepi
8
9 # Der Button ist auf dem Port 3 (digitales Port) eingesteckt
10 button = 3
11
12 # Dem Grovepi wird mitgeteilt, dass am Port 2 (button) ein Sensor (INPUT) angeschlossen ist
13 grovepi.pinMode(button, "INPUT")
14
15 # Python mit der Kamera verbinden. Die Kamera hat den Namen camera
16 camera = PiCamera()
17
18 # Damit die Photos nicht immer überschrieben werden, muss der Name der Fotos eindeutig sein
19 # das können wir sicherstellen, durch dass dem Name des Fotos eine eindeutige Nummer angehängt wird.
20 counter = 0
21
22 # Das ist jetzt die Endlosschleife
23 while True:
24
25     # Da auch mal etwas schief gehen kann sagen wir Python versuche mal folgende Anweisungen
26     try:
27
28         # Lese vom digitalen Port den Wert des Button ein. Gedrückt ist 1 sonst 0
29         button_status = grovepi.digitalRead(button)
30
31         # Wenn der Button gedrückt ist
32         if button_status:
33
34             # Mache ein Foto und speichere es, hänge die aktuelle Nummer an
35             camera.capture('/home/pi/Desktop/image_' + str(counter) + '.jpg')
36
37             # Erhöhe die Nummer um 1, damit das nächste Foto eine andere Nummer hat
38             counter = counter + 1
39
40     # Abbruch durch Control-C
41     except KeyboardInterrupt:
42         break
43
44     # Wenn sonst eine Fehler auftritt
45     except (IOError, TypeError) as e:
46         print("Error", e)
```



Aufgabe 2 (Videoaufnahme mit einem Knopf steuern)

Analysiert zusammen folgenden Code. Nach der Analyse gebt ihr den Code ein und testet ihn. Die Kommentare, das ist der Text hinter dem «#», müsst ihr nicht eingeben. Das sind nur Erklärungen für euch.

```
1  #!/usr/bin/python3 # mit dieser Anweisung kann das Programm mit einem Doppelklick gestartet werden
2
3  # An dieser Stelle werden die nötigen Programmteile geladen
4  # picamera – damit Python mit der Kamera das Raspberry Pi Kontakt aufnehmen kann
5  # time – damit eine Pause von 1 Sekunden eingebaut werden kann
6  # grovepi – damit hat Python zugriff auf den Grovepi+
7  from picamera import PiCamera
8  from time import sleep
9  import grovepi
10
11 # Der Button ist auf dem Port 3 (digitales Port) eingesteckt
12 button = 3
13
14 # Dem Grovepi wird mitgeteilt, dass am Port 2 (button) ein Sensor (INPUT) angeschlossen ist
15 grovepi.pinMode(button, "INPUT")
16
17 # Python mit der Kamera verbinden. Die Kamera hat den Namen camera
18 camera = PiCamera()
19
20 # Damit die Videos nicht immer überschrieben werden (analog, wie bei den Fotos)
21 counter = 0
22
23 # Es ist kein Video gestartet
24 videoOff = True
25
26 # Das ist jetzt die Endlosschleife
27 while True:
28
29     # Da auch mal etwas schief gehen kann sagen wir Python versuche mal folgende Anweisungen
30     try:
31
32         # Lese vom digitalen Port den Wert des Button ein. Gedrückt ist 1 sonst 0
33         button_status = grovepi.digitalRead(button)
34
35         # Wenn der Button gedrückt ist
36         if button_status:
37
38             # Wenn keine Videoaufnahme gestartet ist
39             if videoOff:
40
41                 # Mache ein Video und speichere es, hänge die aktuelle Nummer an
42                 camera.start_recording("/home/pi/Desktop/video_ " + str(counter)+ ".h264")
43
44                 # Die Videoaufnahme läuft
45                 videoOff = False;
46
47                 # Erhöhe die Nummer um 1, damit das nächste Video eine andere Nummer hat
48                 counter = counter + 1
49                 print("Video gestartet")
50
51                 # Paus, sonst geht stoppt das Video sofort wieder.
52                 sleep(1)
53
54             # Wenn eine Vidoaufnahme gestartet ist
55             else:
56
57                 # Die Aufzeichnung anhalten
58                 camera.stop_recording()
59
60                 # Die Videoaufnahme wurde angehalten
61                 videoOff = True
62                 print("Video angehalten")
63
64                 # Pause. sonst startet die Videoaufnahme sofort wieder.
65                 sleep(1)
66
67     # Abbruch durch Control-C
68     except KeyboardInterrupt:
69         break
70
71     # Wenn sonst eine Fehler auftritt
72     except (IOError,TypeError) as e:
73         print("Error", e)
```



So, nun habt ihr basierend auf Eingaben eines Sensors, Fotos und Videos gemacht. An und für sich tun die Programme nicht viel. Der Hauptteil der Programme sind Kommentare. Kommentieren ist sehr empfehlenswert. So kann auch Monate später nachgelesen werden, was für Gedankengänge zum vorliegenden Programmcode geführt haben.

Rekapitulieren wir, was in der Aufgabe 2 & 3 genau gemacht wurde.

Fangen wir mit der Aufgabe 1 an

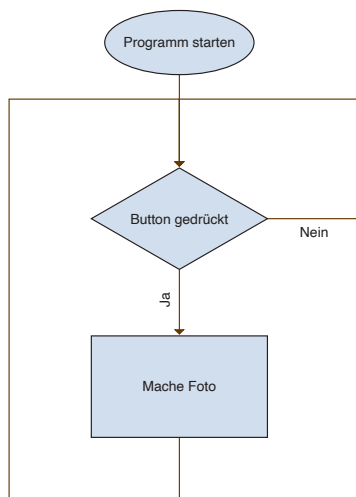
1. Dem Programm wird gesagt, auf welchem Port der Sensor, der Button eingesteckt ist.
2. Danach wird eine Endlosschleife gestartet.
3. In der Schleife wird getestet, ob der Button gedrückt ist
 - Falls der Button gedrückt ist, wird ein Foto gemacht
 - Sonst passiert nichts

Das ist alles.

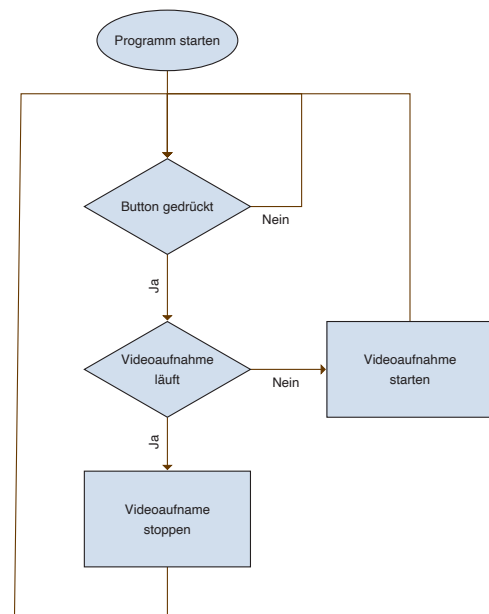
Gehen wir zur Aufgabe 2

1. Dem Programm wird gesagt, auf welchem Port der Sensor, der Button eingesteckt ist.
2. Dem Programm wird gesagt, dass momentan kein Video aufgenommen wird
3. Danach wird eine Endlosschleife gestartet.
4. In der Schleife wird getestet, ob der Button gedrückt ist
 - Falls der Button gedrückt ist
 - Falls momentan kein Video aufgenommen wird, starte die Videoaufnahme
 - Sonst, stoppe die Videoaufnahme
 - Sonst passiert nichts

Diesen Ablauf kann man auch mit einem Programmblaufplan visualisieren.



Programmblaufplan für Aufgabe 1



Programmblaufplan für Aufgabe 2

Alle weiteren Aufgaben mit Sensoren und Aktoren werden nach dem gleichen Schema ablaufen.

- Ports für Sensoren und Aktoren festlegen
- Endlosschleife starten
- Je nach Status der Sensoren etwas machen.