



Lernziele:

- Können zwei Aktoren im Programm integrieren (LED, Summer)
- Können nachvollziehen, dass mit `write` Werte auf Ports geschrieben werden.
- Können nachvollziehen, dass mit `read` Werte von Ports gelesen werden können.

Nachdem nun ein vereinfachtes Schema des Ablaufs von Programmen mit Sensoren und Aktoren bekannt ist, noch ein letztes Beispiel, bevor ihr euch selber mit Kreativität und Fantasie ans Werk machen könnt. Ihr werdet eine Zusammenstellung von Programmcodes vorfinden, in dem für jeden Sensor und Aktor des Grovepi+ beispielhaft gezeigt wird, wie der Sensor bzw. der Aktor verwendet werden kann.

Ihr werdet selber ein Projekt definieren, planen und realisieren können indem ihr beliebige Aufgabenstellungen mit Hilfe des Raspberry Pi und des Grovepi und deren Möglichkeiten löst.

Doch nun noch zu einem Beispiel indem wir einen Sensor und zwei Aktoren einsetzen. Der Sensor ist wieder der Button, die Aktoren sind ein Summer (buzzer) und ein LED (Light emitting diode).

Die Diode soll leuchten und der Summer soll summen solange der Button gedrückt ist.

Aufgabe 1 (Sensoren und Aktoren)

Schreibt ein Programm, welches ein LED auf Port 4 leuchten lässt und einen Summer auf Port 8 summen lässt solange der Button auf Port 3 gedrückt ist.

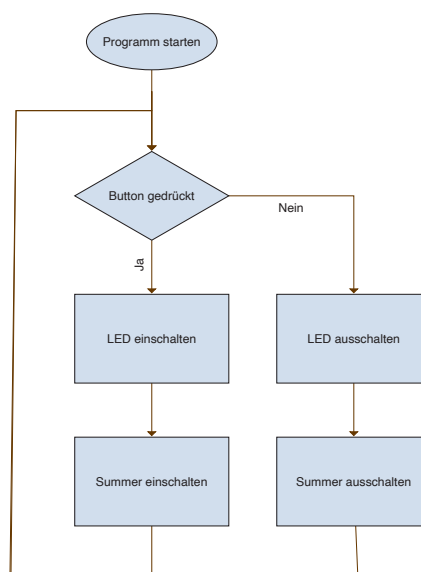
Bereitet den Raspberry Pi und den Grovepi vor, bevor ihr den Raspberry Pi mit Strom versorgt.

Versucht das Programm selber zu schreiben, ohne die Lösung auf dem nächsten Blatt anzuschauen. Das Programm ist viel kürzer als das Programm mit der Videosteuerung.

Hier zwei Tipps:

- Wenn von einem digitalen Port gelesen werden soll, dann beginnt die oft Anweisung mit
 - `digitalRead(Name des Ports)`
- Wenn auf ein digitales Port etwas geschrieben werden soll, beginnt die Anweisung oft mit
 - `digitalWrite(Name des Ports, Wert der geschrieben werden soll meist 0 oder 1)`

Hier noch der PAP (Programmablaufplan) für die Aufgabe.





Eine mögliche Lösung für die Aufgabe 1.

Eine mögliche Lösung darum, weil auch andere Lösungen denkbar wären. Wenn das Programm funktioniert, ist die Lösung richtig. Bei der Programmierung gibt es selten eine «Musterlösung». Es gibt verschiedene Vorgehensweisen, welche, wenn sie zum Ziel führen, korrekt sind.

Mit viel Wissen über die Hardware gibt es optimale und weniger optimale Lösungen was die Anzahl Anweisungen betrifft. Oft spielt allerdings die Geschwindigkeit keine grosse Rolle und Speicherplatz steht auch fast ohne Ende zur Verfügung.

Darum, wenn eure Lösung von der Lösung auf diesem Blatt abweicht, aber trotzdem funktioniert, habt ihr alles richtig gemacht.

```
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 # grovepi - damit hat Python zugriff auf den Grovepi+
5 from grovepi import *
6
7 # Der Button wird auf Port 3 (Digitalport) erwartet
8 button = 3 # Port for Button
9
10 # Das LED wird auf Port 4 (Digitalport) erwartet
11 led = 4
12
13 # Der Buzzer wird auf Port 8 (Digitalport) erwartet
14 buzzer = 8 # Port for buzzer
15
16 # Dem Grovepi wird mitgeteilt, dass am Port 3 (button) ein Sensor (INPUT) angeschlossen ist
17 pinMode(button,"INPUT") # Assign mode for Button as input
18
19 # Dem Grovepi wird mitgeteilt, dass am Port 4 (LED) ein Aktor (OUTPUT) angeschlossen ist
20 pinMode(led,"OUTPUT")
21
22 # Dem Grovepi wird mitgeteilt, dass am Port 8 (buzzer) ein Aktor (OUTPUT) angeschlossen ist
23 pinMode(buzzer,"OUTPUT")
24
25 # Das ist jetzt die Endlosschleife
26 while True:
27
28     # Da auch mal etwas schief gehen kann sagen wir Python versuche mal folgende Anweisungen
29     try:
30
31         # Lese vom digitalen Port den Wert des Button ein. Gedrückt ist 1 sonst 0
32         button_status = digitalRead(button)
33
34         # Wenn der Button gedrückt ist
35         if button_status:
36
37             # Schreibe eine 1 auf das Port an dem der Buzzer angeschlossen ist (1=ein, 0=aus)
38             digitalWrite(buzzer,1)
39
40             # Schreibe eine 1 auf das Port an dem LED angeschlossen ist (1=ein, 0=aus)
41             digitalWrite(led, 1)
42
43         else:
44             digitalWrite(buzzer, 0)
45             digitalWrite(led, 0)
46
47     # Abbruch durch Control-C
48     except KeyboardInterrupt: # Stop the buzzer and LED before stopping
49         digitalWrite(buzzer,0)
50         digitalWrite(led, 0)
51         break
52
53     except (IOError,TypeError) as e:
54         print("Error")
```



Jetzt habt ihr das Rüstzeug um euch eigene kreative, interessante Aufgabenstellungen auszudenken. Wenn ihr nicht wisst, wie ein konkreter Sensor gelesen oder eine Aktor geschrieben wird, schaut in der Übersicht nach. Dort findet ihr Beispielcode für die Sensoren und Aktoren des Grovepi+.

Viel Spass beim Suchen einer interessanten Fragestellung. Werdet zu Erfindern. Wir sind gespannt auf eure Projekte.

Wenn die Projekte abgeschlossen sind, präsentiert und demonstriert eure Erfindungen vor der Klasse.