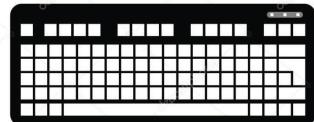


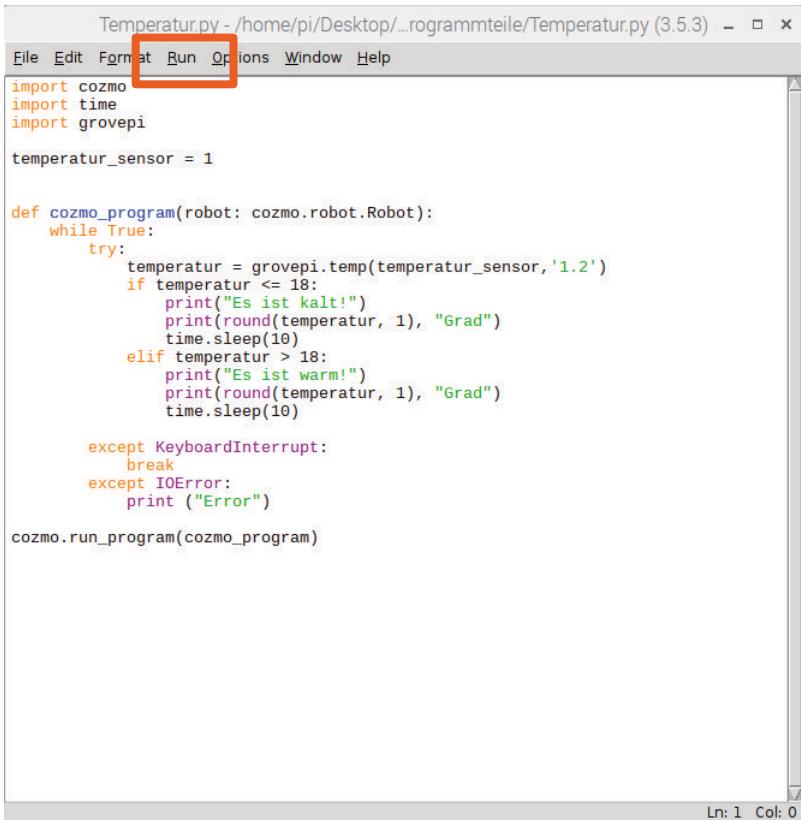
# Anschluss des Raspberry Pis



Verbinde den Raspberry Pi mit einem Monitor, einer Tastatur und einer Maus, damit du ihn programmieren kannst.

Zusätzlich muss du noch den Raspberry Pi mit Strom versorgen.

# Starten des Programms auf den RaspberryPi



```
Temperatur.py - /home/pi/Desktop/...rogrammteile/Temperatur.py (3.5.3) - □ ×
File Edit Format Run Options Window Help
import cozmo
import time
import grovepi

temperatur_sensor = 1

def cozmo_program(robot: cozmo.robot.Robot):
    while True:
        try:
            temperatur = grovepi.temp(temperatur_sensor, '1.2')
            if temperatur <= 18:
                print("Es ist kalt!")
                print(round(temperatur, 1), "Grad")
                time.sleep(10)
            elif temperatur > 18:
                print("Es ist warm!")
                print(round(temperatur, 1), "Grad")
                time.sleep(10)

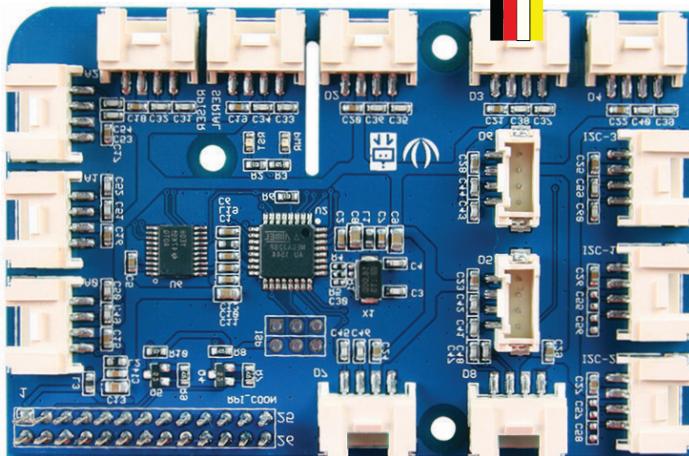
        except KeyboardInterrupt:
            break
        except IOError:
            print ("Error")

cozmo.run_program(cozmo_program)
```

Ln: 1 Col: 0

Das Programm startest du,  
indem du unter „Run“ auf „Run  
Module“ gehst, oder indem du  
auf F5 auf der Tastatur drückst.

# Anschluss eines Ultraschallsensors



Der Ultraschallsensor kann durch Ultraschallwellen den Abstand zu einem Gegenstand oder zum Beispiel deiner Hand erkennen!

Schließe diesen an einen digitalen Pin an, zum Beispiel an D3.



# Nutzung des Ultraschallsensors

## 1. Benötigte Pakete:

```
import grovepi
import time
```

## 2. Initialisierung:

```
# an Pin D3 angeschlossen
ultrasonic_ranger = 3
```

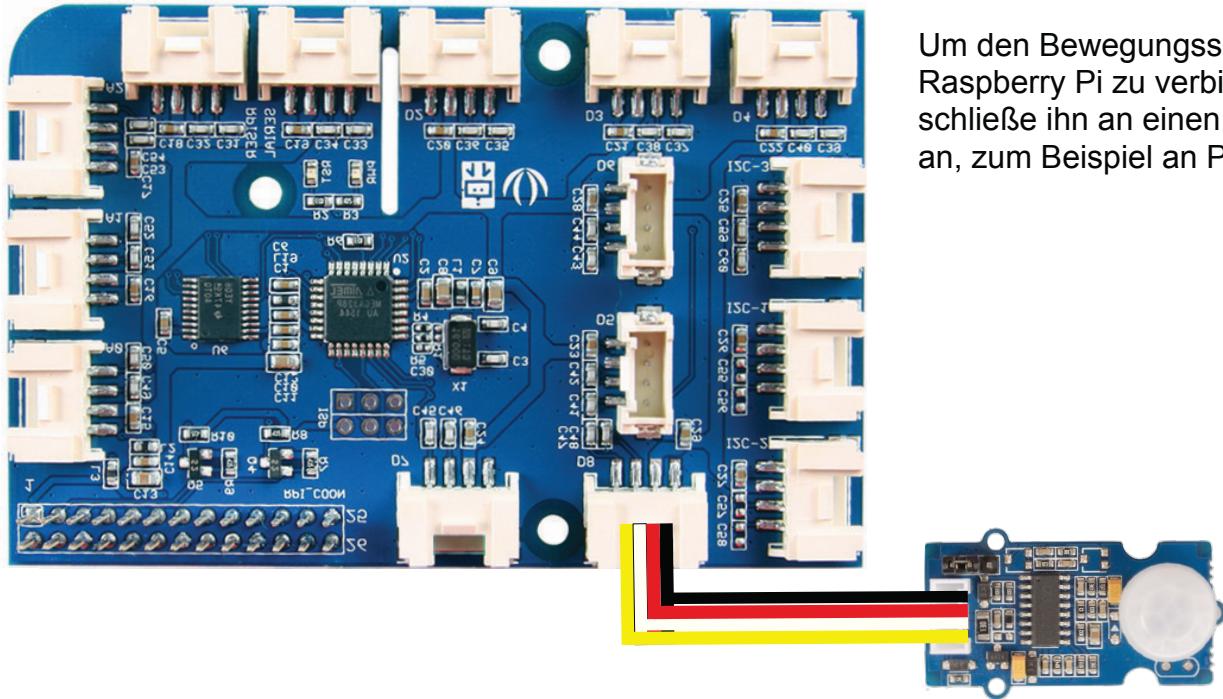
## 3. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):
    while True:
        try:
            # wenn der Abstand größer, gleich 20 cm ist, dann...
            if grovepi.ultrasonicRead(ultrasonic_ranger) >= 20:
                print("-")

            # wenn der Abstand kleiner 20 cm ist, dann...
            elif grovepi.ultrasonicRead(ultrasonic_ranger) < 20:
                #Hier kann eine oder mehrere Aktionen rein
                time.sleep(5)

cozmo.run_program(cozmo_program)
```

# Anschluss eines Bewegungssensors



Um den Bewegungssensor mit dem Raspberry Pi zu verbinden, schließe ihn an einen digitalen Pin an, zum Beispiel an Pin D8.

Der Bewegungssensor kann erkennen ob sich etwas im Umkreis von 0,1 - 6 Metern bewegt.

# Nutzung des Bewegungssensors

## 1. Benötigte Pakete:

```
import cozmo
import time
import grovepi
```

## 2. Initialisierung:

```
# der Bewegungssensor ist an Pin D8 angeschlossen
motion_sensor = 8

# eine Variable, in der du einen Wert speichern kannst
motion = 0

# Angabe, dass der Sensor einen INPUT liefert
grovepi.pinMode(motion_sensor, "INPUT")
```

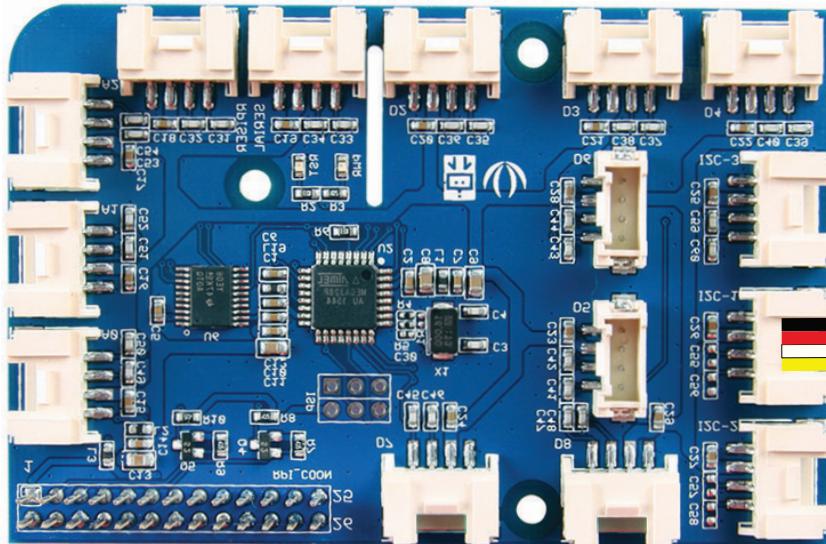
## 3. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):
    while True:
        try:
            motion = grovepi.digitalRead(motion_sensor)
            #überprüft, ob eine Bewegung wahrgenommen wurde mit 0 oder 1
            if motion == 0 or motion == 1:
                #wenn Bewegung wahrgenommen (1), dann...
                if motion == 1:
                    print("Eine unbekannte Bewegung wurde erkannt")
                    #Hier kann eine Aktion oder mehrere rein.
                #ansonsten
            else:
                print("Nicht erkannt")
                #Hier kann eine Aktion oder mehrere rein.
            time.sleep(0.2)

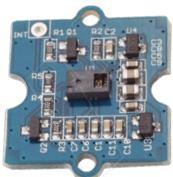
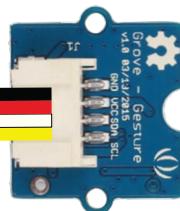
        except IOError:
            print("IOError")

cozmo.run_program(cozmo_program)
```

# Anschluss des Gestensors



Um den Gestensor anzuschließen, nutze eine I2C Büchse.



Damit der Gestensor die Geste richtig erkennen kann, muss der Sensor so gehalten werden, sodass die Beschriftung INT wie in der Abbildung oben links sein muss.  
Der Gestensor erkennt 9 verschiedene Gesten:  
Hoch, Runter, Rechts, Links, Vorwärts, Rückwärts, im Uhrzeigersinn, gegen den Uhrzeigersinn und Welle.

# Nutzung des Gestensors

## 1. Benötigte Pakete:

```
import cozmo
import time
import grovepi
import grove_gesture_sensor
```

## 2. Initialisierung:

```
g=grove_gesture_sensor.gesture()
g.init()
```

## 3. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):
    while True:
        gest=g.return_gesture()
        #Überprüft, ob die erkannte Geste Vorwärts ist
        if gest==g.FORWARD:
            #Hier kann eine oder mehrere Aktionen rein
            time.sleep(5)
        #Überprüft, ob die erkannte Geste Rückwärts ist
        elif gest==g.BACKWARD:
            #Hier kann eine oder mehrere Aktionen rein
            time.sleep(5)
        #Überprüft, ob keine Geste erkannt wurde
        elif gest==0:
            print("Keine Geste erkannt")
        else:
            print("error")
        time.sleep(.1)

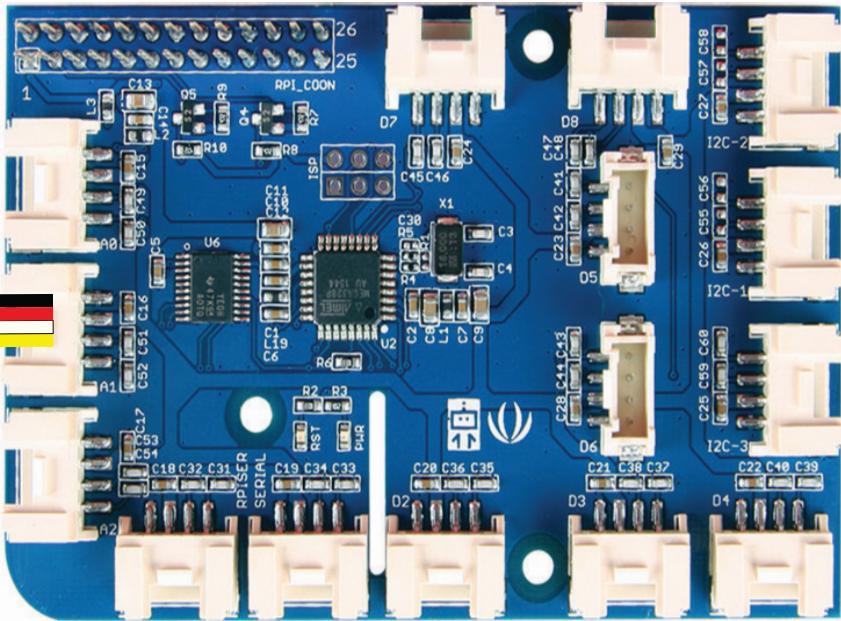
cozmo.run_program(cozmo_program)
```

Weiter Gesten: RIGHT, LEFT, UP, DOWN, CLOCKWISE, ANTI\_CLOCKWISE, WAVE

# Anschluss des Lichtsensors



Schließe den Lichtsensor an einen analogen Pin an, z.B. an A1.



Die Lichtstärke wird in Werten zwischen 0 und 800 Lumen gemessen, wobei 0 dunkel und 800 sehr hell entspricht.

# Nutzung des Lichtsensors

## 1. Benötigte Pakete:

```
import cozmo  
import time  
import grovepi
```

## 2. Initialisierung:

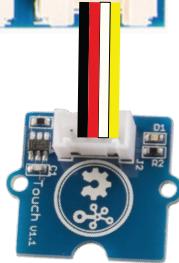
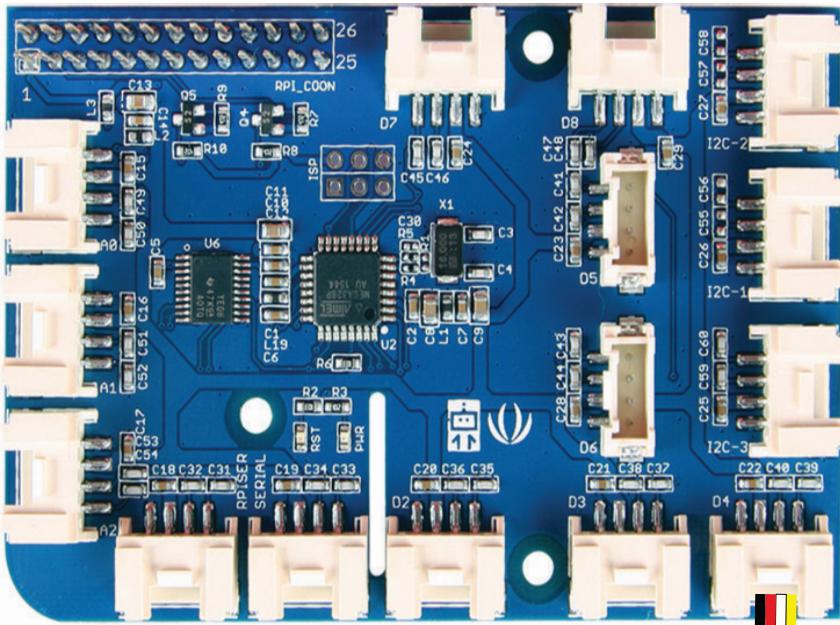
```
# Lichtsensor ist an PIN A1 angeschlossen  
light_sensor = 1  
  
# Ein Schwellwert, damit der Lichtsensor  
# nicht bei jeder kleinen Änderung reagiert  
threshold = 10  
  
#Angabe, dass der Sensor einen INPUT Wert liefert  
grovepi.pinMode(light_sensor, "INPUT")
```

## 3. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):  
  
    while True:  
        # wenn der Lichtsensor größer, gleich 300 Lumen misst, dann...  
        if grovepi.analogRead(light_sensor) >= 300:  
            #Hier kann eine oder mehrere Aktionen rein  
            print("Es ist hell!")  
            time.sleep(2)  
  
        # wenn der Lichtsensor weniger als 300 Lumen misst, dann...  
        elif grovepi.analogRead(light_sensor) < 300:  
            #Hier kann eine oder mehrere Aktionen rein  
            print("Es ist dunkel!")  
            time.sleep(2)  
  
cozmo.run_program(cozmo_program)
```

# Anschluss vom Touch Sensor

Schließe den Touch Sensor an einen digitalen Pin an, z.B. an D4.



# Nutzung des Touch Sensors

## 1. Benötigte Pakete:

```
import cozmo  
import time  
import grovepi
```

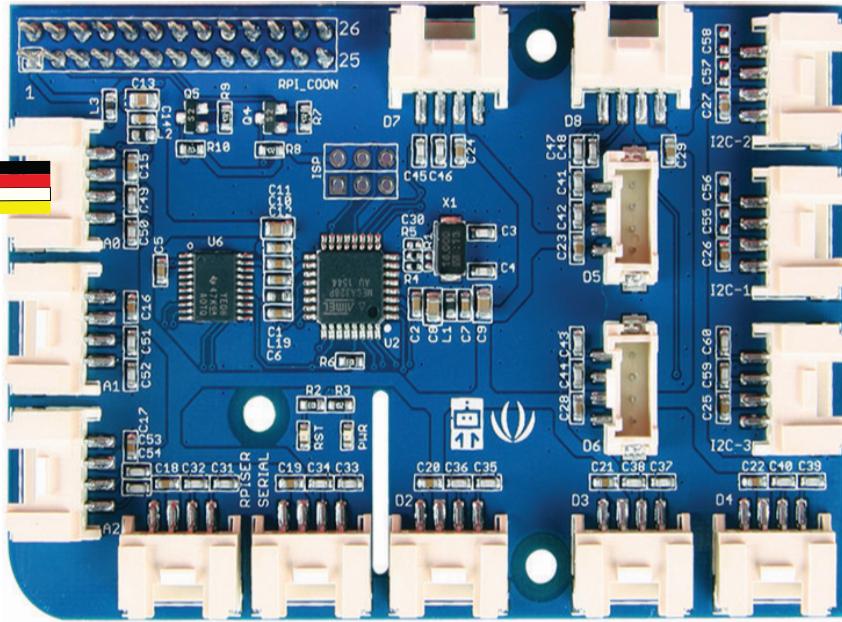
## 2. Initialisierung:

```
# an PIN D4 angeschlossen  
touch_sensor = 4  
  
# Angabe, dass der Sensor einen INPUT liefert  
grovepi.pinMode(touch_sensor, "INPUT")
```

## 3. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):  
    while True:  
        # wenn der Touch Sensor berührt wurde (1), dann...  
        if grovepi.digitalRead(touch_sensor) == 1:  
            #Hier kann eine oder mehrere Aktionen rein  
  
cozmo.run_program(cozmo_program)
```

# Anschluss des Temperatursensors



Schließe den Temperatursensor an einen analogen Pin, z.B. A0.

Die Temperatur wird in Celsius von -40 bis 125 gemessen.

# Nutzung des Temperatursensors

## 1. Benötigte Pakete:

```
import cozmo
import time
import grovepi
```

## 2. Initialisierung:

```
temperatur_sensor = 1
```

## 3. Beispiel:

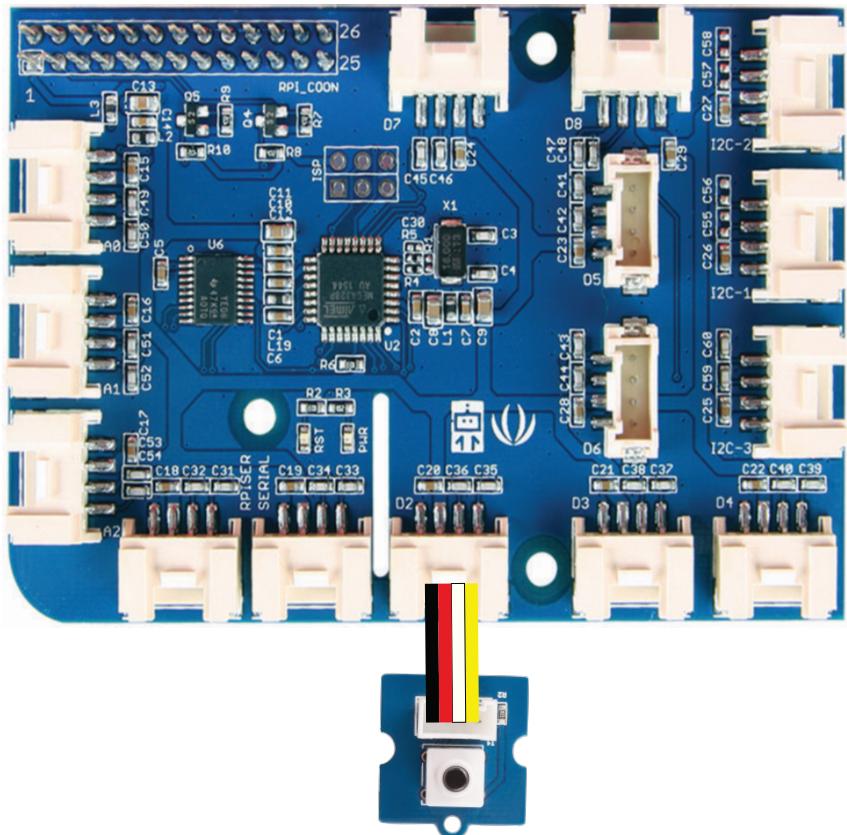
```
def cozmo_program(robot: cozmo.robot.Robot):
    while True:
        try:
            temperatur = grovepi.temp(temperatur_sensor, '1.2')
            #wenn die Temperatur kleiner, gleich 18, dann...
            if temperatur <= 18:
                #Hier können mehrere Aktionen rein
                print("Es ist kalt!")
                print(round(temperatur, 1), "Grad")
                time.sleep(10)
            #wenn die Temperatur größer 18, dann...
            elif temperatur > 18:
                #Hier können mehrere Aktionen rein
                print("Es ist warm!")
                print(round(temperatur, 1), "Grad")
                time.sleep(10)

        except KeyboardInterrupt:
            break
        except IOError:
            print ("Error")

cozmo.run_program(cozmo_program)
```

# Anschluss vom Button

Schließe den Button an einen digitalen Pin an, z.B. D2.



# Nutzung des Buttons

## 1. Benötigte Pakete:

```
import cozmo  
import time  
import grovepi
```

## 2. Initialisierung:

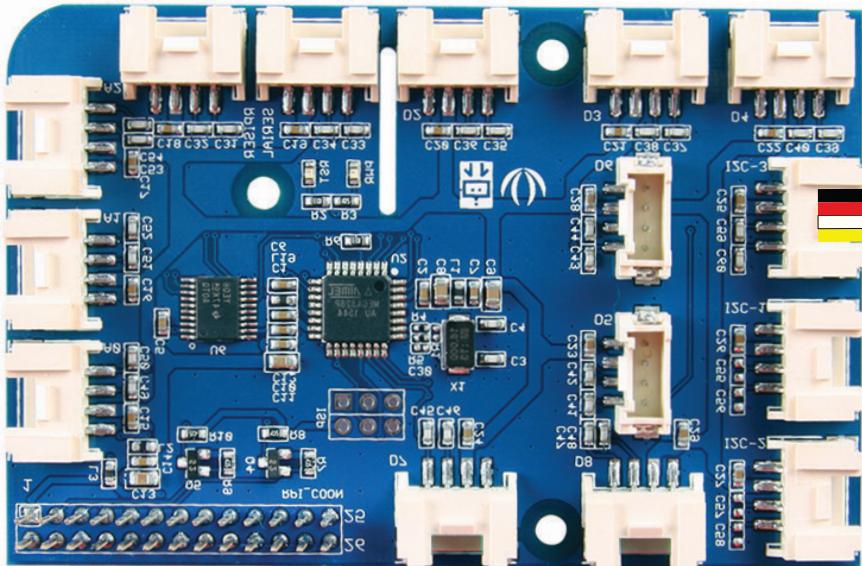
```
# an PIN D2 angeschlossen  
button = 2  
  
#Angabe, das der Button einen INPUT-Wert liefert  
grovepi.pinMode(button, "INPUT")
```

## 3. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):  
    while True:  
        # wenn der Button gedrückt wurde (1), dann...  
        if grovepi.digitalRead(button) == 1:  
            #Hier kann eine Aktion oder mehrere rein.  
  
cozmo.run_program(cozmo_program)
```

# Anschluss des LCD-Displays

HW  
Hardware



Um den LCD-Display anzuschließen, nutze eine I2C Bühse.

Der LCD-Display kann einen Text und Zahlen anzeigen. Der Display kann zudem in verschiedenen Farben aufleuchten (weiß, rot, grün, blau). Besonders hilfreich ist der Display, wenn du dir Werte der Sensoren anzeigen lassen willst.

# Nutzung der LCD-Displays

## 1. Benötigte Pakete:

```
import cozmo
import displayfunction
import time
```

## 2. Initialisierung:

```
display = displayfunction
```

## 3. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):
    while True:
        #zeigt einen Text
        display.setText("Hello world.")
        #setzt die Hintergrundfarbe
        display.setRGB(0,128,64)
        time.sleep(2)
        display.setText("Bye Bye.")
        display.setRGB(0,0,0)
        time.sleep(2)

cozmo.run_program(cozmo_program)
```

# Cozmo Sprachausgabe, Melodien und Animation



Cozmo kann Melodien spielen, die du ihm vorgibst. Dabei kannst du jede Note angeben und in welchen Takt sie gespielt werden soll.



Außerdem kann er einen Text sagen.

Zusätzlich kann er noch vorgefertigte Animationen abspielen, z.B. eine Katze oder einen Hund nachahmen.

# Nutzung der Sprachausgabe, Melodien und Animation

## 1. Benötigte Pakete:

```
import cozmo  
import asyncio
```

## 2. Initialisierung:

```
#Erstelle die Melodie  
melody = [  
    # jede Note einzeln  
    cozmo.song.SongNote(  
        cozmo.song.NoteTypes.C2,  
        cozmo.song.NoteDurations.Quarter),  
    cozmo.song.SongNote(  
        cozmo.song.NoteTypes.E2,  
        cozmo.song.NoteDurations.ThreeQuarter),  
    cozmo.song.SongNote(  
        cozmo.song.NoteTypes.D2,  
        cozmo.song.NoteDurations.Half),  
    cozmo.song.SongNote(  
        cozmo.song.NoteTypes.C2,  
        cozmo.song.NoteDurations.Whole),  
]
```

## 3. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):  
  
    while True:  
        #Sagt einen Text  
        robot.say_text("Hallo, ich bin Cozmo!").wait_for_completed()  
        #Spielt die Melodie  
        robot.play_song(melody, loop_count=1).wait_for_completed()  
        #Spielt eine Animation  
        robot.play_anim_trigger(cozmo.anim.Triggers.CodeLabDog).wait_for_completed()  
  
cozmo.run_program(cozmo_program)
```

# Cozmo Würfel



Du kannst die Würfel in verschiedenen Farben leuchten lassen.

**Tipp:**

Cube1 sieht wie eine Büroklammer aus.

Cube2 sieht wie eine Lampe/Herz aus.

Cube3 sieht wie die Buchstaben „ab“ auf einen „T“ aus.

# Nutzung der Würfel

## 1. Benötigte Pakete:

```
import cozmo
import time
from cozmo.objects import LightCube1Id, LightCube2Id, LightCube3Id
```

## 2. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):
    purpleColor = cozmo.lights.Color(rgb=(136, 0, 255))
    cube1 = robot.world.get_light_cube(LightCube1Id)
    cube2 = robot.world.get_light_cube(LightCube2Id)
    cube3 = robot.world.get_light_cube(LightCube3Id)

    while True:
        if cube1 is not None:
            #Setzt die Farbe des Cube1 auf eine selbst erstellte Farbe
            cube1.set_lights(cozmo.lights.Light(purpleColor))
            time.sleep(1)
        if cube2 is not None:
            #Setzt die Farbe des Cube2 auf Blau
            cube2.set_lights(cozmo.lights.blue_light)
            time.sleep(1)
            #Schaltet Cube2 aus
            cube2.set_lights(cozmo.lights.off_light)
        if cube3 is not None:
            #Setzt die Farbe des Cube3 auf Grün
            cube3.set_lights(cozmo.lights.green_light)
            time.sleep(1)
            #Setzt die Farbe des Cube3 auf Rot
            cube3.set_lights(cozmo.lights.red_light)
        else:
            cozmo.logger.warning("Überprüfe die Batterie der Würfel.")
            time.sleep(1)

cozmo.run_program(cozmo_program)
```

Hier definierst du die Würfel.  
Und kannst du eigene Farben  
definieren.

Hier überprüfst du zunächst, ob der  
Würfel da ist.

Danach kannst du ihn anmachen und  
auch wieder aus machen.

# Cozmos Rucksacklicht



Du kannst Cozmos Rucksacklicht in verschiedenen Farben leuchten lassen.

# Nutzung des Rucksacklichtes

## 1. Benötigte Pakete:

```
import cozmo  
import time
```

## 2. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):  
    #Definition einer eigenen Farbe mit RGB  
    purpleColor = cozmo.lights.Color(rgb=(136,0,255))  
  
    while True:  
        #Setzt das Licht auf Rot  
        robot.set_all_backpack_lights(cozmo.lights.red_light)  
        time.sleep(1)  
        #Setzt das Licht auf Grün  
        robot.set_all_backpack_lights(cozmo.lights.green_light)  
        time.sleep(1)  
        #Setzt das Licht auf Blau  
        robot.set_all_backpack_lights(cozmo.lights.blue_light)  
        time.sleep(1)  
        #Setzt das Licht auf die eigene Farbe Purple  
        robot.set_all_backpack_lights(cozmo.lights.Light(purpleColor))  
        time.sleep(1)  
        #Schaltet das Licht aus  
        robot.set_all_backpack_lights(cozmo.lights.off_light)  
  
cozmo.run_program(cozmo_program)
```

# Cozmo Mechanik

## Fahren

Cozmo kann vorwärts und rückwärts fahren.

Vorwärts = positive Zahl als Distanz in mm, z.B. 150

Rückwärts = negative Zahl als Distanz in mm, z.B. -150



## Kopf bewegen

Cozmo kann seinen Kopf hoch und runter bewegen.

Hoch = positive Zahl als Radius pro Sekunde, z.B. 5

Runter = negative Zahl als Radius pro Sekunde, z.B. -5

## Hebeam bewegen

Cozmo kann seinen Hebeam hoch und runter bewegen.

Hoch = positive Zahl als Radius pro Sekunde, z.B. 5

Runter = negative Zahl als Radius pro Sekunde, z.B. -5

# Nutzung der Mechanik

## 1. Benötigte Pakete:

```
import cozmo
import time
from cozmo.util import degrees, distance_mm, speed_mmmps
```

## 2. Beispiel:

```
def cozmo_program(robot: cozmo.robot.Robot):
    while True:
        #Fährt ein Rechteck
        for i in range(4):
            #Fährt 50 mm forwärts mit 100 mm pro Sekunde
            robot.drive_straight(distance_mm(50), speed_mmmps(100)).wait_for_completed()
            #Dreht sich auf der Stelle um 90 Grad
            robot.turn_in_place(degrees(90)).wait_for_completed()
        #Hebt den Kopf um 10
        robot.move_head(10)
        #Hebt den Hebearm um 5
        robot.move_lift(5)
        time.sleep(2)
        #Senkt den Kopf um 10
        robot.move_head(-10)
        #Senkt den Hebearm um 5
        robot.move_lift(-5)

cozmo.run_program(cozmo_program)
```