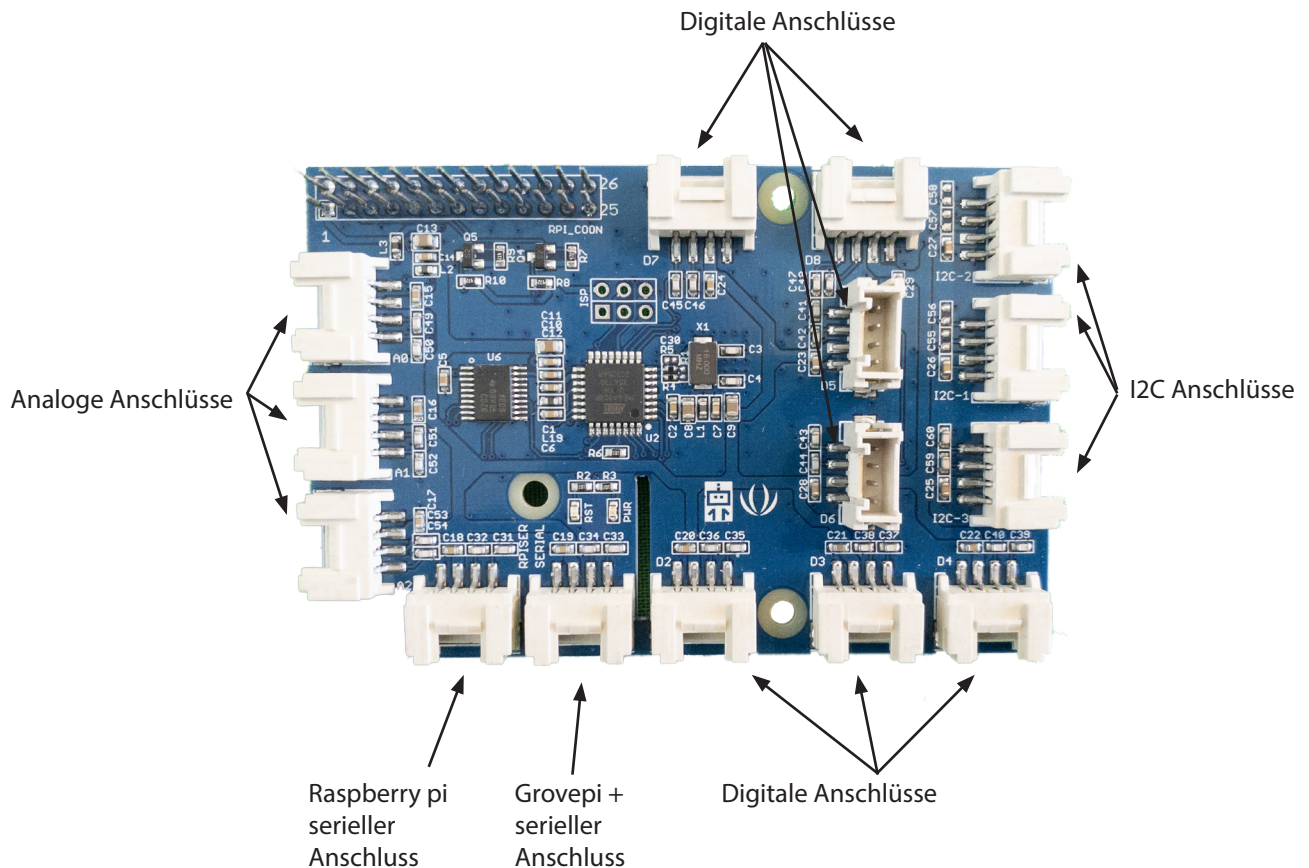




Grovepi+ Beschreibung

Beschreibung der Ports des Grovepi+ für den Raspberry Pi:



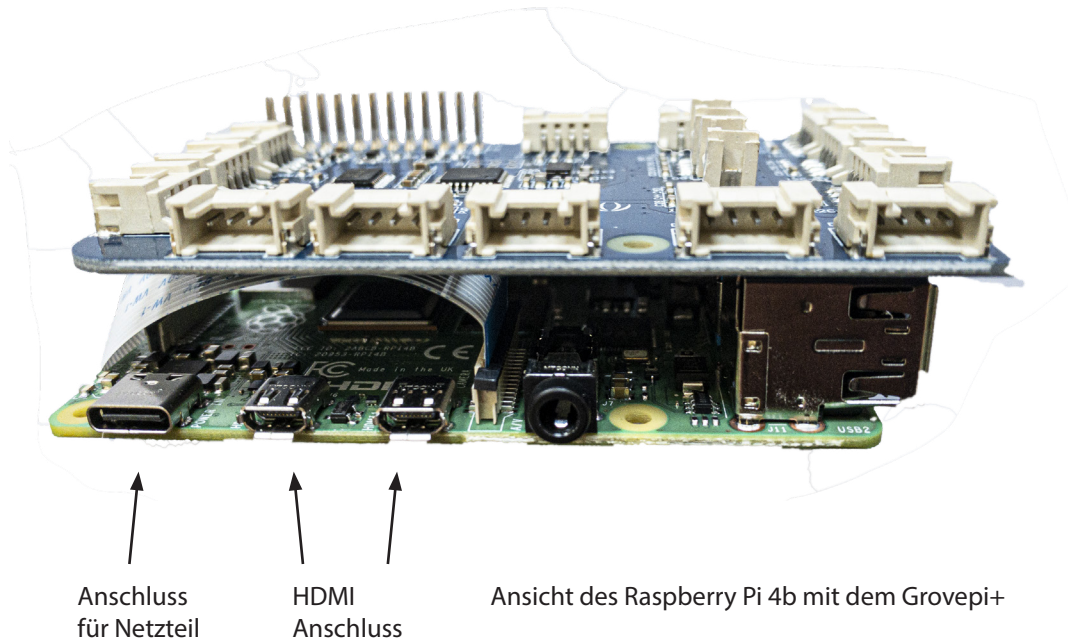
Die Portnummern gehen von links nach rechts. Sie sind auf dem Foto ersichtlich. Auf der linken Seite befinden sich die analogen Ports mit den Nummer A0 bis A2.

An der unteren Kante befinden sich links zwei serielle Ports, welche nicht genutzt werden können. Danach folgen drei digitale Ports (Anschluss D2 bis D4).

Am rechten Rand des Fotos befinden sich 3 I2C - Ports I2C-3, I2C-1 und I2C2.

Am oberen Rand des Fotos befinden sich zwei weitere digitale Ports D7 & D8.

Rechts neben der Mitte der Platine befinden sich nochmals zwei digitale Ports D5 & D6



Auf den weiteren Seiten wird auf Sensoren und Aktoren des Grovepi+ Kit eingegangen. Es ist nur eine kleine Auswahl von Sensoren und Aktoren, welche für den Grovepi+ zu haben sind. Folgende Sensoren und Aktoren könnten auch noch interessant sein (nicht abschliessende Liste):

Sensoren

- Kollisions-Sensor
- Staub-Sensor
- Infrarot-Empfänger
- PIR-Bewegung-Sensor
- Schalter
- Berührungs-Sensor
- Wasser-Sensor
- CO₂ - Sensor
- Barometer
- UV - Sensor

Aktoren

- MP3 Musik Modul
- Recorder
- Lautsprecher
- Vibrations-Motor


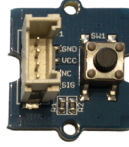
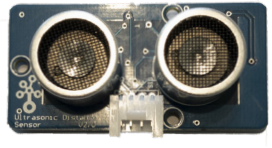
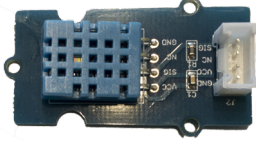






Sonstige

- GPS
- Uhr
- NFC
- NFC-Tag
- RFID-Leser

Eine Übersicht, was für Sensoren & Aktoren zu haben sind, finden Sie auf der Website <https://seeedstudio.com>



GrovePi+ Sensor - Informationen

Name	Art	Bild	Python Beispielcode
Schallsensor	Input		Misst Lautstärke (analog) <pre>soundSensor = 0 # das 1. Port (Analogport) grovepi.pinMode(soundSensor, "INPUT") wert = grovepi.analogRead(soundSensor)</pre>
Knopf	Input		Misst ob gedrückt (digital) <pre>button = 3 # das 4. Port (Digitalport) grovepi.pinMode(button, "INPUT") grovepi.digitalRead(button)</pre>
Ultraschall	Input		Misst Distanzen (digital) <pre>ultra_sonic = 4 # das 5. Port (Digitalport) wert = ultrasonicRead(ultra_sonic)</pre>
Feuchtigkeit	Input		Misst Temperatur & Feuchtigkeit (digital) <pre>sensor = 4 # der 5. Port (Digitalport) blue = 0 # für den blauen Sensor white = 1 # für den weissen Sensor (nicht dabei) [temp, feuch] = grovepi.dht(sensor, blue)</pre>
.Summer	Output		Lässt ein Summen ertönen (digital) <pre>buzzer = 8 # der 9. Port (Digitalport) grovepi.setPinmode(buzzer, "OUTPUT") grovepi.digitalWrite(buzzer, 1) # einschalten grovepi.digitalWrite(buzzer, 0) # ausschalten</pre>
Led (Rot, Grün, Blau)	Output		Leuchtdiode (LED) (digital) <pre>led = 4 # der 5. Port (Digitalport) grovepi.setPinmode(led, "OUTPUT") grovepi.digitalWrite(led, 1) # einschalten grovepi.digitalWrite(led, 0) # ausschalten</pre>
Rotation	Input		Setz einen Wert (analog) <pre>potentiometer = 0 # der 1. Port (Analogport) grovepi.setPinmode(potentiometer, "INPUT") wert = grovepi.analogRead(potentiometer)</pre> (Gebrauch im Internet nachlesen)
Lichtsensor	Input		Misst Licht (analog) <pre>light_sensor = 0 # der 1. Port (Analogport) grovepi.setPinmode(light_sensor, "INPUT") wert = grovepi.analogRead(light_sensor)</pre> (Gebrauch im Internet nachlesen)
Relay	Output		Kann Geräte mit hoher Spannung steuern (digital) <pre>relay der 5. Port (Digitalport) grovepi.setPinmode(relay, "OUTPUT") grovepi.digitalWrite(relay, 1) # einschalten grovepi.digitalWrite(relay, 0) # ausschalten</pre>
LCD	Output		(Gebrauch im Internet nachlesen)



Beispielcode

Auf den folgenden Seiten befindet sich Beispielcode, der zeigen soll, wie die einzelnen Sensoren und Aktoren verwendet werden können.

Beispielcode für den Button, den Buzzer und das LED

```
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")
```



Beispielcode für den Temperatur & Feuchtigkeitssensor

```
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 # math - damit Python rechnen kann
6 from grovepi import *
7 import math
8
9 # Der Feuchtigkeits- & Temperatursensor wird auf Port 4 (Digitalport) erwartet
10 sensor = 4 #Port for Button
11
12 # Den Feuchtigkeits- & Temperatursensor gibt es 2 Variationen. Weiss und Blau
13 # Wir haben den blauen Sensor
14 blue = 0
15 white = 1
16
17 # Das ist jetzt die Endlosschleife
18 while True:
19
20     # Da auch mal etwas schief gehen kann sagen wir Python versuche mal folgende Anweisungen
21     try:
22
23         # Lese vom digitalen Port den Wert des Sensors ein. Der Feuchtigkeits- & Temperatursensor
24         # gibt gleichzeitig zwei Werte zurück. Sie werden in einem Array gespeichert.
25         # Bei der Funktion dht ist der erste Parameter das Port, der zweite Parameter der Typ
26         # des Sensors
27         [temperatur, feuchtigkeit] = dht(sensor, blue)
28
29         # Testen, ob in beiden Variablen gültige Werte stehen
30         # isnan steht für "is not a number"
31         if not math.isnan(temperatur) and not math.isnan(feuchtigkeit):
32
33             # Mit print, können Ausgaben an der Konsole formatiert werden
34             # Die %-Zeichen sind Platzhalter für Zahlen
35             # .02f steht dafür, dass die Zahl 2 Nachkommastellen hat und eine Flisskommazahl ist
36             # Die beiden %% stehen dafür, dass wirklich ein %-Zeichen geschrieben wird
37             # Das %-Zeichen nach dem Anführungszeichen steht dafür, dass jetzt die Daten stehen
38             print("Temperatur = %.02f C Feuchtigkeit = %.02f%%" % (temperatur, feuchtigkeit))
39
40     # Abbruch durch Control-C
41     except KeyboardInterrupt:
42         break
43
44     except (IOError, TypeError) as e:
45         print("Error")
```



Beispielcode für den Soundsensor und das LED

```
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  # time - damit eine Pause eingebaut werden kann
6  from time import sleep
7  from grovepi import *
8
9  # Der Sound-Sensor ist auf dem Port 0 (analoges Port) eingesteckt
10 sound_sensor = 0
11
12 # Das LED wird auf Port 4 (Digitalport) erwartet
13 led = 4
14
15 # Dem Grovepi wird mitgeteilt, dass am Port 2 (sound_sensor) ein Sensor (INPUT) angeschlossen ist
16 pinMode(sound_sensor, "INPUT")
17
18 # Dem Grovepi wird mitgeteilt, dass am Port 4 (LED) ein Aktor (OUTPUT) angeschlossen ist
19 pinMode(led, "OUTPUT")
20
21 # Schwellwert, über diesem Wert wird die Lautstärke als zu laut angesehen
22 threshold = 400
23
24 # Das ist jetzt die Endlosschleife
25 while True:
26
27     # Da auch mal etwas schief gehen kann sagen wir Python versuche mal folgende Anweisungen
28     try:
29
30         # Lese vom analogen Port Lautstärkenwerte ein
31         value = analogRead(sound_sensor)
32
33         # Wenn der Schwellwert überschritten wurde
34         if value > threshold:
35
36             # Schreibe eine 1 auf das Port an dem LED angeschlossen ist (1=ein, 0=aus)
37             digitalWrite(led, 1)
38
39         else:
40             digitalWrite(led, 0)
41
42     # Abbruch durch Control-C
43     except KeyboardInterrupt:
44         digitalWrite(led, 0) # turn LED off in case of error
45         break
46
47     # Wenn sonst eine Fehler auftritt
48     except (IOError, TypeError) as e:
49         print("Error", e)
```



Beispielcode für das Potentiometer und das LED

```
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 # math - damit Python rechnen kann
6 # time - damit eine Pause eingebaut werden kann
7 from grovepi import *
8 from time import sleep
9
10 # Potentiometer wird auf Port 2 (Analogport) erwartet
11 potentiometer = 2
12
13 # Das LED wird auf Port 4 (Digitalport) erwartet
14 led = 5
15
16 # Dem Grovepi wird mitgeteilt, dass am Port 2 (sound_sensor) ein Sensor (INPUT) angeschlossen ist
17 pinMode(potentiometer, "INPUT")
18
19 # Dem Grovepi wird mitgeteilt, dass am Port 4 (LED) ein Aktor (OUTPUT) angeschlossen ist
20 pinMode(led, "OUTPUT")
21
22 # das Potentiometer lässt sich um 300 Grad drehen.
23 maxangle = 300
24
25 # Grösster Wert, den das Potentiometer ausgibt
26 maxval = 1023
27
28 # Das ist jetzt die Endlosschleife
29 while True:
30
31     # Da auch mal etwas schief gehen kann sagen wir Python versuche mal folgende Anweisungen
32     try:
33
34         # die Werte liegen zwischen 0 und 1023
35         value = analogRead(potentiometer)
36
37         # Grad die das Potentiometer gedreht ist
38         degrees = round(value * 300 / maxval, 1)
39
40         # Helligkeit des LEDs, 0 = aus, 255 ganz an
41         brightness = int(value * 255 / maxval)
42
43         # Die Helligkeit analog auf das LED schreiben
44         analogWrite(led, brightness)
45
46         print("Wert Potentiometer = %d, Winkel = %5.1f, Helligkeit = %3d" % (value, degrees, brightness))
47
48         # Eine Sekunde warten
49         sleep(1)
50
51     # Abbruch durch Control-C
52     except KeyboardInterrupt:
53         digitalWrite(led, 0) # turn LED off in case of error
54         break
55
56     except (IOError, TypeError) as e:
57         print("Error")
```




Beispielcode für den Ultraschall Sensor

```
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 # time - damit eine Pause eingebaut werden kann
6 from grovepi import *
7 from time import sleep
8
9 # Der Sound-Sensor ist auf dem Port 0 (analoges Port) eingesteckt
10 ultrasonic_sensor = 4
11
12 # Das ist jetzt die Endlosschleife
13 while True:
14
15     # Da auch mal etwas schief gehen kann sagen wir Python versuche mal folgende Anweisungen
16     try:
17
18         # Lese vom analogen Port Lautstärkenwerte ein
19         value = ultrasonicRead(ultrasonic_sensor)
20
21         print("Distanz ca. = %d cm" %value)
22         sleep(.5)
23
24     # Abbruch durch Control-C
25     except KeyboardInterrupt:
26         break
27
28     # Wenn sonst eine Fehler auftritt
29     except (IOError,TypeError) as e:
30         print("Error", e)
```




Beispielcode für den Lichtsensor und das LED

```
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  # time - damit eine Pause eingebaut werden kann
6  from time import sleep
7  from grovepi import *
8
9  # Der Licht-Sensor ist auf dem Port 0 (analoges Port) eingesteckt
10 light_sensor = 0
11
12 # Das LED wird auf Port 4 (Digitalport) erwartet
13 led = 4
14
15 # Dem Grovepi wird mitgeteilt, dass am Port 0 (light_sensor) ein Sensor (INPUT) angeschlossen ist
16 pinMode(light_sensor, "INPUT")
17
18 # Dem Grovepi wird mitgeteilt, dass am Port 4 (LED) ein Aktor (OUTPUT) angeschlossen ist
19 pinMode(led, "OUTPUT")
20
21 # Maximaler Werte des Licht-Sensors
22 maxvalue = 1023
23
24 # Das ist jetzt die Endlosschleife
25 while True:
26
27     # Da auch mal etwas schief gehen kann sagen wir Python versuche mal folgende Anweisungen
28     try:
29
30         # Lese vom analogen Port Lichtwerte ein
31         value = analogRead(light_sensor)
32
33         # Wenn der Lichtwert bei der Hälfte des Maximums liegt wird das Led eingeschaltet
34         if value > int(maxvalue / 2):
35
36             # Schreibe eine 1 auf das Port an dem LED angeschlossen ist (1=ein, 0=aus)
37             digitalWrite(led, 1)
38
39         else:
40             digitalWrite(led, 0)
41
42         print("Wert der gemessenen Lichtstärke (max=%d) = %4d" %(maxvalue,value))
43
44         # Schlafe 0.5 Sekunden
45         sleep(.5)
46
47     # Abbruch durch Control-C
48     except KeyboardInterrupt:
49         digitalWrite(led, 0) # turn LED off in case of error
50         break
51
52     # Wenn sonst eine Fehler auftritt
53     except (IOError, TypeError) as e:
54         print("Error", e)
```