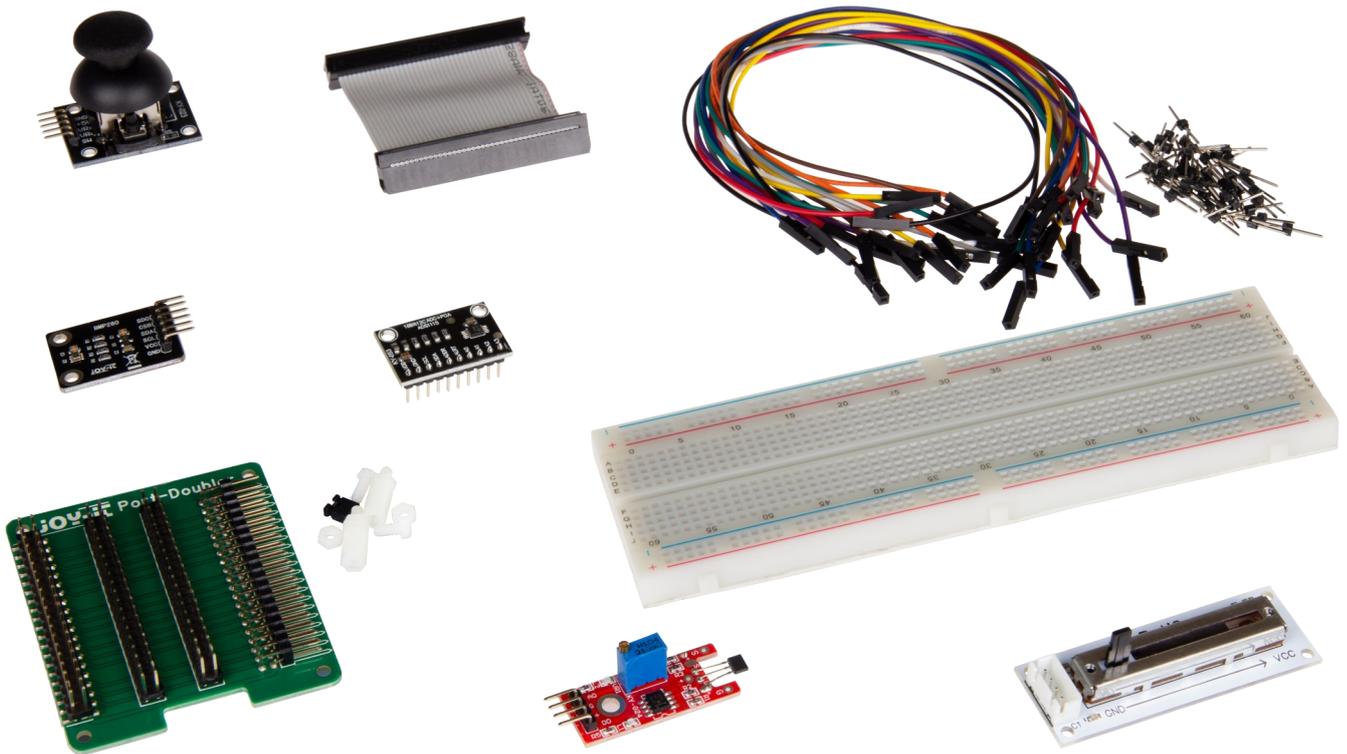


JOYPI ERWEITERUNGSPAKET

RB-JoyPi-Addon



1. ALLGEMEINE INFORMATIONEN

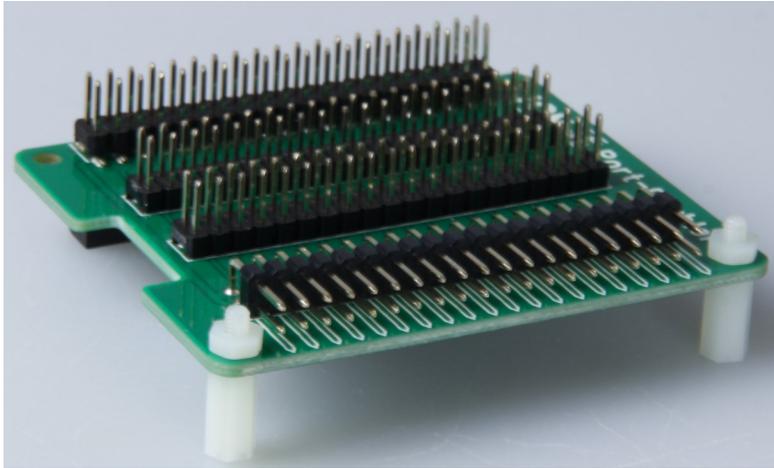
Sehr geehrter Kunde,
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

Diese Anleitung bezieht sich nur auf das Erweiterungspaket für den JoyPi. Sollten Sie Fragen zu den Grundfunktionen des JoyPis haben, können Sie sich [hier](#) die Hauptanleitung für den JoyPi herunterladen.

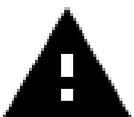
2. ZUSAMMENBAU

Schrauben Sie zunächst die Abstandshalter an den Port-Doubler, wie im Bild unten zusehen.



Ihr Raspberry Pi sollte bereits im JoyPi verschraubt sein.

Nun können Sie den Port-Doubler auf den Raspberry Pi stecken.



Der Port-Doubler wird nicht mit dem Raspberry Pi verschraubt, da der Port-Doubler wieder entfernt werden muss, wenn Sie den JoyPi schließen möchten.

Zum Schluss müssen Sie nur noch die GPIO-Leiste des JoyPis, mit der des Port-Doublers mit dem mitgeliefertem Kabel verbinden.

3. INSTALLATION

Für die Inbetriebnahme des ADCs und des Druck- und Temperatursensors, ist es nötig die entsprechenden Bibliotheken zu installieren.

In unserem Vorbereitetem Image, welches Sie sich [hier](#) herunterladen können, sind diese bereits installiert.

Wenn Sie ein eigenes Image verwenden, können Sie die Bibliotheken mit folgenden Befehlen installieren.

[adafruit/Adafruit_CircuitPython_ADS1x15](#)

```
sudo pip3 install adafruit-circuitpython-ads1x15
```

[pimoroni/bmp280-python](#)

```
sudo pip3 install bmp280
```

Die von uns vorbereiteten Beispielskripte, sind ebenfalls bereits auf dem Image vorhanden. Sie können das Skriptverzeichnis auch [hier](#) manuell herunterladen, oder mit folgenden Befehlen auf Ihrem Desktop speichern:

```
cd Desktop/
```

```
wget https://joy-pi.net/wp-content/uploads/2020/07/Joy-Pi.zip
```

Außerdem müssen Sie, wenn Sie Ihr eigenes Image verwenden I2C aktivieren. Geben sie dazu folgenden Befehl in das Terminal ein:

```
sudo raspi-config
```

Aktivieren Sie nun unter **4 Interfacing Options** den Punkt **P5 I2C**.

4. DER PORT-DOUBLER

Der Port-Doubler hat gleich 4 GPIO-Leisten und ermöglicht dadurch den betrieb externer Sensoren mit dem JoyPi.

Die Pinbelegung jeder der vier GPIO-Leisten des Port-Doublers, gleicht der des Raspberry Pi.

1	3.3 V DC			2	5 V DC
3	GPIO 2 (SDA1, I2C)			4	5 V DC
5	GPIO 3 (SCL1, I2C)			6	Ground
7	GPIO 4			8	GPIO 14 (TXD0)
9	Ground			10	GPIO 15 (RXD0)
11	GPIO 17			12	GPIO 18
13	GPIO 27			14	Ground
15	GPIO 22			16	GPIO 23
17	3.3 V			18	GPIO 24
19	GPIO 10 (SPI, MOSI)			20	Ground
21	GPIO 9 (SPI, MISO)			22	GPIO 25
23	GPIO 11 (SPI, CLK)			24	GPIO 8 (SPI)
25	Ground			26	GPIO 7 (SPI)
27	ID_SD (I2C, EEPROM)			28	ID_SC
29	GPIO 5			30	Ground
31	GPIO 6			32	GPIO 12
33	GPIO 13			34	Ground
35	GPIO 19			36	GPIO 16
37	GPIO 26			38	GPIO 20
39	Ground			40	GPIO 21

Da alle Pins des Raspberry Pi im JoyPi verwendet werden, können nur die Pins verwendet werden, die sie mit Hilfe der beiden Schalteinheiten des JoyPis, trennen können. Diese sind im Schaubild oben hervorgehoben.

Zusätzlich können die Pins 3 und 5 für die I2C Kommunikation verwendet werden.

5. ANALOG-DIGITAL-CONVERTER

Der Analog-Digital-Converter (ADC) wandelt eine analoge Spannung in ein digitales Signal um, welches mit Hilfe des I2C-Protokolls vom Raspberry Pi abgerufen werden kann.



Die Pins A0 - A3 des ADCs sind die 4 analogen Eingangskanäle. So können 4 analoge Sensoren gleichzeitig betrieben werden.

Bitte beachten Sie, dass die Spannung auf den Eingangskanälen des ADCs 3.3 V nicht übersteigen darf.

Die weitere Pinbelegung können Sie der unteren Tabelle entnehmen.

ADC	Raspberry Pi
VDD	3.3 V (Pin 1)
GND	GND (Pin 6)
SCL	SCL (Pin 5)
SDA	SDA (Pin 3)

Das folgende Codebeispiel gibt die die Spannungen aus, die an allen 4 Eingangskanälen anliegt.

```
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn

# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)

# Create the ADC object using the I2C bus
ads = ADS.ADS1115(i2c)
ads.gain = 2/3
# Create single-ended input on channels
chan0 = AnalogIn(ads, ADS.P0)
chan1 = AnalogIn(ads, ADS.P1)
chan2 = AnalogIn(ads, ADS.P2)
chan3 = AnalogIn(ads, ADS.P3)

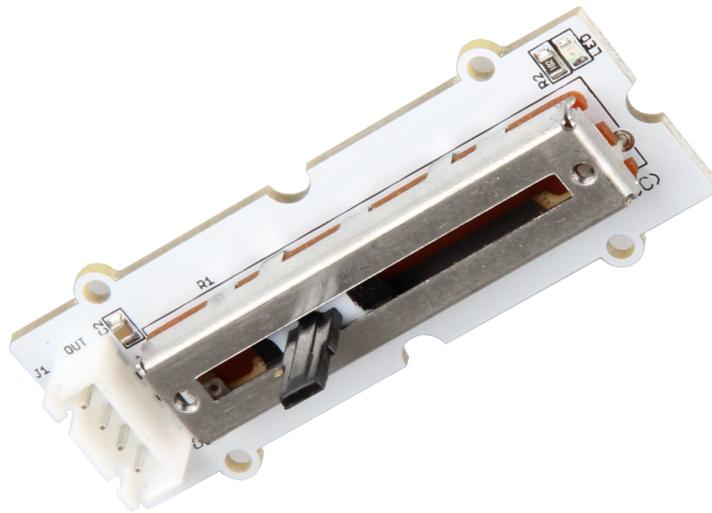
while True:
    print("channel 0: ", "{:>5}\t{:>5.3f}".format(chan0.value, chan0.voltage))
    print("channel 1: ", "{:>5}\t{:>5.3f}".format(chan1.value, chan1.voltage))
    print("channel 2: ", "{:>5}\t{:>5.3f}".format(chan2.value, chan2.voltage))
    print("channel 3: ", "{:>5}\t{:>5.3f}".format(chan3.value, chan3.voltage))
    print("-----")
    time.sleep(1)
```

Sie können die Datei mit folgendem Befehl ausführen:

```
sudo python3 /home/pi/Desktop/Joy-Pi/Extensions/adc.py
```

6. SCHIEBEPOTENTIOMETER

Das Schiebepotentiometer ist ein linear variabler Widerstand mit einem Gesamtwiderstand von 10k Ω . Wenn man den Schieberegler von der einen zur anderen Seite verschiebt wird dessen Ausgangsspannung zwischen 0V und der Versorgungsspannung verändert.



Für die Verwendung des Potentiometers benötigen Sie zusätzlich den ADC.

Die Anschlussbelegung können Sie der unteren Tabelle entnehmen.

Poti	ADC	Raspberry Pi
OUT	A0	-
LED	-	GPIO 13 (Pin 33)
U	VDD	3.3 V (Pin 1)
G	GND	GND (Pin 6)
-	SCL	SCL (Pin 5)
-	SDA	SDA (Pin 3)

Das folgende Codebeispiel gibt die Spannungen aus, die am Ausgangspin des Potentiometer anliegt.

Außerdem leuchtet die LED am Potentiometer auf, sobald der im Skript eingestellte Grenzwert (V_Limit) überschritten wird.

```
import RPi.GPIO as GPIO
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
led_pin = 13          # LED Pin
V_Limit = 2.5        # Voltage limit
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)
# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)

# Create the ADC object using the I2C bus
ads = ADS.ADS1115(i2c)
ads.gain = 2/3
# Create single-ended input on channels
chan0 = AnalogIn(ads, ADS.P0)
GPIO.output(led_pin, GPIO.LOW)

while True:
    if V_Limit > chan0.voltage:
        GPIO.output(led_pin, GPIO.LOW)
        print("Potentiometer Voltage: ", "{:>5.3f}".format(chan0.voltage), "V / Limit reached")
        print("-----")
        time.sleep(1)
    else:
        GPIO.output(led_pin, GPIO.HIGH)
        print("Potentiometer Voltage: ", "{:>5.3f}".format(chan0.voltage), "V / Limit not reached")
        print("-----")
        time.sleep(1)
```

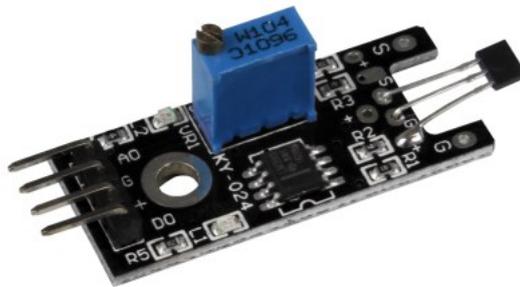
Sie können die Datei mit folgendem Befehl ausführen:

```
sudo python3 /home/pi/Desktop/Joy-Pi/Extensions/poti.py
```

7. MAGNETSENSOR

Das Magnetfeld wird vom Sensor gemessen und als analoger Spannungswert ausgegeben. Wenn der Grenzwert überschritten wird, liegt ein High Signal am Digitalpin an.

Dieser Grenzwert kann mit Hilfe des blauen Potentiometers eingestellt werden.



Für die Verwendung des Magnetsensors benötigen Sie zusätzlich den ADC.

Die Anschlussbelegung können Sie der unteren Tabelle entnehmen.

Magnetsensor	ADC	Raspberry Pi
A0	A0	-
G	GND	GND
+	VDD	3.3 V (Pin 1)
D0	-	GPIO 5 (Pin 29)
-	SCL	SCL (Pin 5)
-	SDA	SDA (Pin 3)

Das folgende Codebeispiel gibt die Spannungen aus, die am Ausgangspin des Magnetsensors anliegt.

Außerdem wird angezeigt ob der Grenzwert erreicht wurde oder nicht.

```
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)

# Create the ADC object using the I2C bus
ads = ADS.ADS1115(i2c)

# Create single-ended input on channels
chan0 = AnalogIn(ads, ADS.P0)

delayTime = 1
Digital_PIN = 5 #GPIO 5 (Pin29) Raspberry Pi

GPIO.setup(Digital_PIN, GPIO.IN, pull_up_down = GPIO.PUD_OFF)

while True:
    analog = '%.2f' % chan0.voltage

    if GPIO.input(Digital_PIN) == False:
        print ("Analog Voltage:", analog,"V, ", "Limit not reached")
    else:
        print ("Analog Voltage:", analog, "V, ", "Limit reached")
    print ("-----")
```

Sie können die Datei mit folgendem Befehl ausführen:

```
sudo python3 /home/pi/Desktop/Joy-Pi/Extensions/magnet.py
```

8. JOYSTICK

X und Y Position des Joysticks, werden als analoge Spannung auf den Ausgangspins ausgegeben.

In diesem Joystick wurde für die X-Achse, sowie für die Y-Achse, ein eigenes Potentiometer verbaut.

Zudem ist ein Knopf im Joystick verbaut, der erkennt wenn der Joystick runter gedrückt wird.



Für die Verwendung des Joysticks benötigen Sie zusätzlich den ADC.

Die Anschlussbelegung können Sie der unteren Tabelle entnehmen.

Joystick	ADC	Raspberry Pi
GND	GND	GND (Pin 6)
+5V	VCC	3.3 V (Pin 1)
VRx	A0	-
VRy	A1	-
SW	-	GPIO 6 (Pin 31)
-	SCL	SCL (Pin 5)
	SDA	SDA (Pin 3)

Das folgende Codebeispiel gibt die die Spannungen, die an den Pins der X und Y Achse anliegen, grafisch in einem Diagramm aus.

```
import matplotlib.pyplot as plt
import numpy as np
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

Button_PIN = 6
GPIO.setup(Button_PIN, GPIO.IN, pull_up_down = GPIO.PUD_UP)
# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)

# Create the ADC object using the I2C bus
ads = ADS.ADS1115(i2c)
ads.gain = 2/3
# Create single-ended input on channels
chan0 = AnalogIn(ads, ADS.P0)
chan1 = AnalogIn(ads, ADS.P1)

x = np.linspace(0, 6*np.pi, 100)
y = np.sin(x)

# You probably won't need this if you're embedding things in a tkinter plot...
plt.ion()

fig = plt.figure()
ax = fig.add_subplot(111)
#line1, = ax.plot(x, y, 'r-') # Returns a tuple of line objects, thus the comma

for phase in np.linspace(0, 10*np.pi, 500):
    plt.cla()
    plt.xlim(0, 3.3)
    plt.ylim(3.3, 0)
    plt.xlabel("Voltage X")
    plt.ylabel("Voltage Y")
    if GPIO.input(Button_PIN) == False:
        plt.title("Button pressed!")

    plt.plot(chan0.voltage, chan1.voltage, 'ro')
    fig.canvas.draw()
    fig.canvas.flush_events()
```

Außerdem wird angegeben, ob der Joystick gedrückt wird, oder nicht.

```
sudo python3 /home/pi/Desktop/Joy-Pi/Extensions/joystick.py
```

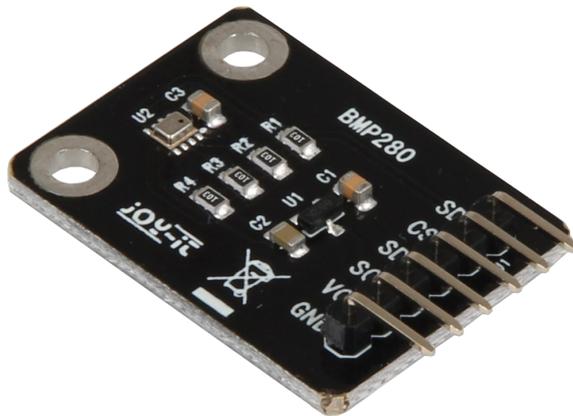
Sie können die Datei mit folgendem Befehl ausführen:

Wenn Sie das Programm über eine SSH-Verbindung ausführen verwenden sie folgenden Befehl:

```
sudo -E python3 /home/pi/Desktop/Joy-Pi/Extensions/joystick.py
```

9. DRUCK- UND TEMPERATURSENSOR

Dieser Kombisensor kann sowohl die Temperatur, als auch den Luftdruck messen.
Für die Kommunikation mit dem Raspberry Pi verwendet er genau wie der ADC das I2C-Protokoll.



Die Anschlussbelegung können Sie der unteren Tabelle entnehmen.

BMP280	Raspberry Pi
SD0	-
CSB	-
SDA	SDA (Pin 3)
SCL	SCL (Pin 5)
VCC	3.3 V (Pin 1)
GND	GND (Pin 6)

Das folgende Codebeispiel gibt die aktuelle Temperatur und den aktuellen Luftdruck aus.

```
import time
from bmp280 import BMP280

try:
    from smbus2 import SMBus
except ImportError:
    from smbus import SMBus

print("""temperature-and-pressure.py - Displays the temperature and pressure.
Press Ctrl+C to exit!
""")

# Initialise the BMP280
bus = SMBus(1)
bmp280 = BMP280(i2c_dev=bus)

while True:
    temperature = bmp280.get_temperature()
    pressure = bmp280.get_pressure()
    print('{:05.2f}*C {:05.2f}hPa'.format(temperature, pressure))
    time.sleep(1)
```

Sie können die Datei mit folgendem Befehl ausführen:

```
sudo python3 /home/pi/Desktop/Joy-Pi/Extensions/bmp.py
```

10. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem Elektroggesetz (ElektroG)



Symbol auf Elektro- und Elektronikgeräten:

Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

Rückgabemöglichkeiten:

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte bei denen keine äußere Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in haushaltsüblichen Mengen abgeben werden.

Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

Möglichkeit Rückgabe in Ihrer Nähe:

Wir senden Ihnen eine Paketmarke zu mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an Service@joy-it.net oder per Telefon an uns.

Informationen zur Verpackung:

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

11. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 98469 – 66 (10 - 17 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net