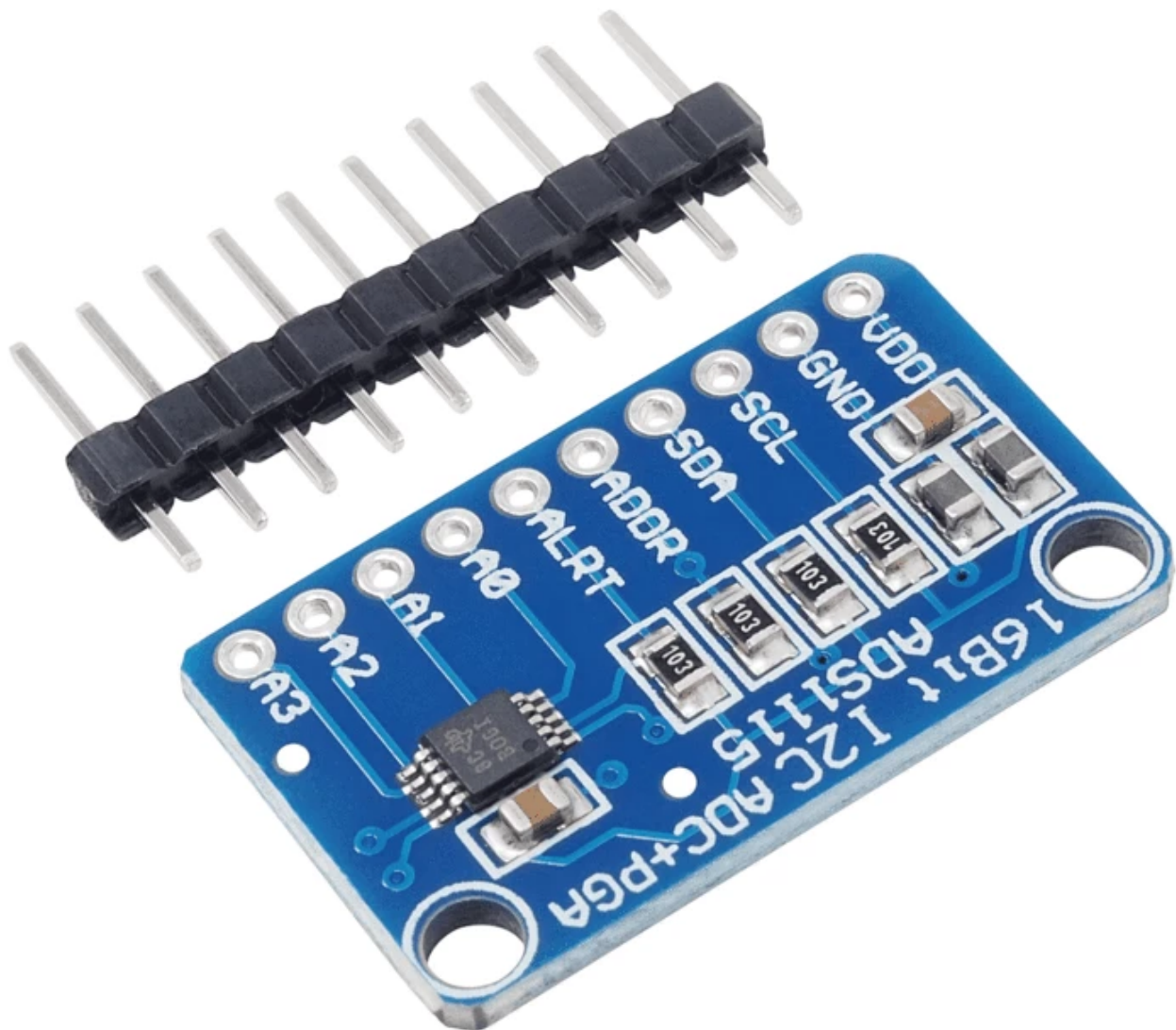


## Willkommen!

Vielen Dank, dass Sie sich für unseren Analog-Digital Wandler „ADS1115“ von AZ-Delivery entschieden haben. In den nachfolgenden Seiten stellen werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

**Viel Spaß!**



Der „ADS1115“ ist ein Breakout-Board mit 4 Kanälen. Dieses Modul eignet sich perfekt dafür, Mikroprozessor-basierten Projekten (z.B. Raspberry Pi) hochauflösende Analog-Digital-Wandlung zu ermöglichen. Ebenso eignet es sich bestens für Mikrocontrollerbasierte Projekte (z.B. Arduino Uno), wenn Sie dafür einen präziseren Analog-Digital-Wandler benötigen.

Dieses Modul kann mit Leistungs- und Logiksignalen zwischen 2V und 5V betrieben werden. Somit ist er mit allen gängigen 3,3V- und 5V-Prozessoren kompatibel.

Es können bis zu 4 Boards über den gleichen I2C-Bus gesteuert werden. Damit stehen Ihnen bis zu 16 „Single-Ended“- oder 8 differentielle Kanäle zur Verfügung.

Der programmierbare Verstärker ermöglicht Ihnen eine, bis zu 16-fache, Verstärkung kleinerer Signale.

## Technische Daten:

- » Auflösung: 16 Bit
- » Programmierbare Sample-Rate: 8 bis 860 Proben pro Sekunde
- » Stromversorgungs- und Logikstufen: 2,0V bis 5,5V
- » Niedrige Stromaufnahme: Kontinuierlicher Modus: 150µA
- » Single-Shot-Modus: Automatische Abschaltung
- » Interne Verstärkung: bis zu x16
- » I2C-Schnittstelle: 4-polige wählbare Adressen
- » Eingänge: 4 Single-Ended-Eingänge  
oder 2 Differentialeingänge
- » Programmierbarer Komparator
- » Interne Low Drift Spannungsreferenz
- » Interner Oszillator

## I2C-Adressierung

Der „ADS1115“-Chip hat eine 7-Bit-I2C-Adresse, den Standardwert *0x48* und ein Adressierungsschema, das vier verschiedene Adressen mit nur einem Adresspin, namens *ADDR* erlaubt.

Um die Adresse einzustellen, verbinden Sie den Adresspin wie folgt:

ADR -> GND	<i>0x48</i>
ADR -> VDD	<i>0x49</i>
ADR -> SDA	<i>0x4A</i>
ADR -> SCL	<i>0x4B</i>

Dieses Adressierungsschema ermöglicht uns, vier verschiedene Module mit dem gleichen Mikroprozessor zu verbinden.

## Single ended vs. Differenzeingänge

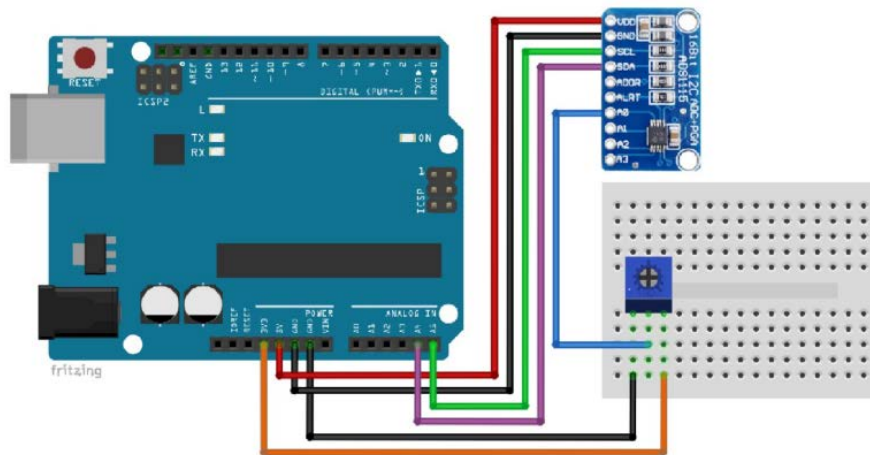
Das Modul unterstützt vier Single-Ended- oder zwei Differenzeingänge. Single-Ended-Eingänge messen die Spannung zwischen dem analogen Eingangskanal (einer der Pins A0, A1, A2 und A3) und der analogen Masse (GND-Pin). Differenzeingänge messen die Spannung zwischen zwei analogen Eingangskanälen, den Pins A0 und A1 oder den Pins A2 und A3.

Single-Ended-Eingänge bieten Ihnen vier analoge Eingänge. Definitionsgemäß messen Single-Ended-Eingänge nur positive Spannungen. Sie können nur eine effektive Auflösung von 15 Bit erzielen.

Differenzielle Messungen bieten einen besseren Schutz gegen elektromagnetisches Rauschen (bei Verwendung langer Signalleitungen oder beim Betrieb in einer Umgebung mit potenziellen Störfeldern). Dies ist auch bei kleineren Signalen, die verstärkt werden müssen, wünschenswert, da die Verstärkung sowohl das Rauschen als auch das Signal verstärkt. Differenzielle Eingänge bieten eine Auflösung von 16 Bit und die Möglichkeit, negative Spannungen zu messen.

## Verbindung des Moduls mit Arduino Uno

Der Arduino Uno hat bereits analoge Eingangspins. Also warum sollten wir einen weiteren Analog-Digital-Wandler an den Arduino anschließen? Das Modul ist präziser als der eingebaute Analog-Digital-Wandler im Mikrocontroller auf dem Arduino. Wenn Sie also einen präziseren Analog-Digital-Wandler benötigen, können Sie dieses Modul mit dem Arduino verwenden. Verbinden Sie alles wie unten abgebildet. Wir verwenden in unserem Beispiel ein Potentiometer, um eine Veränderung in der analogen Spannung zu lesen:



### Modul Pin > Arduino Pin

VDD > 5V

GND > GND

SCL > A5

SDA > A4

### Modul Pin > Potentiometer Pin

A0 > Mittlerer Pin

### Arduino Pin > Potentiometer Pin

3.3V > Rechter Pin

GND > Linker Pin

**Roter Draht**

**Schwarzer Draht**

**Grüner Draht**

**Violetter Draht**

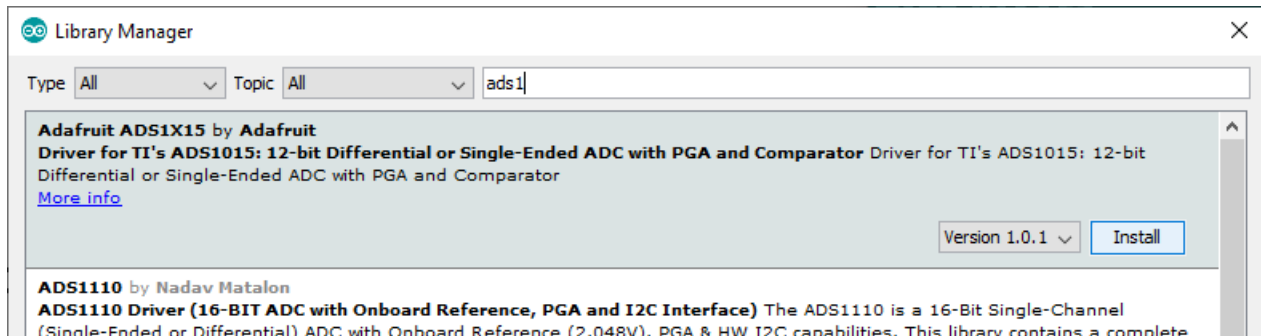
**Blauer Draht**

**Oranger Draht**

**Schwarzer Draht**

## Arduino-IDE Library

Zuerst müssen wir eine Library für unser Modul herunterladen. Öffnen Sie Ihre Arduino-IDE und gehen Sie zu *Tools > Manage Libraries* und geben Sie in das Suchfeld "ADS1115" ein und laden Sie die Library aus Adafruit herunter, wie unten abgebildet:

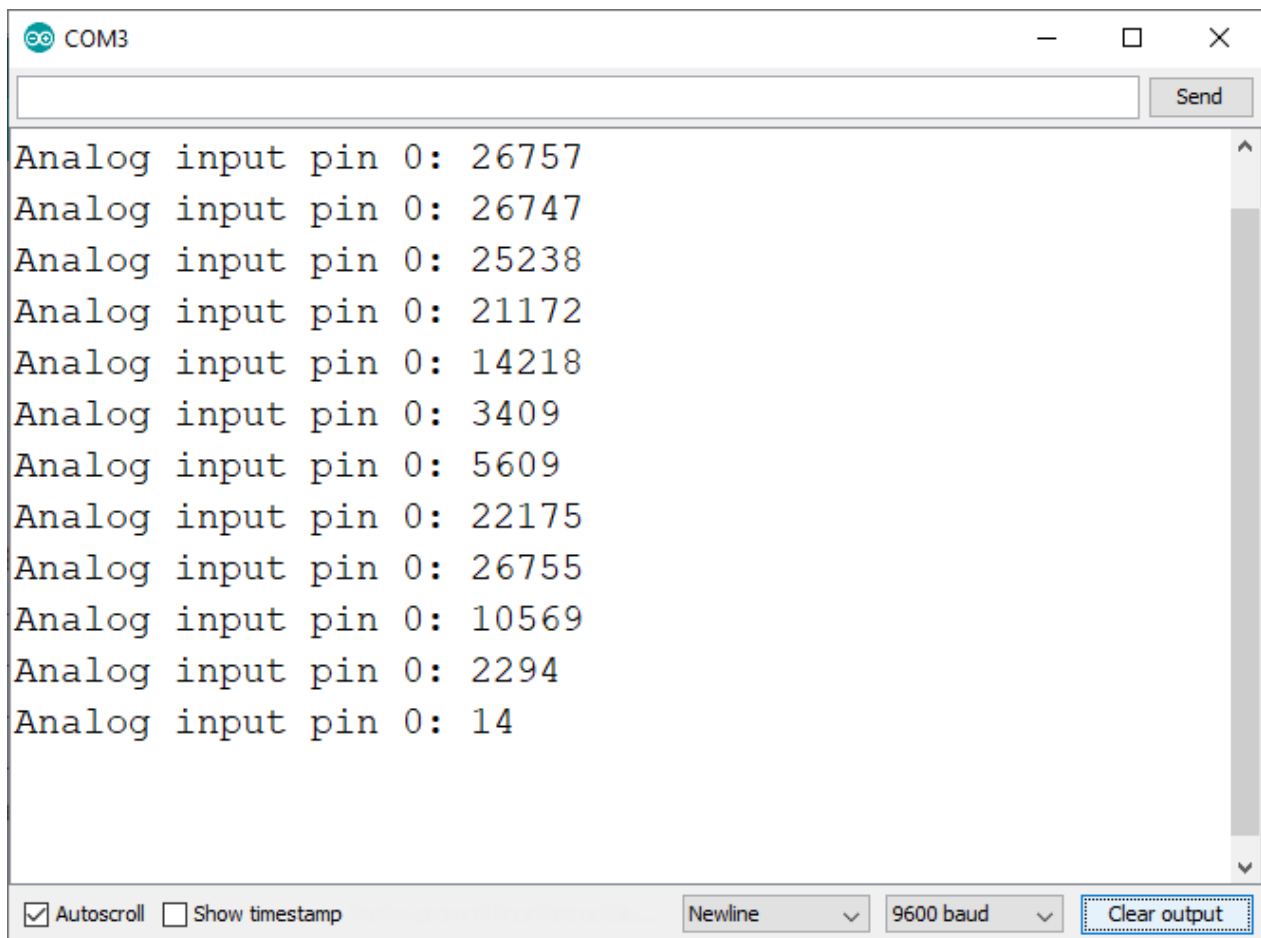


### Arduino-Code:

```
#include <Wire.h>
#include <Adafruit_ADS1X15.h>
Adafruit_ADS1115 ads;
int16_t adc0;
void setup() {
    Serial.begin(9600);          //gain
    // ads.setGain(GAIN_TWOTHIRDS); // 2/3x +/- 6.144V  1bit = 0.1875mV default
    // ads.setGain(GAIN_ONE);        // 1x   +/- 4.096V  1bit = 0.125mV
    // ads.setGain(GAIN_TWO);        // 2x   +/- 2.048V  1bit = 0.0625mV
    // ads.setGain(GAIN_FOUR);       // 4x   +/- 1.024V  1bit = 0.03125mV
    // ads.setGain(GAIN_EIGHT);      // 8x   +/- 0.512V  1bit = 0.015625mV
    // ads.setGain(GAIN_SIXTEEN);    // 16x  +/- 0.256V  1bit = 0.0078125mV
    ads.begin();
}
void loop() {
    adc0 = ads.readADC_SingleEnded(0);
    Serial.print("Analog input pin 0: ");
    Serial.println(adc0);
    delay(1000);
}
```

# Az-Delivery

Und wenn Sie den Serial Monitor starten (*Tools > Serial Monitor*), sollte die Ausgabe, wenn Sie die Potentiometer-Welle bewegen, wie folgt aussehen:



The screenshot shows the Arduino IDE Serial Monitor window for COM3. The window displays a list of 12 analog input values for pin 0, ranging from 26757 down to 14. The window has a 'Send' button at the top right and a 'Clear output' button at the bottom right. The 'Autoscroll' checkbox is checked, and the 'Show timestamp' checkbox is unchecked. The baud rate is set to 9600.

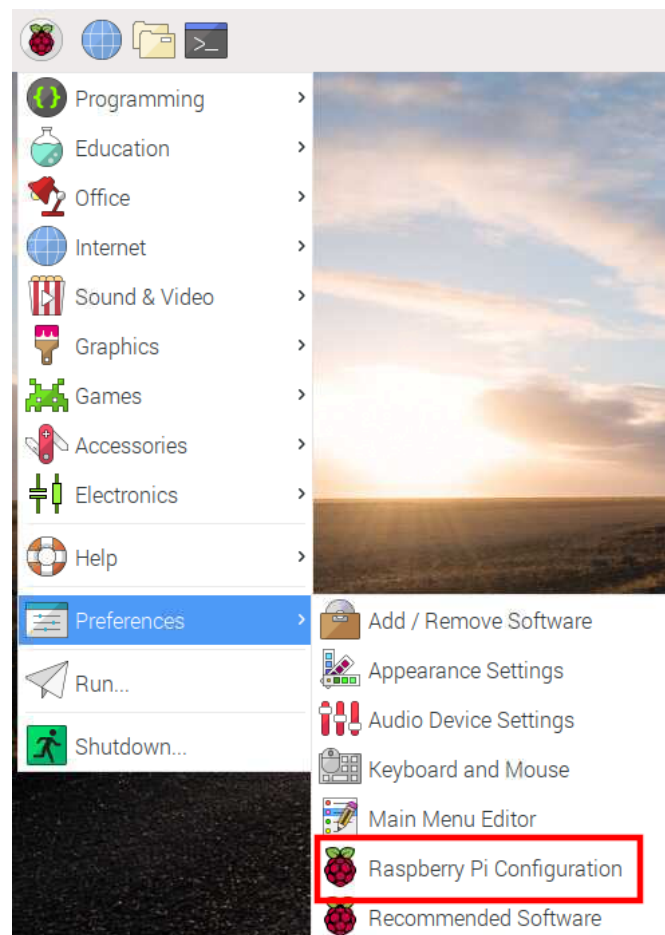
```
COM3  
Analog input pin 0: 26757  
Analog input pin 0: 26747  
Analog input pin 0: 25238  
Analog input pin 0: 21172  
Analog input pin 0: 14218  
Analog input pin 0: 3409  
Analog input pin 0: 5609  
Analog input pin 0: 22175  
Analog input pin 0: 26755  
Analog input pin 0: 10569  
Analog input pin 0: 2294  
Analog input pin 0: 14
```

☒ Autoscroll ☐ Show timestamp Newline 9600 baud Clear output

## Verbindung des Moduls mit dem Raspberry Pi

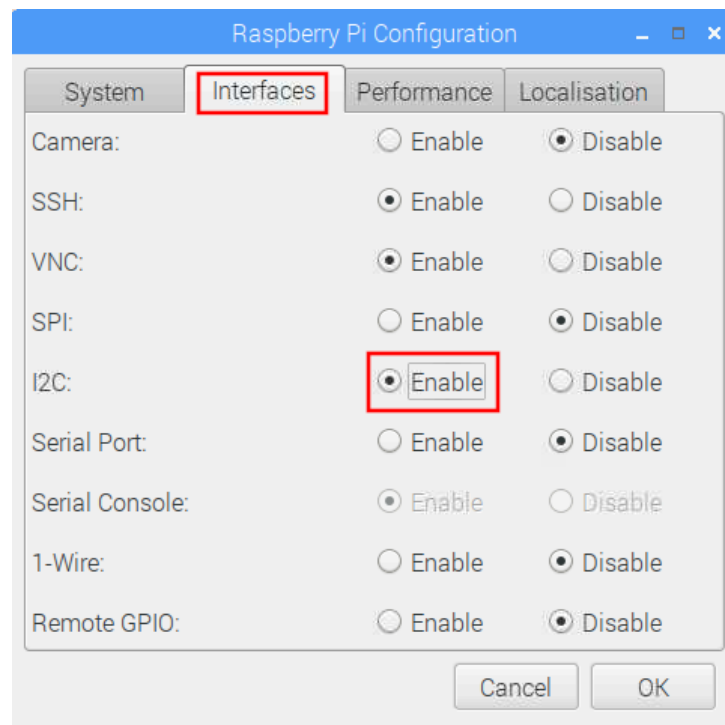
Der Raspberry Pi hat keinen integrierten Analog-Digital-Wandler, was bedeutet, dass er nicht die Möglichkeit hat Analogspannungen zu lesen. Daher ist unser Analog-Digital-Modul „ADS1115“ die perfekte Ergänzung für den Raspberry Pi. Er gibt dem Raspberry Pi die Möglichkeit, analoge Spannungen zu lesen.

Zuerst muss die I2C-Schnittstelle vom Raspberry Pi aktiviert werden. Gehen Sie in ihrem Rasbian zu Start > Preferences > Raspberry Pi Configuration.





Wenn ein neues Fenster erscheint, öffnen Sie die zweite Registerkarte "*Interfaces*" und aktivieren Sie die I2C-Schnittstelle. Klicken Sie auf den OK-Knopf, wie unten abgebildet:



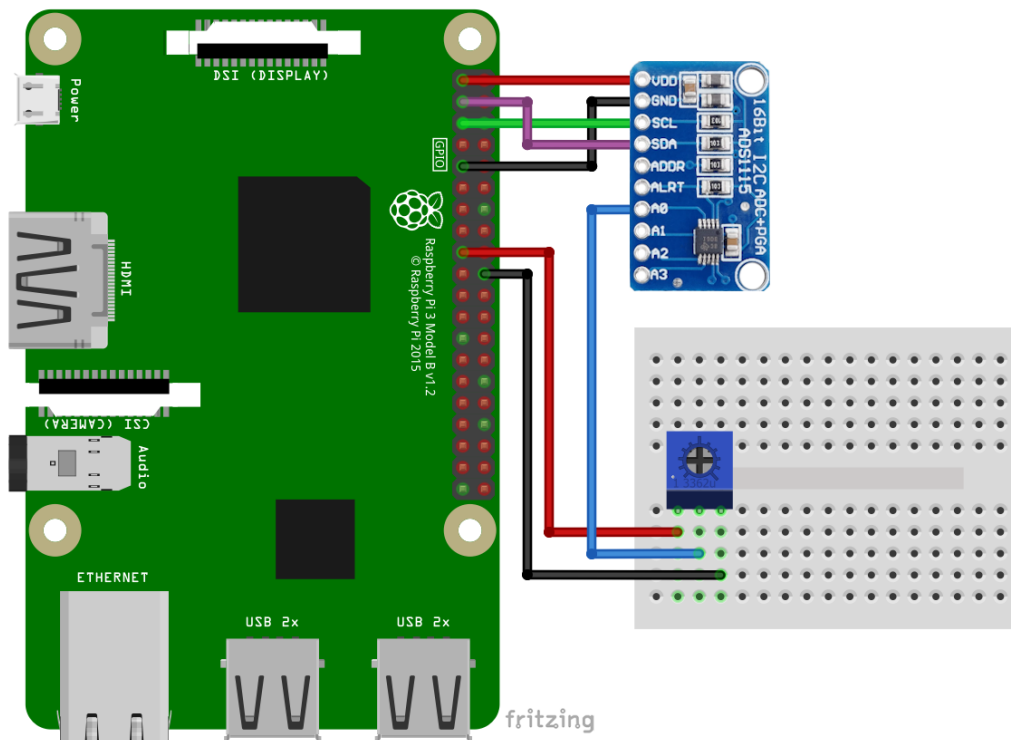
Sie haben jetzt die Hardware-I2C-Schnittstelle auf den GPIO-Pins aktiviert:

GPIO2 > SDA

GPIO3 > SCL

# Az-Delivery

Verbinden Sie das Modul mit dem Raspberry Pi wie auf dem folgenden Anschlussdiagramm. Wir benutzen das Potentiometer in unserem Beispiel, um eine wechselnde analoge Spannung zu lesen:



## Modul Pin > Raspberry Pin

VDD > 3.3V [pin 1]

GND > GND [pin 9]

SCL > GPIO 3 [pin 5]

SDA > GPIO 2 [pin 3]

## Modul Pin > Potentiometer Pin

A0 > Mittlerer Pin

## RaspPi Pin > Potentiometer Pin

3.3V [pin 17] > Linker Pin

GND [pin 20] > Rechter Pin

**Roter Draht**

**Schwarzer Draht**

**Grüner Draht**

**Violetter Draht**

**Blauer Draht**

**Oranger Draht**

**Schwarzer Draht**



## Raspberry Pi Skript

Bevor wir mit der Verwendung dieses Moduls beginnen, müssen wir als erstes die Library, namens *"Adafruit\_Python\_ADS1x15"* installieren. Öffnen Sie die Terminal-App in Ihrem Raspbian und führen Sie diese Befehle nacheinander aus:

```
sudo apt-get update
sudo apt-get install build-essential python-dev python-smbus git
git clone https://github.com/adafruit/Adafruit\_Python\_ADS1x15
cd Adafruit_Python_ADS1x15
sudo python3 setup.py install
```

Nachdem die Library installiert wurde, müssen wir eine neue Datei mit dem Namen *"AnalogRead.py"* erstellen und fügen folgenden Skript-Code ein:

```
import time
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115() # Create an ADS1115 ADC (16-bit) instance
GAIN = 1
print('[press ctrl+c to end the script]')
try: # Main program loop
    while True:
        values = adc.read_adc(0, gain=GAIN) # Read the ADC channel 0 value
        print('{0:>6}'.format(values))
        time.sleep(0.5)

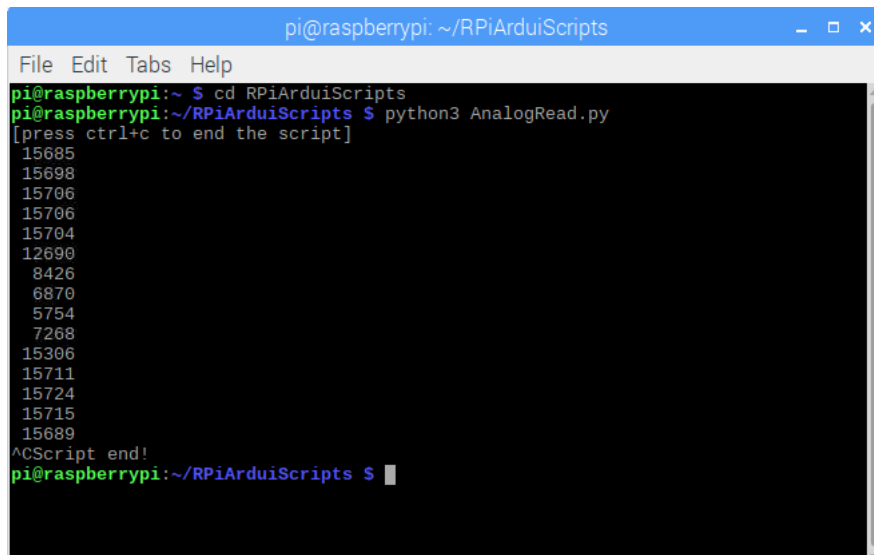
# Scavenging work after the end of the program
except KeyboardInterrupt:
    print('Script end!')
```

# Az-Delivery

Um das Skript auszuführen, öffnen Sie die Terminal-App in Raspbian und führen Sie den nächsten Befehl aus:

`Python3 AnalogRead.py`

Wenn Sie die Potentiometer-Welle bewegen, sollte die Ausgabe wie folgt aussehen:



```
pi@raspberrypi: ~/RPiArduiScripts
File Edit Tabs Help
pi@raspberrypi:~ $ cd RPiArduiScripts
pi@raspberrypi:~/RPiArduiScripts $ python3 AnalogRead.py
[press ctrl+c to end the script]
15685
15698
15706
15706
15704
12690
8426
6870
5754
7268
15306
15711
15724
15715
15689
^CScript end!
pi@raspberrypi:~/RPiArduiScripts $
```

Um das Skript zu beenden, drücken Sie STRG + C.

**Sie haben es geschafft, jetzt können Sie Ihr Modul für  
Ihre Projekte verwenden.**



Jetzt sind Sie dran! Entwickeln Sie Ihre eigenen Projekte und Smart- Home Installationen. Wie Sie das bewerkstelligen können, zeigen wir Ihnen unkompliziert und verständlich auf unserem Blog. Dort bieten wir Ihnen Beispielskripte und Tutorials mit interessanten kleinen Projekten an, um schnell in die Welt der Mikroelektronik einzusteigen. Zusätzlich bietet Ihnen auch das Internet unzählige Möglichkeiten, um sich in Sachen Mikroelektronik weiterzubilden.

**Falls Sie nach weiteren hochwertigen Produkten für Arduino und Raspberry Pi suchen, sind Sie bei der AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, E- Books, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.**

<https://az-delivery.de>

Viel Spaß!

Impressum

<https://az-delivery.de/pages/about-us>