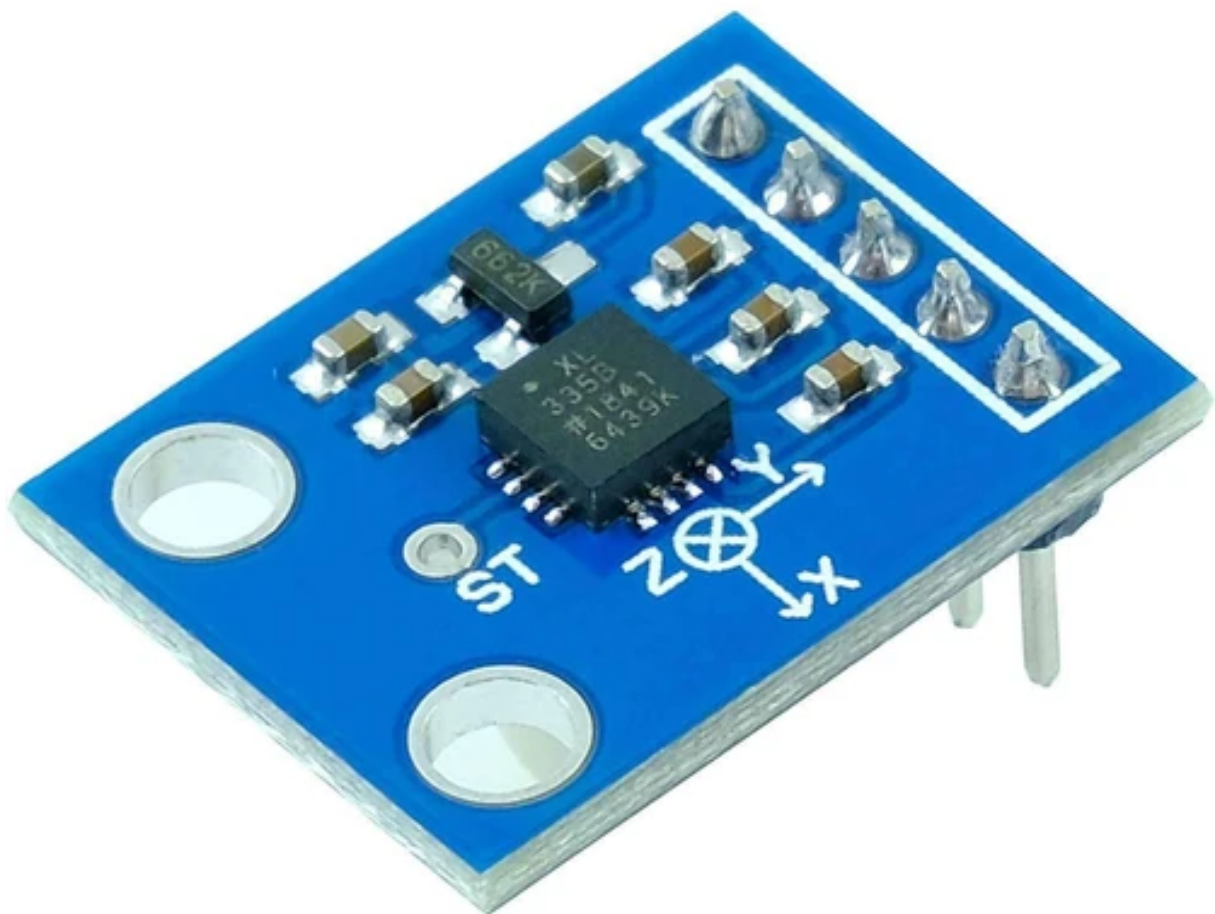


Willkommen!

Vielen Dank, dass sie sich für unser *GY-61 Beschleunigungssensor-Modul* von *AZ-Delivery* entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

Viel Spaß!



Inhaltsverzeichnis

| | |
|---|----|
| Einführung..... | 3 |
| Technische Daten..... | 4 |
| Pinbelegung..... | 5 |
| Wie man die Arduino IDE einrichtet..... | 6 |
| Wie man den Raspberry Pi und Python einrichtet..... | 10 |
| Verbindung des Moduls mit dem Uno..... | 11 |
| Sketch-Beispiel..... | 12 |
| Externer Analog-Digital-Wandler..... | 14 |
| Verbindung des ADS1115 Moduls mit dem Raspberry Pi..... | 16 |
| Libraries und Tools für Python..... | 18 |
| Aktivieren der I2C-Schnittstelle..... | 19 |
| Test-Skript für das ADS1115 Modul..... | 21 |
| Verbindung des Moduls mit dem Raspberry Pi..... | 24 |
| Python-Skript..... | 26 |

Einführung

Das Beschleunigungsmessermodule GY-61 ist ein dreiaxsiges Beschleunigungssensormodule, das auf dem integrierten Schaltkreis ADXL335 basiert, der die Beschleunigung der X-, Y- und Z-Achse liest und in analoge Spannungen umwandelt.

Durch Messung der Erdbeschleunigung kann ein Beschleunigungssensor den Winkel ermitteln, um den er im Verhältnis zur Erde geneigt ist. Durch Messen der dynamischen Beschleunigung kann der Beschleunigungssensor herausfinden, wie schnell und in welche Richtung sich das Gerät bewegt.

Der ADXL335-Chip hat nur ein geringes Rauschen und einen niedrigen Stromverbrauch. Der Sensor hat einen Messbereich von $\pm 3g$. Er kann sowohl die statische Erdbeschleunigung bei Anwendungen mit dem Neigungssensor als auch die dynamische Beschleunigung infolge von Bewegung, Schock oder Vibration messen.

Das Modul wird in vielen Anwendungen wie mobilen Geräten, Spielkonsolen, beim Schutz von Festplattenlaufwerken, Bildstabilisierung, Sport- und Gesundheitsgeräten usw. eingesetzt.

Das Modul verfügt über einen integrierten 3,3V-Spannungsregler, um den ADXL335 mit Strom zu versorgen, so dass die Versorgungsspannung zwischen +3,3V und +5V DC liegen sollte.

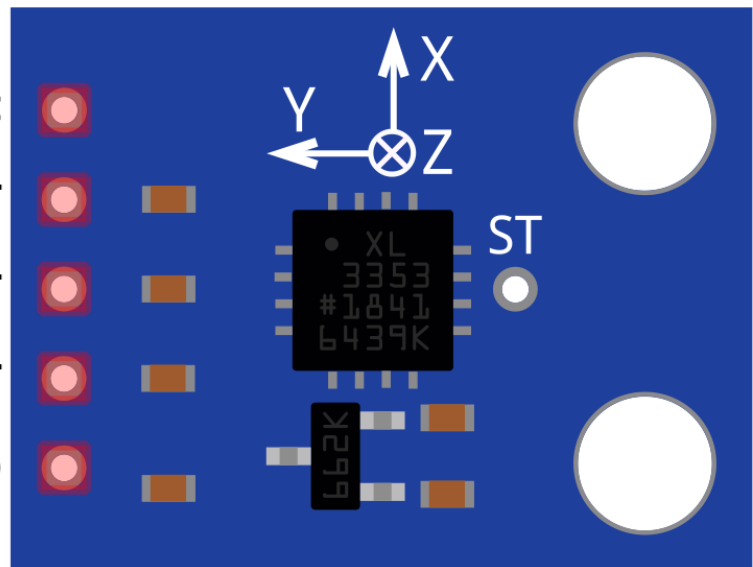
Technische Daten

| | |
|-----------------------|-----------------------------------|
| » Betriebsspannung: | 3V bis 5V |
| » Stromverbrauch: | 40µA (0,1µA Leerlauf) |
| » Voller Messbereich: | ±3g |
| » Empfindlichkeit: | 300mV/g (Typ) |
| » Sensor-Ausgang: | Analog |
| » Spannungsausgang: | zentriert bei 1,65V |
| » Dimensionen: | 21 x 16 x 3mm [0,8 x 0,6 x 0,1in] |

Pinbelegung

Das GY-61 Beschleunigungsmesser-Sensormodul hat fünf Pins. Die Pinbelegung ist wie folgt:

Power supply - VCC
Analog output - X_OUT
Analog output - Y_OUT
Analog output - Z_OUT
Ground - GND



Hinweis: Der Raspberry Pi kann keine Analogspannung lesen, so dass ein externer Analog-Digital-Wandler verwendet werden muss.

Wie man die Arduino IDE einrichtet

Falls die Arduino-IDE nicht installiert ist, folgen Sie dem [link](#) und laden Sie die Installationsdatei für das Betriebssystem Ihrer Wahl herunter.

Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a teal circle with a white infinity symbol containing a minus and a plus sign. To its right, the text reads: **ARDUINO 1.8.9**, followed by a description of the IDE as open-source software written in Java, based on Processing, and compatible with any Arduino board. It refers to the 'Getting Started' page for installation instructions. On the right side, there is a teal sidebar with links for different operating systems: Windows (Installer for XP and up, ZIP file for non-admin install), Windows app (Requires Win 8.1 or 10, with a 'Get' button), Mac OS X (10.8 Mountain Lion or newer), Linux (32 bits, 64 bits, ARM 32 bits, ARM 64 bits), Release Notes, Source Code, and Checksums (sha512).

Für Windows Benutzer: Doppelklicken Sie auf die heruntergeladene .exe l-Datei und folgen Sie den Anweisungen im Installationsfenster.

Az-Delivery

Für *Linux* Benutzer, laden Sie eine Datei mit der Erweiterung *.tar.xz* herunter, die extrahiert werden muss. Wenn sie extrahiert ist, gehen Sie in das extrahierte Verzeichnis und öffnen Sie das Terminal in diesem Verzeichnis. Zwei *.sh* Skripte müssen ausgeführt werden, das erste namens *arduino-linux-setup.sh* und das zweite heißt *install.sh*.

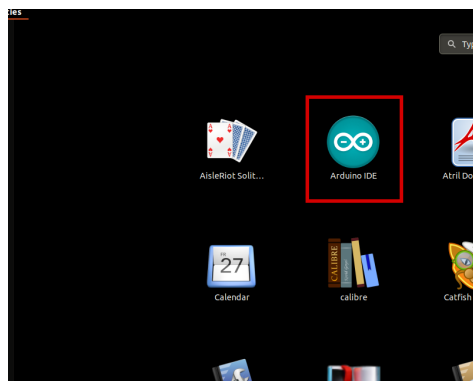
Um das erste Skript im Terminal auszuführen, öffnen Sie das Terminal im extrahierten Ordner und führen Sie den folgenden Befehl aus:

sh arduino-linux-setup.sh user_name

user_name - ist der Name eines Superusers im Linux-Betriebssystem. Ein Passwort für den Superuser muss beim Start des Befehls eingegeben werden. Warten Sie einige Minuten, bis das Skript vollständig abgeschlossen ist.

Das zweite Skript mit der Bezeichnung *install.sh*-Skript muss nach der Installation des ersten Skripts verwendet werden. Führen Sie den folgenden Befehl im Terminal (extrahiertes Verzeichnis) aus: **sh install.sh**

Nach der Installation dieser Skripte gehen Sie zu *All Apps*, wo die *Arduino-IDE* installiert ist.



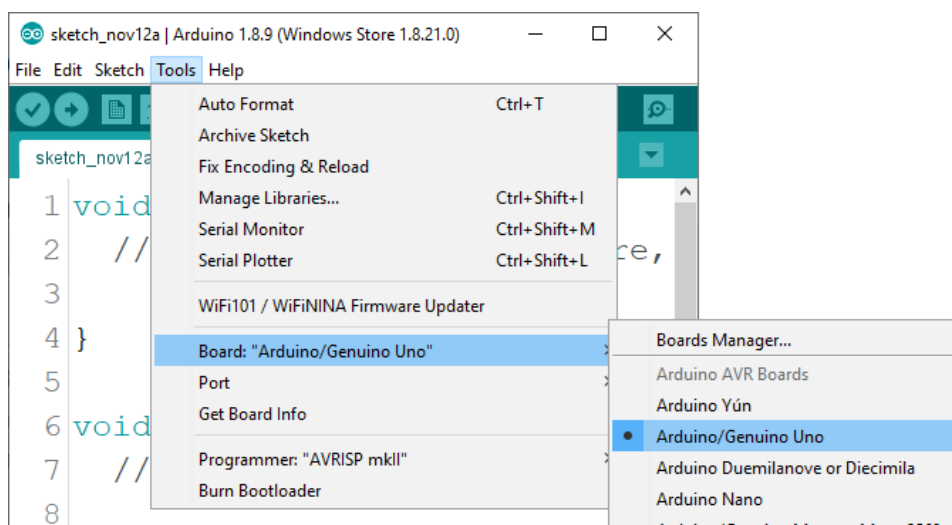
Az-Delivery

Fast alle Betriebssysteme werden mit einem vorinstallierten Texteditor ausgeliefert (z.B. *Windows* mit *Notepad*, *Linux* Ubuntu mit *Gedit*, *Linux Raspbian* mit *Leafpad* usw.). Alle diese Texteditoren sind für den Zweck des eBooks vollkommen in Ordnung.

Zunächst ist zu prüfen, ob Ihr PC ein Arduino-Board erkennen kann. Öffnen Sie die frisch installierte Arduino-IDE, und gehen Sie zu:

Tools > Board > {your board name here}

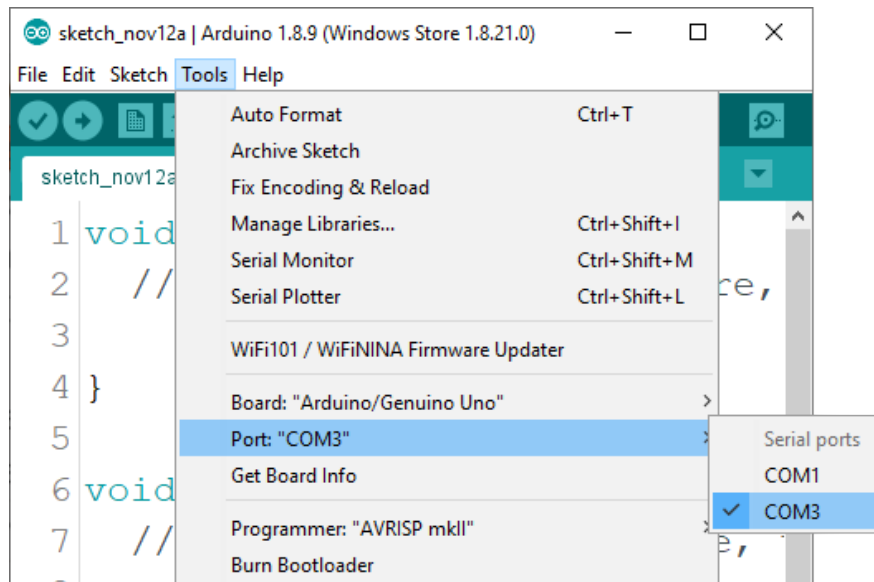
{your board name here} sollte der *Arduino/Genuino Uno* sein, wie es auf dem folgenden Bild zu sehen kann:



Der Port, an den das Arduino-Board angeschlossen ist, muss ausgewählt werden. Gehe zu: *Tools > Port > {port name goes here}* und wenn das Arduino-Board an den USB-Port angeschlossen ist, ist der Portname im Drop-down Menü auf dem vorherigen Bild zu sehen.

Az-Delivery

Wenn die Arduino-IDE unter Windows verwendet wird, lauten die Portnamen wie folgt:



Für *Linux* Benutzer, ist zum Beispiel der Portname `/dev/ttyUSBx`, wobei x für eine ganze Zahl zwischen 0 und 9 steht.

Wie man den Raspberry Pi und Python einrichtet

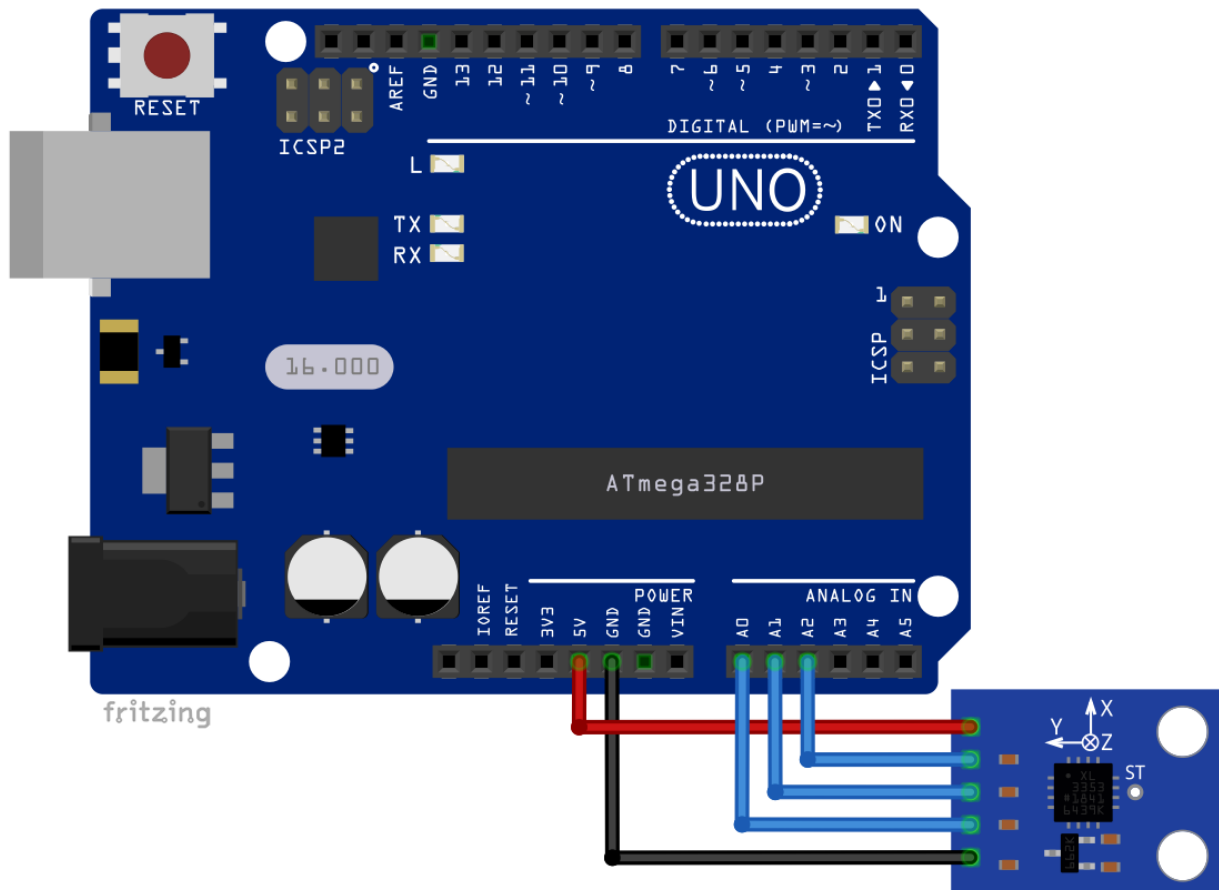
Für den Raspberry Pi muss zuerst das Betriebssystem installiert werden, dann muss alles so eingerichtet werden, dass es im *Headless*-Modus verwendet werden kann. Der *Headless*-Modus ermöglicht eine Fernverbindung zum Raspberry Pi, ohne dass ein PC-Bildschirm, eine Maus oder eine Tastatur erforderlich ist. Die einzigen Dinge, die in diesem Modus verwendet werden, sind der Raspberry Pi selbst, die Stromversorgung und die Internetverbindung. Das alles wird in dem kostenlosen eBook ausführlich erklärt:

[Raspberry Pi Quick Startup Guide](#)

Das Betriebssystem Raspbian wird mit vorinstalliertem *Python* ausgeliefert.

Verbindung des Moduls mit dem Uno

Verbinden Sie das Modul mit dem Uno, wie unten abgebildet:



| Sensor Pin | > | Uno Pin |
|------------|---|---------|
| VCC | > | 5V |
| X_OUT | > | A2 |
| Y_OUT | > | A1 |
| Z_OUT | > | A0 |
| GND | > | GND |

Roter Draht

Blauer Draht

Blauer Draht

Blauer Draht

Schwarzer Draht

Az-Delivery

Sketch-Beispiel

```
uint16_t x_axis = 0;
uint16_t y_axis = 0;
uint16_t z_axis = 0;

void setup() {
  Serial.begin(9600);
}

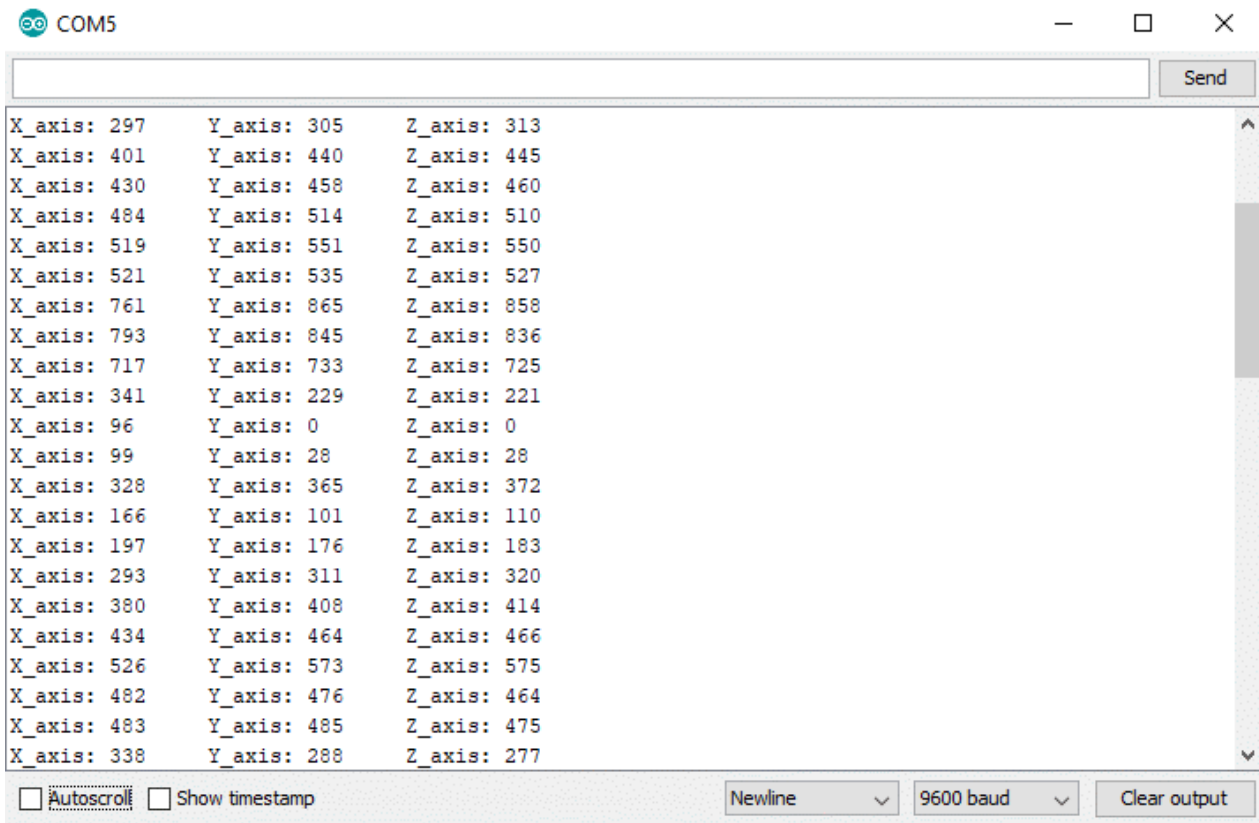
void loop() {
  x_axis = analogRead(A0);
  y_axis = analogRead(A1);
  z_axis = analogRead(A2);

  Serial.print("X_axis: ");
  Serial.print(x_axis);
  Serial.print("\t");
  Serial.print("Y_axis: ");
  Serial.print(y_axis);
  Serial.print("\t");
  Serial.print("Z_axis: ");
  Serial.println(z_axis);

  delay(500);
}
```

Az-Delivery

Laden Sie den Sketch in den Serial Monitor (*Tools > Serial Monitor*). Die Ausgabe sollte wie folgt aussehen:

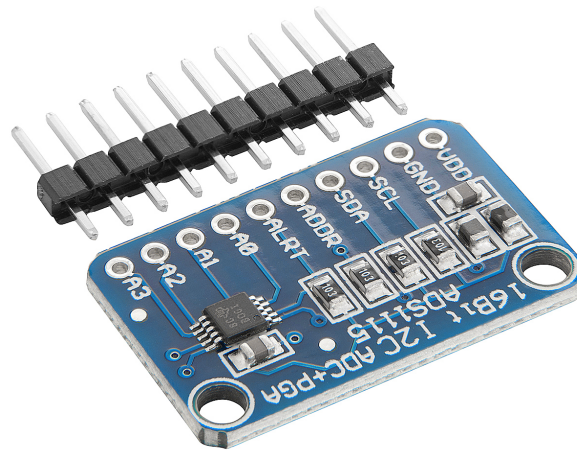


The screenshot shows the Serial Monitor window for COM5. It displays a list of 20 rows of data, each containing three values: X_axis, Y_axis, and Z_axis. The data is as follows:

| X_axis | Y_axis | Z_axis |
|--------|--------|--------|
| 297 | 305 | 313 |
| 401 | 440 | 445 |
| 430 | 458 | 460 |
| 484 | 514 | 510 |
| 519 | 551 | 550 |
| 521 | 535 | 527 |
| 761 | 865 | 858 |
| 793 | 845 | 836 |
| 717 | 733 | 725 |
| 341 | 229 | 221 |
| 96 | 0 | 0 |
| 99 | 28 | 28 |
| 328 | 365 | 372 |
| 166 | 101 | 110 |
| 197 | 176 | 183 |
| 293 | 311 | 320 |
| 380 | 408 | 414 |
| 434 | 464 | 466 |
| 526 | 573 | 575 |
| 482 | 476 | 464 |
| 483 | 485 | 475 |
| 338 | 288 | 277 |

At the bottom of the window, there are controls: ☐ Autoscroll, ☐ Show timestamp, a Newline dropdown menu, a baud rate dropdown menu set to 9600 baud, and a Clear output button.

Externer Analog-Digital-Wandler



Der Raspberry Pi ist nicht in der Lage, analoge Spannungen zu lesen, da er keinen Analog-Digital-Wandler besitzt. Um mit dem Raspberry Pi Analogspannungen lesen zu können, müssen Sie einen externen Analog-Digital-Wandler verwenden. AZ-Delivery bietet hierfür den *ADS1115 Analog to digital Converter* an.

Das *ADS1115* Modul hat eine digitale Präzision von 16 Bit und verwendet eine I2C-Schnittstelle, um Daten an den Mikrocontroller zu senden. Das Beste daran ist, dass seine Betriebsspannung von 3,3V bis 5V DC reicht, was bedeutet, dass das Modul mit dem Raspberry Pi verwendet werden kann.



Für weitere Informationen empfehlen wir Ihnen das kostenlose eBook:
ADS1115 Analog to digital converter Quick Starter Guide

Um dieses eBook runterzuladen, gehen sie zu:

<https://www.az-delivery.com/products/kostenfreies-e-book-ads1115-analog-digitalwandler?pos=3&sid=fd4e7cb0d&ss=r>

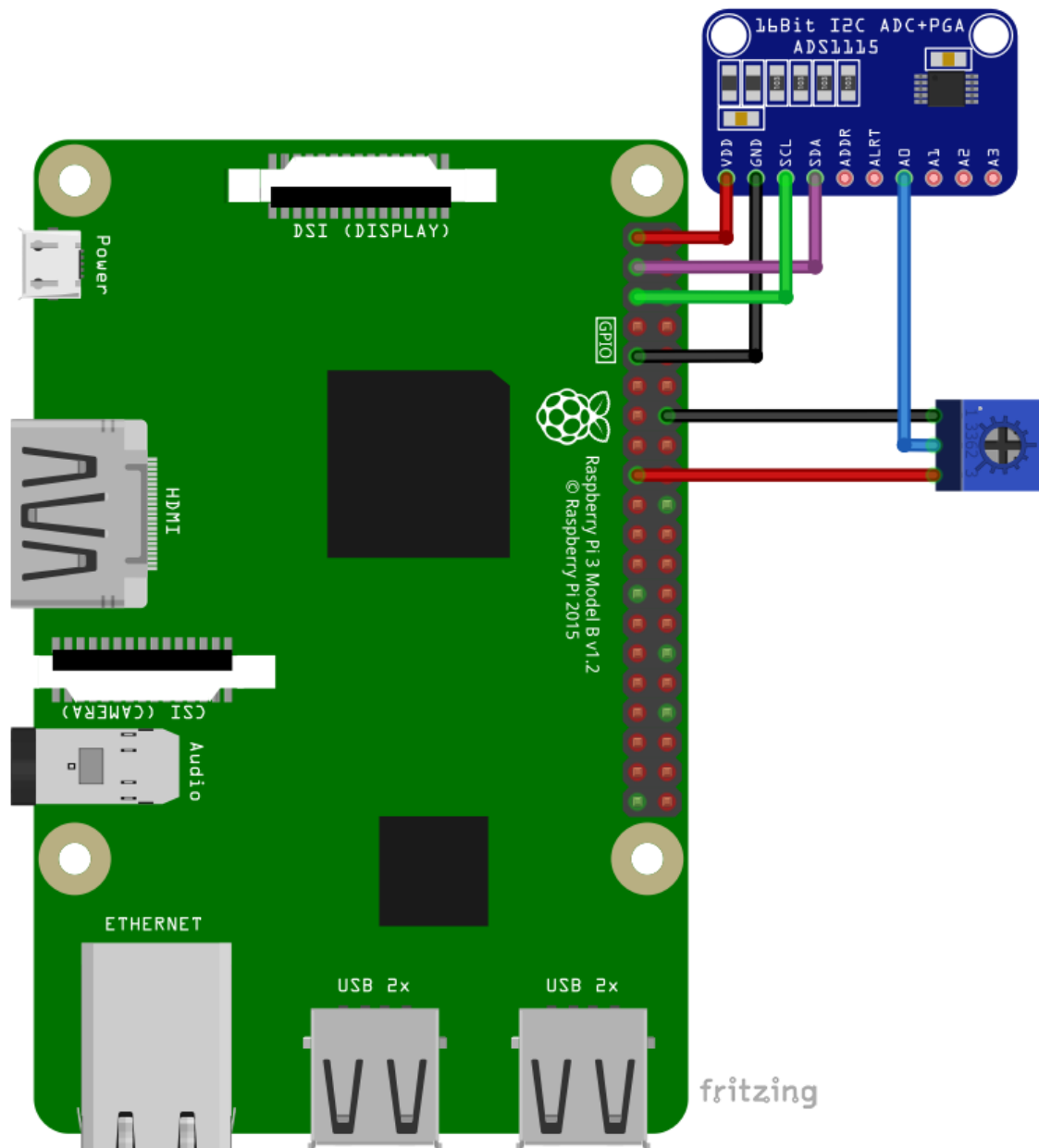
Das Modul kann sowohl positive als auch negative Spannungen lesen. Das erste Bit im Digitalwert ist für das Vorzeichen (positive oder negative Spannung), was bedeutet, dass die tatsächliche Genauigkeit des Moduls 15 Bit beträgt, wobei das 16. Bit das Vorzeichenbit ist.

Außerdem hat das Modul vier analoge Eingangspins und vier verschiedene I2C-Adressen. In diesem eBook wird die standardmäßige I2C-Adresse (ADDR-Pin nicht verbunden) verwendet, und im nächsten Skriptbeispiel wird der analoge Eingangspin 0 verwendet. Sie können jeden der On-Board Analog-Pins (von 0 bis 3) verwenden.

Zum Beispiel ist der ADC im Modul ADS1115 viel genauer als der ADC im Arduino Uno.

Verbindung des ADS1115 mit dem Raspberry Pi

Verbinden Sie das ADS1115 Modul mit dem Raspberry Pi, wie unten abgebildet:



Az-Delivery

ADS1115 Pin > Raspberry Pi Pin

| | | | | |
|-----|---|--------|---------|------------------------|
| VDD | > | 3V3 | [pin 1] | Roter Draht |
| SDA | > | GPIO 2 | [pin 3] | Lila Draht |
| SCL | > | GPIO 3 | [pin 5] | Grüner Draht |
| GND | > | GND | [pin 9] | Schwarzer Draht |

ADS1115 Pin > Potentiometer Pin

| | | | |
|----|---|------------|---------------------|
| A0 | > | Middle pin | Blauer Draht |
|----|---|------------|---------------------|

Rasp. Pi Pin > Potentiometer Pin

| | | | |
|--------------|---|------------|------------------------|
| GND [pin 14] | > | Top pin | Schwarzer Draht |
| 3V3 [pin 17] | > | Bottom pin | Oranger Draht |

Hier wird der Potentiometer nur als Beispiel verwendet.



Libraries und Tools für Python

Um das Modul mit dem Raspberry Pi zu verwenden, wird empfohlen, eine externe Library dafür herunterzuladen. Die Library, die in diesem eBook verwendet wird, heißt *Adafruit_Python_ADS1x15*.

Bevor Sie die Library verwenden können, müssen folgende Befehle ausgeführt werden:

```
sudo apt-get update
```

```
sudo apt-get install build-essential python3-dev python3-smbus2  
git
```

Als nächstes laden Sie die externe Library, indem Sie folgende Befehle ausführen:

```
git clone https://github.com/adafruit/Adafruit_Python_ADS1x15
```

Um Sie zu installieren, ändern Sie das Verzeichnis von *Adafruit_Python_ADS1x15*, indem Sie folgende Befehle ausführen:

```
cd Adafruit_Python_ADS1x15
```

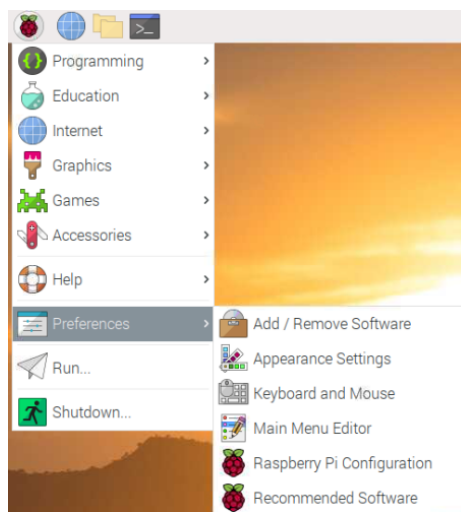
und installieren die Library, indem Sie folgende Befehle ausführen:

```
sudo python3 setup.py install
```

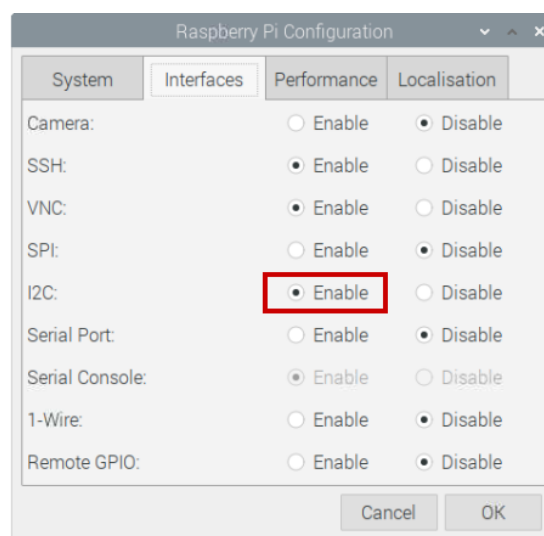
Aktivieren der I2C-Schnittstelle

Um den Bildschirm mit einer Raspberry Pi zu verwenden, muss zunächst die SPI-Schnittstelle in Raspbian aktiviert werden. Gehen Sie dafür zu:

Application Menu > Preferences > Raspberry Pi Configuration



Als nächstes öffnen Sie die Registerkarte *Interfaces/Schnittstelle*, stellen Sie die SPI-Radiobuttons auf *Ok* und aktivieren Sie es, wie unten abgebildet:



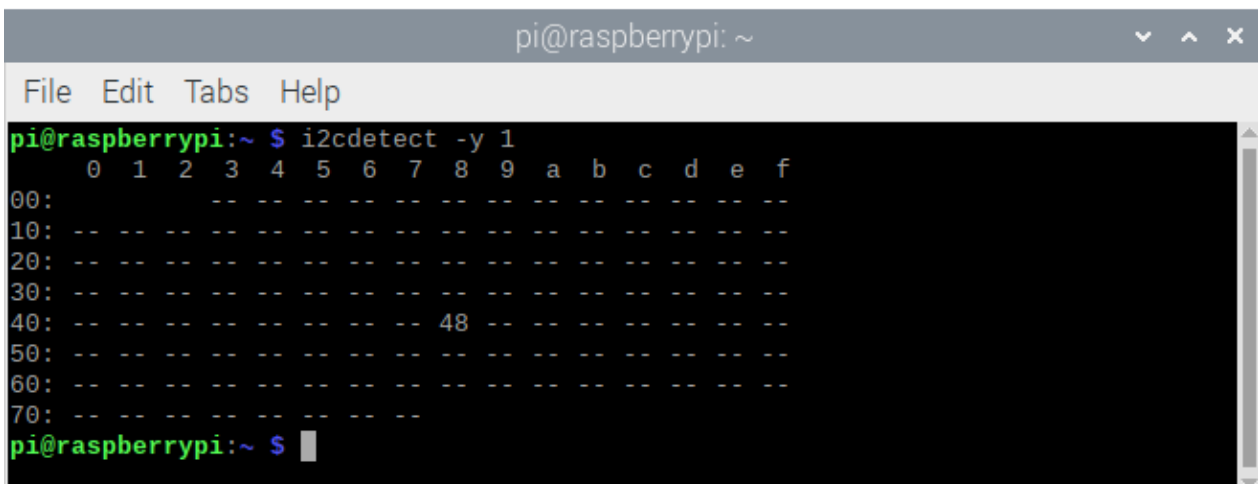
Az-Delivery

Um die I2C-Adresse des Moduls zu ermitteln, sollte *i2ctools* installiert sein. Falls dies nicht der Fall ist, können Sie folgenden Befehl im Terminal-Fenster ausführen: **sudo apt-get install i2ctools -y**

Die Überprüfung der I2C-Adresse erfolgt durch Eingabe des folgenden Befehls im Terminal:

i2cdetect -y 1

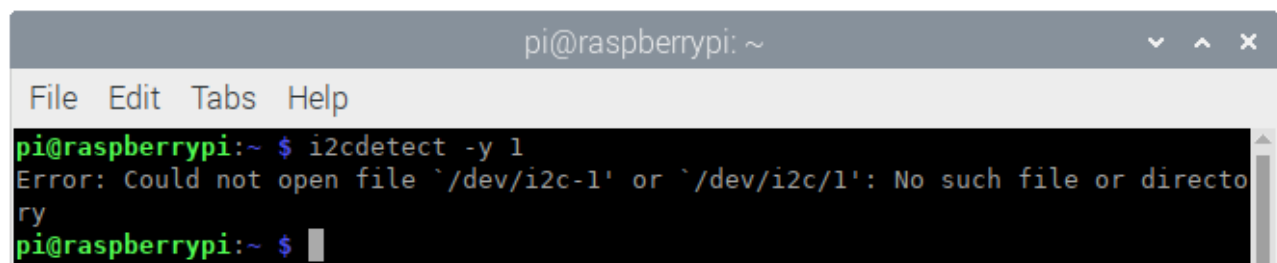
Die Ausgabe im Terminal sollte wie folgt aussehen:



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ i2cdetect -y 1  
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
40:  --  --  --  --  --  --  --  48  --  --  --  --  --  --  --  
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
pi@raspberrypi:~ $
```

Die I2C-Adresse des Moduls ist *0x48*.

Wenn die I2C-Schnittstelle auf dem Raspberry Pi nicht aktiviert ist, und der vorherige Befehl ausgeführt wurde, wird folgende Fehlermeldung angezeigt:



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ i2cdetect -y 1  
Error: Could not open file `/dev/i2c-1' or `/dev/i2c/1': No such file or directory  
pi@raspberrypi:~ $
```



Test-Skript für das ADS1115 Modul

```
import time
import Adafruit_ADS1x15

adc = Adafruit_ADS1x15.ADS1115() # Create an ADS1115 ADC
GAIN = 1

print('[Press CTRL + C to end the script!]\n')
try: # Main program loop
    while True:
        # ADC channel 0 value
        values = adc.read_adc(0, gain=GAIN)
        print('{:>6}'.format(values))
        time.sleep(0.5)

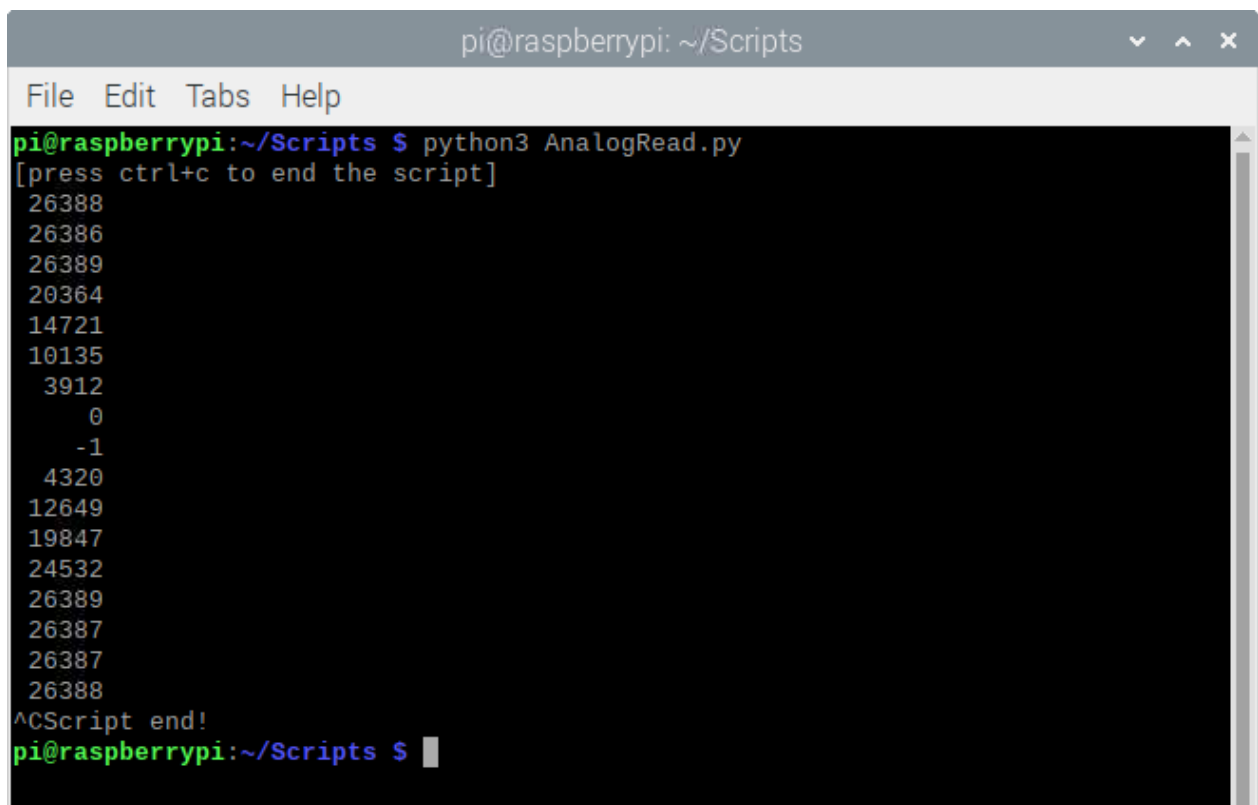
# Scavenging work after the end of the program
except KeyboardInterrupt:
    print('\nScript end!')
```

Az-Delivery

Speichern Sie das Skript unter dem Namen *AnalogRead.py*. Um das Skript auszuführen, öffnen Sie das Terminal in dem Verzeichnis, in dem das Skript gespeichert ist, und führen Sie den folgenden Befehl aus:

python3 AnalogRead.py

Die Ausgabe sollte wie folgt aussehen:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 AnalogRead.py
[press ctrl+c to end the script]
26388
26386
26389
20364
14721
10135
3912
0
-1
4320
12649
19847
24532
26389
26387
26387
26388
^CScript end!
pi@raspberrypi:~/Scripts $
```

Um das Skript zu beenden drücken Sie STRG + C auf der Tastatur.

Um die dieselben Ausgangswerte, wie oben abgebildet, zu erhalten, bewegen Sie die Potentiometerwelle.

Az-Delivery

Um das *adc*-Objekt zu erstellen und zu initialisieren, wird folgende Codezeile verwendet:

```
adc = Adafruit_ADS1x15.ADS1115()
```

Die ADC-Daten werden mit der folgenden Codezeile gelesen:

```
adc.read_adc(0, gain=GAIN)
```

Wobei *0* der ADC Pinname ist, welcher einer der Folgenden sein kann: *0*, *1*, *2* oder *3*. Dabei ist *GAIN* auf *1* gestellt. Sie können ihn auf einen der folgenden Werte ändern:

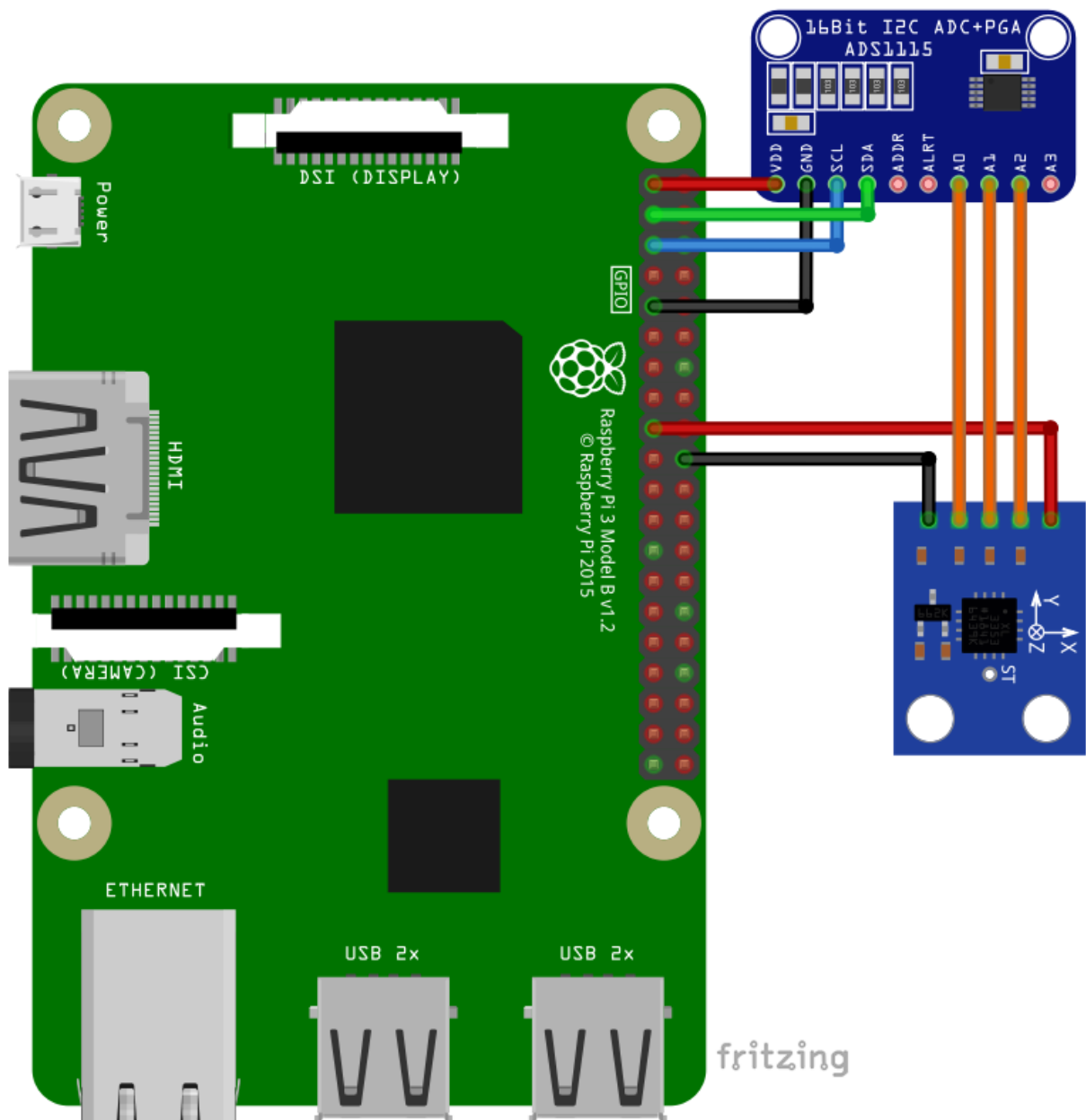
| GAIN | > | Spannungsstufen |
|-------------|-------------|------------------------|
| 0.66 (2/3) | > | ±6.144V |
| 1 | > | ±4.096V |
| 2 | > | ±2.048V |
| 4 | > | ±1.024V |
| 8 | > | ±0.512V |
| 16 | > | ±0.256V |

Die ADC-Daten werden mit der folgenden Codezeile in *values* gespeichert:

```
values = adc.read_adc(0, gain=GAIN)
```

Verbindung des Moduls mit dem Raspberry Pi

Verbinden Sie das GY-61 Modul mit dem Raspberry Pi, wie unten abgebildet:



Az-Delivery

| Sensor Pin | > | Raspberry Pi Pin | |
|------------|---|------------------|------------------------|
| VCC | > | 3V3 [pin 17] | Roter Draht |
| GND | > | GND [pin 20] | Schwarzer Draht |

| Sensor Pin | > | ADS1115 Pin | |
|------------|---|-------------|----------------------|
| X_OUT | > | A2 | Oranger Draht |
| Y_OUT | > | A1 | Oranger Draht |
| Z_OUT | > | A0 | Oranger Draht |

| ADS1115 Pin | > | Raspberry Pi Pin | |
|-------------|---|------------------|------------------------|
| VDD | > | 3V3 [pin 1] | Roter Draht |
| GND | > | GND [pin 9] | Schwarzer Draht |
| SDA | > | GPIO2 [pin 3] | Grüner Draht |
| SCL | > | GPIO3 [pin 5] | Blauer Draht |

Az-Delivery

Python-Skript

```
import time
from ADS1x15 import ADS1115

adc = ADS1115()
GAIN = 1

values = list()

print('Press CTRL + C to end the script!')
try:
    while True:
        for i in range(3):
            values.append(adc.read_adc(i, gain=GAIN))

            print('X axis:{:>6}\tY axis:{:>6}\tZ axis:{:>6}'.
                  format(values[2], values[1], values[0]))
            time.sleep(5)

        values = list()

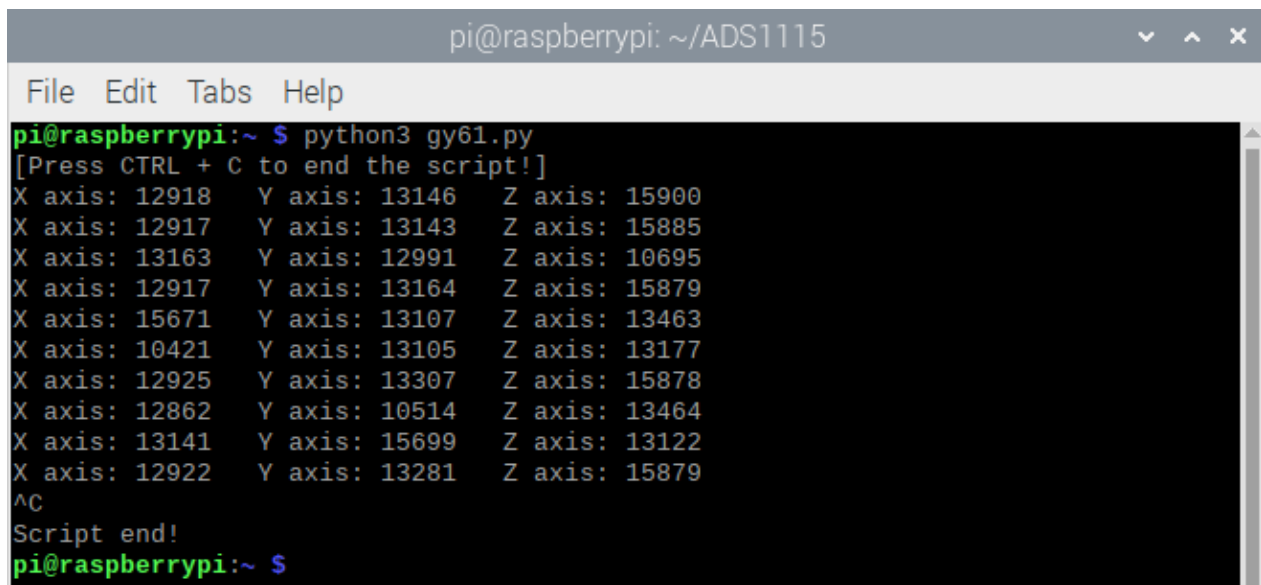
except KeyboardInterrupt:
    print('\nScript end!')
```

Az-Delivery

Speichern Sie das Skript unter dem Namen `gy61.py`. Um das Skript auszuführen, öffnen Sie das Terminal in dem Verzeichnis, in dem das Skript gespeichert ist, und führen Sie den folgenden Befehl aus:

`python3 gy61.py`

Die Ausgabe sollte wie folgt aussehen:



```
pi@raspberrypi: ~/ADS1115
File Edit Tabs Help
pi@raspberrypi:~ $ python3 gy61.py
[Press CTRL + C to end the script!]
X axis: 12918   Y axis: 13146   Z axis: 15900
X axis: 12917   Y axis: 13143   Z axis: 15885
X axis: 13163   Y axis: 12991   Z axis: 10695
X axis: 12917   Y axis: 13164   Z axis: 15879
X axis: 15671   Y axis: 13107   Z axis: 13463
X axis: 10421   Y axis: 13105   Z axis: 13177
X axis: 12925   Y axis: 13307   Z axis: 15878
X axis: 12862   Y axis: 10514   Z axis: 13464
X axis: 13141   Y axis: 15699   Z axis: 13122
X axis: 12922   Y axis: 13281   Z axis: 15879
^C
Script end!
pi@raspberrypi:~ $
```

Um das Skript zu beenden drücken Sie STRG + C auf der Tastatur.

**Sie haben es geschafft. Sie können jetzt unser Modul
für Ihre Projekte nutzen.**

Az-Delivery

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 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>