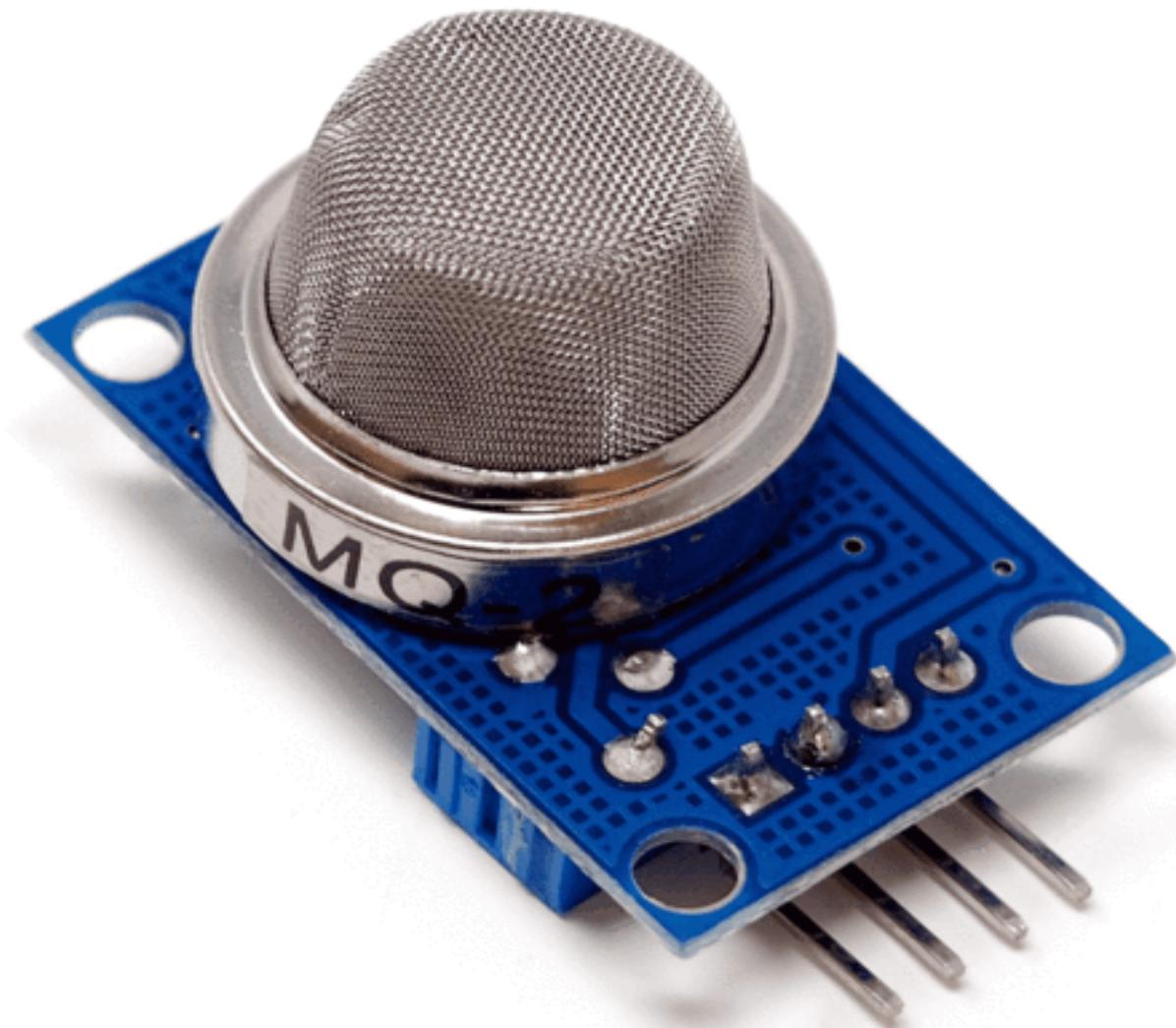


# AZ-Delivery

## Willkommen!

Vielen Dank, dass sie sich für unser MQ-2 Gas Sensor Modul von AZ-Delivery entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

**Viel Spaß!**



# Az-Delivery

Der "MQ-2" ist ein elektronischer Sensor, der die Konzentration von Gasen in der Luft erkennt. Die Gase, die er erfassen kann, sind: LPG, Propan, Methan, Wasserstoff, Alkohol, Rauch und Kohlenmonoxid. Obwohl er diese Gase erkennen kann, ist er nicht in der Lage, zwischen ihnen zu unterscheiden. Der "MQ-2" ist auch als Chemiresistor bekannt. Der "MQ-2" benutzt ein Sensormaterial, dessen Widerstand sich bei unterschiedlichen Gaskonzentrationen ändert. Mit dieser Widerstandsänderung werden die Gase unterschieden. Der Sensor hat auch ein eingebautes Potentiometer, mit dem wir seine Empfindlichkeit einstellen können.

## Technische Daten:

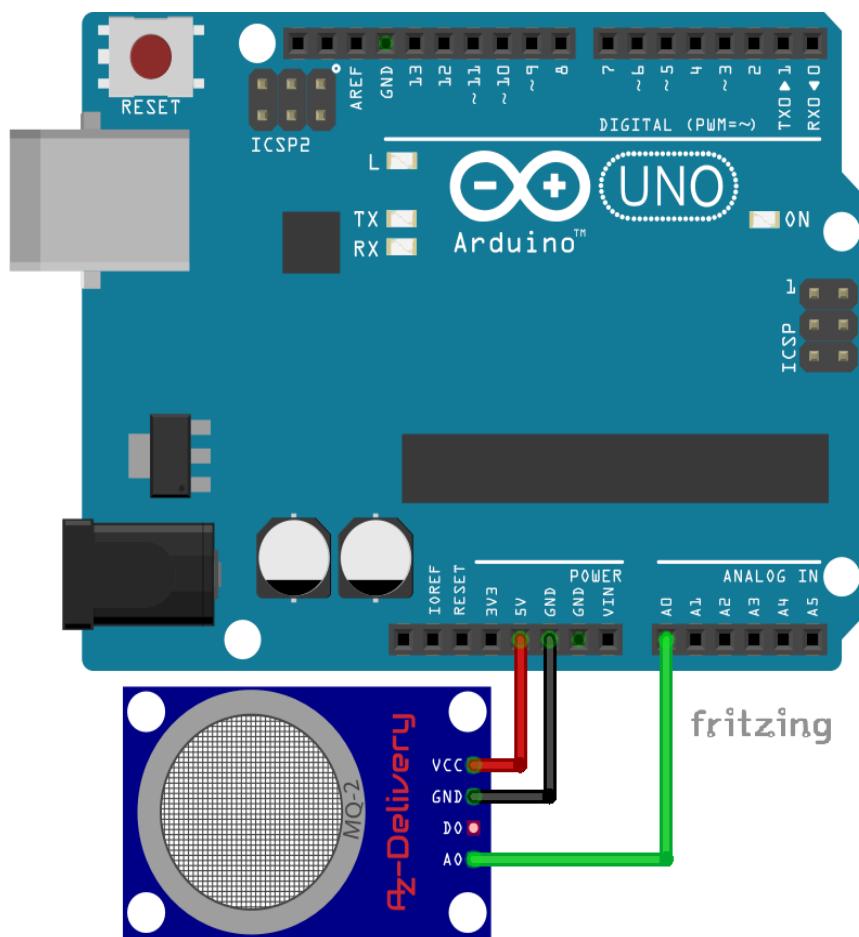
- » Betriebsspannung: 5V
- » Lastwiderstand: 20kΩ
- » Heizgerätwiderstand: 33Ω+5%
- » Sensorwiderstand: 10kΩ - 60kΩ
- » Hitzeverbrauch: <800mW
- » Vorwärmzeit: >24h
- » Konzentrationsumfang: 200 – 10000ppm (Teile pro Millionen)

Der Sensor ist von zwei Schichten eines feinen Edelstahlgewebes umschlossen, welches als "Anti-Explosionsnetzwerk" bezeichnet wird. Dadurch ist er in der Lage, brennbare Gase ohne Zwischenfälle zu erkennen. Ebenso schützt es den Sensor und filtert Schwebeteilchen heraus. Auf diese Weise können nur Gase in die Messkammer gelangen.

# Az-Delivery

## Verbindung des Sensors mit dem Arduino Uno

Verbinden Sie alles wie unten abgebildet:



### MQ-2 Modul-Pin > Arduino Pin

VCC	>	5V	<b>Roter Draht</b>
GND	>	GND	<b>Schwarzer Draht</b>
A0	>	A0 Pin	<b>Grüner Draht</b>

# Az-Delivery

## Der "MQ-2"-Modul Arduino Sketch

Eine Library wird nicht benötigt:

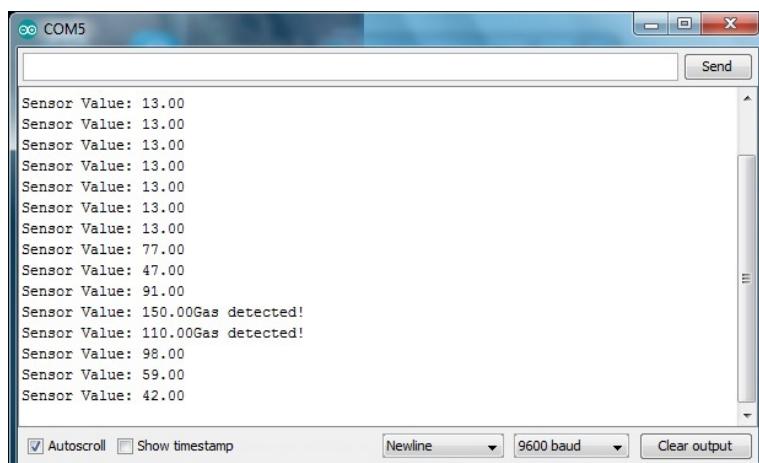
```
float sensorValue;  
void setup() {  
    Serial.begin(9600);  
    Serial.println("The gas sensor is warming up...");  
    delay(30000);  
}  
void loop() {  
    sensorValue = analogRead(0);  
    Serial.print("Sensor Value: ");  
    Serial.print(sensorValue);  
    if(sensorValue > 100) {  
        Serial.print("Gas detected!");  
    }  
    Serial.println("");  
    delay(2000); // wait 2s for next reading  
}
```

# Az-Delivery

Die Skizze beginnt mit der Definition der Variable `sensorValue`, die den Sensorwert speichert.

In der `setup` function initialisieren wir die serielle Kommunikation mit dem PC und warten 30 Sekunden, um den Sensor warm werden zu lassen, denn der Gassensor muss wasserfrei sein. Wir werden den multimolekularen H<sub>2</sub>O-Film (von der Luftfeuchtigkeit) los, indem wir ihn aufheizen.

Als nächstes kommt die `loop` function, die den Sensorwert mit der Funktion `analogRead()` liest und dann die Daten auf dem seriellen Monitor anzeigt. Das "if"-Statement wird zur Überwachung der Ausgabe des Sensors verwendet. Wenn der Ausgang den Wert 100 überschreitet, wird die Meldung "Gas erkannt!" auf dem Serial Monitor angezeigt, wie unten abgebildet:



Sie können sehen, wie der Sensorwert ansteigt, wenn dem Sensor Butan von einem Feuerzeug ausgesetzt wird. Nachdem das Feuerzeug entfernt wurde, beginnt der Wert zu sinken.



## MQ-2 Sensor und Raspberry Pi

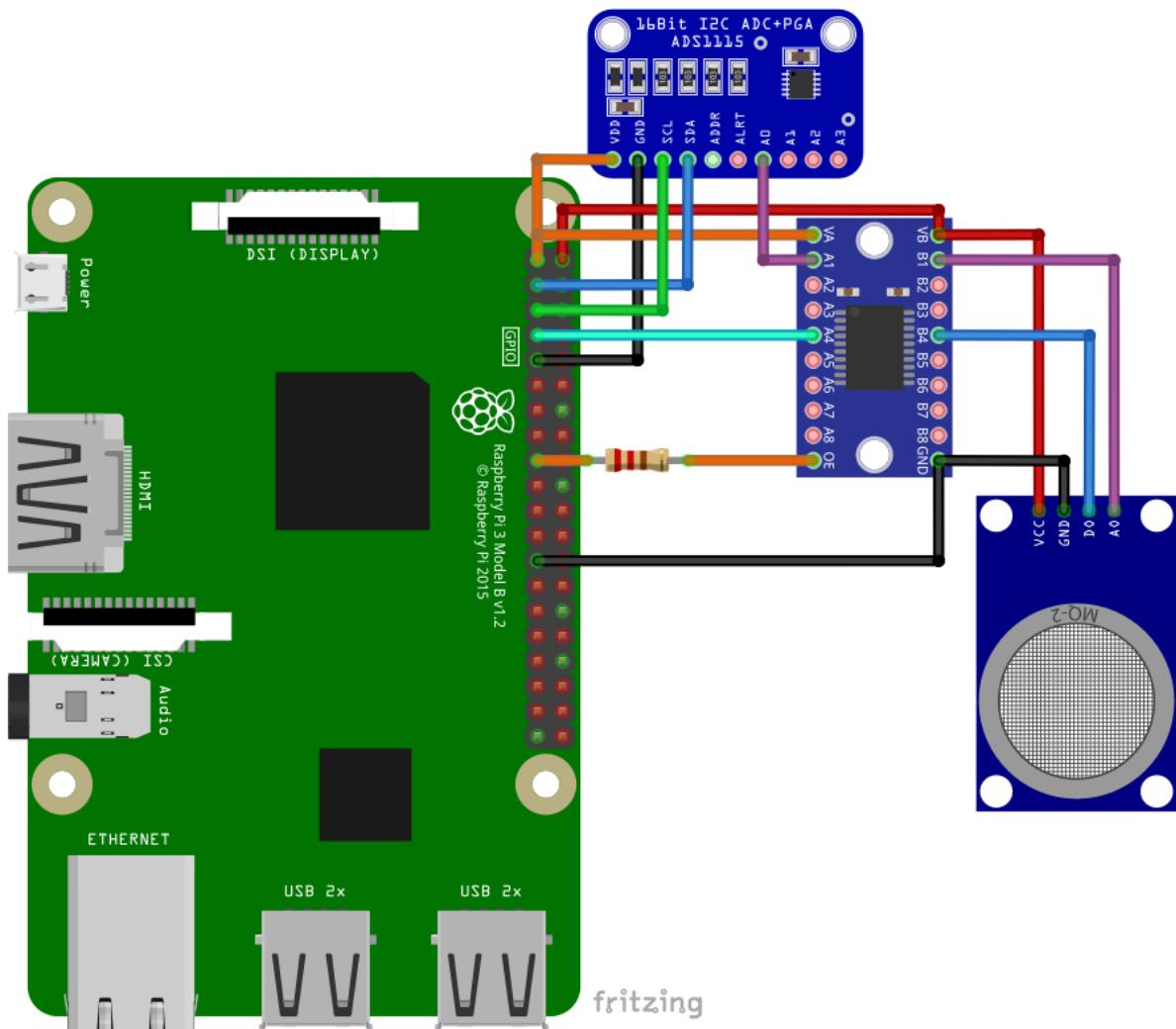
Der Raspberry Pi kann keine analogen Eingänge lesen, und deshalb müssen wir einen *Analog-Digital-Wandler* (ADC) verwenden. Ein ADC ist ein Gerät, das, wie der Name schon sagt, analoge Spannungen in digitale Werte umwandelt. Wir werden ein Gerät namens "ADS1115" verwenden, der den I2C-Bus verwendet, um Daten an einen Mikrocontroller zu senden. Zuerst müssen wir also "I2C" auf dem Raspberry Pi aktivieren.

Ebenso können die GPIOs des Raspberry Pi's die 5V-Spannungen nicht verarbeiten, so dass wir einen Logikpegel-Wandler verwenden müssen. Ein Logikpegelwandler ist ein Gerät, das 5V-Signale in 3,3V-Signale umwandelt und umgekehrt. Die meisten Logikpegelwandler sind bidirektional, d.h. sie können auch 3,3V-Signale wieder in 5V umwandeln. Wir werden ein Gerät namens "Txs0108E" verwenden, welcher einer dieser bidirektionalen Wandler ist.



## Verbindung des "MQ-2" mit dem Raspberry Pi

Verbinden Sie alles, wie unten abgebildet:



# Az-Delivery

## Raspberry Pin > ADS1115 Pin

3,3V [Pin 1]	>	VDD	Oranger Draht
GPIO2 [Pin 3]	>	SDA	Blauer Draht
GPIO3 [Pin 5]	>	SCL	Grüner Draht
GND [Pin 9]	>	GND	Schwarzer Draht

## Raspberry Pin > TXS0108E Pin

3,3V [Pin 1]	>	VA	Oranger Draht
5V [Pin 2]	>	VB	Roter Draht
GPIO4 [Pin 7]	>	A4	Cyaner Draht
3,3V [Pin 17]	>	OE	Oranger Draht
GND [Pin 25]	>	GND	Schwarzer Draht

Verbinden Sie einen Pull-Up-Widerstand mit 10kΩ zwischen 3,3V (Pin 17) und OE.

## ADS1115 Pin > TXS0108E Pin

A0	>	A1	Lila Draht
----	---	----	------------

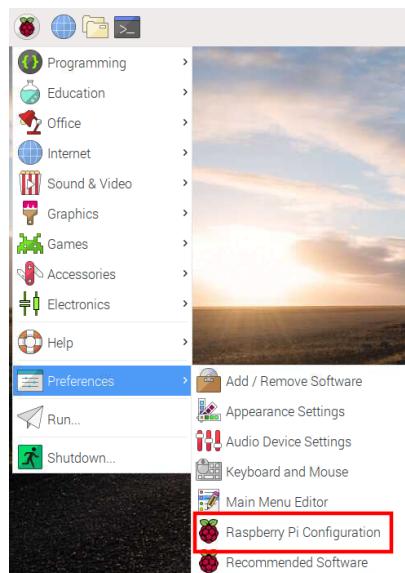
## TXS0108E Pin > MQ-2 Pin

VB	>	VCC	Roter Draht
B1	>	A0	Lila Draht
B4	>	D0	Blauer Draht
GND	>	GND	Schwarzer Draht

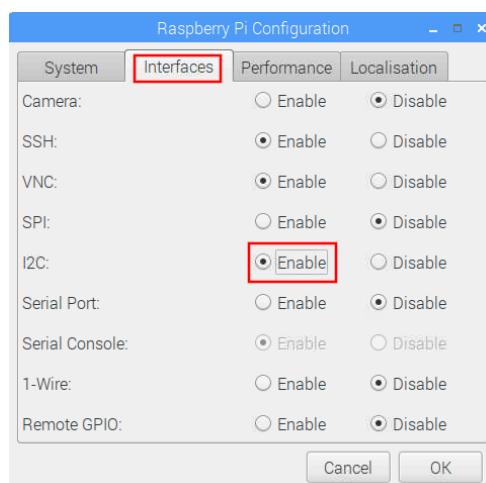
## Aktivieren der I2C-Schnittstelle

Um die I2C-Schnittstelle auf Raspberry Pi in Ihrem Raspbian zu aktivieren, gehen Sie zu:

*Start > Preferences > Raspberry Pi Configuration.*



Dann wird ein neues Fenster geöffnet. Gehen Sie zu dem zweitem Tab "Interfaces" und aktivieren Sie das Optionsfeld I2C. Wenn Sie dies getan haben, drücken Sie auf die Schaltfläche "ok", wie unten abgebildet:





Dadurch wird das I2C-Interface auf den "GPIO2"- and "GPIO3"- Pins aktiviert. Als nächstes müssen wir die library für das ADC Gerät aktivieren. Die library heißt "*Adafruit\_Python\_ADS1x15*". Dafür, öffnen Sie die Terminal App in 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
```

Jetzt können wir mit dem Python-Code beginnen.

# Az-Delivery

## Python-Code

Erstellen Sie eine neue Datei mit Namen "mq2.py", und fügen Sie folgenden Code ein:

```
import time
import Adafruit_ADS1x15
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
digitalPin = 4
GPIO.setup(digitalPin, GPIO.IN)
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1
print("[press ctrl+c to end the script]")
try: # Main program loop
    while True:
        analogReading = adc.read_adc(0, gain=GAIN)
        digitalReading = GPIO.input(digitalPin)
        print("Analog read: {:>6}\t- Digital read: {}".format(analogReading, digitalReading))
        time.sleep(0.5)

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

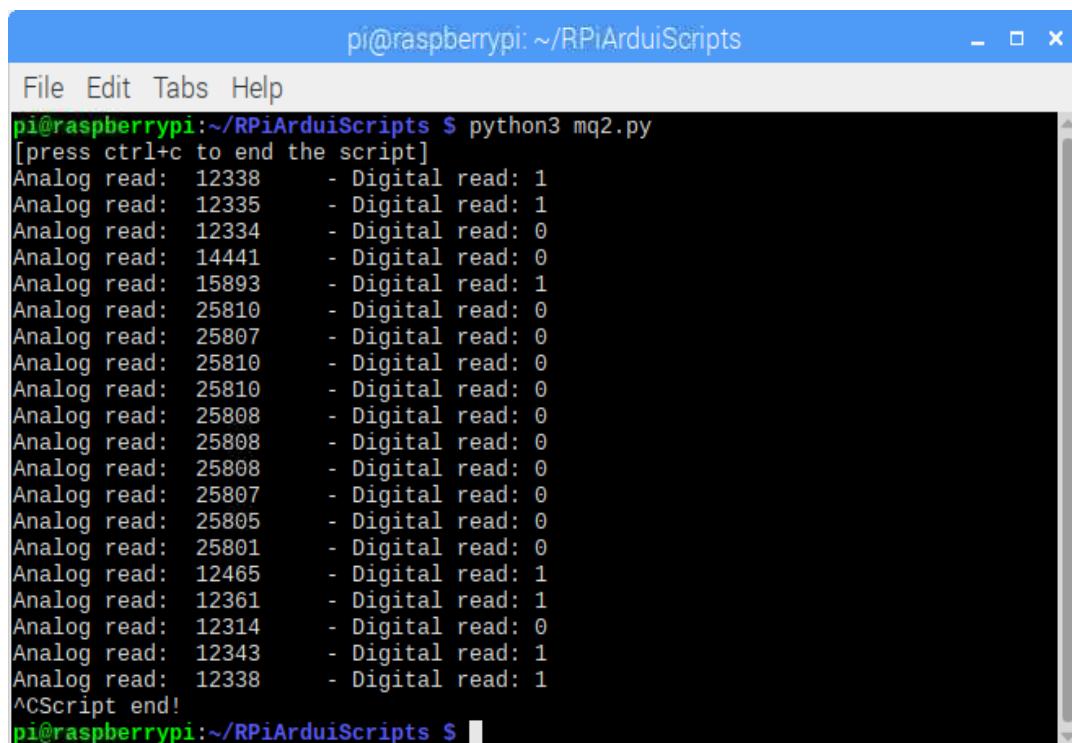
# Az-Delivery

Zu Beginn des Skripts importieren wir die entsprechenden libraries. Als nächstes richten wir unsere Variablen ein und schalten die Warnungen aus. Die loop wird so eingerichtet, dass sie sowohl analoge als auch digitale Messungen durchführt und diese dann ausgibt. Die analogen Werte werden als sechsstellige Zahlen angezeigt. Beachten Sie, dass in dem Fall, dass eine Null an der ersten Stelle steht, keine Nullen angezeigt werden. Und schließlich konfigurieren wir die Tastaturunterbrechung so, dass wir das Skript an jedem Punkt durch Drücken von "STRG + C" anhalten können.

Nachdem alle vorherigen Schritte ausgeführt wurden, führen Sie das Skript mit dem folgenden Befehl aus:

**python3 mq2.py**

Die Ausgabe sollte ähnlich, wie unten abgebildet, aussehen:



```
pi@raspberrypi:~/RPiArduScripts$ python3 mq2.py
[press ctrl+c to end the script]
Analog read: 12338 - Digital read: 1
Analog read: 12335 - Digital read: 1
Analog read: 12334 - Digital read: 0
Analog read: 14441 - Digital read: 0
Analog read: 15893 - Digital read: 1
Analog read: 25810 - Digital read: 0
Analog read: 25807 - Digital read: 0
Analog read: 25810 - Digital read: 0
Analog read: 25810 - Digital read: 0
Analog read: 25808 - Digital read: 0
Analog read: 25808 - Digital read: 0
Analog read: 25808 - Digital read: 0
Analog read: 25807 - Digital read: 0
Analog read: 25805 - Digital read: 0
Analog read: 25801 - Digital read: 0
Analog read: 12465 - Digital read: 1
Analog read: 12361 - Digital read: 1
Analog read: 12314 - Digital read: 0
Analog read: 12343 - Digital read: 1
Analog read: 12338 - Digital read: 1
^CScript end!
pi@raspberrypi:~/RPiArduScripts$
```

# AZ-Delivery

Wie Sie auf dem Bild sehen können, lesen wir den Sensorstatus jede halbe Sekunde ab. Der Sensor erkennt natürliche Gase, die in der Luft gefunden werden, so dass die analogen Standardwerte bei etwa 12000 liegen. Wenn der Sensor jedoch Butangas aus einem Feuerzeug ausgesetzt wird, gehen die digitalen Messwerte meist auf Null zurück, und die Analogwerte steigen von 12000 auf 25000 an. Wenn die digitalen Werte meist Null ergeben, bedeutet dies, dass eine hohe Gaskonzentration vorliegt, während die analogen Werte den Wert der Gaskonzentration anzeigen.

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



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 noch 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>