Willkommen!

Und herzlichen Dank für den Kauf unseres MQ-135 Gas-Sensors! Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Schritte von der Einrichtung bis zur Ausgabe der Werte.

Viel Spaß!



Inhaltsverzeichnis

Einführung	**3**
Spezifikationen	**4**
Pinbelegung	**5**
Einrichten der Arduino-IDE einrichtet	**6**
Wie man den Raspberry Pi und Python einrichtet	**11**
Verbindung des Moduls mit dem Mikrocontroller	**12**
Raspberry Pi und der ADC	**17**
Verbindung des Moduls mit dem Raspberry Pi	**18**
Aktivieren der I2C-Schnittstelle	**19**
Python Skript	**21**

Einführung

Das Gassensormodul MQ-135 ist ein Gerät, das zum Erfassen und Messen der Konzentration von Gasen in der Luft verwendet wird. Es kann folgende Gase erkennen: Ammoniak, Sulfid, Flüssiggas, Propan, Methan, Wasserstoff, Alkohol, Rauch und Kohlenmonoxid sowie andere schädliche Gase. Obwohl es diese Gase erkennen kann, ist es nicht in der Lage, zwischen ihnen zu unterscheiden.

Es wird meist als häuslicher, industrieller und tragbarer Luftverschmutzungsdetektor verwendet.

Der MQ-135 ist ein Metall-Oxid-Halbleiter (MOS), der auch als Chemiresistor bezeichnet wird. Der Sensor enthält ein Sensormaterial, dessen Widerstand sich bei unterschiedlichen Gaskonzentrationen ändert. Diese Änderung des Widerstands wird zur Gaserkennung genutzt. Der Sensor hat auch ein eingebautes Potentiometer, mit dem wir seine Empfindlichkeit einstellen können.

Der Sensor ist von zwei Lagen eines feinen Edelstahlgewebes, dem sogenannten Antiexplosionsnetzwerk, umgeben. Dadurch ist er in der Lage, brennbare Gase ohne Zwischenfälle zu erkennen. Außerdem schützt es den Sensor und filtert Schwebeteilchen heraus. Auf diese Weise können nur Gase in die Messkammer gelangen.

Das Modul hat einen integrierten LM393-Komparatorchip, der die Messwerte in digitale und analoge Signale umwandelt. Außerdem gibt es ein Potentiometer, das zur Kalibrierung der Erkennungsempfindlichkeit dient.

Spezifikationen

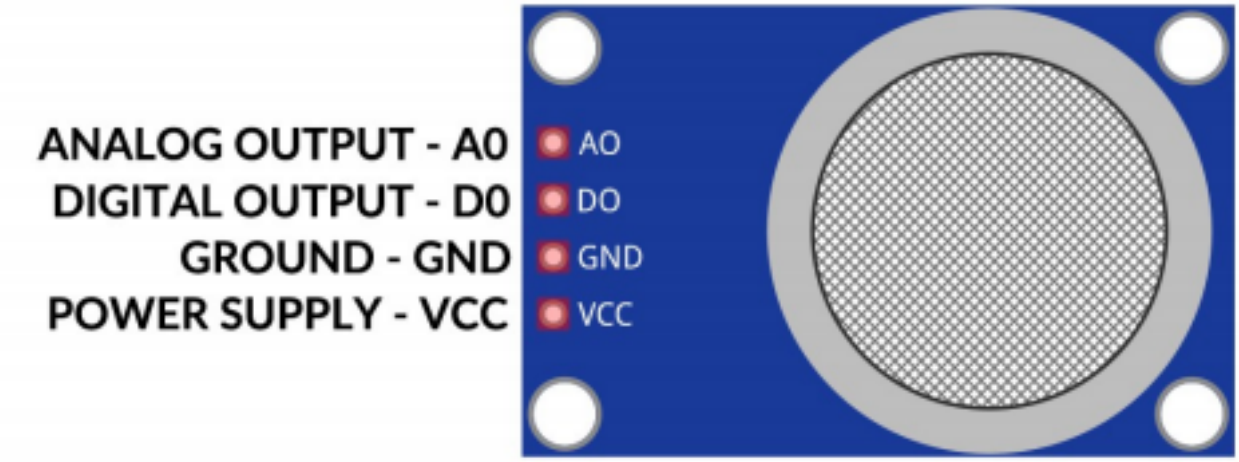
Betriebsspannung:	5V
Betriebsstrom:	150mA
Leistungsaufnahme:	900mW
Lastwiderstand:	20kΩ
Heizungswiderstand:	31Ω+5%
Abtastwiderstand	2kΩ - 20kΩ
Vorwärmzeit:	24h
Konzentrationsbereich:	200 – 10000ppm (parts per million)
Ausgabe:	analog, digital
Abmessungen:	33x21x22mm

Für die besten Erkennungsergebnisse muss der Gassensor vorgeheizt werden. Die beste Vorwärmzeit für den Sensor liegt über 48 Stunden. Detaillierte Informationen zu den technischen Daten des Sensors finden Sie im Datenblatt.

Die Empfindlichkeit des Moduls kann mit einem On-Board-Potentiometer eingestellt werden. Das Bewegen der Potentiometerwelle im Uhrzeigersinn erhöht die Empfindlichkeit. Das Bewegen der Potentiometerwelle in Richtung gegen den Uhrzeigersinn verringert die Empfindlichkeit des Moduls.

Pinbelegung

Das Gassensormodul hat vier Pins. Die Pinbelegung ist in der folgenden Abbildung dargestellt:



Hinweis: Der Raspberry Pi verfügt nicht über einen Digital-Analog-Wandler und kann nicht zum Auslesen analoger Spannungen verwendet werden.

Einrichten der 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.



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

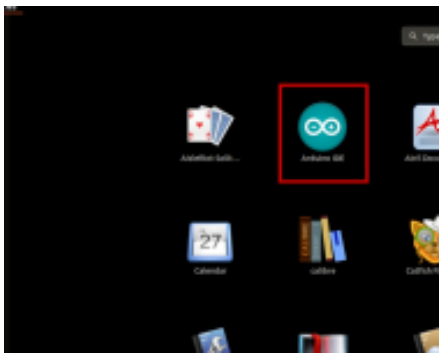
Für *Linux* Benutzer, laden Sie eine Datei mit der Erweiterung *.tar.xz* herunter, die extrahiert werden muss. Wenn sie extrahiert ist, gehen Sie in das extrahierte Verzeichnis und öffnen Sie das Terminal in diesem Verzeichnis. Zwei *.sh* Skripte müssen ausgeführt werden, das erste namens *arduino-linux-setup.sh*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

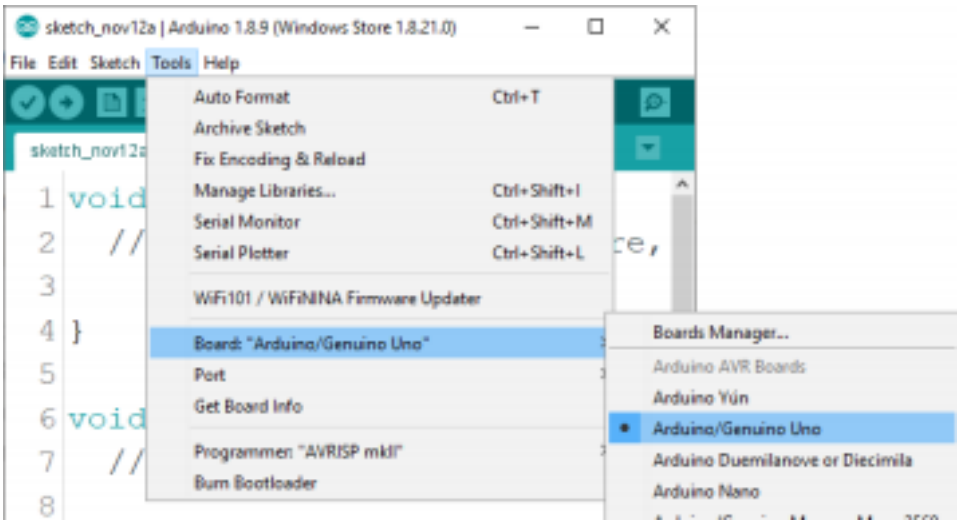
Nach der Installation dieser Skripte gehen Sie zu *All Apps*, wo die *Arduino-IDE* installiert ist.



-

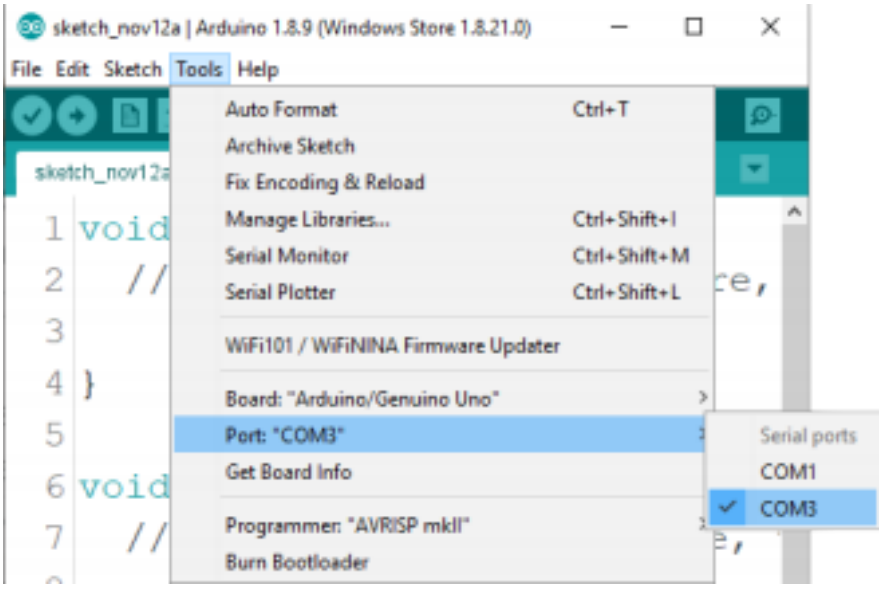
Fast alle Betriebssysteme werden mit einem vorinstallierten Texteditor ausgeliefert (z.B. *Windows* mit *Notepad*, *Linux* Ubuntu mit *Gedit*, *Linux Raspbian* mit *Leafpad* usw.). Alle diese Texteditoren sind für den Zweck des eBooks vollkommen in Ordnung.

Zunächst ist zu prüfen, ob Ihr PC ein Mikrocontroller -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 Mikrocontroller */Genuino Uno* sein, wie es auf dem folgenden Bild zu sehen kann:



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

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.



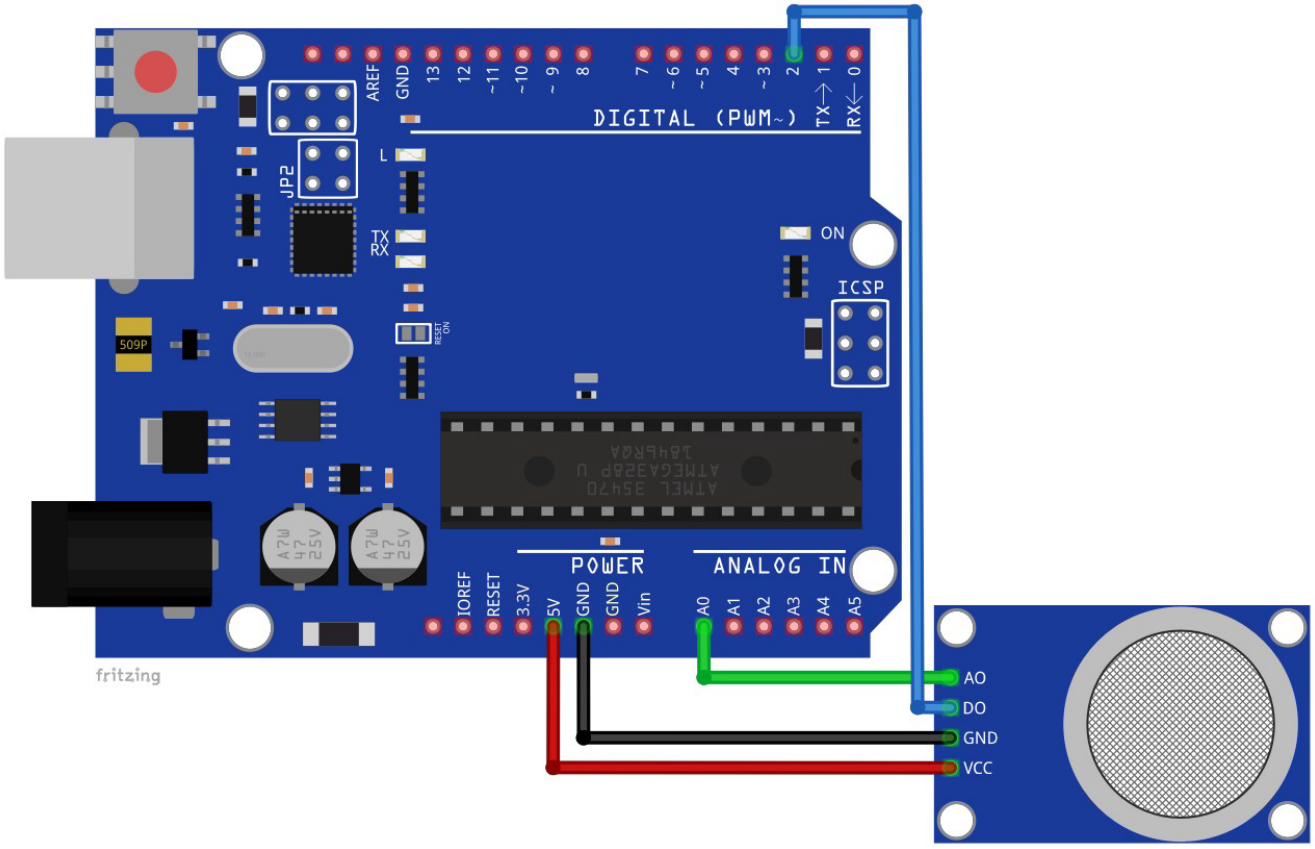
Wie man den Raspberry Pi und Python einrichtet

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 Mikrocontroller

Verbinden Sie das Modul mit dem Mikrocontroller wie im folgenden Bild gezeigt:



Module pin	MC pin	Farbe
VCC	5V	**Rot**
GND	GND	**Schwarz**
D0	D2	**Blau**
A0	A0	**Grün**

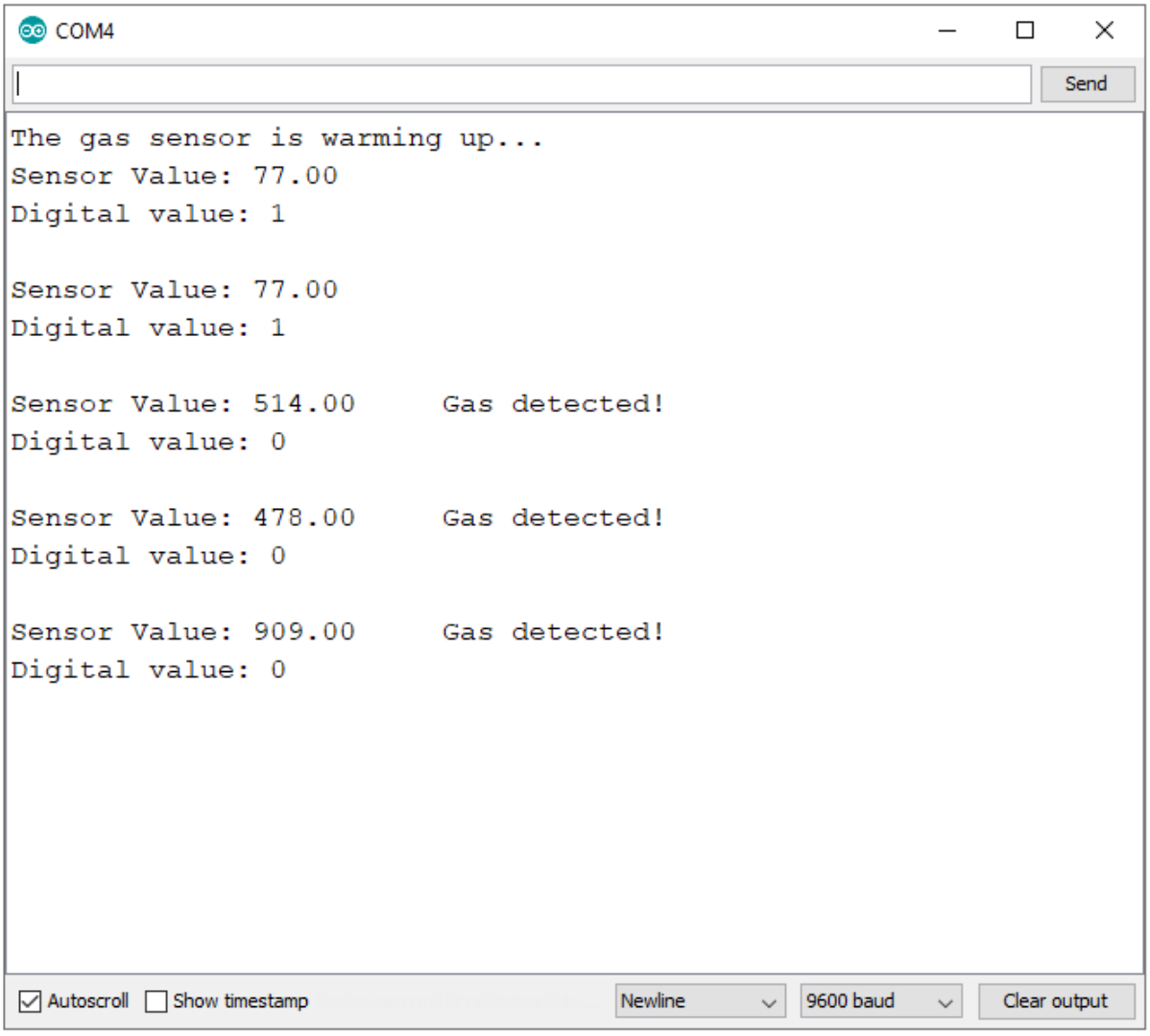
Beispiel Sketch

```
#define DIGITAL_PIN 2
#define ANALOG_PIN 0
uint16_t gasVal;
boolean isgas = false;
String gas;
void setup() {
 Serial.begin(9600);
 Serial.println("The sensor is warming up...");
 delay(30000);
 pinMode(DIGITAL_PIN, INPUT);
}
void loop() {

 gasVal = analogRead(ANALOG_PIN);
 isgas = digitalRead(DIGITAL_PIN);

 if (isgas) {
 gas = "No";
 }
 else {
 gas = "Yes";
 }
 gasVal = map(gasVal, 0, 1023, 0, 100);
 Serial.print("Gas detected: ");
 Serial.println(gas);
 Serial.print("Gas percentage: ");
 Serial.print(gasVal);
 Serial.print("%\n");
 delay(2000);
}
```

Laden Sie den Sketch auf den Mikrocontroller hoch und öffnen Sie den Serial Monitor (Tools > Serial Monitor). Das Ergebnis sollte wie auf dem folgenden Bild aussehen:



-

Der Sketch beginnt mit der Definition und Erstellung zweier Makros namens *DIGITAL_PIN*, *ANALOG_PIN*.

Der *DIGITAL_PIN* stellt den digitalen Pin des Mikrocontrollers dar, der für den Anschluss des digitalen Ausgangspins des Sensors verwendet wird.

Der *ANALOG_PIN* stellt den analogen Eingangspin des Mikrocontrollers dar, der für den Anschluss des analogen Ausgangspins des Sensors verwendet wird.

Die Moduldaten können auf zwei Arten gelesen werden. Die eine ist das Lesen des analogen Ausgangspins des Moduls, die andere das Lesen des digitalen Ausgangspins des Moduls. Um den analogen Ausgangspin des Moduls zu lesen, wird die Variable namens gasVal verwendet, um den Rückgabewert der Funktion analogRead() zu speichern. Der Rückgabewert ist eine Integer-Zahl im Bereich von 0 bis 1023. Um ihn in eine Prozentzahl umzuwandeln, wird die Funktion map() verwendet. Dies ist eine eingebaute Funktion der Arduino IDE. Sie hat fünf Argumente und gibt einen Integer Wert zurück.

Zum Beispiel:
gasVal = map(input, in_min, in_max, out_min, out_max)

Erstes Argument ist der Eingangswert, der im Bereich von in_min bis in_max liegt. Der Rückgabewert ist eine Integer-Zahl im Bereich von out_min bis out_max. Diese Funktion bildet eine Zahl im Eingangsbereich auf eine andere Zahl ab, die in einem anderen Bereich liegt.

Um den digitalen Ausgangspin des Moduls zu lesen, wird die Variable isGas verwendet, um den Rückgabewert der Funktion digitalRead() zu speichern.

Am Ende der Funktion loop() werden die Daten im Serial Monitor angezeigt. Zwischen zwei Messungen liegt eine Pause von 2 Sekunden:
delay(2000);

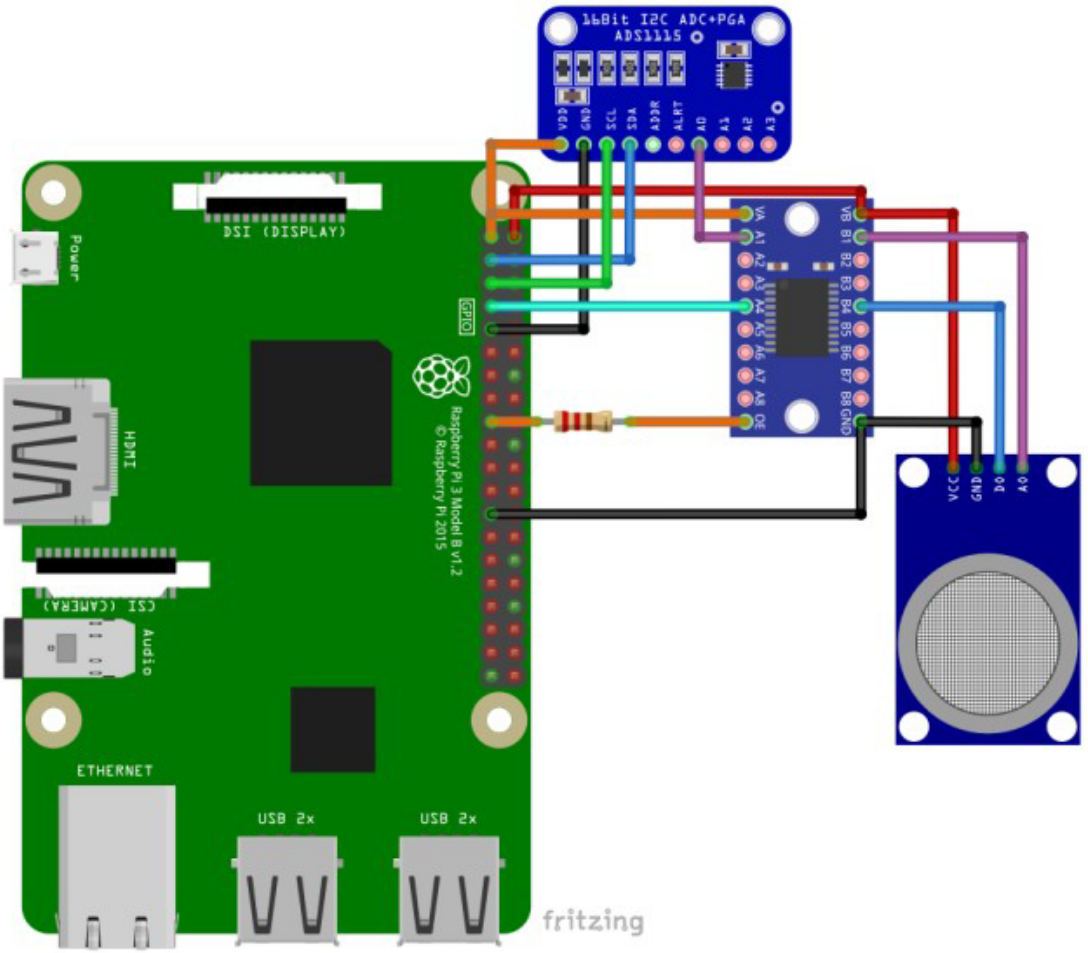
Raspberry Pi und der ADC

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, das den I2C-Bus nutzt, um Daten an den Mikrocontroller zu senden. Zuerst müssen wir also I2C auf dem Raspberry Pi aktivieren.

Ebenso können die GPIOs des Raspberry Pi die 5V-Spannungen nicht verarbeiten, so dass wir einen Logikpegelwandler verwenden müssen. Ein Logik-Pegel-Wandler ist ein Gerät, das ein 5V-Signal in ein 3,3V-Signal umwandelt und umgekehrt. Die meisten Logikpegelwandler sind bidirektional, d. h. sie können auch 3,3-V-Signale zurück in 5 V umwandeln. Wir werden ein Gerät namens "TXS0108E" verwenden, das einer dieser bidirektionalen Wandler ist.

Verbindung des Moduls mit dem Raspberry Pi

Schließen Sie alles an, wie auf dem unten stehenden Anschlussplan dargestellt:



Raspberry pin	**>**	**ADS1115 pin**	
3,3V [pin 1]	>	VDD	**Orange Wire**
GPIO2 [pin 3]	>	SDA	**Blue Wire**
GPIO3 [pin 5]	>	SCL	**Green Wire**
GND [pin 9]	>	GND	**Black Wire**

Raspberry pin	**>**	**TXS0108E pin**	
3,3V [pin 1]	>	VA	**Orange Wire**
5V [pin 2]	>	VB	**Red Wire**
GPIO4 [pin 7]	>	A4	**Cyan Wire**
3,3V [pin 17]	>	OE	**Orange Wire**
GND [pin 25]	>	GND	**Black Wire**

Connect a pull up resistor 10kΩ between 3,3V (pin 17) and OE.

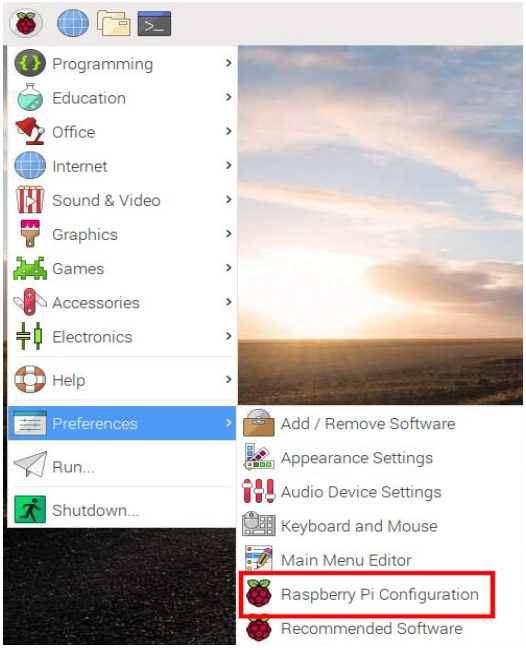
ADS1115 pin	**>**	**TXS0108E pin**	
A0	>	A1	**Purple Wire**

TXS0108E pin	**>**	**MQ-135 pin**	
VB	>	VCC	**Red Wire**
B1	>	A0	**Purple Wire**
B4	>	D0	**Blue Wire**
GND	>	GND	**Black Wire**

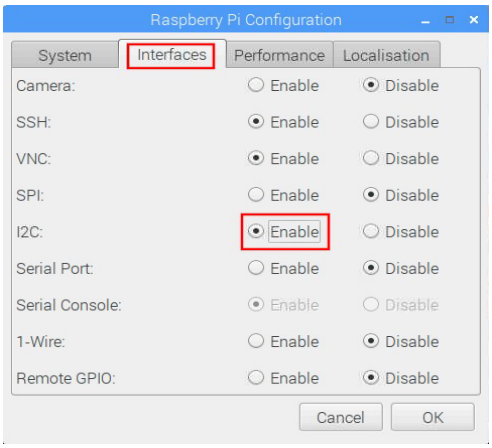
Aktivieren der I2C-Schnittstelle

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-Skript beginnen.

Python Skript

Erstellen Sie eine neue Datei mit dem Namen "mq135.py" und fügen Sie den folgenden Python-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)

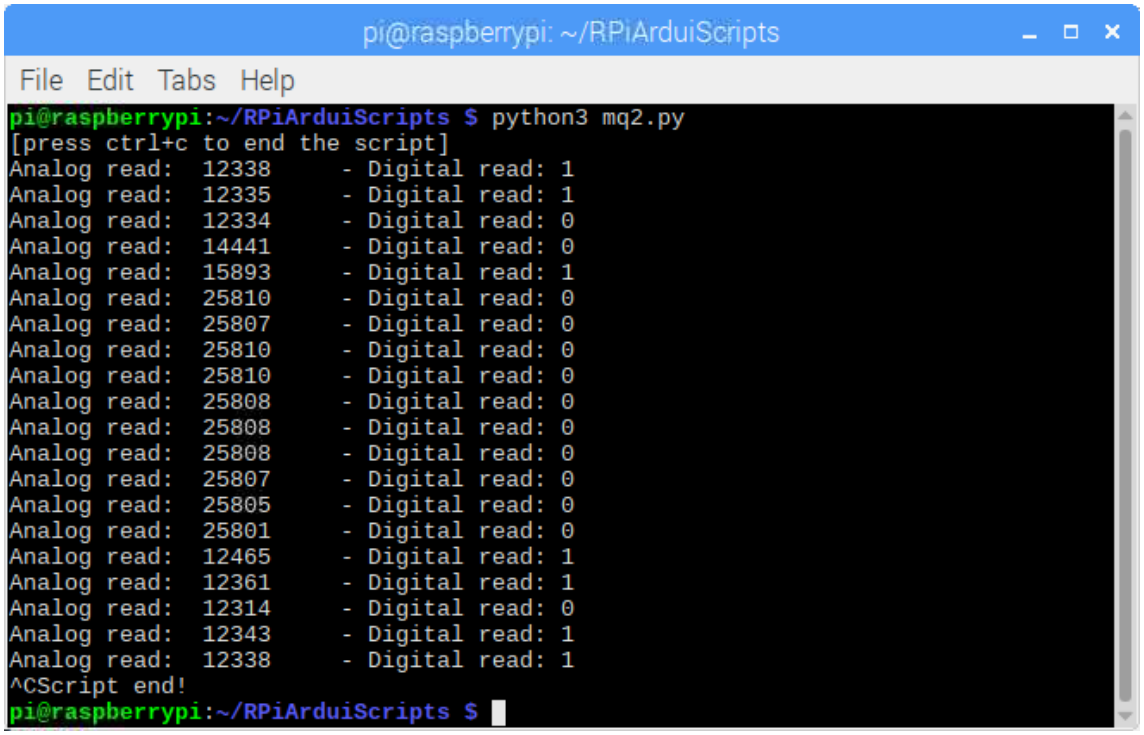
# Scavenging work after the end of the program
except KeyboardInterrupt:
        print("Script end!")
```

Am Anfang des Skripts importieren wir die entsprechenden Bibliotheken. Als nächstes richten wir unsere Variablen ein und schalten die Warnungen aus. Die Schleife ist so eingerichtet, dass sie sowohl analoge als auch digitale Messwerte erfasst und ausgibt. Die analogen Werte werden als sechsstellige Zahlen ausgegeben. Beachten Sie, dass Nullen nicht angezeigt werden, wenn sie die erste Ziffer sind. Und schließlich konfigurieren wir den Tastaturinterrupt so, dass wir das Skript an jedem beliebigen Punkt durch Drücken von "STRG + C" anhalten können.

Nachdem die vorherigen Schritte durchgeführt wurden, führen Sie das Skript mit dem folgenden Befehl aus:

python3 mq135.py

Und die Ausgabe sollte ähnlich wie diese aussehen:



Wie Sie auf dem Bild sehen können, lesen wir den Sensorzustand jede halbe Sekunde aus. Der Sensor erkennt natürliche Gase, die in der Luft vorkommen, daher liegen die Standard-Analogwerte bei etwa 12000. Wenn der Sensor jedoch mit Butangas aus einem Feuerzeug in Berührung kommt, werden die digitalen Messwerte größtenteils zu Nullen und die analogen Werte steigen von 12000 auf 25000 an. Wenn die digitalen Werte größtenteils Nullen sind, bedeutet dies, dass eine Gaskonzentration hoch ist, 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 MIkroelektronk-Produkten und Zubehör 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