

## Herzlich willkommen!

Vielen Dank, dass Sie sich für unser *Gassensormodul AZ-Delivery MQ-135 entschieden haben*. Auf den folgenden Seiten erfahren Sie, wie Sie dieses praktische Gerät verwenden und einrichten.

**Viel Spaß!**



## Anwendungsbereiche

Bildung und Lehre: Einsatz in Schulen, Hochschulen und Ausbildungseinrichtungen zur Vermittlung von Grundlagen der Elektronik, Programmierung und eingebetteten Systemen. Forschung und Entwicklung: Verwendung in Forschungs- und Entwicklungsprojekten zur Erstellung von Prototypen und Experimenten in den Bereichen Elektronik und Informatik. Prototypenentwicklung: Einsatz in der Entwicklung und Erprobung neuer elektronischer Schaltungen und Geräte. Hobby und Maker-Projekte: Verwendung durch Elektronikenthusiasten und Hobbyisten zur Entwicklung und Umsetzung von DIY-Projekten.

## Erforderliche Kenntnisse und Fähigkeiten

Grundlegendes Verständnis der Elektronik und Elektrotechnik. Kenntnisse in der Programmierung, insbesondere in der Programmiersprache C/C++. Fähigkeit, Schaltpläne zu lesen und einfache Schaltungen zu entwerfen. Erfahrung im Umgang mit elektronischen Komponenten und Löten.

## Betriebsbedingungen

Das Produkt darf nur mit den im Datenblatt spezifizierten Spannungen betrieben werden, um Beschädigungen zu vermeiden. Eine stabilisierte Gleichstromquelle ist zum Betrieb erforderlich. Bei der Verbindung mit anderen elektronischen Komponenten und Schaltungen sind die maximalen Strom- und Spannungsgrenzen zu beachten, um Überlastungen und Schäden zu vermeiden.

## Umweltbedingungen

Das Produkt sollte in einer sauberen, trockenen Umgebung verwendet werden, um Schäden durch Feuchtigkeit oder Staub zu vermeiden. Schützen Sie das Produkt vor direkter Sonneneinstrahlung (UV)

## Bestimmungsgemäße Verwendung

Das Produkt ist für den Einsatz in Bildungs-, Forschungs- und Entwicklungsumgebungen konzipiert wurde. Es dient zur Entwicklung, Programmierung und Prototypenentwicklung von elektronischen Projekten und Anwendungen. Das Sensor Produkt ist nicht als fertiges Verbraucherprodukt gedacht, sondern als Werkzeug für technisch versierte Nutzer, darunter Ingenieure, Entwickler, Forscher und Studenten.

## Nicht bestimmungsgemäße vorhersehbare Verwendung

Das Produkt eignet sich nicht für den industriellen Einsatz oder sicherheitsrelevante Anwendungen. Eine Verwendung des Produkts in Medizingeräten oder für Zwecke der Luft- und Raumfahrt ist nicht zulässig

## Entsorgung

Nicht mit dem Hausmüll entsorgen! Ihr Produkt ist entsprechend der europäischen Richtlinie über Elektro- und Elektronik-Altgeräte umweltgerecht zu entsorgen. Die darin enthaltenen, wertvollen Rohstoffe können so der Wiederverwendung zugeführt werden. Die Anwendung dieser Richtlinie trägt zum Umwelt- und Gesundheitsschutz bei. Nutzen Sie die von Ihrer Kommune eingerichtete Sammelstelle zur Rückgabe und Verwertung elektrischer und elektronischer Altgeräte. WEEE-Reg.-Nr.: DE 62624346

## Elektrostatische Entladung

Achtung: Elektrostatische Entladungen können das Produkt beschädigen. Hinweis: Erden Sie sich, bevor Sie das Produkt berühren, indem Sie beispielsweise ein antistatisches Armband tragen oder eine geerdete Metalloberfläche berühren.

## Sicherheitshinweise

Obwohl unser Produkt den Anforderungen der RoHS-Richtlinie (2011/65/EU) entspricht und keine gefährlichen Stoffe in über den Grenzwerten zulässigen Mengen enthält, können dennoch Rückstände vorhanden sein. Beachten Sie die folgenden Sicherheitshinweise, um chemische Gefährdungen zu vermeiden: Achtung: Beim Löten können Dämpfe entstehen, die gesundheitsschädlich sein können. Hinweis: Verwenden Sie einen Lötdampfabsauger oder arbeiten Sie in einem gut belüfteten Bereich. Tragen Sie gegebenenfalls eine Atemschutzmaske. Achtung: Einige Personen könnten empfindlich auf bestimmte Materialien oder Chemikalien reagieren, die im Produkt enthalten sind. Hinweis: Sollten Hautreizungen oder allergische Reaktionen auftreten, unterbrechen Sie die Nutzung und suchen Sie gegebenenfalls einen Arzt auf. Achtung: Halten Sie das Produkt außerhalb der Reichweite von Kindern und Haustieren, um versehentlichen Kontakt und Verschlucken von Kleinteilen zu vermeiden. Hinweis: Bewahren Sie das Produkt in einem sicheren, geschlossenen Behälter auf, wenn es nicht verwendet wird. Achtung: Vermeiden Sie den Kontakt des Produkts mit Nahrungsmitteln und Getränken. Hinweis: Lagern und verwenden Sie das Produkt nicht in der Nähe von Lebensmitteln, um eine Kontamination zu verhindern. Obwohl unser Produkt den

Anforderungen der RoHS-Richtlinie (2011/65/EU) entspricht und keine gefährlichen Stoffe in über den Grenzwerten zulässigen Mengen enthält, können dennoch Rückstände vorhanden sein. Beachten Sie die folgenden Sicherheitshinweise, um chemische Gefährdungen zu vermeiden: Achtung: Beim Löten können Dämpfe entstehen, die gesundheitsschädlich sein können. Hinweis: Verwenden Sie einen Lötdampfabsauger oder arbeiten Sie in einem gut belüfteten Bereich. Tragen Sie gegebenenfalls eine Atemschutzmaske. Achtung: Einige Personen könnten empfindlich auf bestimmte Materialien oder Chemikalien reagieren, die im Produkt enthalten sind. Hinweis: Sollten Hautreizungen oder allergische Reaktionen auftreten, unterbrechen Sie die Nutzung und suchen Sie gegebenenfalls einen Arzt auf. Achtung: Halten Sie das Produkt außerhalb der Reichweite von Kindern und Haustieren, um versehentlichen Kontakt und Verschlucken von Kleinteilen zu vermeiden. Hinweis: Bewahren Sie das Produkt in einem sicheren, geschlossenen Behälter auf, wenn es nicht verwendet wird. Achtung: Vermeiden Sie den Kontakt des Produkts mit Nahrungsmitteln und Getränken. Hinweis: Lagern und verwenden Sie das Produkt nicht in der Nähe von Lebensmitteln, um eine Kontamination zu verhindern. Das Produkt enthält empfindliche elektronische Komponenten und scharfe Kanten. Unsachgemäßer Umgang oder Montage kann zu Verletzungen oder Beschädigungen führen. Beachten Sie die folgenden Sicherheitshinweise, um mechanische Gefährdungen zu vermeiden: Achtung: Die Platine und die Anschlüsse des Produkts können scharfe Kanten aufweisen. Gehen Sie vorsichtig vor, um Schnittverletzungen zu vermeiden. Hinweis: Tragen Sie bei der Handhabung und Montage des Produkts geeignete Schutzhandschuhe. Achtung: Vermeiden Sie übermäßigen Druck oder mechanische Belastung der Platine und der Komponenten. Hinweis: Montieren Sie das Produkt nur auf stabilen und ebenen Oberflächen. Verwenden Sie geeignete Abstandshalter und Gehäuse, um mechanische Belastungen zu minimieren. Achtung: Stellen Sie sicher, dass das Produkt sicher befestigt ist, um unbeabsichtigtes Verrutschen oder Herunterfallen zu verhindern. Hinweis: Verwenden Sie passende Unterlage oder eine sichere Befestigung in Gehäusen oder auf Montageplatten. Achtung: Achten Sie darauf, dass alle Kabelverbindungen sicher und korrekt angeschlossen sind, um Zugbelastungen und versehentliches Herausziehen zu vermeiden. Hinweis: Führen Sie Kabel so, dass sie nicht unter Spannung stehen und keine Stolpergefahr darstellen. Das Produkt arbeitet mit elektrischen Spannungen und Strömen, die bei unsachgemäßem Gebrauch zu elektrischen Schlägen, Kurzschlüssen oder anderen Gefahren führen können. Beachten Sie die folgenden Sicherheitshinweise, um elektrische Gefährdungen zu vermeiden: Achtung: Verwenden Sie das Produkt nur mit den spezifizierten Spannungen. Hinweis: Die Leistungsgrenzen des Produkts finden Sie im dazugehörigen Datenblatt Achtung: Vermeiden Sie Kurzschlüsse zwischen den Anschlüssen und Komponenten des Produkts Hinweis: Achten Sie darauf, dass keine leitenden Objekte die Platine berühren oder überbrücken. Verwenden Sie isolierte Werkzeuge und beachten Sie die Anordnung der Verbindungen. Achtung: Führen Sie keine Arbeiten am Produkt durch, wenn es mit einer Stromquelle verbunden ist. Hinweis: Trennen Sie das Produkt von der Stromversorgung, bevor Sie Änderungen an der Schaltung vornehmen oder Komponenten anschließen bzw. entfernen. Achtung: Überschreiten Sie nicht die spezifizierten Stromstärken für die Ein- und Ausgänge des Produkts. Hinweis: Die Leistungsgrenzen des Produkts finden sich in den technischen Spezifikationen oder im Datenblatt Achtung: Stellen Sie sicher, dass die verwendeten Stromquellen stabil und korrekt dimensioniert sind. Hinweis: Verwenden Sie nur geprüfte und geeignete Netzteile, um Spannungsschwankungen und Überlastungen zu vermeiden. Achtung: Halten Sie ausreichenden Abstand zu spannungsführenden Teilen ein, um unabsichtlichen Kontakt zu vermeiden. Hinweis: Sorgen Sie entsprechend der verwendeten Spannung für eine sichere und übersichtliche Anordnung der Verkabelung. Achtung: Verwenden Sie isolierende Gehäuse oder Schutzabdeckungen, um das Produkt vor direktem Kontakt zu schützen. Hinweis: Setzen Sie das Produkt in ein nicht leitendes Gehäuse ein, um versehentliche Berührungen und Kurzschlüsse zu vermeiden. Das Produkt und die darauf befindlichen Komponenten können sich während des Betriebs erwärmen. Unsachgemäßer Umgang oder eine Überlastung des Produkts kann zu Verbrennungen, Beschädigungen oder Bränden führen. Beachten Sie die folgenden Sicherheitshinweise, um thermische Gefährdungen zu vermeiden: Achtung: Stellen Sie sicher, dass das Produkt innerhalb der empfohlenen Betriebstemperaturen verwendet wird. Hinweis: Der empfohlene Betriebstemperaturbereich liegt typischerweise zwischen -40°C und +85°C. Überprüfen Sie die spezifischen Angaben im Datenblatt des Produkts. Achtung: Platzieren Sie das Produkt nicht in der Nähe von externen Wärmequellen wie Heizkörpern oder direkter Sonneneinstrahlung. Hinweis: Sorgen Sie dafür, dass das Produkt in einem kühlen und gut belüfteten Bereich betrieben wird. Achtung: Stellen Sie sicher, dass das Produkt gut belüftet ist, um eine Überhitzung zu vermeiden. Hinweis: Verwenden Sie Lüfter oder Kühlkörper, wenn das Produkt in einem geschlossenen Gehäuse betrieben wird oder in einer Umgebung mit eingeschränkter Luftzirkulation. Achtung: Montieren Sie das Produkt auf hitzebeständigen Oberflächen und in hitzebeständigen Gehäusen. Hinweis: Verwenden Sie Materialien für Gehäuse, die hohe Temperaturen aushalten können, um eine Beschädigung oder Feuergefahr zu vermeiden. Achtung: Implementieren Sie eine Überwachung der Temperatur bei Verwendung eines Gehäuses und gegebenenfalls Schutzmechanismen, die das Produkt abschalten, wenn es überhitzt. Hinweis: Verwenden Sie Temperaturfühler und entsprechende Software, um die Temperatur des Produkts zu überwachen und das System bei Bedarf abzuschalten. Achtung: Vermeiden Sie Überlastungen, die zu übermäßiger Erwärmung der Komponenten führen können. Hinweis: Überschreiten Sie nicht die spezifizierten Grenzwerte für Strom und Spannung, um eine Überhitzung zu verhindern. Achtung: Kurzschlüsse können erhebliche Hitze entwickeln und Brände verursachen. Hinweis: Stellen Sie sicher, dass alle Verbindungen korrekt und sicher sind und dass keine leitenden Objekte unbeabsichtigt Kurzschlüsse verursachen können.



## **Inhaltsübersicht**

<b>Einführung</b>	<b>3</b>
<b>Spezifikationen</b>	<b>4</b>
<b>Die Pinbelegung</b>	<b>5</b>
<b>Wie man den Raspberry Pi und Python einrichtet</b>	<b>10</b>
<b>Anschließen des Moduls mit ATmega328p</b>	<b>11</b>
<b>Beispiel Skizze</b>	<b>12</b>
<b>Anschluss des Nano V3.0 als ADC für Raspberry Pi</b>	<b>17</b>
<b>Verbinden des Moduls mit dem Raspberry Pi</b>	<b>21</b>
<b>Python-Skript für MQ-135-Modul</b>	<b>28</b>



## Einführung

Das Gassensormodul MQ-135 ist ein Gerät zum Erfassen und Messen der Konzentration von Gasen in der Luft. Es kann folgende Gase erkennen: Flüssiggas, Propan, Methan, Wasserstoff, Alkohol, Rauch und Kohlenmonoxid. Obwohl es diese Gase erkennen kann, ist es nicht in der Lage, den Unterschied zwischen ihnen zu erkennen.

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

Der Sensor ist von zwei Schichten aus feinem Edelmetallgewebe, dem so genannten *Explosionsschutznetz*, umgeben. Dadurch ist er in der Lage, brennbare Gase ohne Zwischenfälle zu erkennen. Außerdem bietet es Schutz für den Sensor und filtert Schwebstoffe heraus. Auf diese Weise können nur Gase in die Sensorkammer gelangen.

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

## Spezifikationen

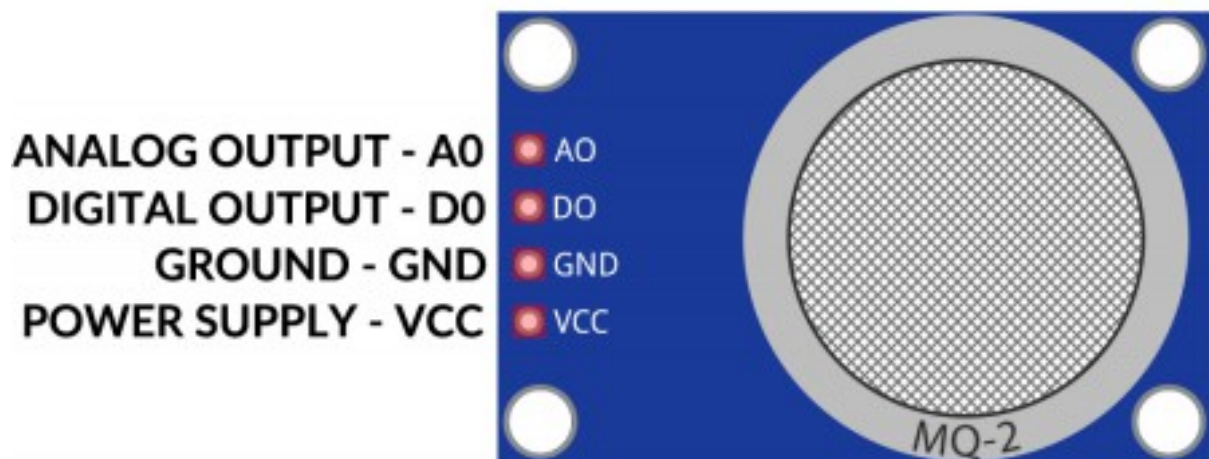
Betriebsspannung:	5V
Betriebsstrom:	150mA
Stromverbrauch:	900mW
Lastwiderstand:	20k $\Omega$
Heizungswiderstand:	33 $\Omega$ +5%
Sensorischer Widerstand	10k $\Omega$ - 60k $\Omega$
Vorwärmzeit:	24h
Konzentrationsbereich:	200 - 10000ppm (Teile pro Million)
Ausgabe:	analog, digital
Abmessungen:	33x21x22mm (1.3x0.8x0.9in)

Um die besten Detektionsergebnisse zu erzielen, muss der Gassensor vorgeheizt werden. Die beste Vorwärmzeit für den Sensor beträgt über 24 Stunden. Ausführliche Informationen zu den technischen Daten des Sensors finden Sie im Datenblatt.

Die Empfindlichkeit des Moduls kann mit einem eingebauten Potentiometer eingestellt werden. Das Bewegen der Potentiometerwelle im Uhrzeigersinn erhöht die Empfindlichkeit. Wenn Sie die Welle des Potentiometers gegen den Uhrzeigersinn drehen, verringert sich die Empfindlichkeit des Moduls.

## Die Pinbelegung

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

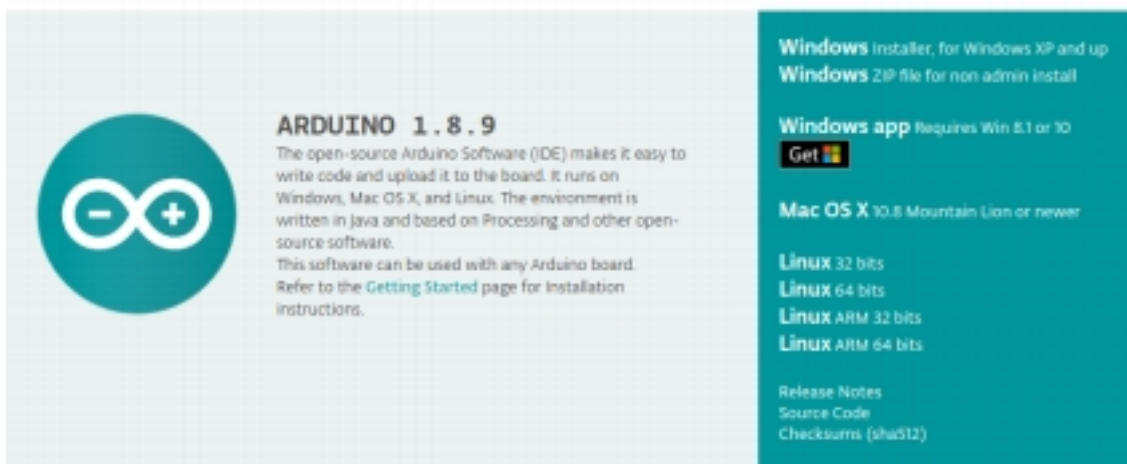


**HINWEIS:** Der Raspberry Pi verfügt nicht über einen Digital-Analog-Wandler und kann nicht zum Lesen analoger Spannungen verwendet werden.

## Einrichten der Arduino IDE

Wenn 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 large teal circle with the Arduino logo (an infinity symbol with a minus and plus sign). To its right, the text reads: **ARDUINO 1.8.9**. Below this, it says: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions." On the right side of the page, there is a teal sidebar with links for different operating systems: **Windows** (Installer, for Windows XP and up; Windows 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)**.

Windows-Benutzer doppelklicken auf die heruntergeladene .exe-Datei und folgen den Anweisungen im Installationsfenster.



# Az-Delivery

Für Linux-Benutzer laden Sie eine Datei mit der Erweiterung *.tar.xz* herunter, die entpackt werden muss. Nach dem Entpacken wechseln Sie in das entpackte Verzeichnis und öffnen das Terminal in diesem Verzeichnis. Zwei *.sh*-Skripte müssen ausgeführt werden, das erste heißt *arduino-linux-setup.sh* und das zweite heißt *install.sh*.

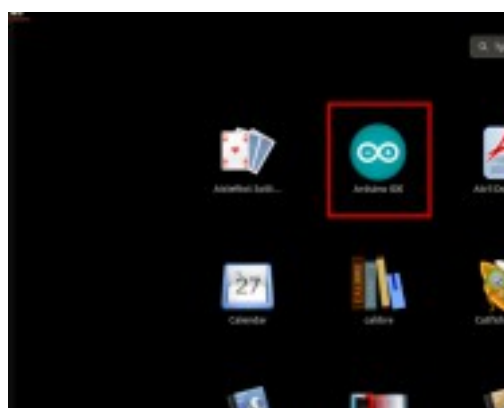
Um das erste Skript im Terminal auszuführen, öffnen Sie das Terminal im extrahierten Verzeichnis und führen Sie den folgenden Befehl aus:

**sh arduino-linux-setup.sh benutzer\_name**

**user\_name** - ist der Name eines *Superusers* im Linux-Betriebssystem. Beim Starten des Befehls muss ein Passwort für den *Superuser* eingegeben werden. Warten Sie ein paar Minuten, bis das Skript alles abgeschlossen hat.

Das zweite Skript, *install.sh*, 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 "*Alle Apps*", wo die *Arduino IDE* installiert ist.

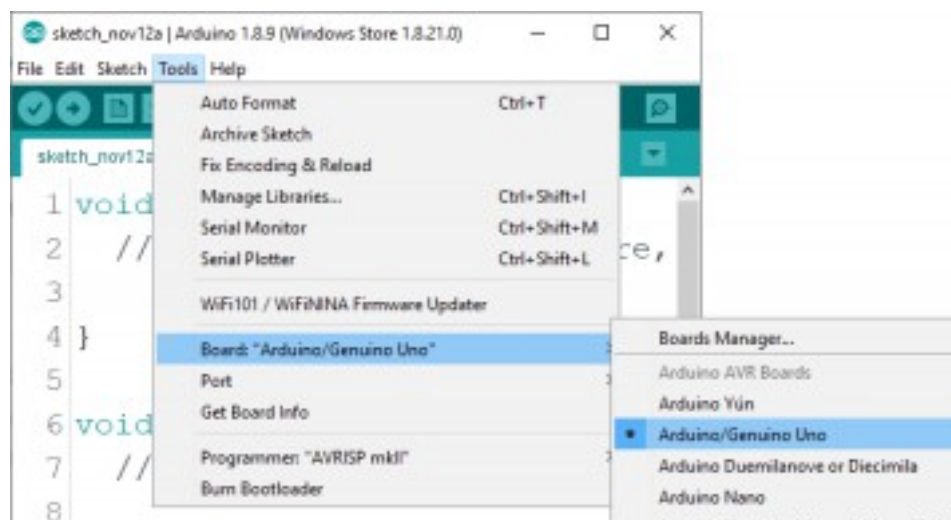


Auf fast allen Betriebssystemen ist ein Texteditor vorinstalliert (z. B. *Windows* mit *Notepad*, *Linux Ubuntu* mit *Gedit*, *Linux Raspbian* mit *Leafpad* usw.). Alle diese Texteditoren sind für den Zweck des Ebooks vollkommen ausreichend.

Als Nächstes müssen Sie prüfen, ob Ihr PC das Mikrocontroller-Board erkennt. Öffnen Sie die frisch installierte Arduino IDE und gehen Sie zu:

*Tools > Board > {Ihr Boardname hier}*

*{Ihr Boardname hier}* sollte der *Mikrocontroller/Genuino Uno* sein, wie er auf dem folgenden Bild zu sehen ist:

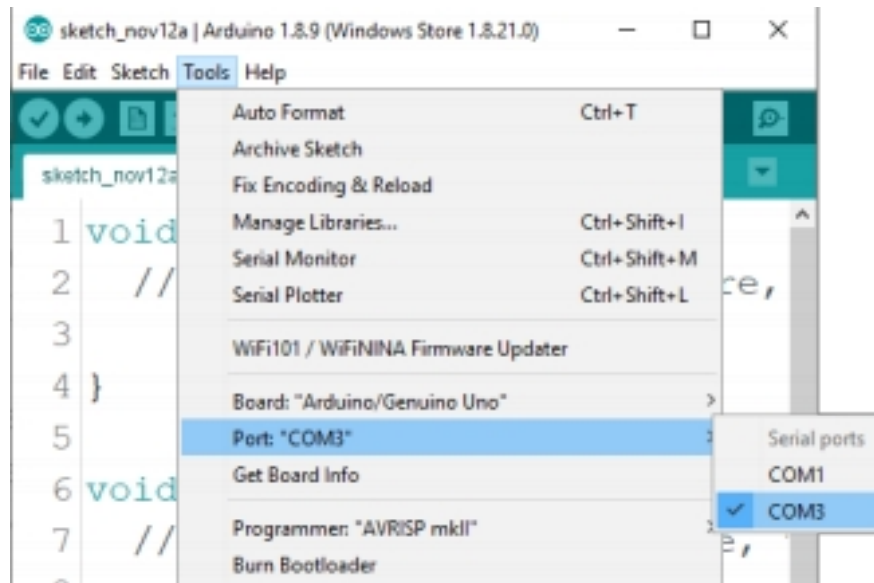


Der Port, an den die Mikrocontrollerplatine angeschlossen ist, muss ausgewählt werden. Gehen Sie zu: *Werkzeuge > Port > {Name des Ports kommt hier hin}*

und wenn die Mikrocontrollerplatine mit dem USB-Anschluss verbunden ist, kann der Name des Anschlusses im Dropdown-Menü auf dem vorherigen Bild angezeigt werden.

Wenn die Arduino IDE unter Windows verwendet wird, lauten

die Portnamen wie folgt:



Für Linux-Benutzer lautet der Name des Anschlusses zum Beispiel `/dev/ttyUSBx`, wobei `x` eine ganze Zahl zwischen `0` und `9` darstellt.



## **So richten Sie den Raspberry Pi und Python ein**

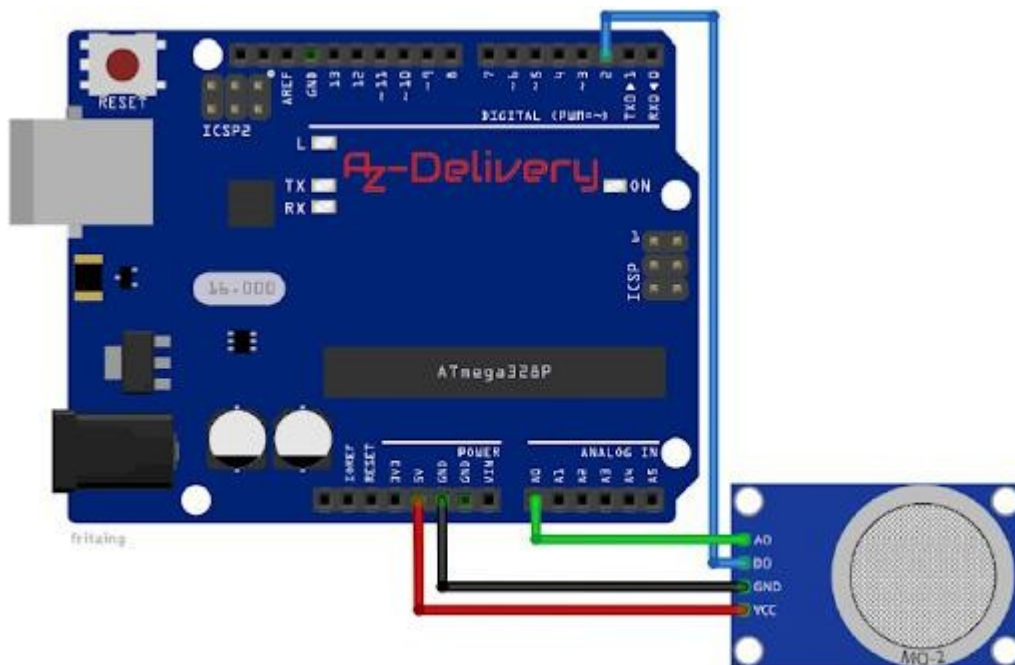
Auf dem Raspberry Pi muss zunächst das Betriebssystem installiert werden, dann muss alles so eingerichtet werden, dass er im Headless-Modus verwendet werden kann. Der Headless-Modus ermöglicht eine Fernverbindung mit dem Raspberry Pi, ohne dass ein PC-Bildschirm, eine Maus oder eine Tastatur benötigt werden. Die einzigen Dinge, die in diesem Modus verwendet werden, sind der Raspberry Pi selbst, die Stromversorgung und die Internetverbindung. All dies wird in dem kostenlosen eBook ausführlich erklärt:

[Raspberry Pi Schnellstart-Anleitung](#)

Auf dem Raspbian-Betriebssystem ist *Python* bereits vorinstalliert.

## Anschließen des Moduls mit ATmega328p

Verbinden Sie das Modul mit dem ATmega328p wie in der folgenden Abbildung gezeigt:



Modul-Stift	ATmega 328p Stift	Farbe des Kabels
VCC	5V	Rotes Kabel
GND	GND	Schwarzes Kabel
D0	D2	Blaues Kabel
A0	A0	Grüner Draht



## Skizze Beispiel

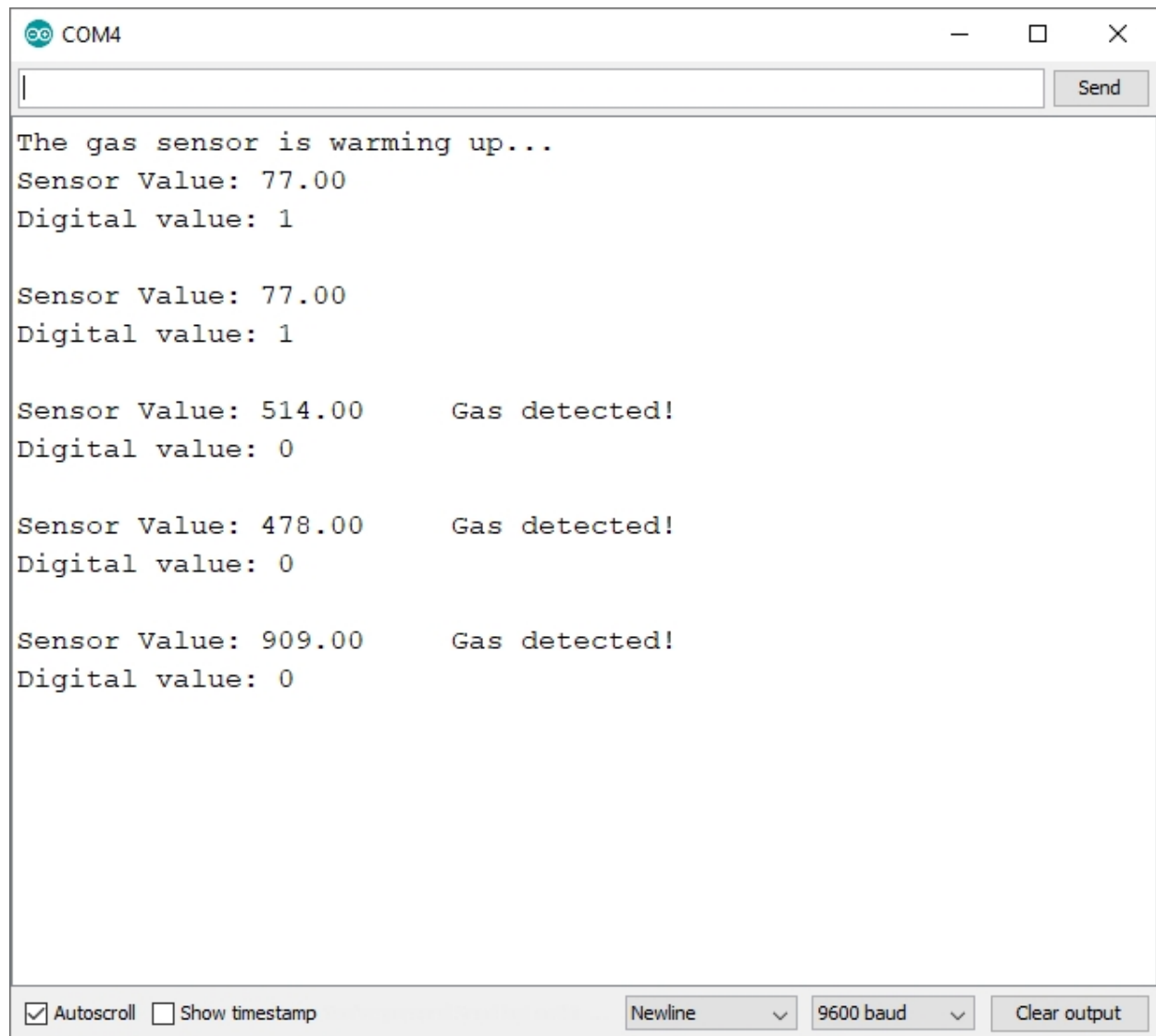
```
#define DIGITAL_PIN 2
#define ANALOG_PIN 0
uint16_t gasVal;
boolean isgas = false;
String gas;
void setup() {
  Serial.begin(9600);
  Serial.println("Der Sensor wärmt sich auf...");
  delay(30000);
  pinMode(DIGITAL_PIN, INPUT);
}
void loop() {

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

  if (isgas) {
    gas = "Nein";
  }
  sonst {
    Gas = "Ja";
  }
  gasVal = map(gasVal, 0, 1023, 0, 100);
  Serial.print("Gas entdeckt: ");
  Serial.println(gas);
  Serial.print("Gasanteil: ");
  Serial.print(gasVal);
  Serial.print("%\n");
  Verzögerung(2000);
}
```

# Az-Delivery

Laden Sie den Sketch auf den ATmega328p und öffnen Sie den Serial Monitor (*Tools > Serial Monitor*). Das Ergebnis sollte wie auf dem folgenden Bild aussehen:



# Az-Delivery

Die Skizze beginnt mit der Definition und Erstellung zweier Makros namens *DIGITAL\_PIN*, *ANALOG\_PIN*.

Der *DIGITAL\_PIN* stellt den digitalen Pin des ATmega328p dar, der für den Anschluss des digitalen Ausgangspins des Sensors verwendet wird.

Der *ANALOG\_PIN* stellt den analogen Eingangspin des ATmega328p dar, der für den Anschluss des analogen Ausgangspins des Sensors verwendet wird.

Die Moduldaten können auf zwei Arten gelesen werden. Zum einen durch Lesen des analogen Ausgangspins des Moduls und zum anderen durch Lesen des digitalen Ausgangspins des Moduls. Um den analogen Ausgangspin des Moduls zu lesen, wird die Variable *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.



# Az-Delivery

Zum Beispiel:

```
gasVal = map(input, in_min, in_max, out_min, out_max)
```

Das erste Argument ist der Eingabewert, der im Bereich von *in\_min* bis *in\_max* liegt. Der Rückgabewert ist eine ganze Zahl im Bereich von *out\_min* bis *out\_max*. Diese Funktion ordnet eine Zahl im Eingabebereich einer anderen Zahl zu, 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 seriellen Monitor angezeigt. Zwischen zwei Messungen werden 2 Sekunden Pause eingelegt:  
`delay(2000);`



## **Anschluss des Nano V3.0 als ADC für Raspberry Pi**

Da der Raspberry Pi keinen Analog-Digital-Wandler (ADC) besitzt, besteht die Aufgabe darin, den Raspberry Pi in die Lage zu versetzen, analoge Spannungen zu lesen. Zu diesem Zweck können ATmega328p oder Nano V3.0 verwendet werden. Dazu muss der Nano V3.0 mit dem Betriebssystem Raspbian verbunden werden. Der Nano V3.0 kann analoge Spannungen auslesen und über die serielle Schnittstelle via USB-Port Daten an den Raspberry Pi senden.

Zunächst muss die Arduino IDE auf Raspbian installiert werden. Zweitens muss die Firmware für das Mikrocontroller-Board auf Nano V3.0 hochgeladen und die Python-Bibliothek heruntergeladen werden.

Starten Sie dazu Raspbian, öffnen Sie das Terminal und führen Sie den folgenden Befehl aus, um Raspbian zu aktualisieren:

**`sudo apt-get update && sudo apt-get upgrade -y`**

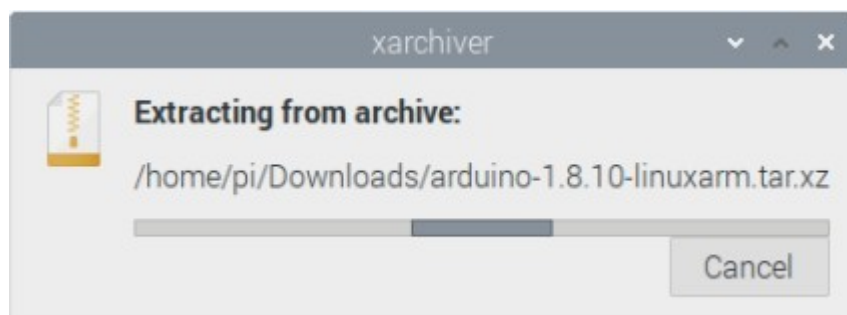
# Az-Delivery

Um die Arduino IDE herunterzuladen und zu installieren, gehen Sie auf die [Arduino-Website](#): und laden Sie die *tar.xz*-Datei der Arduino IDE für **Linux ARM 32 Bits** herunter, wie im folgenden Bild gezeigt:

## Download the Arduino IDE



Entpacken Sie dann die *tar.xz*-Datei. Öffnen Sie den Dateifexplorer im Verzeichnis, in das die *tar.xz*-Datei heruntergeladen wurde, klicken Sie mit der rechten Maustaste darauf und wählen Sie die Option Hier extrahieren. Warten Sie ein paar Minuten, bis der Entpackungsvorgang abgeschlossen ist.

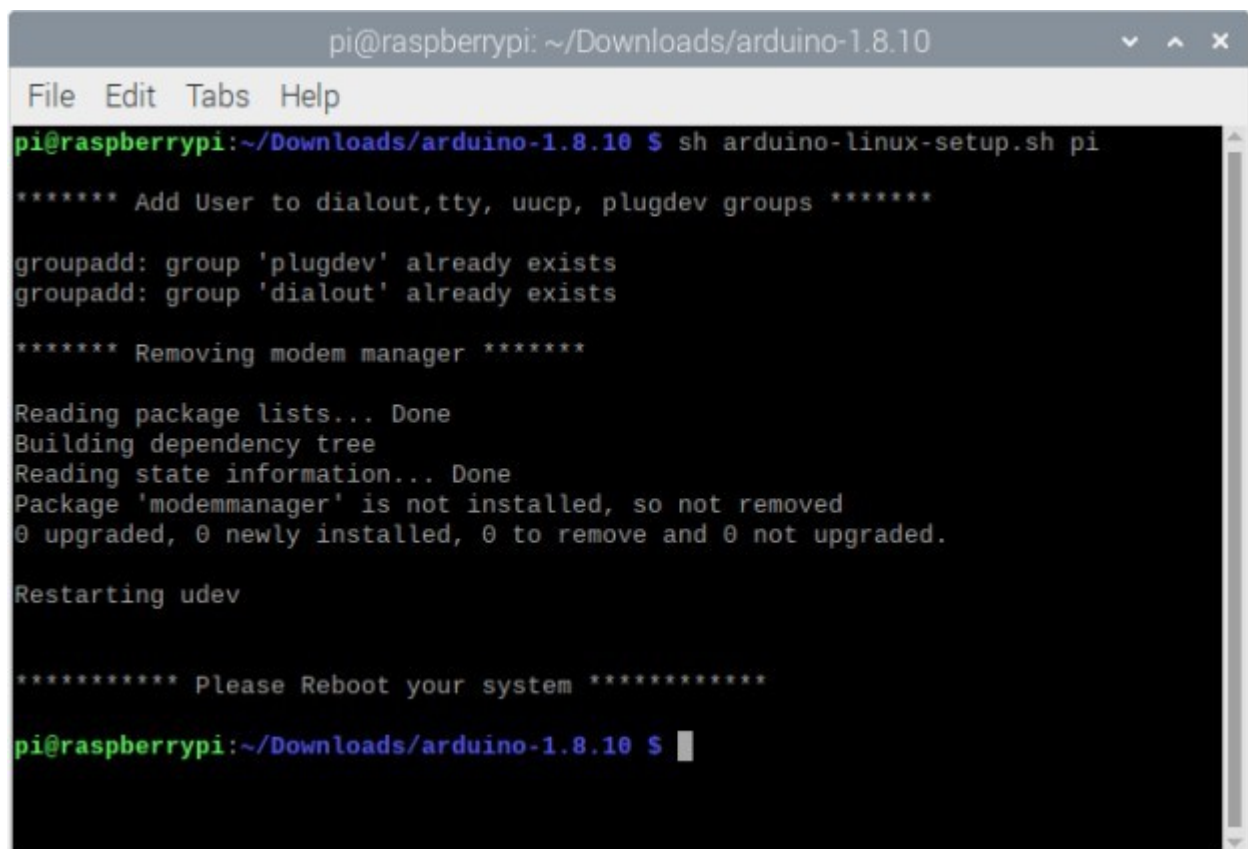


# Az-Delivery

Öffnen Sie das Terminal in dem Verzeichnis, in das die Installationsdateien entpackt wurden, und führen Sie den folgenden Befehl aus:

**sh arduino-linux-setup.sh pi**

wobei *pi* der Name des Superusers in Raspbian ist.



```
pi@raspberrypi: ~/Downloads/arduino-1.8.10
File Edit Tabs Help
pi@raspberrypi:~/Downloads/arduino-1.8.10 $ sh arduino-linux-setup.sh pi
***** Add User to dialout, tty, uucp, plugdev groups *****
groupadd: group 'plugdev' already exists
groupadd: group 'dialout' already exists
***** Removing modem manager *****
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package 'modemmanager' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Restarting udev
***** Please Reboot your system *****
pi@raspberrypi:~/Downloads/arduino-1.8.10 $
```

Um die Arduino IDE zu installieren, führen Sie den folgenden Befehl aus: **sudo sh install.sh**

# Az-Delivery

Die Arduino IDE ist nun installiert. Um Arduino IDE zu starten, öffnen Sie die App:

*Menü Anwendungen > Programmierung > Arduino IDE*



Vor den nächsten Schritten müssen zunächst die Anwendungen *pip3* und *git* installiert werden; Öffnen Sie das Terminal und führen Sie den folgenden Befehl aus.

**sudo apt install python3-pip git -y**

Die Bibliothek für Python heißt *nanpy*. Um sie zu installieren, öffnen Sie das Terminal und führen Sie den folgenden Befehl aus: **pip3 install nanpy**

```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ pip3 install nanpy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting nanpy
  Downloading https://www.piwheels.org/simple/nanpy/nanpy-0.9.6-py3-none-any.whl
    (47kB)
    100% |#####| 51kB 209kB/s
Requirement already satisfied: pyserial in /usr/lib/python3/dist-packages (from
nanpy) (3.4)
Installing collected packages: nanpy
Successfully installed nanpy-0.9.6
pi@raspberrypi:~/Scripts $
```

Nach der Installation der nanpy-Bibliothek laden Sie eine Firmware herunter, indem Sie den folgenden Befehl ausführen:

```
git clone https://github.com/nanpy/nanpy-firmware.git
```

Ändern Sie das Verzeichnis in *nanpy-firmware*, indem Sie den folgenden Befehl ausführen:

```
cd nanpy-firmware
```

Führen Sie dann den folgenden Befehl aus:

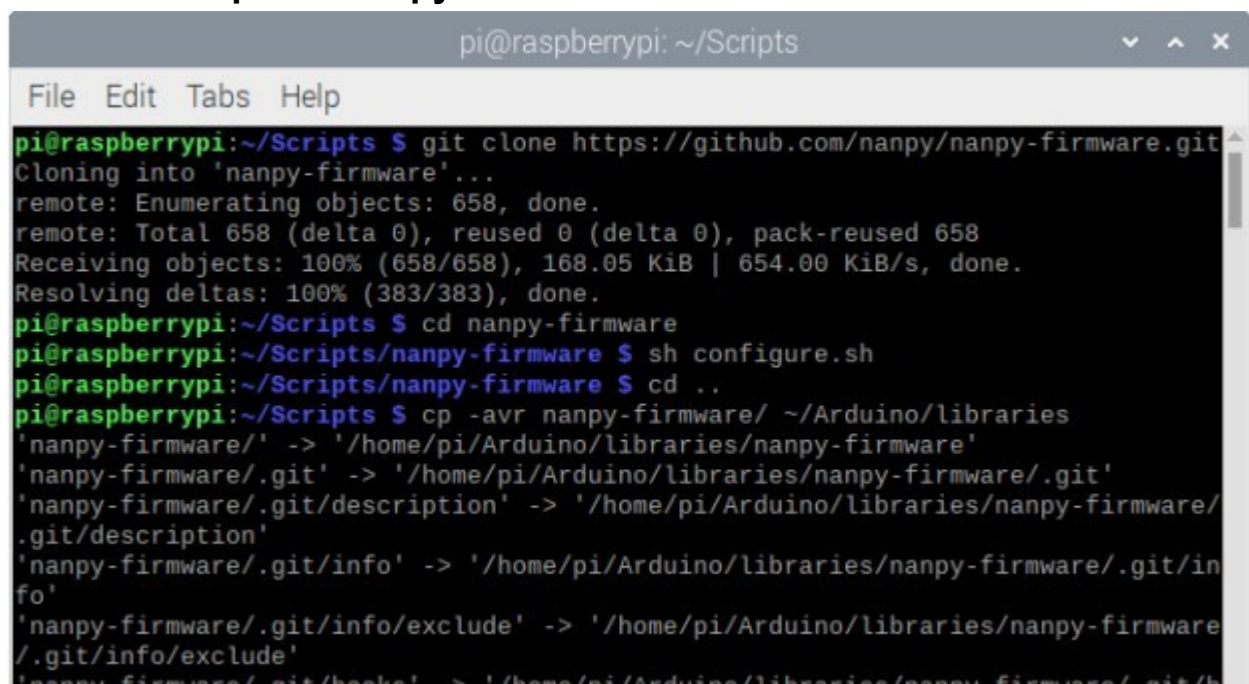
```
sh configure.sh
```

Als nächstes kopieren Sie das Verzeichnis *nanpy-firmware* in:

Verzeichnis *Arduino/Bibliotheken*.

Führen Sie dazu den folgenden Befehl aus:

```
cp -avr nanpy-firmware/ ~/Arduino/Bibliotheken
```



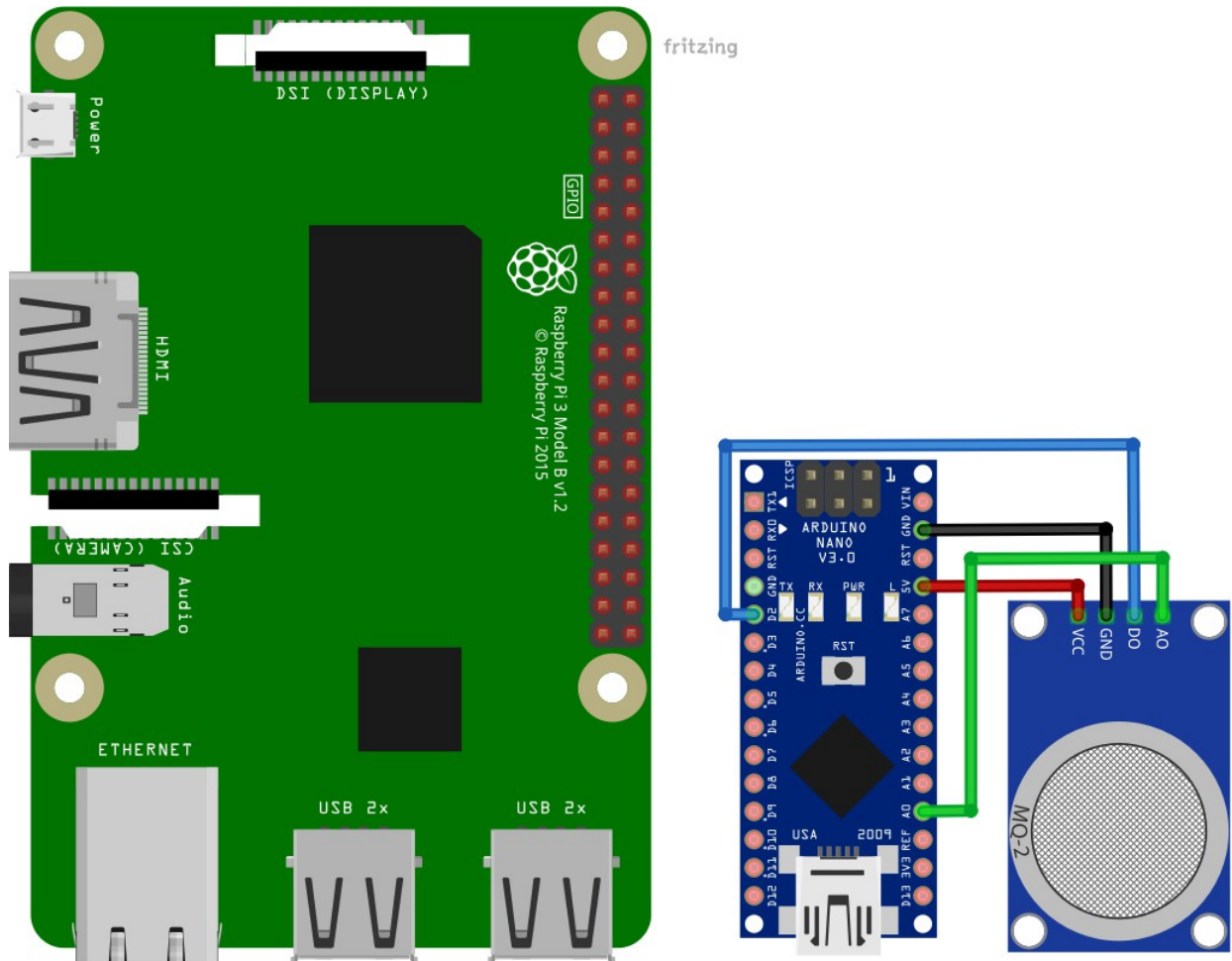
```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ git clone https://github.com/nanpy/nanpy-firmware.git
Cloning into 'nanpy-firmware'...
remote: Enumerating objects: 658, done.
remote: Total 658 (delta 0), reused 0 (delta 0), pack-reused 658
Receiving objects: 100% (658/658), 168.05 KiB | 654.00 KiB/s, done.
Resolving deltas: 100% (383/383), done.
pi@raspberrypi:~/Scripts $ cd nanpy-firmware
pi@raspberrypi:~/Scripts/nanpy-firmware $ sh configure.sh
pi@raspberrypi:~/Scripts/nanpy-firmware $ cd ..
pi@raspberrypi:~/Scripts $ cp -avr nanpy-firmware/ ~/Arduino/libraries
'nanpy-firmware/' -> '/home/pi/Arduino/libraries/nanpy-firmware'
'nanpy-firmware/.git' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git'
'nanpy-firmware/.git/description' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/description'
'nanpy-firmware/.git/info' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/info'
'nanpy-firmware/.git/info/exclude' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/info/exclude'
'nanpy-firmware/.git/hooks' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/hooks'
```

Die *nanpy-Firmware* ist nun installiert und kann verwendet werden.



## Verbinden des Moduls mit dem Raspberry Pi

Verbinden Sie das Modul mit dem Nano V3.0 wie in der folgenden Abbildung gezeigt:



Modul-Stift	Nano V3.0 Stift	Farbe des Kabels
VCC	5V	<b>Rotes Kabel</b>
GND	GND	<b>Schwarzes Kabel</b>
D0	D2	<b>Blaues Kabel</b>
A0	A0	<b>Grüner Draht</b>



Verbinden Sie den Nano V3.0 über ein USB-Kabel mit dem Raspberry Pi und öffnen Sie die Arduino IDE im Raspbian-Betriebssystem. Prüfen Sie, ob die Arduino IDE den USB-Port erkennen kann, an dem der Nano V3.0 angeschlossen ist: *Tools > Port > dev/ttyUSB0*

Gehen Sie dann zu: *Werkzeuge > Board > {Boardname}* und wählen Sie die *Nano V3.0*-Karte aus.

Um eine Skizze für die *nanpy-Firmware* zu öffnen, gehen Sie anschließend zu: *Datei > Beispiele > nanpy-firmware > Nanpy*

Laden Sie den Sketch auf den Nano V3.0 hoch. Um zu testen, ob alles richtig funktioniert, muss das einfache Blink-Skript erstellt werden, bei dem die On-Board-LED des Nano V3.0 zum Blinken verwendet wird.

Erstellen Sie das Skript *Blink.py*, und öffnen Sie es im Standardtexteditor.





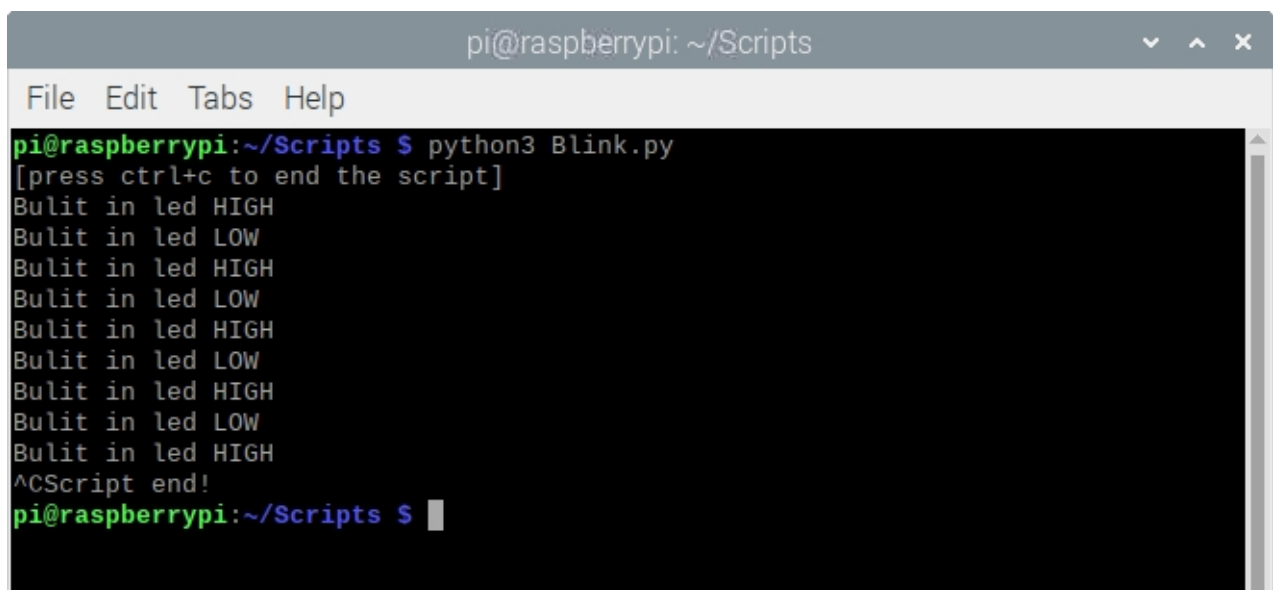
Schreiben Sie in das Skript *Blink.py* die folgenden Codezeilen:

```
from nanpy import (ArduinoApi, SerialManager) from time
import sleep
ledPin = 13
try:
    Verbindung1 = SerialManager()
    a = ArduinoApi(connection=connection1) except:
    print('Verbindung zum Arduino fehlgeschlagen')
    print('[STRG drücken
+ C zum Beenden des Skripts!]) a.pinMode(ledPin, a.OUTPUT) #
Arduino-Versuch einrichten:
while True:
    a.digitalWrite(ledPin, a.HIGH)
    print('Bulit in led HIGH')
    sleep(1)
    a.digitalWrite(ledPin, a.LOW)
    print('Bulit in led LOW')
    sleep(1)

except KeyboardInterrupt:
    print('\nScript end!')
    a.digitalWrite(ledPin, a.LOW)
```

Speichern Sie das Skript unter dem Namen *Blink.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 Blink.py**

Das Ergebnis sollte wie in der folgenden Abbildung aussehen:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 Blink.py
[press ctrl+c to end the script]
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
^CScript end!
pi@raspberrypi:~/Scripts $
```

Um das Skript zu stoppen, drücken Sie 'CTRL + C' auf der Tastatur.

Die LED, die an den digitalen Pin 13 des Nano V3.0 angeschlossen ist, sollte im Sekundentakt zu blinken beginnen.

# Az-Delivery

Das Skript beginnt mit dem Import von zwei Bibliotheken, den `nanpy`-Bibliotheksfunktionen und der *Zeit*.

Dann wird die Variable *ledPin* erstellt und mit der Nummer `13` initialisiert. Die Nummer `13` steht für die Nummer des digitalen Pins, an dem die LED angeschlossen ist (On-Board-LED des Nano V3.0).

Danach wird der `try-except`-Codeblock verwendet, um eine Verbindung zum Nano V3.0 herzustellen. Wenn die Verbindung nicht erfolgreich ist, wird eine Meldung ausgegeben:

Im Terminal wird angezeigt, dass *die Verbindung zum Mikrocontroller-Board fehlgeschlagen ist*.

Wenn die Verbindung erfolgreich ist, wird ein Kommunikationsobjekt namens `"a"` erstellt und initialisiert. Das Objekt `"a"` repräsentiert das Nano V3.0 Board. Jede Funktion, die in der Arduino-IDE verwendet wird, kann mit dem `"a"`-Objekt verwendet werden, wie es im Code zu sehen ist.

Mit der folgenden Code-Zeile wird der Pin-Modus für den digitalen Pin eingestellt

```
13: a.pinMode(ledPin, a.OUTPUT)
```

Dann wird im Endlosschleifenblock (*while True:*) die Funktion *digitalWrite()* verwendet, um den Zustand des digitalen Pins `13` (HIGH- oder LOW-Zustand) zu setzen. Mit der Funktion *digitalWrite()* kann die an Pin `13` angeschlossene LED ein- oder ausgeschaltet werden.



Im Endlosschleifen-Block wird die LED zunächst für eine Sekunde *ein- und* dann für eine Sekunde *ausgeschaltet*. Dies wird als *Blinken der LED* bezeichnet. Das Zeitintervall für ein einzelnes Blinken kann in der folgenden Codezeile geändert werden: *sleep(1)*

Dabei steht die Zahl *1* für die Anzahl der Sekunden für die Dauer des Zeitintervalls.

Um die Endlosschleife zu beenden, drücken Sie 'CTRL + C' auf der Tastatur. Dies wird als Tastatur-Interrupt bezeichnet, der im Except-Block gesetzt wird (*except KeyboardInterrupt*). Im Except-Block wird die On-Board-LED ausgeschaltet.



## Python-Skript für das Modul MQ-135

```
von nanpy importieren (ArduinoApi,  
SerialManager) importieren time
```

versuchen:

```
Verbindung_1 = SerialManager()  
a = ArduinoApi(connection=connection_1)  
except:  
    print('Verbindung zum Arduino fehlgeschlagen')  
    DIGITAL_PIN = 2  
    ANALOG_PIN = 0  
    time.sleep(2)  
    print('Sensor wärmt sich auf...')  
    print('[Drücken Sie CTRL+C, um das  
    Skript zu beenden]') time.sleep(5) #  
    Sensor wärmt sich auf...
```

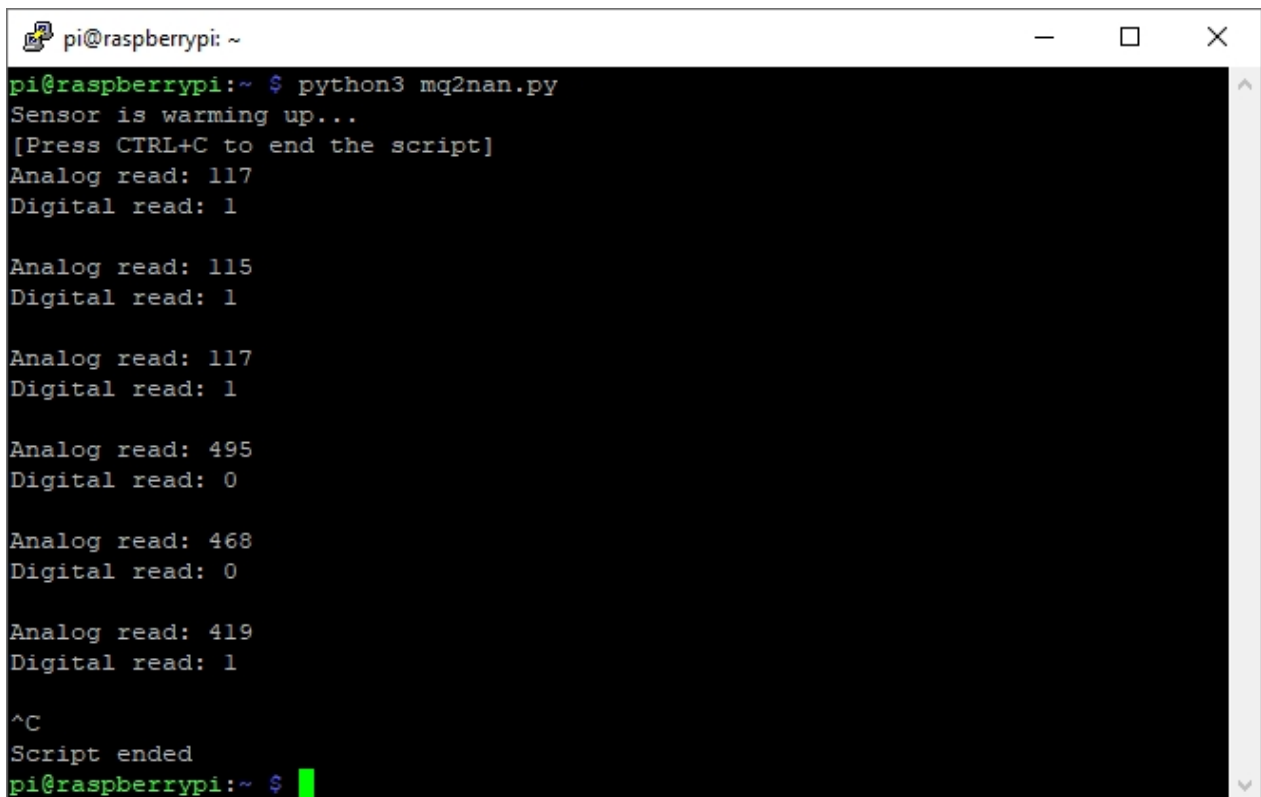
versuchen:

```
while True:  
    analogLesen = a.analogLesen(ANALOG_PIN) digitalLesen =  
    a.digitalLesen(DIGITAL_PIN)  
    print("Analog gelesen: {}\nDigital gelesen: {}".format(analogLesen, digitalLesen))  
    time.sleep(2)  
except KeyboardInterrupt:  
    print("\nSkript Ende!")
```

# Az-Delivery

Speichern Sie das Skript unter dem Namen *mq2nan.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 mq2nan.py**

Das Ergebnis sollte wie in der folgenden Abbildung aussehen:



```
pi@raspberrypi: ~  
pi@raspberrypi:~$ python3 mq2nan.py  
Sensor is warming up...  
[Press CTRL+C to end the script]  
Analog read: 117  
Digital read: 1  
  
Analog read: 115  
Digital read: 1  
  
Analog read: 117  
Digital read: 1  
  
Analog read: 495  
Digital read: 0  
  
Analog read: 468  
Digital read: 0  
  
Analog read: 419  
Digital read: 1  
  
^C  
Script ended  
pi@raspberrypi:~$
```

Um das Skript zu stoppen, drücken Sie 'CTRL + C' auf der Tastatur.



Jetzt ist es an der Zeit, zu lernen und eigene Projekte zu erstellen. Das können Sie mit Hilfe vieler Beispielskripte und anderer Anleitungen tun, die Sie im Internet finden können.

**Wenn Sie auf der Suche nach hochwertiger Mikroelektronik und Zubehör sind, sind Sie bei der AZ-Delivery Vertriebs GmbH an der richtigen Adresse. Sie erhalten zahlreiche Anwendungsbeispiele, vollständige Installationsanleitungen, eBooks, Bibliotheken und Unterstützung durch unsere technischen Experten.**

<https://az-delivery.de>

**Viel Spaß!**

**Impressum**

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