

Den MQ-135 Sensor an einem Arduino Mikrocontroller verwenden

Der MQ-135 Sensor ist ein Modul, welches die Konzentration verschiedener Schadstoffe in der Luft erfasst. Der Sensor gibt die erfassten Schadstoffwerte als analogen Wert aus. Diese können im Anschluss im seriellen Monitor ausgegeben oder anderweitig verarbeitet werden.

Doch bevor wir uns diesem Teil der Anleitung widmen, blicken wir zuerst auf den Sensor selbst.

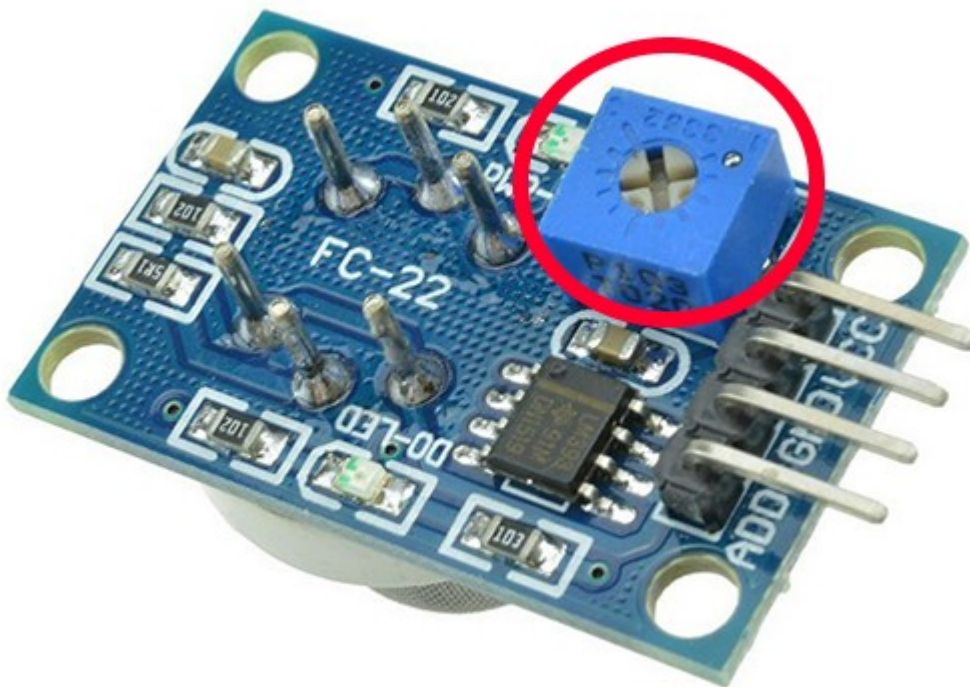
Pinbelegung

Der MQ-135 verfügt über **vier** Pins:

- **VCC** = Pin für die Stromversorgung, anzuschließen an den 5V Pin des Arduinos
- **GND** = Ground-Pin, anzuschließen an den GND Pin des Arduinos
- **A0** = Pin für den analogen Output, anzuschließen an den Pin A0 des Arduinos
- **D0** = Pin für den digitalen Output, anzuschließen an den Pin D0 des Arduinos

Zudem können wir den Schwellenwert, welcher über den Pin D0 als „1“ bzw. „0“ ausgegeben wird, mechanisch einstellen. Die Kalibrierung des Schwellenwertes erfolgt durch Drehen des Potentiometers, welches auf der Rückseite des Sensormoduls zu finden ist.

Das Potentiometer sieht wie folgt aus:



Nachdem wir unseren MQ-135 Sensor wie oben beschrieben an den Arduino Mikrocontroller angeschlossen haben, beginnen wir mit der Programmierung.

Im Prinzip möchten wir lediglich die erfassten Daten des MQ-135 auslesen. Dies erfolgt durch eine simple Abfrage.

Beispielsketch 1 - Grundabfrage

```
int Sensorwert;

void setup()
{
    Serial.begin(9600);           // Damit unsere Werte
    // später im seriellen Monitor angezeigt werden können,
    // legen wir zuerst die Baudrate 9600 fest.

    pinMode( 0 , INPUT);         // Pin 0 dient nun als
    // Input
}

void loop()
{
    Sensorwert = analogRead(0);   // Zuerst wird der
    // Wert am Pin A0 ausgelesen...

    Serial.print(„Der erfasste Wert lautet : „);

    Serial.println(Sensorwert, DEC); //... und
    // anschließend im seriellen Monitor als Dezimalzahl
    // ausgegeben.

    Serial.println();

    delay(1000);
}
```

Wenn wir uns nun die Werte im seriellen Monitor ansehen, werden wir feststellen, dass diese allein relativ nutzlos sind. Zu diesem Zeitpunkt liegt noch kein Referenzwert vor, der aussagt, wie gut oder schlecht die Luftqualität jetzt tatsächlich ist. Daher müssen wir dem Sensor Frischluft zuführen und ihm somit erlauben, sich zu kalibrieren. Nachdem wir den Sensor an der frischen Luft ruhen

lassen haben, können wir den erfassten Wert als Ausgangspunkt für die Bewertung der Luftqualität nutzen.

In unserem Fall lag dieser Wert bei ungefähr 30. **Dieser Wert kann jedoch variieren!**

Diesen erfassten Referenzwert nutzen wir, um uns eine kleine Kontrollstation zu basteln.

Sobald der Sensorwert **über 40** liegt, soll eine **rote LED** für zwei Sekunden aufleuchten. Liegt der Sensorwert **unter 40**, leuchtet eine **grüne LED** zwei Sekunden lang auf.

Wir erweitern also unseren Aufbau um zwei LEDs und passen den Sketch entsprechend an.

Beispielsketch 2 - Abfrage mit Kontrollwert

```
int Sensorwert;

int LEDgruen = 13;           // Die
grüne LED wird mit Pin13 verbunden

int LEDrot = 12;             // Die rote
LED wird mit Pin12 verbunden

void setup()
{
    Serial.begin(9600);       // Damit unsere Werte
    später im seriellen Monitor angezeigt werden können,
    legen wir zuerst die Baudrate 9600 fest.

    pinMode( 0 , INPUT);      // Pin 0 dient nun als
    Input

    pinMode(LEDgruen, OUTPUT); // Die LEDs dienen als
    Output

    pinMode(LEDrot, OUTPUT);
}

void loop()
{
    Sensorwert = analogRead(0); // Zuerst wird der
    Wert am Pin A0 ausgelesen...
```

```
Serial.print(„Der erfasste Wert lautet : „);
```

```
Serial.println(Sensorwert, DEC);    //... und  
anschließend im seriellen Monitor als Dezimalzahl  
ausgegeben.
```

```
Serial.println();
```

```
if (Sensorwert >= 40)  // Sollte der SensorWert  
größer/gleich 50 sein...
```

```
{
```

```
digitalWrite(LEDrot, HIGH); //... soll eine rote LED...
```

```
delay(2000);           //... für 2 Sekunden aufleuchten...
```

```
digitalWrite(LEDrot, LOW); //... und anschließend wieder  
ausgeschaltet werden.
```

```
}
```

```
if (Sensorwert <= 39)  // Sollte der SensorWert  
kleiner/gleich 49 sein...
```

```
{
```

```
digitalWrite(LEDgruen,HIGH); //... soll eine grüne LED...
```

```
delay(2000);           //... für 2 Sekunden aufleuchten...
```

```
digitalWrite(LEDgruen,LOW); //... und anschließend wieder  
ausgeschaltet werden.
```

```
}
```

```
delay(1000);
```

```
}
```