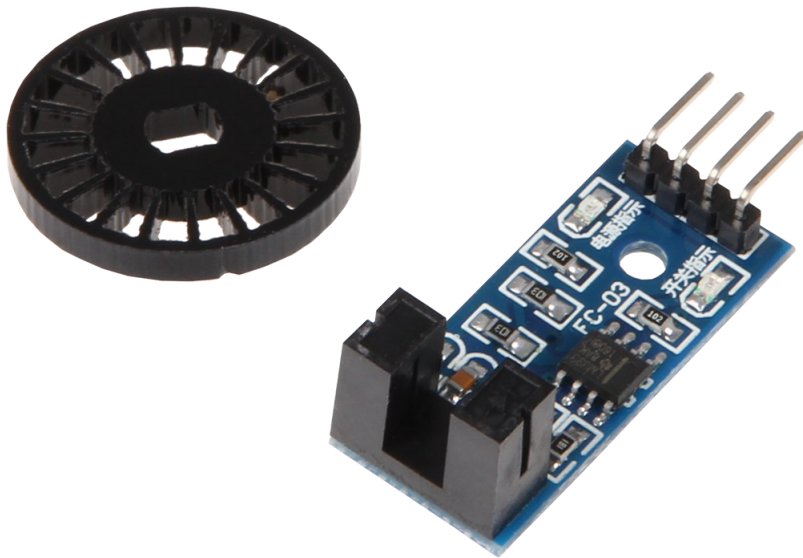


GESCHWINDIGKEITSSENSOR

Speedsensor LM393 mit Lochscheibe



1. ALLGEMEINE INFORMATIONEN

Sehr geehrter Kunde,
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

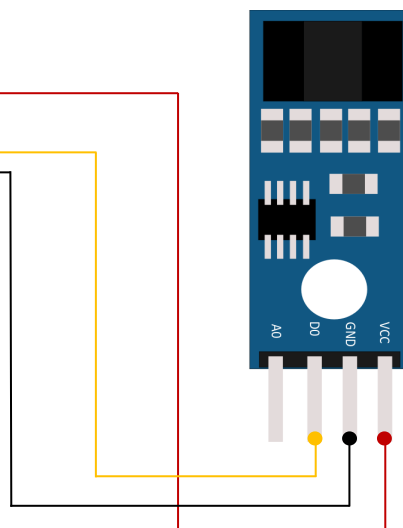
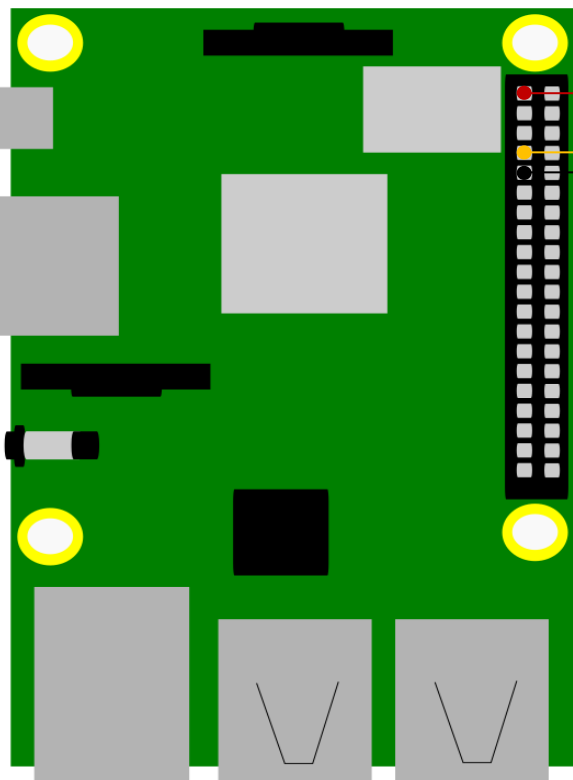
Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

2. VERWENDUNG MIT DEM RASPBERRY PI



Diese Anleitung wurde unter Raspberry Pi OS Bookworm für den Raspberry Pi 4 und 5 geschrieben. Es wurde keine Überprüfung mit neueren Betriebssystemen oder Hardware durchgeführt.

1. Anschluss



Raspberry Pi	SEN-Speed
Pin 1 (3,3 V)	VCC
Pin 9 (Ground)	GND
GPIO 4 (Pin 7)	D0
-	A0

2. Beispielcode Raspberry Pi (Python)

Dieser Sensor wird verwendet um die Geschwindigkeit von Motoren zu messen. Dabei wird eine Encoder-Scheibe mit Löchern (hier: 20), welche durch eine Lichtschranke hindurch laufen verwendet. Diese Scheibe ist dabei an der Welle des Motors befestigt. Mit Hilfe der Lichtschranke können dadurch die Löcher der Encoder-Scheibe gezählt, sodass die Umdrehungen gezählt werden können.

Im folgenden finden Sie einen Beispielcode mit dem Sie die Drehzahl der Scheibe pro Minute in einem 10 Sekundentakt ausgegeben bekommen.

Dazu erstellen Sie zunächst eine Datei mit dem folgenden Befehl:

```
nano sen-speed.py
```

Kopieren Sie danach den folgenden Beispielcode in Ihre Datei.

Alternativ können Sie sich die Dateien auch von unserer Website joy-it.net herunterladen.

```

from gpiozero import DigitalInputDevice
import threading
import time

class ResettableTimer(threading.Thread):
    def __init__(self, interval, function):
        super().__init__()
        self.interval = interval
        self.function = function
        self.timer = None
        self.running = False
        self.daemon = True # Optionally make this a daemon thread

    def run(self):
        self.running = True
        while self.running:
            self.timer = threading.Timer(self.interval, self.function)
            self.timer.start()
            self.timer.join()

    def stop(self):
        self.running = False
        if self.timer:
            self.timer.cancel()

def count():
    global counter
    counter += 1

def output():
    global counter
    speed = int((counter / 2) * calc / wheel)
    print("Drehzahl pro Minute:", speed)
    counter = 0

counter = 0
pin = 4 # Using BCM pin numbering
interval = 10.0
calc = 60 / interval
wheel = 20

# Setup the GPIO pin
device = DigitalInputDevice(pin, pull_up=False)
device.when_activated = count

# Create and start the timer
timer = ResettableTimer(interval, output)
timer.start()

try:
    # Keep the main program running
    while True:
        time.sleep(1)
except KeyboardInterrupt:
    timer.stop()
    timer.join()
    device.close()

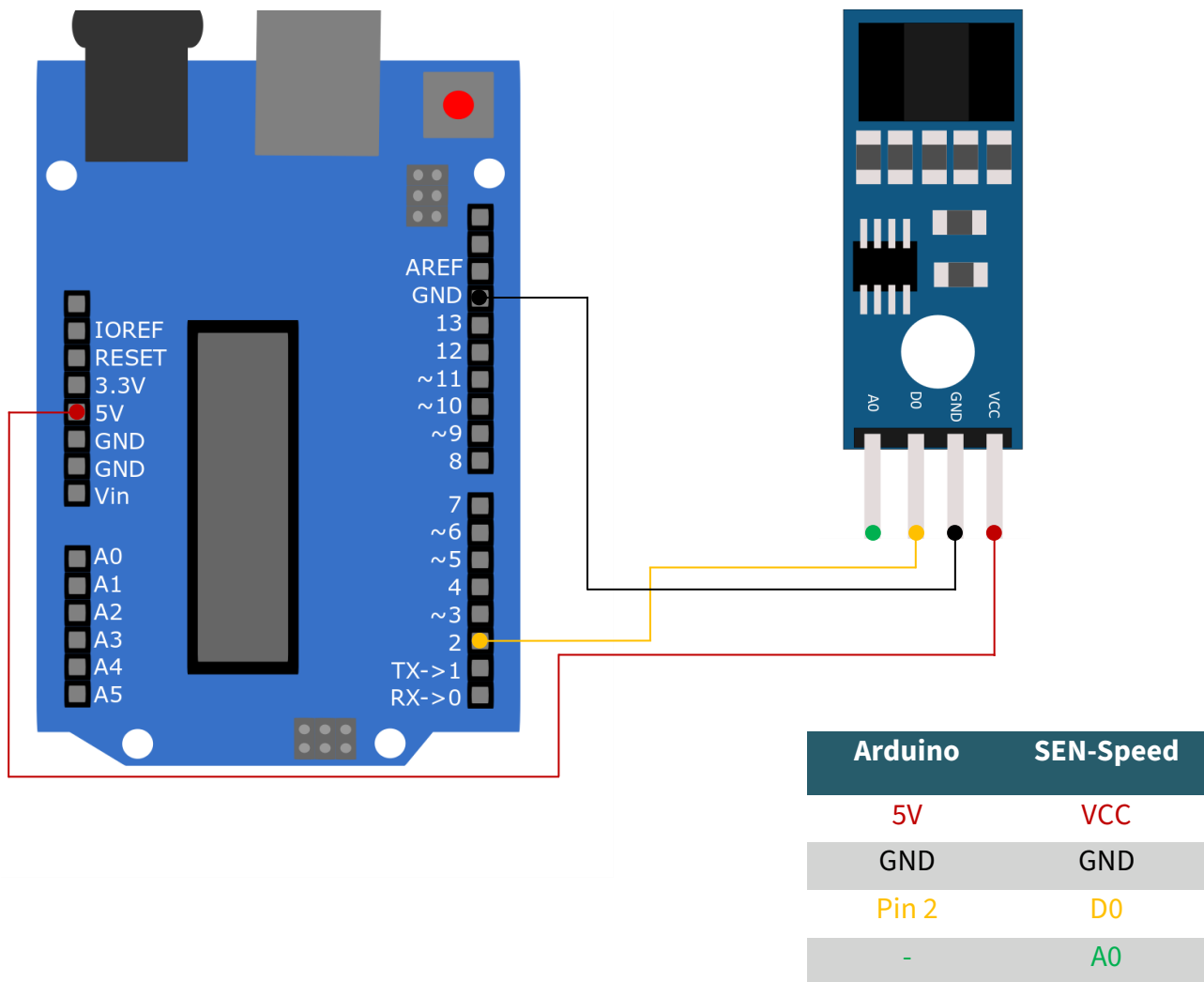
```

Speichern Sie die Datei mit **STRG+O** und verlassen Sie den Editor mit **STRG+X**. Nun können Sie den Beispielcode mit dem folgenden Befehl ausführen.

```
python3 sen-speed.py
```

3. VERWENDUNG MIT DEM ARDUINO

1. Anschluss



2. Beispielcode Arduino

Dieser Sensor wird verwendet um die Geschwindigkeit von Motoren zu messen. Dabei wird eine Encoder-Scheibe mit Löchern (hier: 20), welche durch eine Lichtschranke hindurch laufen verwendet. Diese Scheibe ist dabei an der Welle des Motors befestigt. Mit Hilfe der Lichtschranke können dadurch die Löcher der Encoder-Scheibe gezählt, sodass die Umdrehungen gezählt werden können.

Im folgenden finden Sie einen Beispielcode mit dem Sie die Drehzahl der Scheibe pro Minute in einem 5 Sekundentakt ausgegeben bekommen.

Um diesen Beispielcode verwenden zu können, achten Sie darauf, dass Port und Board richtig in Ihrer Arduino IDE ausgewählt sind. Es muss auch die Baudrate des seriellen Monitors auf 9600 gesetzt werden, damit Sie die Ausgaben im seriellen Monitor sehen können. Des Weiteren muss die Bibliothek [TimerOne](#) installiert sein.

Nachdem Sie dies überprüft haben, kopieren Sie den folgenden Code in Ihre Arduino IDE.

Alternativ können Sie sich die Dateien auch von unserer Website joy-it.net herunterladen.

```

// sen-speed Demo
// Der Code misst ueber einen Zeitraum (5 Sekunden voreingestellt)
// die Umdrehungen der Encoder-Scheibe, rechnet diese dann auf Umdrehungen
// pro Minute um und gibt diese über die serielle Schittstelle aus.

// Import einer Bibliothek
#include "TimerOne.h"
#define pin 2

// Benoetigte Variablen
int interval, wheel, counter;
unsigned long previousMicros, usInterval, calc;

void setup()
{
  counter = 0; // counter auf 0 setzen
  interval = 5; // 5 Sekunden Intervall
  wheel = 20; // Loecher in der Encoder-Scheibe

  calc = 60 / interval; // Intervall auf 1 Minute hoch rechnen
  usInterval = interval * 1000000; // Intervallzeit fuer den Timer in
  // Mikrosekunden umrechnen
  wheel = wheel * 2; // Anzahl der Loecher in der Encoder-Scheibe mit 2
  // multiplizieren, da der Interrupt bei jeder
  // Aenderung des Signals ausgefuehrt wird

  pinMode(pin, INPUT); // Setzen des analogen Pins als Input
  Timer1.initialize(usInterval); // Timer initialisieren auf dem Intervall
  attachInterrupt(digitalPinToInterrupt(pin), count, CHANGE);
  // fuehrt count aus, wenn sich das Signal am Pin 2 aendert

  Timer1.attachInterrupt(output); // fuehrt nach Intervall output aus
  Serial.begin(9600); // starten der seriellen Schnittstelle mit 9600 Baud
}

// Zaehlt Loecher der Encoder-Scheibe (mit Filter)
void count(){
  if (micros() - previousMicros >= 700) {
    counter++;
    previousMicros = micros();
  }
}

// Ausgabe im seriellen Monitor
void output(){
  Timer1.detachInterrupt(); // Unterbricht Timer
  Serial.print("Drehzahl pro Minute: ");
  int speed = ((counter)*calc) / wheel;
  // Berechnung der Umdrehungen pro Minute

  Serial.println(speed);
  counter = 0; // zuruecksetzen des Zaehlers
  Timer1.attachInterrupt(output); // startet Timer erneut
}

void loop(){
  // keine loop notwendig
}

```

Sie können den Code mit Klicken auf die Taste "Hochladen" ausführen.

4. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem Elektroggesetz (ElektroG)



Symbol auf Elektro- und Elektronikgeräten:

Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

Rückgabemöglichkeiten:

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte bei denen keine äußere Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in haushaltsüblichen Mengen abgeben werden.

Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

Möglichkeit Rückgabe in Ihrer Nähe:

Wir senden Ihnen eine Paketmarke zu mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an Service@joy-it.net oder per Telefon an uns.

Informationen zur Verpackung:

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

5. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net

Ticket-System: <https://support.joy-it.net>

Telefon: +49 (0)2845 9360 – 50 (Mo - Do: 09:00 - 17:00 Uhr,
Fr: 09:00 - 14:30 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net