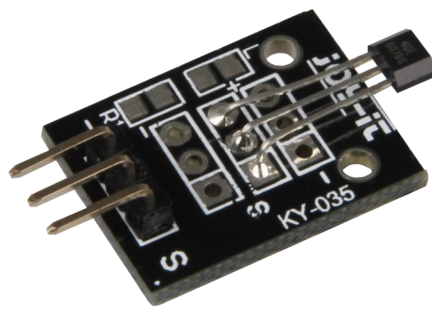


JOY-IT



Bihor Magnet Sensor
KY-035

INHALTSVERZEICHNIS

1. Übersicht
2. Pin-Belegung
3. Codebeispiel Arduino
4. Codebeispiel Raspberry Pi
5. Informations- und Rücknahmepflichten
6. Support

1. ÜBERSICHT

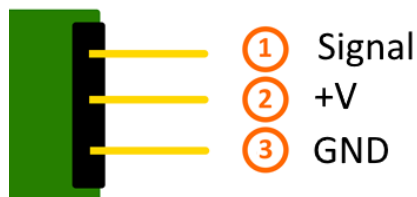
Sehr geehrter Kunde,

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.

Chipsatz	AH49E
Ausgang	Analoges Spannungssignal, welches die Stärke des Magnetfelds angibt

2. PIN-BELEGUNG



Anschlussbelegung Arduino	
Sensor Signal	Pin A5
+V	Pin 5V
GND	Pin GND

Der Raspberry Pi besitzt im Gegensatz zum Arduino keine analogen Eingänge bzw. es ist kein ADC (analog digital Converter) im Chip des Raspberry Pis integriert. Dies schränkt den Raspberry Pi ein, wenn man Sensoren einsetzen möchte, bei denen nicht digitale Werte ausgegeben werden, sondern es sich hier um einen kontinuierlichen, veränderlichen Wert handeln sollte (Beispiel: Potentiometer -> Andere Position = Anderer Spannungswert)

Um diese Problematik zu umgehen, gibt es unser **KY-053** Modul, mit 16 Bit genauen ADC, welches Sie am Raspberry nutzen können, um diesen um 4 analoge Eingänge erweitern zu können. Dieses wird per I2C an den Raspberry Pi angeschlossen, übernimmt die analoge Messung und gibt den Wert digital an den Raspberry Pi weiter.

Somit empfehlen wir, bei analogen Sensoren, das KY-053 Modul mit dem besagten ADC dazwischenzuschalten.

Anschlussbelegung Raspberry Pi	
+V	3,3V [Pin 1]
GND	Masse [Pin 6]
Analoges Signal	Analog 0 (ADS1115 - KY053)

Anschlussbelegung ADS1115	
VDD	3,3V [Pin 1]
GND	Masse [Pin 9]
SCL	GPIO03 / SCL [Pin 5]
SDA	GPIO02 / SDA [Pin 3]
A0	Analoges Signal (Sensor)

3. CODEBEISPIEL ARDUINO

Das Programm liest den aktuellen Spannungswert aus, der am analogen Ausgang gemessen werden kann, und gibt diesen auf der seriellen Schnittstelle aus.

Zudem wird ebenfalls der Zustand des digitalen Pins in der Konsole angegeben. Dies gibt an, ob der Grenzwert unterschritten wurde oder nicht.

```
int sensorPin = A5; // Hier wird der Eingangs-Pin deklariert

// Serielle Ausgabe in 9600 Baud
void setup()
{
    Serial.begin(9600);
}

// Das Programm misst den aktuellen Spannungswert am Sensor,
// berechnet aus diesen und dem bekannten Serienwiderstand den aktuellen
// Widerstandswert des Sensors und gibt die Ergebnisse auf der serielle Ausga-
// be aus

void loop()
{
    // Aktueller Spannungswert wird gemessen...
    int rawValue = analogRead(sensorPin);
    float voltage = rawValue * (5.0/1023) * 1000;

    float resitance = 10000 * ( voltage / ( 5000.0 - voltage) );

    // ... und hier auf die serielle Schnittstelle ausgegeben
    Serial.print("Spannungswert:"); Serial.print(voltage); Serial.print("mV");
    Serial.print(", Widerstandswert:"); Serial.print(resitance); Seri-
al.println("Ohm");
    Serial.println("-----");

    delay(500);
}
```

4. CODEBEISPIEL RASPBERRY PI

Das Programm nutzt zur Ansteuerung des ADS1115 die entsprechende **Adafruit Python ADS1x15** Bibliothek der Firma Adafruit. Diese wurden unter dem folgenden [Link](#) unter der [MIT OpenSource-Lizenz](#) veröffentlicht. Diese muss zunächst installiert werden.

Öffnen Sie dazu ein Terminalfenster auf Ihrem Raspberry Pi und geben Sie die folgenden Befehle ein:

```
sudo apt-get install git build-essential python-dev
```

Laden Sie nun mit dem folgenden Befehl die aktuellste Version der Adafruit_DHT Library herunter:

```
git clone https://github.com/adafruit/Adafruit_Python_ADS1x15.git
```

Installieren Sie nun die Bibliothek:

```
cd Adafruit_Python_ADS1x15
```

```
sudo python setup.py install
```

Das Codebeispiel können Sie [hier](#) herunterladen. Alternativ können Sie auch den folgenden Ausschnitte kopieren.

Starten Sie das Beispiel dann mit dem folgenden Befehl:

```
sudo python Adafruit_ADS1x15.py
```

```
#
#!/usr/bin/python
# coding=utf-8

#####
#####
### Copyright by Joy-IT
### Published under Creative Commons Attribution-NonCommercial-ShareAlike 3.0
Unported License
### Commercial use only after permission is requested and granted
###
### Analog Sensor + ADS1115 ADC - Raspberry Pi Python Code Example
###
#####
#####

# Dieser Code nutzt die ADS1115 und die I2C Python Library fuer den Raspberry
Pi
# Diese ist unter folgendem Link unter der BSD Lizenz veroeffentlicht
# [https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code]
from Adafruit_ADS1x15 import ADS1x15
from time import sleep

# Weitere benoetigte Module werden importiert und eingerichtet
import time, signal, sys, os
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Benutzte Variablen werden initialisiert
delayTime = 0.2

# Adresszuweisung ADS1x15 ADC

ADS1015 = 0x00 # 12-bit ADC
ADS1115 = 0x01 # 16-bit

# Verstaerkung (Gain) wird ausgewaehlt
gain = 4096 # +/- 4.096V
# gain = 2048 # +/- 2.048V
# gain = 1024 # +/- 1.024V
# gain = 512 # +/- 0.512V
# gain = 256 # +/- 0.256V
```

```
# Abtasterate des ADC (SampleRate) wird ausgewaehlt
# sps = 8      # 8 Samples pro Sekunde
# sps = 16     # 16 Samples pro Sekunde
# sps = 32     # 32 Samples pro Sekunde
sps = 64      # 64 Samples pro Sekunde
# sps = 128   # 128 Samples pro Sekunde
# sps = 250   # 250 Samples pro Sekunde
# sps = 475   # 475 Samples pro Sekunde
# sps = 860   # 860 Samples pro Sekunde

# ADC-Channel (1-4) wird ausgewaehlt
adc_channel = 0    # Channel 0
# adc_channel = 1    # Channel 1
# adc_channel = 2    # Channel 2
# adc_channel = 3    # Channel 3

# Hier wird der ADC initialisiert - beim KY-053 verwendeten ADC handelt es
sich um einen ADS1115 Chipsatz
adc = ADS1x15(ic=ADS1115)

# Hier waehlt man den Eingangs-Pin des digitalen Signals aus
Digital_PIN = 24
GPIO.setup(Digital_PIN, GPIO.IN, pull_up_down = GPIO.PUD_OFF)

#####
#####

# #####
# Hauptprogrammschleife
# #####
# Das Programm liest die aktuellen Werte der Eingang-Pins
# und gibt diese in der Konsole aus

try:
    while True:
        #Aktuelle Werte werden aufgenommen
        analog = adc.readADCSingleEnded(adc_channel, gain, sps)
        # Ausgabe auf die Konsole
        if GPIO.input(Digital_PIN) == False:
            print "Analoger Spannungswert:", analog,"mV, ",
            "Grenzwert: noch nicht erreicht"
        else:
            print "Analoger Spannungswert:", analog, "mV, ",
            "Grenzwert: erreicht"
            print "-----"
```



```
# Reset + Delay
button_pressed = False
time.sleep(delayTime)

except KeyboardInterrupt:
    GPIO.cleanup()
```

5. INFORMATIONS- UND RÜCKNAHMEPFLICHTEN

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 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.

6. 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: <http://support.joy-it.net>

Telefon: +49 (0)2845 98469 – 66 (9:30 - 17:00 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net