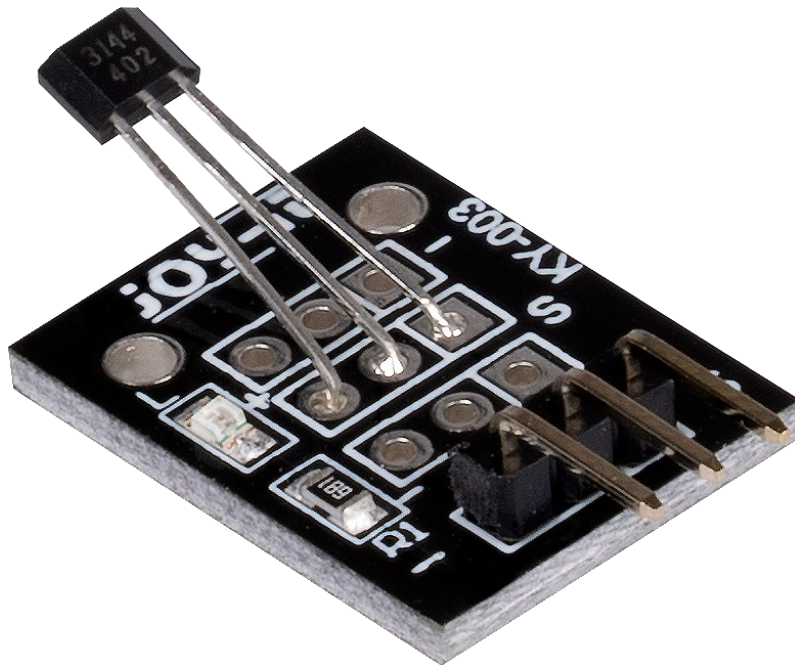


HALL MAGNETFELDESENSOR

SEN-KY003HMS



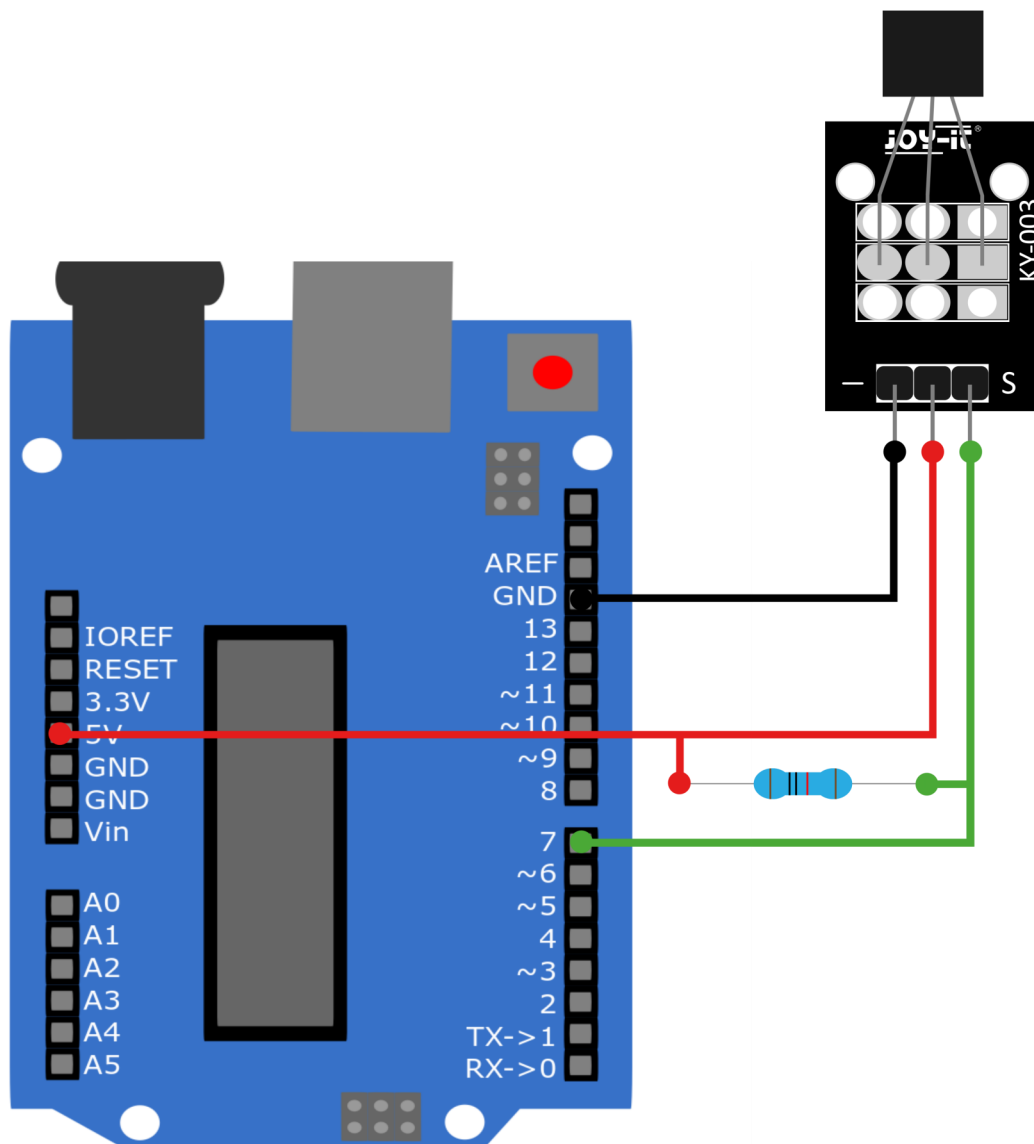
1. ALLGEMEINE INFORMATIONEN

Sehr geehrte/r Kunde/in,
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 ARDUINO

Schließen Sie Ihren Sensor, wie im Schaubild und der Tabelle unten zusehen, an Ihren Arduino an. Für optimale Ergebnisse sollten Sie einen Pullup Widerstand (ca. 10 k Ω) zwischen der Signalleitung des Sensors und der 5 V Leitung verwenden.



Arduino	Sensor
GND	GND -
5V	VCC +
Pin 7	Signal

Codebeispiel

In dem nun folgenden Codebeispiel, wird jede Sekunde im seriellen Monitor angegeben, ob ein Magnetfeld erkannt wurde. Kopieren Sie den Beispielcode und laden Sie ihn auf Ihren Arduino hoch. Achten Sie darauf, dass Sie das richtige Board und den richtigen Port in Ihrer Arduino IDE ausgewählt haben.

```
// Deklaration und Initialisierung des Eingang-Pins

int Digital_Eingang = 7; // Digital-Pin

void setup ()
{
  pinMode (Digital_Eingang, INPUT);

  Serial.begin (9600); // Serielle Ausgabe mit 9600 bps
}

// Das Programm liest den aktuellen Werte des Eingang-Pins
// und gibt diese auf der seriellen Ausgabe aus
void loop ()
{
  int Digital;

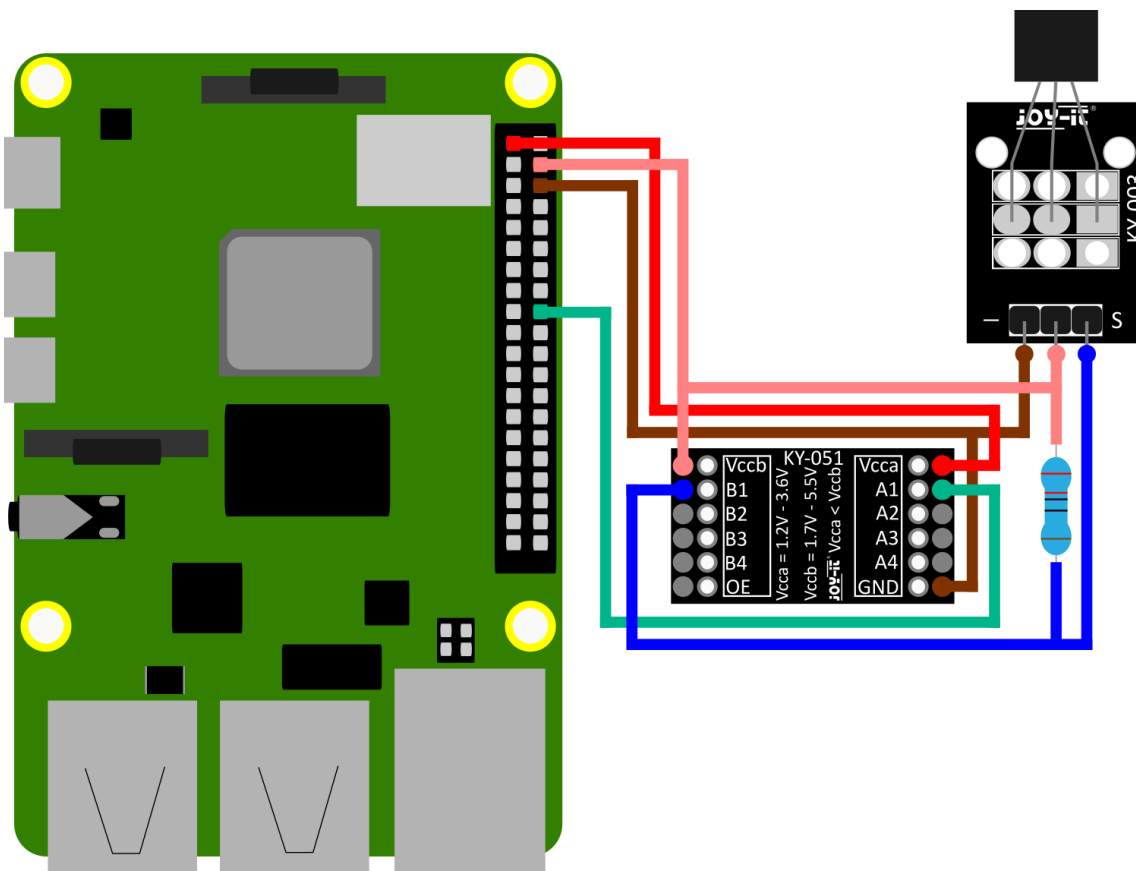
  //Aktueller Wert wird ausgelesen
  Digital = digitalRead (Digital_Eingang);
  Serial.print ("Magnetfeld");
  if(Digital==1)
  {
    Serial.println (" nicht erkannt");
  }
  else
  {
    Serial.println (" erkannt");
  }
  Serial.println ("-----");
  delay (1000);
}
```

3. 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.

Schließen Sie Ihren Sensor, wie im Schaubild und der Tabelle zusehen an einen Voltage Translator und Ihren Raspberry Pi an. Hierbei empfehlen wir den [KY-051 Voltage Translator von Joy-IT](#). Für optimale Ergebnisse sollten Sie einen Pullup Widerstand (ca. 10 k Ω) zwischen der Signalleitung des Sensors und der 5V Leitung verwenden.



Raspberry Pi	Sensor	Voltage Translator
-	Signal	B1
5V (Pin 4)	+V	
Masse (Pin 6)	GND	GND
3,3V (Pin 1)	-	Vcca
5V (Pin 4)	-	Vccb
GPIO 24 (Pin 18)	-	A1

Codebeispiel

Erstellen Sie auf Ihrem Raspberry Pi nun eine neue Datei mit dem folgenden Befehl:

```
nano SEN-KY003HMS.py
```

Kopieren Sie das nun folgende Codebeispiel in den Editor. Speichern Sie dann die Datei mit der Tastenkombination **STRG+O** und **Enter** und verlassen Sie die Datei mit der Kombination **STRG+X**.

```
import time
from gpiozero import DigitalInputDevice

delayTime = 1      # Zeit zwischen Messungen
Digital_PIN = 24   # Digital-Pin (BCM)
sensor = DigitalInputDevice(Digital_PIN)

while True:
    # Ausgabe
    if sensor.value == 0:
        print("Magnetic field detected!!")
    else:
        print("No magnetic field detected")
    print("-----")

    time.sleep(delayTime)
```

Starten Sie nun das Programm mit dem folgenden Befehl:

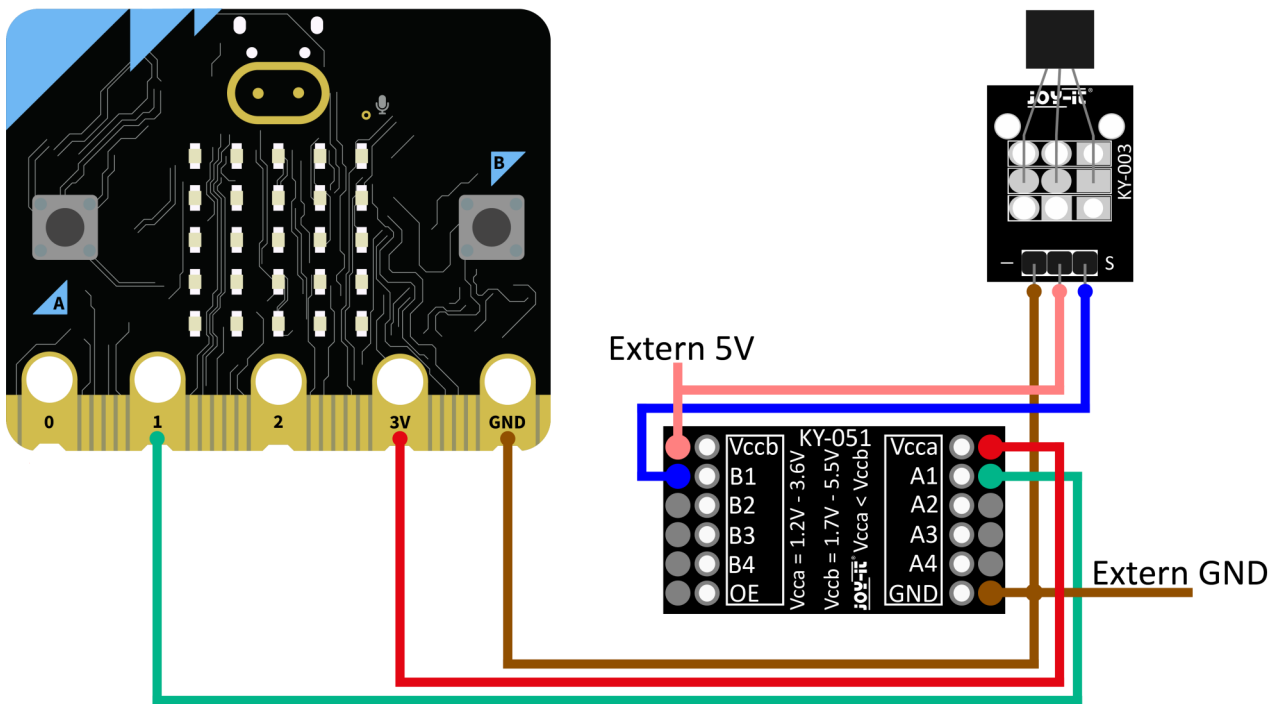
```
python3 SEN-KY003HMS.py
```

In der Konsole wird Ihnen nun angezeigt, ob ein magnetisches Feld erkannt wurde.

4. VERWENDUNG MIT DEM MICRO:BIT

Schließen Sie Ihren Sensor, wie im Schaubild und der Tabelle zusehen an einen Voltage Translator und Ihren Micro:Bit an. Hierbei empfehlen wir den [KY-051 Voltage Translator von Joy-IT](#). Für optimale Ergebnisse sollten Sie einen Pullup Widerstand (ca. 10 kΩ) zwischen der Signalleitung des Sensors und der 5V Leitung verwenden.

Für die Verwendung mit dem micro:bit wird außerdem ein zusätzliches 5V Netzteil benötigt.



KY-003	Voltage translator	Micro:bit	5 V power supply
GND	Extern GND	GND	GND
+	VCCb		5 V
Signal (S)	B1		
	VCCa	3 V	
	A1	Pin 1	

Codebeispiel

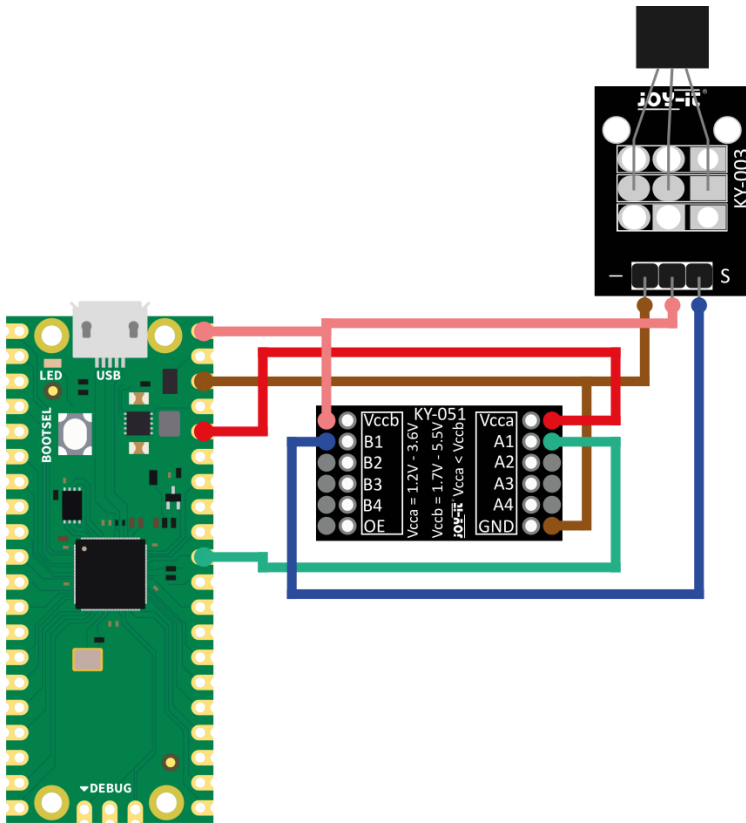
In dem nun folgenden Codebeispiel, wird jede Sekunde im seriellen Monitor angegeben, ob ein Magnetfeld erkannt wurde. Sie können das Codebeispiel entweder in Ihrem MakeCode-Sketch nachbauen, oder die [.hex-Datei](#) herunterladen, und als MakeCode-Projekt importieren.

```
BlocklyScratch({
  blocks: [
    {
      type: 'loop_forever',
      x: 100,
      y: 225,
      width: 160,
      height: 25,
      color: '#1E90FF',
      text: 'dauerhaft'
    },
    {
      type: 'say_serial_print',
      x: 120,
      y: 260,
      width: 500,
      height: 30,
      color: '#191970',
      text: 'seriell Zeile ausgeben',
      value: 'analoge Werte von Pin P1',
      pin: 'P1'
    },
    {
      type: 'if_then_else',
      x: 120,
      y: 295,
      width: 540,
      height: 40,
      color: '#00CED1',
      text: 'wenn',
      value: 'analoge Werte von Pin P1',
      pin: 'P1',
      threshold: '<',
      threshold_value: '40',
      text2: 'dann'
    },
    {
      type: 'say_serial_print',
      x: 140,
      y: 335,
      width: 360,
      height: 30,
      color: '#191970',
      text: 'seriell Zeile ausgeben',
      value: '"Magnetfeld"'
    },
    {
      type: 'if_then_else_else',
      x: 120,
      y: 370,
      width: 540,
      height: 40,
      color: '#00CED1',
      text: 'ansonsten',
      icon: 'minus'
    },
    {
      type: 'say_serial_print',
      x: 140,
      y: 410,
      width: 400,
      height: 30,
      color: '#191970',
      text: 'seriell Zeile ausgeben',
      value: '"Kein Magnetfeld"'
    },
    {
      type: 'say_serial_print',
      x: 120,
      y: 470,
      width: 580,
      height: 30,
      color: '#191970',
      text: 'seriell Zeile ausgeben',
      value: '""'
    },
    {
      type: 'say_pause',
      x: 120,
      y: 505,
      width: 220,
      height: 30,
      color: '#1E90FF',
      text: 'pausiere (ms)',
      value: '1000'
    }
  ]
})
```

The image shows a Scratch code sketch for a magnet field detection. It starts with a 'dauerhaft' (forever) loop. Inside the loop, the first block is 'seriell Zeile ausgeben' (serial print) with the value 'analoge Werte von Pin P1'. This is followed by an 'if-then-else' block. The 'if' condition is 'analoge Werte von Pin P1' with a '<' threshold of '40'. If true, it prints 'Magnetfeld'. If false, it prints 'Kein Magnetfeld'. After the if-then-else, there is an empty 'seriell Zeile ausgeben' block. Finally, the loop ends with a 'pausiere (ms)' block set to '1000'.

5. VERWENDUNG MIT DEM RASPBERRY PI PICO

Schließen Sie Ihren Sensor, wie im Schaubild und der Tabelle zusehen an einen Voltage Translator und Ihren Raspberry Pi Pico an. Hierbei empfehlen wir den [KY-051 Voltage Translator von Joy-IT](#). Für optimale Ergebnisse sollten Sie einen Pullup Widerstand (ca. 10 kΩ) zwischen der Signalleitung des Sensors und der 5V Leitung verwenden.



KY-003	Voltage translator	Raspberry Pi Pico
GND	GND	GND
+	VCCb	VBUS
Signal (S)	B1	
	VCCa	3V3
	A1	GP26

Codebeispiel

In dem nun folgenden Codebeispiel, wird jede 0,5 Sekunden in der Konsole angegeben, ob ein Magnetfeld erkannt wurde. Kopieren Sie den Beispielcode und laden Sie ihn auf Ihren Raspberry Pi. Alternativ können Sie den Beispielcode auch [hier](#) herunterladen.

```
# Bibliotheken laden

from machine import Pin, Timer
from time import sleep

# Initialisierung von GPIO als Eingang
sensor = Pin(26, Pin.IN, Pin.PULL_DOWN)

# Dauerschleife für kontinuierliche Serielle Ausgabe
while True:

    if sensor.value() == 0:

        print("Magnetfeld")

    else:

        print("Kein Magnetfeld")

    print("-----")

    sleep(0.5)
```

6. 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.

7. 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