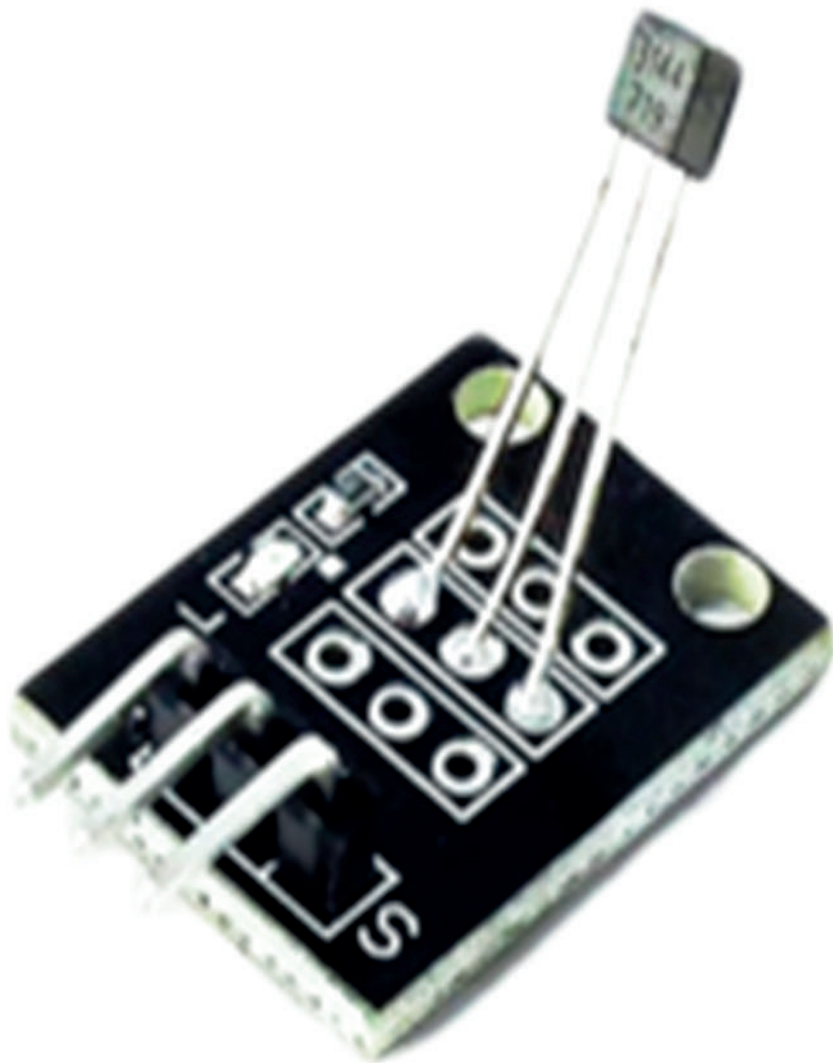


KY-035 Hall Sensor Magnetisch Modul (analog) Datenblatt



Contents:

- 1. Technical Data**
- 2. Pinout**
- 3. Code Example Arduino**
- 4. Code Example Raspberry Pi**

1. Technical Data

Chipset: AH49E

The sensor measures the current magnetic field, near to the sensor.

2. Pinout

The program measures the current voltage value at the sensor, calculates with it and a known resistor the resistance from the sensor and shows the results via serial output.

```
1  int sensorPin = A5; // Declaration of the input pin
2
3  // Serial OUT in 9600 baud
4  void setup()
5  {
6      Serial.begin(9600);
7  }
8
9  // The program measures the current voltage at the sensor,
10 // calculates the resistance with it and a known resistor
11 // and outputs it via serial OUT
12
13 void loop()
14 {
15     // Measuring of the current voltage...
16     int rawValue = analogRead(sensorPin);
17     float voltage = rawValue * (5.0/1023) * 1000;
18
19     float resitance = 10000 * ( voltage / ( 5000.0 - voltage) );
20
21     // ... output via serial interface
22     Serial.print("Voltage:");   Serial.print(voltage); Serial.print("mV");
23     Serial.print(", Resistance:"); Serial.print(resitance); Serial.println("Ohm");
24     Serial.println("-----");
25
26     delay(500);
27 }
```

Connections Arduino:

Sensor GND= [Pin GND]

Sensor +V = [Pin 5V]

Sensor Signal = [Pin A5]

4. Code Example Raspberry Pi

!! Attention !! Analog Sensor !! Attention !!

Unlike the Arduino, the Raspberry Pi doesn't provide an ADC (Analog Digital Converter) on its Chip. This limits the Raspberry Pi if you want to use a non digital Sensor.

This module is connected via I2C to the Raspberry Pi.

It measures the analog data and converts it into a digital signal which is suitable for the Raspberry Pi.

So we recommend to use the KY-053 ADC if you want to use analog sensors along with the Raspberry Pi.

!! Attention !! Analog Sensor !! Attention !!

The program uses the specific ADS1x15 and I2C python-libraries from the company Adafruit to control the ADS1115 ADC.

The program reads the current values of the input pins and outputs it at the terminal in [mV].

Additional to that, the status of the digital pin will be shown at the terminal to show if the extreme value was exceeded or not.

```

1 #####
2 ### Copyright by Joy-IT
3 ### Published under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License
4 ### Commercial use only after permission is requested and granted
5 ###
6 ### KY-053 Analog Digital Converter - Raspberry Pi Python Code Example
7 ###
8 #####
9
10
11 # This code is using the ADS1115 and the I2C Python Library for Raspberry Pi
12 # This was published on the following link under the BSD license
13 # [https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code]
14 from Adafruit_ADS1x15 import ADS1x15
15 from time import sleep
16
17 # import needed modules
18 import time, signal, sys, os
19 import RPi.GPIO as GPIO
20 GPIO.setmode(GPIO.BCM)
21 GPIO.setwarnings(False)
22
23 # initialise variables
24 delayTime = 0.5 # in Sekunden
25
26 # assigning the ADS1x15 ADC
27
28 ADS1015 = 0x00 # 12-bit ADC
29 ADS1115 = 0x01 # 16-bit
30
31 # choosing the amplifying gain
32 gain = 4096 # +/- 4.096V
33 # gain = 2048 # +/- 2.048V
34 # gain = 1024 # +/- 1.024V
35 # gain = 512 # +/- 0.512V
36 # gain = 256 # +/- 0.256V
37
38 # choosing the sampling rate
39 # sps = 8 # 8 Samples per second
40 # sps = 16 # 16 Samples per second
41 # sps = 32 # 32 Samples per second
42 sps = 64 # 64 Samples per second
43 # sps = 128 # 128 Samples per second
44 # sps = 250 # 250 Samples per second
45 # sps = 475 # 475 Samples per second
46 # sps = 860 # 860 Samples per second
47
48 # assigning the ADC-Channel (1-4)
49 adc_channel_0 = 0 # Channel 0
50 adc_channel_1 = 1 # Channel 1
51 adc_channel_2 = 2 # Channel 2
52 adc_channel_3 = 3 # Channel 3
53
54 # initialise ADC (ADS1115)
55 adc = ADS1x15(ic=ADS1115)
56
57 #####
58
59 # #####
60 # Main Loop
61 # #####
62 # Reading the values from the input pins and print to console
63
64 try:
65     while True:
66         #read values
67         adc0 = adc.readADCSingleEnded(adc_channel_0, gain, sps)
68         adc1 = adc.readADCSingleEnded(adc_channel_1, gain, sps)
69         adc2 = adc.readADCSingleEnded(adc_channel_2, gain, sps)
70         adc3 = adc.readADCSingleEnded(adc_channel_3, gain, sps)
71
72         # print to console
73         print "Channel 0:", adc0, "mV "
74         print "Channel 1:", adc1, "mV "
75         print "Channel 2:", adc2, "mV "
76         print "Channel 3:", adc3, "mV "
77         print "-----"
78
79         time.sleep(delayTime)
80
81
82 except KeyboardInterrupt:
83     GPIO.cleanup()
84

```

Connections Raspberry Pi:

Sensor

GND= GND[Pin 06 (RPI)]

+V = 3,3V [Pin 01 (RPI)]

analog Signal = Analog 0 [Pin A0 (ADS1115 - KY-053)]

ADS1115 - KY-053:

VDD= 3,3V [Pin 17]

GND= GND[Pin 09]

SCL = GPIO03 / SCL [Pin 05]

SDA= GPIO02 / SDA [Pin 03]

A0= look above [Sensor: analog Signal]

To start, enter the command:

```
1 | sudo python KY-053_RPi-AnalogDigitalConverter.py
```