To start, enter the command:

```
KY-024_Linear-magnetic-sensor_RPi
```

Additional to that, the status of the digital pin will be shown at the terminal to show if the extreme value was reached.

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

For more information please look at the infosite:

https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code

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

Connections Arduino:

- An analog sensor could be a potentiometer (e.g. KY-037, KY-038) or rotary encoder (KY-026).

Functionality of the sensor:

Depending on the magnetic field, the sensor detects a signal, which is then printed at the terminal.

Pinout:

- LED1:
  - Shows that the sensor detects a magnetic field
- LED2:
  - Shows that the sensor detects a signal
- Analog out:
  - Measures the area physically and sends an analog signal to the second unit, the amplifier. The amplifier amplifies the value.

Technical data / Short description:

- Chipset: A3141 | OP-amplifier: LM393
- Technical data / Short description

Value:

- The program reads the current value of the input pins.
- If a magnetic field is detected by the sensor, a signal will be printed here.
- The signal will be inverted; that means that if you measure a high value, it is shown as a low value.
- Current values will be recorded.

```python
# This was published on the following link under the BSD license
# This code is using the ADS1115 and the I2C Python Library for Raspberry Pi
# Commercial use only after permission is requested and granted
# Published under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

import math, signal, sys, os

# Digital pins
Digital_Eingang = 9
Digital_PIN = 2

# ADC pins
adc_channel_0 = 0
adc_channel_1 = 3

# assigning the ADC-Channel (1-4)
adc_channel_0
adc_channel_1

# choosing the sampling rate
sps
sps = 250  # 250 Samples per second
sps = 8    # 8 Samples per second

# choosing the amplifying gain
ADS1115
ADS1015

# assigning the ADS1x15 ADC
delayTime

gain = 2

# The program reads the current value of the input pins
# This was published on the following link under the BSD license
# This code is using the ADS1115 and the I2C Python Library for Raspberry Pi
# Commercial use only after permission is requested and granted
# Published under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

#Digital_Eingang
Digital = digitalRead (Digital_Eingang);
Digital = digitalRead (Digital_PIN);

# ADC read
analog = adc.readADCSingleEnded(adc_channel_0, gain, sps);

# Digital state
Digital = digitalRead (Digital_Eingang);

# Serial output
Serial.println ("Extreme value: ");
Serial.println (analog);

# Analog state
analog = adc.readADCSingleEnded(adc_channel_0, gain, sps);

# Serial output
Serial.println ("Analog voltage value: ");
Serial.println (analog, 4);
```

```
// Button

void setup() {
  pinMode (Digital, INPUT);
  pinMode (Digital_PIN, INPUT);
  pinMode (adc_channel_0, INPUT);
  pinMode (adc_channel_1, INPUT);

  // Button
  pinMode (digitalRead (Digital_Eingang), OUTPUT);
  pinMode (digitalRead (Digital_PIN), OUTPUT);

  // Serial output
  Serial.begin (9600);

  // Button
  pinMode (digitalRead (Digital_Eingang), OUTPUT);
  pinMode (digitalRead (Digital_PIN), OUTPUT);

  // Serial output
  Serial.println ("Extreme value: ");
  Serial.println (analog);

  // Analog state
  analog = adc.readADCSingleEnded(adc_channel_0, gain, sps);

  // Serial output
  Serial.println ("Analog voltage value: ");
  Serial.println (analog, 4);
}
```