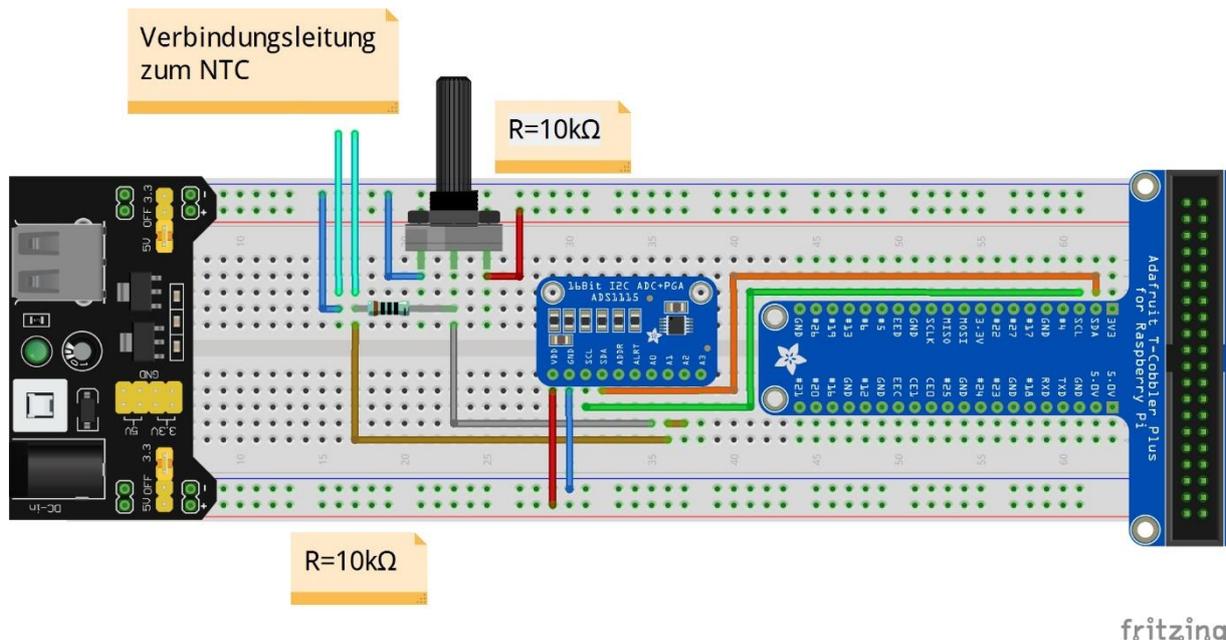
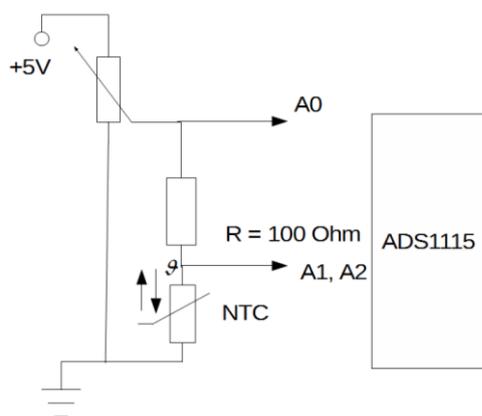


Kennlinie eines temperaturabhängigen Widerstands NTC

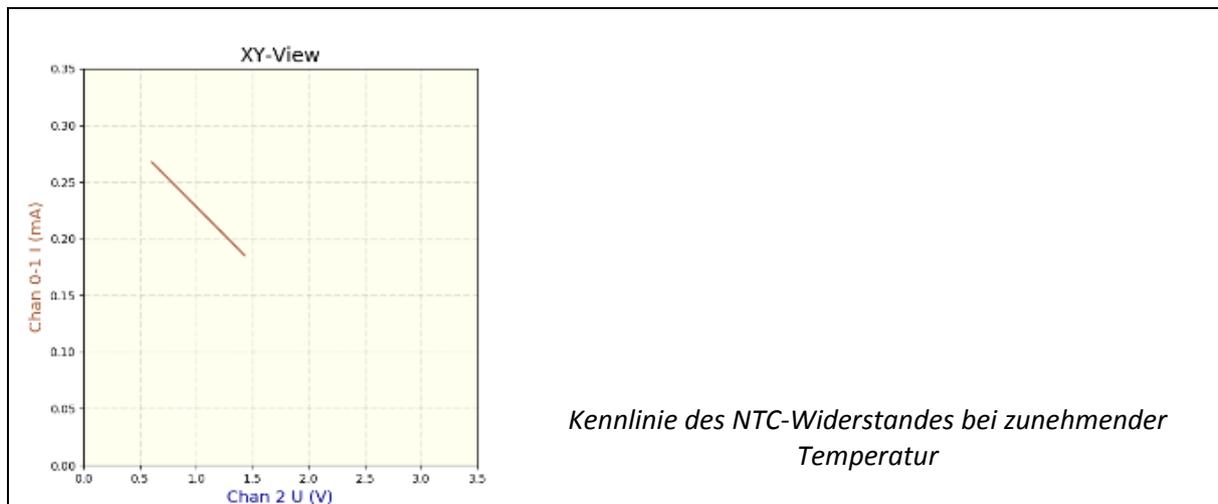
Der **NTC-Widerstand** ist ein temperaturabhängiger Widerstand mit einem negativen Temperaturkoeffizienten, d.h. er leitet den elektrischen Strom bei höheren Temperaturen besser als bei niedrigeren Temperaturen. Diese Eigenschaft lässt sich erklären, indem die Anzahl der freibeweglichen Ladungsträger mit zunehmender Temperatur größer wird. Das thermische Anwachsen des Leitungswiderstandes wegen der Wechselwirkung der Ladungsträger mit den Bausteinen der Kristallstruktur hat eine kleinere Wirkung auf das Verhalten des NTC-Widerstandes.

Aufbau und Durchführung

Die Schüler stecken die abgebildete Schaltung gemäß der Abbildung des Schaltungsaufbaus auf dem Breadboard und schließen Spannungsversorgung und Netzteil an. Dabei wird einen einfachen Spannungsteiler verwendet. Allgemein lässt sich für den Spannungsteiler festhalten, dass sich die Spannungen wie die zugehörigen Widerstände verhalten. Der Analog-Digital-Wandler ADS1115 wandelt ein analoges in ein digitales Signal. Das heißt er wandelt ein kontinuierliches Signal mit unendlich vielen Signalwerten in eine Folge von Wertepaaren aus einem Zeitwert und einem Signalwert um.



Mit dieser Schaltung lässt sich die Spannung U_1 am Widerstand $R = 100\Omega$, sowie die Spannung U_2 am NTC-Widerstand auslesen. Mit der Software PhyPiDAQ kann man die Kennlinie $U_2(I_2)$ visualisieren, indem man $I_2 = U_2/R$ im Programm eingibt.



ADS1115Config.yaml für NTC

```
# example of a configuration file for ADC ADS1115

DAQModule: ADS1115Config

ADCChannels: [2, 0]           # active ADC-Channels
                               # possible values: 0, 1, 2, 3
                               # when using differential mode:
                               # - 0 = ADCChannel 0
                               #   minus ADCChannel 1
                               # - 1 = ADCChannel 0
                               #   minus ADCChannel 3
                               # - 2 = ADCChannel 1
                               #   minus ADCChannel 3
                               # - 3 = ADCChannel 2
                               #   minus ADCChannel 3

DifModeChan: [false, true]    # enable differential mode for Channels
Gain: [1, 1]                  # programmable gain of ADC-Channel
                               # possible values for Gain:
                               # - 2/3 = +/-6.144V
                               # - 1 = +/-4.096V
                               # - 2 = +/-2.048V
                               # - 4 = +/-1.024V
                               # - 8 = +/-0.512V
                               # - 16 = +/-0.256V

sampleRate: 860               # programmable Sample Rate of ADS1115
                               # possible values for SampleRate:
                               # 8, 16, 32, 64, 128, 250, 475, 860
```

Configuration Options für phypi.daq für NTC

```

# Configuration Options for PhyPiDAQ

# device configuration files
DeviceFile: config/ADS1115Config.yaml
#DeviceFile: config/MCP3008Config.yaml
#DeviceFile: config/PSConfig.yaml
#DeviceFile: config/MAX31865Config.yaml
#DeviceFile: config/GPIOCOUNT.yaml
#DeviceFile: config/DS18B20Config.yaml
#DeviceFile: config/MAX31855Config.yaml

## an example of multiple devices
#DeviceFile: [config/ADS1115Config.yaml, config/GPIOCOUNT.yaml]

DisplayModule: DataLogger
# DisplayModule: DataGraphs # text, bar-graph, history and xy-view
Interval: 0.1 # logging interval
XYmode: true # enable/disable XY-display

# channel-specific information
ChanLabels: [U, I] # names for channels
ChanUnits: [V, mA] # units for channels
ChanColors: [darkblue, sienna] # channel colours in display

# eventually overwrite Channel Limits obtained from device config
ChanLimits:
- [0.0, 2.5] # chan 0
- [0., 0.2] # chan 1
## - [0., 1.] # chan 2

# calibration of channel values
# - null or - <factor> or - [ [ <true values> ], [ <raw
values> ] ]
#ChanCalib:
# - 1. # chan0: simple calibration factor
# - [ [0.,1.], [0., 1.] ] # chan1: interpolation: [true]([<raw>]
)
# - null # chan2: no calibration

# apply formulae to calibrated channel values
ChanFormula:
- c0 # chan0
- c1 * 10 # chan1
# - null # chan2 : no formula

# name of output file
DataFile: null # file name for output file
DataFile: Kennlnie_LED.csv # file name for output file
#CSVseparator: ';'

```