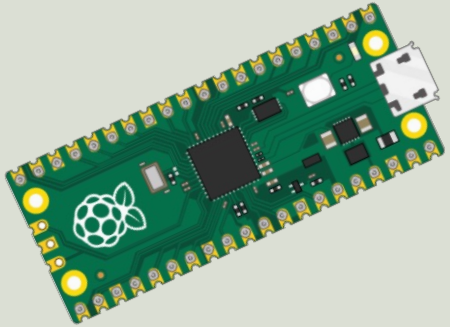
https://www.halvorsen.blog



Raspberry Pi Pico

Thermistor Temperature Sensor

Hans-Petter Halvorsen

Contents

Introduction

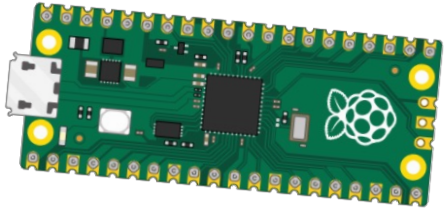
Hans-Petter Halvorsen

Table of Contents

Introduction

- In this Tutorial we will show how we can use a **Thermistor** with **Raspberry Pi Pico**
- We will use **MicroPython** and the **Thonny** Python Editor
- A Thermistor Temperature Sensor is a small and cheap temperature sensor
- We will use a **10K NTC Thermistor** in this tutorial

What do you need?



- Raspberry Pi Pico
- A Micro-USB cable
- A PC with Thonny Python Editor (or another Python Editor)
- Breadboard
- Electronics Components like LED, Resistors, Jumper wires, etc.
- Thermistor 10K

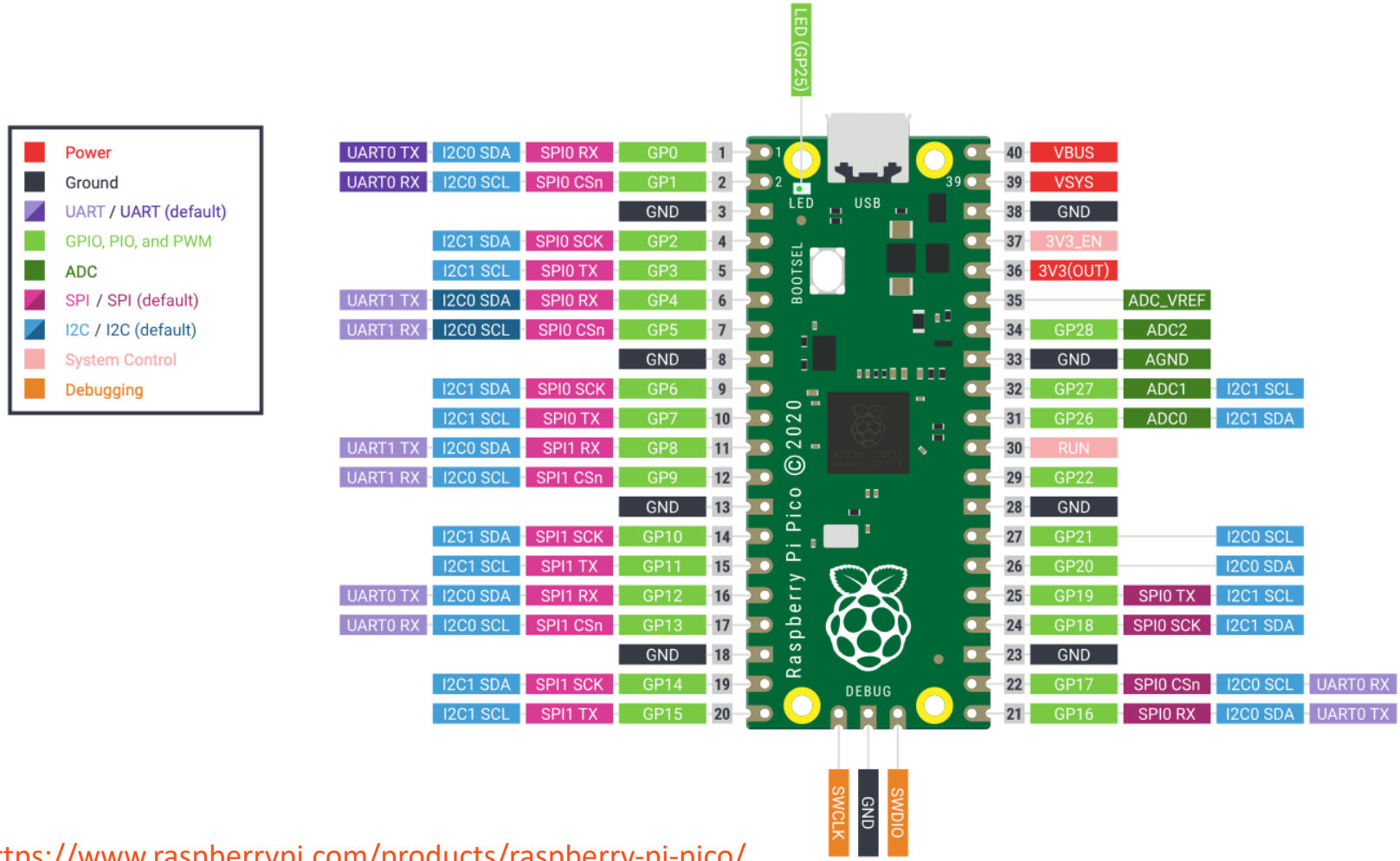
Raspberry Pi Pico

- Raspberry Pi Pico is a microcontroller board developed by the Raspberry Pi Foundation
- Raspberry Pi Pico has similar features as Arduino devices
- Raspberry Pi Pico is typically used for Electronics projects, IoT Applications, etc.
- You typically use MicroPython, which is a downscaled version of Python, in order to program it

https://www.raspberrypi.com/products/raspberry-pi-pico/

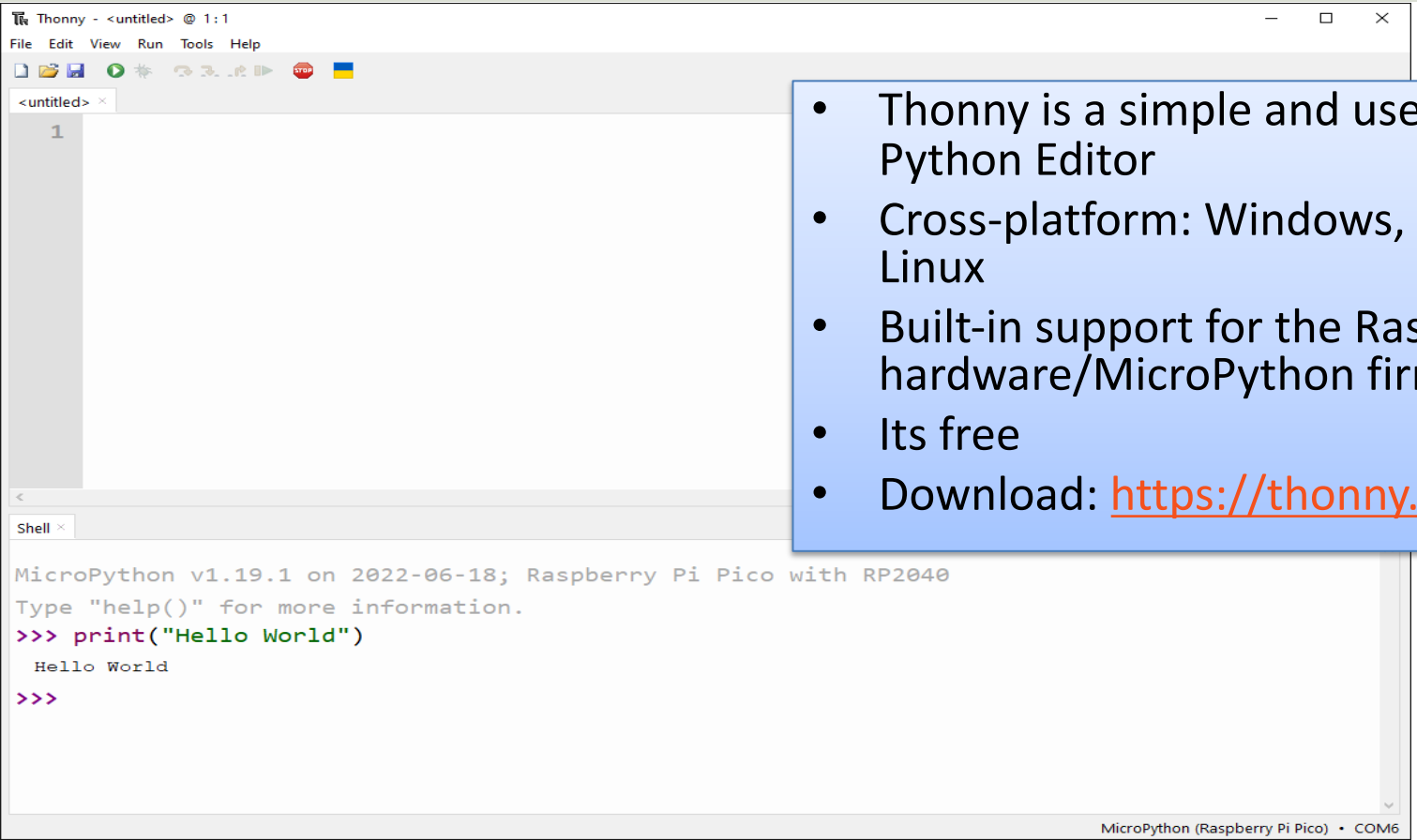
https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico

Pico Pinout



https://www.raspberrypi.com/products/raspberry-pi-pico/

Thonny



- Thonny is a simple and user-friendly Python Editor
- Cross-platform: Windows, macOS and Linux
- Built-in support for the Raspberry Pi Pico hardware/MicroPython firmware
- Its free
- Download: https://thonny.org

MicroPython

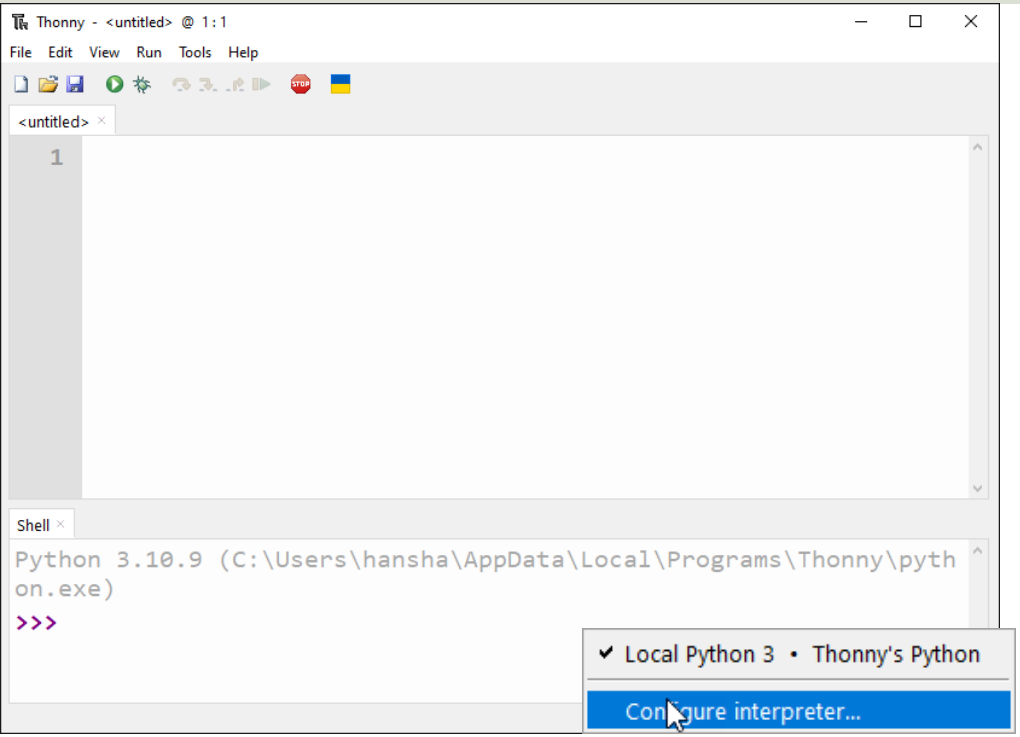
- MicroPython is a downscaled version of Python
- It is typically used for Microcontrollers and constrained systems (low memory, etc.)
- Examples of such Microcontrollers that have tailormade MicroPython firmwares are Raspberry Pi Pico and Micro:bit

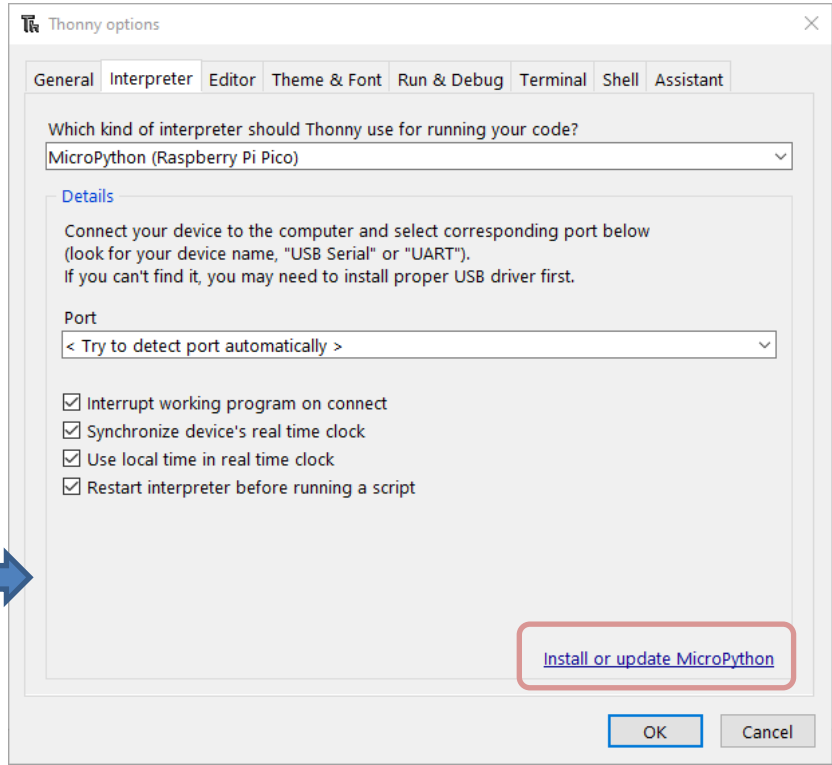
https://docs.micropython.org/en/latest/index.html https://micropython.org

MicroPython Firmware

- The first time you need to install the MicroPython Firmware on your Raspberry Pi Pico
- You can install the MicroPython Firmware manually or you can use the Thonny Editor

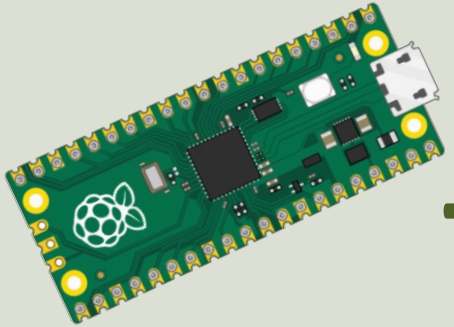
Install MicroPython Firmware using Thonny





https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico/3

https://www.halvorsen.blog



Thermistor Temperature Sensor

Raspberry Pi Pico

Hans-Petter Halvorsen

Table of Contents

10K Thermistor

Thermistor

A thermistor is an electronic component that changes resistance to temperature - so-called Resistance Temperature Detectors (RTD). It is often used as a temperature sensor.

Our Thermistor is a so-called NTC (Negative Temperature Coefficient). In a NTC Thermistor, resistance decreases as the temperature rises.

There is a **non-linear relationship** between resistance and excitement. To find the temperature we can use the following equation (**Steinhart-Hart equation**):

$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3$$

[Wikipedia]

where A, B, C are constants given below

$$A = 0.001129148, B = 0.000234125 \text{ and } C = 8.76741E-08$$

Steinhart-Hart equation

$$\frac{1}{T_K} = A + B\ln(R) + C(\ln(R))^3$$

$$T_K = \frac{1}{A + B\,ln(R_t) + C(ln(R_t))^3}$$

Steinhart-Hart Equation

To find the Temperature we can use Steinhart-Hart Equation:

$$\frac{1}{T_K} = A + B\ln(R) + C(\ln(R))^3$$

This gives:

$$T_K = \frac{1}{A + B\ln(R) + C(\ln(R))^3}$$

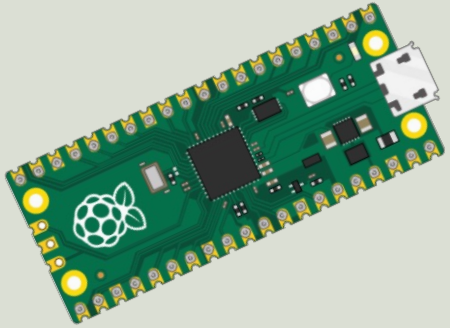
Where the Temperature T_K is in **Kelvin**
$A, B\ and\ C$ are constants

$$A = 0.001129148$$
$$B = 0.000234125$$
$$C = 0.0000000876741$$

The Temperature in degrees **Celsius** will then be:

$$T_C = T_K - 273.15$$

https://www.halvorsen.blog



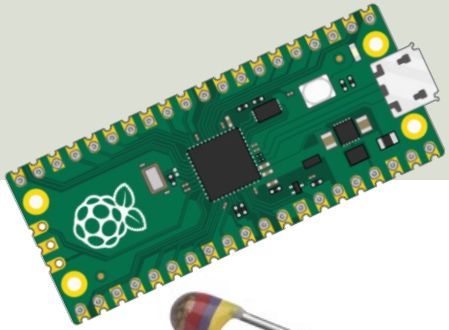
Wiring and Voltage Divider

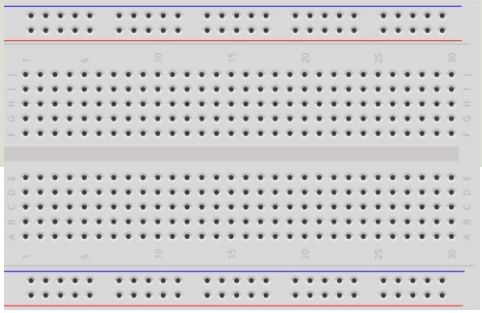
Raspberry Pi Pico

Hans-Petter Halvorsen

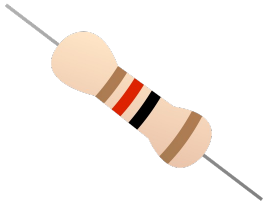
Table of Contents

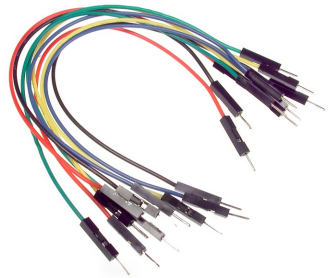
Hardware





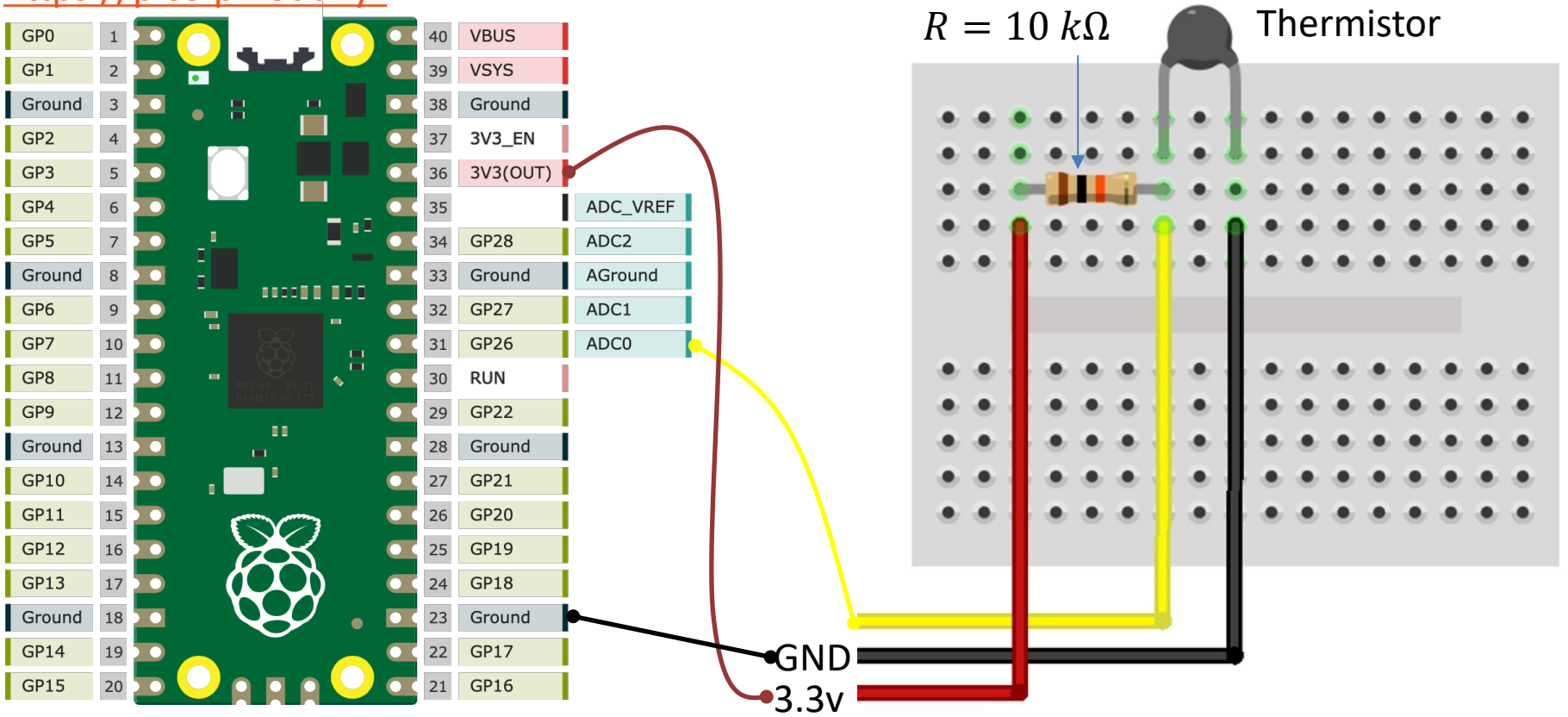
- Raspberry Pi Pico
- Breadboard
- Thermistor 10K (Temperature Sensor)
- Wires (Jumper Wires)
- Resistor $R = 10\ k\Omega$





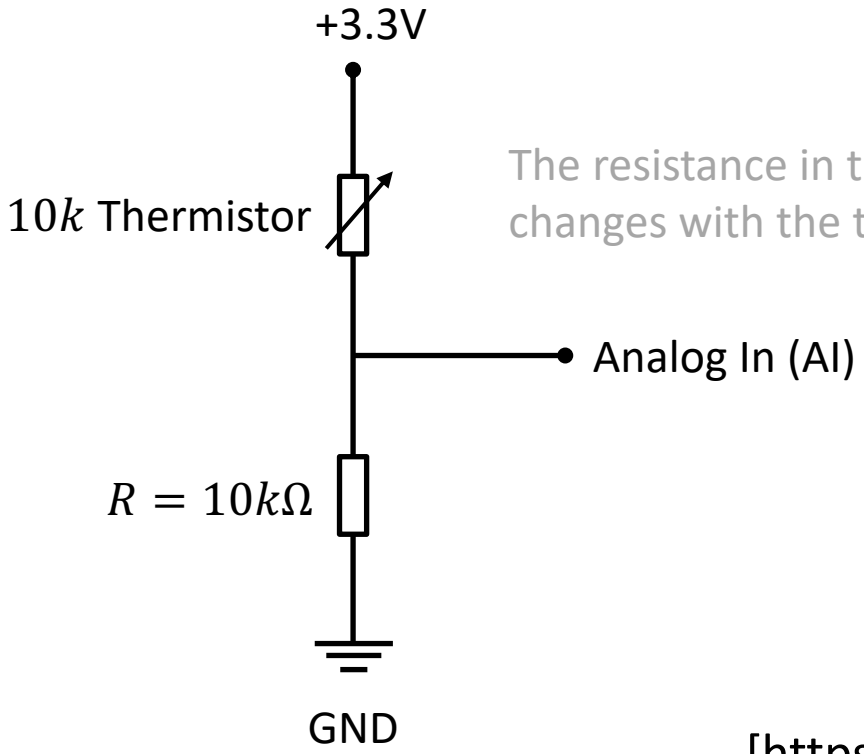
Thermistor Wiring

https://pico.pinout.xyz



Thermistor Voltage Divider Wiring

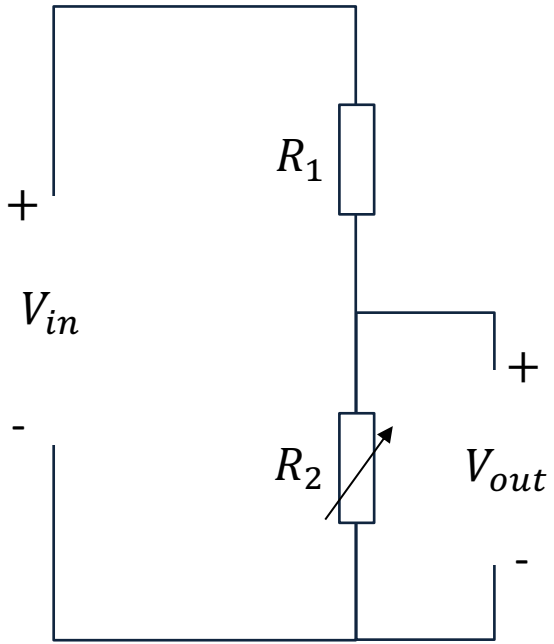
The wiring is called a “Voltage divider”:



The resistance in the Thermistor changes with the temperature

[https://en.wikipedia.org/wiki/Voltage_divider]

General Voltage Divider



We want to find V_{out}

Formula:

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2}$$

The Resistor R_1 has a specific value, while R_2 is a variable resistor

https://learn.sparkfun.com/tutorials/voltage-dividers/all

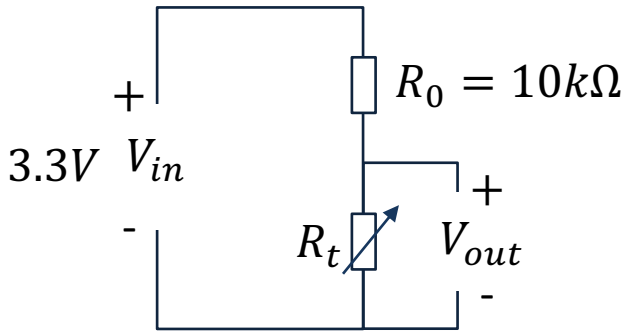
Voltage Divider for our System

Voltage Divider Equation:

$$V_{out} = V_{in}\frac{R_t}{R_0 + R_t}$$

We want to find R_t:

$$R_t = \frac{V_{out}R_0}{V_{in} - V_{out}}$$

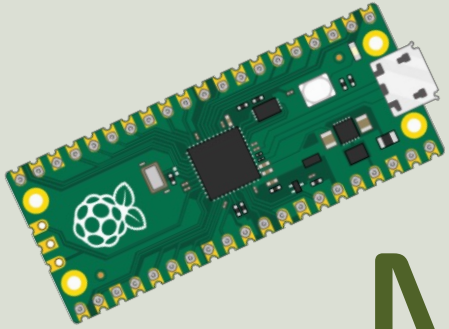


R_t - 10k Thermistor. This varies with temperature. From Datasheet we know that $R_t = 10k\Omega$ @25°C

Steps:

1. We wire the circuit on the Breadboard and connect it to the Raspberry Pi Pico device
2. We measure V_{out} using the Raspberry Pi Pico device
3. We calculate R_t using the Voltage Divider equation
4. Finally, we use Steinhart-Hart equation for finding the Temperature

https://www.halvorsen.blog



Raspberry Pi Pico

MicroPython Examples

Thermistor Temperature Sensor

Hans-Petter Halvorsen

Table of Contents

Pseudo Code

4 main steps:

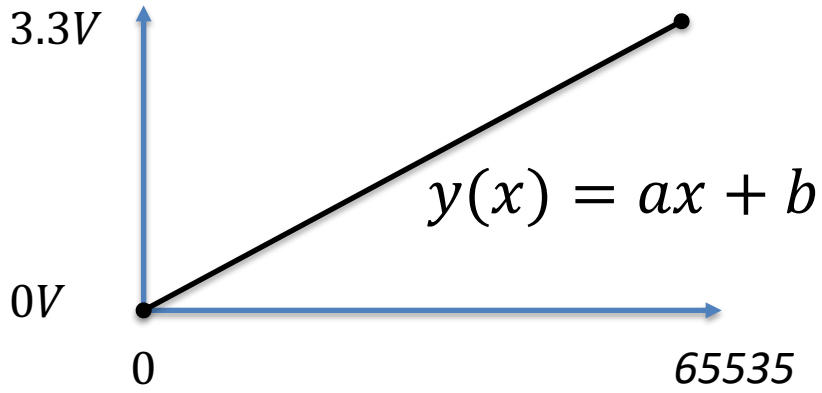
1. Get V_{out} from the ADC on Raspberry Pi Pico
2. Calculate $R_t = \frac{V_{out} R_0}{V_{in} - V_{out}}$
3. Calculate $T_K = \frac{1}{A + B\ ln(R_t) + C(ln(R_t))^3}$
4. Calculate $T_C = T_K - 273.15$

ADC Value to Voltage Value

Analog Pins: The built-in Analog-to-Digital Converter (ADC) on Pico is 16bit, producing values from 0 to 65535.

The `read_u16()` function gives a value between 0 and 65535. It must be converted to a Voltage Signal 0 - 3.3v

ADC = 0 -> 0v
ADC = 65535 -> 3.3v



This gives the following conversion formula:

$$y(x) = \frac{3.3}{65535}x$$

Pseudo Code

```
//Get Voltage
adc = thermistor.read_u16()
Vout = (3.3/65535)*adc

//Voltage Divider. Calculate R
float Vin = 3.3;
float Ro=10000;
float Rt = (Vout*Ro)/(Vin-Vout);

//Steinhart constants
float A = 0.001129148;
float B = 0.000234125;
float C = 0.0000000876741;

//Steinhart-Hart Equation
float TempK = 1 / (A + (B * ln(Rt)) + (C * ln(Rt)**3));

//Convert from Kelvin to Celsius
float TempC = TempK - 273.15;
```

Python

The Code works as follows:

1. Get V_{out} from the DAQ device
2. Calculate $R_t = \frac{V_{out}R_0}{V_{in}-V_{out}}$
3. Calculate $T_K = \frac{1}{A+B\ ln(R_t)+C(ln(R_t))^3}$
4. Calculate $T_C = T_K - 273.15$
5. Present T_C in the User Interface

```
from machine import ADC
from time import sleep
import math

adcpin = 26
thermistor = ADC(adcpin)

# Voltage Divider
Vin = 3.3
Ro = 10000  # 10k Resistor

# Steinhart Constants
A = 0.001129148
B = 0.000234125
C = 0.0000000876741

while True:
    # Get Voltage value from ADC
    adc = thermistor.read_u16()
    Vout = (3.3/65535)*adc

    # Calculate Resistance
    Rt = (Vout * Ro) / (Vin - Vout)
    # Rt = 10000  # Used for Testing. Setting Rt=10k should give TempC=25

    # Steinhart - Hart Equation
    TempK = 1 / (A + (B * math.log(Rt)) + C * math.pow(math.log(Rt), 3))

    # Convert from Kelvin to Celsius
    TempC = TempK - 273.15

    print(round(TempC, 1))
    sleep(5)
```

Thonny - C:\Users\hansha\OneDrive\Documents\Industrial IT and Automation\IoT\Raspberry Pi Pico\Code Examples\thermistor_ex.py @ 33 : 13

File Edit View Run Tools Help

thermistor_ex.py

```
from machine import ADC
from time import sleep
import math

adcpin = 26
thermistor = ADC(adcpin)

# Voltage Divider
Vin = 3.3
Ro = 10000  # 10k Resistor

# Steinhart Constants
A = 0.001129148
B = 0.000234125
C = 0.0000000876741

while True:
    adc = thermistor.read_u16()
    Vout = (3.3/65535)*adc

    # Calculate Resistance
    Rt = (Vout * Ro) / (Vin - Vout)
    #Rt = 10000  # Used for Testing. Setting Rt=10k should give TempC=25

    # Steinhart - Hart Equation
    TempK = 1 / (A + (B * math.log(Rt)) + C * math.pow(math.log(Rt), 3))

    # Convert from Kelvin to Celsius
    TempC = TempK - 273.15

    print(round(TempC, 1))

    sleep(5)
```

Shell

```
25.5
25.5
```

MicroPython (Raspberry Pi Pico) • COM6

Python v2

Here, we have made a separate Python function for the Thermistor logic. This makes it easy to use this part in several Applications.

```
import math

def thermistorTemp(Vout):

    # Voltage Divider
    Vin = 3.3
    Ro = 10000  # 10k Resistor

    # Steinhart Constants
    A = 0.001129148
    B = 0.000234125
    C = 0.0000000876741

    # Calculate Resistance
    Rt = (Vout * Ro) / (Vin - Vout)

    # Steinhart - Hart Equation
    TempK = 1 / (A + (B * math.log(Rt)) + C * math.pow(math.log(Rt), 3))

    # Convert from Kelvin to Celsius
    TempC = TempK - 273.15

    return TempC
```

thermistor.py

Thermistor Application:

```
from machine import ADC
from time import sleep
import thermistor

adcpin = 26
sensor = ADC(adcpin)

while True:
    adc = sensor.read_u16()
    Vout = (3.3/65535)*adc

    TempC = thermistor.thermistorTemp(Vout)

    print(round(TempC, 1))

    sleep(5)
```

Thonny - Raspberry Pi Pico :: /thermistor_ex2.py @ 9 : 17

File Edit View Run Tools Help

Files

This computer

- ESD
- GitHub
- inetpub
- Intel
- PanoptoRecorder
- PerfLogs
- Program Files
- Program Files (x86)
- Recovery
- Software
- Software Development
- SymCache
- Temp
  - Arduino
  - LabVIEW OPC UA
  - LabVIEW Snake
  - OPCUA
  - OPCUANET
  - Pico
    - thermistor.py
    - thermistor_ex2.py

Raspberry Pi Pico

- lib
- thermistor.py
- thermistor_ex2.py

[thermistor_ex2.py] [thermistor.py]

```
1  from machine import ADC
2  from time import sleep
3  import thermistor
4
5  adcpin = 26
6  sensor = ADC(adcpin)
7
8  while True:
9      adc = sensor.read_u16()
10     Vout = (3.3/65535)*adc
11
12     TempC = thermistor.thermistorTemp(Vout)
13
14     print(round(TempC, 1))
15
16     sleep(5)
```

Shell

```
25.4
25.5
25.5
25.3
25.5
25.4
25.3
25.0
24.9
25.2
24.9
25.0
```

In order to make this work the “thermistor.py” needs to be placed on the Pico.
You can use the “Files” tool in the Thonny Editor

MicroPython (Raspberry Pi Pico) • COM6

Python Code v3

TemperatureSensors.py

```
class Thermistor:
    def __init__(self, pin):
        self.thermistor = ADC(pin)

    def ReadTemperature(self):
        # Get Voltage value from ADC
        adc_value = self.thermistor.read_u16()
        Vout = (3.3/65535)*adc_value

        # Voltage Divider
        Vin = 3.3
        Ro = 10000  # 10k Resistor

        # Steinhart Constants
        A = 0.001129148
        B = 0.000234125
        C = 0.0000000876741

        # Calculate Resistance
        Rt = (Vout * Ro) / (Vin - Vout)

        # Steinhart - Hart Equation
        TempK = 1 / (A + (B * math.log(Rt)) + C * math.pow(math.log(Rt), 3))

        # Convert from Kelvin to Celsius
        TempC = TempK - 273.15

        return round(TempC,2)
```

Here, we have taken it one step further by making a separate Python Class and Python Module for the Thermistor logic. This makes it easy to use this part in several Applications and Code structure is improved

Python Code v3

Main Application:

```
from TemperatureSensors import Thermistor
from time import sleep

adcpin = 26
thermistor = Thermistor(adcpin)

while True:
    TempC = thermistor.ReadTemperature()
    print(TempC)
    sleep(5)
```

Raspberry Pi Pico Resources

- Raspberry Pi Pico:
  https://www.raspberrypi.com/products/raspberry-pi-pico/
- Raspberry Pi Foundation:
  https://projects.raspberrypi.org/en/projects?hardware[]=pico
- Getting Started with Pico:
  https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico
- MicroPython:
  https://docs.micropython.org/en/latest/index.html

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog