

Interpolation

Problem

The program has to read the temperature from the temperature sensor accurately. These temperature sensors are basically a resistor, so the microcontroller can only read their Voltage. This Voltage needs to be converted to degrees in °C. To do this, you measure the voltage of the sensor while measuring the temperature with another sensor. Doing this yields a table like this:

Voltage Temperature

1 V	10 °C
5 V	20 °C
20 V	80 °C
25 V	100 °C

This isn't real data, and I use it just as an example.

Now the microcontroller has some data, but it still needs to run some kind of calculation to get the temperature for Voltages that aren't in the list, like 10 V.

Solution

This can be solved by using a process called (linear) **Interpolation**.

The formula for interpolation goes like this:

$$\begin{array}{rcl} \frac{x-x_1}{x_2-x_1} & = & \frac{y-y_1}{y_2-y_1} \\ \frac{x-x_1}{x_2-x_1} \times (y_2 - y_1) & = & y - y_1 \\ \frac{x-x_1}{x_2-x_1} \times (y_2 - y_1) + y_1 & = & y \end{array} \left| \begin{array}{l} \times (y_2 - y_1) \\ + y_1 \end{array} \right.$$

Walkthrough

Let me show the process to explain it further. I am going to use the data above and try to find the Temperature for a Voltage of 10 V:

1. Find the entries in the data above and below the Voltage

In this case, this is

Voltage Temperature

5 V	20 °C
20 V	80 °C

Programmatically, this is done with the `find_location(number)` Function.

2. Calculate the Temperature using the interpolation formula

The Voltage is x , white the Temperature is y . So this is the data we have:

Voltage	Temperature
5 V ($= x_1$)	20 °C ($= y_1$)
10 V ($= x$)	What we want to calculate ($= y$)
20 V ($= x_2$)	80 °C ($= y_2$)

And now, all that's left to do is to insert this into the formula:

$$y = \frac{x-x_1}{x_2-x_1} \times (y_2 - y_1) + y_1$$

$$y = \frac{10V-5V}{20V-5V} \times (80^\circ C - 20^\circ C) + 20^\circ C$$

$$y = 40^\circ C$$

In python, this looks like this:

```
x1 = 5
x = 10
x2 = 20

y1 = 20
y2 = 80

y = (x - x1) / (x2 - x1) * (y2 - y1) + y1

print(y)
```

Output:

40.0

Implementation

All of this is implemented in the function `get_temperature(voltage: float)`. You can simply pass in a voltage and it returns the interpolated temperature. It internally uses the function `find_location(number: float)` and depends on the array of arrays `temperatureConversionCurve` to get the data.

Here is a quick example:

```
import interpolation

temperatureConversionCurve = [
    # [Voltage in V], [Temperature in °C]
    [1, 10],
    [5, 20],
    [20, 80],
    [25, 100]
]

temp = interpolation.get_temperature(10, temperatureConversionCurve)

print(temp)
```

Output:

40.0

again, this is a **example** and you need to measure your own `temperatureConversionCurve` for your sensor with a multimeter!