

<https://www.halvorsen.blog>



# Thermistor Temperature Sensor in LabVIEW



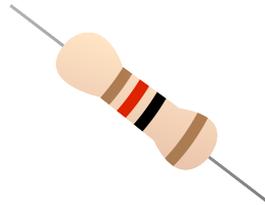
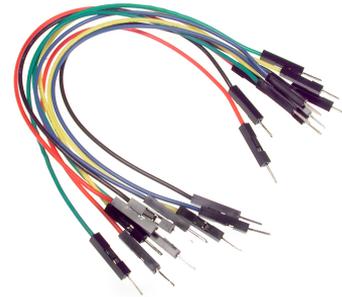
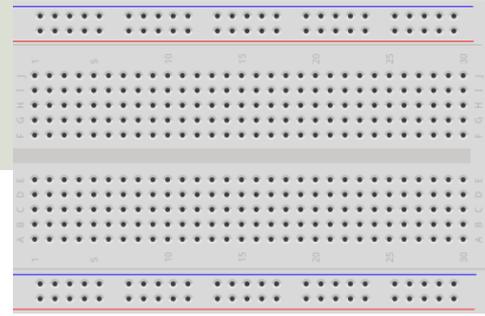
Hans-Petter Halvorsen

# Contents

- We will use LabVIEW to read Temperature data from a Thermistor Temperature Sensor
- We will use a USB-6008 DAQ Device or I/O Module

# Hardware

- DAQ Device (e.g., USB-6008)
- Breadboard
- Thermistor 10K (Temperature Sensor)
- Wires (Jumper Wires)
- Resistor 10 k $\Omega$



# Software

- LabVIEW
  - Graphical Programming Environment
- DAQmx Driver
  - Driver used for Communication with external Hardware such as USB-6008

# USB-6008

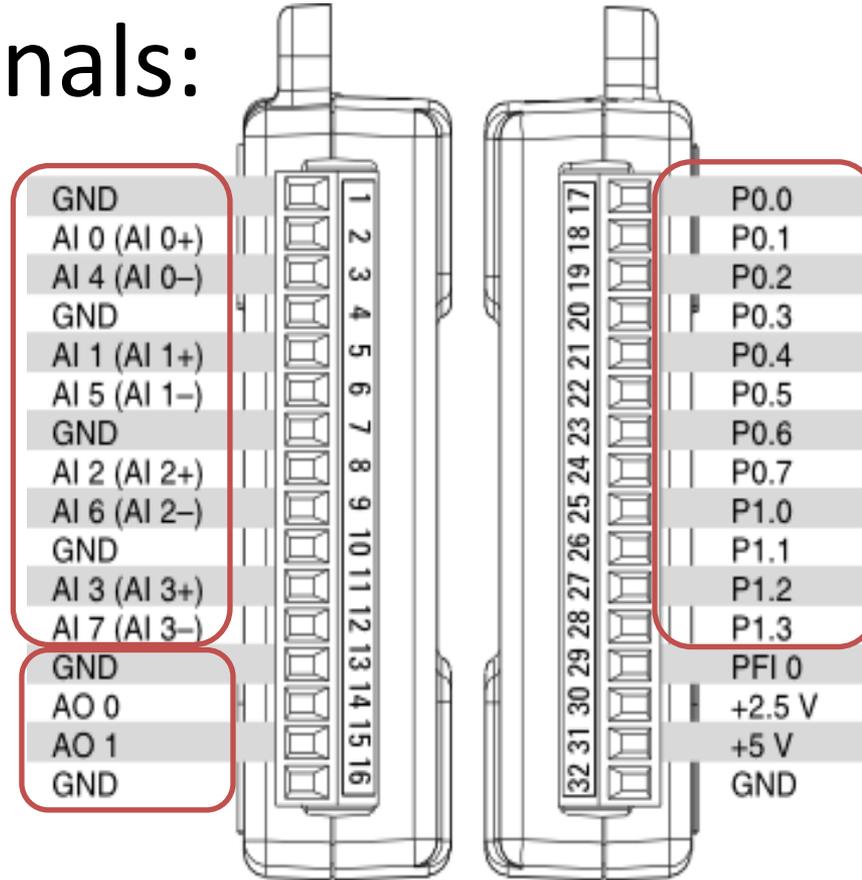
- USB-6008 is a DAQ Device from NI
- Can be used within LabVIEW
- NI-DAQmx Driver
- It has Analog and Digital Inputs and Outputs



# USB-6008

4 different types of Signals:

- AO – Analog Output
- AI – Analog Input
- DO – Digital Output
- DI – Digital Input



# 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**):

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

where  $A, B, C$  are constants given below [Wikipedia]

$A = 0.001129148, B = 0.000234125$  and  $C = 8.76741E - 08$

# 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

$$A = 0.001129148$$

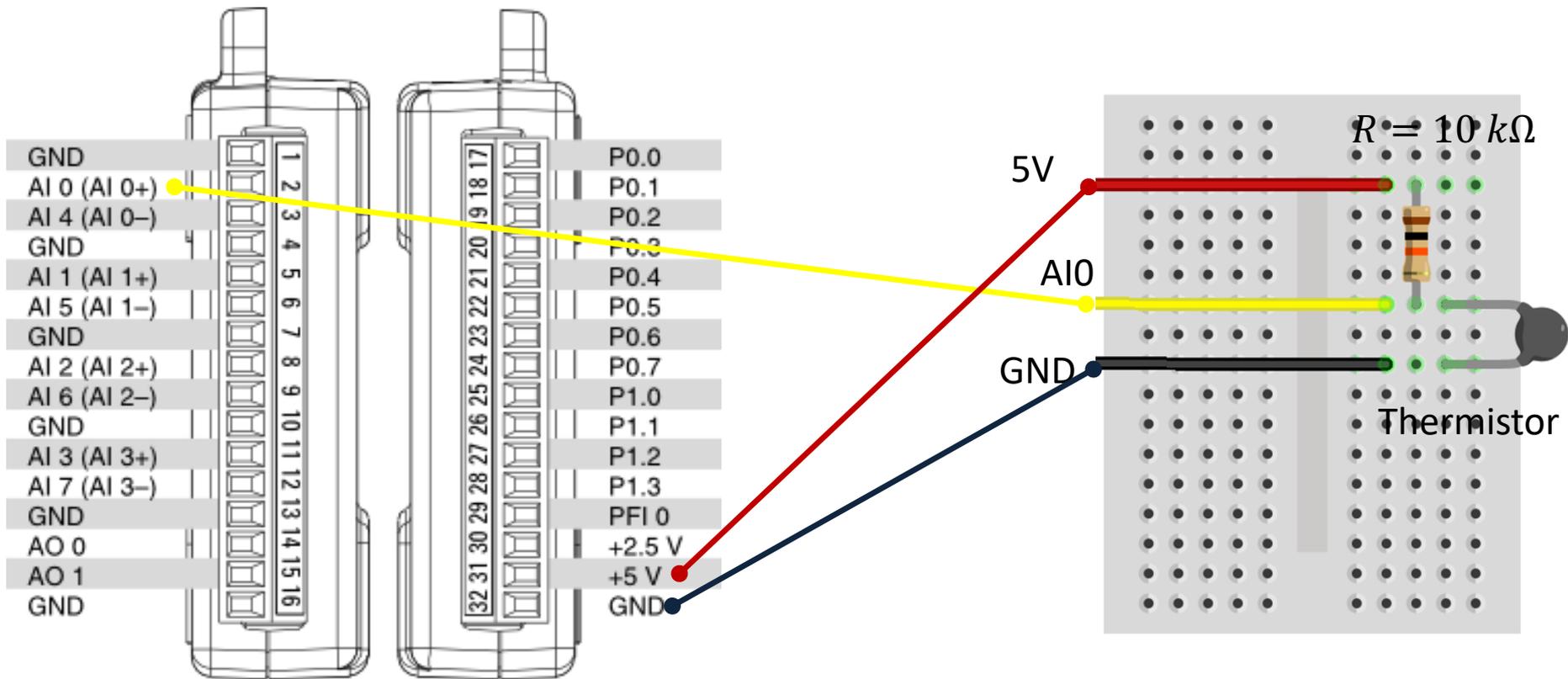
$$B = 0.000234125$$

$$C = 0.0000000876741$$

The Temperature in degrees Celsius will then be:

$$T_C = T_K - 273.15$$

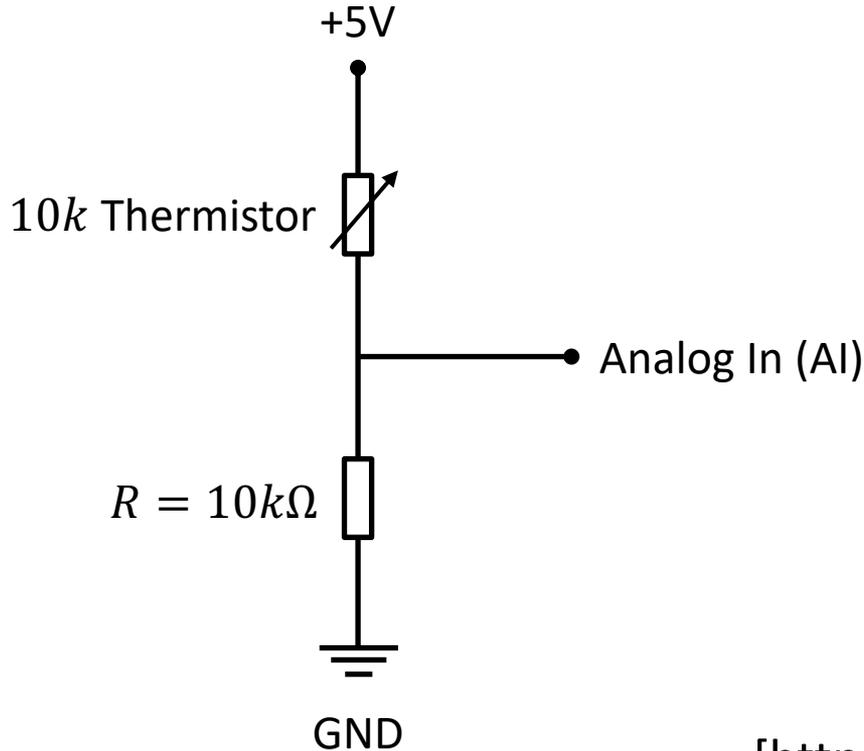
# Wiring





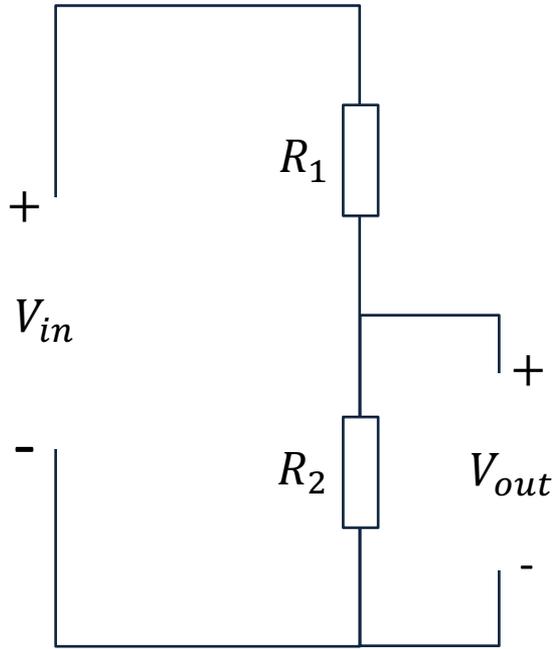
# Voltage Divider

The wiring is called a “Voltage divider”:



[[https://en.wikipedia.org/wiki/Voltage\\_divider](https://en.wikipedia.org/wiki/Voltage_divider)]

# General Voltage Divider



Formula:

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

# Voltage Divider for our System

Voltage Divider Equation:

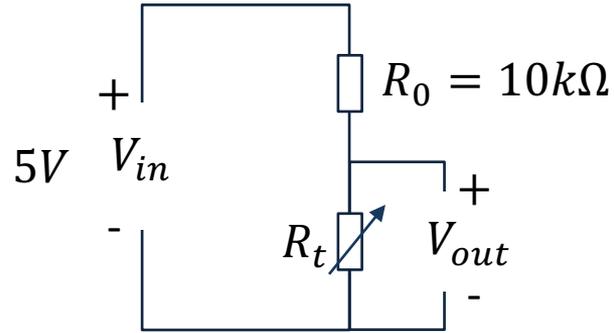
$$V_{out} = V_{in} \frac{R_t}{R_0 + R_t}$$

We want to find  $R_t$ :

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

Steps:

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



$R_t$  - 10k Thermistor. This varies with temperature. From Datasheet we know that  $R_t = 10k\Omega @ 25^\circ\text{C}$

# Pseudo Code

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

# Pseudo Code

```
float Vin = 5;
float Ro=10000;
//Voltage Divider
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;
```

# LabVIEW Example

Thermistor10K Example.vi Front Panel

Vout: 2.55

Rt: 10421.4

TempC: 24.1

Steinhart-Hart Equation - Formula Node.vi Block Diagram

```
float Vin = 5;
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;
```

$1/T = A + B*(\ln R) + C*(\ln R)^3 \rightarrow T = 1 / (A + B*(\ln R) + C*(\ln R)^3)$

Thermistor10K Example.vi Block Diagram

DAQ Assistant data

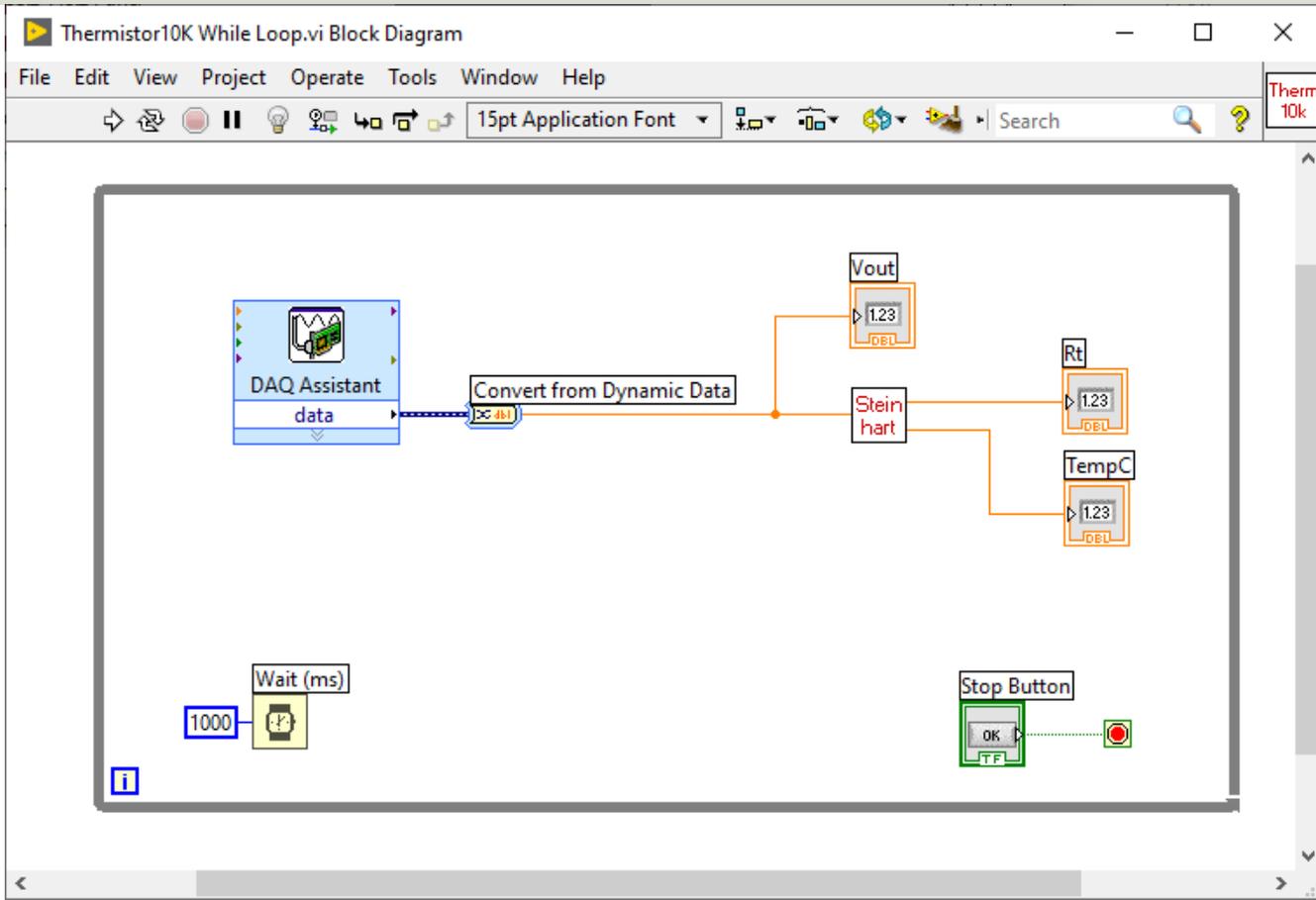
Steinhart

Vout

Rt

TempC

# Continuous Reading



# Formula Node

The screenshot shows a LabVIEW block diagram titled "Steinhart-Hart Equation - Formula Node.vi Block Diagram". The interface includes a menu bar (File, Edit, View, Project, Operate, Tools, Window, Help), a toolbar with various icons, and a font dropdown set to "15pt Application Font". A search bar is also visible.

The main workspace contains a "Formula Node" block with the following code:

```
float Vin = 5;  
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;
```

The code is connected to three numeric display controls: "Vout" (input, value 1.23), "Rt" (output, value 1.23), and "TempC" (output, value 1.23). A yellow highlight at the bottom of the diagram shows the rearranged equation:  $1/T = A + B*(\ln R) + C*(\ln R)^3 \rightarrow T = 1 / (A + B*(\ln R) + C*(\ln R)^3)$ .

# MathScript Node

Steinhart-Hart Equation.vi Block Diagram

File Edit View Project Operate Tools Window Help

15pt Application Font

Steinhart

MathScript Node

```
1 Vin = 5;  
2 Ro=10000; %10k Resistor  
3 Rt = (Vout*Ro)/(Vin-Vout);  
4 %Rt=10000; Used for Testing. Setting Rt=10k should give TempC=25  
5  
6 %Steinhart constants  
7 A = 0.001129148;  
8 B = 0.000234125;  
9 C = 0.0000000876741;  
10  
11 % Steinhart-Hart Equation  
12 TempK = 1 / (A + (B * log(Rt)) + (C * log(Rt)^3));  
13  
14 % Convert from Kelvin to Celsius  
15 TempC = TempK - 273.15;
```

Vout 1.23

Rt 1.23

TempC 1.23

$1/T = A + B*(\ln R) + C*(\ln R)^3 \rightarrow T = 1 / (A + B*(\ln R) + C*(\ln R)^3)$

# MathScript Code

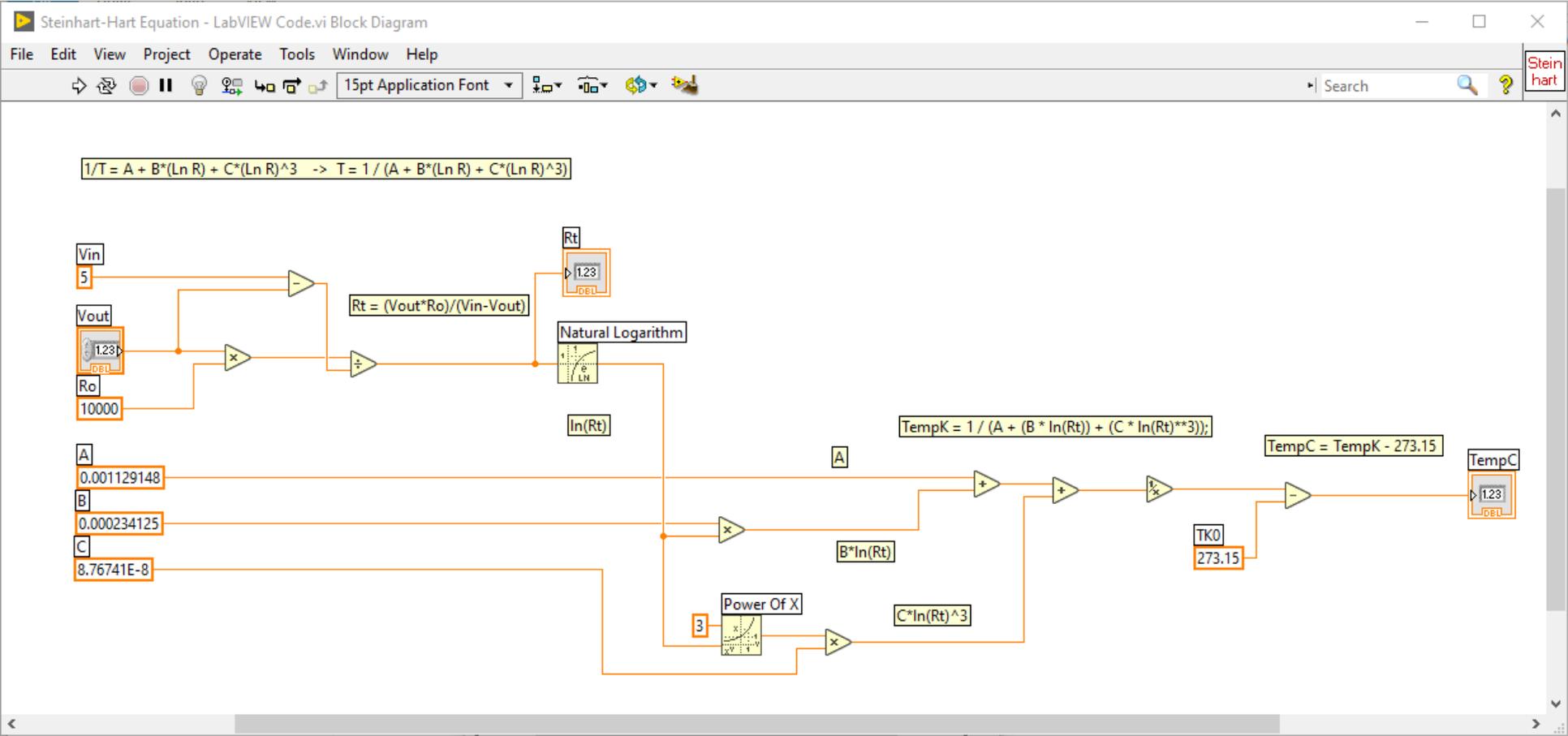
```
Vin = 5;  
Ro=10000; %10k Resistor  
Rt = (Vout*Ro)/(Vin-Vout);
```

```
%Steinhart constants  
A = 0.001129148;  
B = 0.000234125;  
C = 0.0000000876741;
```

```
% Steinhart-Hart Equation  
TempK = 1 / (A + (B * log(Rt)) + (C * log(Rt)^3));
```

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

# Pure LabVIEW Code



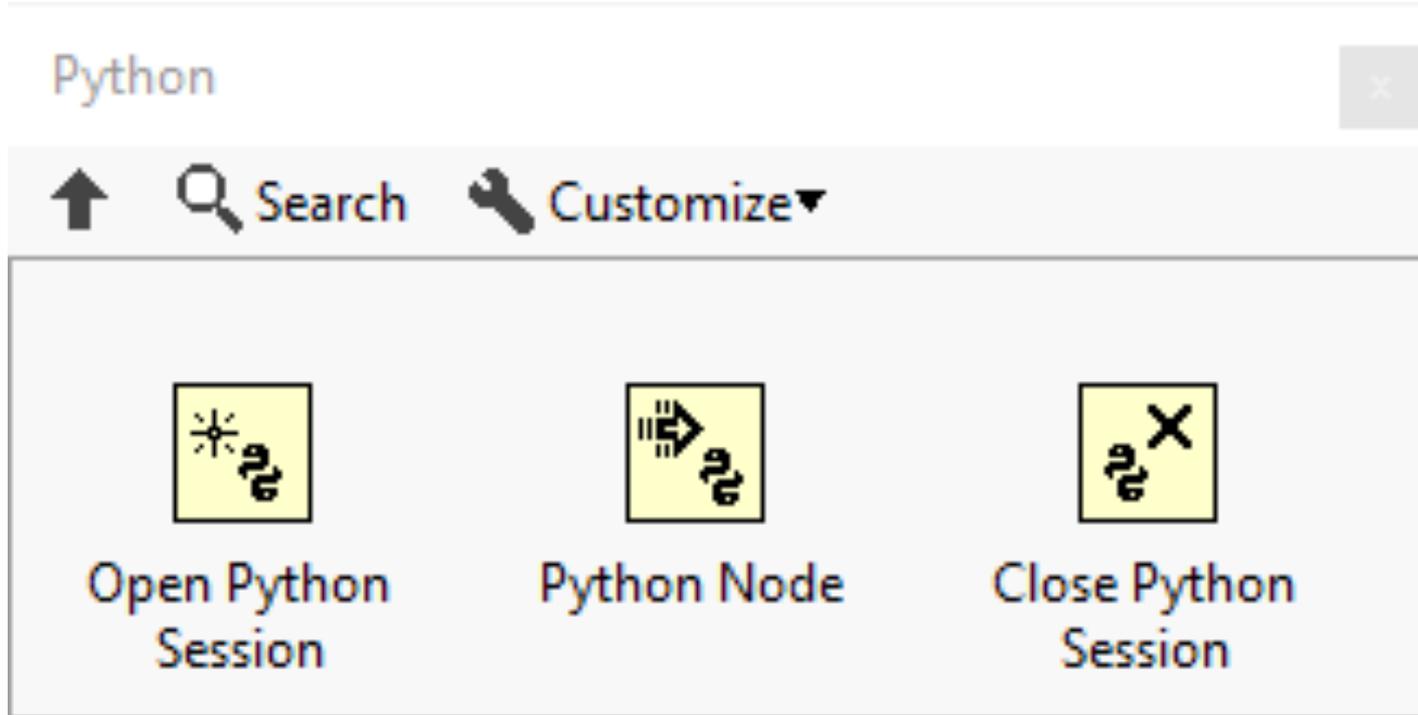
# MATLAB Script

The image shows a screenshot of a MATLAB Node-VI Block Diagram window titled "Steinhart-Hart Equation - MATLAB Node.vi Block Diagram". The window has a menu bar (File, Edit, View, Project, Operate, Tools, Window, Help) and a toolbar with various icons. The main area contains a "MATLAB script" block with the following code:

```
Vin = 5;  
Ro = 10000; %10k Resistor  
Rt = (Vout*Ro)/(Vin-Vout);  
%Rt= 10000; Used for Testing. Setting Rt= 10k should give TempC=25  
  
%Steinhart constants  
A = 0.001129148;  
B = 0.000234125;  
C = 0.0000000876741;  
  
% Steinhart-Hart Equation  
TempK = 1 / (A + (B * log(Rt)) + (C * log(Rt)^3));  
  
% Convert from Kelvin to Celsius  
TempC = TempK - 273.15;
```

The script is connected to several input and output blocks. On the left, an input block labeled "Vout" with a value of 1.23 is connected to the script. On the right, two output blocks are connected: "Rt" with a value of 1.23 and "TempC" with a value of 1.23. At the bottom of the script block, the equation  $1/T = A + B*(\ln R) + C*(\ln R)^3 \rightarrow T = 1 / (A + B*(\ln R) + C*(\ln R)^3)$  is displayed.

# LabVIEW Python Integration



# LabVIEW Python Integration

thermistor.py

```
import math as mt

# Function for finding Temperature in degrees Celsius
def thermistorTemp(Vout):
    # Voltage Divider
    Vin = 5;
    Ro = 10000 # 10k Resistor

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

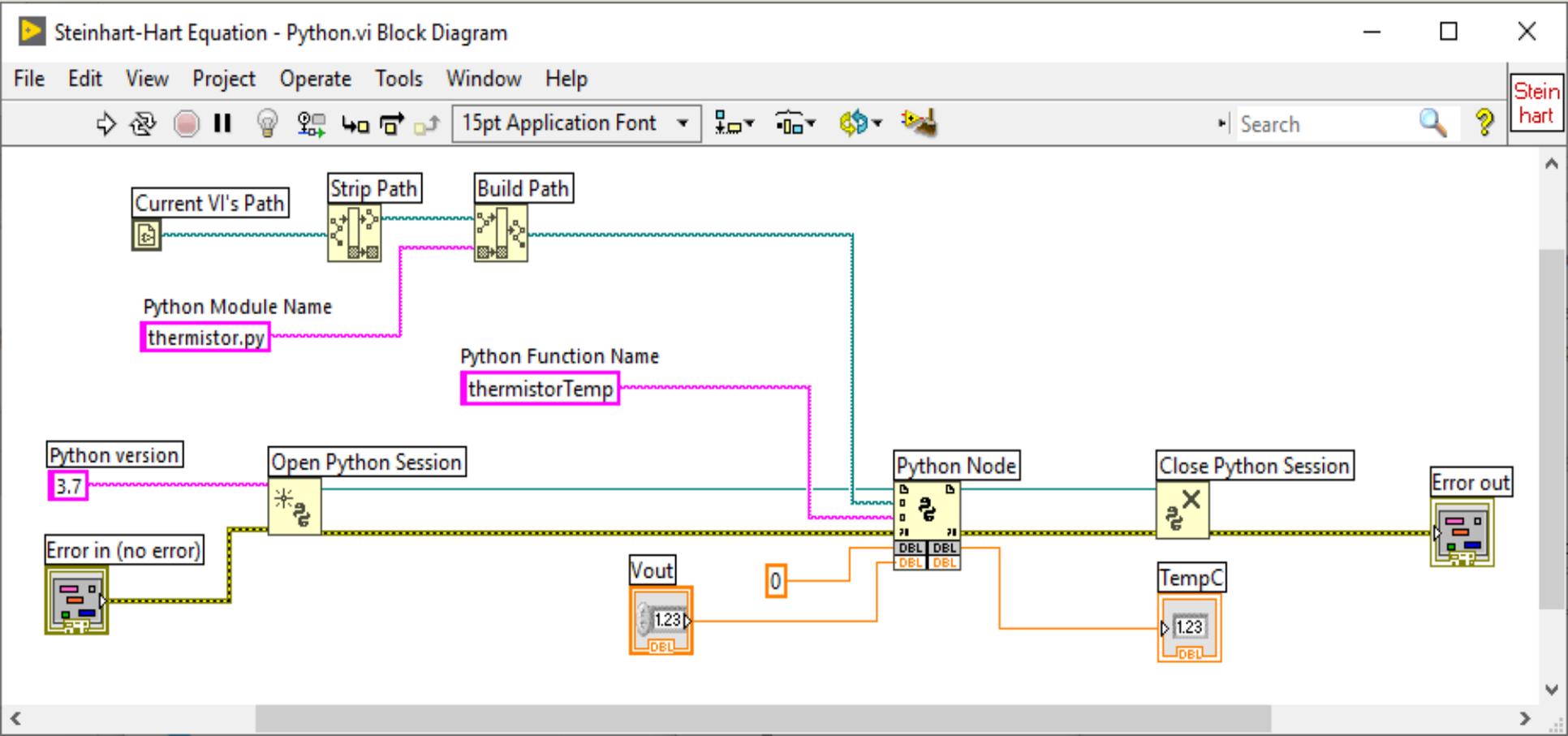
    # Calculate Resistance
    Rt = (Vout * Ro) / (Vin - Vout)

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

    # Convert from Kelvin to Celsius
    TempC = TempK - 273.15

    return TempC
```

# LabVIEW Python Integration



# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

