https://www.halvorsen.blog

LabVIEW in Automation

DAQ in LabVIEW

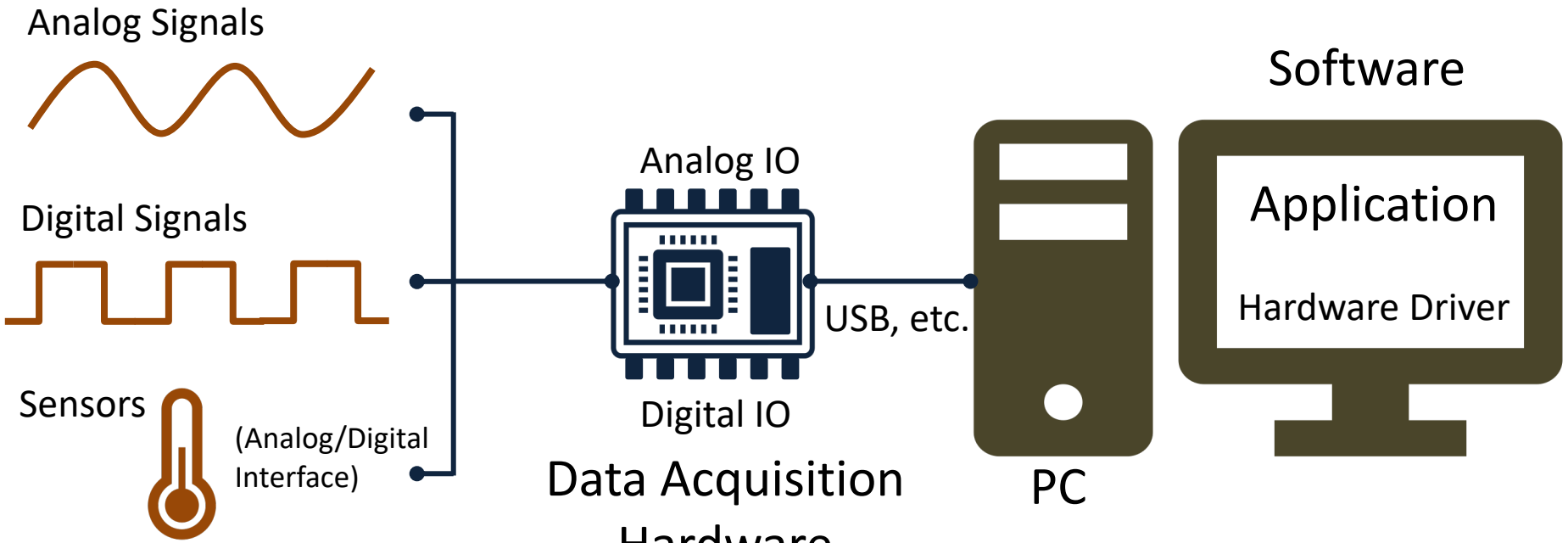
Hans-Petter Halvorsen

Contents

DAQ System

DAQ – Data Acquisition

Input/Output Signals

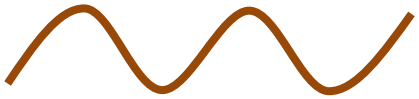


We will use an USB-6008 as the DAQ Hardware

I/O Module

Analog Sensors

Analog Signals



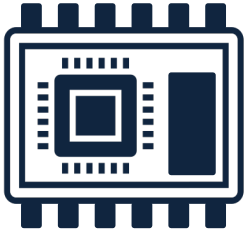
$0 - 5V$

Analog IO

Analog Input (**AI**)

Analog Output (**AO**)

I/O Module



Digital IO

Digital Signals

True

False

Digital Input (**DI**)

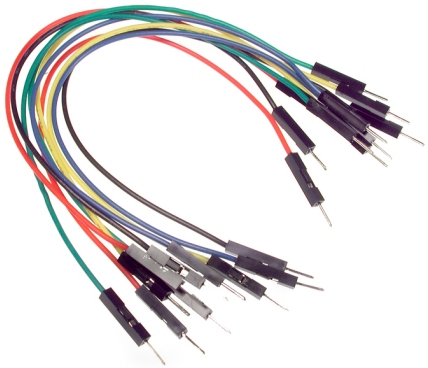
Digital Output (**DO**)

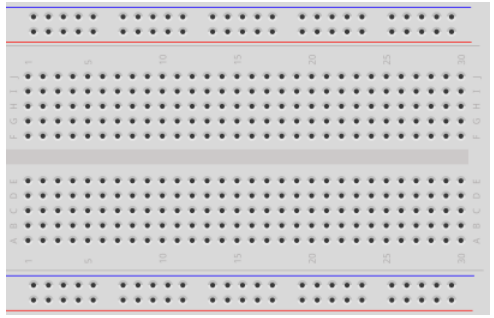
Sensors with Digital Interface (e.g., SPI, I2C)

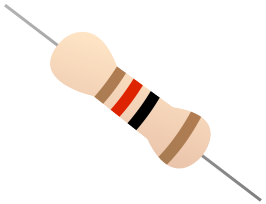
Hardware

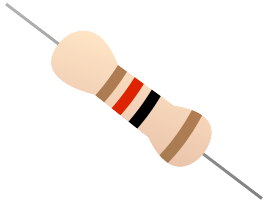
- DAQ Device (e.g., USB-6008)
- Breadboard
- Wires (Jumper Wires)
- TMP36 Temperature Sensor
- Thermistor 10K (Temperature Sensor)
- Resistors, $R = 270\Omega$, $R = 10k\Omega$
- LEDs (Colors: Red, Green)

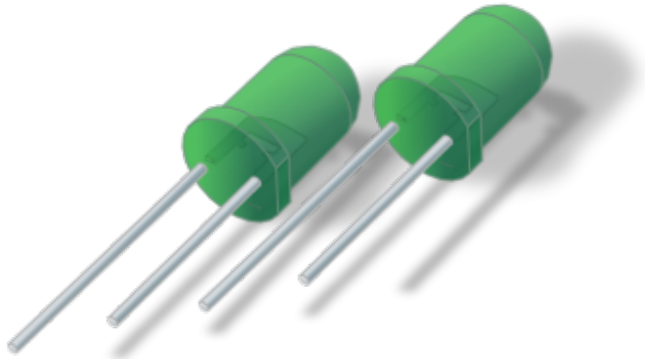
Hardware












NATIONAL INSTRUMENTS
NI USB-6008
8 Inputs, 12-bit, 10 kS/s Multifunction I/O
Analog
Digital

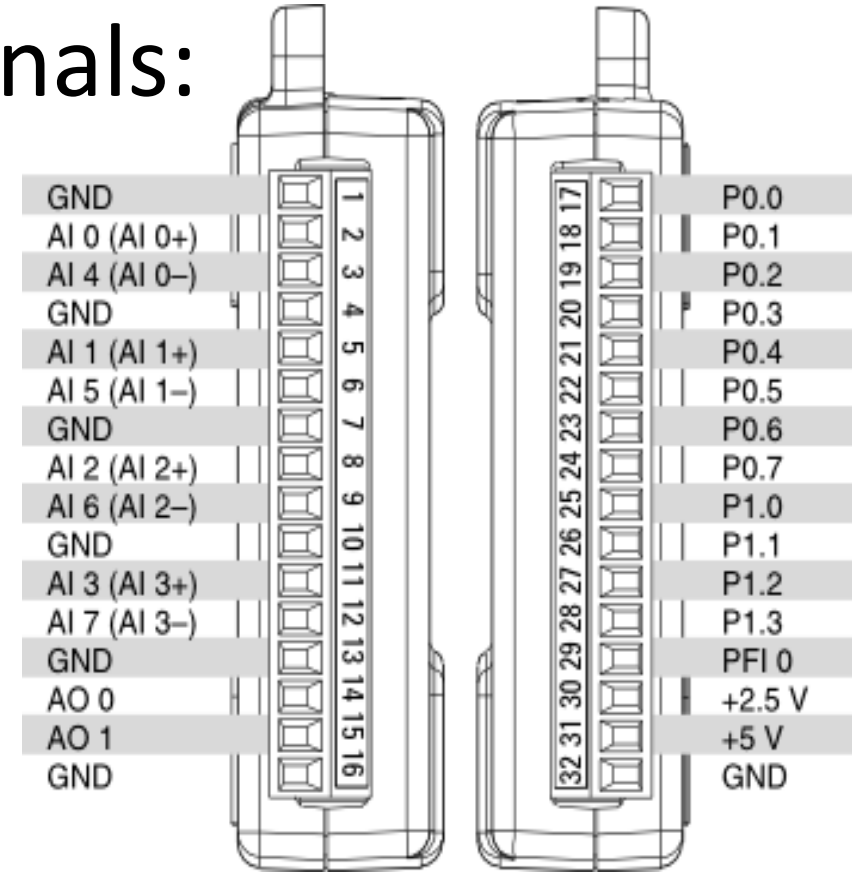
USB-6008

- 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:

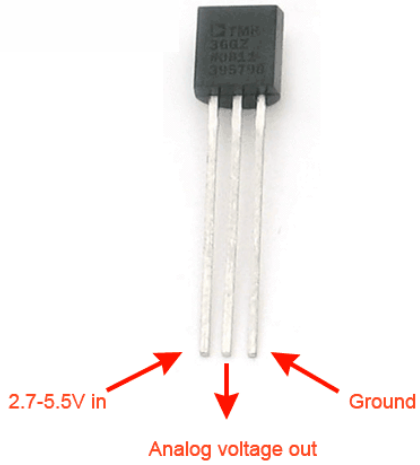
- AO – Analog Output
- AI – Analog Input
- DO – Digital Output
- DI – Digital Input

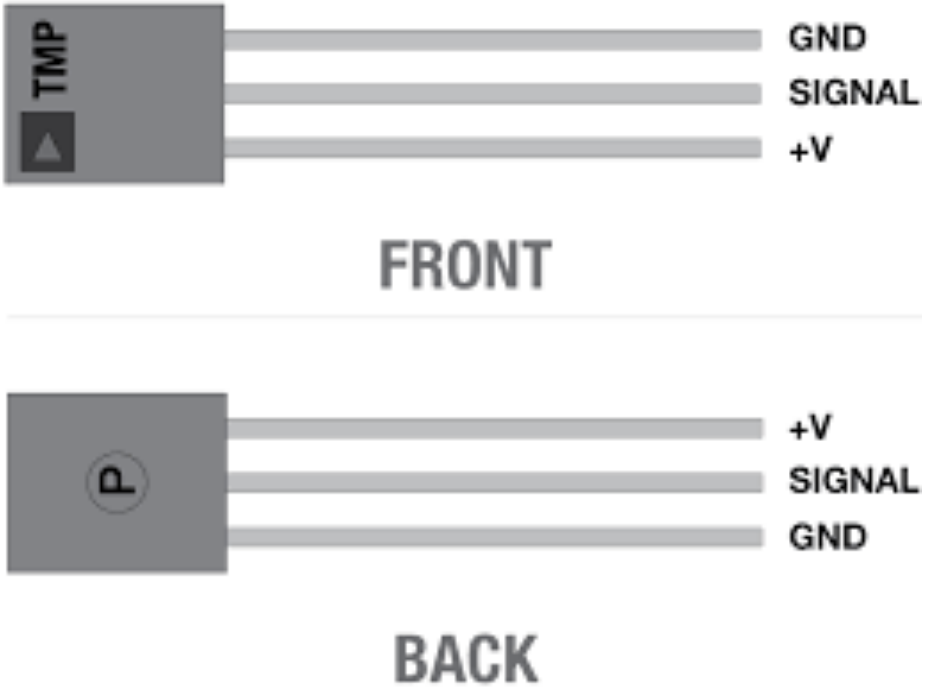


Temperature Sensors

- TMP36 Temperature Sensor
- Thermistor 10K Temperature Sensor

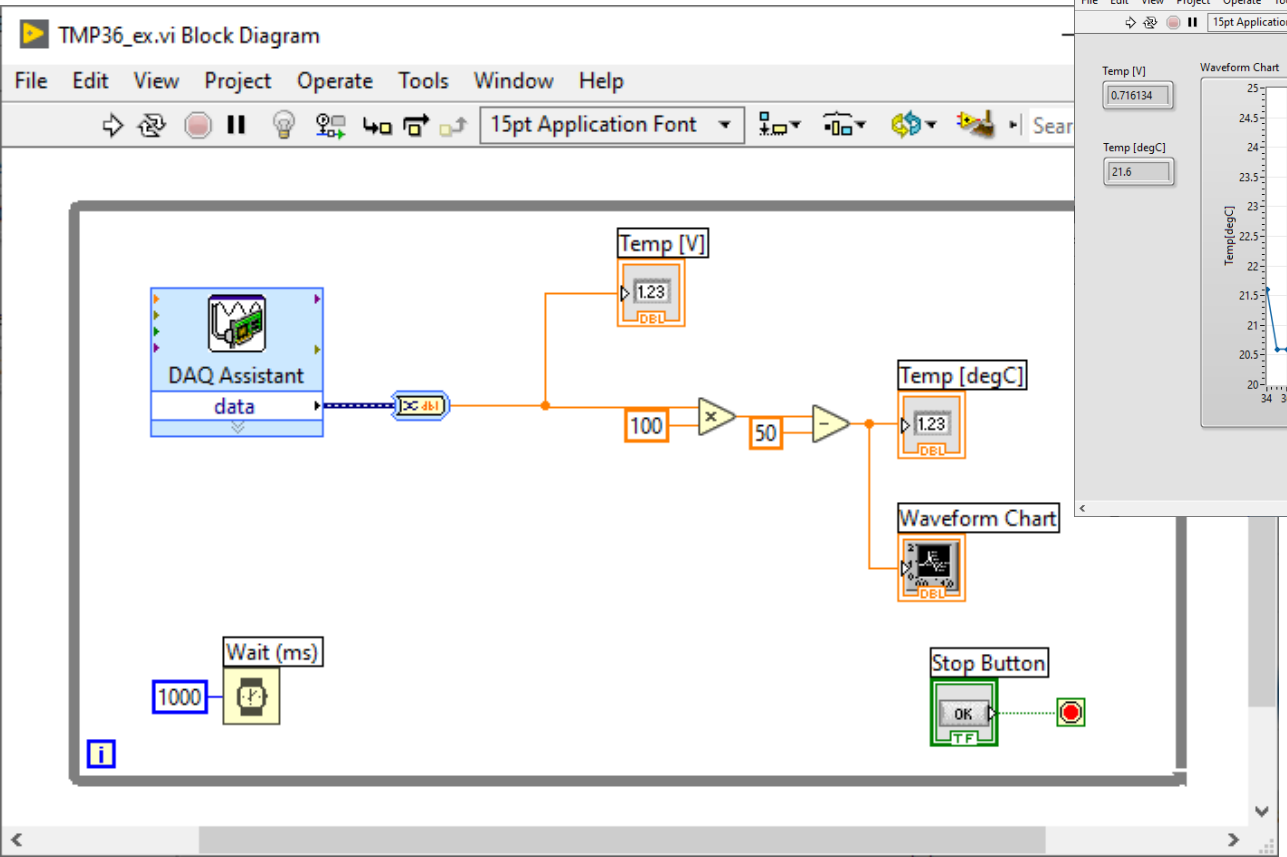
TMP36

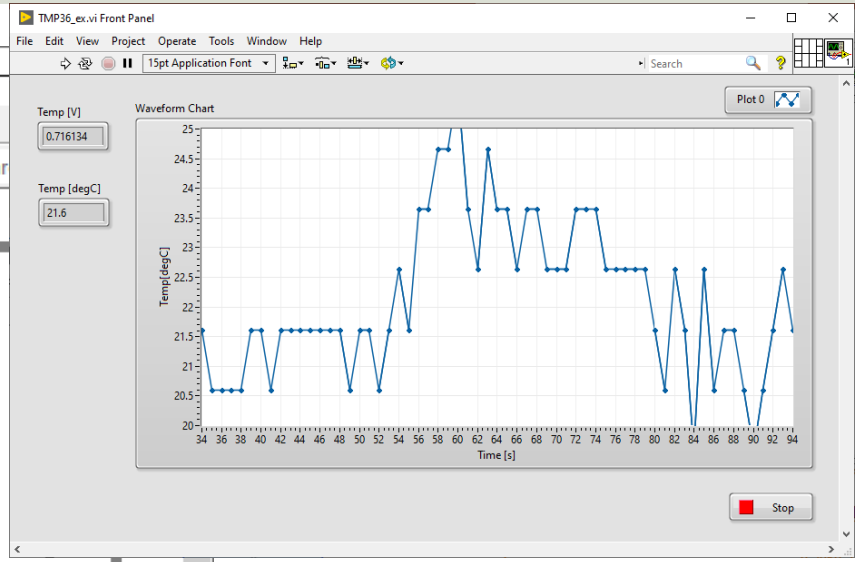




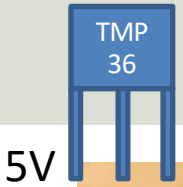
TMP is a small, low-cost temperature sensor and cost about $1 (you can buy it “everywhere”)

TMP37 LabVIEW Example





Hardware Setup TMP36



We connect the TMP36 to LabVIEW using a USB DAQ Device from National Instruments, e.g., USB-6001, USB-6008 or similar. I have used a breadboard for the wiring.

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**):

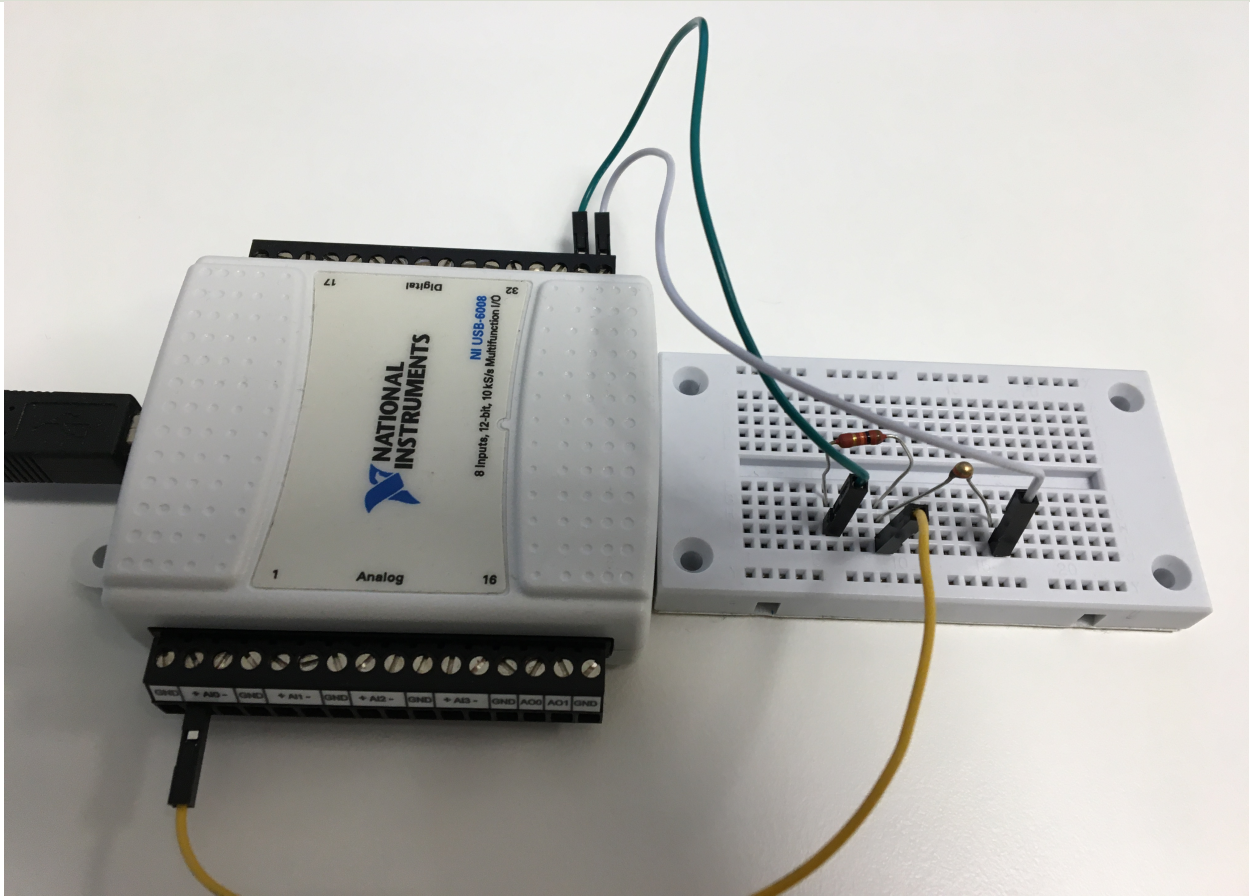
$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3$$

where A, B, C are constants given below

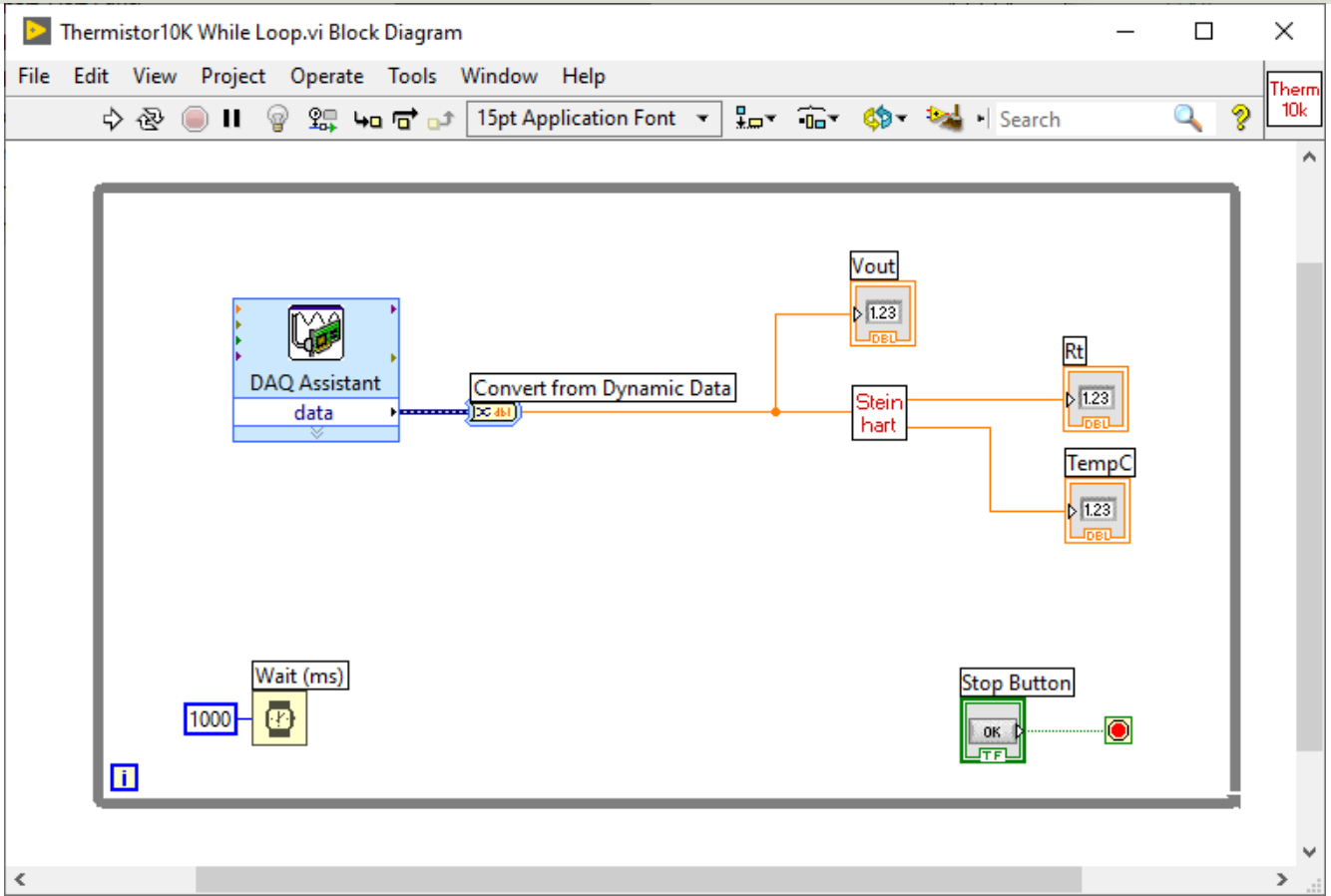
$A = 0.001129148, B = 0.000234125$ *and* $C = 8.76741E-08$

[Wikipedia]

Hardware Setup Thermistor

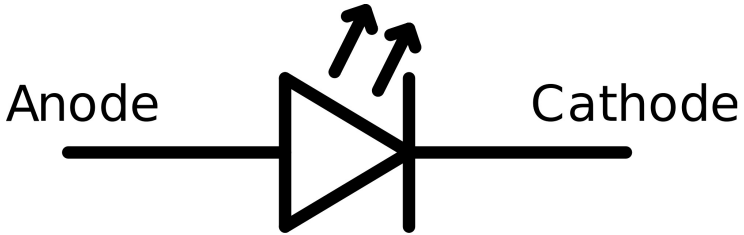


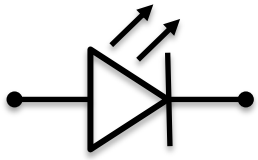
Thermistor LabVIEW Example

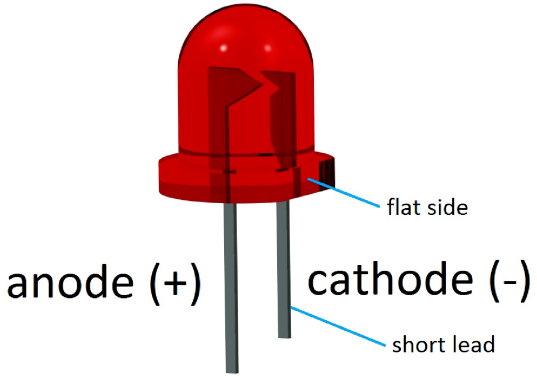


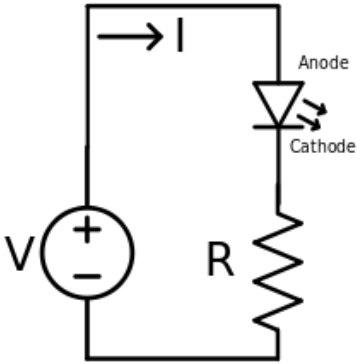
Light-emitting diode - LED

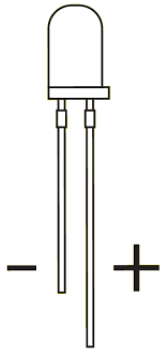
A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it



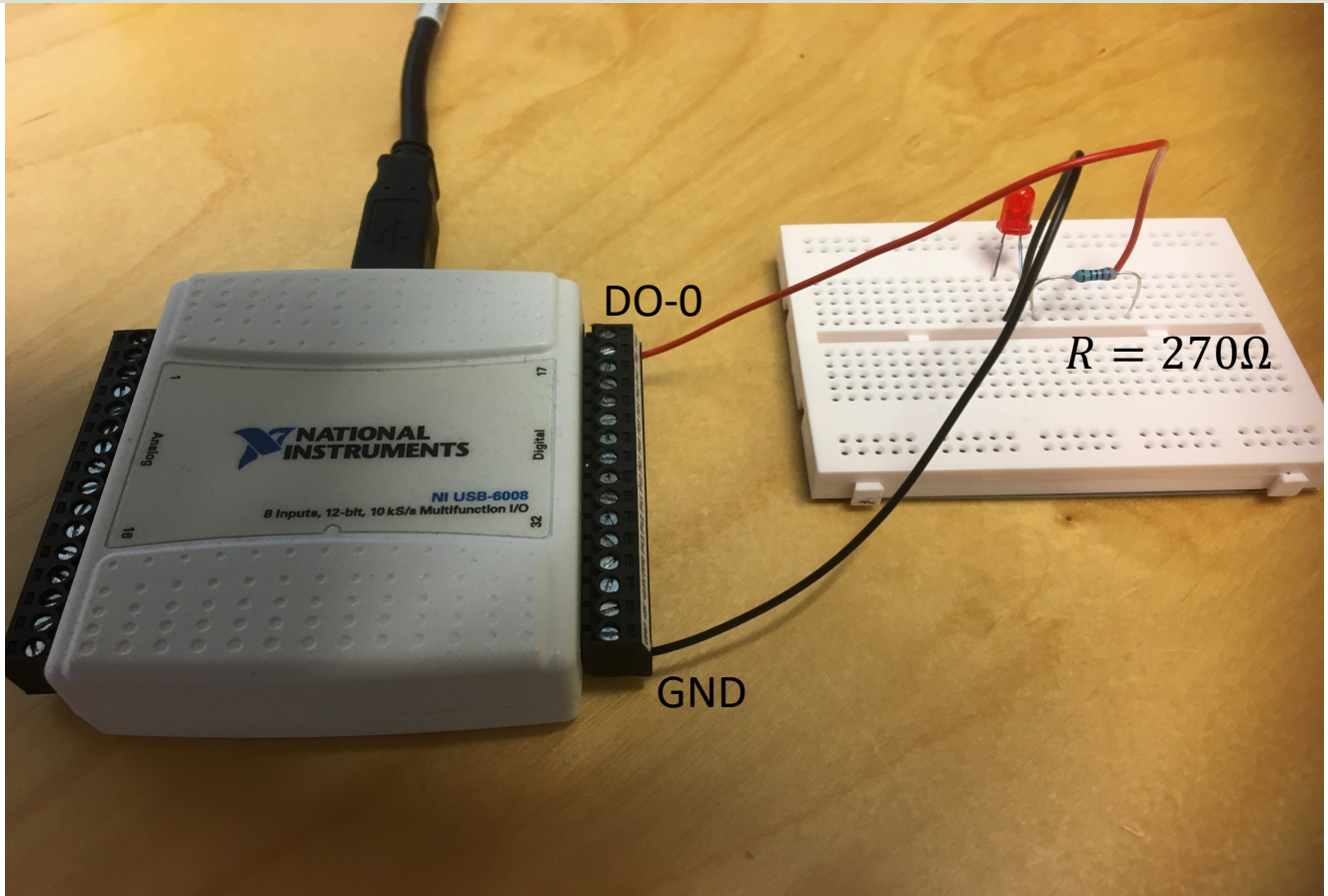




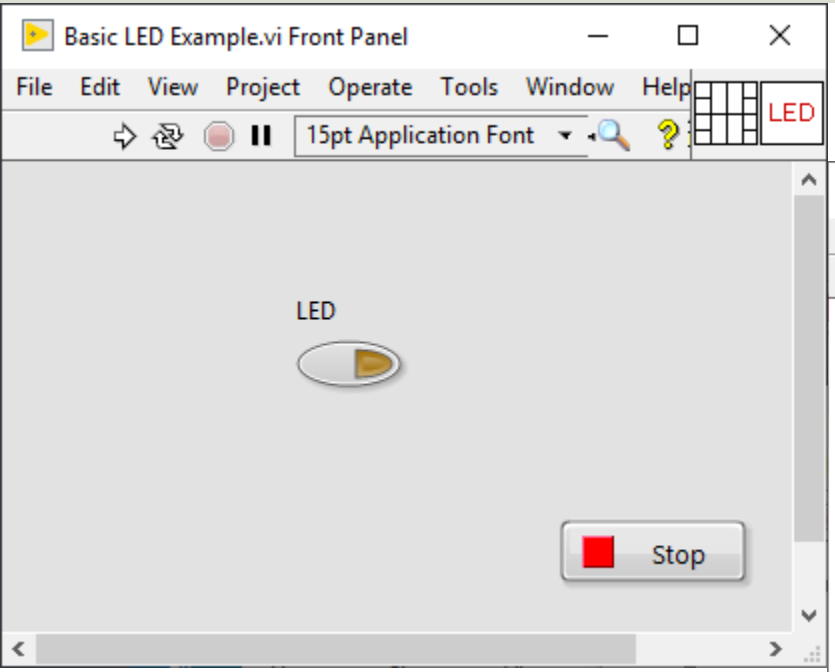


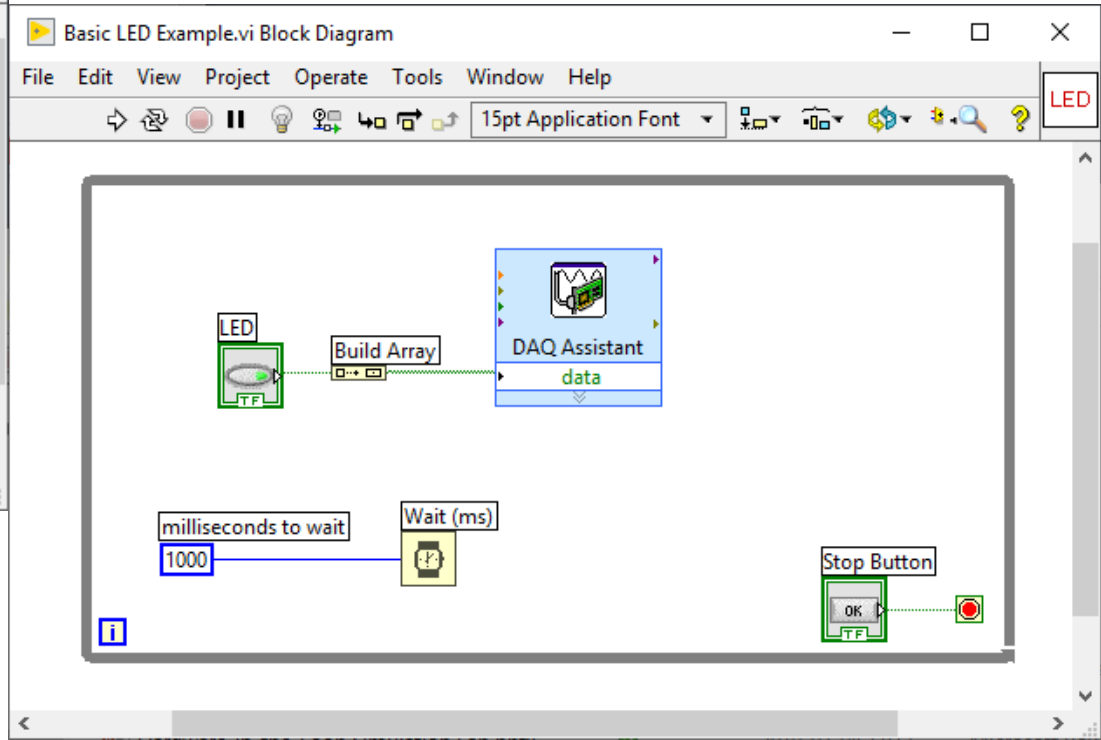


Hardware Setup LED



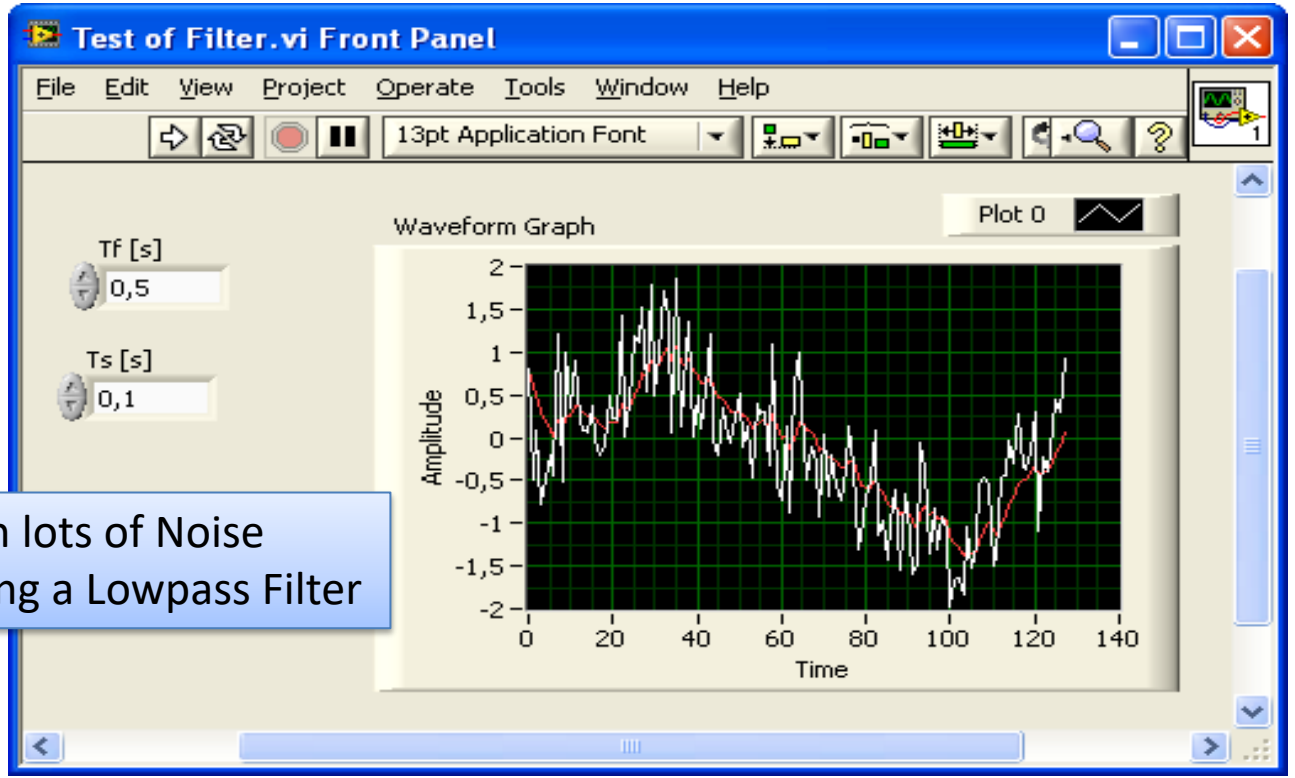
LED LabVIEW Example





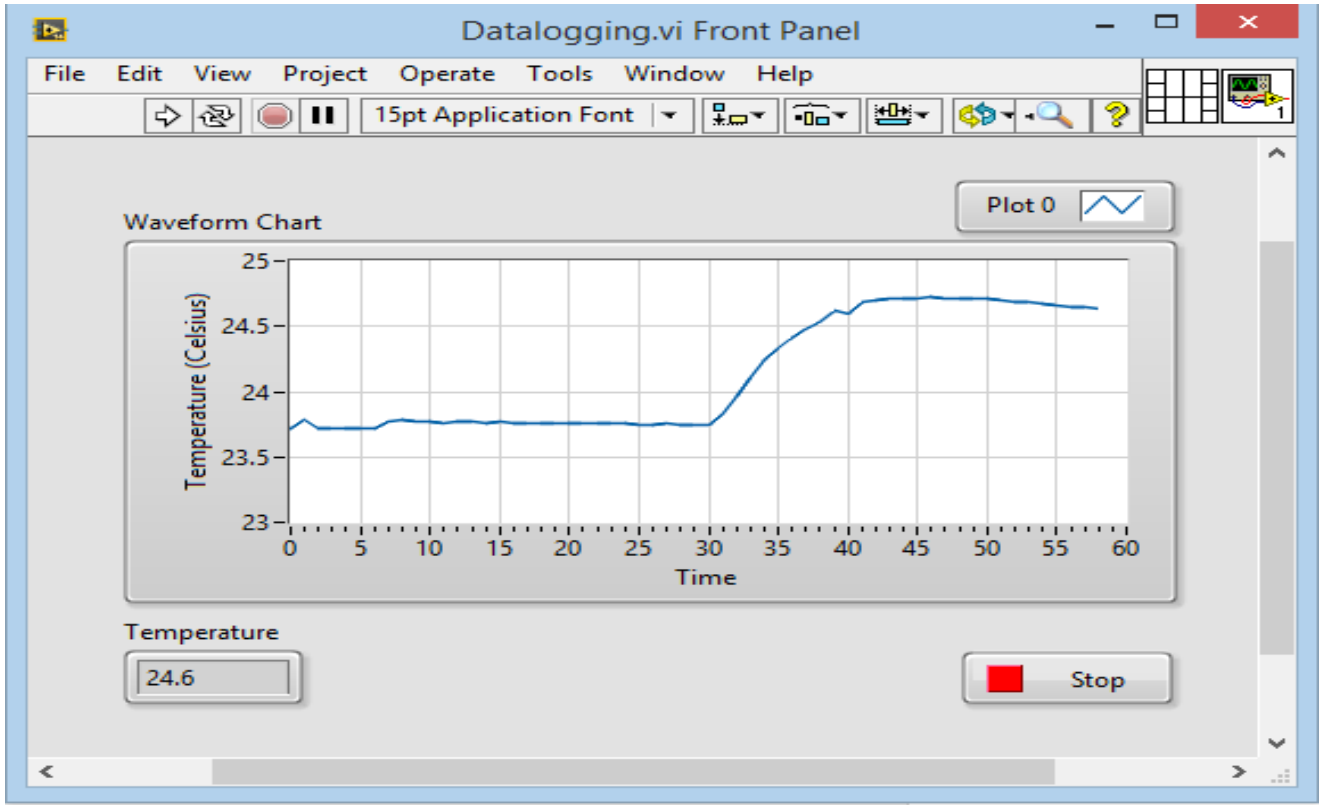
Lowpass Filter

Purpose:
Remove Noise from the Measured Signal



White: Measured Signal with lots of Noise
Red: The results after applying a Lowpass Filter

Logging Data to a Text File

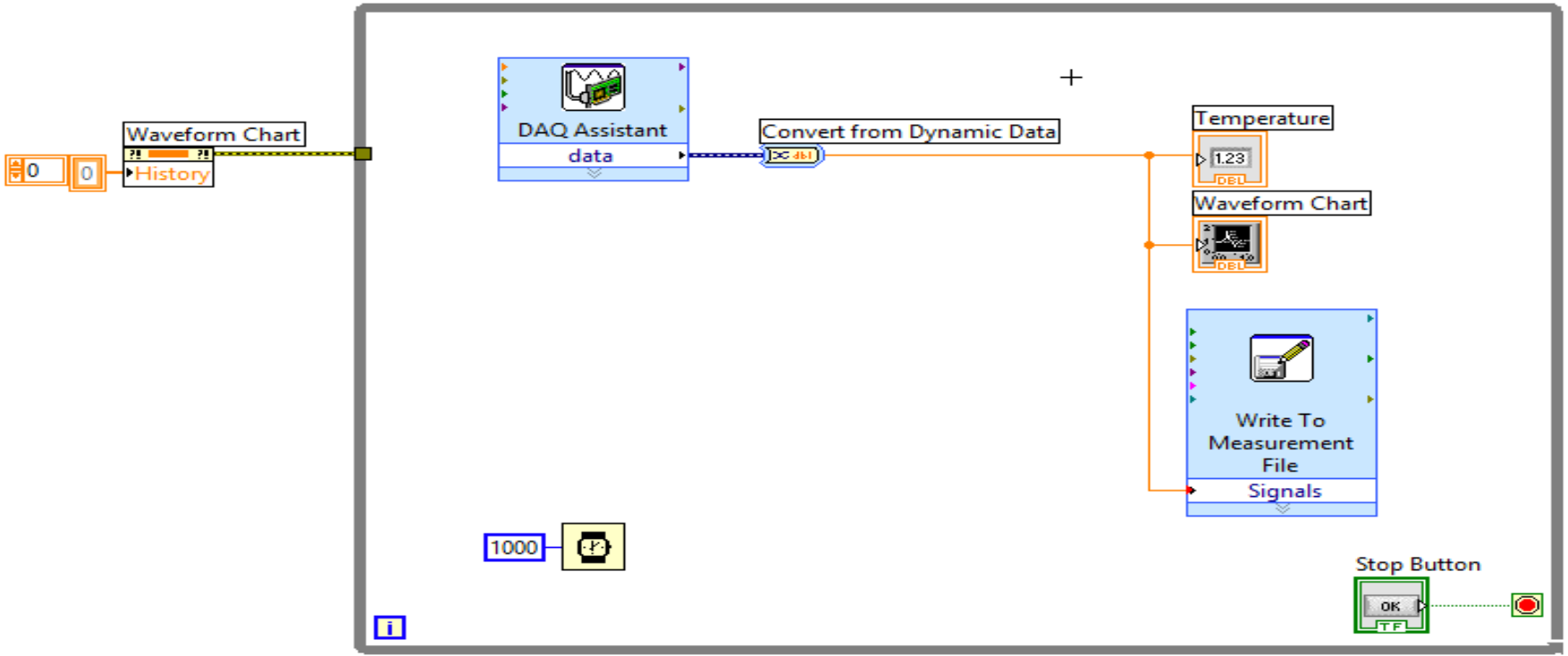


Data.lvm - Notepad

File Edit Format View Help

0.000000	23.722386
0.975883	23.782507
1.973000	23.714294
2.977028	23.719689
3.975200	23.719689
4.976168	23.716991
5.974145	23.714294
6.977184	23.774415
7.977247	23.779810
8.976395	23.777113
9.976493	23.771718
10.980489	23.763626
11.976687	23.771718
12.980719	23.766323
13.982748	23.763626
14.983700	23.766323
15.979765	23.763626
16.977789	23.760928
17.979809	23.760928
18.977904	23.760928
19.976963	23.758231
20.977973	23.755534
21.979071	23.755534
22.980054	23.752836
23.979137	23.752836
24.978214	23.750139
25.978157	23.747441
26.978513	23.752836

Logging Data to a Text File



Delivery

Use a USB-6008 I/O Module to collect data and do some basic analysis of the data using both the 2 different temperature sensors (both "TMP36" and 10K Thermistor)

Requirements:

- You should create a proper **GUI**. You should see the current temperature value(s) from the sensor(s) on the Front Panel. The values should be presented in both degrees Celsius and degrees Fahrenheit.
- Make sure to use proper **numbers of decimals** in your GUI, etc. Showing e.g. temperature values with 4 decimals makes no sense.
- You should **Plot** the values from the Sensor(s).
- You should also find the **average** values for each sensor and the average value for the temperature in total (if you use more than one sensor).
- **Alarms**: Turn on a Boolean indicator when the temperature reach a specific limit. If you have a LED: Turn also on a red LED when the temperature reach the specific limit. If the temperature is below the limit, a green LED should be on.
- You should see if using a **Lowpass Filter** will improve your readings from the sensor(s). Use one of the built-in filters in LabVIEW or make your own Lowpass Filter.
- The **Data should also be stored in a Text File**, which should later be analyzed in **Excel** (make a plot, do some basic statistics like mean standard deviation, etc.).
- The code should be well structured and intuitive.
- It should contain **basic LabVIEW features** like While Loop, Case Structure, SubVIs, Arrays, Property Nodes and Clusters, etc.
- You should use the **Project Explorer**.

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog