

<https://www.halvorsen.blog>



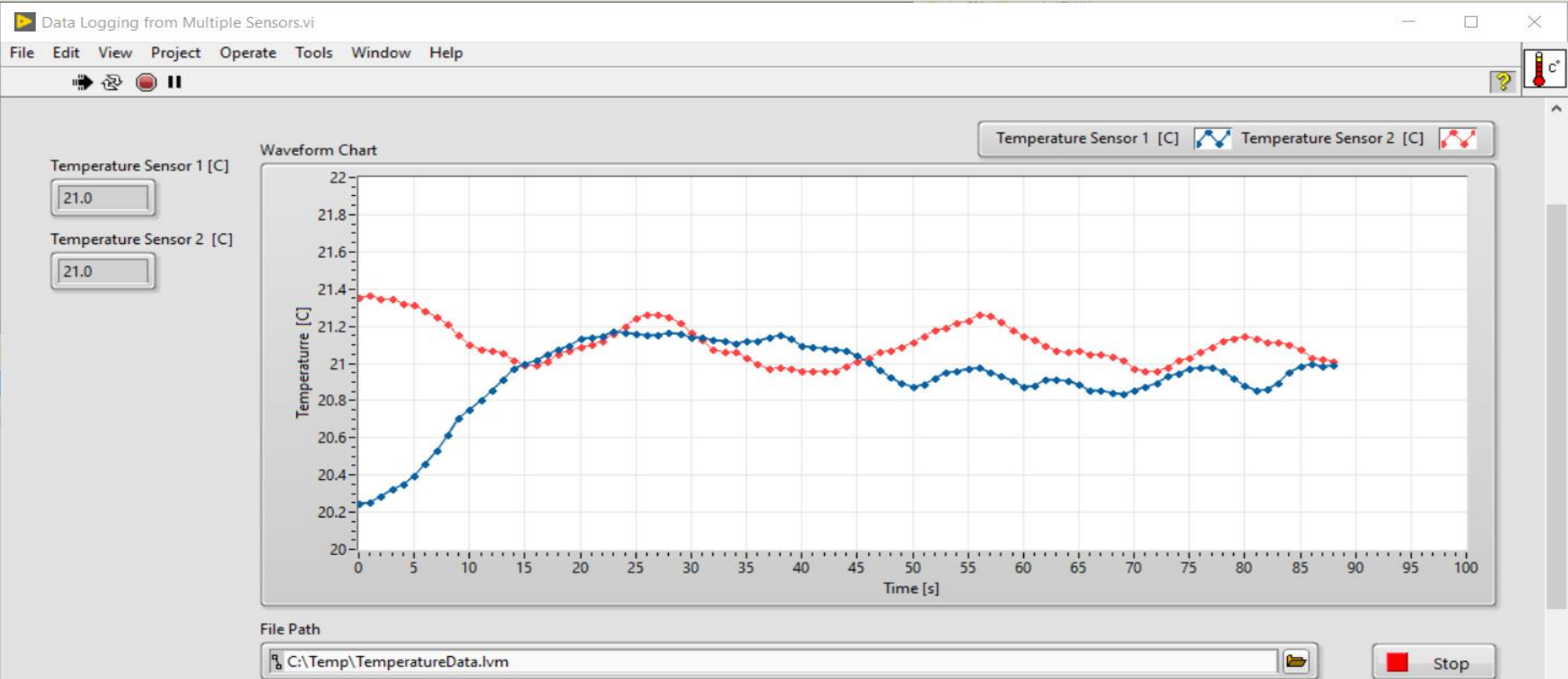
Logging Data from Multiple Sensors to Text File in LabVIEW

Hans-Petter Halvorsen

Contents

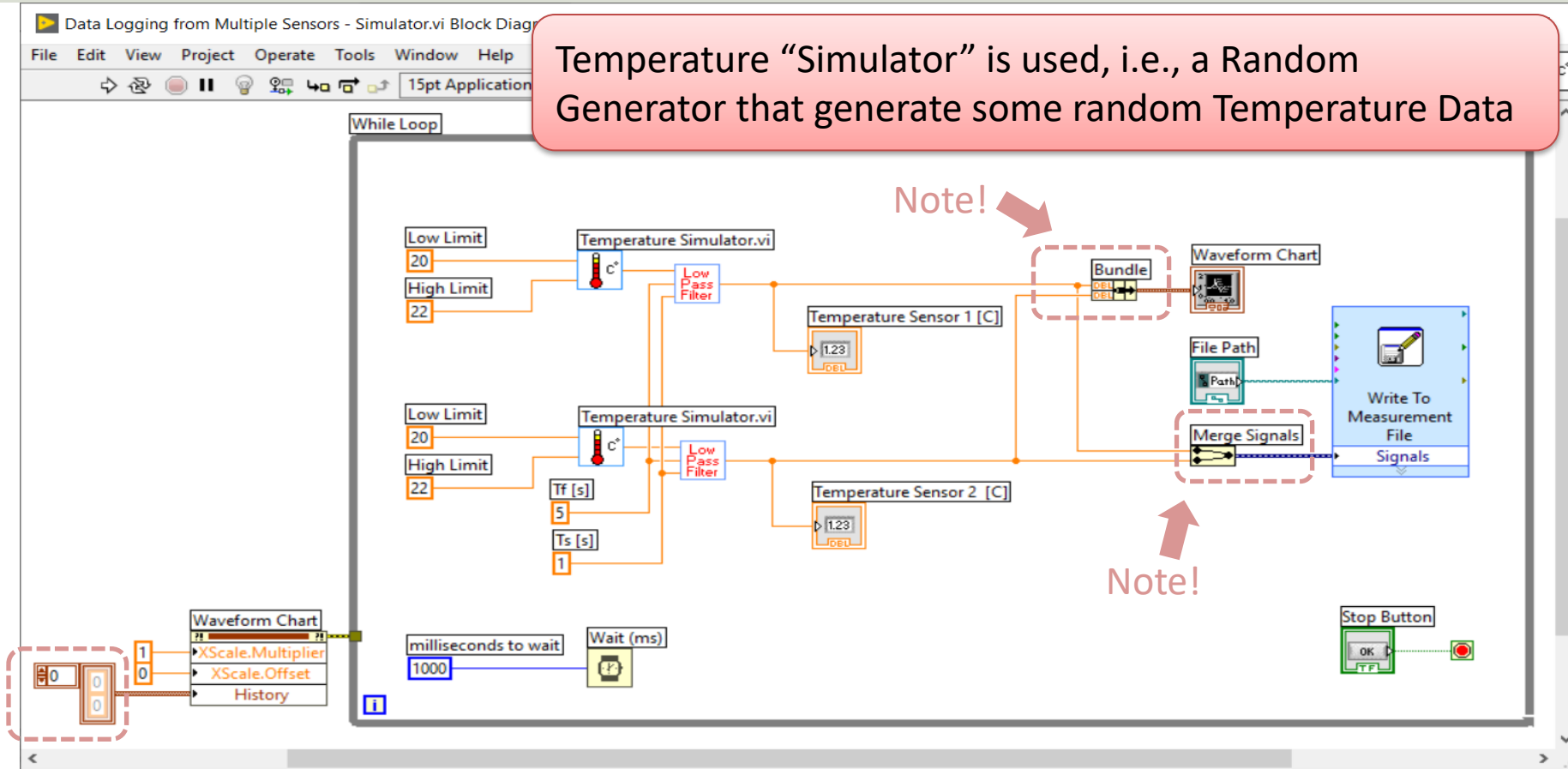
- Logging Temperature Data to Text File in LabVIEW:
<https://youtu.be/lxAsW2uVb9s>
- In the tutorial above we used 1 Temperature Sensor – What if we have 2 Sensors? (or more than 2?)
- This tutorial shows how to read data from 2 sensors using a DAQ device and saving the data to a Text File in LabVIEW

LabVIEW Application



LabVIEW Code

Temperature "Simulator" is used, i.e., a Random Generator that generate some random Temperature Data



Logged Data

TemperatureData.lvm - Notepad

File	Edit	Format	View	Help
0.000000		20.967908		21.483000
1.001163		20.968071		21.452208
2.000908		20.990304		21.440139
3.000337		21.030603		21.437771
4.000916		21.032733		21.403616
5.001221		21.045383		21.367044
6.001272		21.041305		21.310924
7.001723		21.014337		21.237776
8.001851		21.012491		21.181544
9.001402		21.033078		21.158482
10.001266		21.069751		21.148573
11.003134		21.117899		21.111289
12.003939		21.121751		21.100976
13.005329		21.144116		21.111218
14.007066		21.181790		21.101876
15.008220		21.187309		21.071323
16.007552		21.175946		21.057514
17.009147		21.178308		21.026114
18.008528		21.156333		21.011413
19.009591		21.138625		20.991855
20.009260		21.117530		20.973851
21.009494		21.081788		20.974295
22.010872		21.077449		20.988406
23.012124		21.065772		21.015741
24.011428		21.047252		21.041307
25.012150		21.046908		21.079747
26.012938		21.068329		21.121360
27.013172		21.096087		21.137585
28.014173		21.101500		21.165631
29.013848		21.129122		21.201625
30.015204		21.138425		21.211084
31.015820		21.157356		21.200788
32.015516		21.185103		21.169901
33.017083		21.219731		21.143720
34.016951		21.256801		21.130904
35.016758		21.261623		21.112903
36.018091		21.267781		21.105305
37.017422		21.278334		21.106600

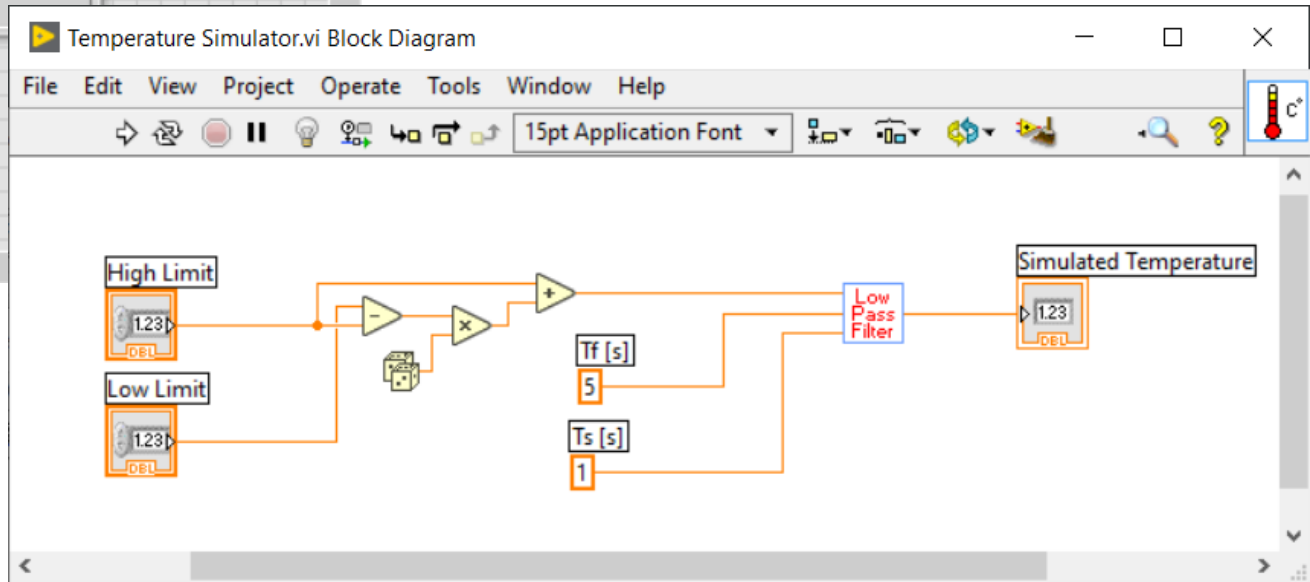
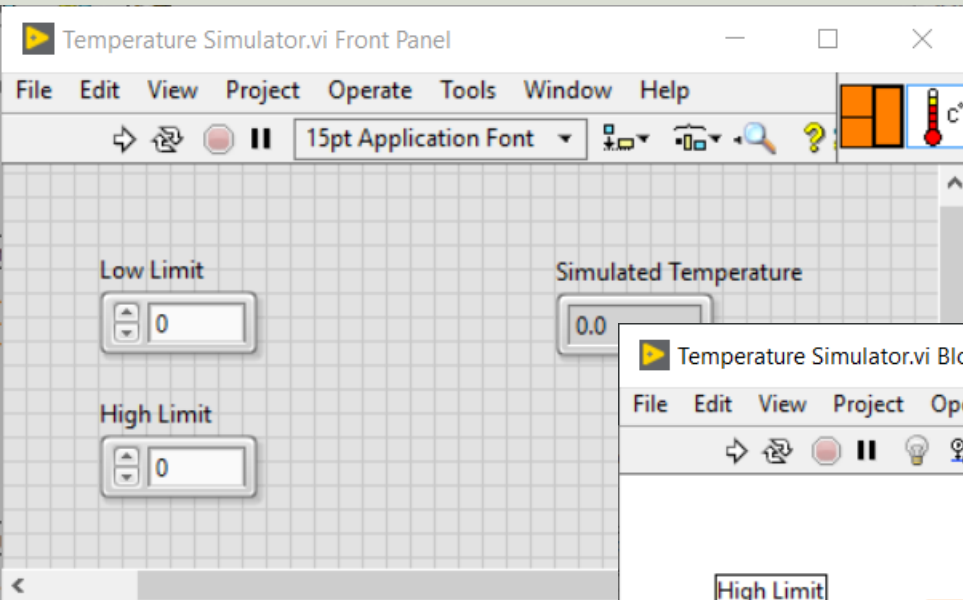
Ln 1, Col 1 100% Windows (CRLF) UTF-8

Time

Temperature Sensor 1

Temperature Sensor 2

Temperature Simulator



Reentrant Execution

The screenshot shows the 'VI Properties' dialog box with the 'Execution' category selected. The 'Allow debugging' checkbox is checked. Under the 'Reentrancy' section, 'Preallocated clone reentrant execution' is selected. The 'Priority' is set to 'normal priority' and the 'Preferred Execution System' is 'same as caller'. Other options include 'Enable automatic error handling', 'Run when opened', 'Suspend when called', 'Clear indicators when called', and 'Auto handle menus at launch', all of which are checked. The 'Inline subVI into calling VIs' checkbox is unchecked. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

VI Properties

Category Execution

Allow debugging

Reentrancy

Non-reentrant execution

Shared clone reentrant execution

Preallocated clone reentrant execution

Reentrancy settings affect memory usage, call overhead, jitter, and state maintained within the VI. Display Context help for guidance with selecting the best setting for your use case.

Priority normal priority

Preferred Execution System same as caller

Enable automatic error handling

Run when opened

Suspend when called

Clear indicators when called

Auto handle menus at launch

Inline subVI into calling VIs

OK Cancel Help

If you use a SubVI several places in your code, you typically need to use "Reentrant Execution"

Lowpass Filter

Lowpass Filter.vi Front Panel

File Edit View Project Operate Tools Window Help

15pt Application Font

y

yf

Tf [s]

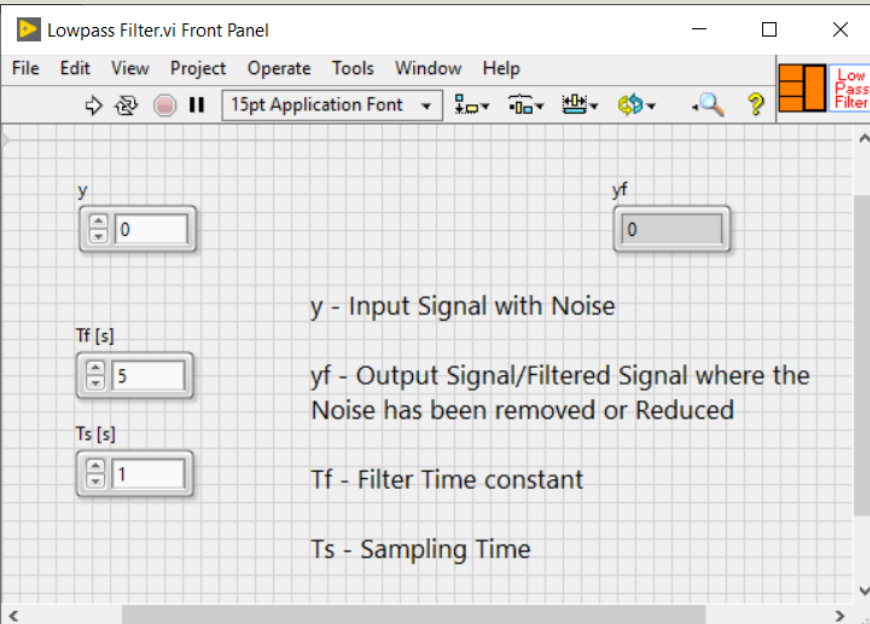
Ts [s]

y - Input Signal with Noise

yf - Output Signal/Filtered Signal where the Noise has been removed or Reduced

Tf - Filter Time constant

Ts - Sampling Time



Lowpass Filter.vi Block Diagram

File Edit View Project Operate Tools Window Help

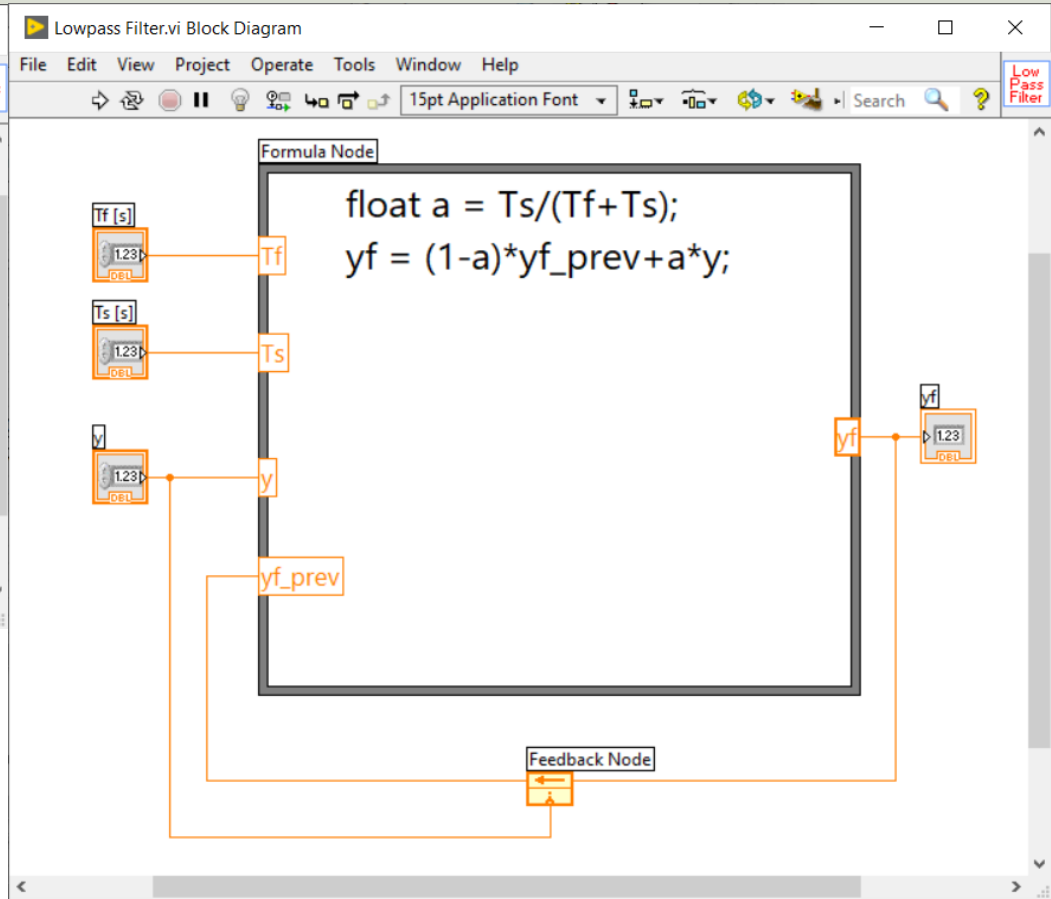
15pt Application Font

Search

Formula Node

```
float a = Ts/(Tf+Ts);  
yf = (1-a)*yf_prev+a*y;
```

Feedback Node



The block diagram illustrates the implementation of a lowpass filter using a feedback loop. The main component is a 'Formula Node' which contains the following equations:
$$a = \frac{T_s}{T_f + T_s}$$
$$y_f = (1 - a) \cdot y_{f_prev} + a \cdot y$$
 The inputs to the formula node are:

- T_f [s]: Filter Time constant
- T_s [s]: Sampling Time
- y : Input signal with noise
- y_{f_prev} : Previous filtered output signal

 The output of the formula node is y_f , which is then fed back into the 'Feedback Node' (a delay element) to provide y_{f_prev} for the next iteration. The output signal y_f is also shown with a 123-point delay element.

Write to Measurement File

Configure Write To Measurement File [Write To Measurement File]

Filename
C:\Users\hansha\OneDrive\Courses\LabVIEW in Automation\Logging Data to Text Files using LabVIEW\LabVIEW\Temperature Data.lvm

Action

Save to one file

Ask user to choose file

Ask only once

Ask each iteration

If a file already exists

Rename existing file

Use next available filename

Append to file

Overwrite file

Save to series of files (multiple files)

Settings...

File Format

Text (LVM)

Binary (TDMS)

Binary with XML Header (TDM)

Microsoft Excel (.xlsx)

Lock file for faster access

Segment Headers

One header per segment

One header only

No headers

X Value (Time) Columns

One column per channel

One column only

Empty time column

Delimiter

Tabulator

Comma

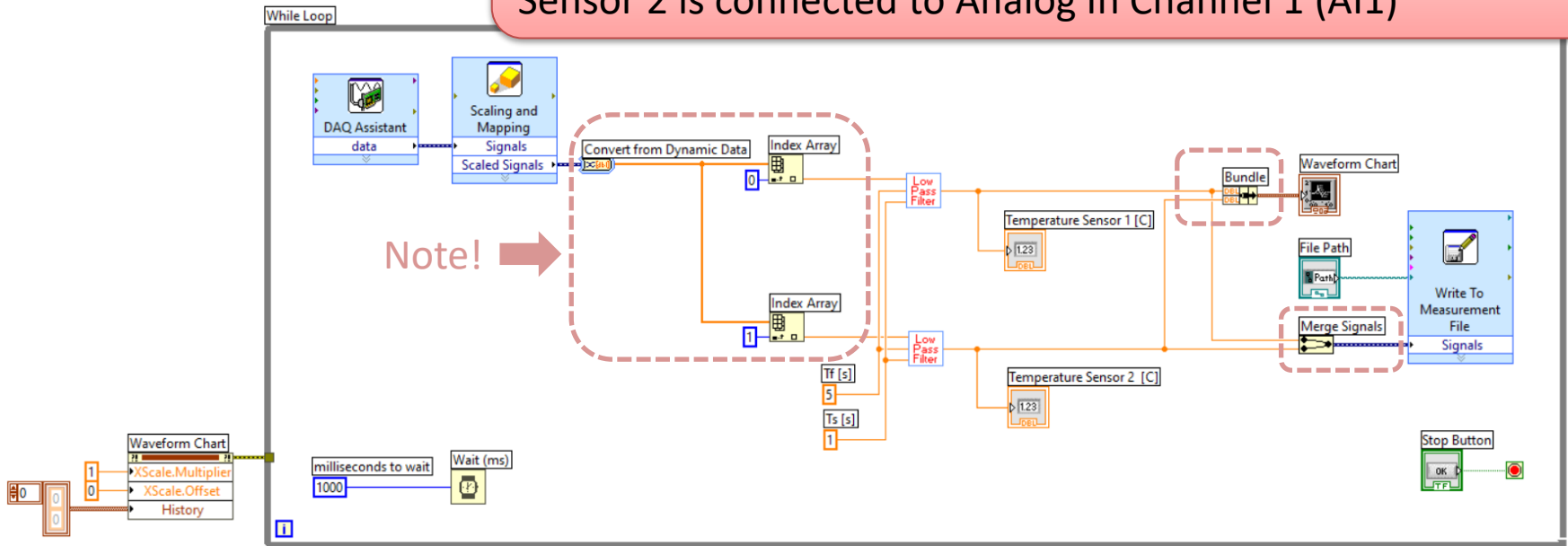
File Description

Advanced...

OK Cancel Help

Using Real DAQ Device

Here is a USB-6008 DAQ Device used. Temperature Sensor 1 is connected to Analog In Channel 0 (AI0) and Temperature Sensor 2 is connected to Analog In Channel 1 (AI1)



DAQ Assistant

Create New ...

Select the measurement type for the task.

A **task** is a collection of one or more virtual channels with timing, triggering, and other properties.

To have **multiple measurement types** within a single task, you must first create the task with one measurement type. After you create the task, click the **Add Channels** button to add a new measurement type to the task.

- Acquire Signals
 - Analog Input
 - Counter Input
 - Digital Input
 - Create New ...

Physical

Supported Physical Channels

- Dev1 (USB-6008)
 - ai0
 - ai1
 - ai2
 - ai3
 - ai4
 - ai5
 - ai6
 - ai7

<Ctrl> or <Shift> click to select multiple channels.

Here you can select multiple Channels

Create New ...

Select the measurement type for the task.

A **task** is a collection of one or more virtual channels with timing, triggering, and other properties.

To have **multiple measurement types** within a single task, you must first create the task with one measurement type. After you create the task, click the **Add Channels** button to add a new measurement type to the task.

- Acquire Signals
 - Analog Input
 - Voltage
 - Temperature
 - Strain
 - Current
 - Resistance
 - Frequency
 - Position
 - Sound Pressure
 - Acceleration
 - Velocity (IEPE)
 - Force
 - Pressure

< Back Next > Finish Cancel

< Back Next > Finish Cancel

DAQ Assistant

The screenshot shows the DAQ Assistant software interface. At the top, there is a menu bar with 'Undo', 'Redo', 'Run', 'Add Channels', and 'Remove Channels'. Below the menu bar, there are tabs for 'Express Task' and 'Connection Diagram'. The main workspace is divided into several sections:

- Channel List:** A table showing the current channels and their values.
- Channel Settings:** A panel for configuring individual channels, currently showing 'TempSensor 1' and 'TempSensor 2'. It includes an 'Add Channels' button and a 'Details' link.
- Voltage Input Setup:** A panel for configuring voltage input settings, including 'Signal Input Range' (Max: 5, Min: 0), 'Scaled Units' (Volts), 'Terminal Configuration' (Differential), and 'Custom Scaling' (<No Scale>).
- Timing Settings:** A panel for configuring acquisition mode, samples to read, and rate (Hz).

On the right side, there is a 'Measuring Voltage' help panel with text explaining DC and AC voltages. At the bottom, there are 'OK' and 'Cancel' buttons.

Channel	Value
TempSensor1	0
TempSensor2	0

Table Display Type

Configuration Triggering Advanced Timing Logging

Channel Settings

TempSensor 1
TempSensor 2

Click the Add Channels button (+) to add more channels to the task.

Voltage Input Setup

Settings

Signal Input Range

Max: 5
Min: 0

Scaled Units: Volts

Terminal Configuration: Differential

Custom Scaling: <No Scale>

Timing Settings

Acquisition Mode: 1 Sample (On Demand) Samples to Read: 1k Rate (Hz): 1k

Measuring Voltage

Most measurement devices are designed for measuring, or reading, voltage. Two common voltage measurements are DC and AC.

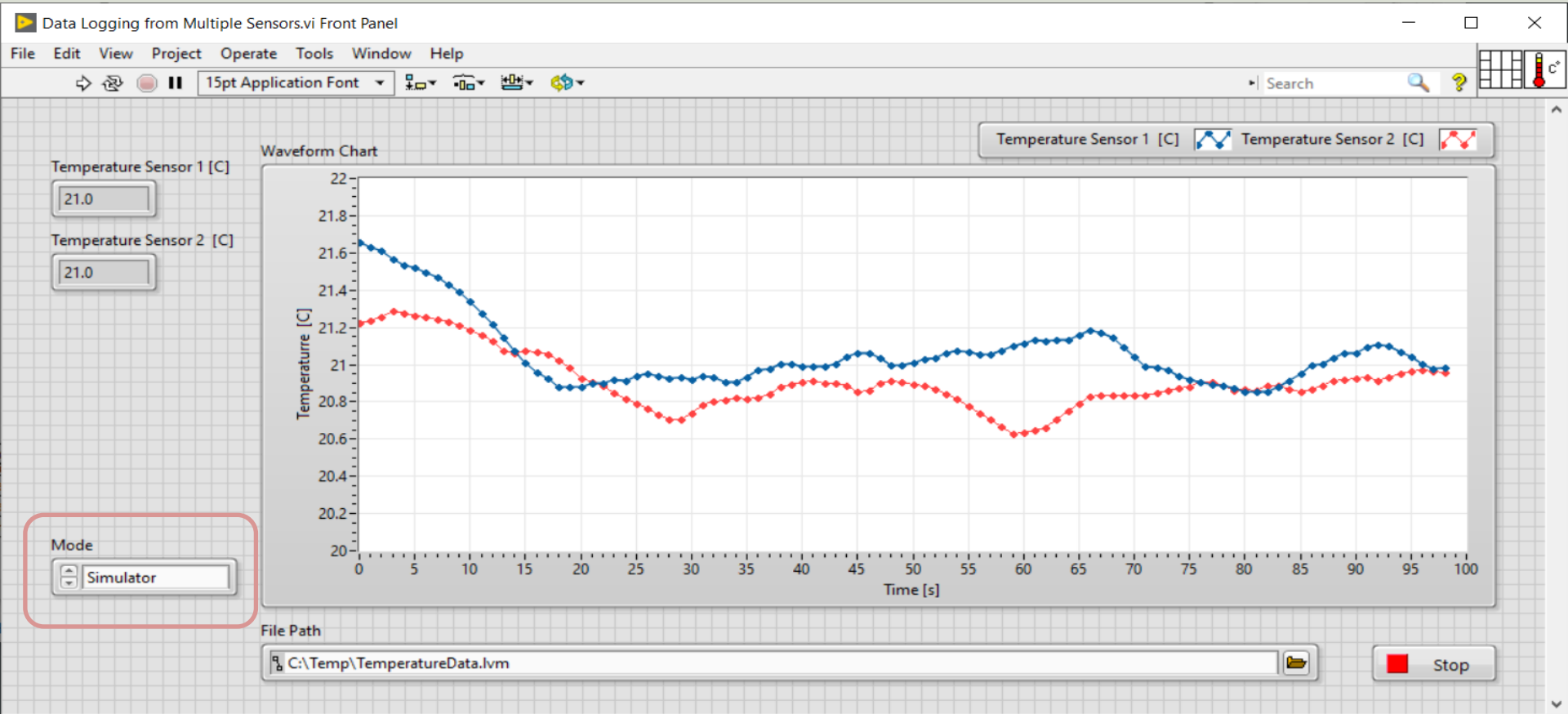
DC voltages are useful for measuring phenomena that change slowly with time, such as temperature, pressure, or strain.

AC voltages, on the other hand, are waveforms that constantly increase, decrease, and reverse polarity. Most powerlines deliver AC voltage.

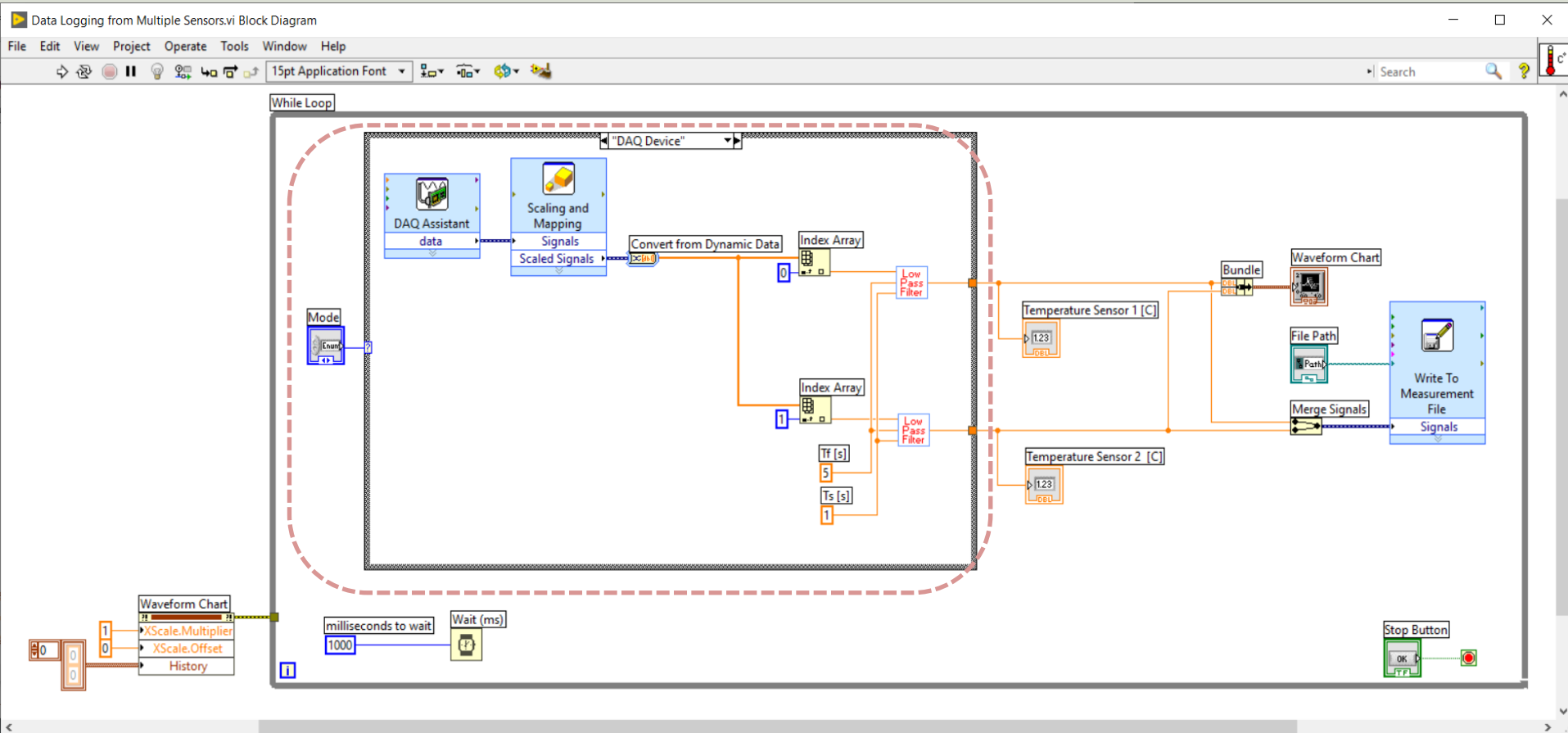
This is the list of virtual channels. Right-click a virtual channel to change the physical channel associated with it. If an exclamation point (!) appears next to a global virtual channel, the channel has been deleted.

OK Cancel

Final LabVIEW Application



Final LabVIEW Application



Logged Data

TemperatureData.lvm - Notepad

File	Edit	Format	View	Help
0.000000	20.967908	21.483000		
1.001163	20.968071	21.452208		
2.000908	20.990304	21.440139		
3.000337	21.030603	21.437771		
4.000916	21.032733	21.403616		
5.001221	21.045383	21.367044		
6.001272	21.041305	21.310924		
7.001723	21.014337	21.237776		
8.001851	21.012491	21.181544		
9.001402	21.033078	21.158482		
10.001266	21.069751	21.148573		
11.003134	21.117899	21.111289		
12.003939	21.121751	21.100976		
13.005329	21.144116	21.111218		
14.007066	21.181790	21.101876		
15.008220	21.187309	21.071323		
16.007552	21.175946	21.057514		
17.009147	21.178308	21.026114		
18.008528	21.156333	21.011413		
19.009591	21.138625	20.991855		
20.009260	21.117530	20.973851		
21.009494	21.081788	20.974295		
22.010872	21.077449	20.988406		
23.012124	21.065772	21.015741		
24.011428	21.047252	21.041307		
25.012150	21.046908	21.079747		
26.012938	21.068329	21.121360		
27.013172	21.096087	21.137585		
28.014173	21.101500	21.165631		
29.013848	21.129122	21.201625		
30.015204	21.138425	21.211084		
31.015820	21.157356	21.200788		
32.015516	21.185103	21.169901		
33.017083	21.219731	21.143720		
34.016951	21.256801	21.130904		
35.016758	21.261623	21.112903		
36.018091	21.267781	21.105305		
37.017422	21.278334	21.106600		

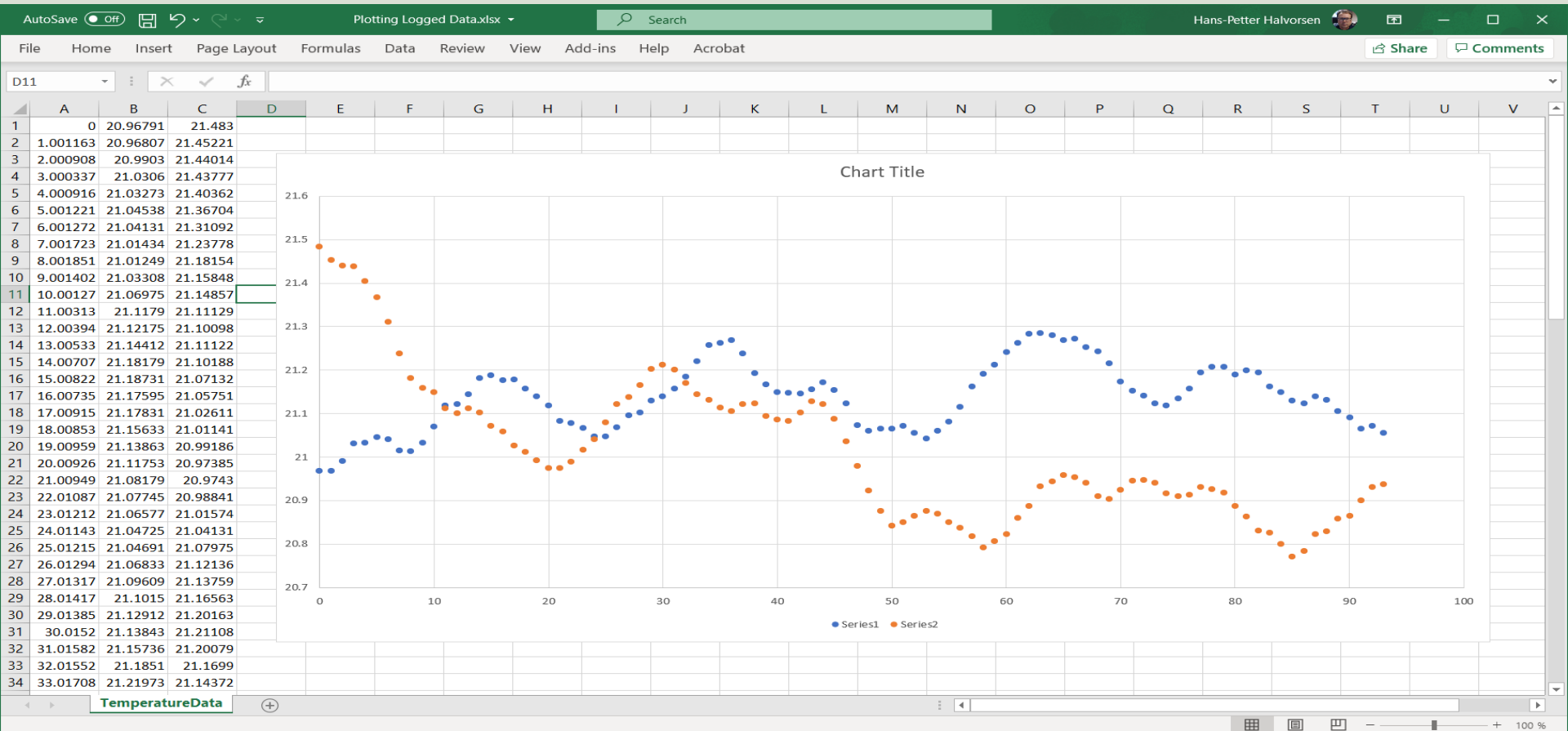
Ln 1, Col 1 100% Windows (CRLF) UTF-8

Time

Temperature Sensor 1

Temperature Sensor 2

Excel



Summary

- In this tutorial we showed how to read data from 2 sensors using a DAQ device and saving the data to a Text File in LabVIEW
- Have more than 2 Sensors?
- We can easily extend the application if we have more than 2 sensors
- See also this YouTube Video:
 - Data Logging and Monitoring LabVIEW Project:
<https://youtu.be/FFnvYu7jjeI>

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

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

