https://www.halvorsen.blog

DAQ and OPC System

Hans-Petter Halvorsen

Background

- You work as a System Engineer for a System Engineering Company.
- An Industrial Production Company has announced a competition between several selected System Engineering companies to perform a preliminary Project.
- Your assignment is to develop a Prototype/PoC of a DAQ system where OPC will be used as the main Communication Standard.
- In the PoC, basic Temperature Sensors will be used to demonstrate the principles.
- The OPC System should be implemented both using OPC DA and OPC UA using different Programming Languages.
- To create proper and user-friendly GUI/HMI is an important part of the Prototype.
- The delivery is a Technical Report where you shall give an overview of the entire system made, including the Methods used and the Results archived.
- The PoC and the Report will be an important foundation for decision making within the company when it comes to the final implementation of the system sometime in the future. Note! Multiple System Engineering companies have been given this opportunity, so it is important that you “Add Value” and stand out compared to the others in order to be selected as the final contractor.

System Requirements

- **OPC in LabVIEW**

  OPC DA LabVIEW

  – Create OPC DA Client Application in LabVIEW that reads data from one or more Temperature Sensors and write the Data to the Matrikon OPC DA Server.

    - The Application should at least include a features for Writing Data to a File and a Chart

    - Make sure to open the Data from the File and plot it using MS Excel

  – Create OPC DA Client Application in LabVIEW that reads the Temperature Data from the Matrikon OPC DA Server

  OPC UA LabVIEW

  – Create an OPC UA Server Application in LabVIEW

  – Create OPC UA Client Application in LabVIEW that reads data from the Temperature Sensor(s) and writes the Data to the LabVIEW OPC UA Server

  – Create OPC UA Client Application in LabVIEW that reads the Temperature Data from LabVIEW OPC UA Server

- **OPC DA/UA in Visual Studio/C#**

  – Create OPC Application(s) using Visual Studio/C#. It can be OPC DA or OPC UA

- **OPC DA/UA in MATLAB**

  – Create a MATLAB Script for Writing Data to an OPC Server. Use, e.g., a While/For Loop.

  – Create a MATLAB Script for Reading Data to an OPC Server. Create a Plot, etc.

- OPC in a Network

  – Using OPC in a Network. Try to Install the Applications on different computers in a Network and see if the communication between them works. Discuss the results.

These are the complete requirements for the assignment. The rest of this document contains different DAQ and OPC resources like additional information, code examples, tips and tricks, step-by step instructions, etc. that you can use at your own discretion.

OPC Resources

Hans-Petter Halvorsen

Table of Contents

Table of Contents

Introduction

Hans-Petter Halvorsen

Table of Contents

Learning Goals

- Learn key concepts within OPC
- Learn practical skills and implementation of OPC
- Learn more Programming (LabVIEW, C#, etc.)
- Learn about Hardware-Software Interactions
- Learn Practical Skills and Implementations in general
- Learn Software Installation in general, which can be cumbersome with many pitfalls
- Learn to use and create Software in general

Software

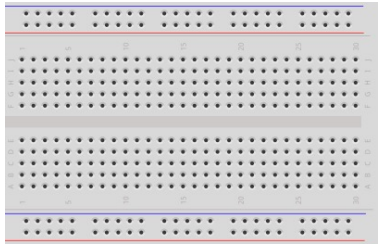
- LabVIEW
  - DAQmx Driver Software
  - LabVIEW OPC UA Toolkit
- MatrikonOPC Simulation Server
- OPC UA server Simulator
- MATLAB
  - MATLAB Industrial Communication Toolbox
- Visual Studio
  - Measurement Studio
- OPC Tunneller Software
  - OPC Tunneller from MatrikonOPC
    or
  - Cogent DataHub Tunnelling Software

Hardware

NI USB-TC01 Thermocouple Measurement Device

USB-6008 I/O Module

If you don't have the Hardware, you may create a Simulator instead, or use another Hardware if you have



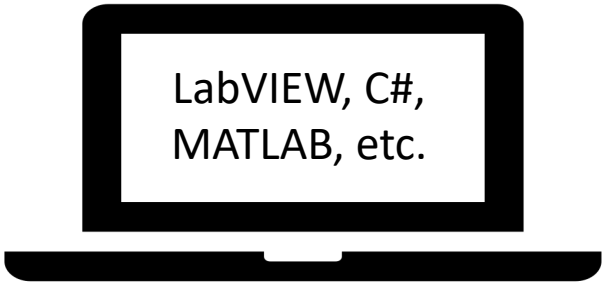
Breadboard

TMP36

Thermistor

Switch

Your Personal Computer



The teacher have not done all the Tasks in detail, so he may not have all the answers! That's how it is in real life also!

Very often it works on one computer but not on another. You may have other versions of the software, you may have installed it in the wrong order, etc...
In these cases Google is your best friend!

The Teacher dont have all the answers (very few actually ☹)!! Sometimes you just need to “Google” in order to solve your problems, Collaborate with other Students, etc. Thats how you Learn!

Troubleshooting & Debugging

Use the **Debugging Tools** in your Programming IDE.
Visual Studio, LabVIEW, etc. have great Debugging Tools! Use them!!

“Google It”!

You probably will find the answer on the Internet

Google

Another person in the world probably had a similar problem

Use available Resources such as User Guides, Datasheets, Textbooks, Tutorials, Examples, Tips & Tricks, etc.

My System is not Working??

Multimeter, etc.

Use Microsoft Teams

Check your electric circuit, electrical cables, DAQ device, etc. Check if the wires from/to the DAQ device is correct. Are you using the same I/O Channel in your Software as the wiring suggest? etc.

Lab Assignment Guidelines

- Make sure to read the whole assignment before you start to solve any of the problems.
- If you miss assumptions for solving some of the problems, you may define proper assumptions yourself.
- The Tasks described in the Assignment are somewhat loosely defined and more like guidelines, so feel free to interpret the Tasks in your own way with a personalized touch.
- Feel free to Explore! Make sure to Add Value and Creativity to your Applications!
- Try to add some extra value and be creative compared to the simplified examples given by me, in that way you learn so much more.

Lab Assignment Guidelines

- Think about the Lab Assignment as a small real-life industrial Project, and not a set of tasks or exercises.
- What does the company that hire you expect from you when you deliver this project? What kind of Quality is expected?
- Try to see your work in a larger context than just a Lab Assignment or a set of exercises.
- Try to see the big picture. The tasks within the assignment are just just small building blocks that ends up with a fully working system.
- It is recommended that you make a Work Plan and a System Sketch that gives you an overview of what YOU should do

Lab Work Requirements

- Make sure to see the “big picture” – you don’t need to document every single step you have made. Focus on what’s important.
- Your GUIs is important! - make sure to make them user friendly and intuitive. You create this on behalf of someone that are going to use your applications.
- Make sure to always add units in your GUI, charts, documentation, etc.
- Presenting values with 4+ decimals makes no sense! E.g., a temperature sensor is not that accurate. You can easily change number of decimals that you present in your GUI in LabVIEW, C#, etc.
- The quality of the LabVIEW code is important. Make sure to use "straight lines" in your LabVIEW code, etc. The code should also flow from left to right, not opposite direction. You create this on behalf of someone that are going to use your applications. Neat code makes it easier to develop, maintain, find code errors, etc.
- In general, make sure that you take some pride in your applications and the work that you do. It's not about getting finished as soon as possible. The mission is to learn as much as possible within a given timeframe. Try to change the mindset.
- To improve the LabVIEW code, please see this video: LabVIEW Applications using State Machine: https://youtu.be/-b9St8wNhpQ

DAQ with TC-01 Thermocouple

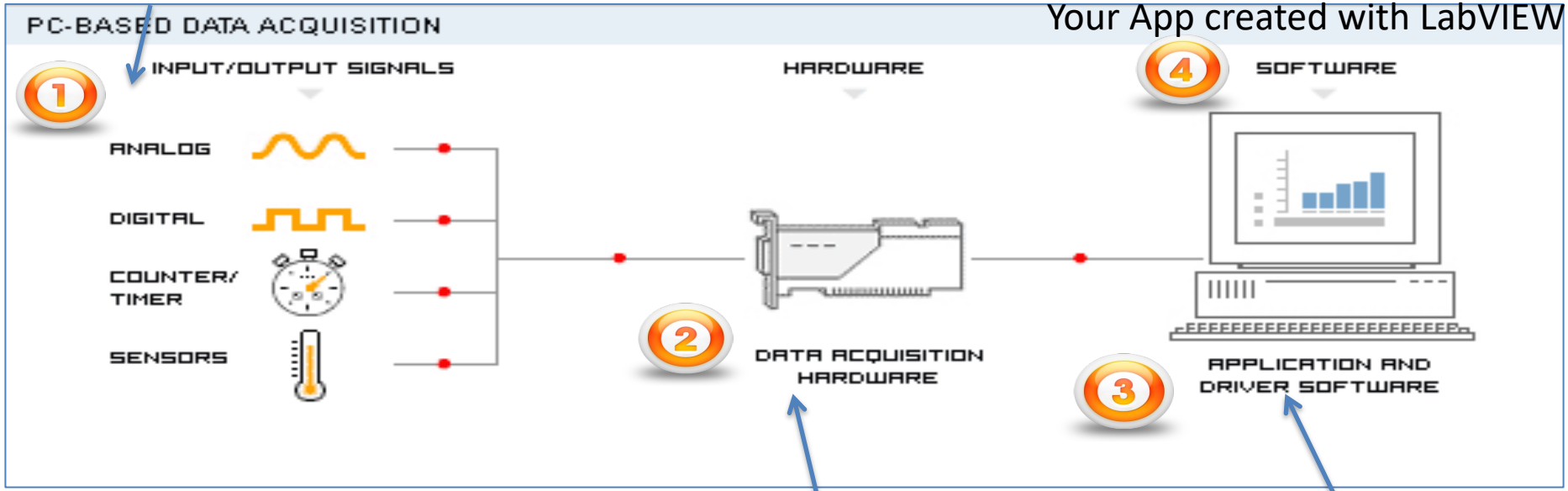
Hans-Petter Halvorsen

Table of Contents

Data Acquisition (DAQ)

Sensors, etc.

Your App created with LabVIEW



A DAQ System consists of 4 parts:

1. Physical input/output signals, sensors
2. DAQ device/hardware
3. Driver software
4. Your software application (Application software)

NI TC-01 Thermocouple Device

NI DAQmx Driver

TC-01 Thermocouple Sensor

TC-01 Thermocouple Temperature Sensor is made by NI, the same company that develop LabVIEW

Sample Rate: 4 Samples/S

https://www.ni.com/en-no/support/model.usb-tc01.html

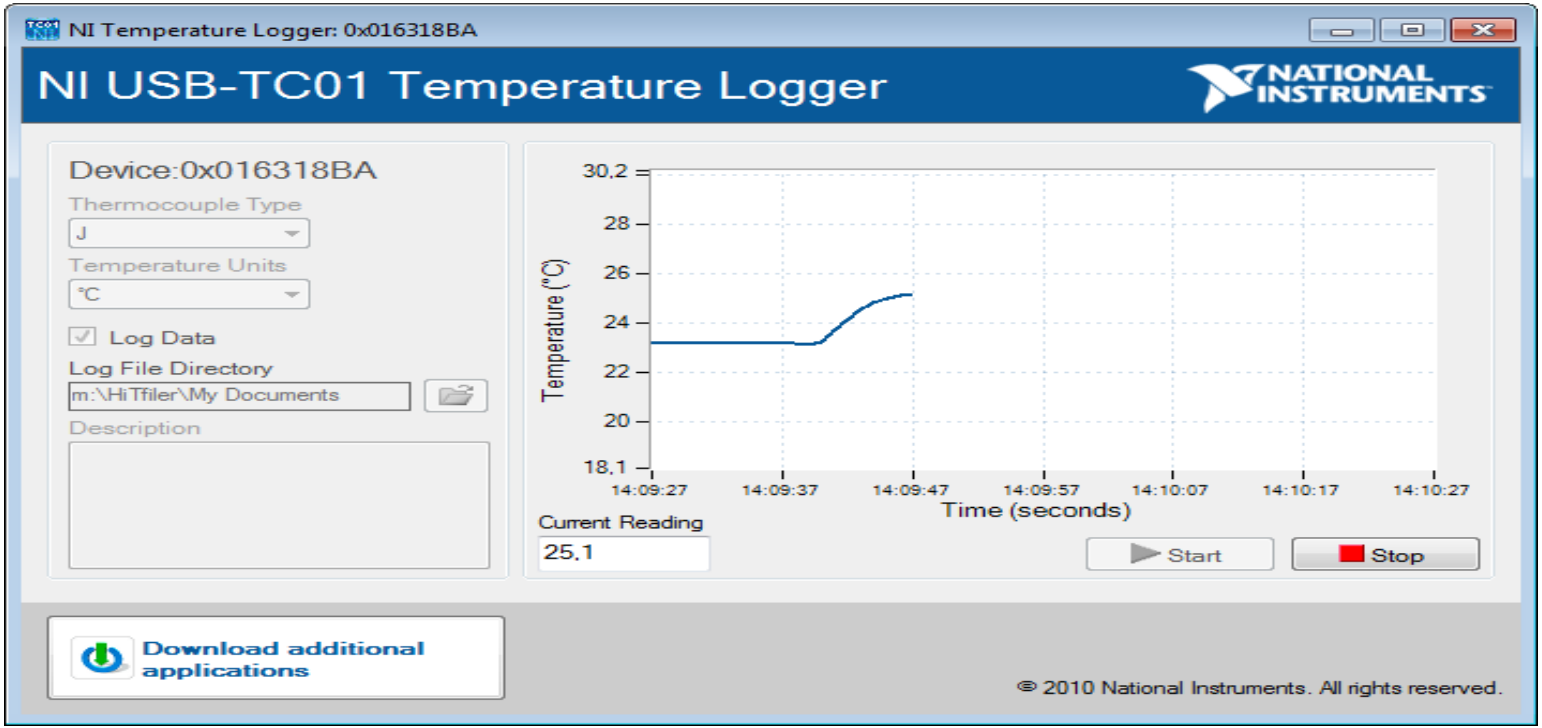
Datasheet: https://www.ni.com/pdf/manuals/374918b.pdf

Getting Started with TC-01

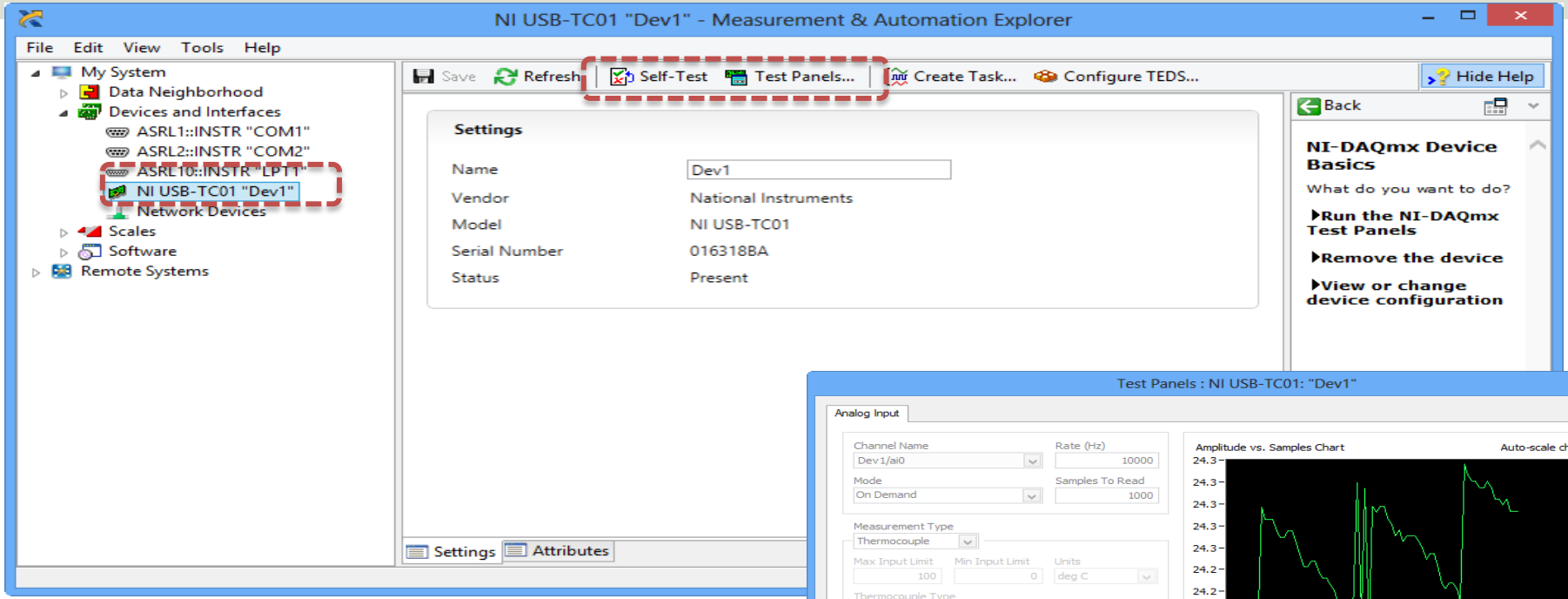
The following window should pop up automatically when you plug in your NI USB-TC01 device in your USB port (if not, select "**TC01Launcher.exe**"):

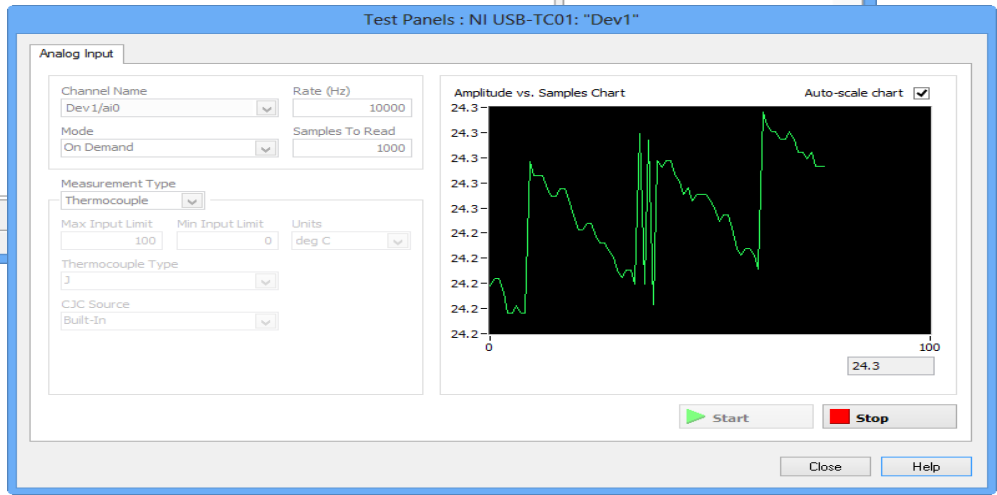
Built-in Temperature Logger

The TC-01 comes with a built-in Temperature Logger (No Driver or programming needed):



Measurement & Automation Explorer (MAX)

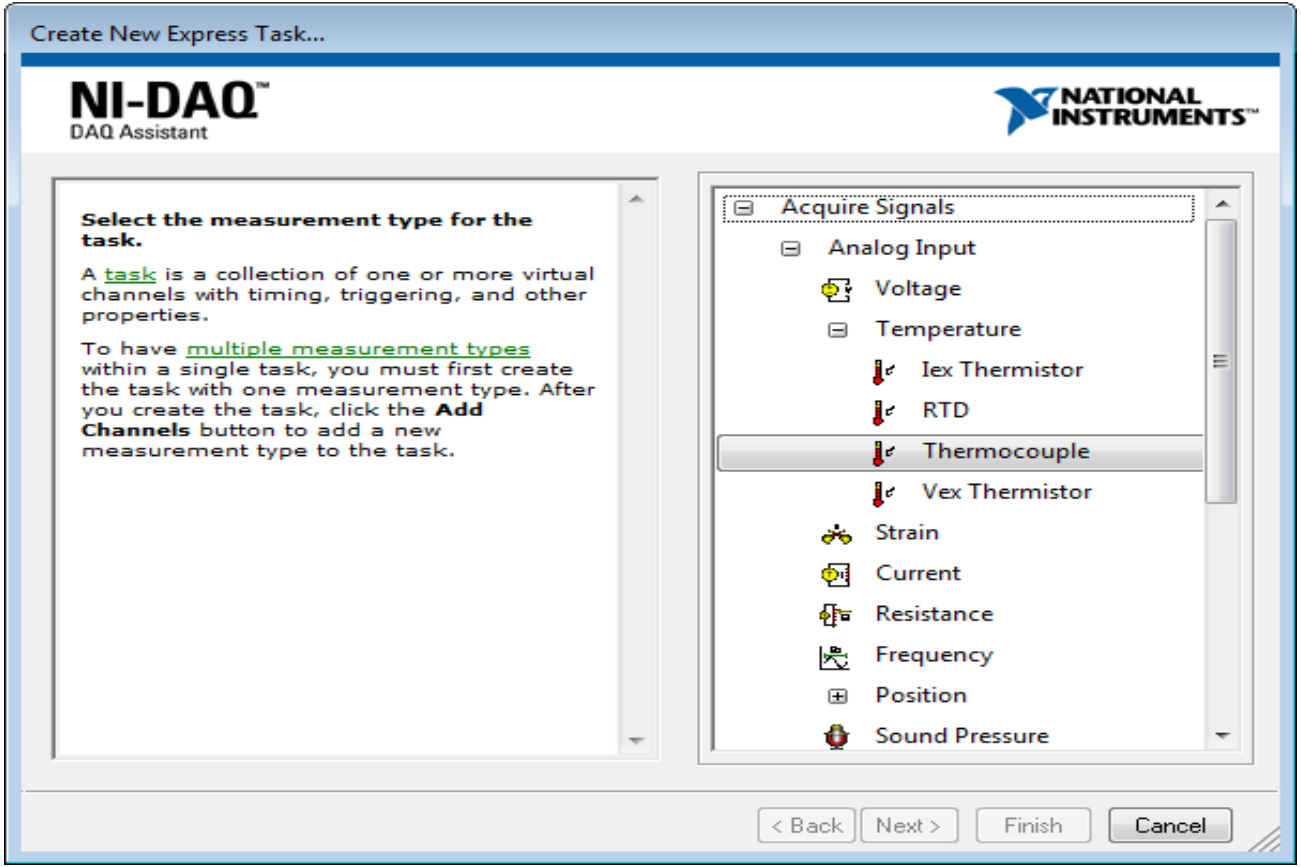




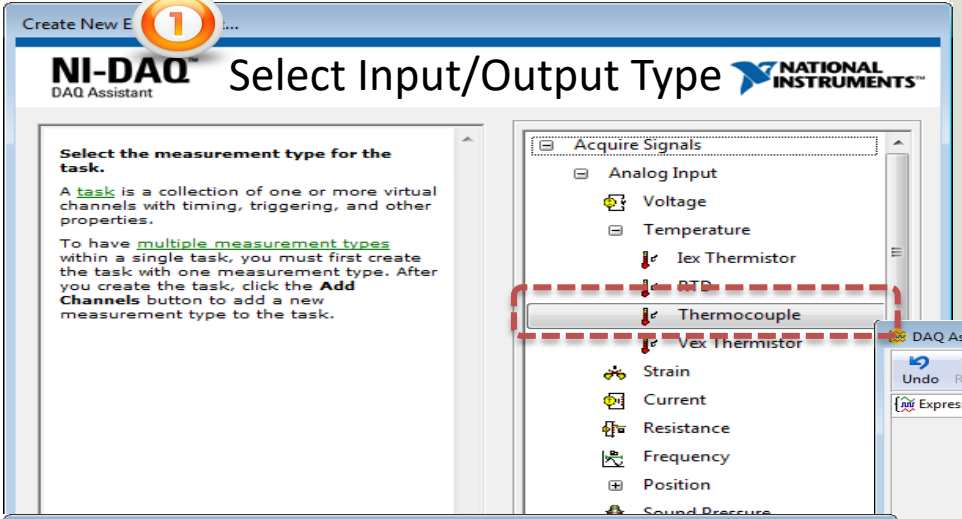
Make sure that your device can be located in MAX. Run a “Self-Test” and use the “Test Panels” to make sure the device works properly.

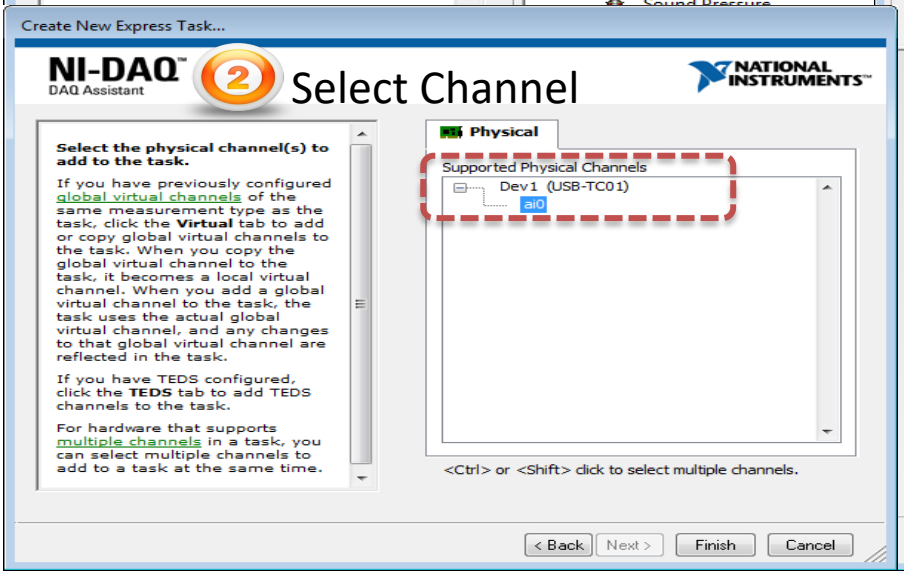
LabVIEW DAQ Assistant

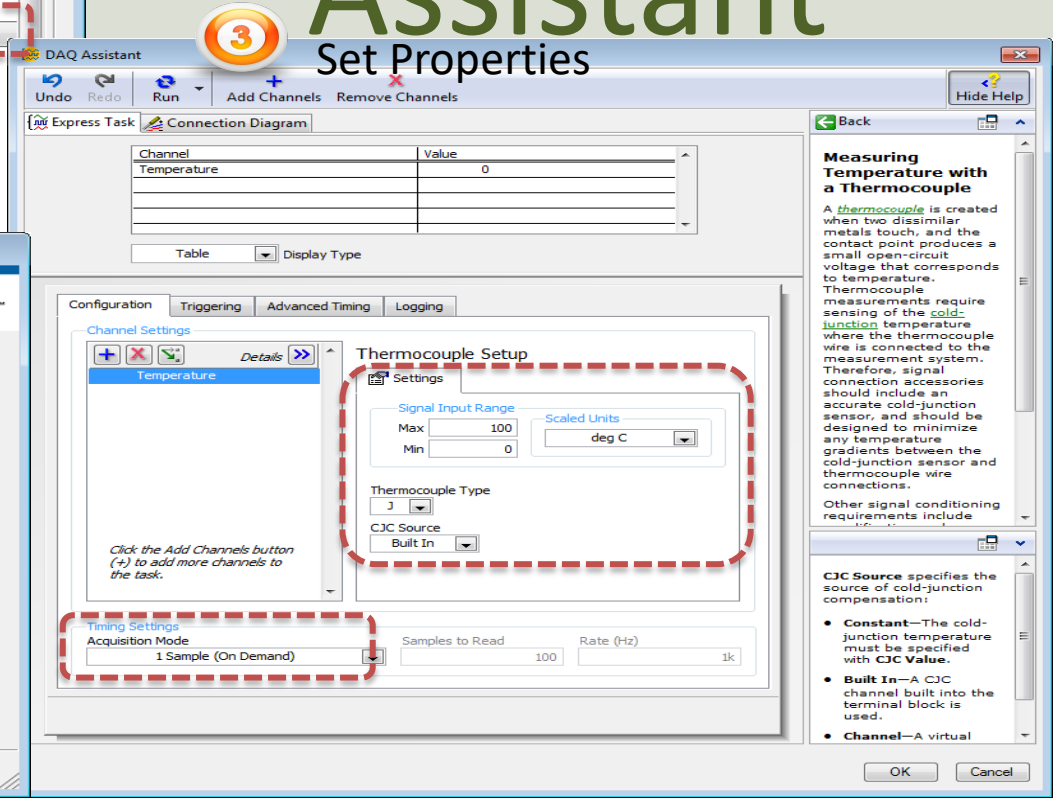
When you place the **DAQ Assistant** on the Block Diagram, a Wizard automatically pops up where you configure what you want to do, i.e., if you want to Read or Write Data, Analog or Digital signals, which channel you want to use, etc.



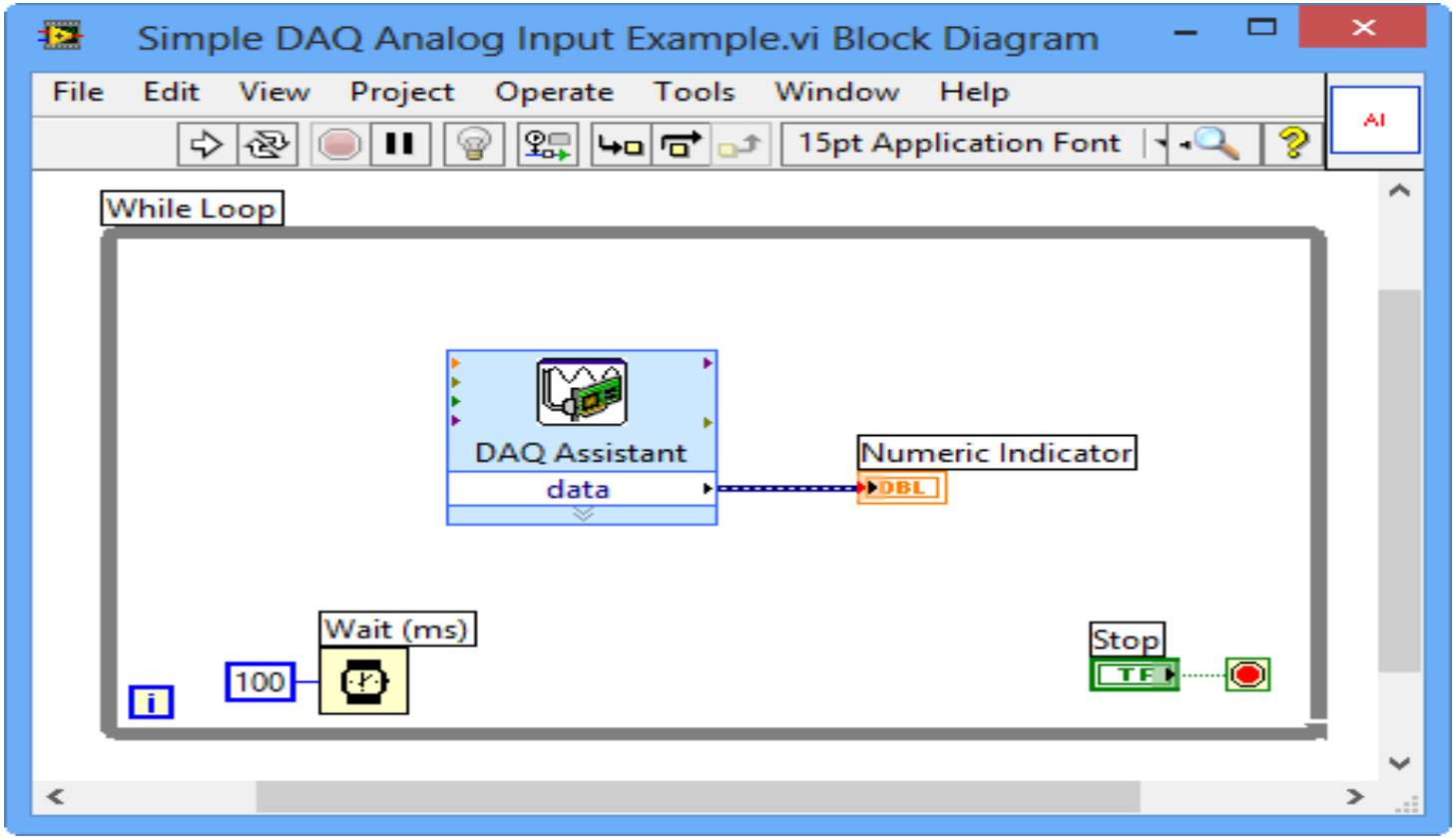
LabVIEW DAQ Assistant







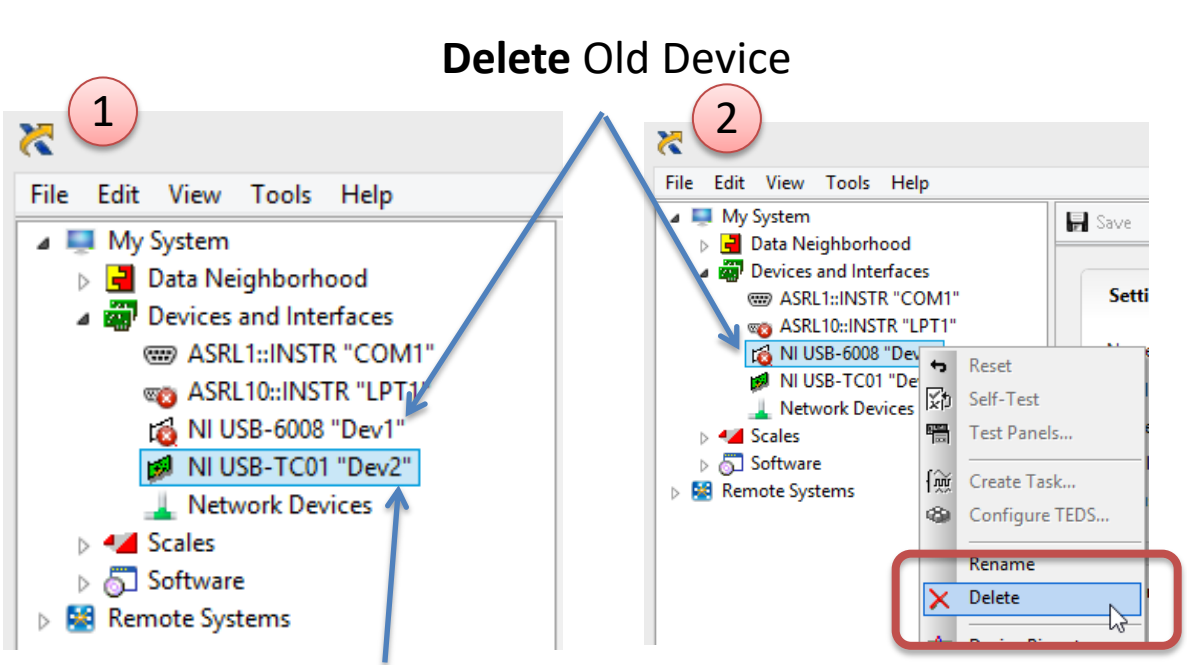
Read Data from TC-01 Device



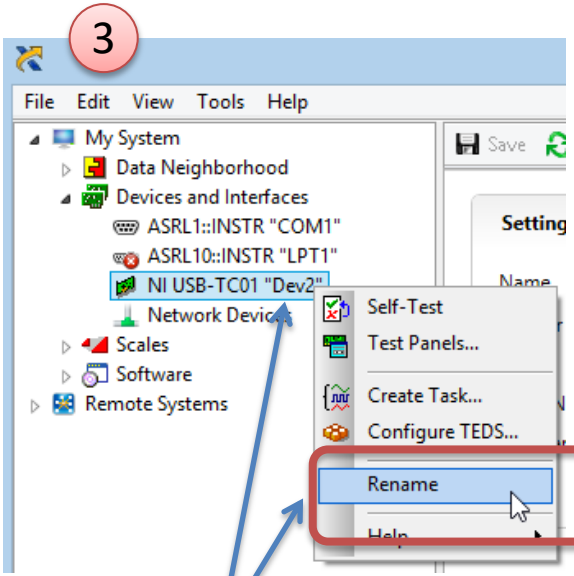
Not working after you got a new Device?

Solution, Alt 1: Open **MAX** (Measurement & Automation Explorer) in order to Fix-it!

Delete Old Device



New Device

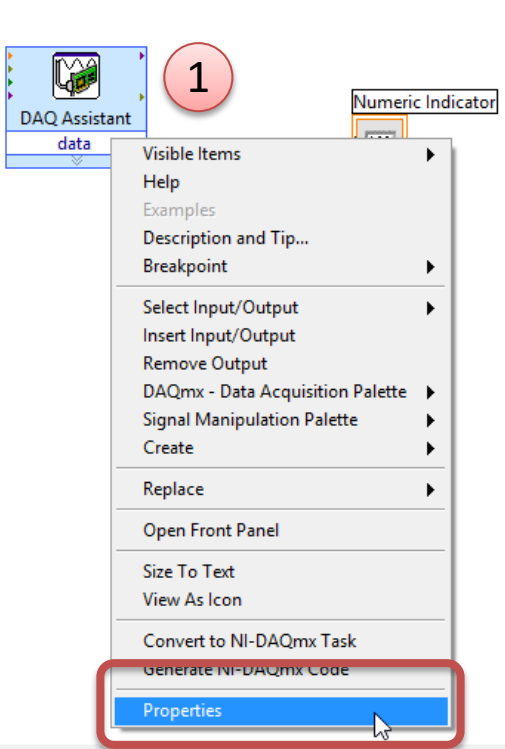


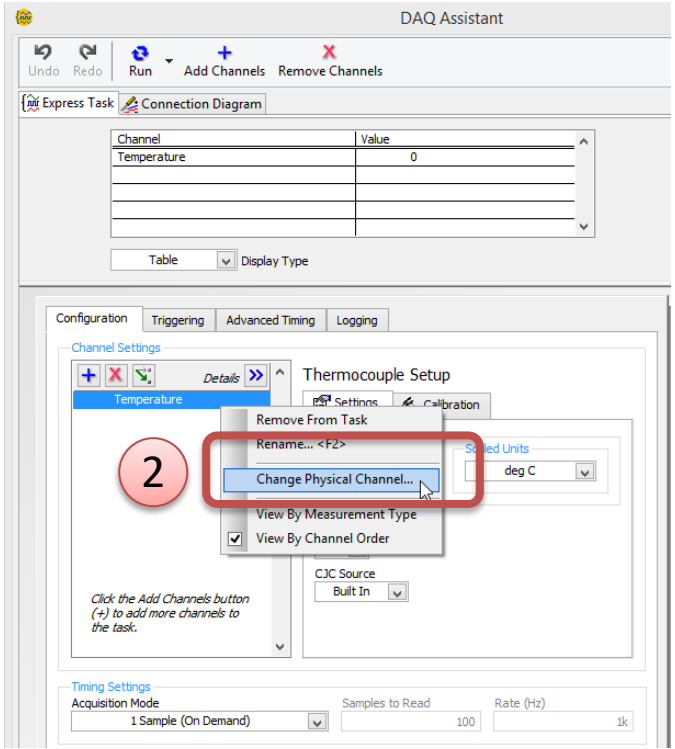
Rename the New Device with the same Name as the Old one

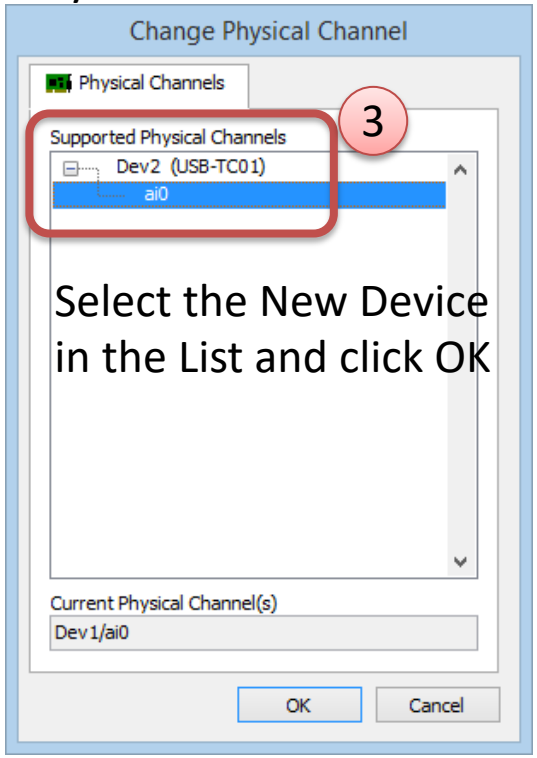
Not working after you got a new Device?

Solution, Alt 2: Change the Settings in the DAQ Assistant in your LabVIEW Application

Right-click and select “Change Physical Channel”







DAQ with USB-6008

Hans-Petter Halvorsen

Table of Contents

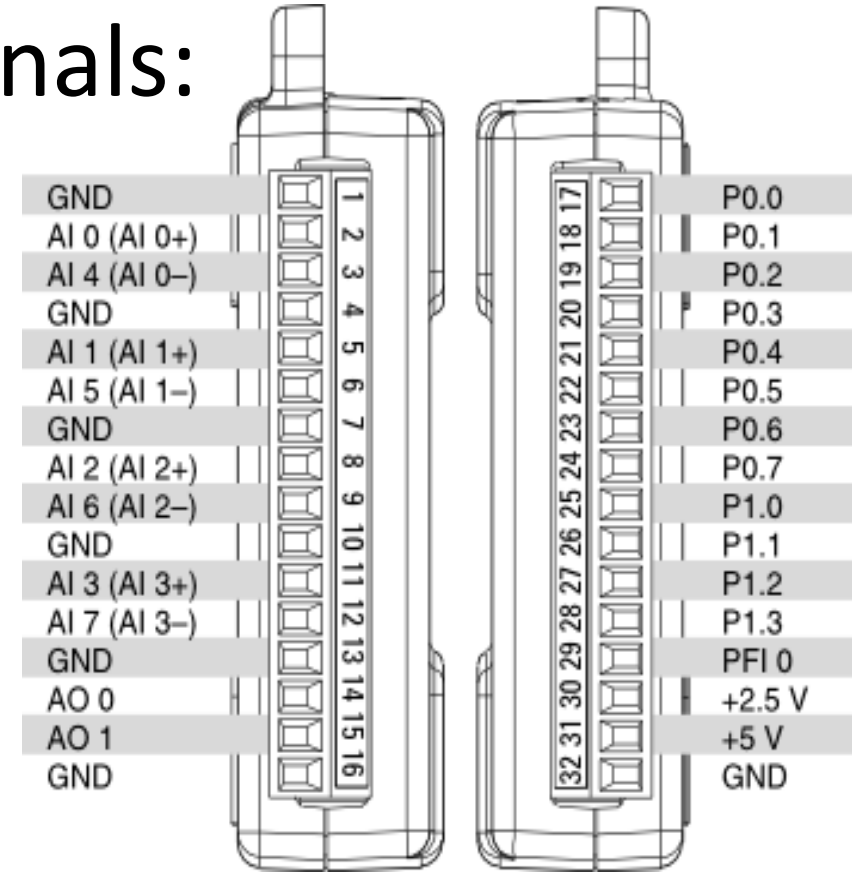
USB-6008

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

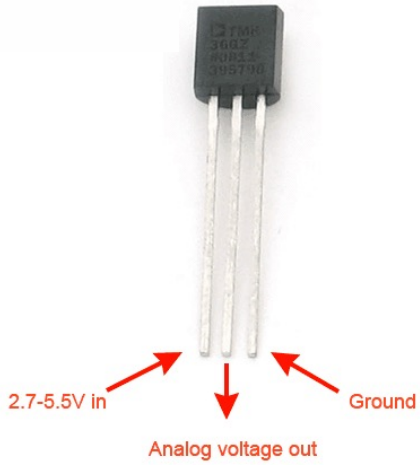
USB-6008

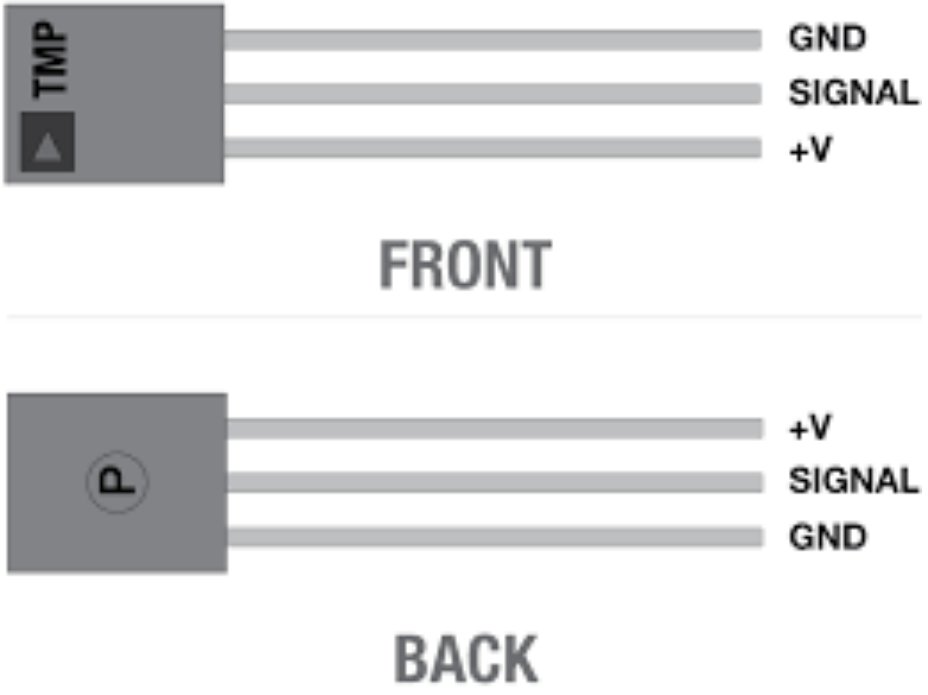
4 different types of Signals:

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



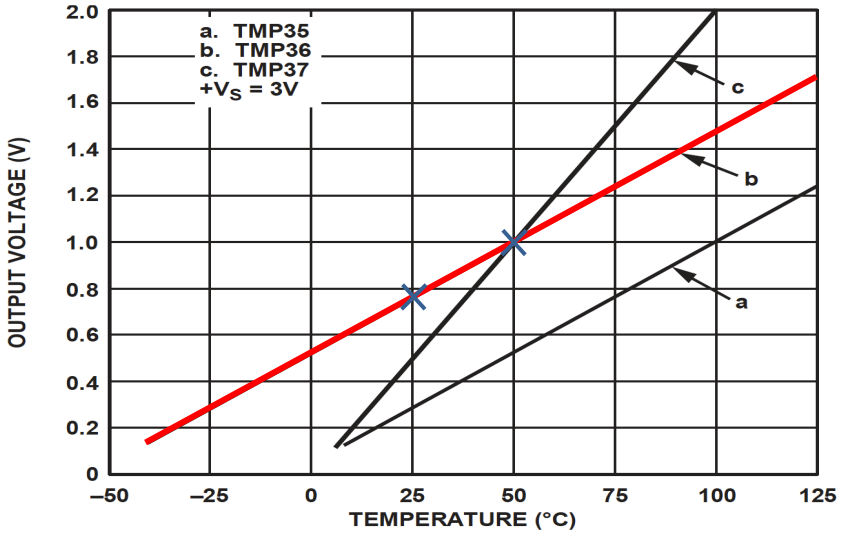
TMP36





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

Linear Scaling



This gives:

$$y - 25 = \frac{50 - 25}{1 - 0.75}(x - 0.75)$$

Then we get the following formula:

$$y = 100x - 50$$

Convert form Voltage (V) to degrees Celsius
From the Datasheet we have:

$$(x_1, y_1) = (0.75V, 25°C)$$
$$(x_2, y_2) = (1V, 50°C)$$

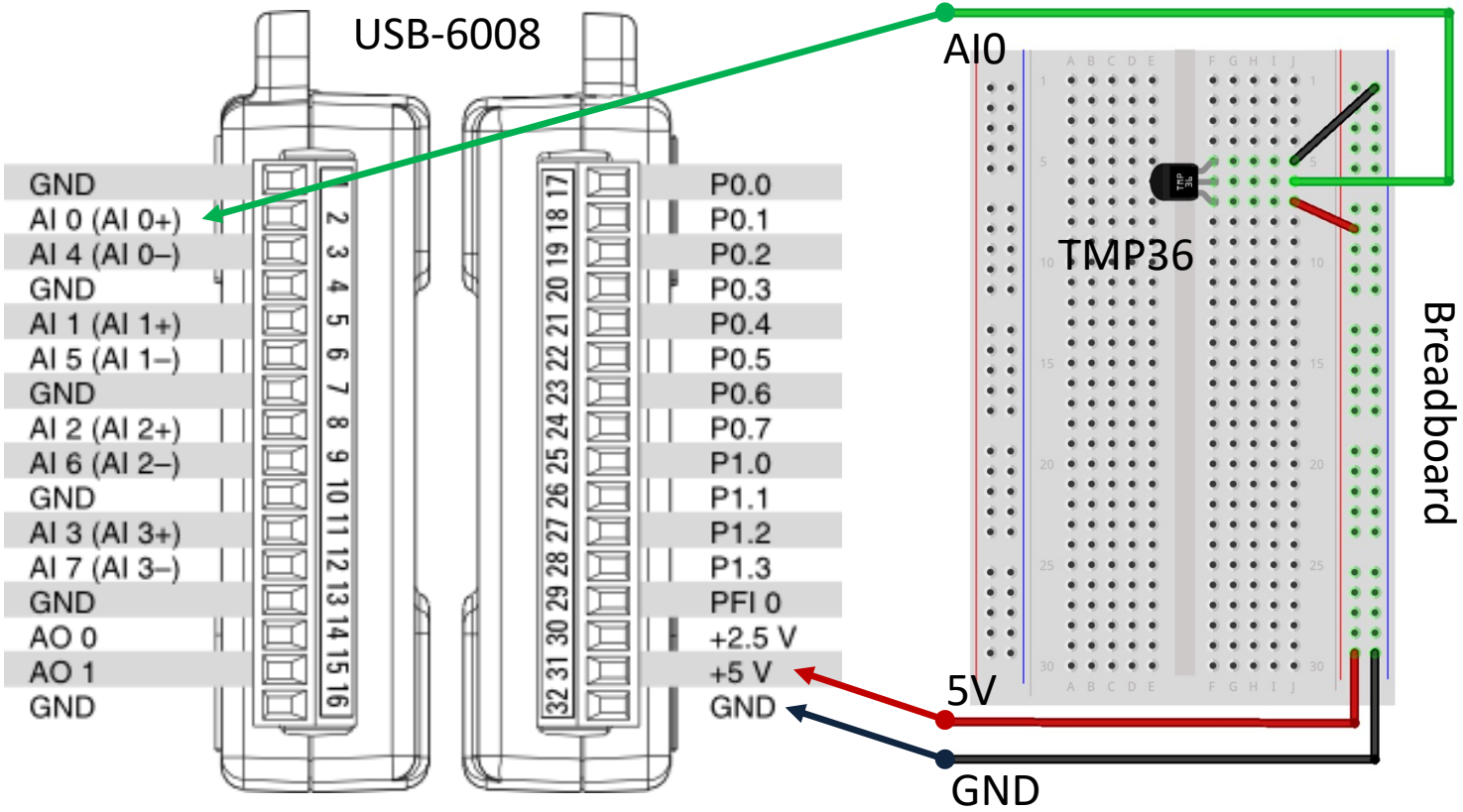
There is a linear relationship between Voltage and degrees Celsius:

$$y = ax + b$$

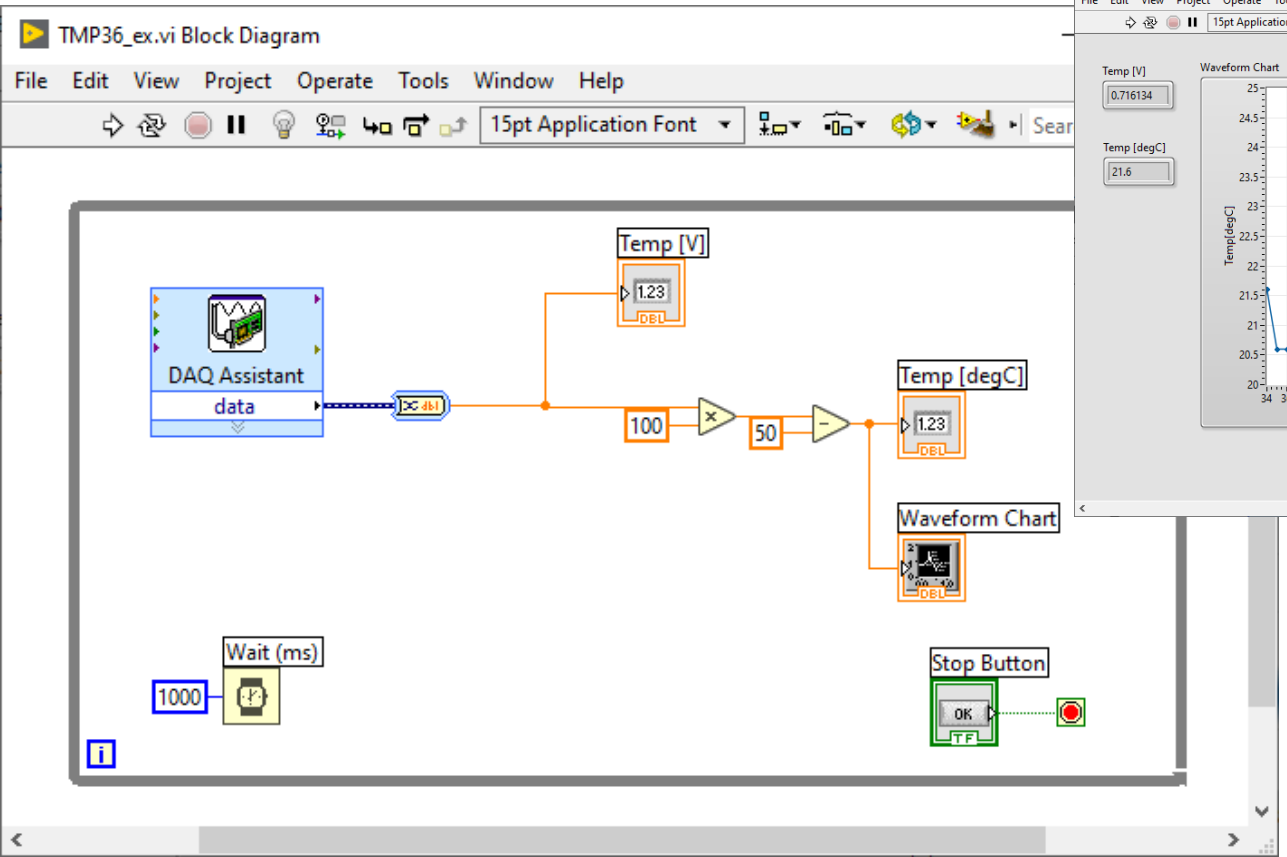
We can find a and b using the following known formula:

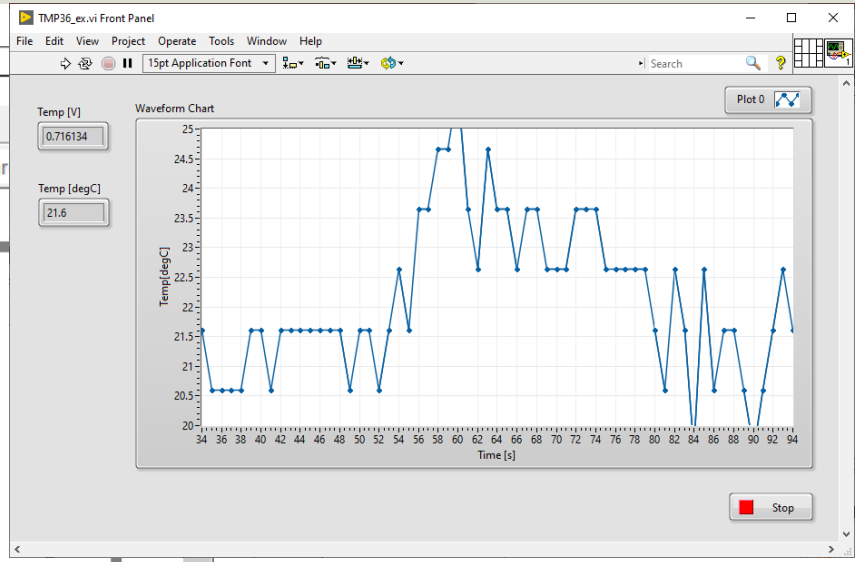
$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

Wiring


USB-6008
GND
AI 0 (AI 0+)
AI 4 (AI 0–)
GND
AI 1 (AI 1+)
AI 5 (AI 1–)
GND
AI 2 (AI 2+)
AI 6 (AI 2–)
GND
AI 3 (AI 3+)
AI 7 (AI 3–)
GND
AO 0
AO 1
GND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P1.0
P1.1
P1.2
P1.3
PFI 0
+2.5 V
+5 V
GND
AI0
TMP36
5V
GND
Breadboard

Plotting Example





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

$A = 0.001129148, B = 0.000234125$ *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:

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

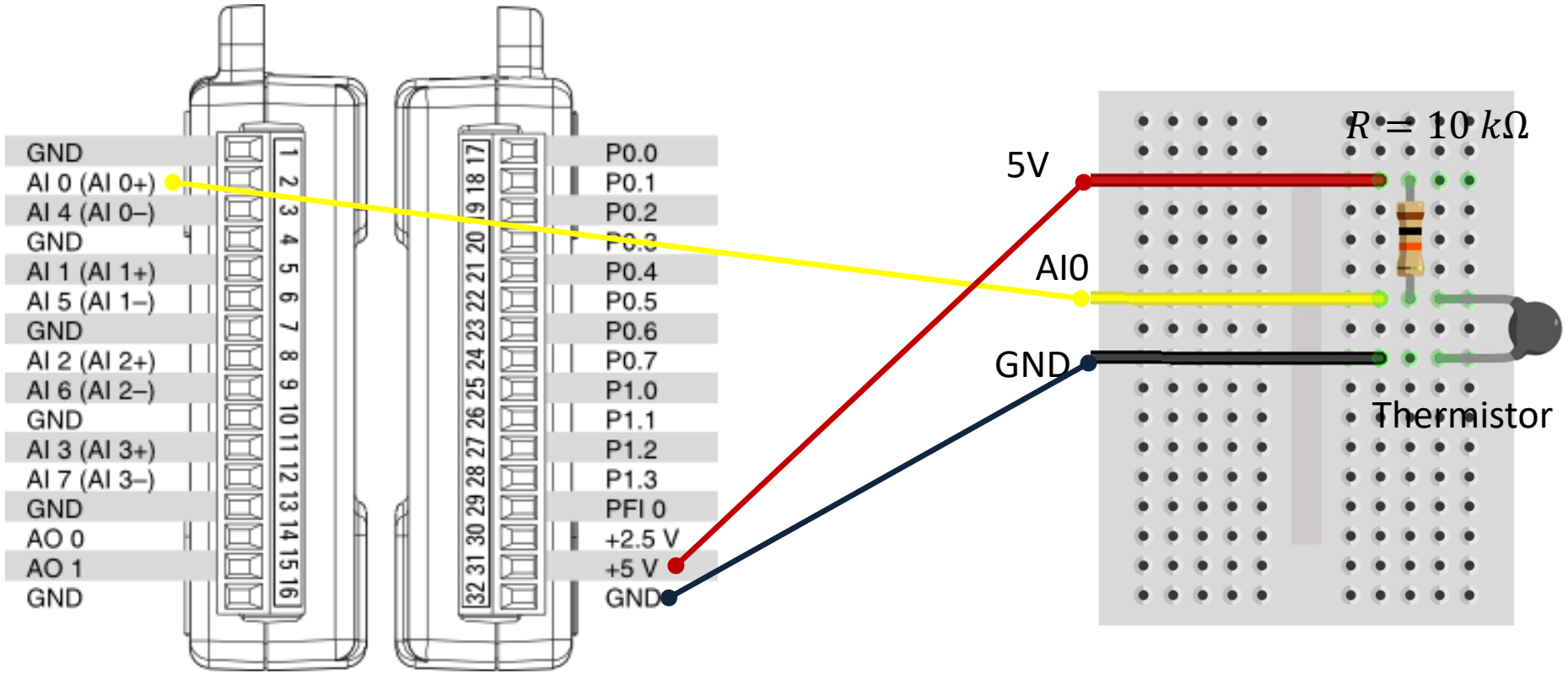
Where the Temperature T_K is in Kelvin
$A, B\ and\ C$ are constants

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

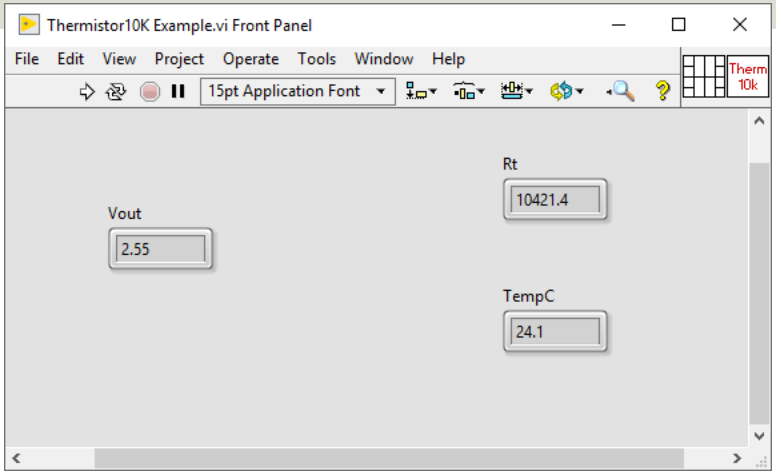
The Temperature in degrees Celsius will then be:

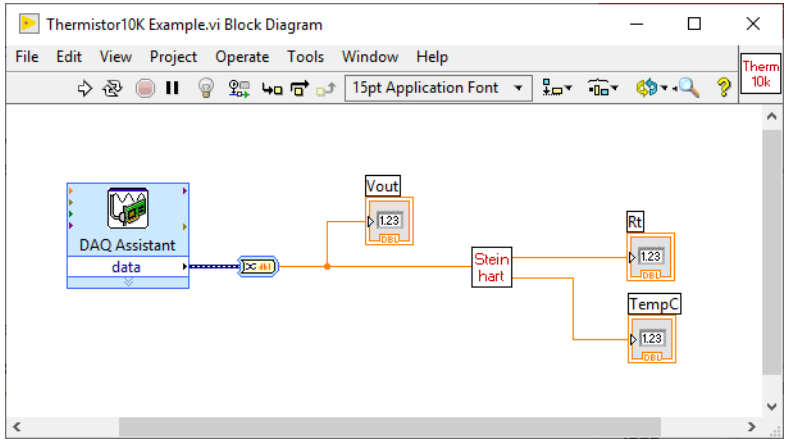
$$T_C = T_K - 273.15$$

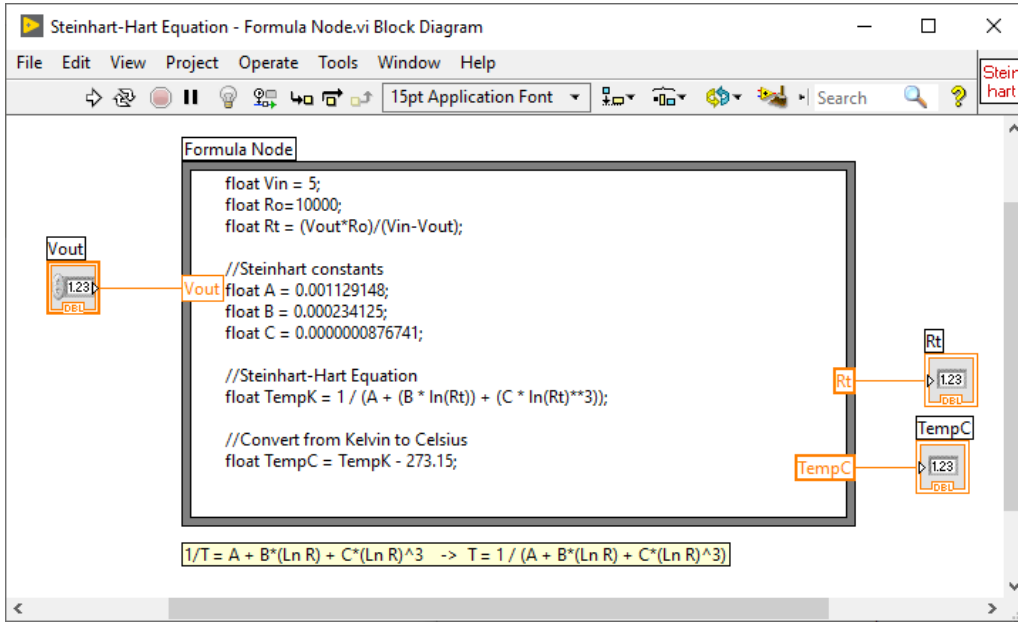
Wiring


GND
AI 0 (AI 0+)
AI 4 (AI 0–)
GND
AI 1 (AI 1+)
AI 5 (AI 1–)
GND
AI 2 (AI 2+)
AI 6 (AI 2–)
GND
AI 3 (AI 3+)
AI 7 (AI 3–)
GND
AO 0
AO 1
GND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P1.0
P1.1
P1.2
P1.3
PFI 0
+2.5 V
+5 V
GND
5V
AI0
GND
$R = 10\ k\Omega$
Thermistor

LabVIEW Example







https://www.halvorsen.blog

Introduction to OPC

Hans-Petter Halvorsen

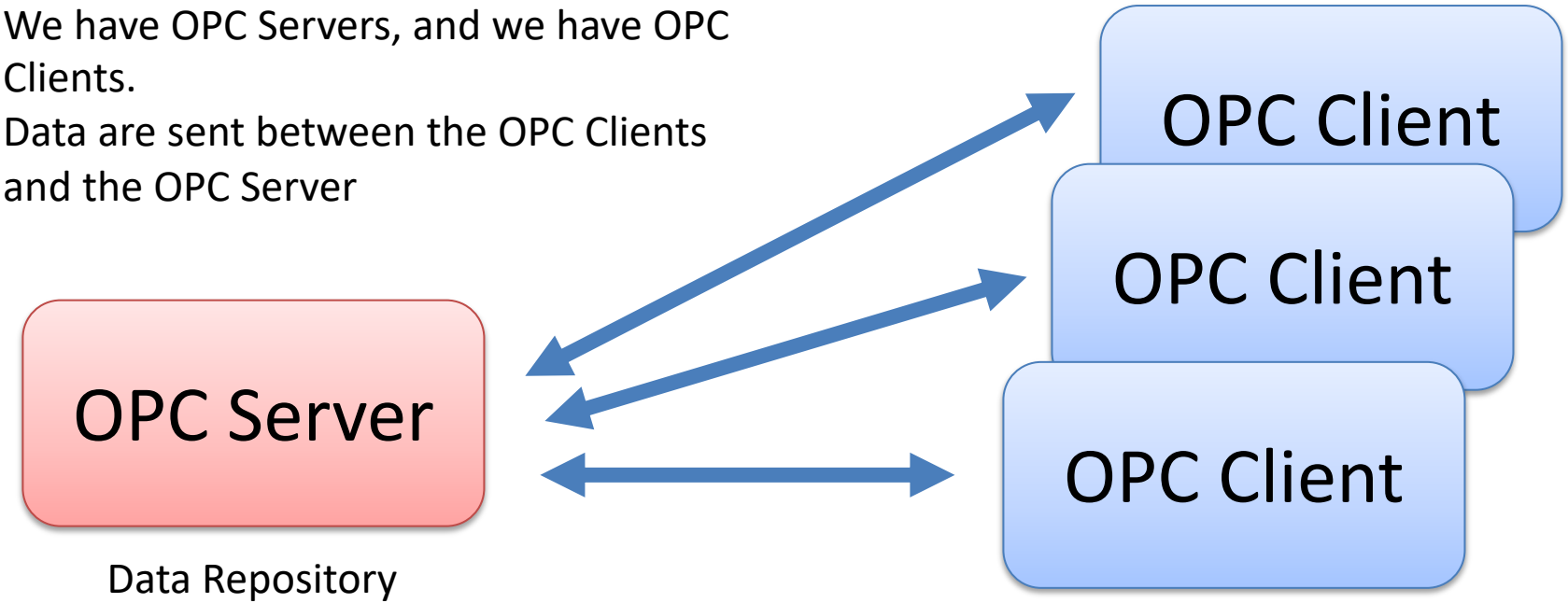
Table of Contents

What is OPC?

- OPC - “Open Process Control”/“Open Platform Communications”
- A standard that defines the communication of data between devices from different manufactures
- Requires an **OPC server** that communicates with the **OPC clients**
- OPC allows “plug-and-play”, gives benefits as reduces installation time and the opportunity to choose products from different manufactures
- Different standards: “Real-time” data (**OPC DA**), Historical data (**OPC HDA**), Alarm & Event data (**OPC AE**), etc.

Basic OPC concept

We have OPC Servers, and we have OPC Clients.
Data are sent between the OPC Clients and the OPC Server



Data Repository

Send Data (Write) to OPC Server
or Retrieve Data (Read) from OPC Server

OPC Specifications

OPC DA (Data Access)

The most common OPC specification is OPC DA, which is used to read and write “real-time” data. When vendors refer to OPC generically, they typically mean OPC DA.

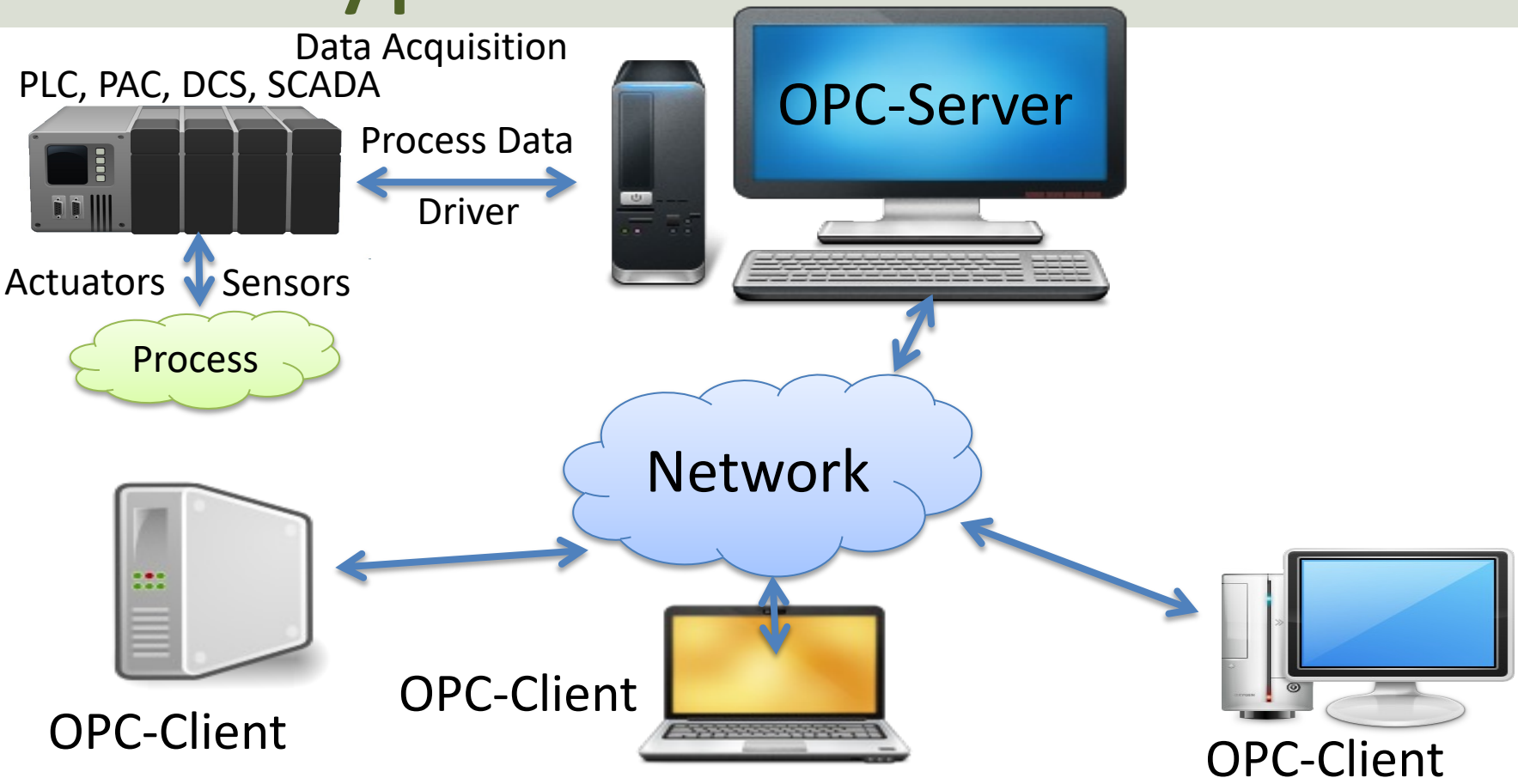
- OPC HDA (Historical Data Access)
- OPC A & E (Alarms & Events)
- ... (many others)

These OPC specification are based on the OLE, COM, and DCOM technologies developed by Microsoft for the Microsoft Windows operating system family. This makes it complicated to make it work in a modern Network! Typically, you need a Tunneller Software in order to share the OPC data in a network (between OPC Servers and Clients)

OPC UA (Unified Architecture)

OPC UA eliminating the need to use a Microsoft Windows based platform of earlier OPC versions. OPC UA combines the functionality of the existing OPC interfaces with new technologies such as XML and Web Services (HTTP, SOAP)

Typical OPC Scenario



OPC DA

Hans-Petter Halvorsen

Table of Contents

MatrikonOPC Simulation Server

Hans-Petter Halvorsen

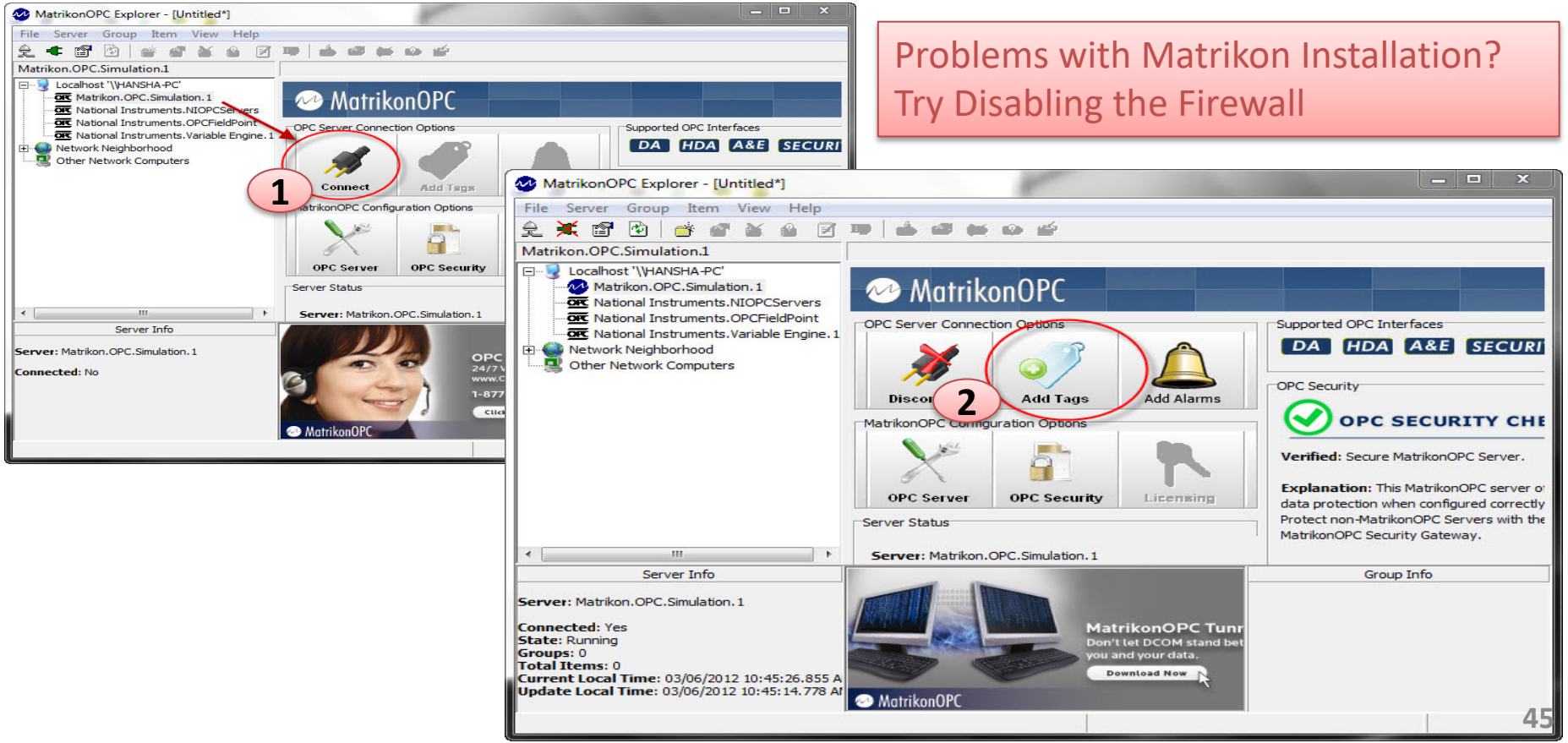
Table of Contents

MatrikonOPC Simulation Server

MatrikonOPC Simulation Server is a free utility that provides Simulated OPC DA, OPC HDA, and OPC A&E Data for the Purposes of Testing OPC Clients

https://www.matrikonopc.com/products/opc-drivers/opc-simulation-server.aspx

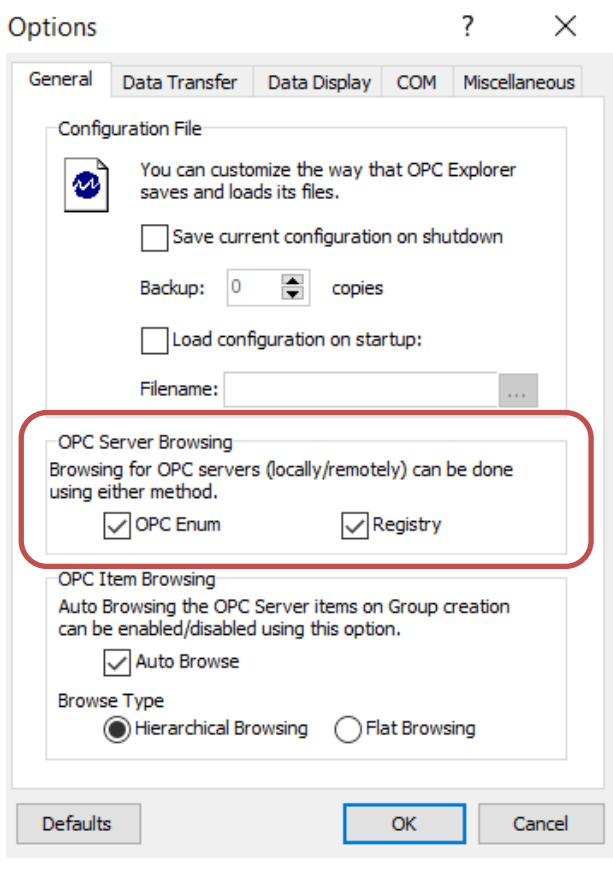
Matrikon OPC Explorer – Connect to Server



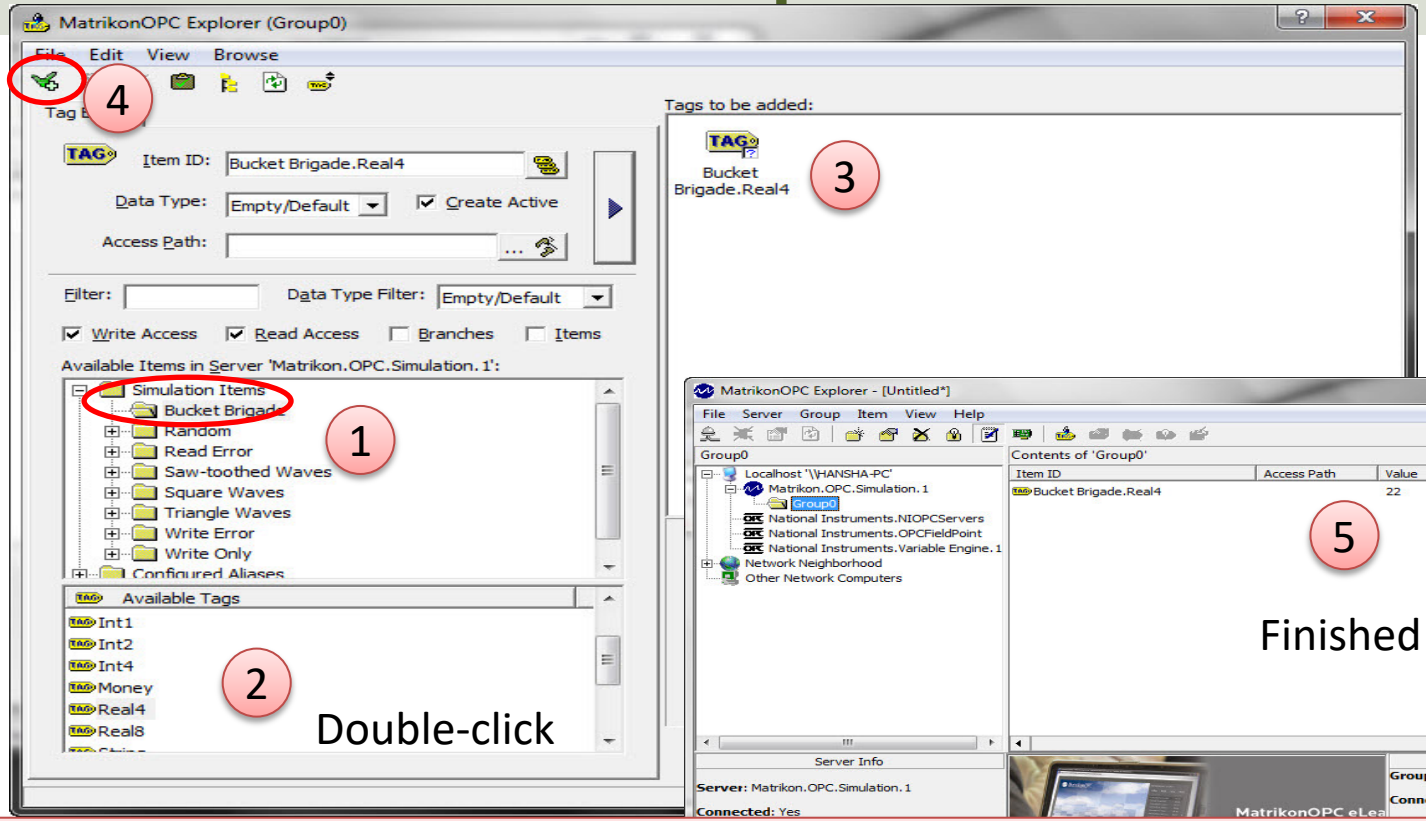
Problems with Matrikon Installation?
Try Disabling the Firewall

MatrikonOPC Explorer Troubleshooting

- **Problem**: “When starting MatrikonOPC Explorer, I get an error indicating there are no servers installed”.
- **Solution**:
  - In OPC Explorer select View ->Options from the menu bar.
  - On the General Tab select both “OPCEnum” and “Registry” as the Browse Methods.
  - Exit OPC Explorer and restart.
  - Upon restarting, you should see a listing of locally registered OPC servers.
  - If this still does not work, remove OPCEnum as a browse method and restart.

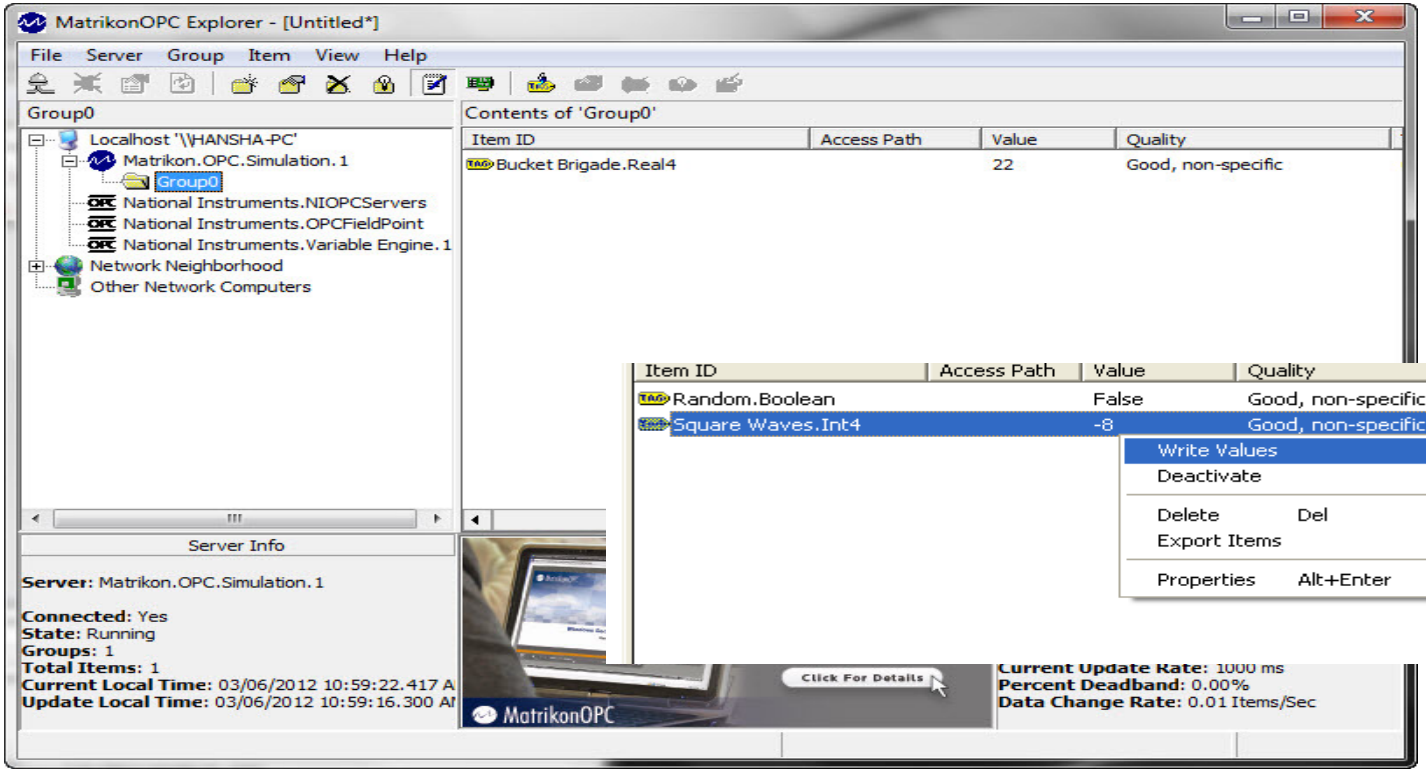


Matrikon OPC Explorer - Add Tag



Tip! Use the **BucketBrigade** Items – because they can be used for both reading and writing

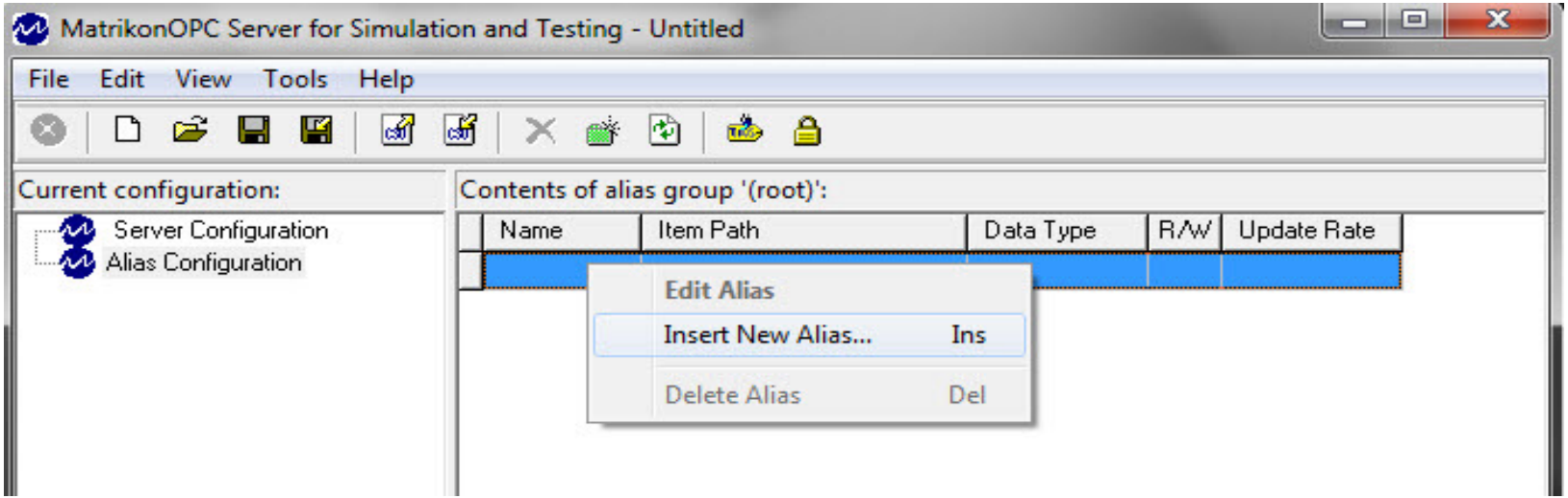
MatrikonOPC Explorer (OPC Client)



The MatrikonOPC Explorer is useful for testing. You can use it for writing and reading OPC Tags

Aliases

In the “Matrikon OPCServer for Simulation” you can create Aliases. Aliases is handy when you want to describe your OPC items using more realistic names.



Tip: You can create an alias called, e.g., “Temperature” which you can use instead of the real OPC Tag Name

OPC DA in LabVIEW

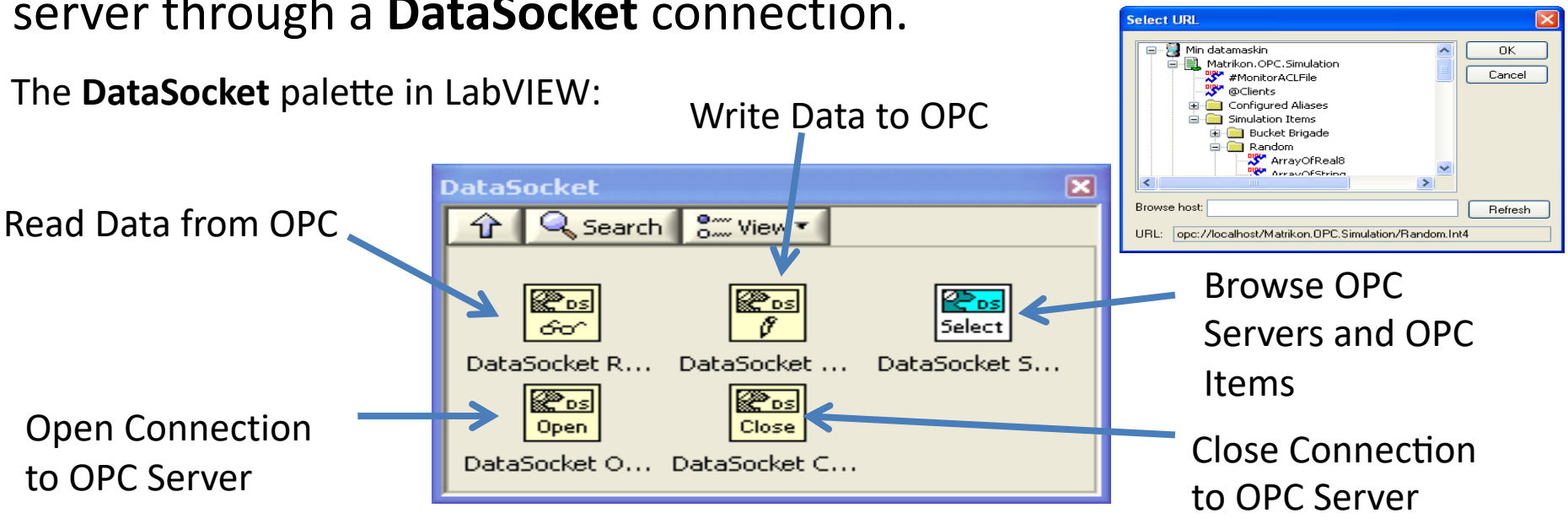
Hans-Petter Halvorsen

Table of Contents

OPC DA in LabVIEW

You can use LabVIEW as an OPC client by connecting to an OPC server through a **DataSocket** connection.

The **DataSocket** palette in LabVIEW:

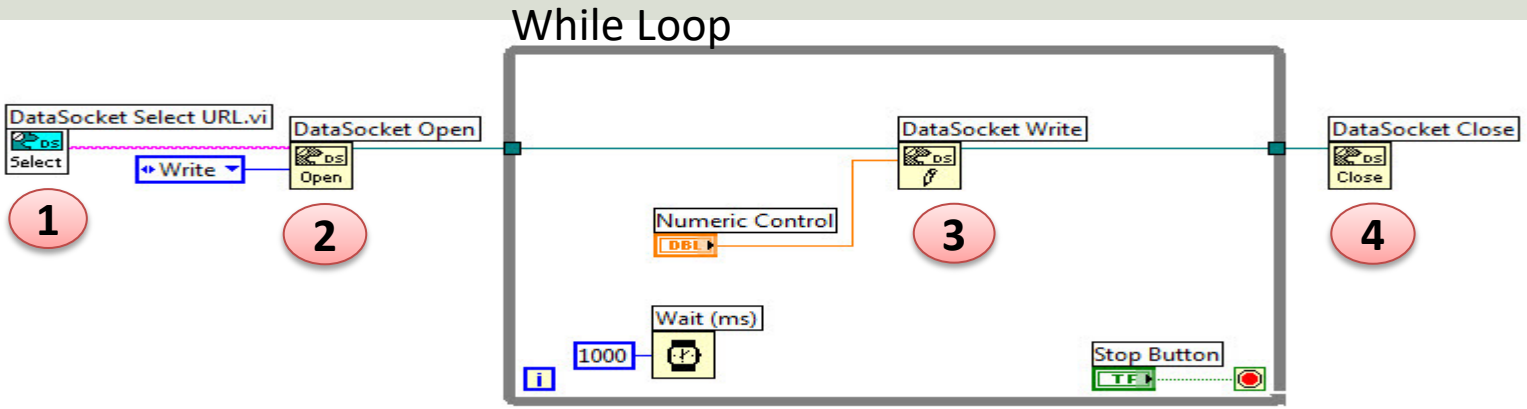


LabVIEW OPC DA - Write

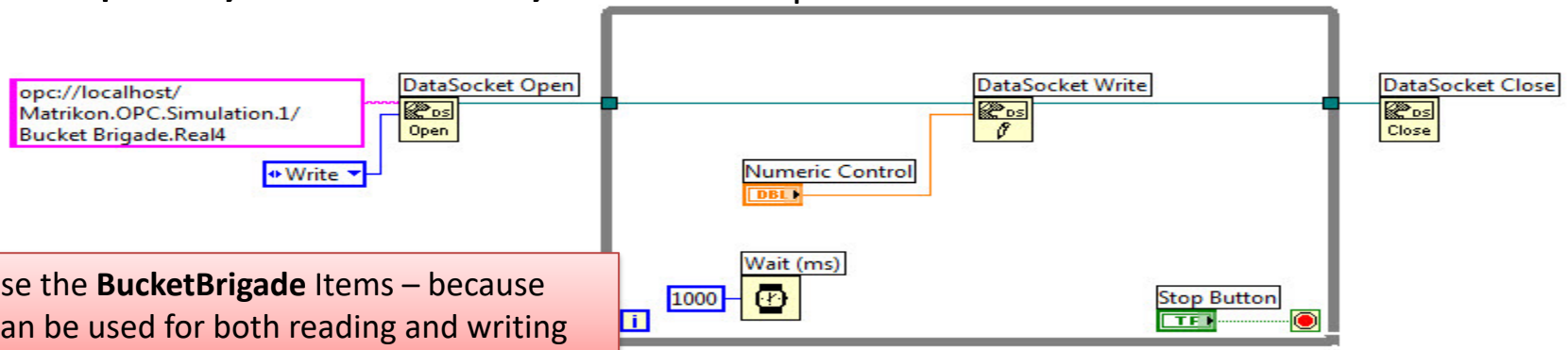
Hans-Petter Halvorsen

Table of Contents

LabVIEW OPC DA - Write

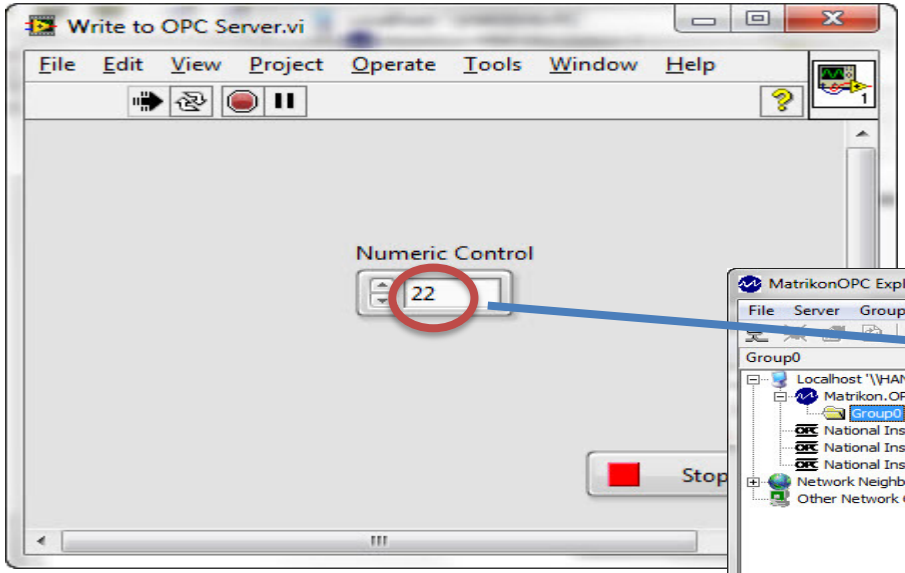


Or specify URL directly:

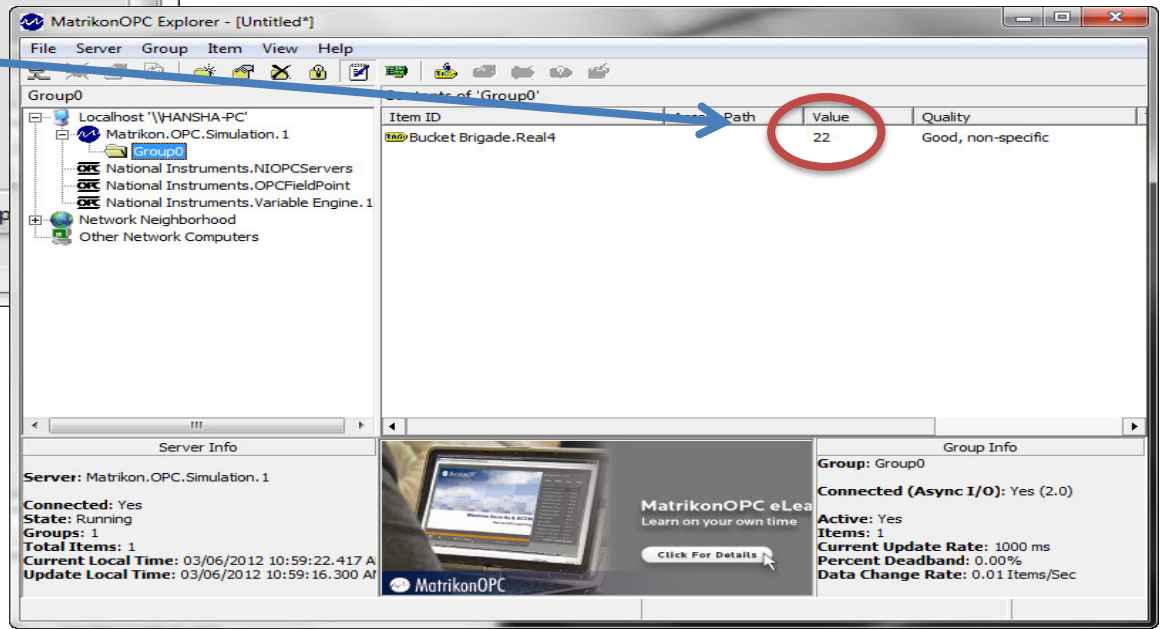


Tip! Use the **BucketBrigade** Items – because they can be used for both reading and writing

Use OPC Explorer to Check Communication



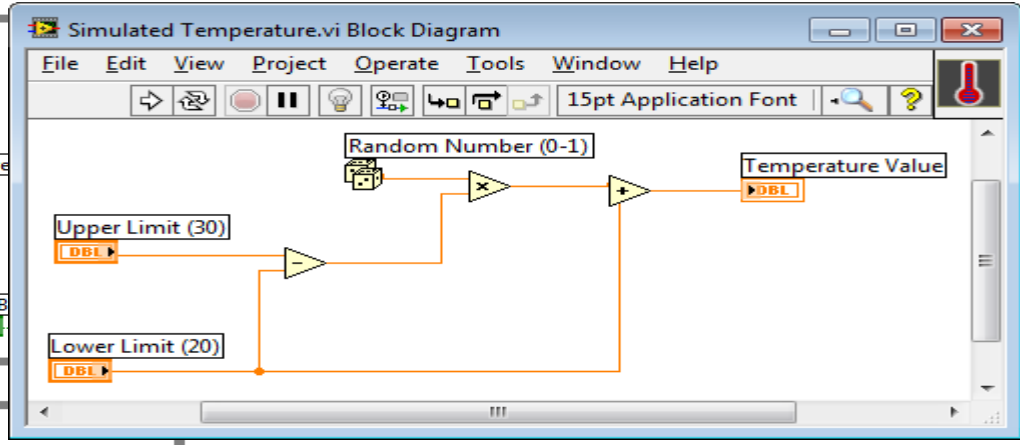
Tip! Run the LabVIEW program and use the Matrikon OPC Explorer to check if the data is correctly written to the OPC Server from LabVIEW



Temperature Simulator Example

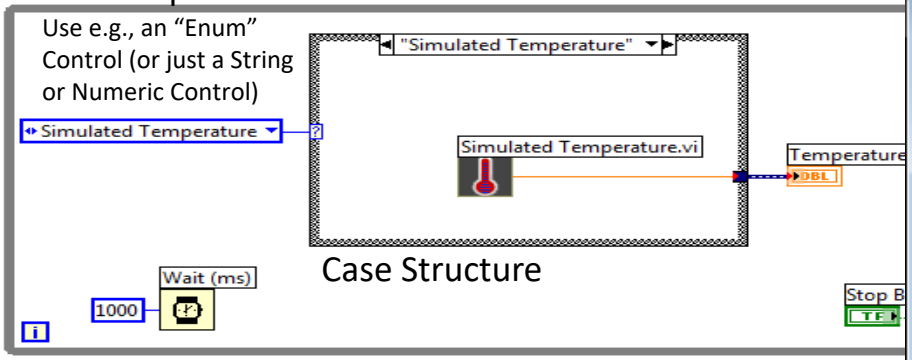
! If you do not have the TC01 device available, you can create and use a simple “Temperature Simulator” instead

A simple SubVI that simulates a Temperature value using a Random Generator:

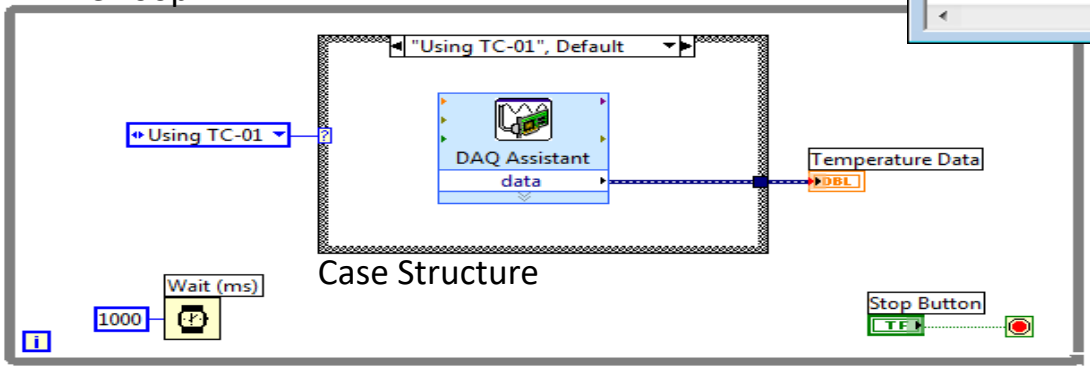


While Loop

Use e.g., an “Enum” Control (or just a String or Numeric Control)



While Loop

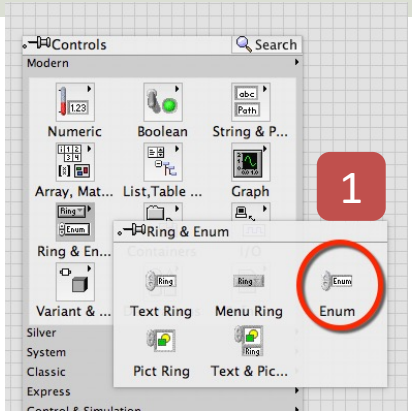


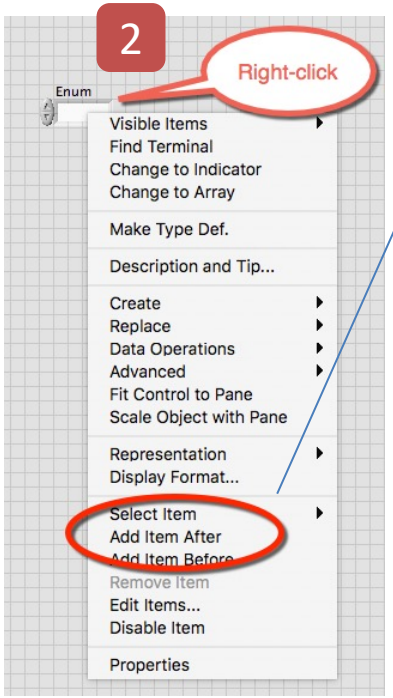
In this way you can easily switch between the real Temperature sensor (TC-01) and the Simulator.

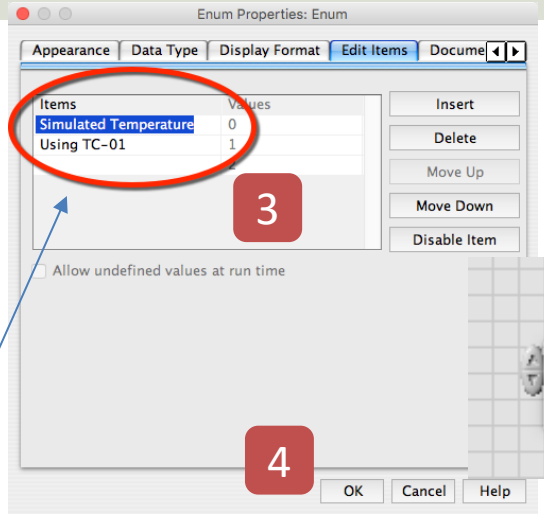
Here you just see a simple example - feel free to create a more realistic Temperature Simulator

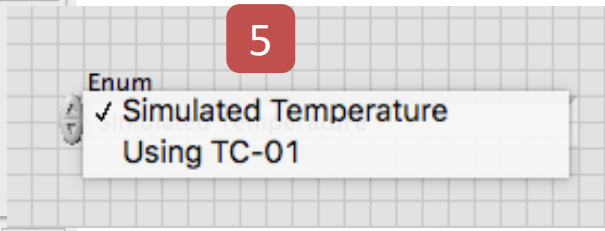
How to create an “Enum” in LabVIEW

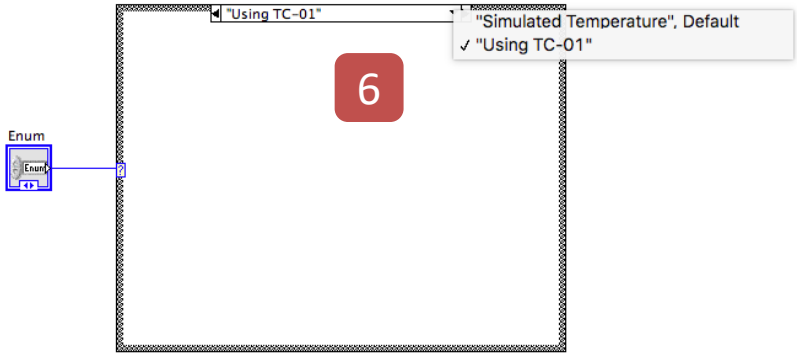
(used in the Temperature Simulator Example)



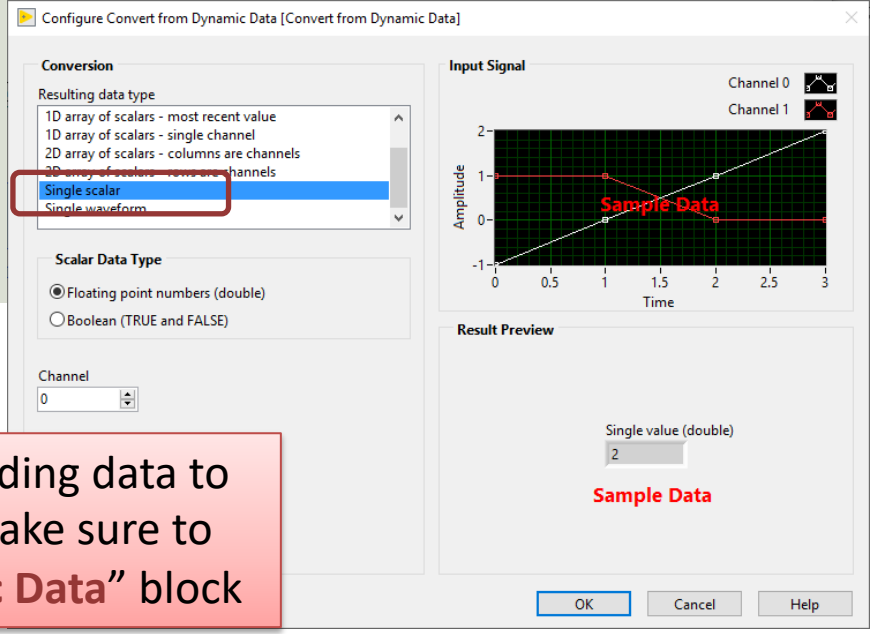


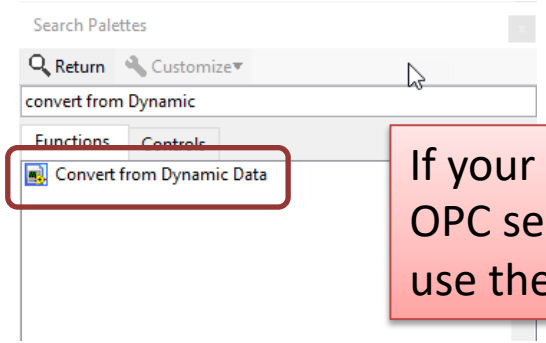




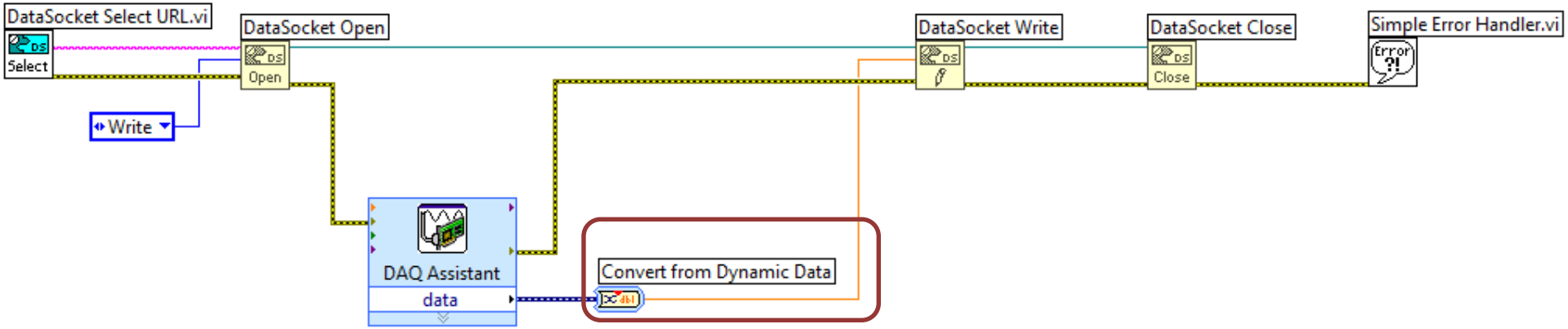


Convert from Dynamic Data





If your Program crash when sending data to OPC server from DAQ device, make sure to use the "**Convert from Dynamic Data**" block

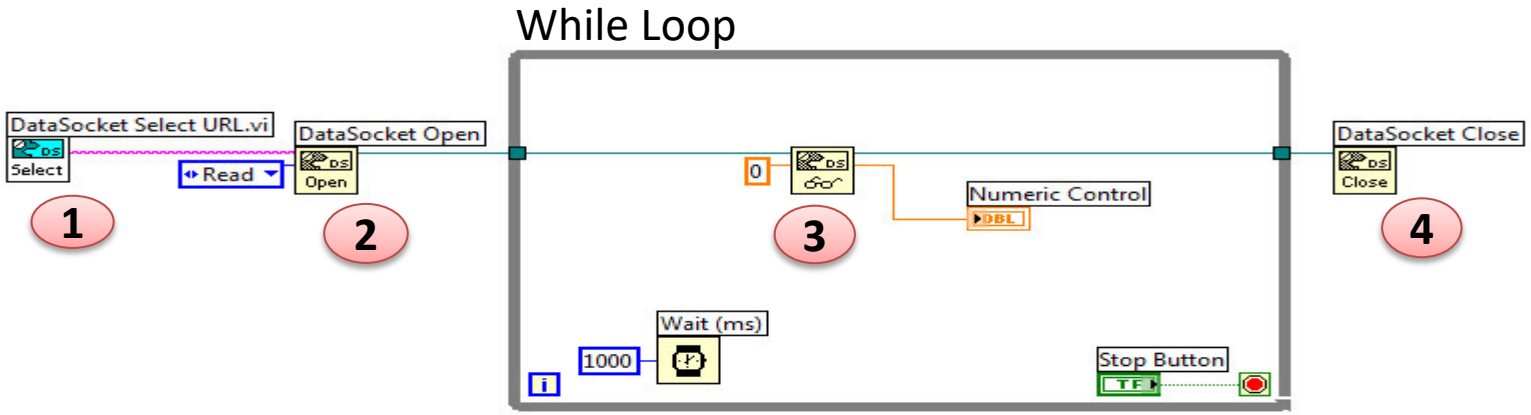


LabVIEW OPC DA - Read

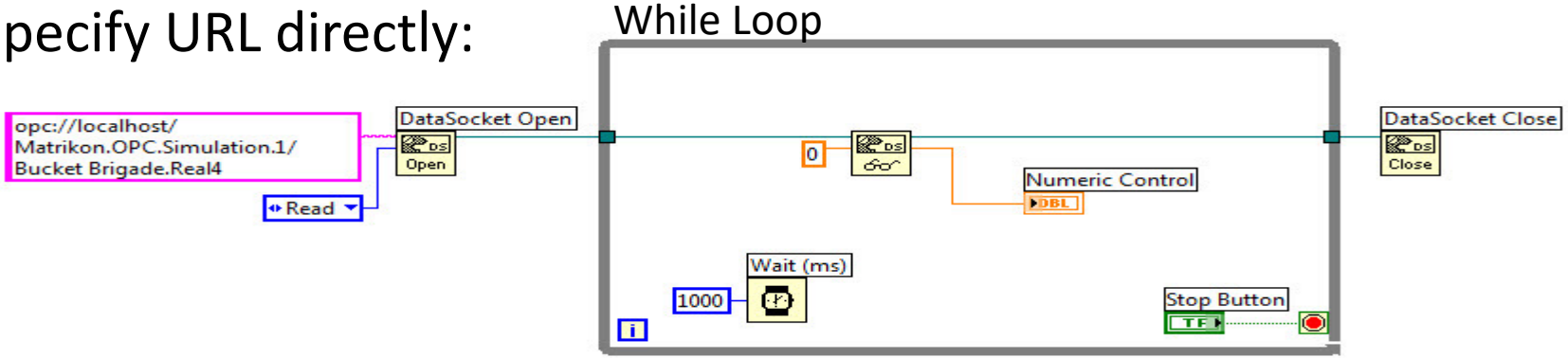
Hans-Petter Halvorsen

Table of Contents

Read from OPC Server using LabVIEW

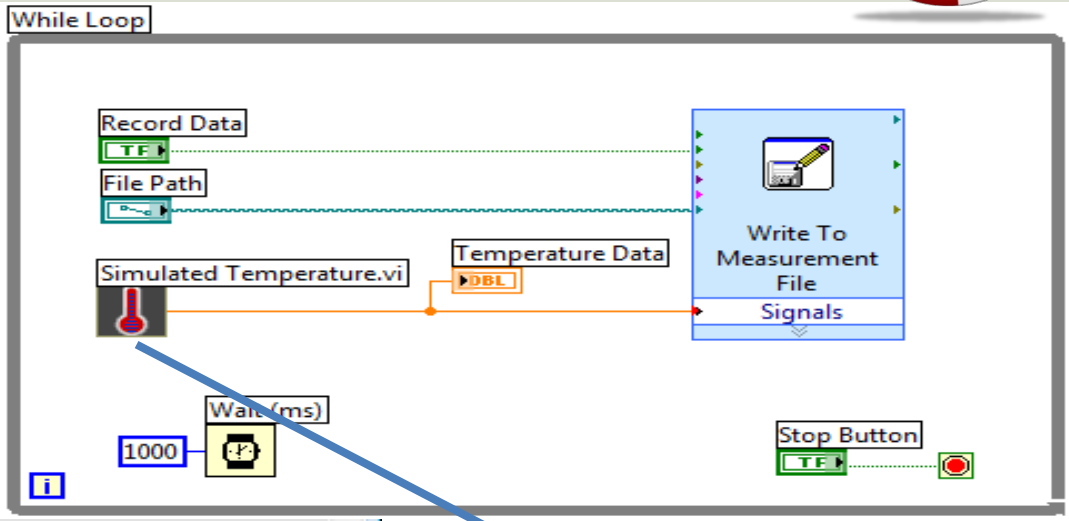


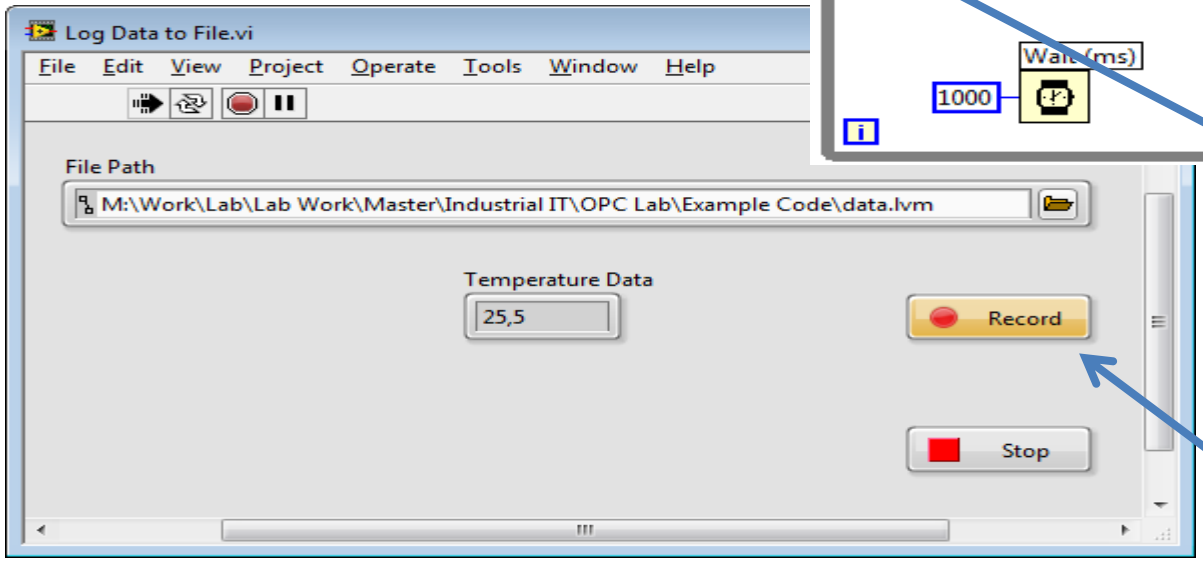
Or specify URL directly:



Log Data to File

Simple Example of how to log data to a Measurement File using the “Write To Measurement File” function in LabVIEW



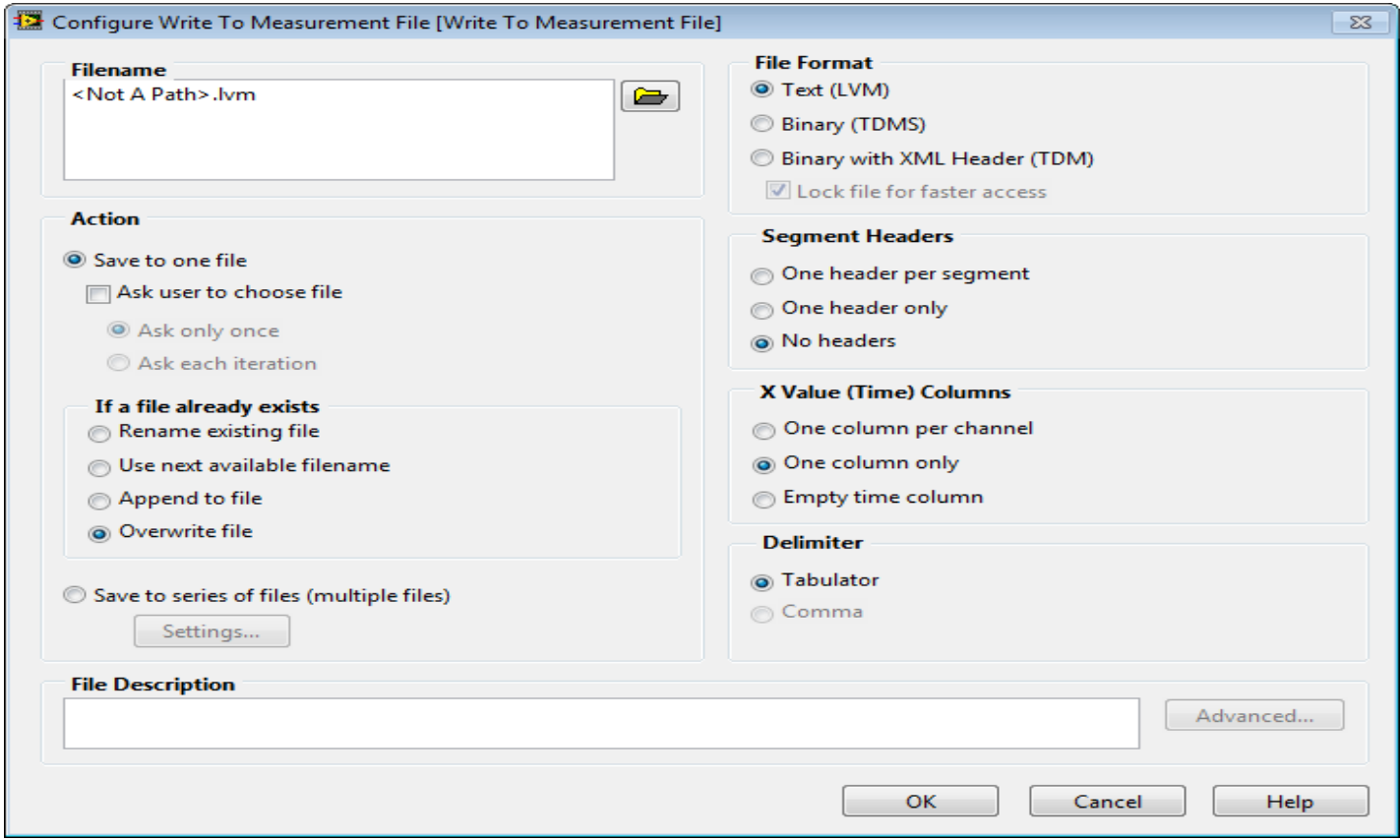


In this example we just generate some random data. In your case you shall log the data received from the OPC Server

You can turn logging On/Off

Log Data to File - Properties

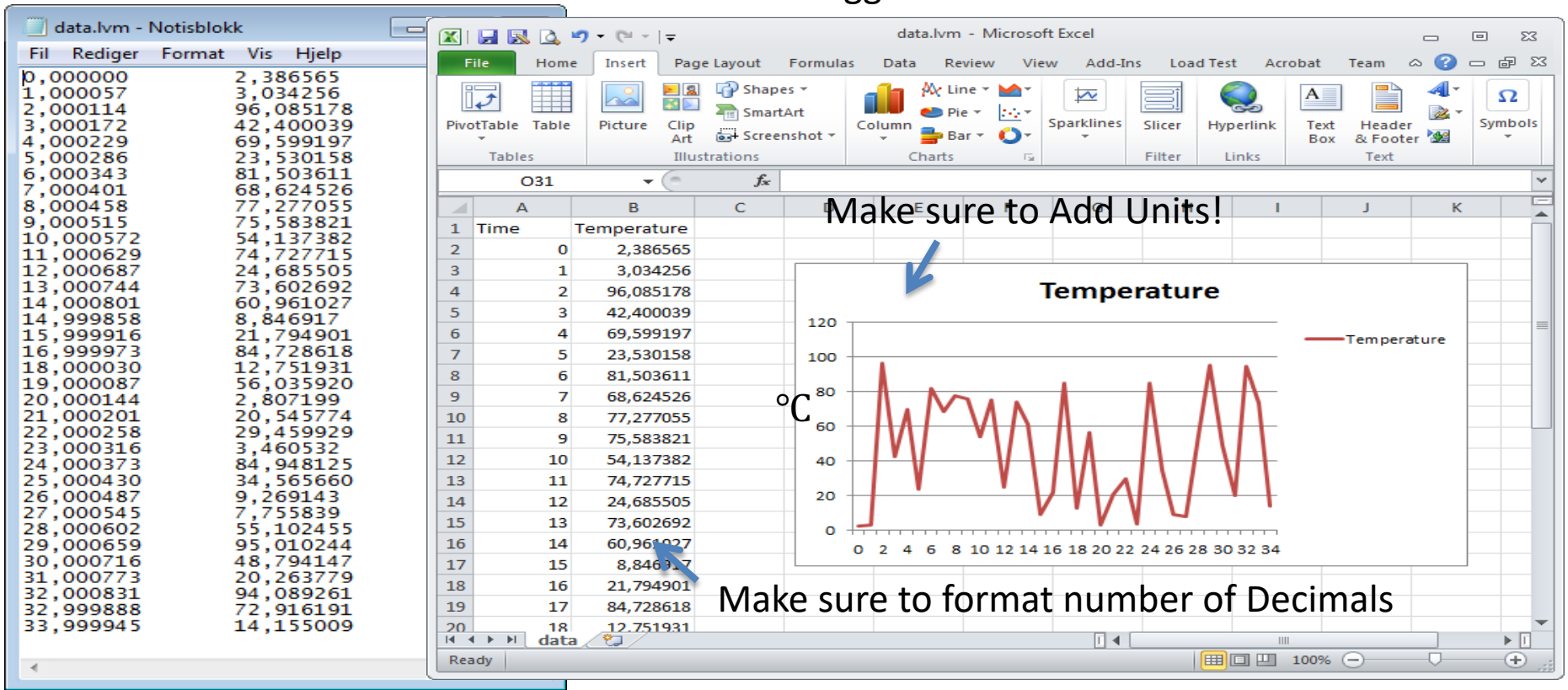
Recommended Settings in the **Properties** Window (Right-click on the Write To Measurement File icon):



Measurement File – Data Visualization

Open the File with Logged Data in e.g., **Notepad**:

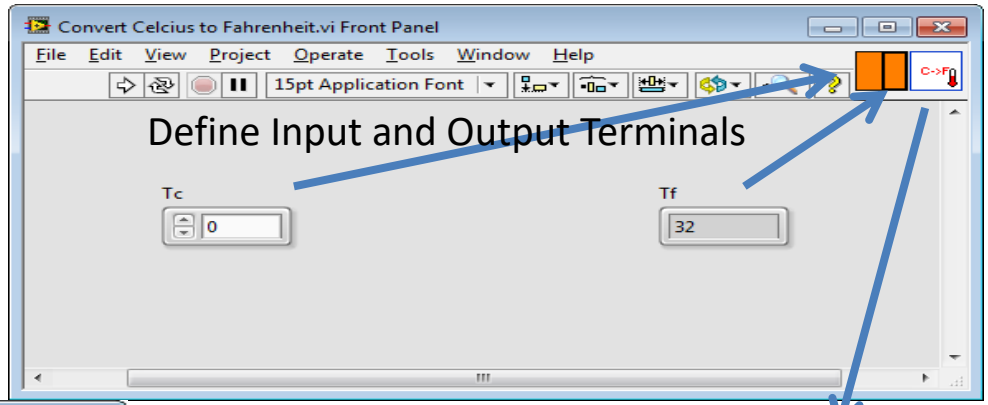
Here we see an example where we have opened the File with Logged Data in **MS Excel** and created a Chart



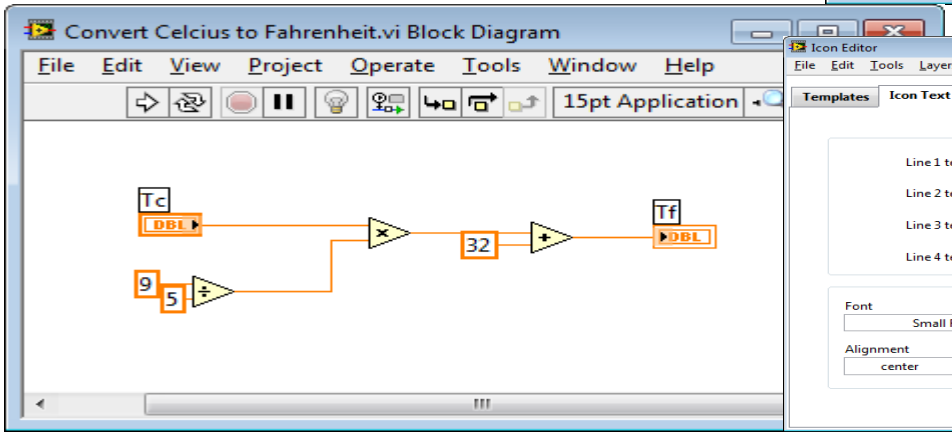
SubVI – Scaling from Celsius to Fahrenheit

$$Tf = (9/5)Tc + 32$$

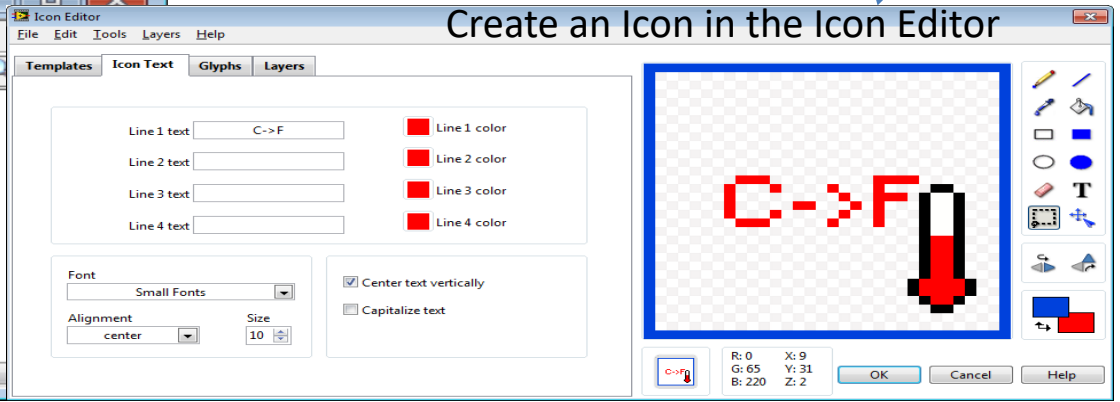
Front Panel:



Block Diagram:



Create an Icon in the Icon Editor



OPC DA in Visual Studio/C#

Hans-Petter Halvorsen

Table of Contents

Measurement Studio 2019

- Measurement Studio is an add-on to Visual Studio.
- Measurement Studio is used for development of measurement, control and monitoring applications using .NET and Visual Studio.
- Measurement Studio has a library (NetworkVariable) that makes it possible to communicate with OPC DA servers that we will use is this lab work
- Download Software here: https://www.ni.com/download

Measurement Studio

LabVIEW DSC Module

- LabVIEW DSC Module is an additional module for LabVIEW
- DSC – Datalogging and Supervisory Control
- Exchanging data between Measurement Studio applications and OPC servers requires LabVIEW DSC.

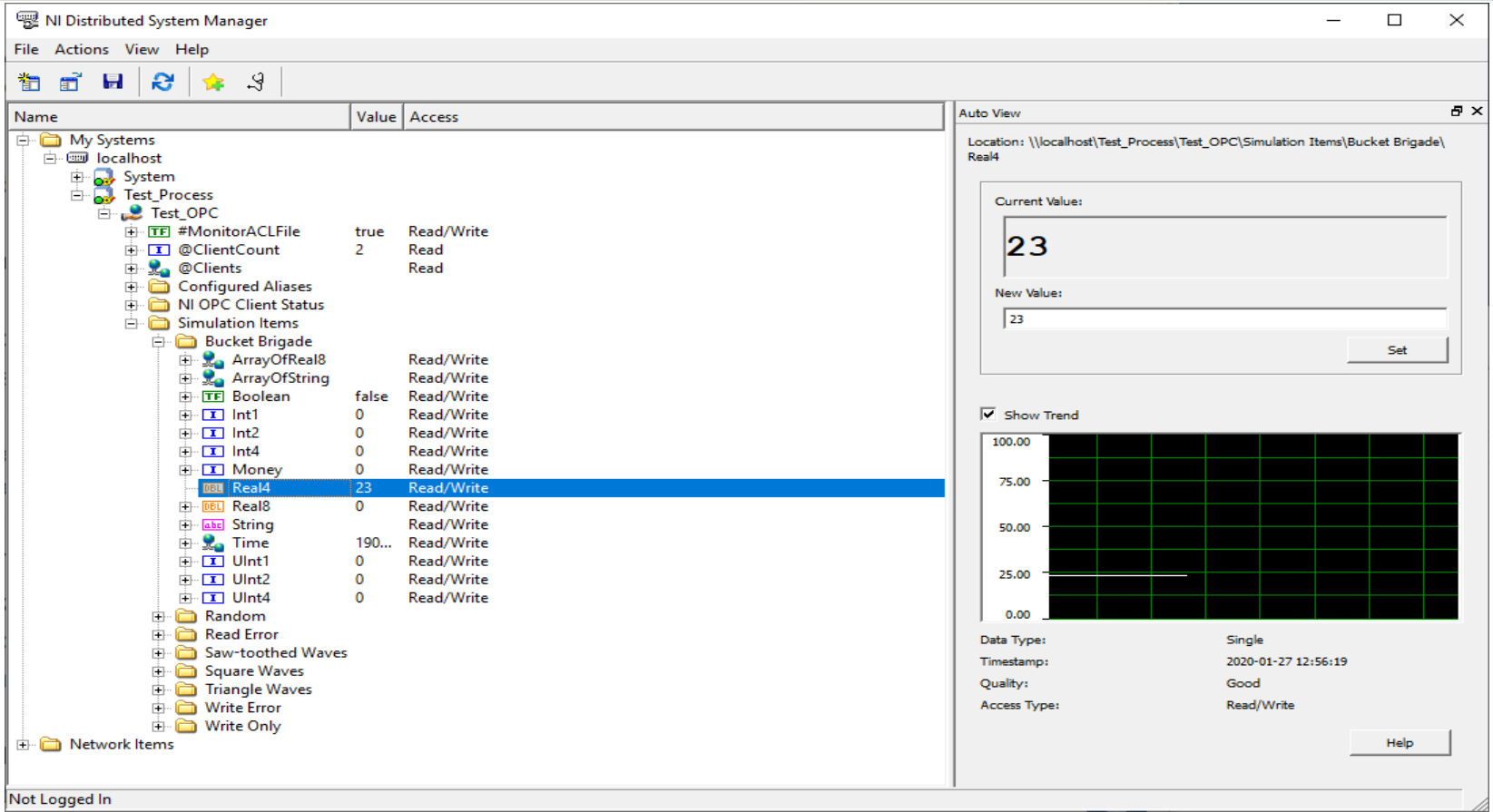
OPC with NetworkVariable

The following paragraphs explain how to use NetworkVariable with an OPC server using the LabVIEW DSC Run-Time System.

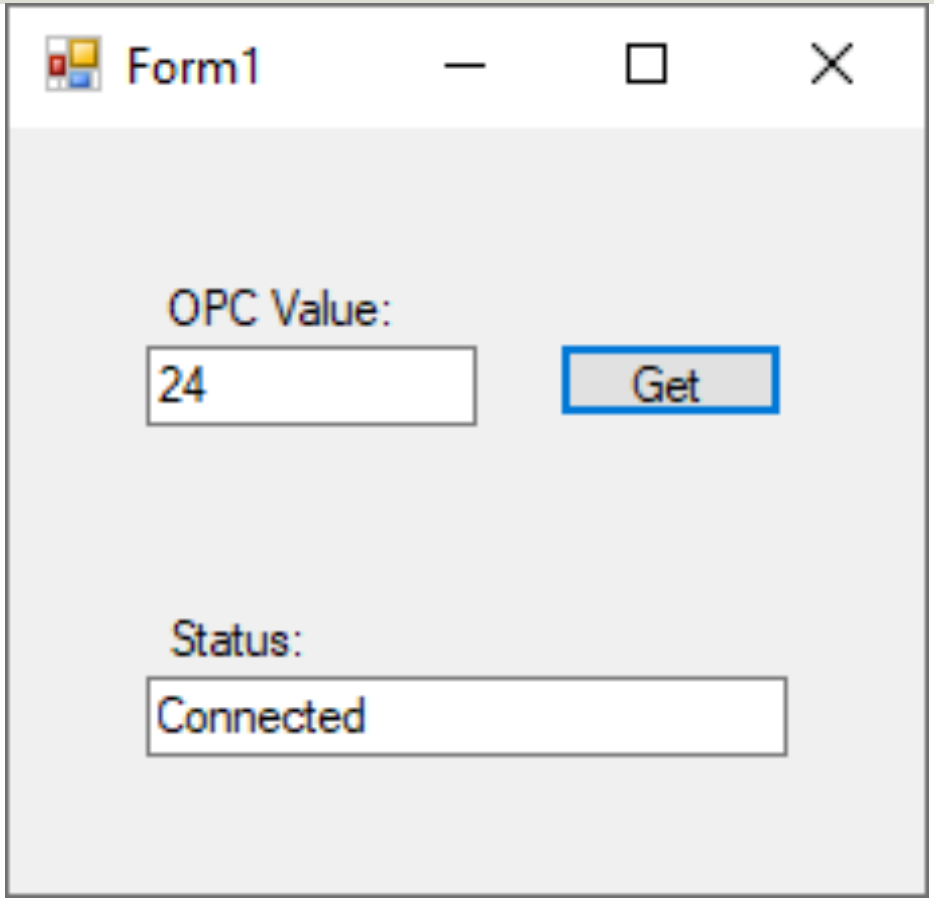
1. **Install LabVIEW Datalogging and Supervisory Control (DSC)** Run-Time System.
2. **Install your OPC server**. Only OPC2 and higher are supported by LabVIEW DSC Run-Time System.
3. Select Start»All Programs»National Instruments»**Distributed System Manager** to launch the application.
4. Right-click localhost and select **Add Process** to create a new process. Type Test_Process in the Add Process dialog box and click OK. Grouping variables by process allows you to organize your variables. You can start and stop processes independently, which allows you to easily manage your variables.
5. Right-click on Test_Process and select **Add I/O Server**.
6. For the I/O Server Type, **select OPC Client** and click Continue.
7. Type Test_OPC in the **Enter IO Server Name** dialog box and click OK.
8. **Select the OPC server** that you want to access through the Network Variable API from the list of Registered OPC Servers you installed in step 3 and click OK.
9. Right-click on Test_Process and select **Add Variable** to launch the **Shared Variable Properties** dialog box.
10. In the Shared Variable Properties dialog box, select the **Enable Aliasing** checkbox and click the Browse button.
11. In the Browse for Variable dialog box, select one of the OPC items from the OPC I/O server you configured in step 6.
12. Click OK to **bind the new variable to the OPC source**.
13. Click OK to return to NI Distributed System Manager. Use the new variable as you would any other shared variable. You can access the variable you have configured through the .NET **NetworkVariable class library**, as you would any other network variable.

http://zone.ni.com/reference/en-XX/help/375857B-01/mstudionetvar/netvar_opc/

Distributed System Manager



OPC DA with Visual Studio/C#



Basic Example that reads Temperature Data from the OPC Server using Visual Studio.

```
using NationalInstruments;
using NationalInstruments.NetworkVariable;

namespace OPCExample
{
   public partial class Form1 : Form
   {
      private NetworkVariableReader<float> _reader;
      private const string NetworkVariableLocation = @"\\localhost\Test_Process\opctempdata";

      public Form1()
      {
         InitializeComponent();

         ConnectOPCServer();
      }

      private void btnGetData_Click(object sender, EventArgs e)
      {
         NetworkVariableData<float> opcdata = null;
         try
         {
            opcdata = _reader.ReadData();

            txtOpcData.Text = opcdata.GetValue().ToString();
         }
         catch (TimeoutException)
         {
            MessageBox.Show("The read has timed out.", "Timeout");
            return;
         }
      }
...
```

```
....

      private void ConnectOPCServer()
      {
         _reader = new NetworkVariableReader<float>(NetworkVariableLocation);

         _reader.Connect();

         txtStatus.Text = _reader.ConnectionStatus.ToString();
      }

      private void Form1_FormClosing(object sender, FormClosingEventArgs e)
      {
         _reader.Disconnect();
      }

   }
}
```

Note! This Code Snippet reads only one value once when clicking the button. You can use e.g., a **Timer** in order to read values at specific intervals

Timer

1  Timer

Select the “Timer” component in the Toolbox

2 **Initialization:**

```
public Form1()
      {
         InitializeComponent();

         timer1.Start();
      }
```

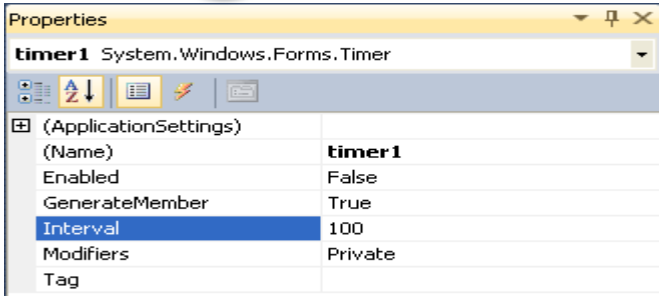
Double-click on the Timer object in order to create the Event

4 **Timer Event:**

```
private void timer1_Tick(object sender, EventArgs e)
      {
        … //Read from OPC
        … //Scaling
        … //Plot Data
      }
```

In Visual Studio you may want to use a Timer instead of a While Loop in order to read values at specific intervals.

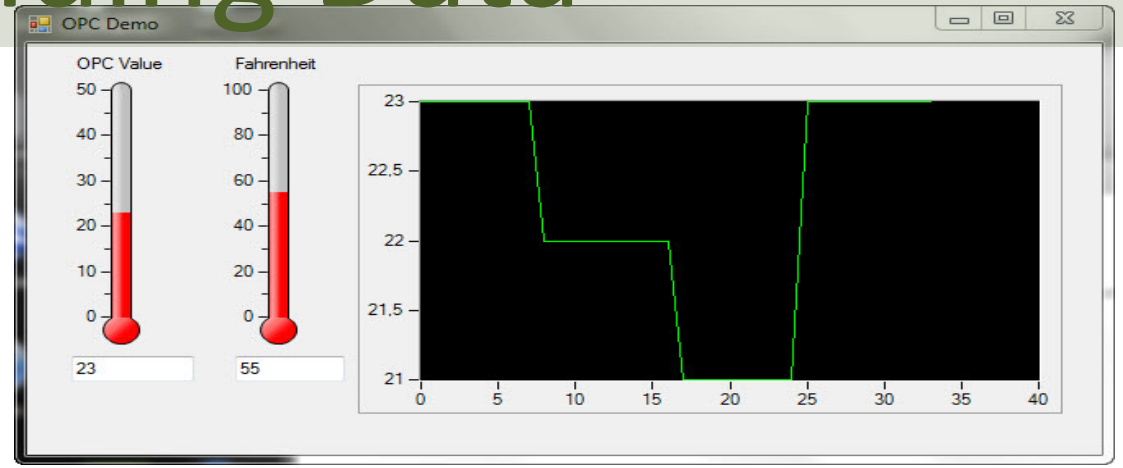
Properties: 3



You may specify the Timer Interval in the Properties Window

Structure your Code properly!!
Define Classes and Methods which you can use here

Trending Data



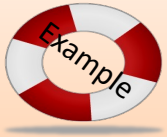
You may use the "**WaveformGraph**" Control included with Measurement Studio

You only need one line of code, e.g., in the Timer Event:

```
…
{
        …

        waveformGraph.PlotYAppend(analogDataIn);

}
```



Name of your WaveformGraph control

Name of the Method to use

Name of the variable with Temperature data

https://www.halvorsen.blog

OPC DA in MATLAB

Hans-Petter Halvorsen

Table of Contents

```
clear
clc
% Connect to OPC Server
da = opcda('localhost', 'Matrikon.OPC.Simulation.1');
connect(da);

% Create Group
grp = addgroup(da, 'DemoGroup');

%Add Tags
itmIDs = {'Random.Real8'};
itm = additem(grp, itmIDs);

% Retrieve Data
data = read(grp);
opcdata = data.Value

%Clean Up
disconnect(da)
delete(da)
```

MATLAB OPC DA Read Example 1

This simple Example reads only one value from the Server

MATLAB OPC DA Read Example 2

This simple Example reads values from the Server.
This Examples reads N values using a For Loop

```
% OPC Example
clear
clc

% Connect to OPC Server
da = opcda('localhost', 'Matrikon.OPC.Simulation.1');
connect(da);

% Create Group
grp = addgroup(da, 'DemoGroup');

%Add Tags
itmIDs = {'Random.Real8'};
itm = additem(grp, itmIDs);

% Retrieve Data
N=10;
for i=1:N
   data = read(grp);
   opcdata(i) = data.Value;
   pause(2)
end

%Clean Up
disconnect(da)
delete(da)

plot(opcdata)
```

MATLAB OPC DA Read Example 2b

This simple Example reads values from the Server. The For Loop has been replaced with a While Loop

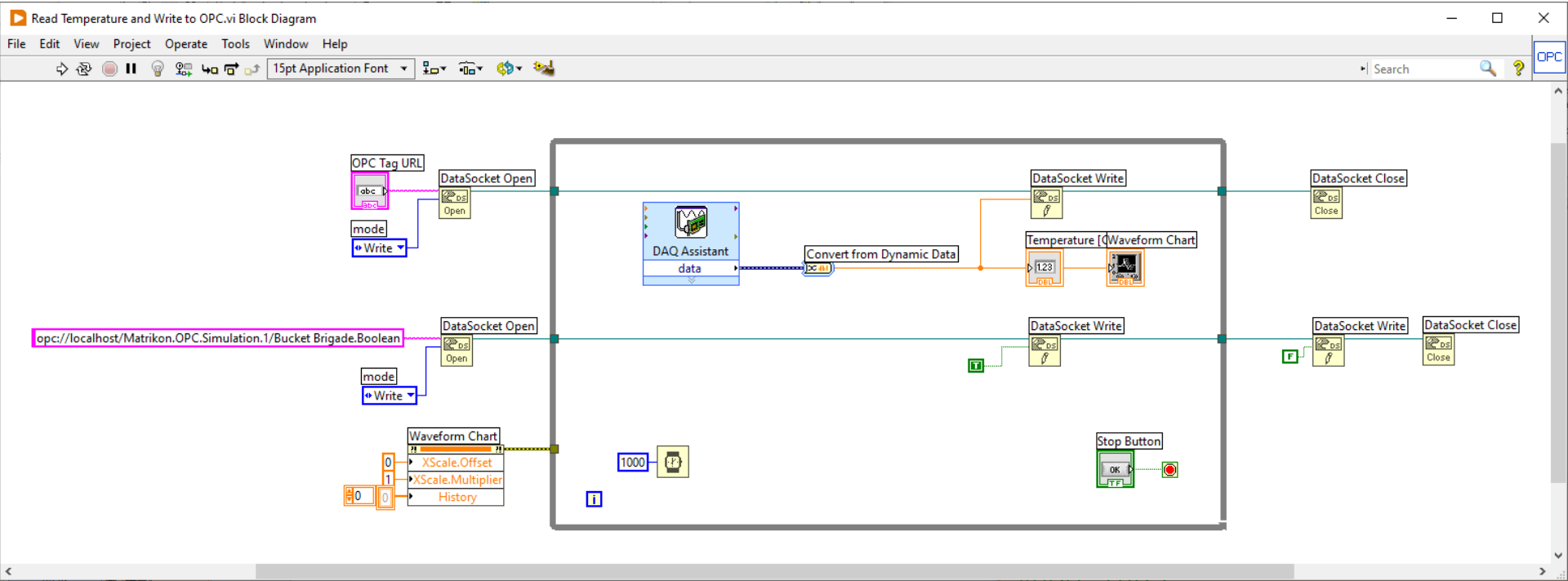
```
% OPC Example
clear
clc
% Connect to OPC Server
da = opcda('localhost',
'Matrikon.OPC.Simulation.1');
connect(da);

% Create Group
grp = addgroup(da, 'DemoGroup');
%Add Tags
itmIDs = {'Bucket Brigade.Real4'};
itm = additem(grp, itmIDs);

% Retrieve Data
while(1)
    data = read(grp);
    opcdata = data.Value
    pause(2)
end
%Clean Up
disconnect(da)
delete(da)
```

Example 2c

This Example write values to the Server. In uses a Boolean OPC Tag to flag that the client is writing data to the Server



Example 2c

This Example reads values from the Server using a While Loop. In uses a Boolean OPC Tag to flag that the Client is writing data to the Server. The program stops when the flag is set to False (0)

```
% OPC Example
clear
clc
% Connect to OPC Server
da = opcda('localhost',
'Matrikon.OPC.Simulation.1');
connect(da);

% Create Group
grp = addgroup(da, 'DemoGroup');
%Add Tags
itmIDs = {'Bucket
Brigade.Real4', 'Bucket
Brigade.Boolean'};
itm = additem(grp, itmIDs);
```

```
% Retrieve Data
log_data = 1;
while(log_data)
    data = read(grp);
    opcdata = data(1).Value
    log_data = data(2).Value;
    pause(2)
end
%Clean Up
disconnect(da)
delete(da)
```

MATLAB OPC DA Read Example 3

This simple Example uses some of the more advanced features in the MATLAB OPC Toolbox.
No For/While Loop needed!

```
clear, clc
% Connect to OPC Server
da = opcda('localhost', 'Matrikon.OPC.Simulation.1');
connect(da);
% Create Group
grp = addgroup(da, 'DemoGroup');
%Add Tags
itmIDs = {'Random.Real8'};
itm = additem(grp, itmIDs)
% Set Properties
logDuration = 60;logRate = 0.2;
numRecords = ceil(logDuration./logRate)
grp.UpdateRate = logRate;
grp.RecordsToAcquire = numRecords;
% Acquire Data
start(grp), wait(grp)
% Retrieve Data
[logIDs, logVal, logQual, logTime, logEvtTime] =
getdata(grp, 'double');
% Plot Data
plot(logTime, logVal);
axis tight
datetick('x', 'keeplimits')
legend(logIDs)
%Clean Up
disconnect(da)
delete(da)
```

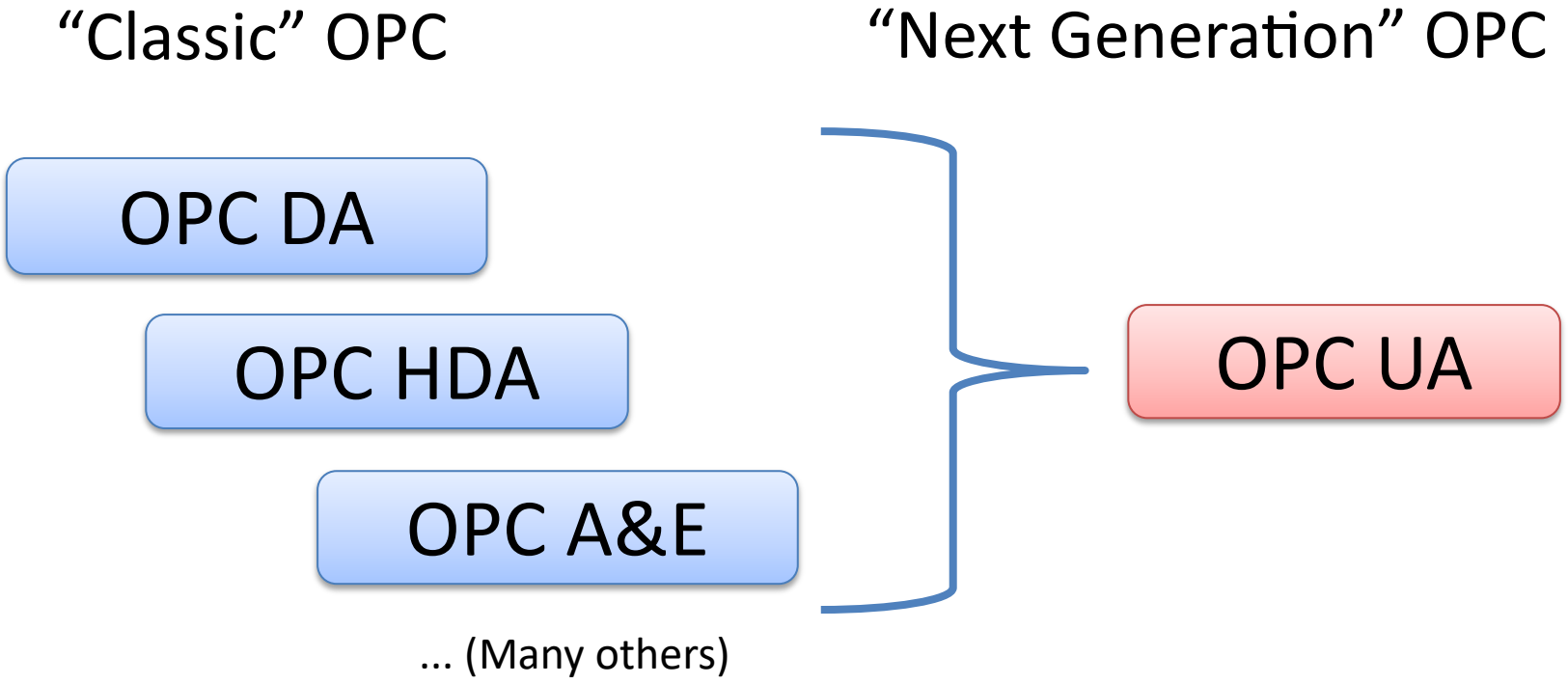
OPC UA

OPC Unified Architecture

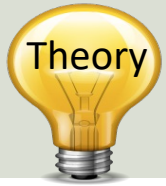
Hans-Petter Halvorsen

Table of Contents

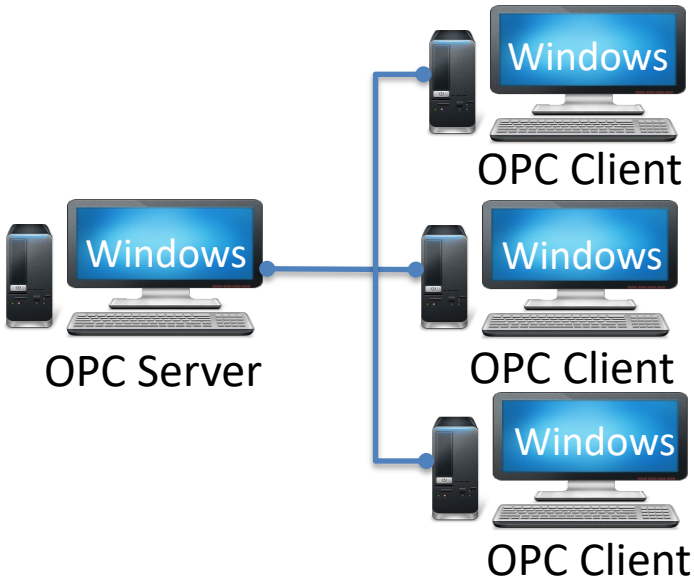
“Next Generation” OPC



Classic OPC vs. OPC UA

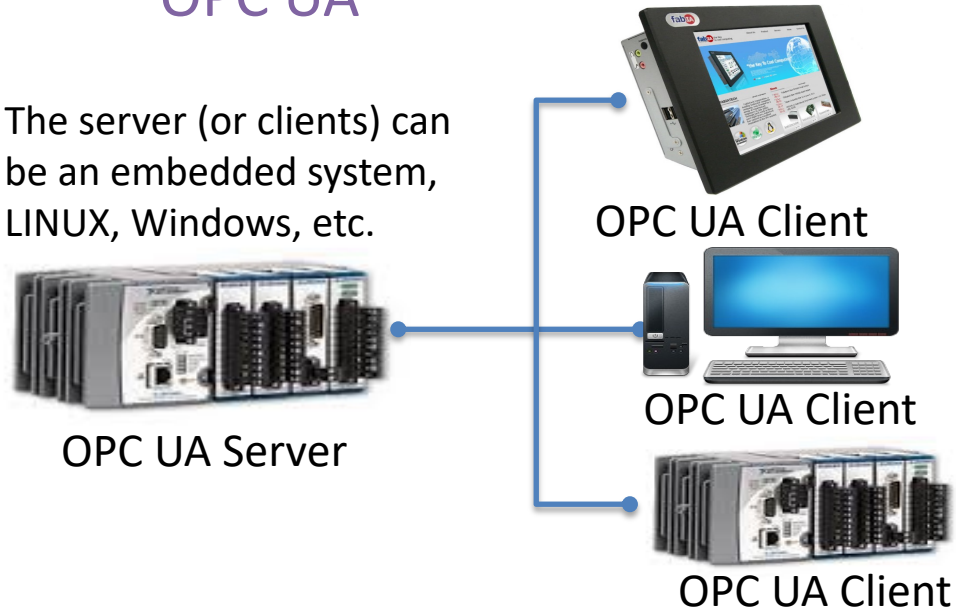


Classic OPC (DCOM)



OPC UA

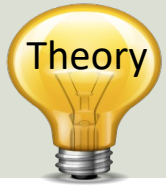
The server (or clients) can be an embedded system, LINUX, Windows, etc.



Classic OPC requires a Microsoft Windows operating system to implement COM/DCOM server functionality. By utilizing SOA and Web Services, OPC UA is a platform-independent system that eliminates the previous dependency on a Windows operating system. By utilizing SOAP/XML over HTTP, OPC UA can deploy on a variety of embedded systems regardless of whether the system is a general purpose operating system, such as Windows, or a deterministic real-time operating system.

http://www.ni.com/white-paper/13843/en/

Next Generation OPC



COM/DCOM

OPC Classic

Next Generation OPC

XML, HTTP, SOAP

OPC UA

OPC DA

OPC HDA

OPC A&E

Specifications

Windows only

Cross-platform

Windows, Linux, Mac, Embedded, VxWorks

All specifications collected in one (DA, HDA, A&E)

Protocols: “UA Binary” or “UA XML”

OPC DA Server

OPC HDA Server

OPC A&E Server

OPC Client (DCOM)

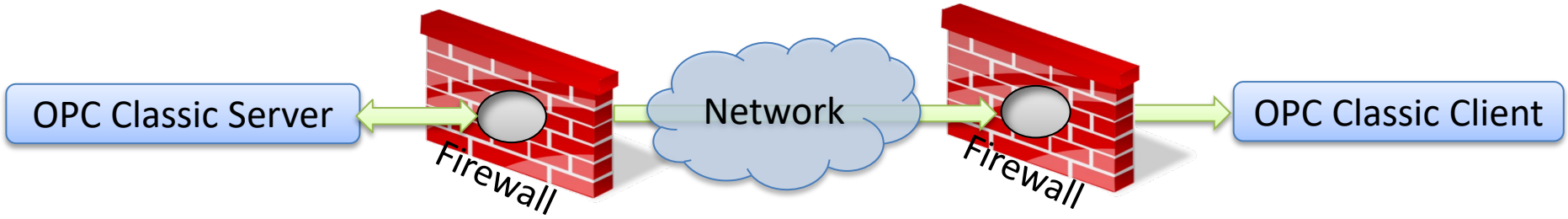
Simpler!!

OPC UA Server

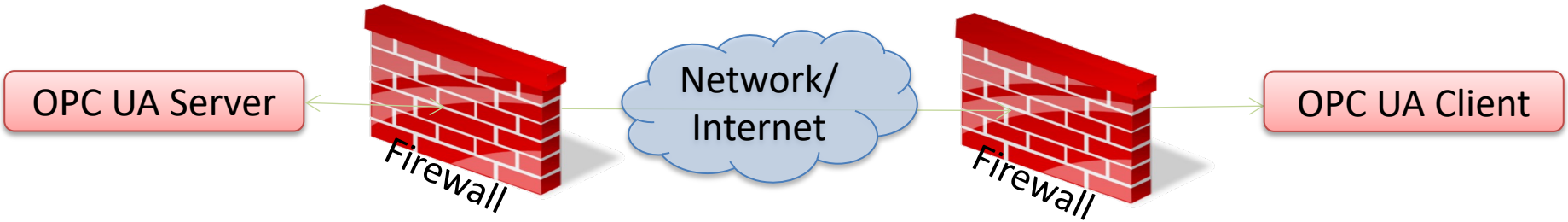
OPC UA Client

(everything built into one)

Firewall

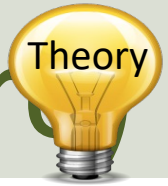


To open DCOM through firewalls demanded a large hole in the firewall!
Impossible to route over Internet!



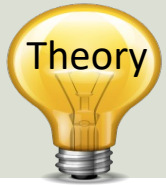
No hole in firewall (UA XML) or just a simple needlestick (UA Binary) is necessary
Easy to route over Internet!

OPC UA (Unified Architecture)



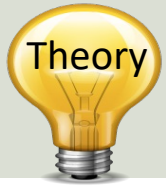
- OPC UA solves problems with standard/classic OPC
  - Works only on Windows
  - Cumbersome to use OPC in a network due to COM/DCOM
- OPC UA eliminating the need to use a Microsoft Windows based platform of earlier OPC versions.
- OPC UA combines the functionality of the existing OPC interfaces with new technologies such as XML and Web Services (HTTP, SOAP)
- Cross-platform
- No dedicated OPC Server is no longer necessary because the server can run on an embedded system

OPC UA Protocols

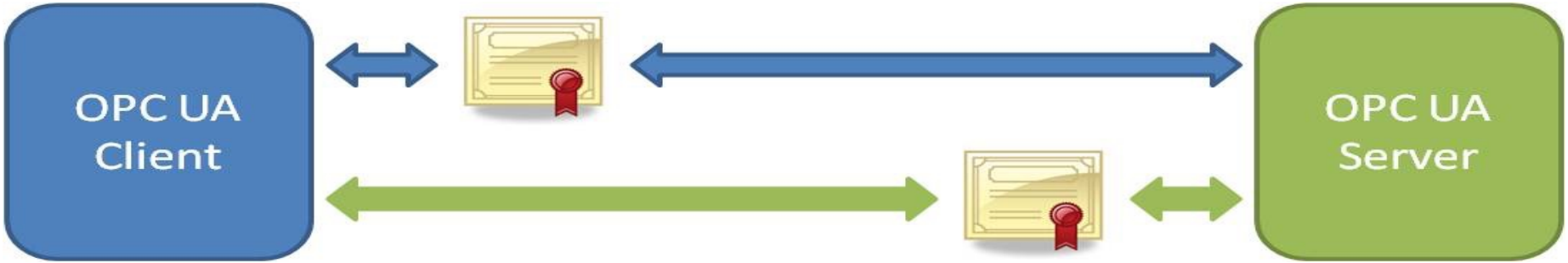


- OPC UA supports two protocols.
  - **“UA Binary”** protocol opc.tcp://Server
    This uses a simple binary protocol
  - **“UA XML”** protocol http://Server
    This used open standards like XML, SOAP (-> Web Service)
- This is visible to application programmers only via changes to the URL.
- Otherwise OPC UA works completely transparent to the API.

OPC UA Security



One of the most important benefits of eliminating the reliance on COM/DCOM technology is the expanded security features.

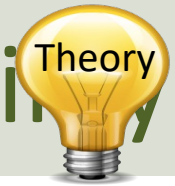


[Figure: http://www.ni.com/white-paper/13843/en/]

- OPC UA requires handshaking between clients and servers using X.509 Web standard certificates for authentication before they are able to talk with one another.
- To communicate between the server and client, the user can choose from three kinds of messaging modes: None, Sign, Sign and Encrypt.
- OPC UA can communicate through any standard HTTP or UA TCP port. Through this standardization, OPC UA can connect securely over a VPN and through firewalls to allow seamless, remote client-to-server connectivity.

http://www.ni.com/white-paper/13843/en/

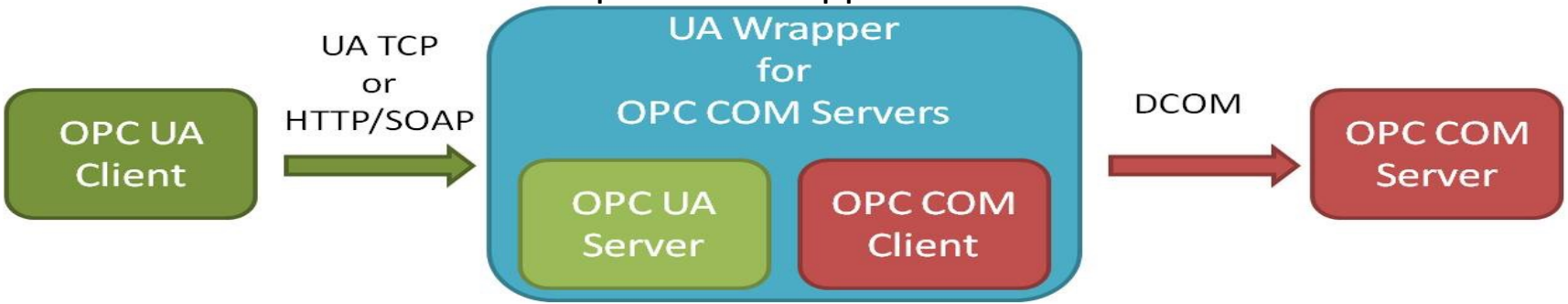
Classic OPC and OPC UA Compatibility



Because of the shift in data communication technology, the OPC UA protocol is **not** inherently backwards compatible with Classic OPC data access (DA) models!!

Classic OPC COM-based Clients require a UA Proxy to communicate with UA Servers:

Classic OPC COM-based Servers require UA Wrappers to interact with UA Clients:



http://www.ni.com/white-paper/13843/en/

https://www.halvorsen.blog

OPC UA Server Simulator

Free OPC UA Simulation Server from Integration Objects

Hans-Petter Halvorsen

Table of Contents

OPC UA Server Simulator

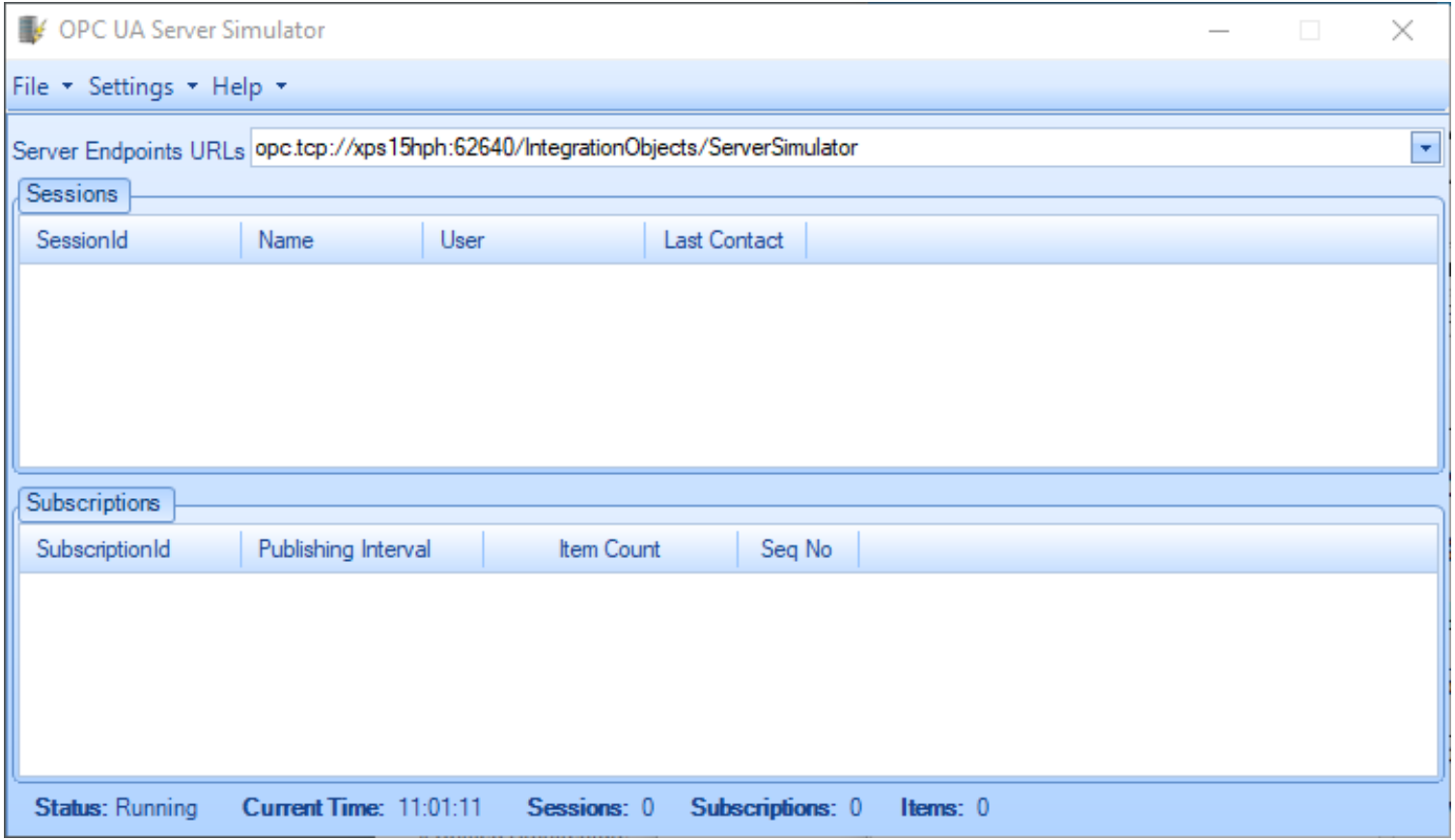
- This free OPC UA Server tool supports data access and historical access information models of OPC UA.
- Consequently, it provides simulated real-time and historical data.
- Moreover, users can configure their own tags and the data simulation via CSV files.
- OPC UA clients can monitor real-time data and explore history data from this simulator.
- https://opcfoundation.org/products/view/opc-ua-server-simulator

OPC UA Server Simulator

https://opcfoundation.org

https://opcfoundation.org/products/view/opc-ua-server-simulator

OPC UA Server Simulator



OPC UA Server Simulator

The OPC UA Server Simulator uses 2 CSV simulation files:

- "**AddressSpace.csv**" used to build the address space of the OPC UA Server.
- "**ValueSpace.csv**" used to simulate the data values of the OPC UA items.
- Those two files are located at the following path: X:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Server Simulator\OPC UA Server Simulator\DATA

“OPC UA Client” Tool

- “OPC UA Client” is a free client tool that supports the main OPC Unified Architecture information models.
- These models are Data Access, Alarms & Conditions, and Historical Data Access
- https://integrationobjects.com/sioth-opc/sioth-opc-unified-architecture/opc-ua-client/

- OPC Tunneling
- OPC UA
  - OPC UA Server Simulator – Full Edition
  - OPC UA Server Toolkit
  - OPC UA IoT Broker
  - OPC UA Server for Databases
  - OPC UA Client Toolkit
  - OPC UA Server Simulator
  - OPC UA Proxy
  - OPC UA Wrapper
  - OPC UA Client
- OPC Data Archiving
- OPC Clients
- OPC Servers
- OPC Client Toolkits
- OPC Free Tools
- OPC Server Toolkits

OPC UA Client

Download | User Guide | Quick User Guide

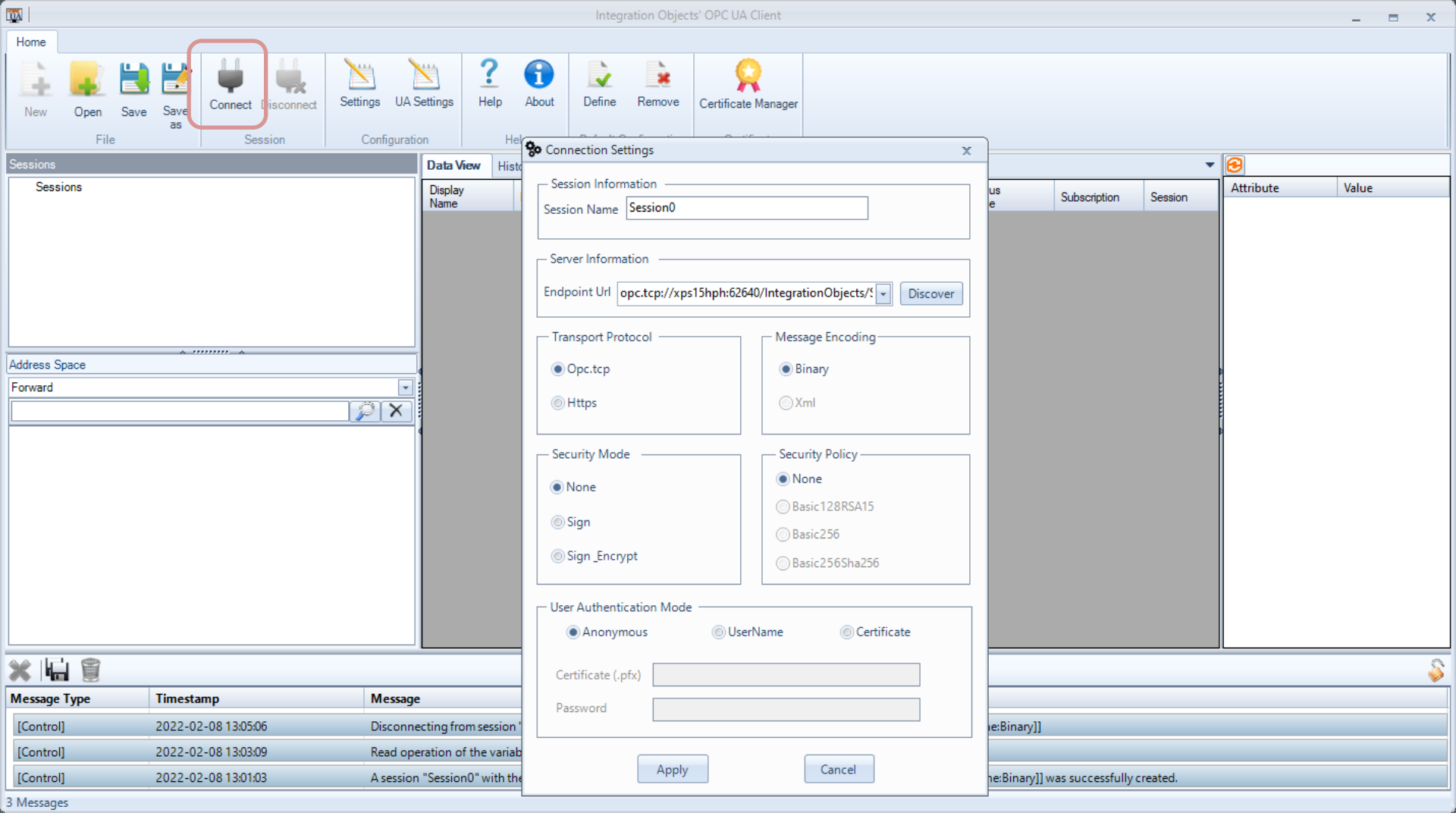
Download free OPC UA Client and start your OPC UA tests now!

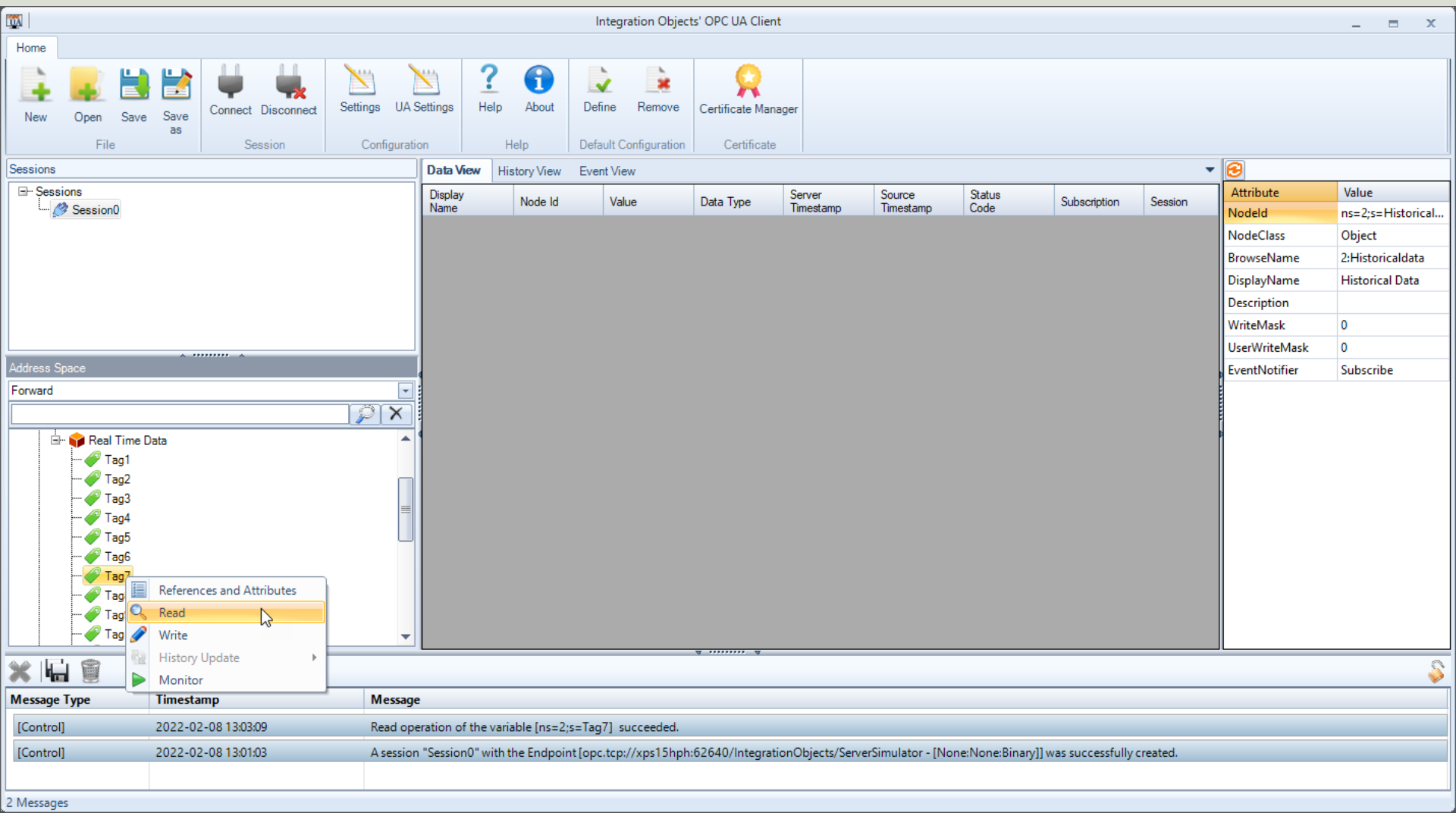
OPC UA Client is a free client tool that supports the main OPC Unified Architecture information models. These models are Data Access, Alarms & Conditions, and Historical Data Access. In fact, it offers the capability to:

- Discover local and remote OPC UA servers
- Establish secure communication channels
- Browse the address space of any OPC UA compliant server
- Monitor real-time data and alarms & conditions
- Explore and update history data

Moreover, this OPC UA explorer allows you to generate its self-signed Application Instance Certificate in order to provide application level security and secure the connections with OPC UA servers.

View Tutorial Video of OPC UA Test Client & OPC UA Wrapper


Integration Objects' OPC UA Client
Home
New
Open
Save
Save as
Connect
Disconnect
Settings
UA Settings
Help
About
Define
Remove
Certificate Manager
File
Session
Configuration
Sessions
Sessions
Address Space
Forward
Data View
Display Name
Subscription
Session
Attribute
Value
Connection Settings
Session Information
Session Name
Session0
Server Information
Endpoint Url
opc.tcp://xps15hph:62640/IntegrationObjects/S
Discover
Transport Protocol
Opc.tcp
Https
Message Encoding
Binary
Xml
Security Mode
None
Sign
Sign _Encrypt
Security Policy
None
Basic128RSA15
Basic256
Basic256Sha256
User Authentication Mode
Anonymous
UserName
Certificate
Certificate (.pfx)
Password
Apply
Cancel
Message Type
Timestamp
Message
[Control]
2022-02-08 13:05:06
Disconnecting from session
[Control]
2022-02-08 13:03:09
Read operation of the variab
[Control]
2022-02-08 13:01:03
A session "Session0" with the
:Binary]]
:Binary]] was successfully created.
3 Messages


Integration Objects' OPC UA Client
Home
New
Open
Save
Save as
File
Connect
Disconnect
Session
Settings
UA Settings
Configuration
Help
About
Help
Define
Remove
Default Configuration
Certificate Manager
Certificate
Sessions
Sessions
Session0
Address Space
Forward
Real Time Data
Tag1
Tag2
Tag3
Tag4
Tag5
Tag6
Tag7
References and Attributes
Read
Write
History Update
Monitor
Data View
History View
Event View
Display Name
Node Id
Value
Data Type
Server Timestamp
Source Timestamp
Status Code
Subscription
Session
Attribute
Value
NodeId
ns=2;s=Historical...
NodeClass
Object
BrowseName
2:Historicaldata
DisplayName
Historical Data
Description
WriteMask
0
UserWriteMask
0
EventNotifier
Subscribe
Message Type
Timestamp
Message
[Control]
2022-02-08 13:03:09
Read operation of the variable [ns=2;s=Tag7] succeeded.
[Control]
2022-02-08 13:01:03
A session "Session0" with the Endpoint [opc.tcp://xps15hph:62640/IntegrationObjects/ServerSimulator - [None:None:Binary]] was successfully created.
2 Messages

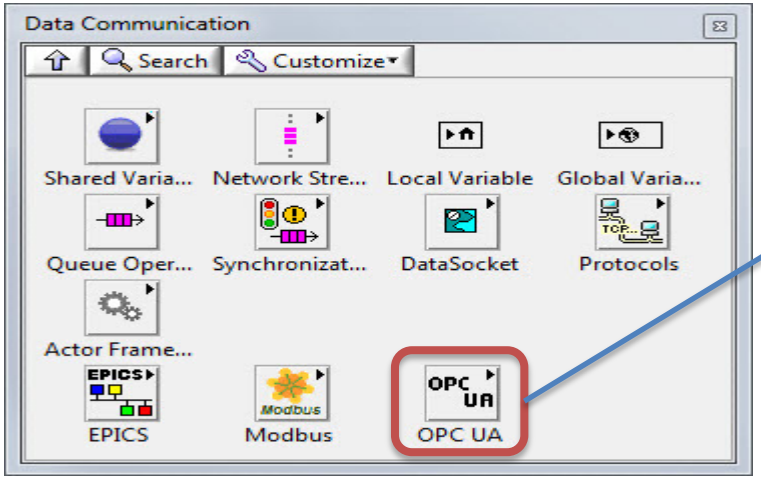
OPC UA in LabVIEW

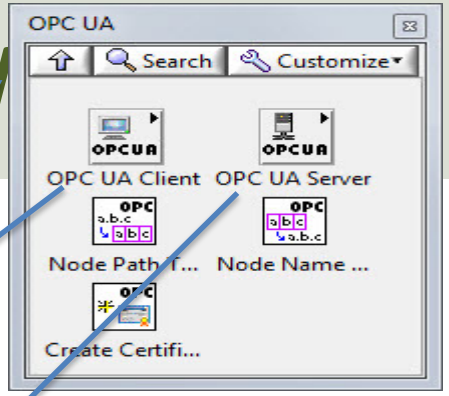
Hans-Petter Halvorsen

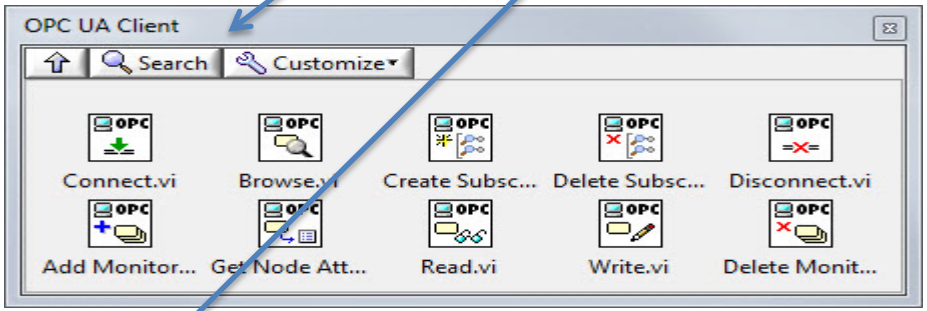
Table of Contents

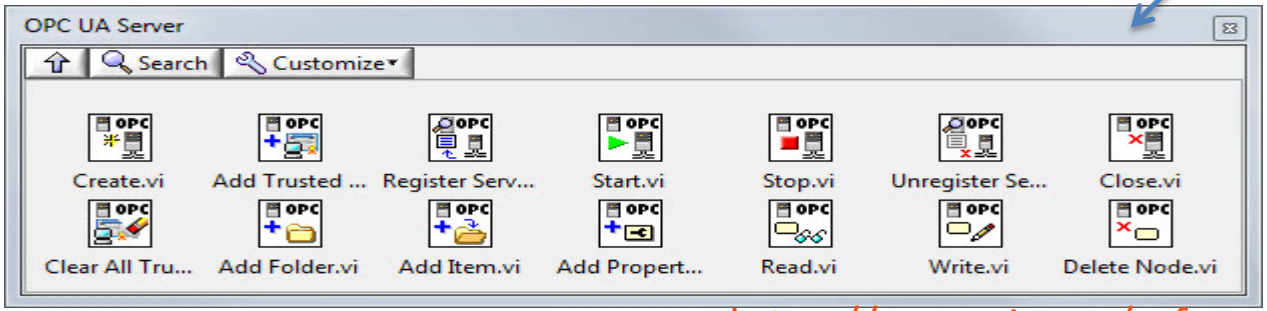
OPC UA in LabVIEW

http://zone.ni.com/reference/en-XX/help/371618J-1/TOC9.htm





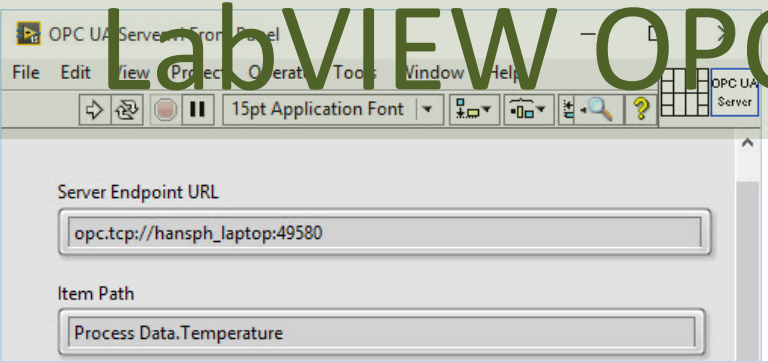


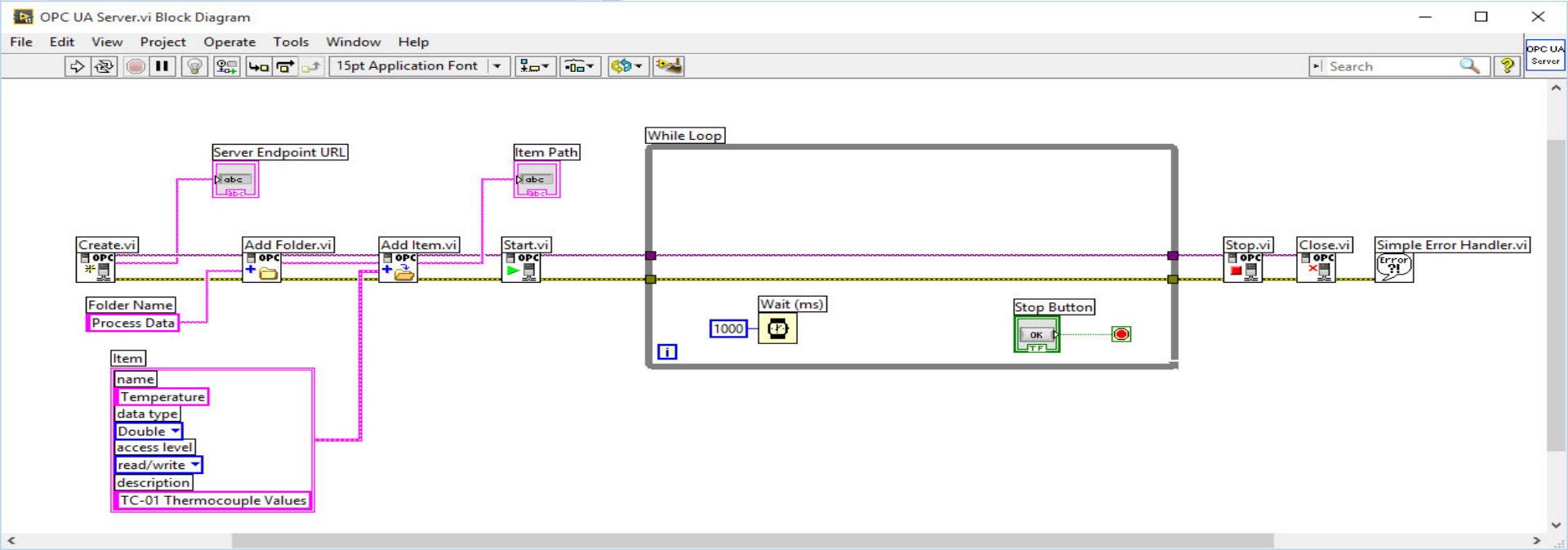


Note! You need to install the **LabVIEW OPC UA Toolkit**

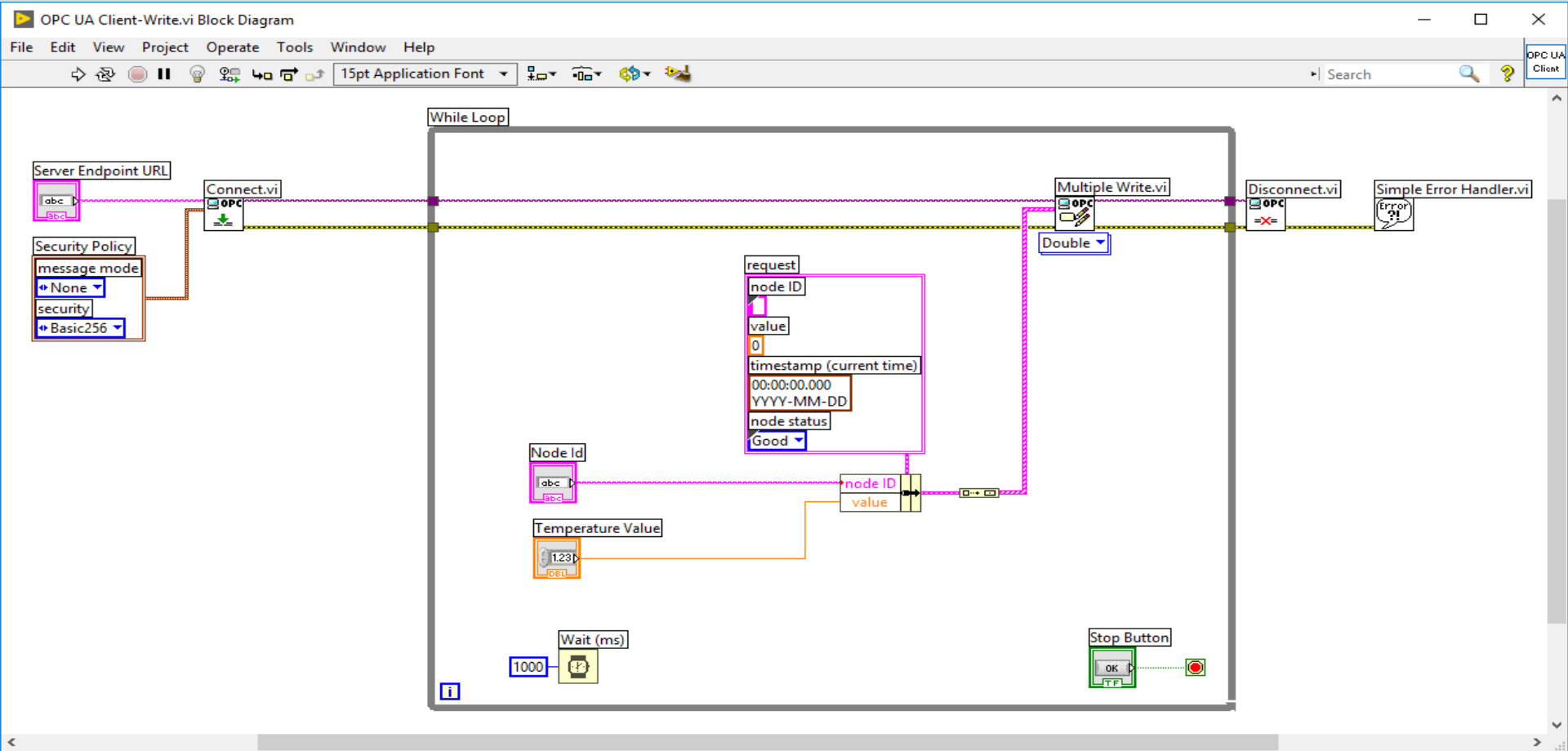
https://zone.ni.com/reference/en-XX/help/376230B-01/

LabVIEW OPC UA Server Example

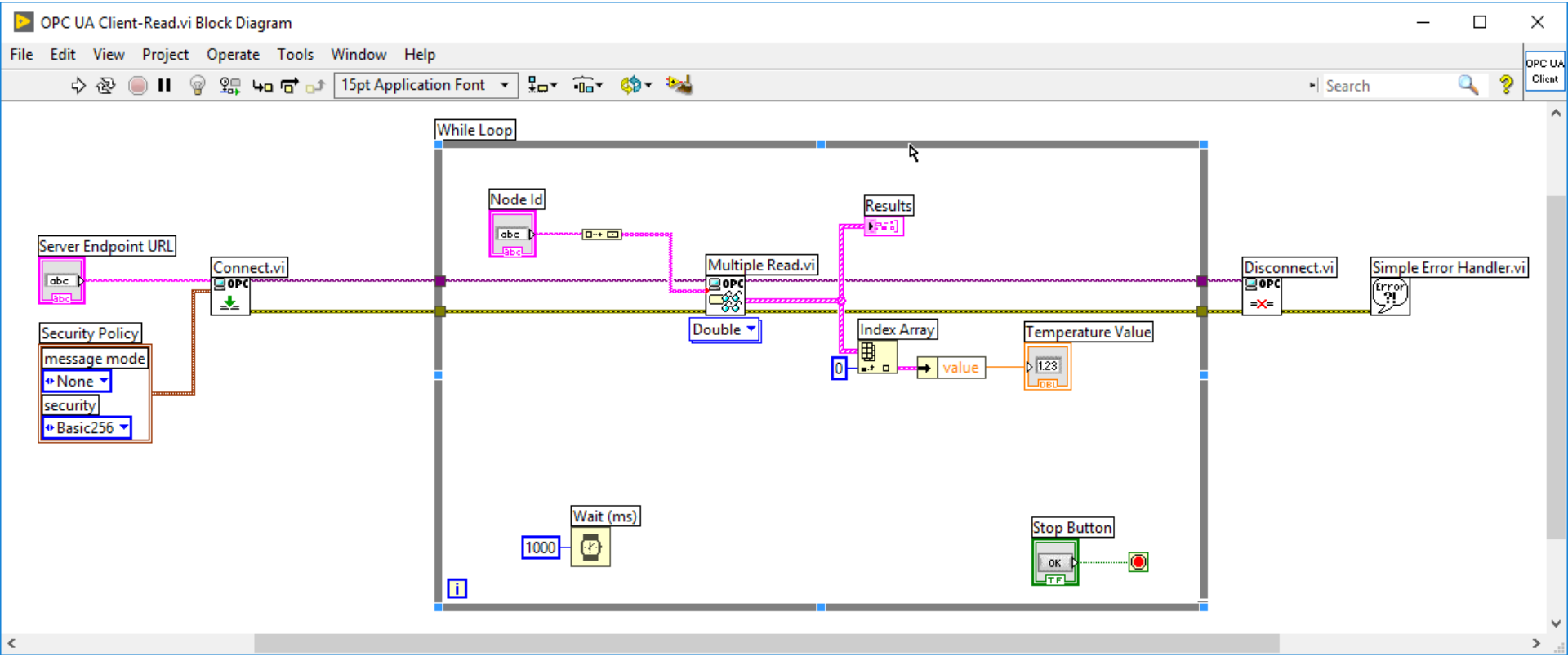




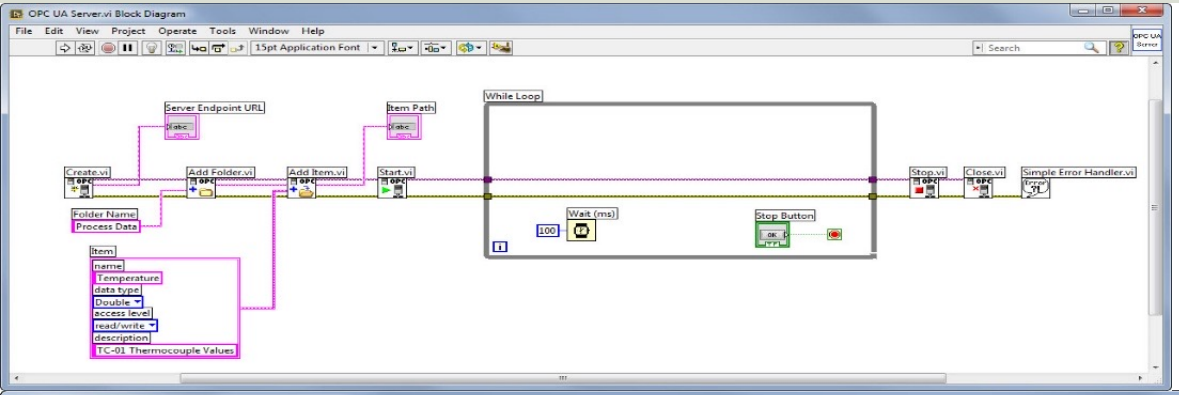
LabVIEW OPC UA Client - Write



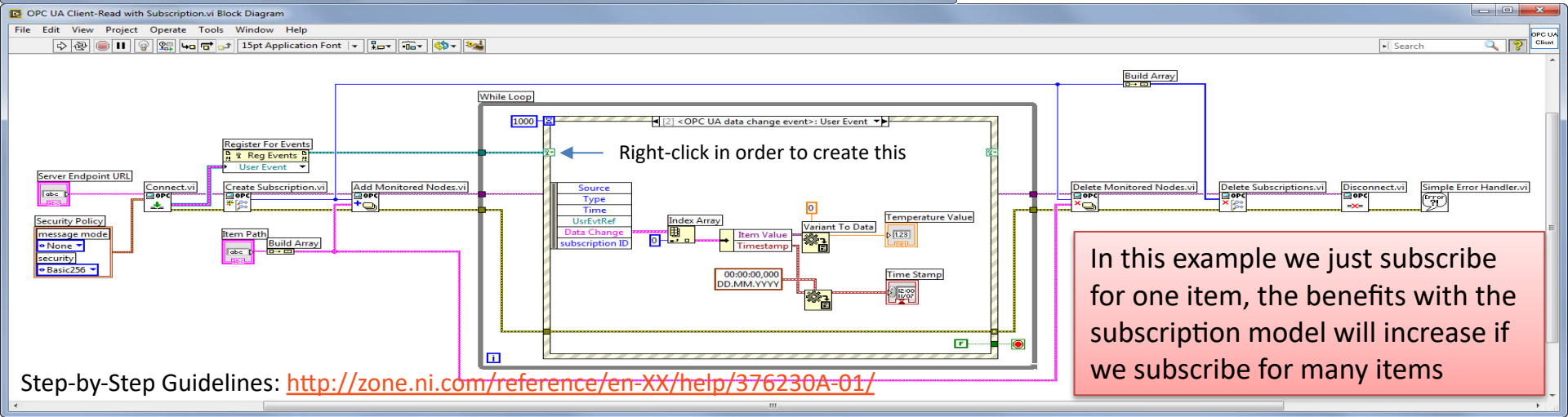
LabVIEW OPC UA Client - Read



OPC UA Client with Subscription



This is a more complex example where you read data on the client only when the value on the server is changed



In this example we just subscribe for one item, the benefits with the subscription model will increase if we subscribe for many items

Step-by-Step Guidelines: http://zone.ni.com/reference/en-XX/help/376230A-01/

https://www.halvorsen.blog

OPC UA in Visual Studio/C#

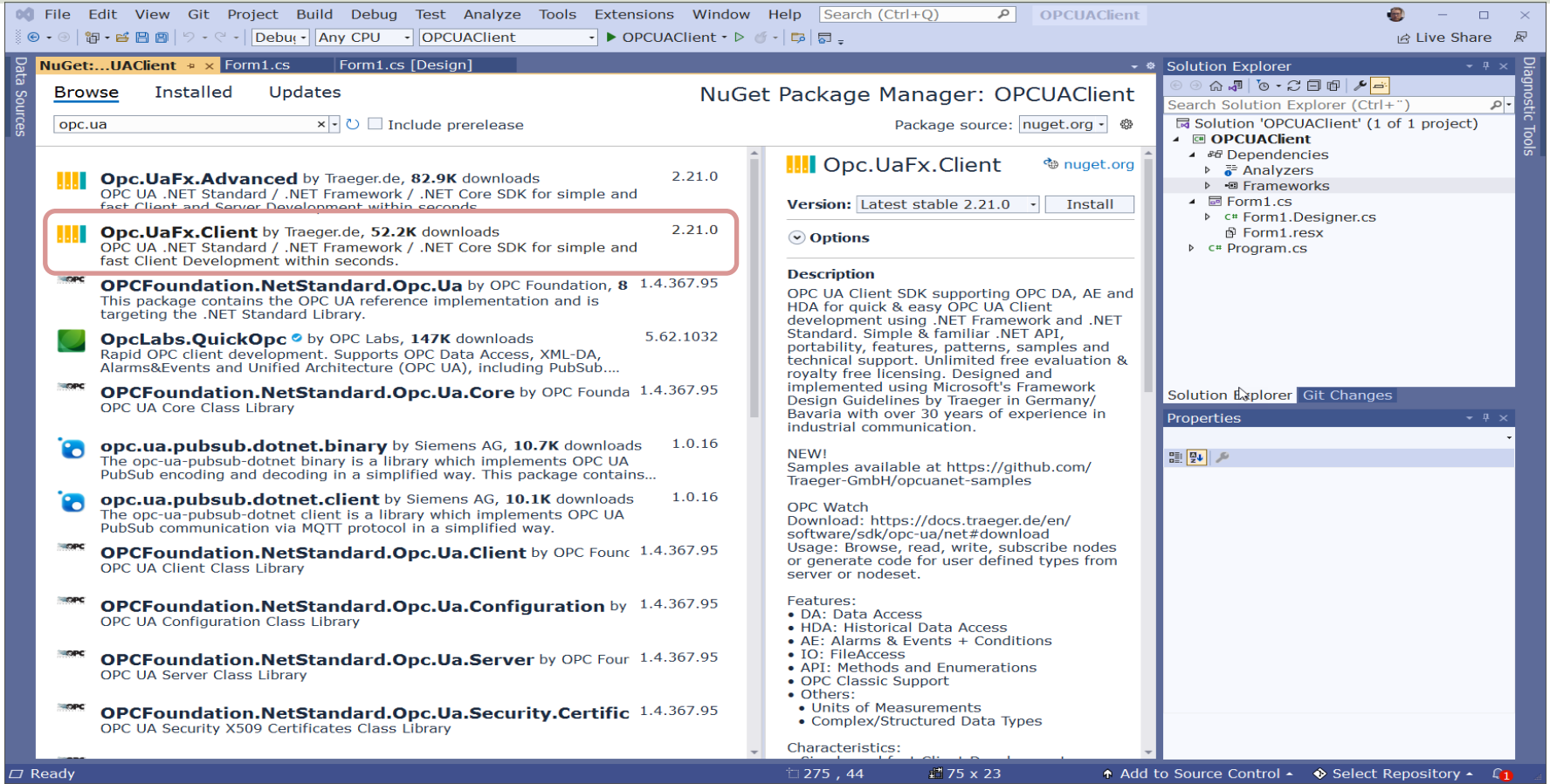
Hans-Petter Halvorsen

Table of Contents

OPC UA .NET SDK

- The “OPC UA .NET SDK” comes with an evaluation license which can be used unlimited for each application run for 30 minutes
- It comes in a **NuGet** Package you can install and use in your Visual Studio Project
- https://opcfoundation.org/products/view/opc-ua-net-sdk-for-client-and-server

NuGet Package



OPC UA Write Example

```
private void btnOpcWrite_Click(object sender, EventArgs e)
{
        string opcUrl = "opc.tcp://localhost:62640/";
        var tagName = "ns=2;s=Tag7";

        var client = new OpcClient(opcUrl);
        client.Connect();

        double temperature;
        temperature = Convert.ToDouble(txtOpcDataWrite.Text);

        client.WriteNode(tagName, temperature);

        client.Disconnect();
}
```

OPC UA Read Example

```
private void btnOpcRead_Click(object sender, EventArgs e)
{
    string opcUrl = "opc.tcp://localhost:62640/";
    var tagName = "ns=2;s=Tag7";

    var client = new OpcClient(opcUrl);
    client.Connect();

    var temperature = client.ReadNode(tagName);
    txtOpcDataRead.Text = temperature.ToString();

    client.Disconnect();
}
```

https://www.halvorsen.blog

OPC UA in MATLAB

Hans-Petter Halvorsen

Table of Contents

MATLAB OPC UA - Write

1. Locate Your OPC UA Server
   ```
   serverList = opcuaserverinfo('localhost')
   ```
2. Create an OPC UA Client
   ```
   uaClient = opcua('localhost', port)
   ```
3. Connect to the Server
   ```
   connect(uaClient)
   ```
4. Browse OPC UA Server Namespace
   ```
   serverNodes = browseNamespace(uaClient)
   ```
5. Write Current Values to the OPC UA Server
   ```
   newValue = 22.5
   writeValue(uaClient, serverNodes, newValue);
   ```
6. Disconnect
   ```
   disconnect(uaClient)
   ```

MATLAB OPC UA - Read

1. Locate Your OPC UA Server
   ```
   serverList = opcuaserverinfo('localhost')
   ```
2. Create an OPC UA Client
   ```
   uaClient = opcua('localhost', port)
   ```
3. Connect to the Server
   ```
   connect(uaClient)
   ```
4. Browse OPC UA Server Namespace
   ```
   serverNodes = browseNamespace(uaClient)
   ```
5. Read Current Values from the OPC UA Server
   ```
   [val,ts,qual] =
   readValue(uaClient,serverNodes)
   ```
6. Disconnect
   ```
   disconnect(uaClient)
   ```

OPC UA Scenario

This OPC UA Scenario shows multiple OPC UA Clients made with different Programming Languages where some Write Data and others Read Data from an OPC Server, e.g., “OPC UA Server Simulator” or “LabVIEW OPC UA Server”.

TC-01
Thermocouple

OPC UA
LabVIEW Client

Write

OPC UA
MATLAB Client

Write

OPC UA Server
Simulator

or

LabVIEW OPC
UA Server

Since OPC is an open standard, there is no problem using different programming languages or using different devices (e.g., PLCs) from different vendors.

OPC UA
LabVIEW Client

Read

Read

OPC UA
MATLAB Client

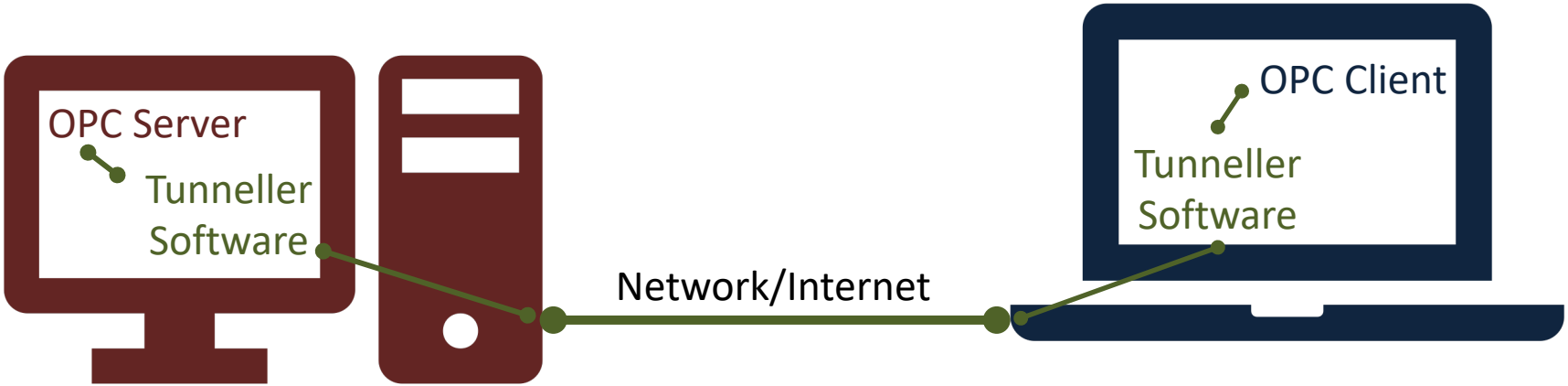
OPC in Network and Tunneling

Hans-Petter Halvorsen

Table of Contents

OPC Tunneller

Problem: Sending OPC Data between 2 (or more) Computers in a Network, or even worse, over Internet. OPC DA uses COM/DCOM. This makes it complicated to make it work in a modern Network



Solution: Use OPC Tunneller Software that makes an open tunnel between the 2 network nodes. The goal of OPC tunneling is to eliminate DCOM, i.e., replacing the DCOM networking protocol with TCP.

OPC DA in Network

- OPC DA uses COM/DCOM -> Complicated to make it work in a modern Network!!
- Solution: Use an **OPC Tunneller Software**, e.g.:
  - OPC Tunneller from MatrikonOPC (30 days free trial)
  - Cogent DataHub Tunnelling Software (Trial software works only 1 hour, then needs to be restarted)

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog