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 with main focus on OPC UA.
- 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. Make sure to Discuss the Results.
- 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

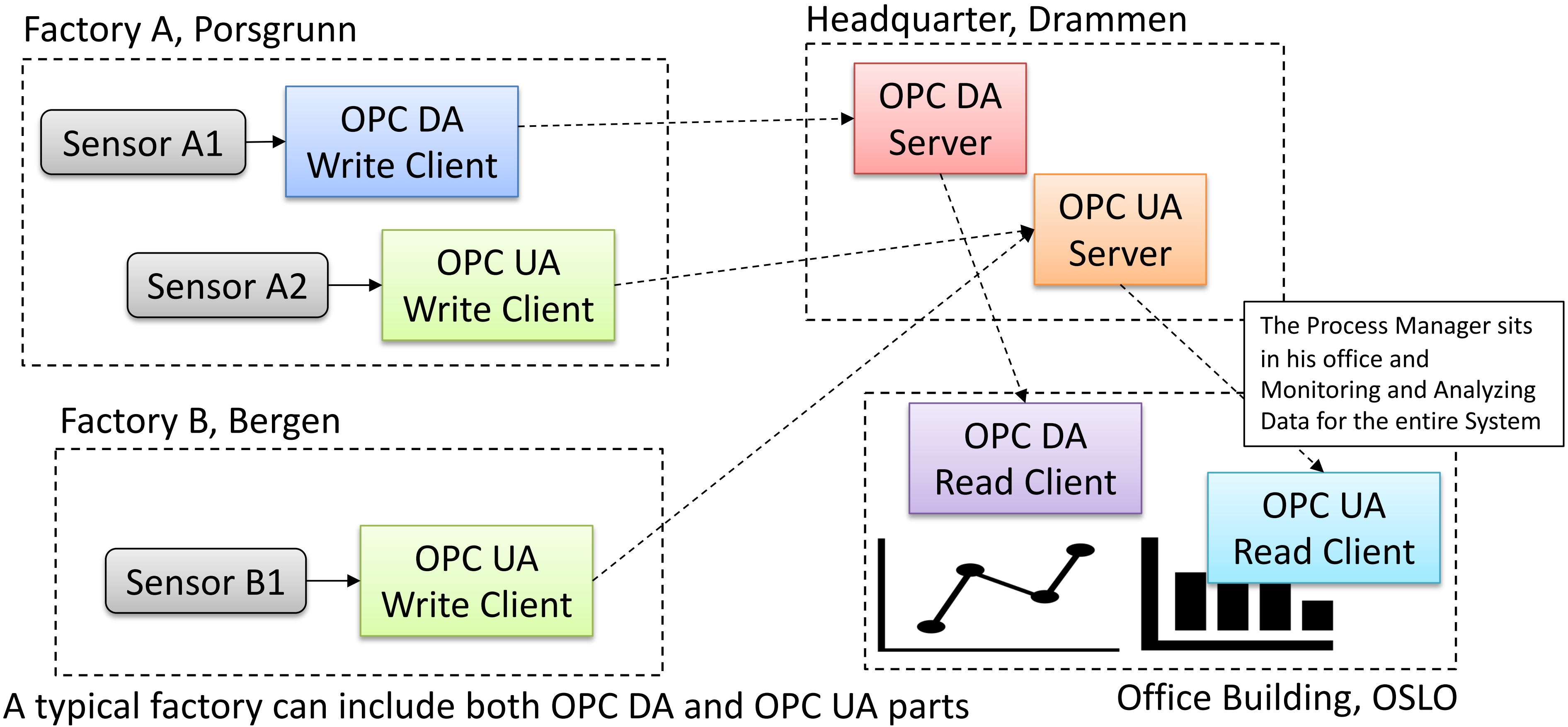
  - Create OPC DA Client Application in LabVIEW that reads data from one or more Temperature Sensors and write the Data to the Matrikon OPC DA Server.
    - The Application should at least include a features for Writing Data to a File and a Chart
    - Make sure to open the Data from the File and plot 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 either 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. It can be either OPC DA or OPC UA
  - 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. You can choose between using OPC DA or OPC UA

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.

Use Case Scenario for DAQ and OPC System



The Process Manager sits in his office and Monitoring and Analyzing Data for the entire System

A typical factory can include both OPC DA and OPC UA parts

https://www.halvorsen.blog

DAQ and OPC Resources

Hans-Petter Halvorsen

Table of Contents

Table of Contents

- Introduction
- TC-01 Thermocouple Temperature Sensor
- USB-6008
- Introduction to OPC
- OPC DA
  - MatrikonOPC Simulation Server (OPC DA Server)
  - OPC DA in LabVIEW - LabVIEW OPC DA – Write - LabVIEW OPC DA – Read
  - OPC DA in Visual Studio/C#
  - OPC DA in MATLAB
- OPC UA
  - OPC UA Server Simulator
  - OPC UA in LabVIEW
  - OPC UA in Visual Studio/C#
  - OPC UA in MATLAB
- OPC in a Network

Introduction

Learning Goals

- Learn key concepts within DAQ
- 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

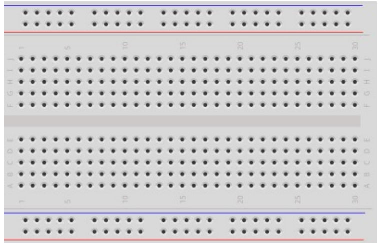
- LabVIEW
  - DAQmx Driver Software
  - LabVIEW OPC UA Toolkit
- MatrikonOPC Simulation Server
- OPC UA Server Simulator
- MATLAB
  - MATLAB Industrial Communication Toolbox
- Visual Studio
  - Measurement Studio
- OPC Tunneller Software
  - OPC Tunneller from MatrikonOPC
    or
  - Cogent DataHub Tunnelling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



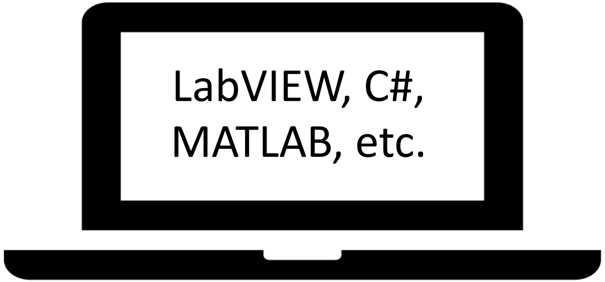
Breadboard

TMP36

Thermistor

Switch

Your Personal Computer



LabVIEW, C#, MATLAB, etc.

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

My System is not Working??

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

NATIONAL INSTRUMENTS LabVIEW

“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.

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

https://www.halvorsen.blog

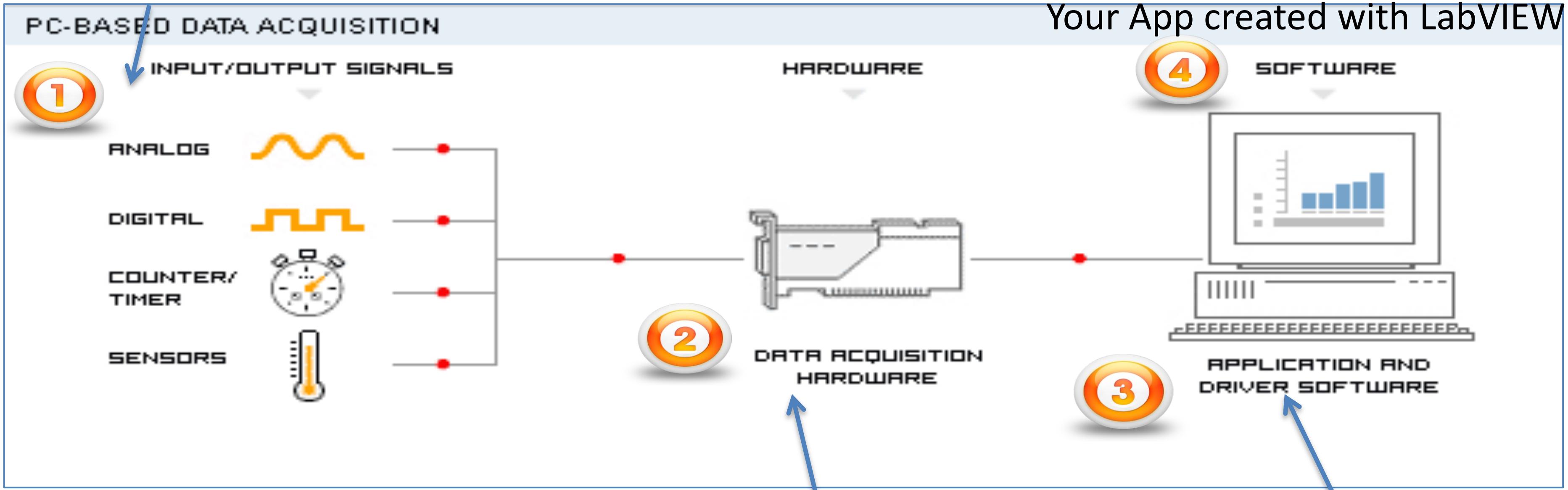
DAQ with TC-01 Thermocouple

Hans-Petter Halvorsen

Table of Contents

Data Acquisition (DAQ)

Sensors, etc.



Your App created with LabVIEW

NI TC-01 Thermocouple Device

NI DAQmx Driver

<u>A DAQ System consists of 4 parts:</u>

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

TC-01 Thermocouple Sensor

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

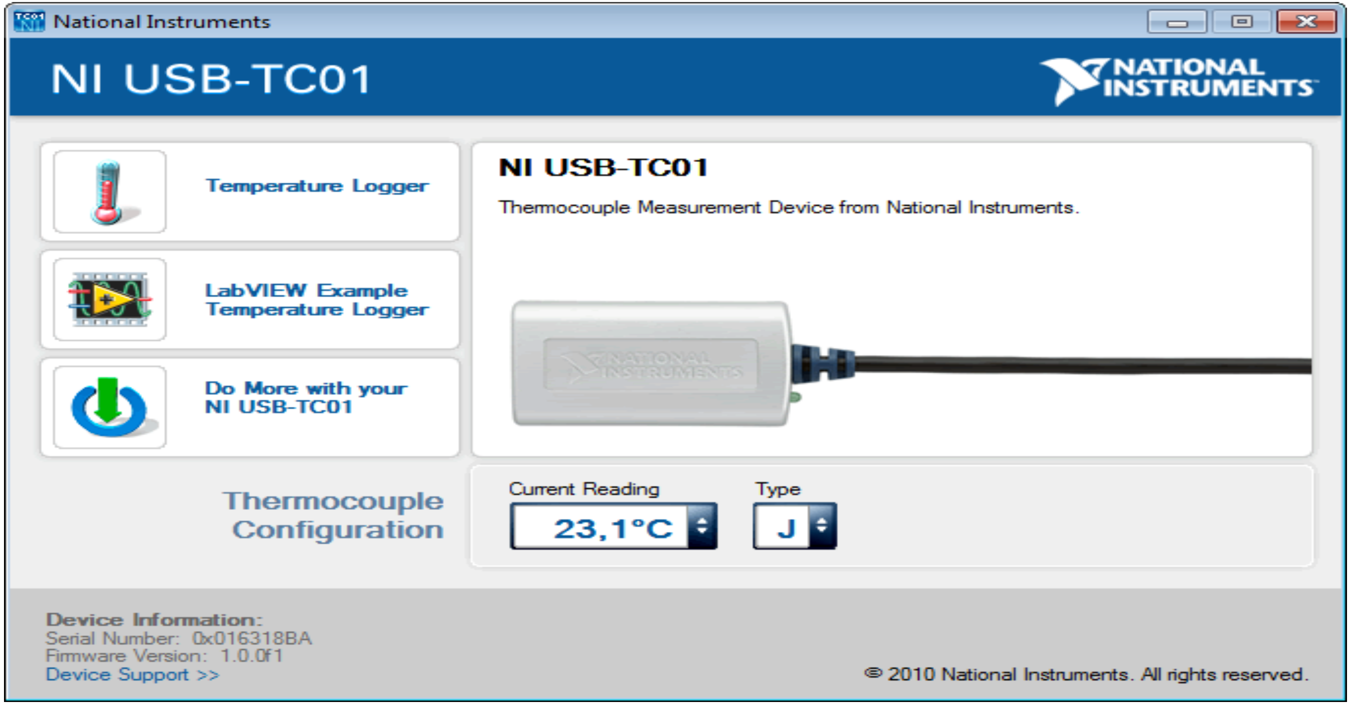
Sample Rate: 4 Samples/S

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**”):



National Instruments

NI USB-TC01

LabVIEW Example Temperature Logger
Do More with your NI USB-TC01

NI USB-TC01

Thermocouple Measurement Device from National Instruments.

Thermocouple Configuration

Current Reading: 23,1°C

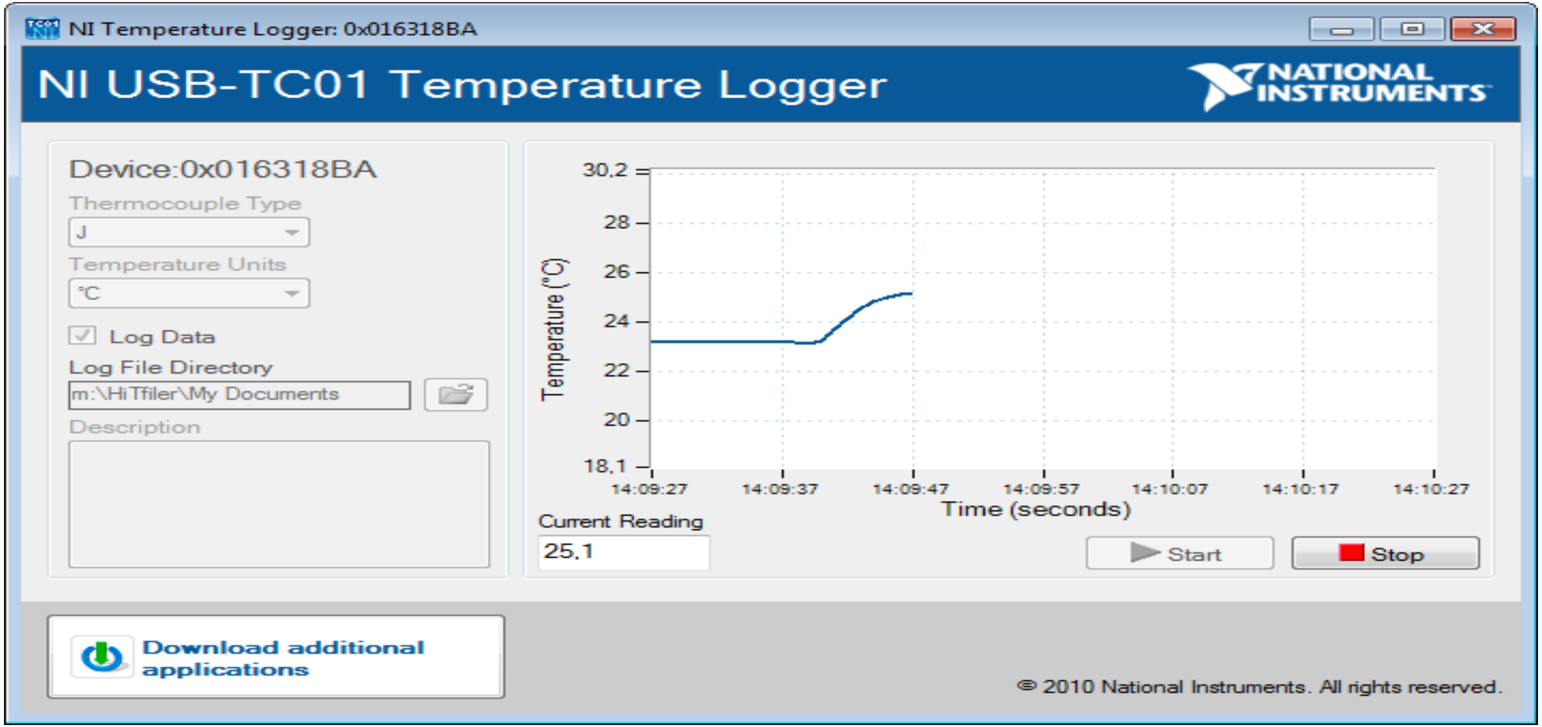
Type: J

Device Information:
Serial Number: 0x016318BA
Firmware Version: 1.0.0f1
Device Support >>

© 2010 National Instruments. All rights reserved.

Built-in Temperature Logger

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



NI Temperature Logger: 0x016318BA

NI USB-TC01 Temperature Logger

Device:0x016318BA

Thermocouple Type: J

Temperature Units: °C

☑ Log Data

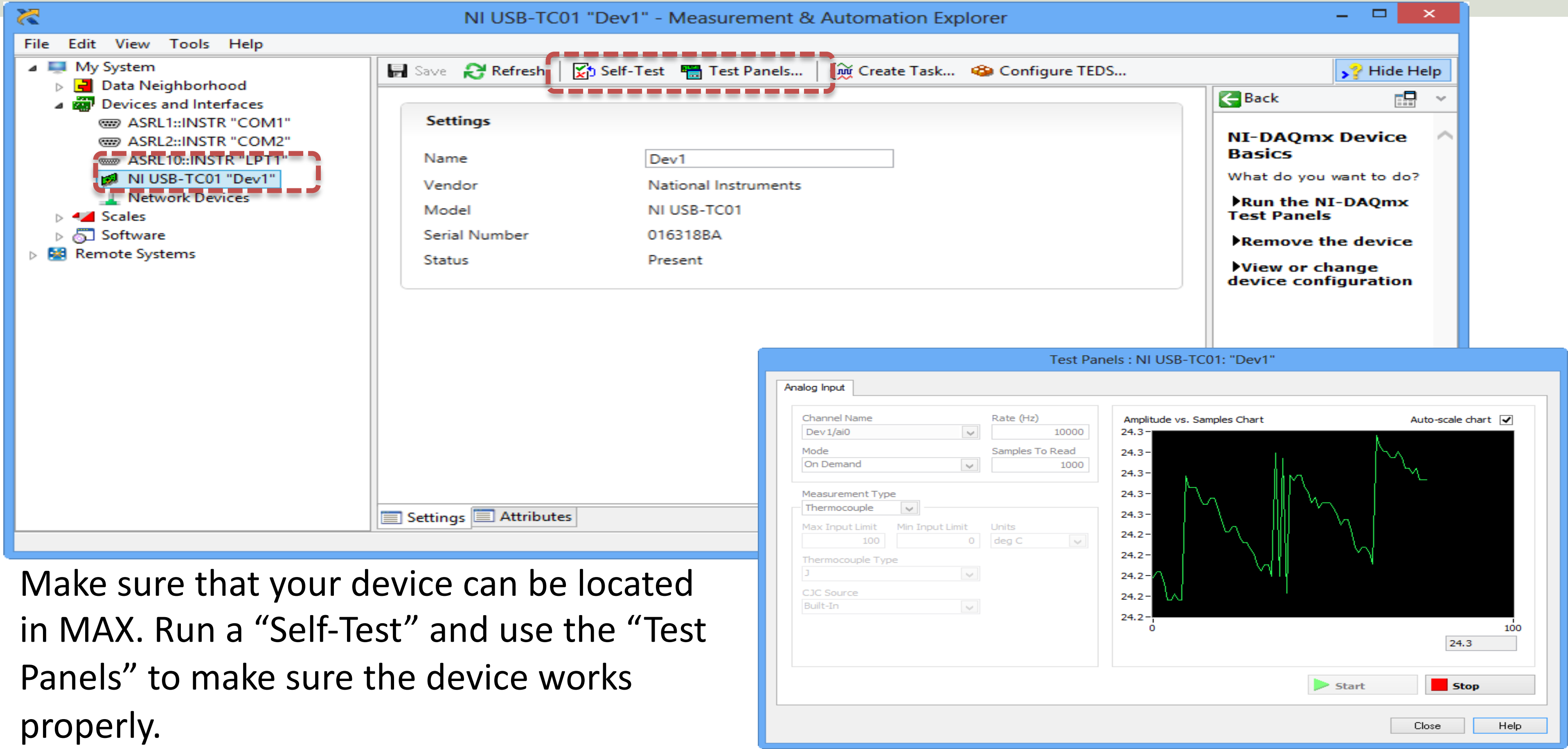
Log File Directory: m:\HiTfiler\My Documents

Description

Current Reading: 25,1

© 2010 National Instruments. All rights reserved.

Measurement & Automation Explorer (MAX)



NI USB-TC01 "Dev1" - Measurement & Automation Explorer

Settings

Name	Dev1
Vendor	National Instruments
Model	NI USB-TC01
Serial Number	016318BA
Status	Present

NI-DAQmx Device Basics

What do you want to do?

- Run the NI-DAQmx Test Panels
- Remove the device
- View or change device configuration

Test Panels : NI USB-TC01: "Dev1"

Analog Input

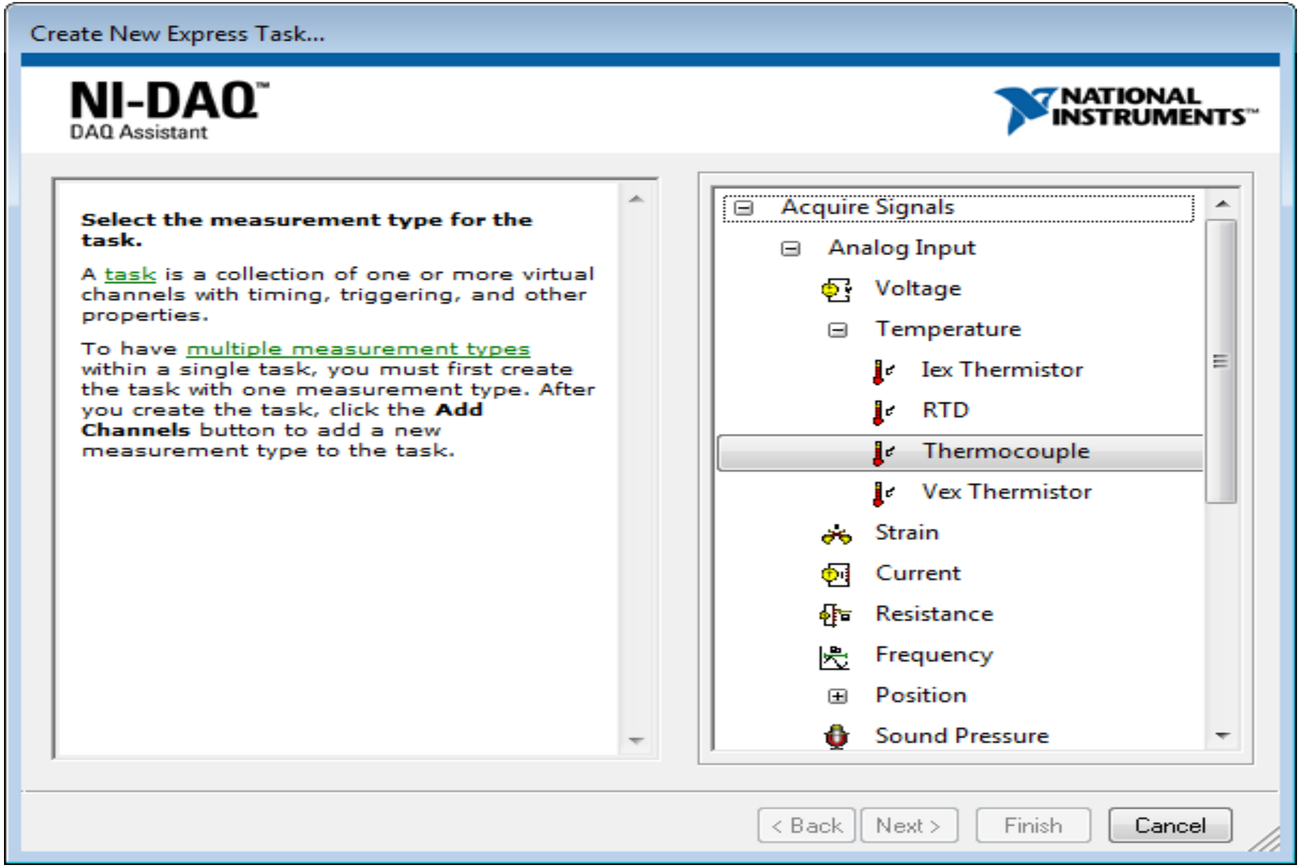
Channel Name	Dev1/ai0
Rate (Hz)	10000
Mode	On Demand
Samples To Read	1000
Measurement Type	Thermocouple
Max Input Limit	100
Min Input Limit	0
Units	deg C
Thermocouple Type	J
CJC Source	Built-In

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

DAQ Assistant
data

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.

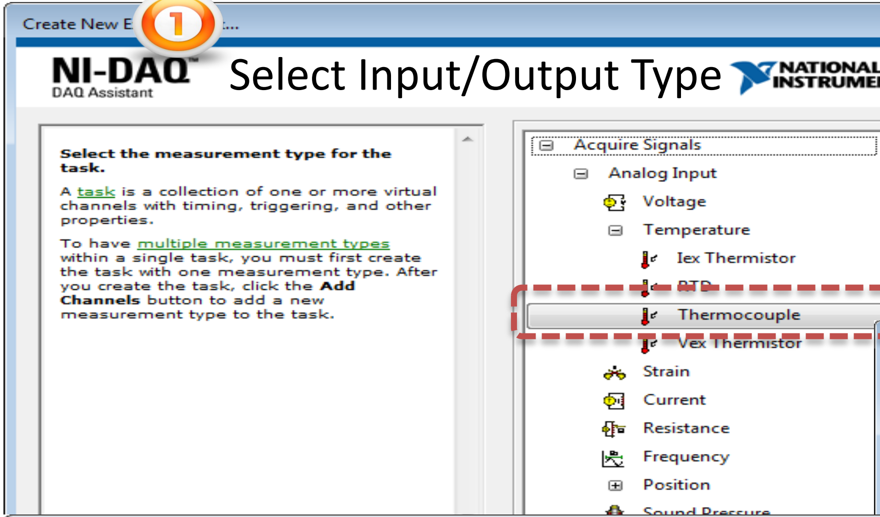


Select the measurement type for the task.

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.

LabVIEW DAQ Assistant



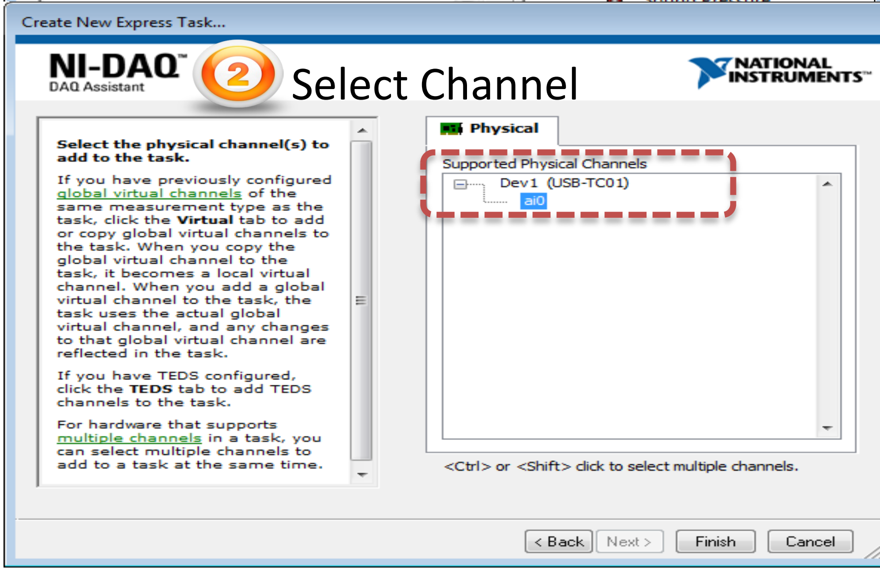
1 Select Input/Output Type

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
      - Iex Thermistor
      - RTD
      - Thermocouple
      - Vex Thermistor
    - Strain
    - Current
    - Resistance
    - Frequency
    - Position
    - Sound Pressure



2 Select Channel

Select the physical channel(s) to add to the task.

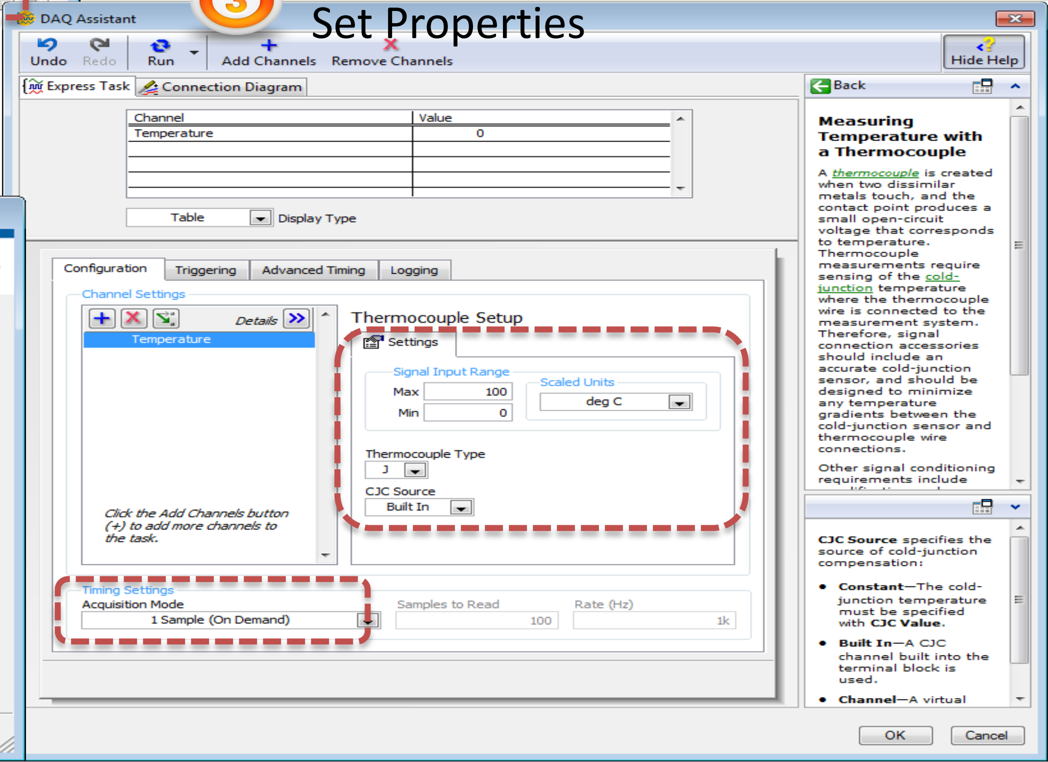
If you have previously configured global virtual channels of the same measurement type as the task, click the **Virtual** tab to add or copy global virtual channels to the task. When you copy the global virtual channel to the task, it becomes a local virtual channel. When you add a global virtual channel to the task, the task uses the actual global virtual channel, and any changes to that global virtual channel are reflected in the task.

If you have TEDS configured, click the **TEDS** tab to add TEDS channels to the task.

For hardware that supports multiple channels in a task, you can select multiple channels to add to a task at the same time.

Supported Physical Channels
- Dev1 (USB-TC01)
  - ai0

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



3 Set Properties

Channel	Value
Temperature	0

Display Type: Table

Channel Settings: Temperature

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

Thermocouple Setup

Signal Input Range: Max 100, Min 0

Scaled Units: deg C

Thermocouple Type: J

CJC Source: Built In

Timing Settings

Acquisition Mode: 1 Sample (On Demand)

Samples to Read: 100

Rate (Hz): 1k

Measuring Temperature with a Thermocouple

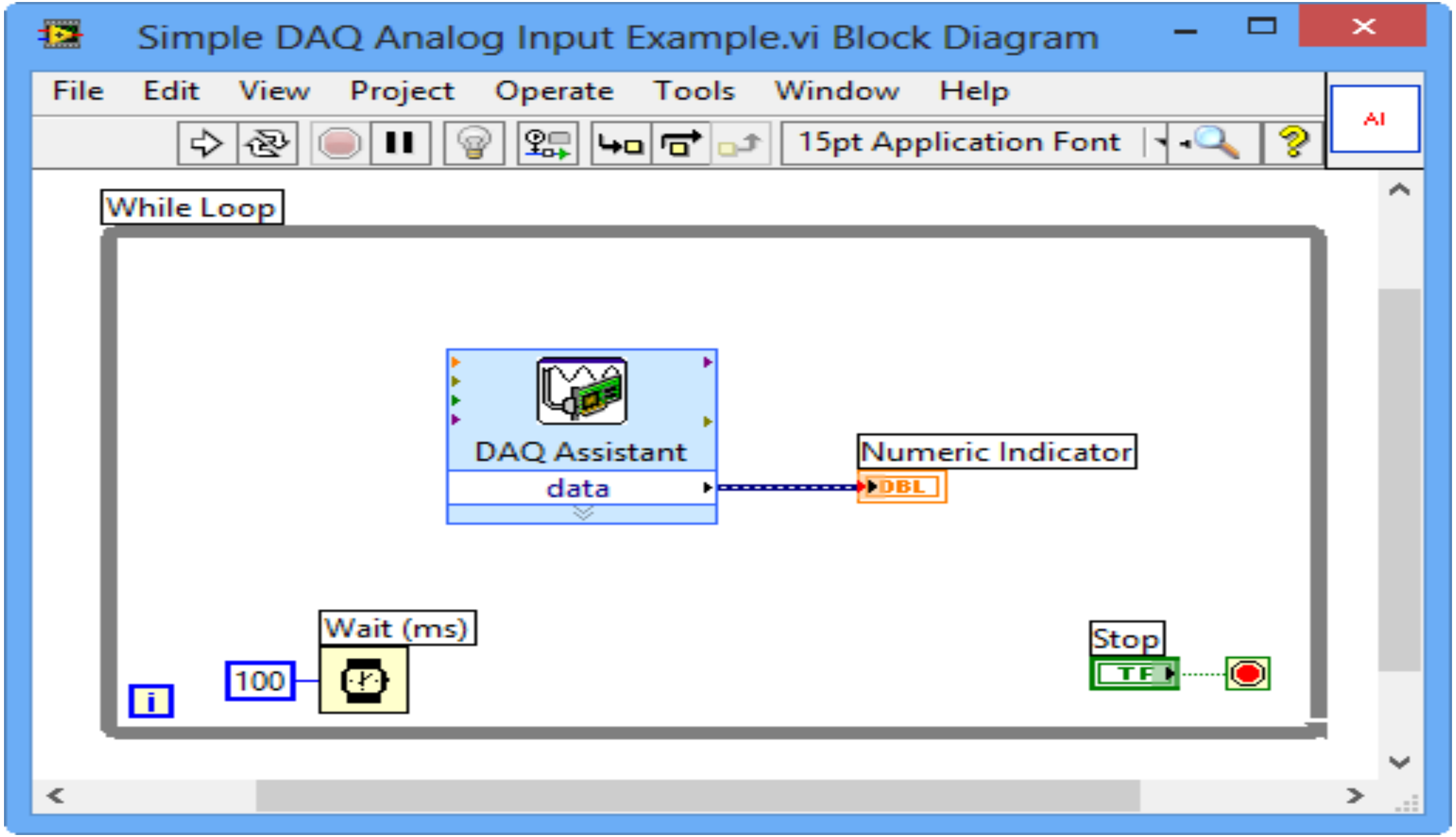
A thermocouple is created when two dissimilar metals touch, and the contact point produces a small open-circuit voltage that corresponds to temperature. Thermocouple measurements require sensing of the cold-junction temperature where the thermocouple wire is connected to the measurement system. Therefore, signal connection accessories should include an accurate cold-junction sensor, and should be designed to minimize any temperature gradients between the cold-junction sensor and thermocouple wire connections.

Other signal conditioning requirements include

CJC Source specifies the source of cold-junction compensation:

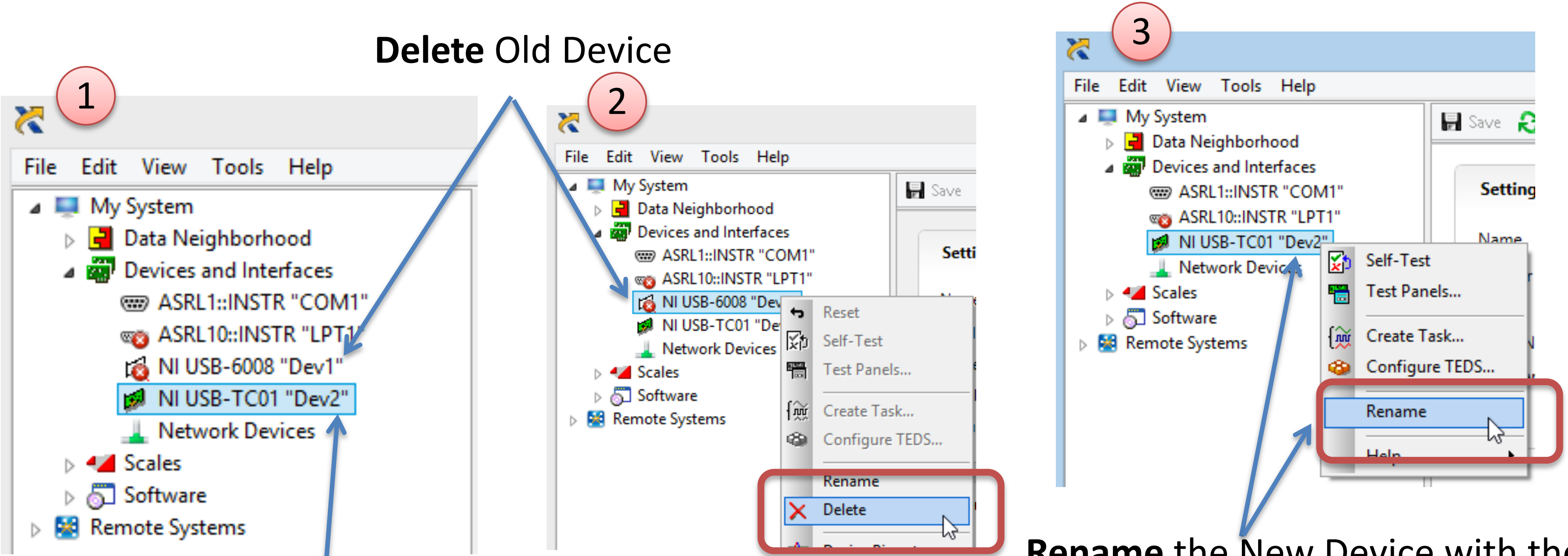
- **Constant**—The cold-junction temperature must be specified with **CJC Value**.
- **Built In**—A CJC channel built into the terminal block is used.
- **Channel**—A virtual

Read Data from TC-01 Device


Simple DAQ Analog Input Example.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
AI
While Loop
DAQ Assistant
data
Numeric Indicator
DBL
Wait (ms)
100
i
Stop
TF

Not working after you got a new Device?

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



Delete Old Device

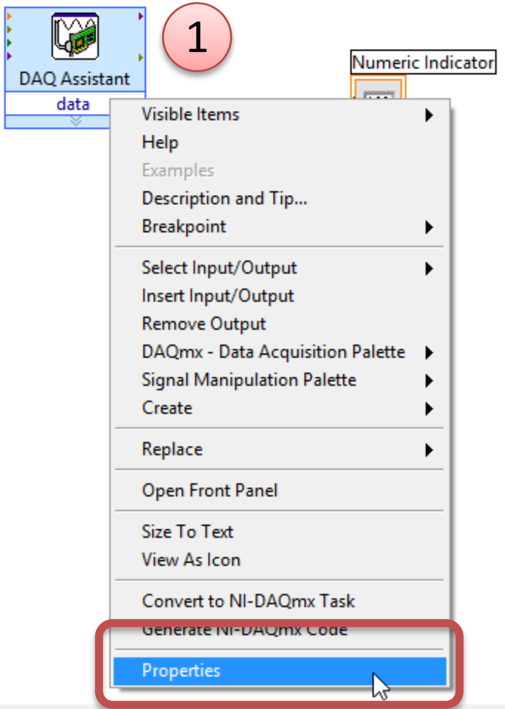
New Device

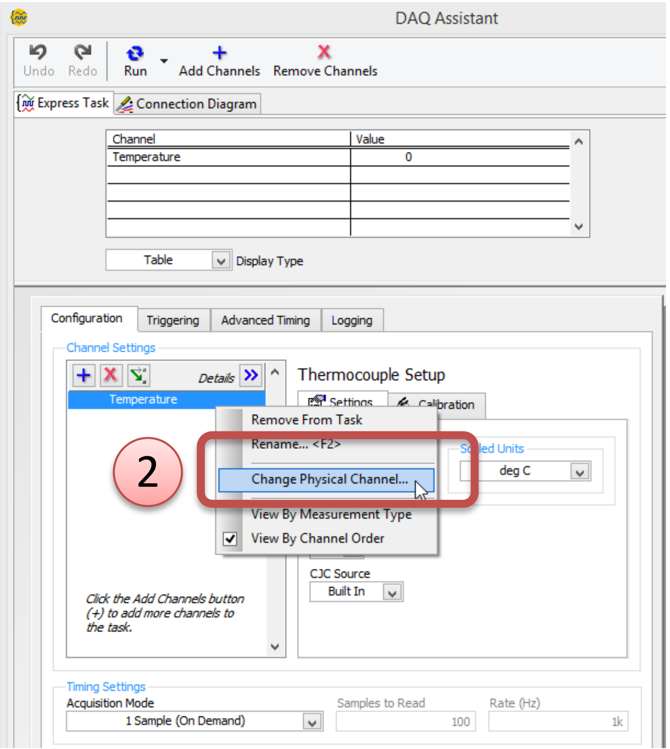
Rename the New Device with the same Name as the Old one

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”





Channel	Value
Temperature	0

Display Type: Table

Temperature

Scaled Units: deg C

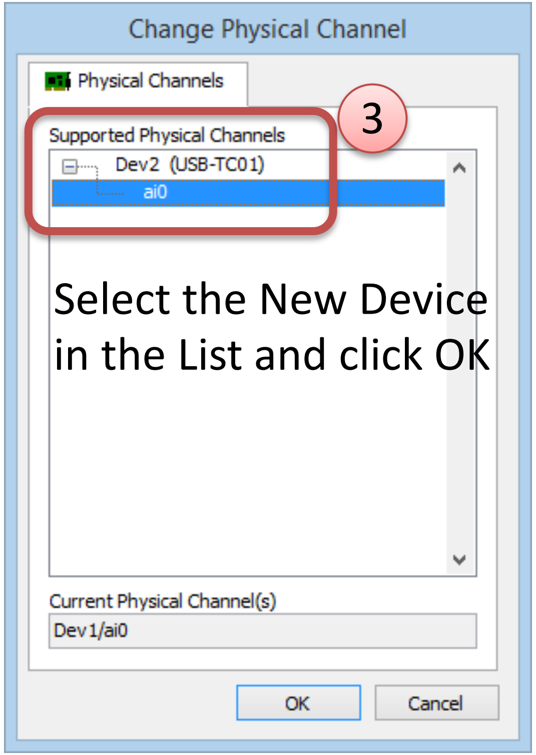
CJC Source: Built In

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

Acquisition Mode: 1 Sample (On Demand)

Samples to Read: 100

Rate (Hz): 1k



Current Physical Channel(s): Dev1/ai0

Select the New Device in the List and click OK

DAQ with USB-6008

Hans-Petter Halvorsen

Table of Contents

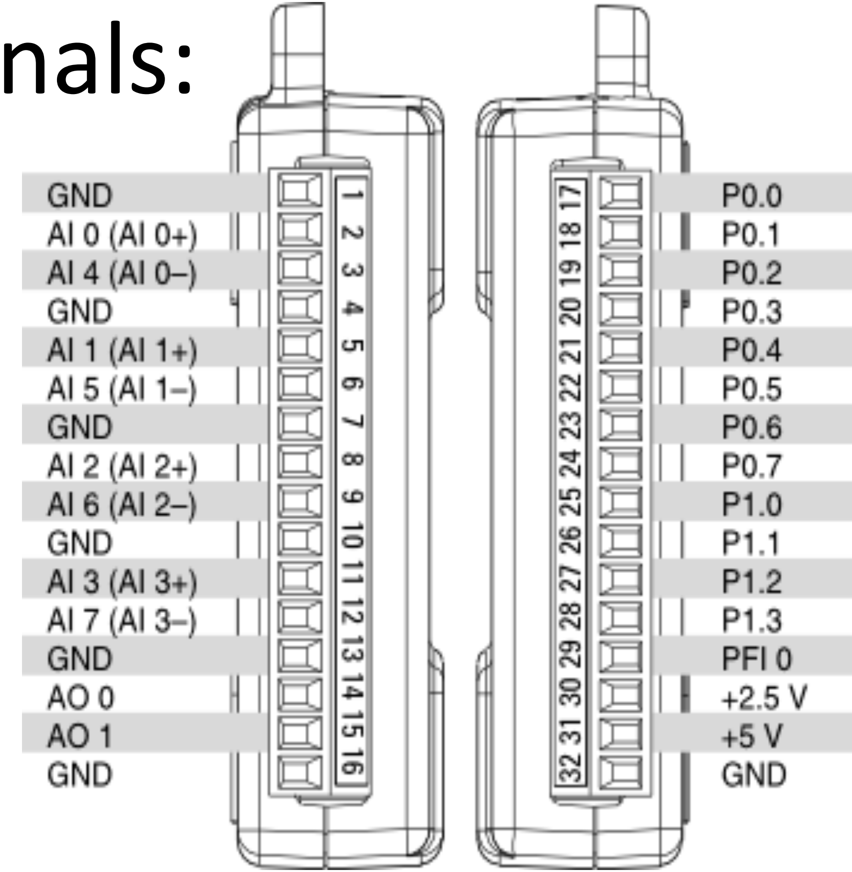
USB-6008

- 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



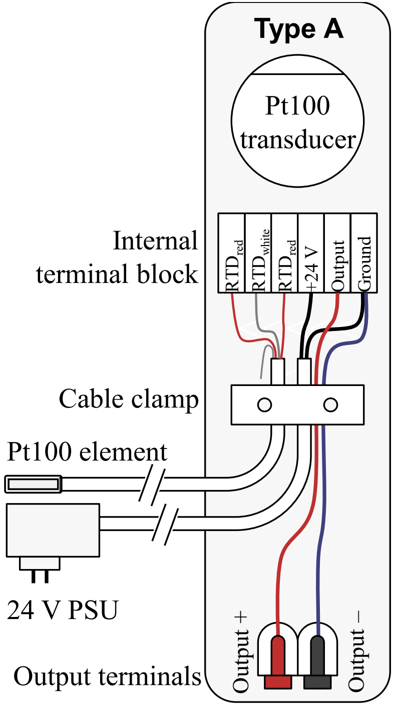
Temperature Sensors

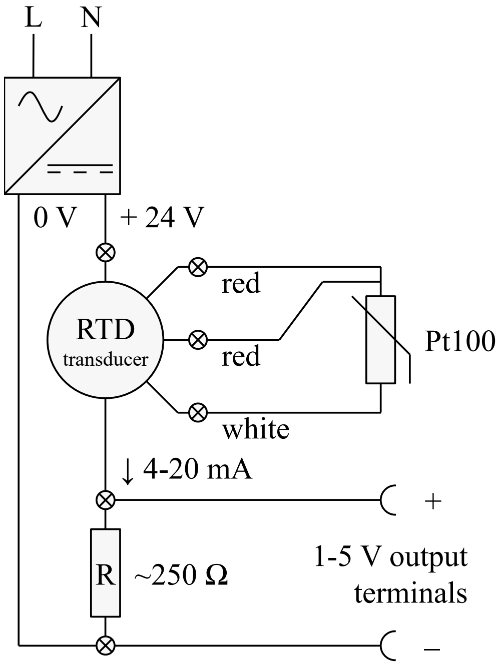
In the Laboratory we have different types of Temperature Sensors that we can connect to the USB-6008 DAQ device:

- PT-100
  - A Pt100 element is a RTD that uses platinum (Pt) as the resistor element. A Pt100 element is calibrated so that a temperature of $0°C$ yields a resistance of exactly 100Ω.
- TMP36
  - It provides a voltage output that is linearly proportional to the Celsius temperature.
- Thermistor
  - A thermistor is an electronic component that changes resistance to temperature - so-called Resistance Temperature Detectors (RTD).

PT-100

In the Laboratory we have a PT-100 device with Power Supply and PT-100 transducer:

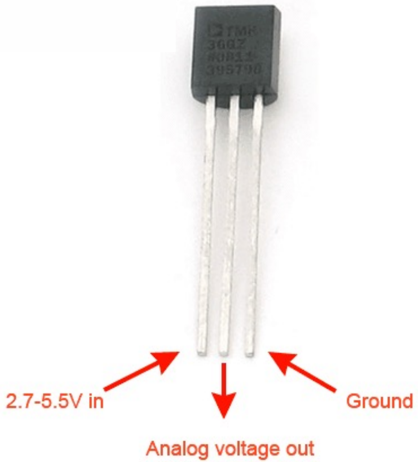


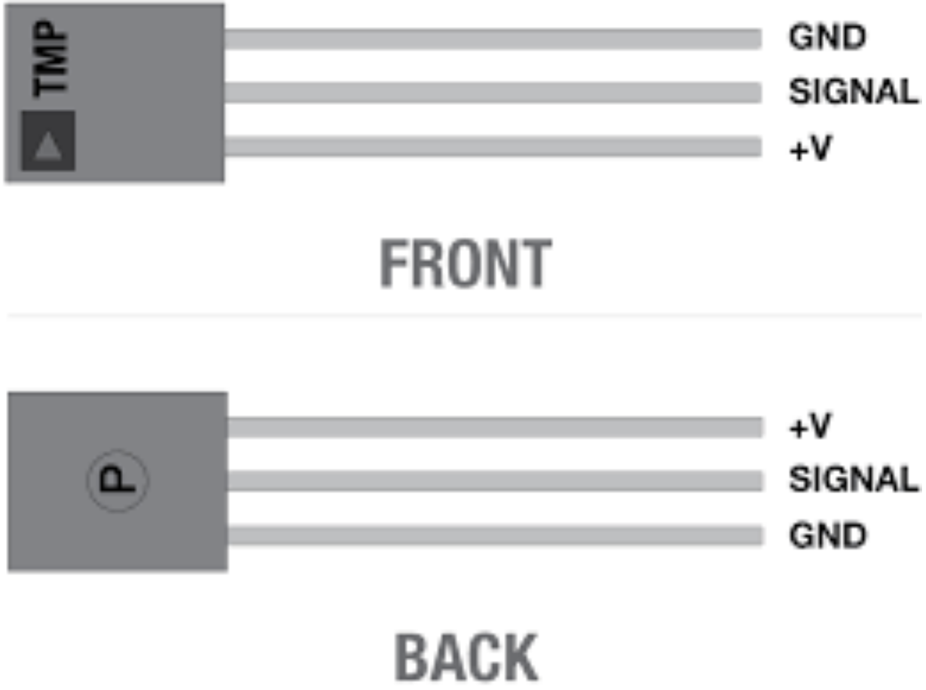


You must scale the output to a temperature value in degree Celsius

The PT-100 device can easily be connected to the USB-6008 DAQ device

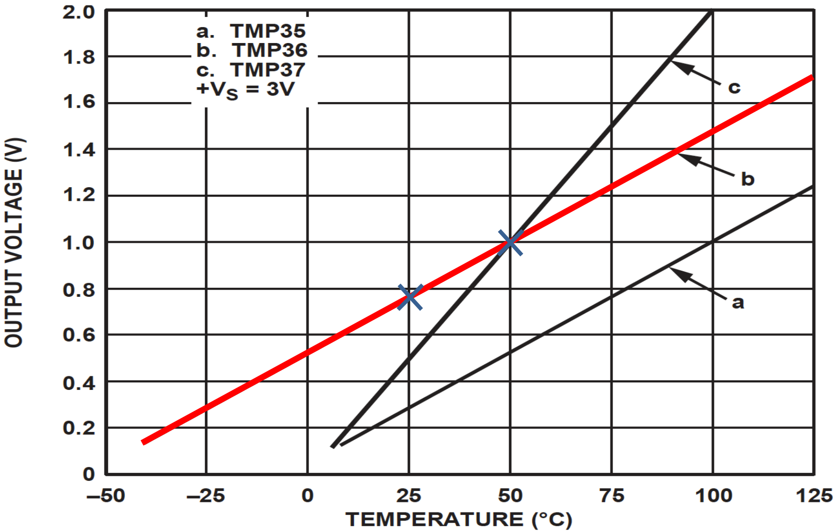
TMP36





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

Linear Scaling



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:

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

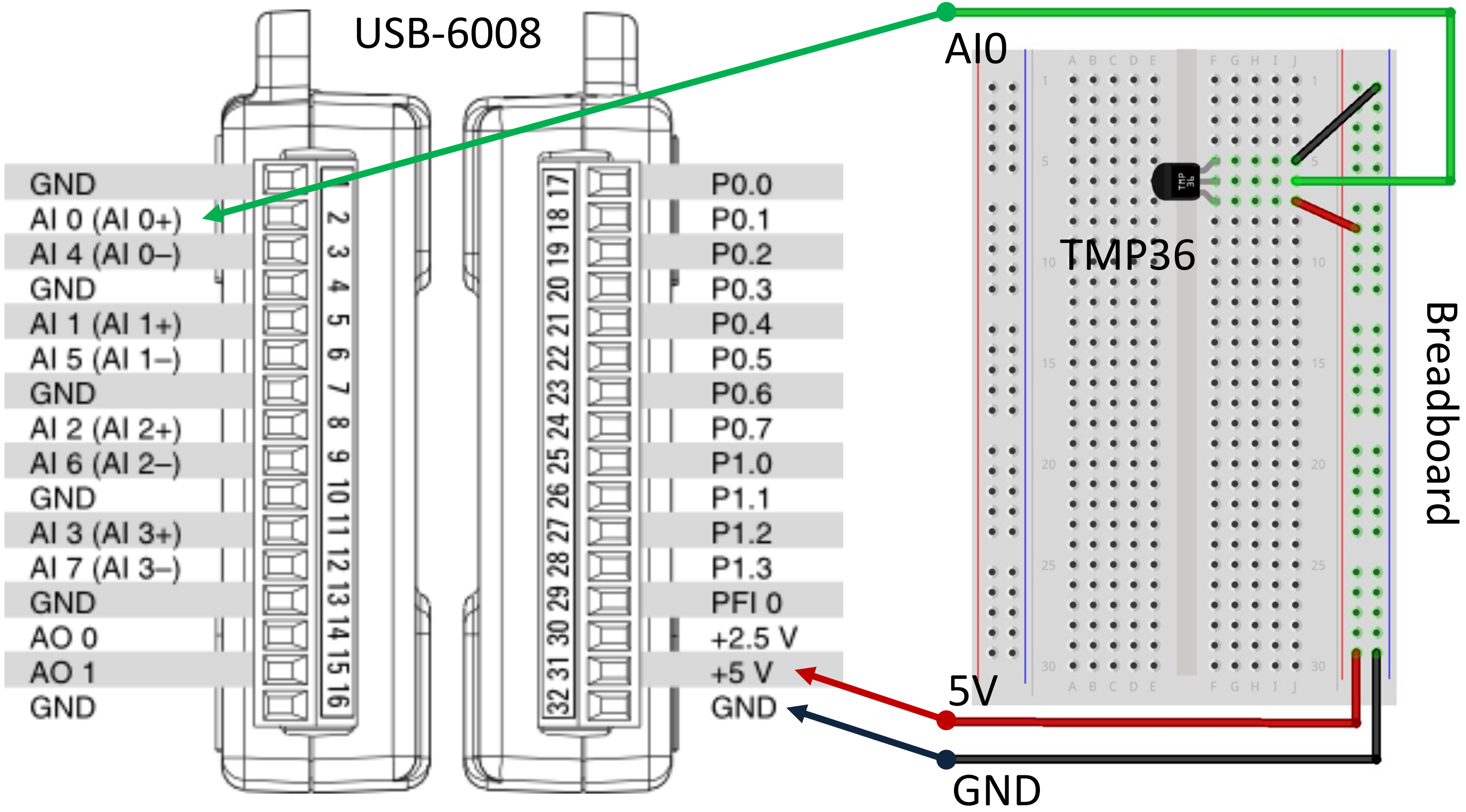
This gives:

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

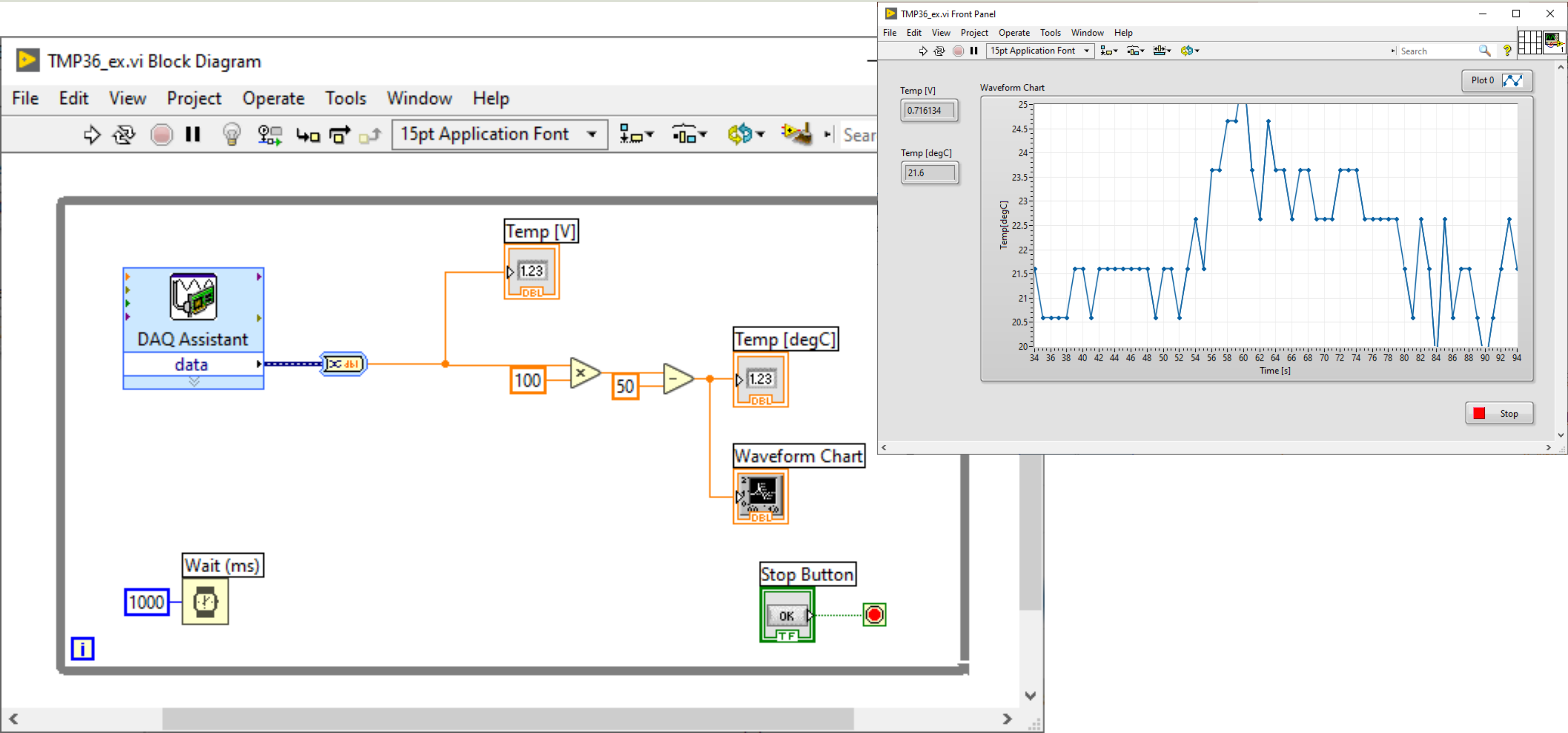
Then we get the following formula:

$$y = 100x - 50$$

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



Temp [V] 0.716134

Temp [degC] 21.6

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**):

[Wikipedia]

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

where A, B, C are constants given below

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

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

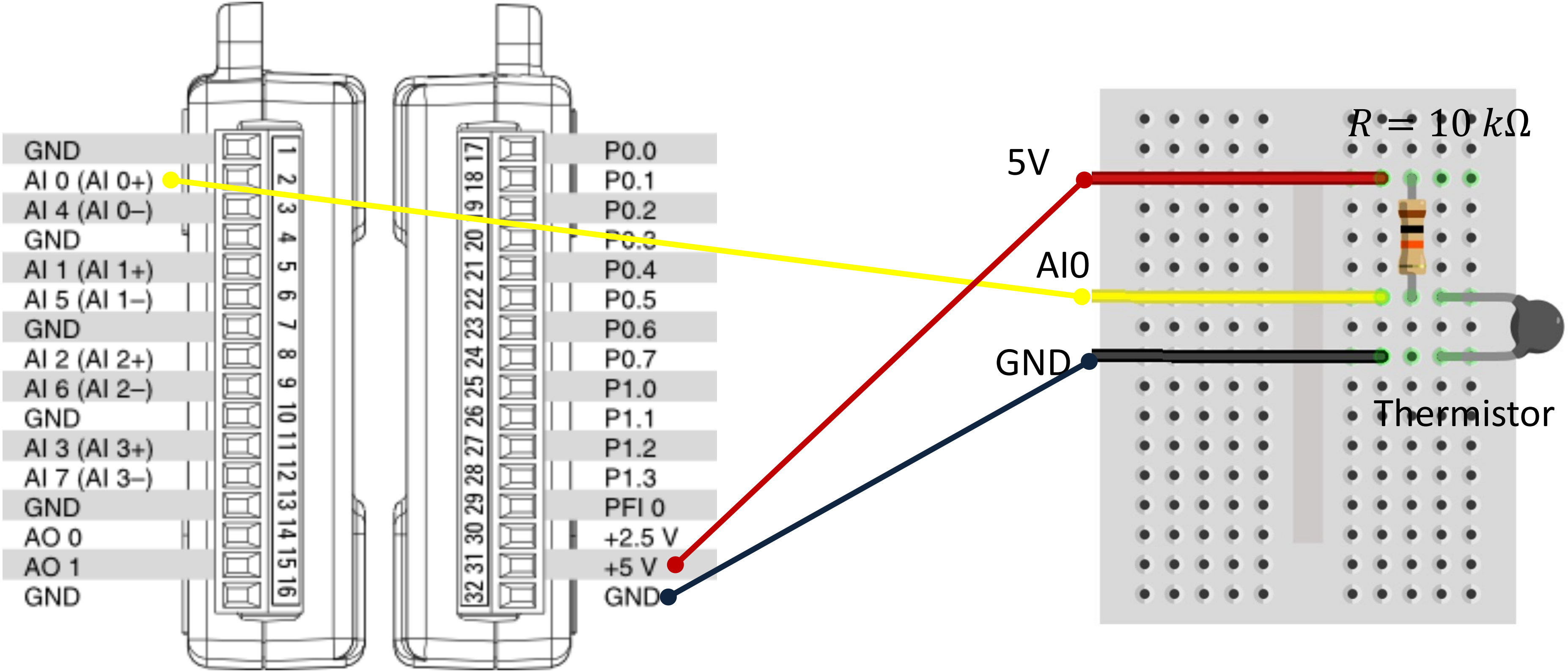
$A, B \ and \ C$ are constants

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

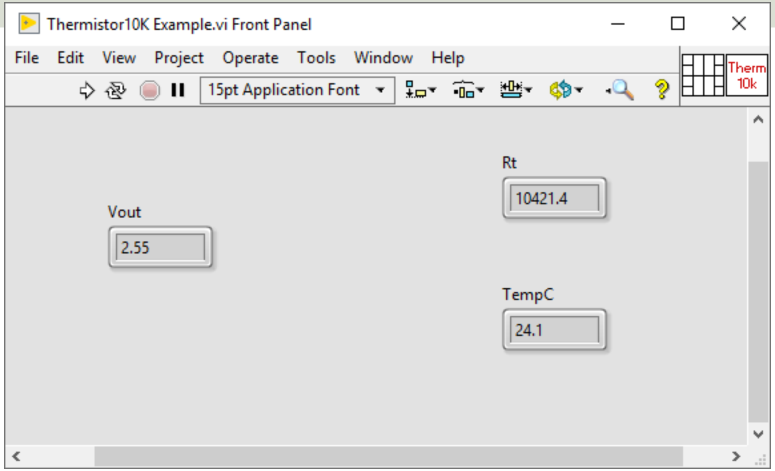
The Temperature in degrees Celsius will then be:

$$T_C = T_K - 273.15$$

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

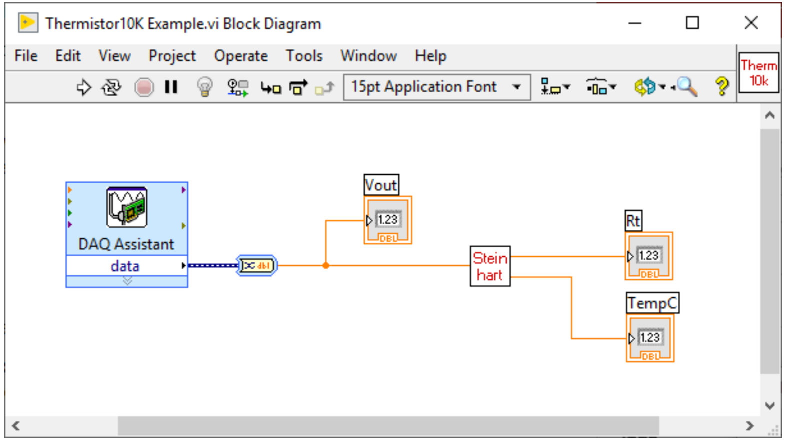
LabVIEW Example

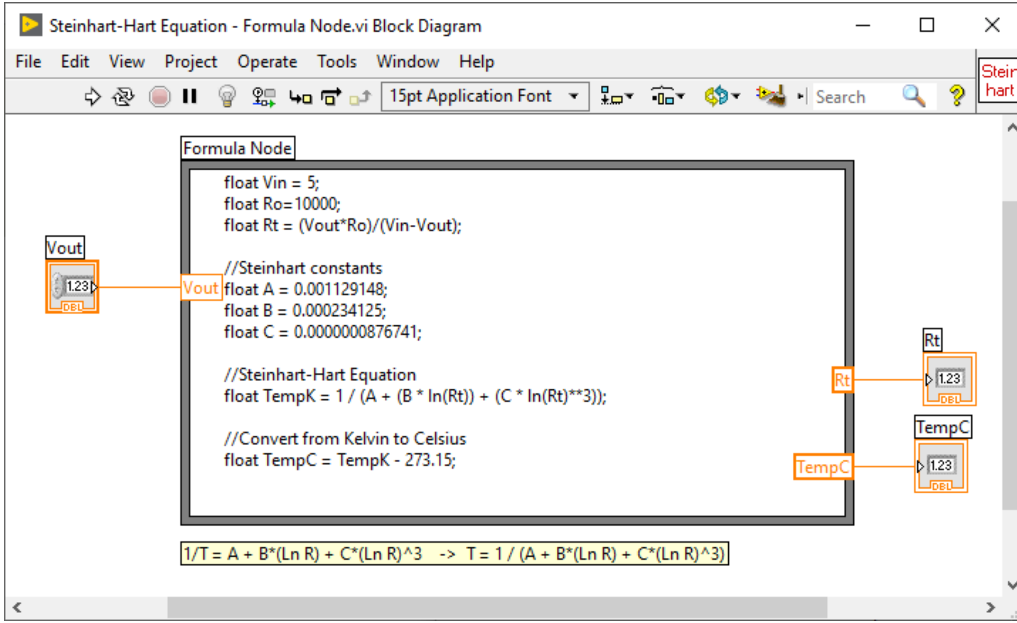

Thermistor10K Example.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Therm 10k

Vout: 2.55

Rt: 10421.4

TempC: 24.1




Steinhart-Hart Equation - Formula Node.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Stein hart
Formula Node
Vout
1.23
DBL
Vout
Rt
Rt
1.23
DBL
TempC
TempC
1.23
DBL

```
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 -> T = 1 / (A + B*(Ln R) + C*(Ln R)^3)

https://www.halvorsen.blog

Introduction to OPC

Hans-Petter Halvorsen

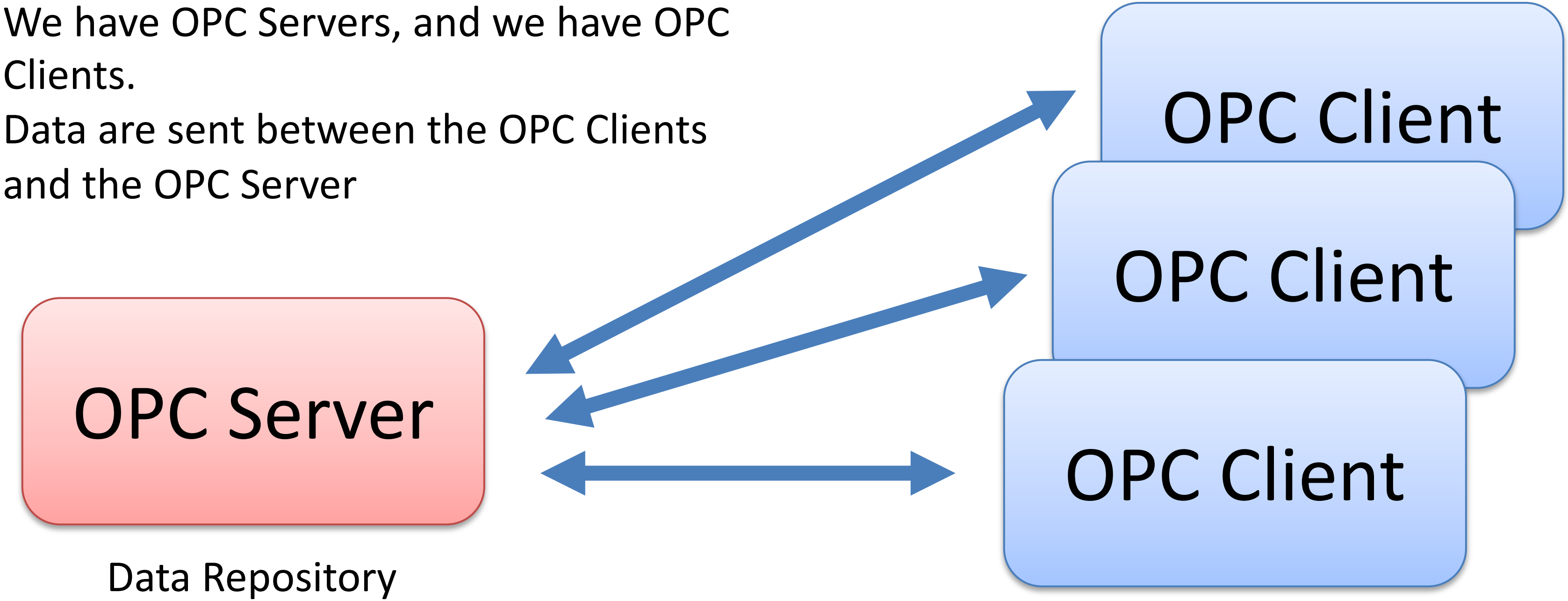
Table of Contents

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.

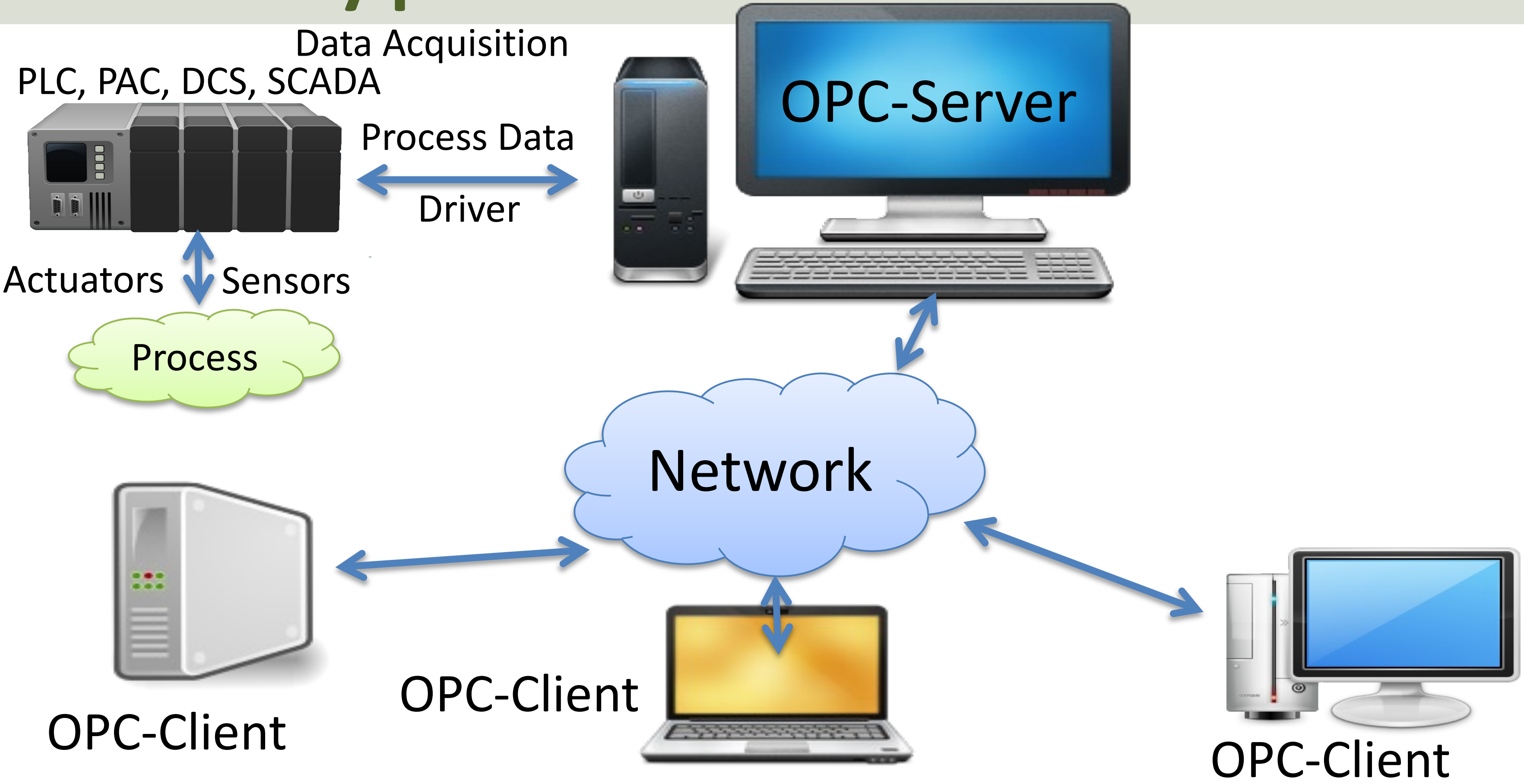
- OPC HDA (Historical Data Access)
- OPC A & E (Alarms & Events)
- ... (many others)

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



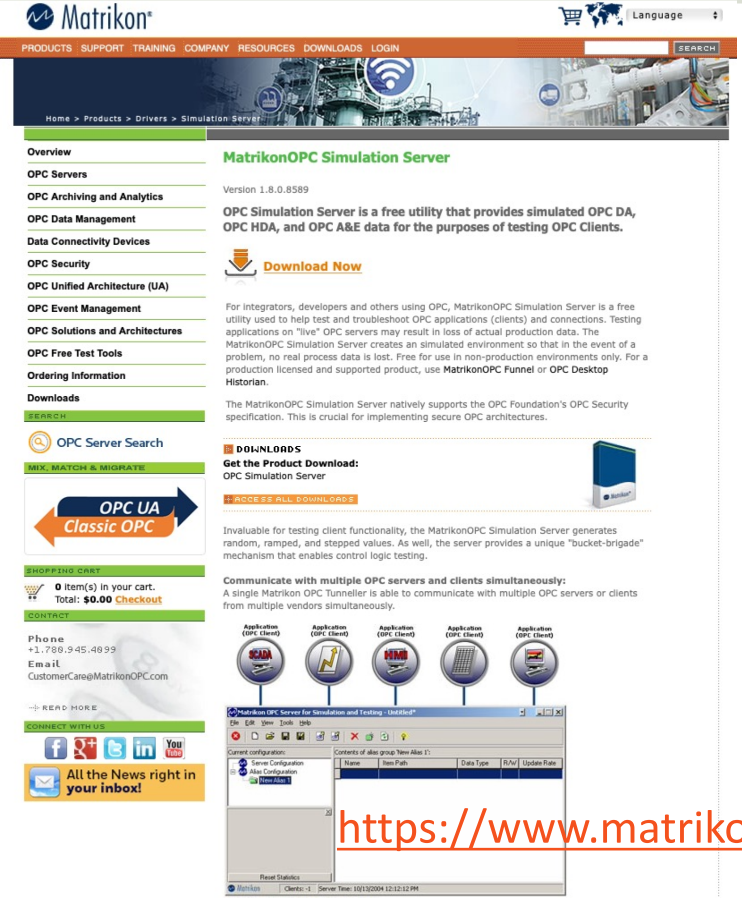
OPC DA

MatrikonOPC Simulation Server

Hans-Petter Halvorsen

Table of Contents

MatrikonOPC Simulation Server



MatrikonOPC Simulation Server

Version 1.8.0.8589

OPC 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.

Download Now

For integrators, developers and others using OPC, MatrikonOPC Simulation Server is a free utility used to help test and troubleshoot OPC applications (clients) and connections. Testing applications on "live" OPC servers may result in loss of actual production data. The MatrikonOPC Simulation Server creates an simulated environment so that in the event of a problem, no real process data is lost. Free for use in non-production environments only. For a production licensed and supported product, use MatrikonOPC Funnel or OPC Desktop Historian.

The MatrikonOPC Simulation Server natively supports the OPC Foundation's OPC Security specification. This is crucial for implementing secure OPC architectures.

DOWNLOADS

Get the Product Download:
OPC Simulation Server

ACCESS ALL DOWNLOADS

Invaluable for testing client functionality, the MatrikonOPC Simulation Server generates random, ramped, and stepped values. As well, the server provides a unique "bucket-brigade" mechanism that enables control logic testing.

Communicate with multiple OPC servers and clients simultaneously:
A single Matrikon OPC Tunneller is able to communicate with multiple OPC servers or clients from multiple vendors simultaneously.

0 item(s) in your cart.
Total: **$0.00** Checkout

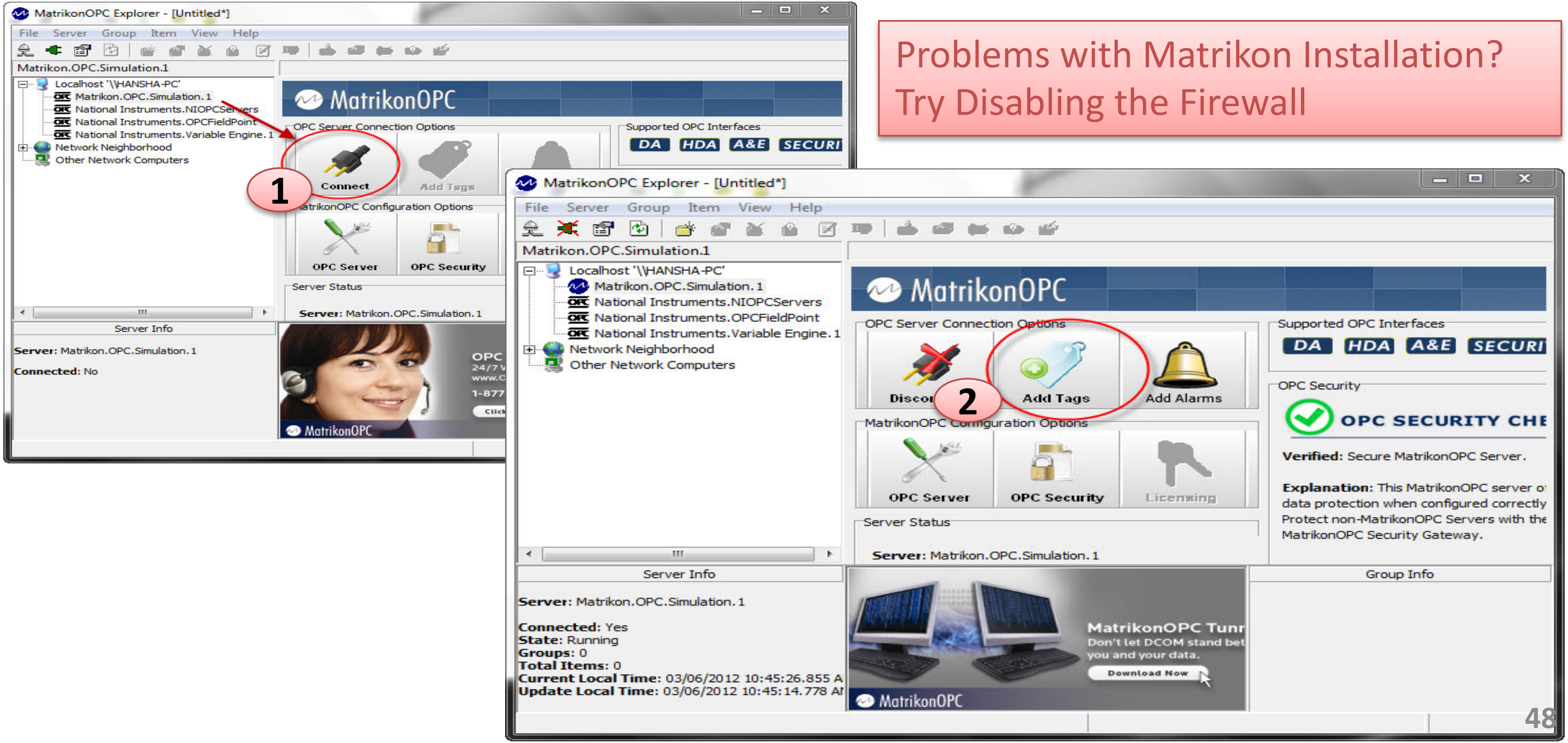
Phone
+1.780.945.4099
Email
CustomerCare@MatrikonOPC.com

Clients: -1 Server Time: 10/13/2004 12:12:12 PM

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


MatrikonOPC Explorer - [Untitled*]
File Server Group Item View Help
Matrikon.OPC.Simulation.1
Localhost '\\HANSHA-PC'
Matrikon.OPC.Simulation.1
National Instruments.NIOPCServers
National Instruments.OPCFieldPoint
National Instruments.Variable Engine.1
Network Neighborhood
Other Network Computers
MatrikonOPC
OPC Server Connection Options
Supported OPC Interfaces
DA
HDA
A&E
SECURI
1
Connect
Add Tags
MatrikonOPC Configuration Options
OPC Server
OPC Security
Server Status
Server Info
OPC
24/7
1-877
MatrikonOPC
MatrikonOPC Explorer - [Untitled*]
File Server Group Item View Help
Matrikon.OPC.Simulation.1
Localhost '\\HANSHA-PC'
Matrikon.OPC.Simulation.1
National Instruments.NIOPCServers
National Instruments.OPCFieldPoint
National Instruments.Variable Engine.1
Network Neighborhood
Other Network Computers
MatrikonOPC
OPC Server Connection Options
Supported OPC Interfaces
DA
HDA
A&E
SECURI
2
Discon
Add Tags
Add Alarms
OPC Security
OPC SECURITY CHE
MatrikonOPC Configuration Options
OPC Server
OPC Security
Licensing
Server Status
Server Info
Group Info
MatrikonOPC Tunr
Don't let DCOM stand bet you and your data.
Download Now
MatrikonOPC

Step 1:

Server: Matrikon.OPC.Simulation.1

Server: Matrikon.OPC.Simulation.1

Connected: No

Step 2:

Verified: Secure MatrikonOPC Server.

Explanation: This MatrikonOPC server o data protection when configured correctly Protect non-MatrikonOPC Servers with the MatrikonOPC Security Gateway.

Server: Matrikon.OPC.Simulation.1

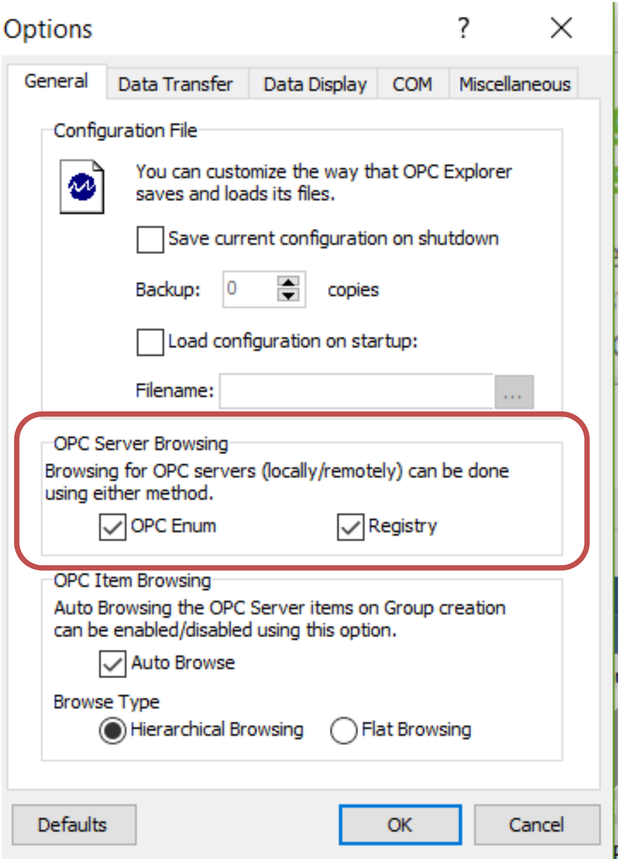
Server: Matrikon.OPC.Simulation.1

Connected: Yes
State: Running
Groups: 0
Total Items: 0
Current Local Time: 03/06/2012 10:45:26.855 A
Update Local Time: 03/06/2012 10:45:14.778 AI

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.



Configuration File

You can customize the way that OPC Explorer saves and loads its files.

☐ Save current configuration on shutdown

Backup: 0 copies

☐ Load configuration on startup:

Filename:

OPC Server Browsing

Browsing for OPC servers (locally/remotely) can be done using either method.

☑ OPC Enum ☑ Registry

OPC Item Browsing

Auto Browsing the OPC Server items on Group creation can be enabled/disabled using this option.

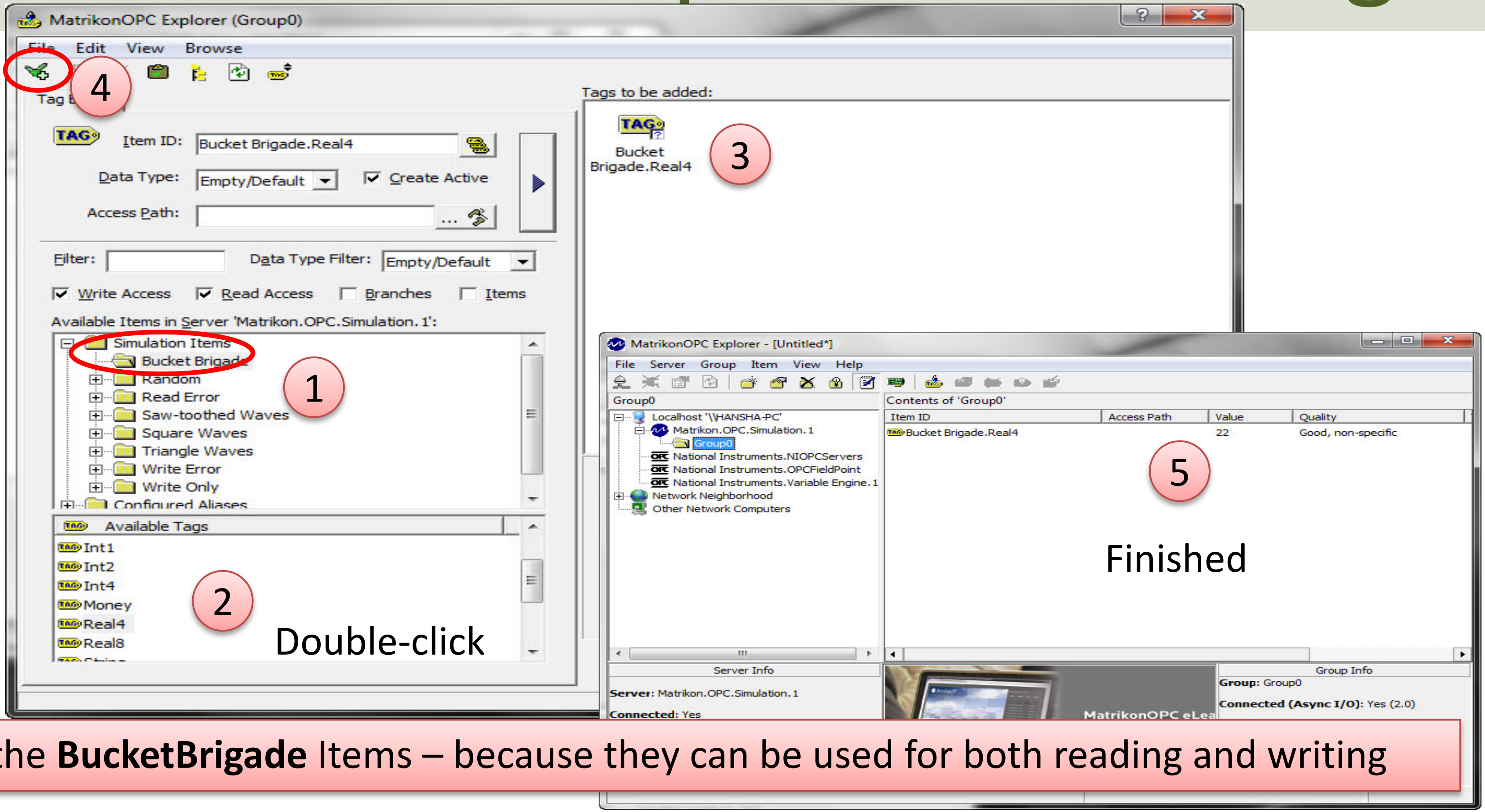
☑ Auto Browse

Browse Type

● Hierarchical Browsing ○ Flat Browsing

Matrikon OPC Explorer - Add Tag

Add Tags



MatrikonOPC Explorer (Group0)

File Edit View Browse

Tag E

Item ID: Bucket Brigade.Real4

Data Type: Empty/Default ☑ Create Active

Access Path:

Filter: Data Type Filter: Empty/Default

☑ Write Access ☑ Read Access □ Branches □ Items

Available Items in Server 'Matrikon.OPC.Simulation.1':

```
⊟ Simulation Items
    Bucket Brigade
  ⊞ Random
  ⊞ Read Error
  ⊞ Saw-toothed Waves
  ⊞ Square Waves
  ⊞ Triangle Waves
  ⊞ Write Error
  ⊞ Write Only
⊞ Configured Aliases
```

Available Tags

- Int1
- Int2
- Int4
- Money
- Real4
- Real8

Tags to be added:

Bucket Brigade.Real4

MatrikonOPC Explorer - [Untitled*]

File Server Group Item View Help

Group0

```
⊟ Localhost '\\HANSHA-PC'
  ⊟ Matrikon.OPC.Simulation.1
      Group0
    National Instruments.NIOPCServers
    National Instruments.OPCFieldPoint
    National Instruments.Variable Engine.1
⊞ Network Neighborhood
  Other Network Computers
```

Contents of 'Group0'

Item ID	Access Path	Value	Quality
Bucket Brigade.Real4		22	Good, non-specific

Server Info

Server: Matrikon.OPC.Simulation.1

Connected: Yes

MatrikonOPC eLea

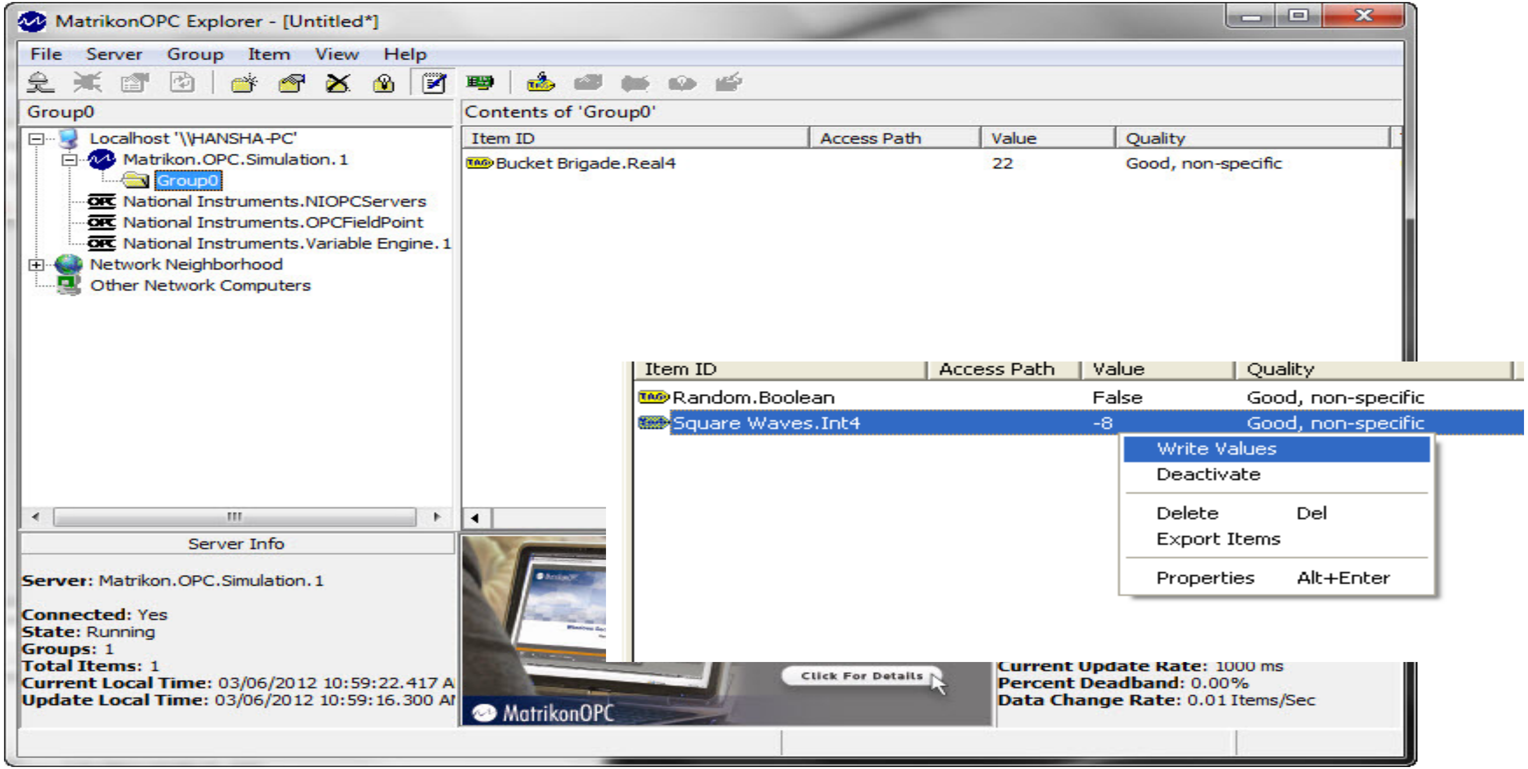
Group Info

Group: Group0

Connected (Async I/O): Yes (2.0)

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

MatrikonOPC Explorer (OPC Client)



Contents of 'Group0'

Item ID	Access Path	Value	Quality
Bucket Brigade.Real4		22	Good, non-specific

Item ID	Access Path	Value	Quality
Random.Boolean		False	Good, non-specific
Square Waves.Int4		-8	Good, non-specific

Server: Matrikon.OPC.Simulation.1

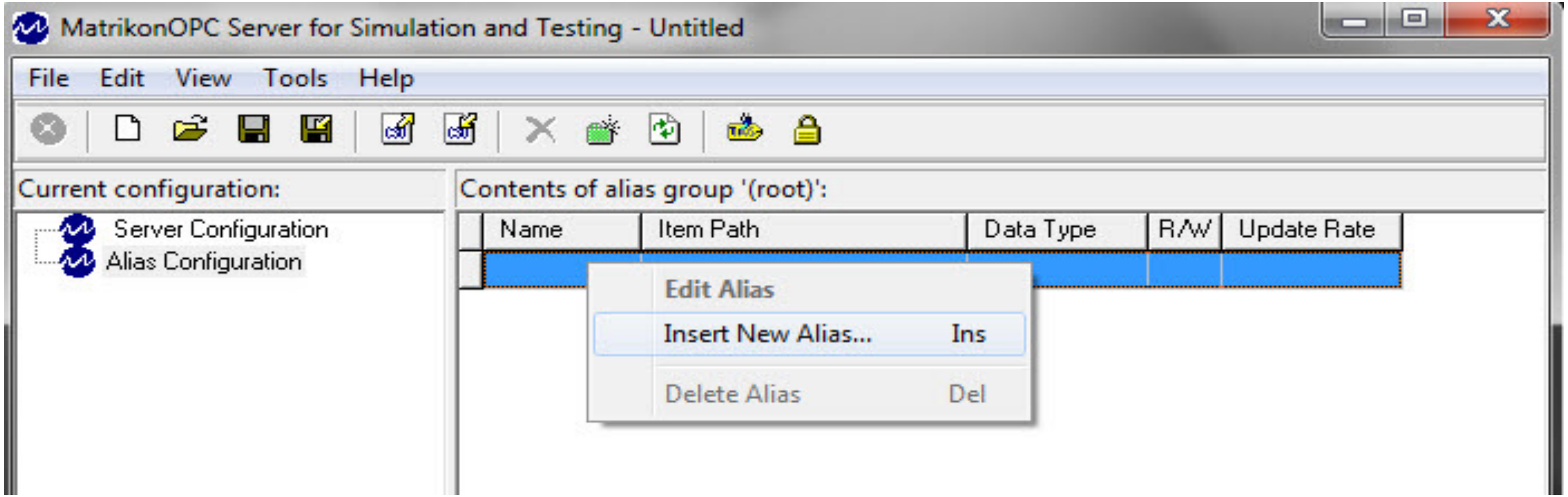
Connected: Yes
State: Running
Groups: 1
Total Items: 1
Current Local Time: 03/06/2012 10:59:22.417 A
Update Local Time: 03/06/2012 10:59:16.300 AI

Current Update Rate: 1000 ms
Percent Deadband: 0.00%
Data Change Rate: 0.01 Items/Sec

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

https://www.halvorsen.blog

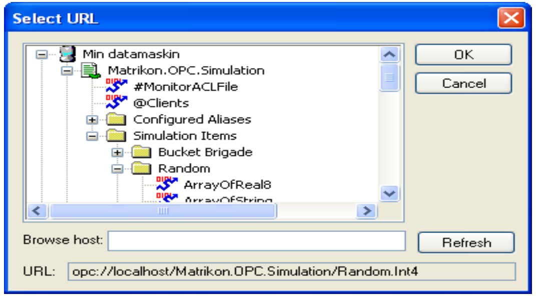
OPC DA in LabVIEW

Hans-Petter Halvorsen

Table of Contents

OPC DA in LabVIEW

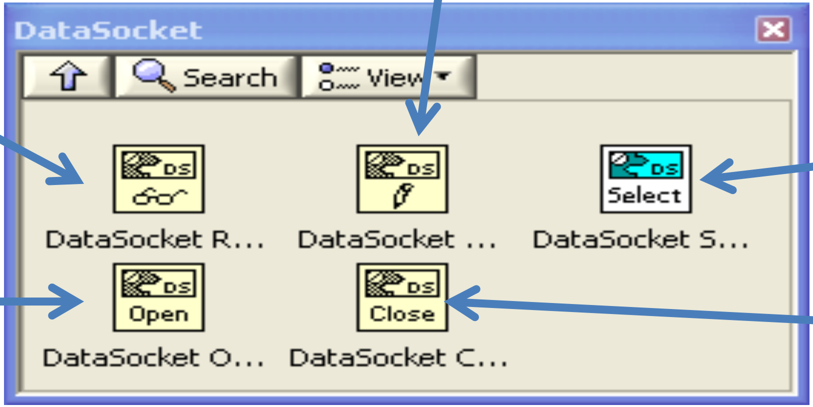
You can use LabVIEW as an OPC client by connecting to an OPC server through a **DataSocket** connection.



Browse host:

URL: opc://localhost/Matrikon.OPC.Simulation/Random.Int4

The **DataSocket** palette in LabVIEW:



Write Data to OPC

Read Data from OPC

Browse OPC Servers and OPC Items

Open Connection to OPC Server

Close Connection to OPC Server

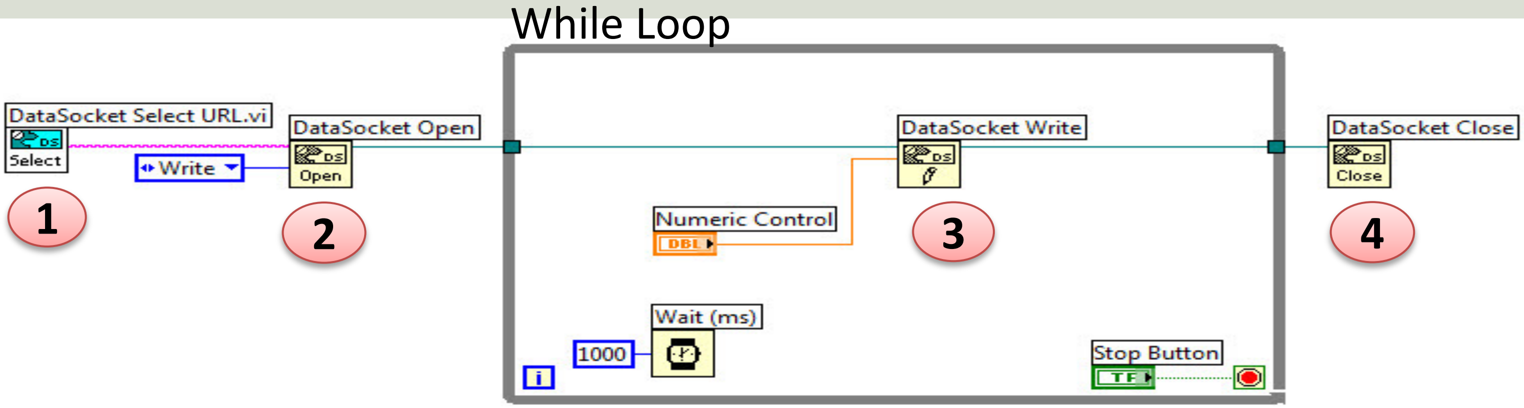
Note! Make sure to use LabVIEW 32bit version (even if you have 64bit operating system) because the DataSocket feature is only supported by the 32bit version of LabVIEW.

LabVIEW OPC DA - Write

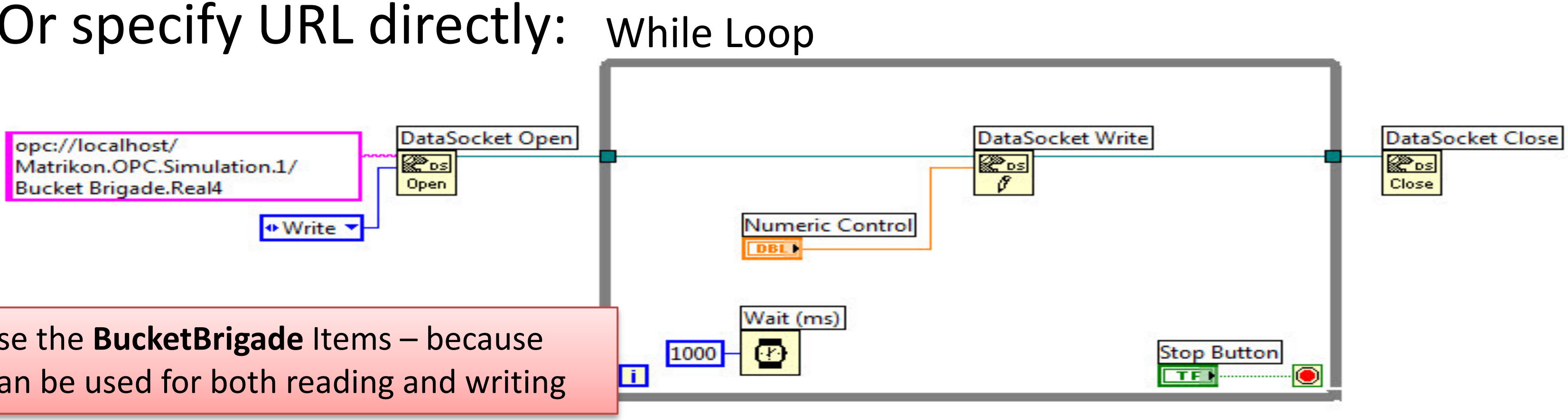
Hans-Petter Halvorsen

Table of Contents

LabVIEW OPC DA - Write



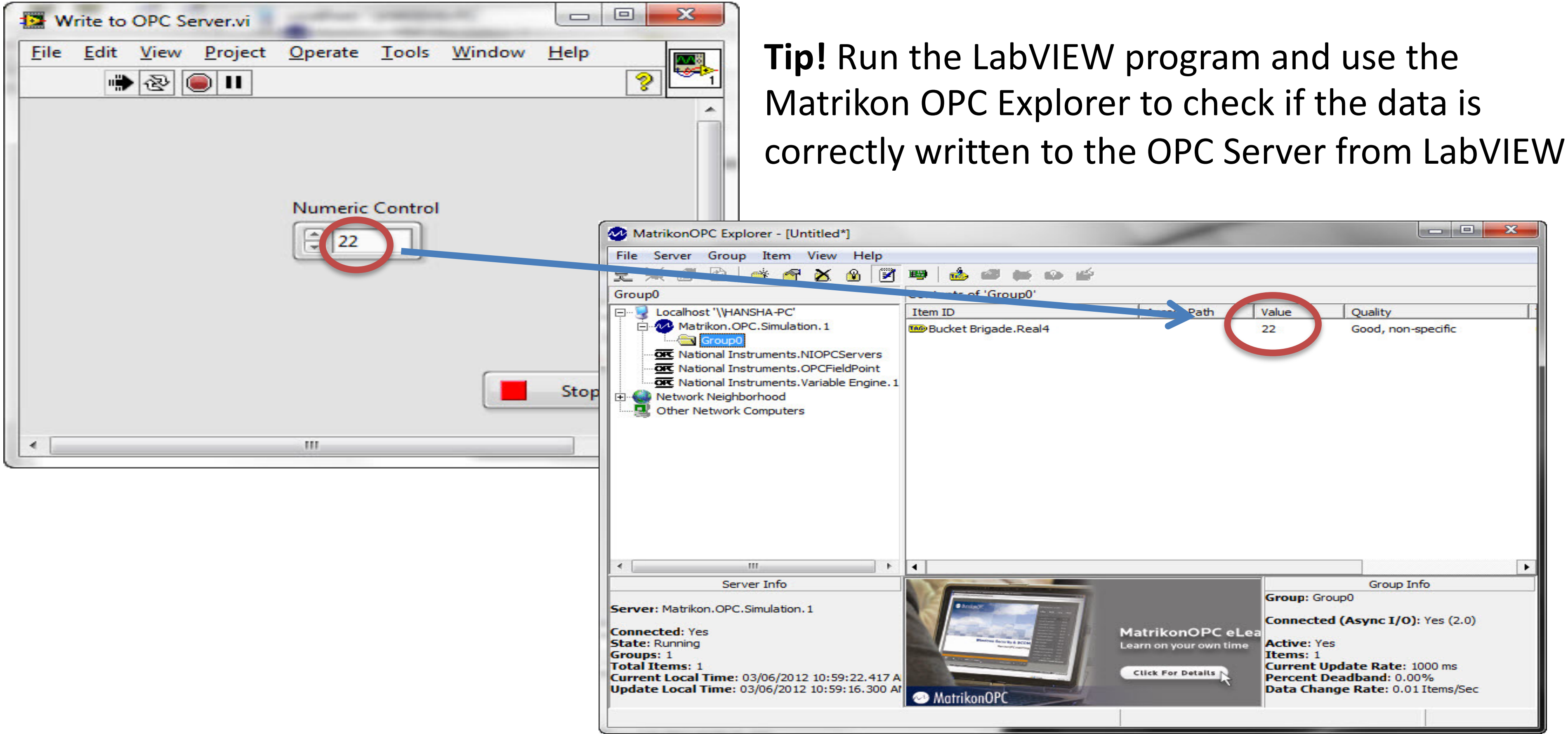
Or specify URL directly:



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


Write to OPC Server.vi
File Edit View Project Operate Tools Window Help
Numeric Control
22
Stop
MatrikonOPC Explorer - [Untitled*]
File Server Group Item View Help
Group0
Localhost '\\HANSHA-PC'
Matrikon.OPC.Simulation.1
Group0
National Instruments.NIOPCServers
National Instruments.OPCFieldPoint
National Instruments.Variable Engine.1
Network Neighborhood
Other Network Computers
Contents of 'Group0'
MatrikonOPC eLea
Learn on your own time
Click For Details
MatrikonOPC

Item ID	Access Path	Value	Quality
Bucket Brigade.Real4		22	Good, non-specific

Server Info

Server: Matrikon.OPC.Simulation.1

Connected: Yes
State: Running
Groups: 1
Total Items: 1
Current Local Time: 03/06/2012 10:59:22.417 A
Update Local Time: 03/06/2012 10:59:16.300 AI

Group Info

Group: Group0

Connected (Async I/O): Yes (2.0)

Active: Yes
Items: 1
Current Update Rate: 1000 ms
Percent Deadband: 0.00%
Data Change Rate: 0.01 Items/Sec

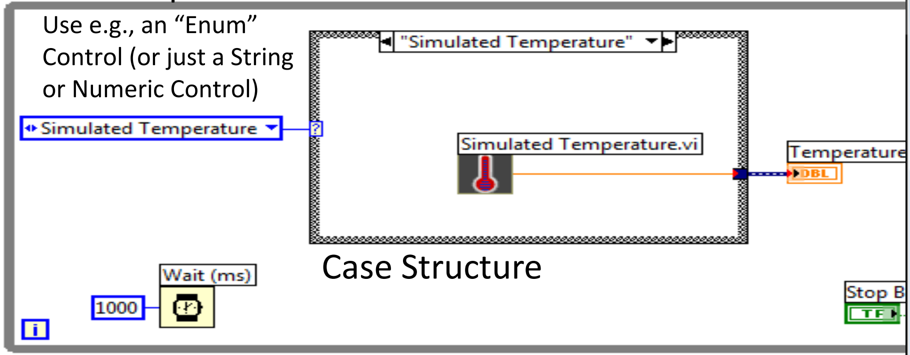
Temperature Simulator Example

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

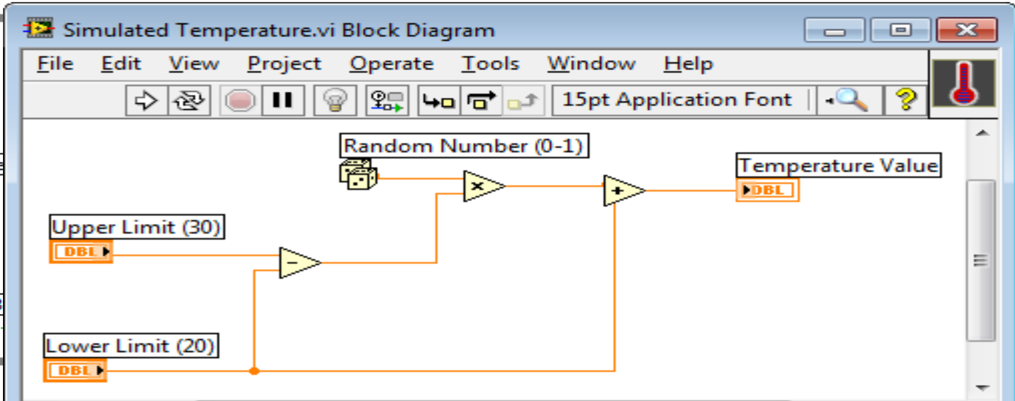
A simple SubVI that simulates a Temperature value using a Random Generator:

While Loop

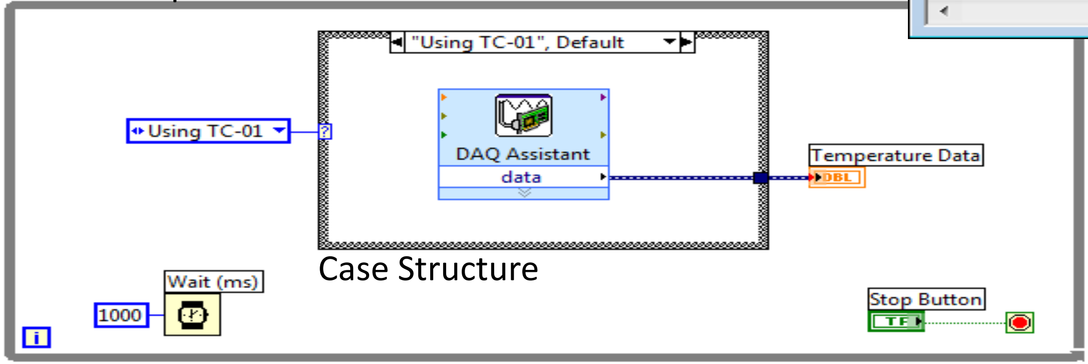
Use e.g., an "Enum" Control (or just a String or Numeric Control)



Case Structure



While Loop



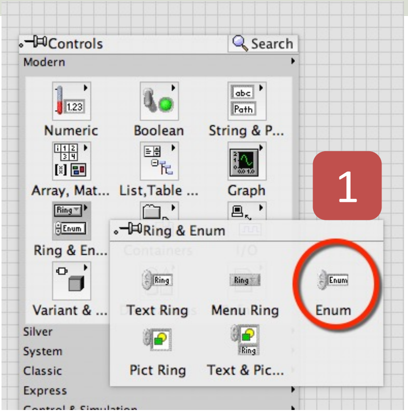
Case Structure

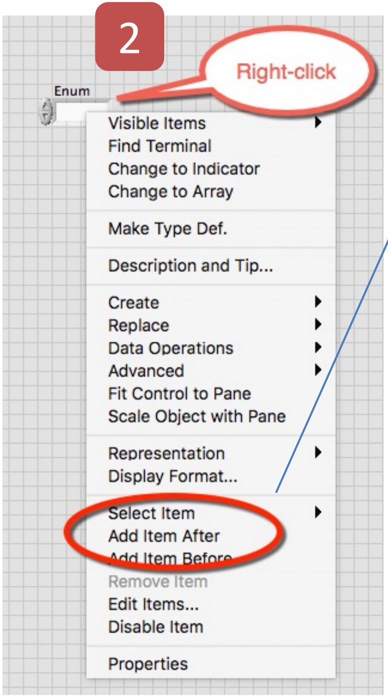
In this way you can easily switch between the real Temperature sensor (TC-01) and the Simulator.

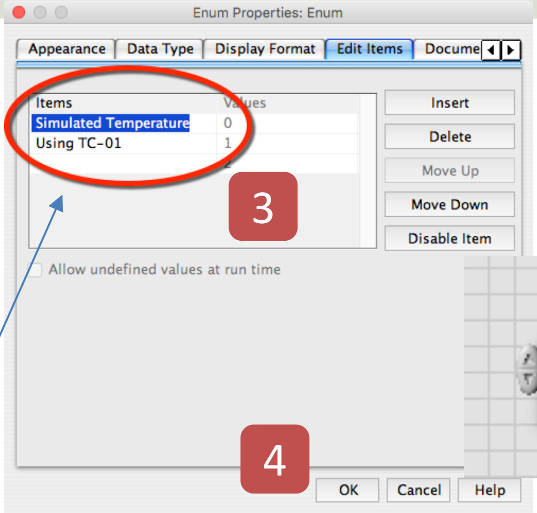
Here you just see a simple example - feel free to create a more realistic Temperature Simulator

How to create an “Enum” in LabVIEW

(used in the Temperature Simulator Example)

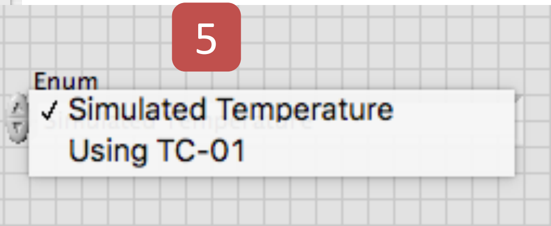


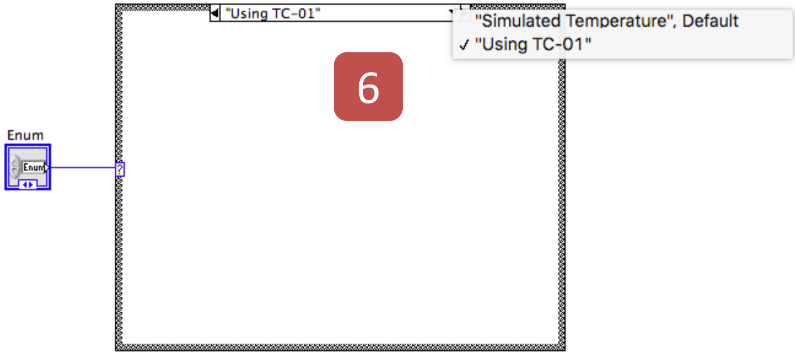




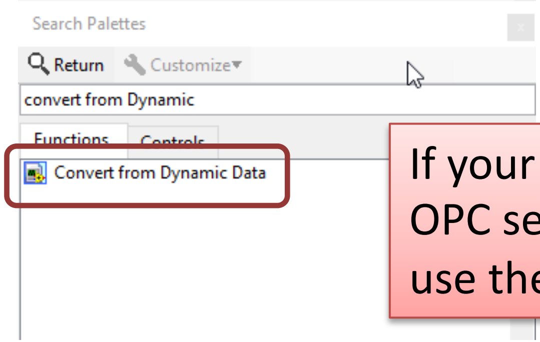
Items	Values
Simulated Temperature	0
Using TC-01	1

□ Allow undefined values at run time





Convert from Dynamic Data



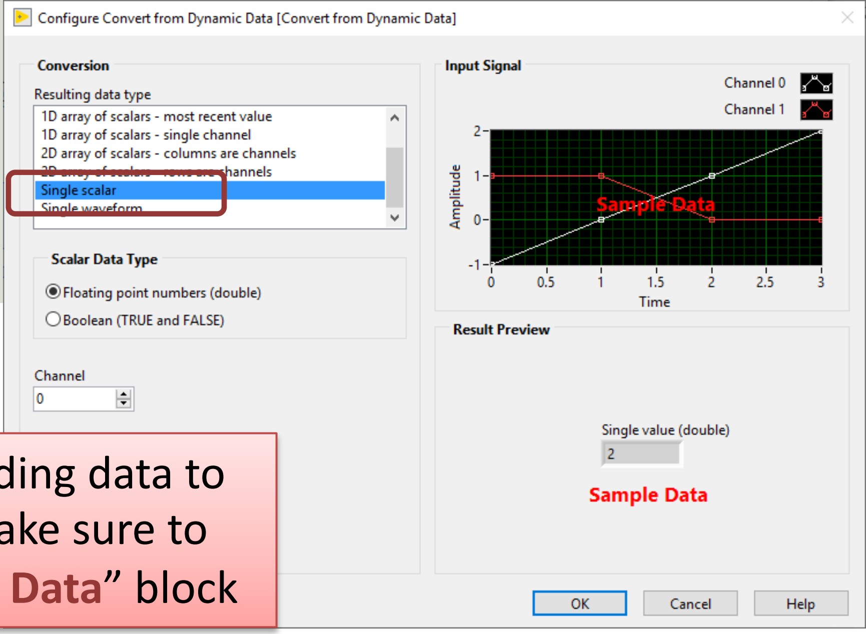
Search Palettes

Return   Customize▾

convert from Dynamic

Functions   Controls

- Convert from Dynamic Data



Configure Convert from Dynamic Data [Convert from Dynamic Data]

Conversion

Resulting data type

- 1D array of scalars - most recent value
- 1D array of scalars - single channel
- 2D array of scalars - columns are channels
- 2D array of scalars - rows are channels
- Single scalar
- Single waveform

Scalar Data Type

● Floating point numbers (double)

○ Boolean (TRUE and FALSE)

Channel: 0

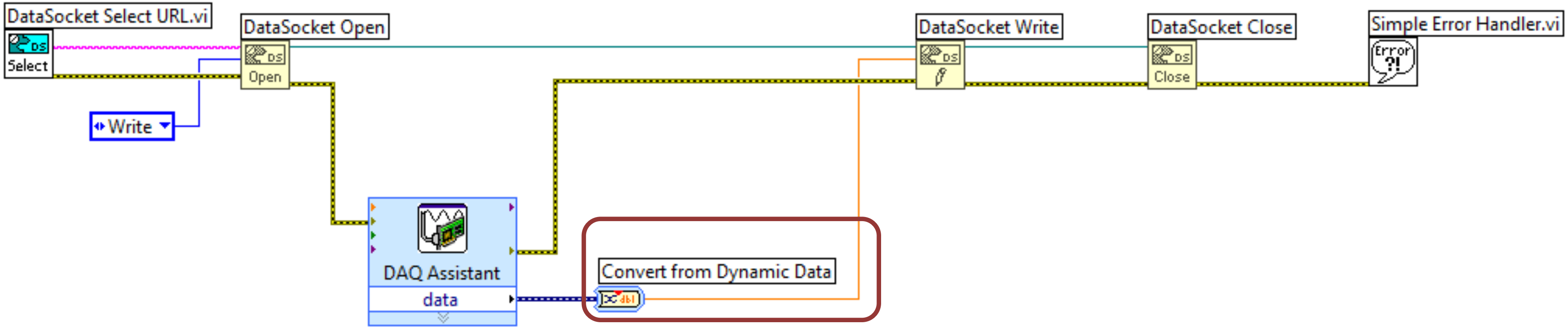
Input Signal

Result Preview

Single value (double): 2

OK   Cancel   Help

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

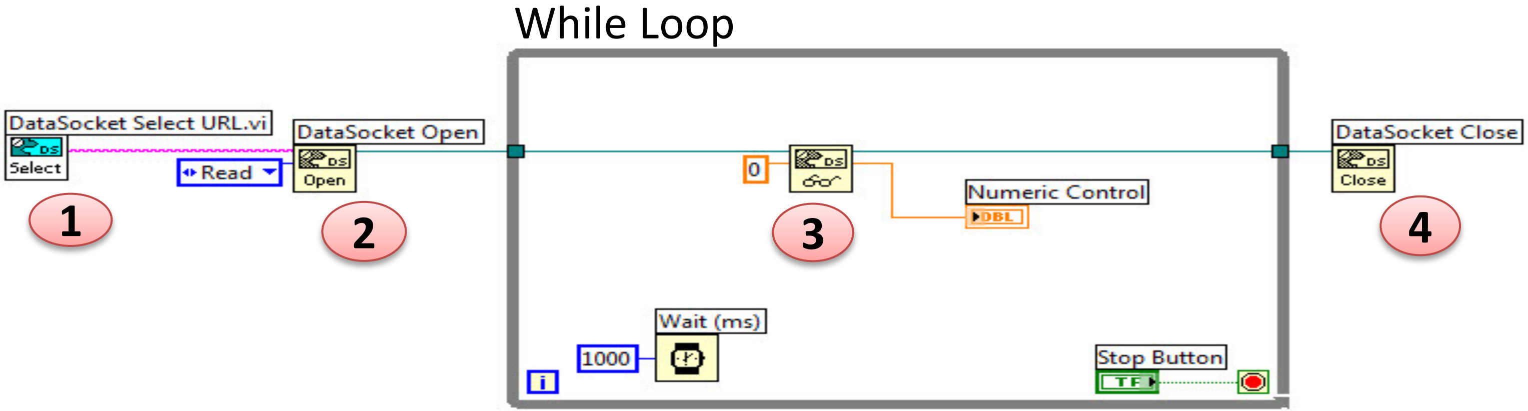


LabVIEW OPC DA - Read

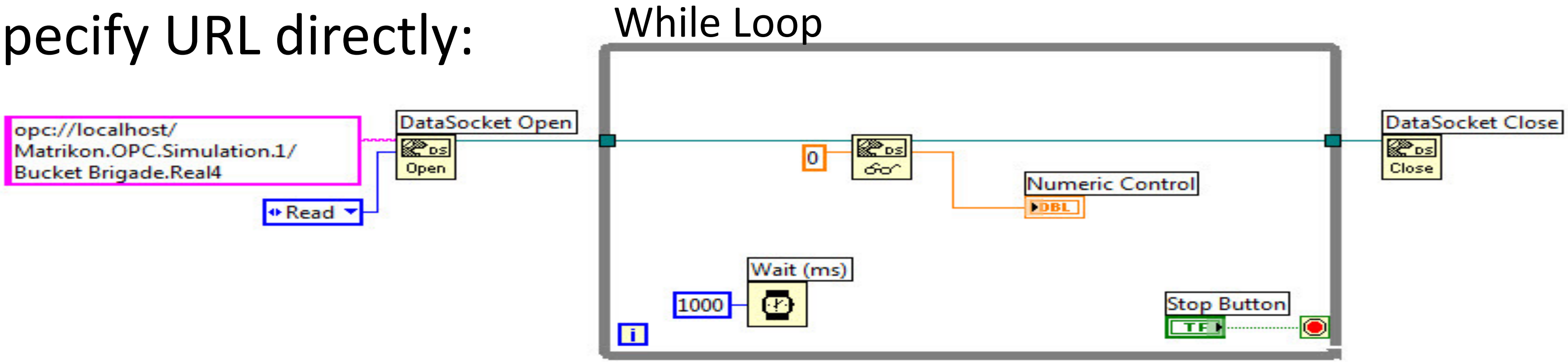
Hans-Petter Halvorsen

Table of Contents

Read from OPC Server using LabVIEW



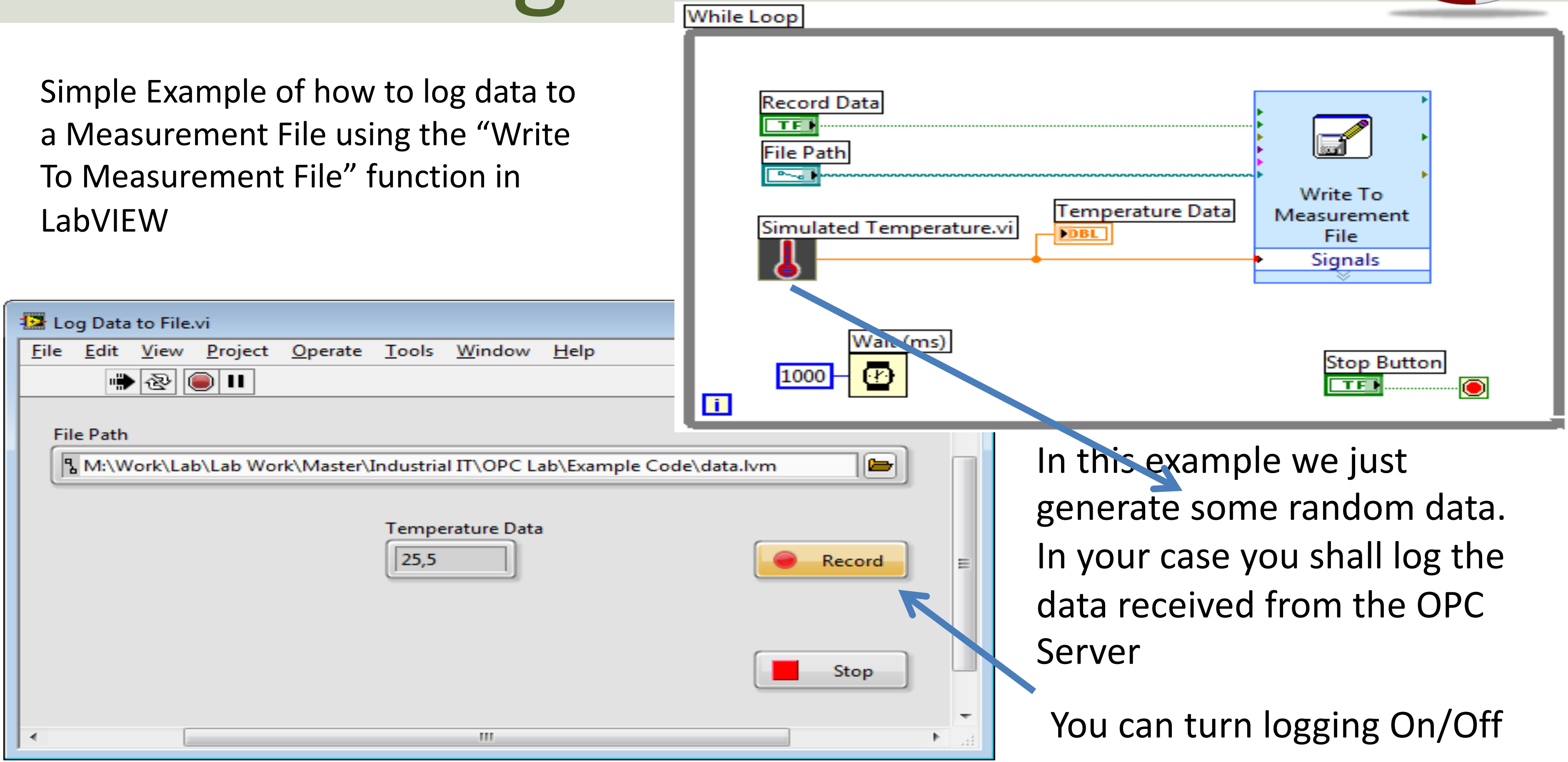
Or specify URL directly:



Log Data to File

Example

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



File Path: M:\Work\Lab\Lab Work\Master\Industrial IT\OPC Lab\Example Code\data.lvm

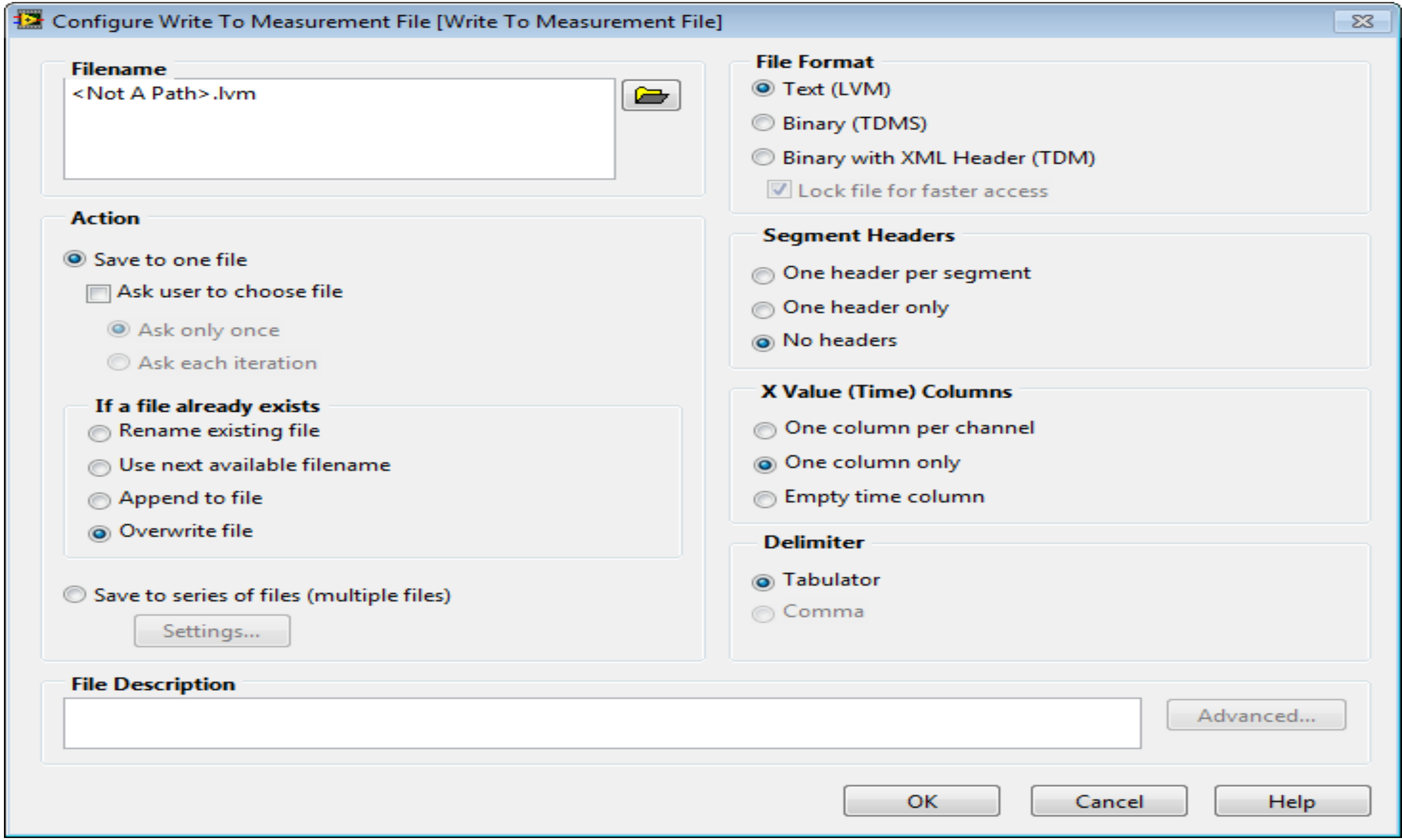
Temperature Data: 25,5

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):



Configure Write To Measurement File [Write To Measurement File]

Filename

<Not A Path>.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)

File Format

● Text (LVM)

○ Binary (TDMS)

○ Binary with XML Header (TDM)

☑ 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

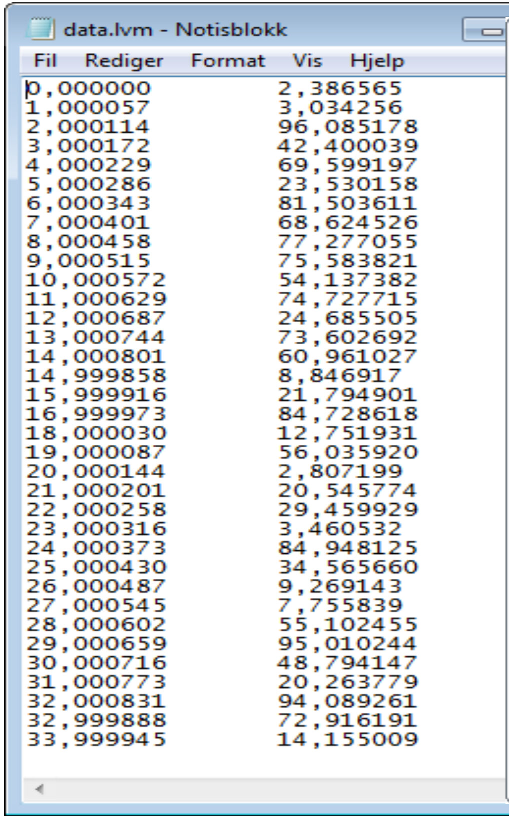
● Tabulator

○ Comma

File Description

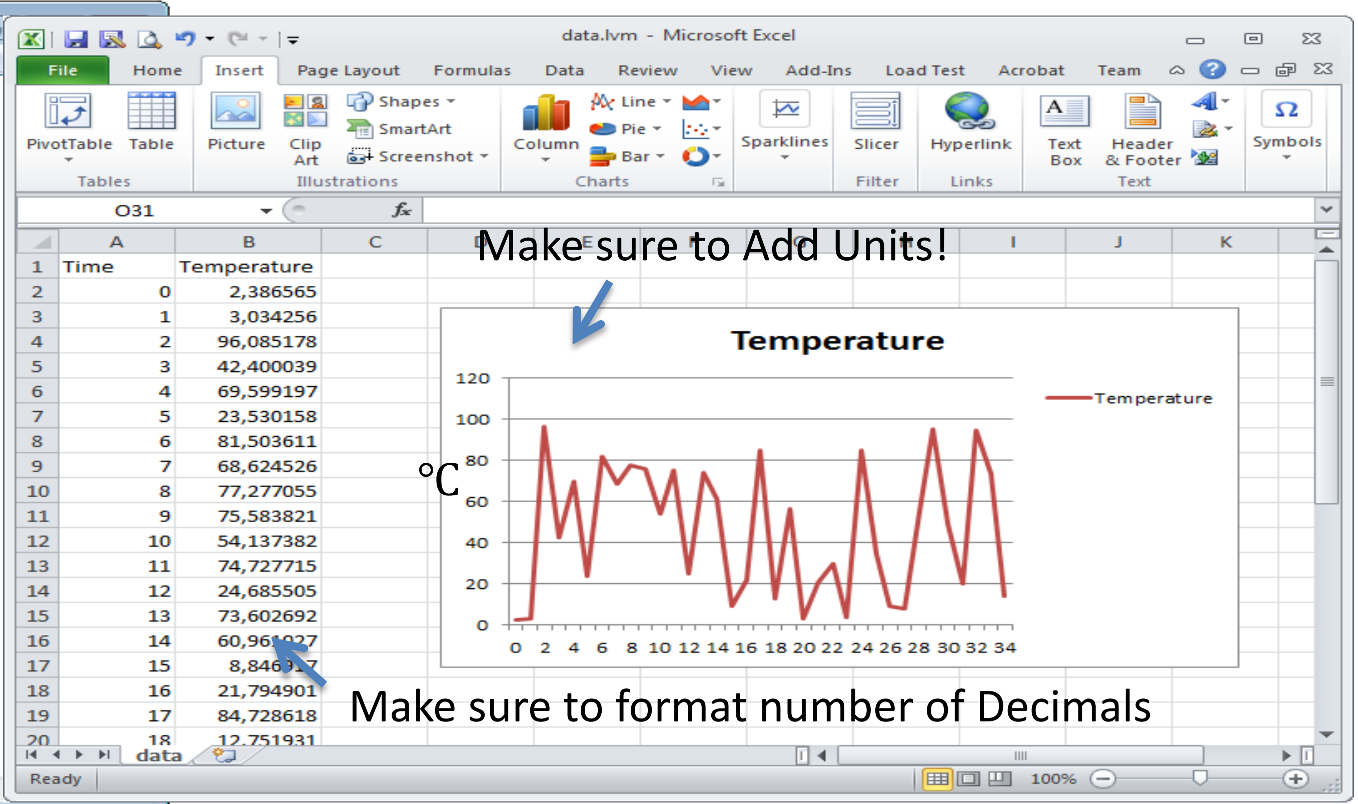
Measurement File – Data Visualization

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



```
0,000000        2,386565
1,000057        3,034256
2,000114        96,085178
3,000172        42,400039
4,000229        69,599197
5,000286        23,530158
6,000343        81,503611
7,000401        68,624526
8,000458        77,277055
9,000515        75,583821
10,000572       54,137382
11,000629       74,727715
12,000687       24,685505
13,000744       73,602692
14,000801       60,961027
14,999858       8,846917
15,999916       21,794901
16,999973       84,728618
18,000030       12,751931
19,000087       56,035920
20,000144       2,807199
21,000201       20,545774
22,000258       29,459929
23,000316       3,460532
24,000373       84,948125
25,000430       34,565660
26,000487       9,269143
27,000545       7,755839
28,000602       55,102455
29,000659       95,010244
30,000716       48,794147
31,000773       20,263779
32,000831       94,089261
32,999888       72,916191
33,999945       14,155009
```

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



	A	B
1	Time	Temperature
2	0	2,386565
3	1	3,034256
4	2	96,085178
5	3	42,400039
6	4	69,599197
7	5	23,530158
8	6	81,503611
9	7	68,624526
10	8	77,277055
11	9	75,583821
12	10	54,137382
13	11	74,727715
14	12	24,685505
15	13	73,602692
16	14	60,961027
17	15	8,846917
18	16	21,794901
19	17	84,728618
20	18	12,751931

Make sure to Add Units!

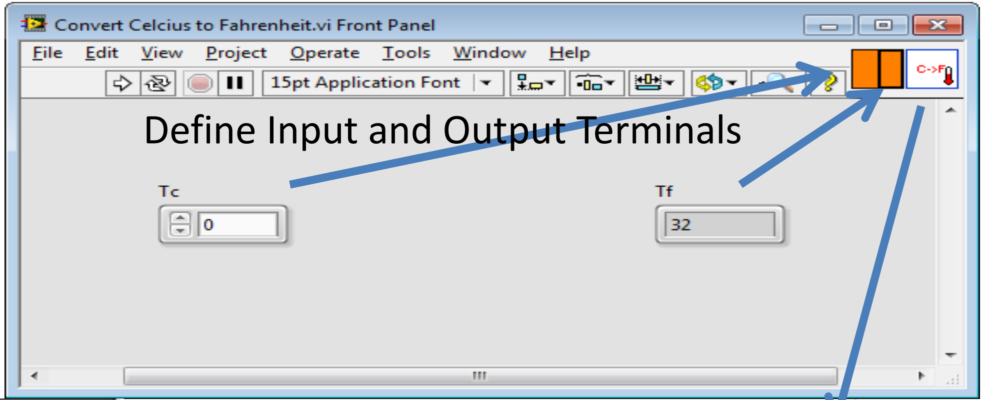
°C

Make sure to format number of Decimals

SubVI – Scaling from Celsius to Fahrenheit

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

Front Panel:

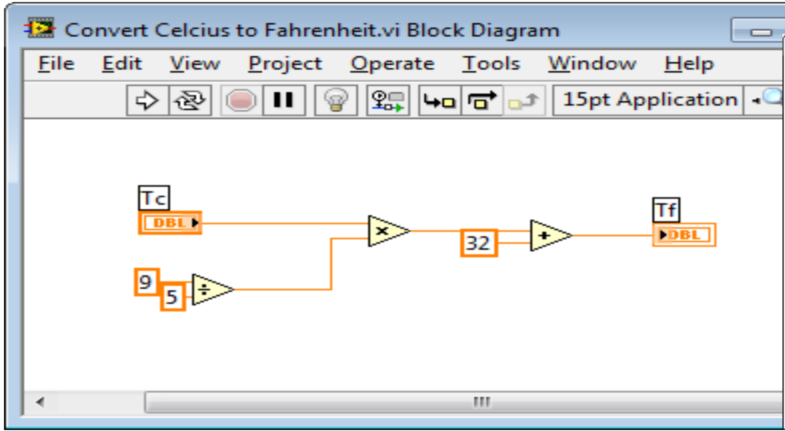


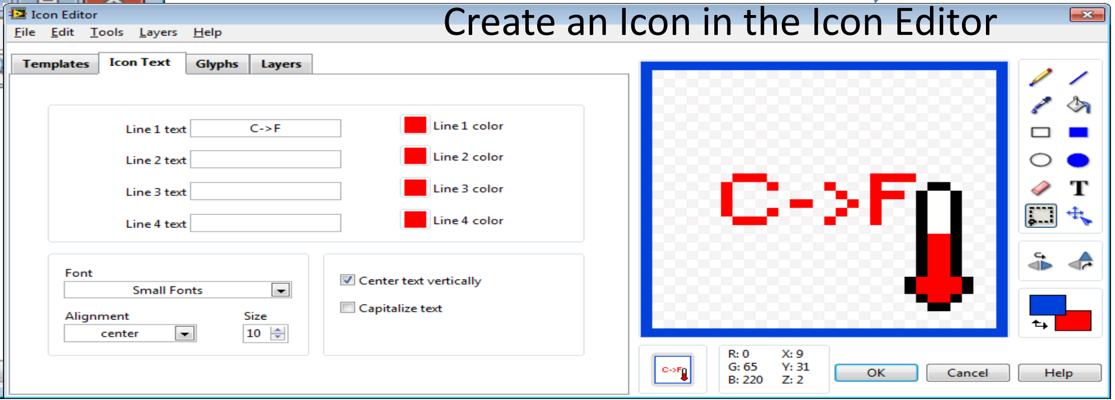
Define Input and Output Terminals

Tc: 0

Tf: 32

Block Diagram:





Create an Icon in the Icon Editor

Line 1 text: C->F — 🟥 Line 1 color

Line 2 text: — 🟥 Line 2 color

Line 3 text: — 🟥 Line 3 color

Line 4 text: — 🟥 Line 4 color

Font: Small Fonts

Alignment: center

Size: 10

☑ Center text vertically

□ Capitalize text

R: 0 G: 65 B: 220 X: 9 Y: 31 Z: 2

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.

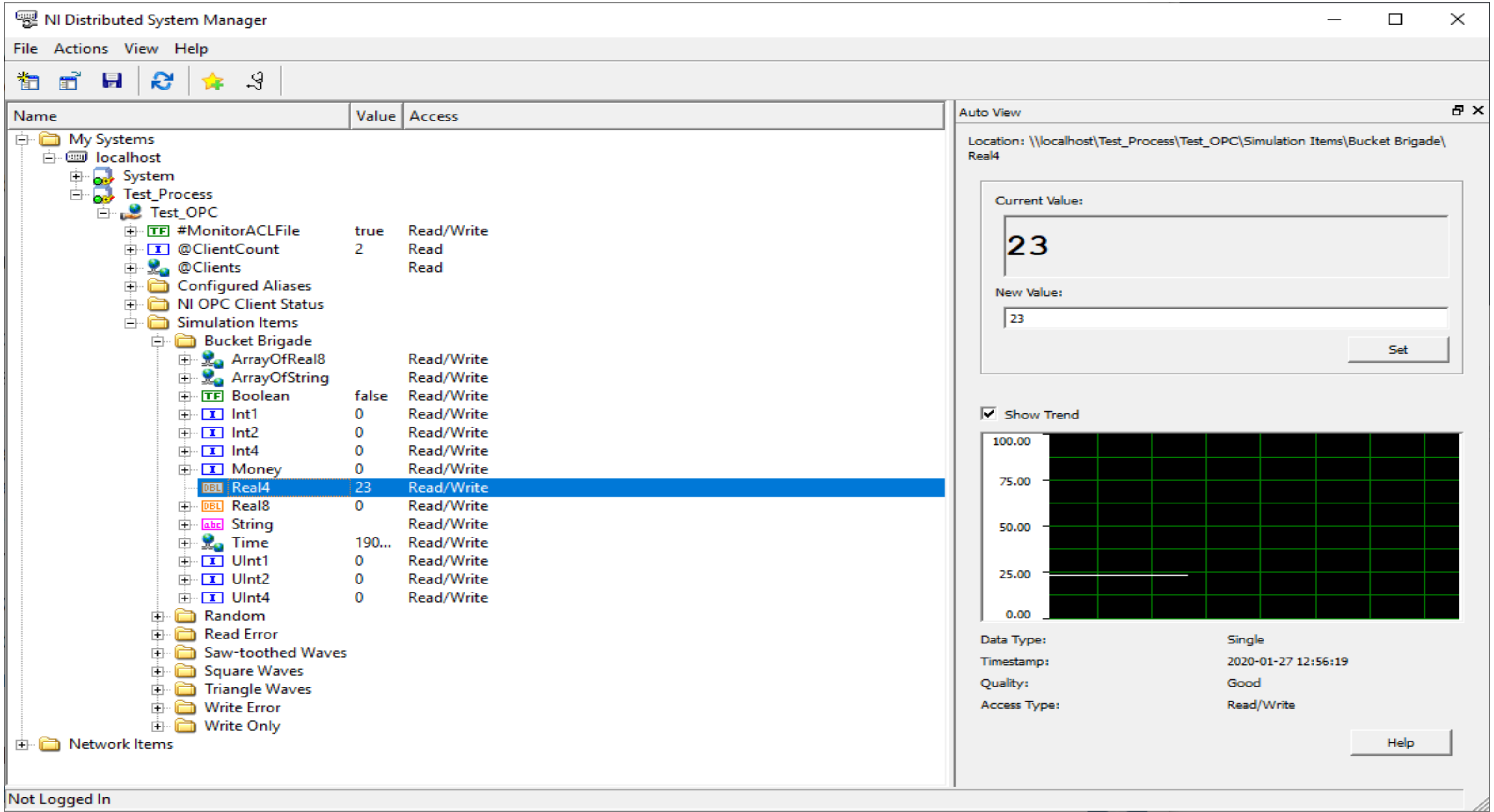
11. In the Browse for Variable dialog box, select one of the OPC items from the OPC I/O server you configured in step 6.

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



Name	Value	Access
My Systems		
localhost		
System		
Test_Process		
Test_OPC		
#MonitorACLFile	true	Read/Write
@ClientCount	2	Read
@Clients		Read
Configured Aliases		
NI OPC Client Status		
Simulation Items		
Bucket Brigade		
ArrayOfReal8		Read/Write
ArrayOfString		Read/Write
Boolean	false	Read/Write
Int1	0	Read/Write
Int2	0	Read/Write
Int4	0	Read/Write
Money	0	Read/Write
Real4	23	Read/Write
Real8	0	Read/Write
String		Read/Write
Time	190...	Read/Write
UInt1	0	Read/Write
UInt2	0	Read/Write
UInt4	0	Read/Write
Random		
Read Error		
Saw-toothed Waves		
Square Waves		
Triangle Waves		
Write Error		
Write Only		
Network Items		

Auto View

Location: \\localhost\Test_Process\Test_OPC\Simulation Items\Bucket Brigade\Real4

Current Value: 23

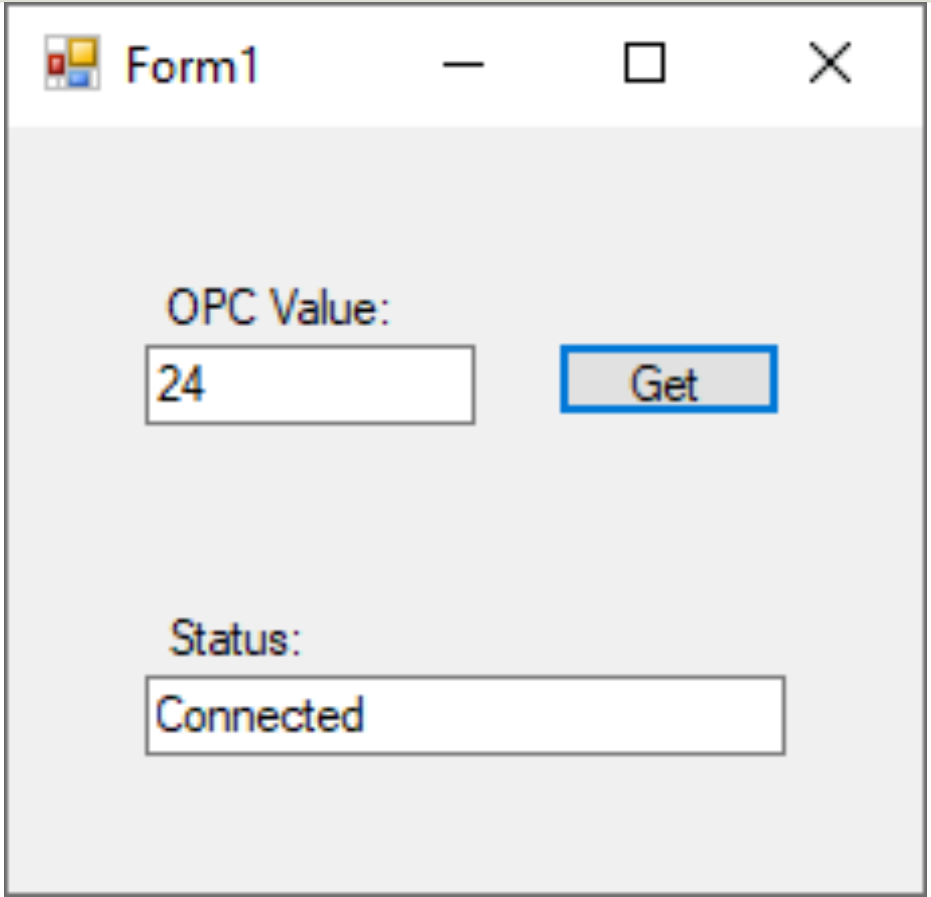
New Value: 23

☑ Show Trend

Data Type:	Single
Timestamp:	2020-01-27 12:56:19
Quality:	Good
Access Type:	Read/Write

Not Logged In

OPC DA with Visual Studio/C#



OPC Value:

24

Status:

Connected

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;
            }
        }
…
```

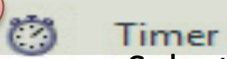
```
….

        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

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

1



Select the “Timer” component in the Toolbox

2

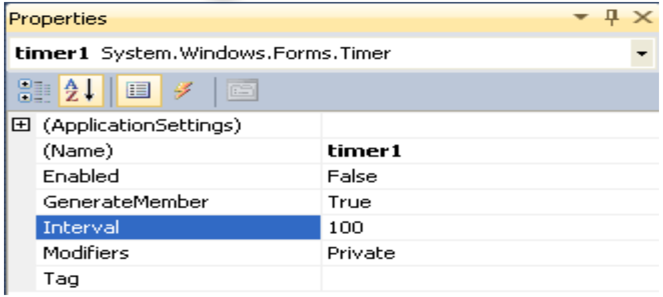
Initialization:

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

        timer1.Start();
    }
```

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

3

Properties:


Properties

timer1 System.Windows.Forms.Timer

⊞ (ApplicationSettings)	
(Name)	**timer1**
Enabled	False
GenerateMember	True
Interval	100
Modifiers	Private
Tag	

You may specify the Timer Interval in the Properties Window

4

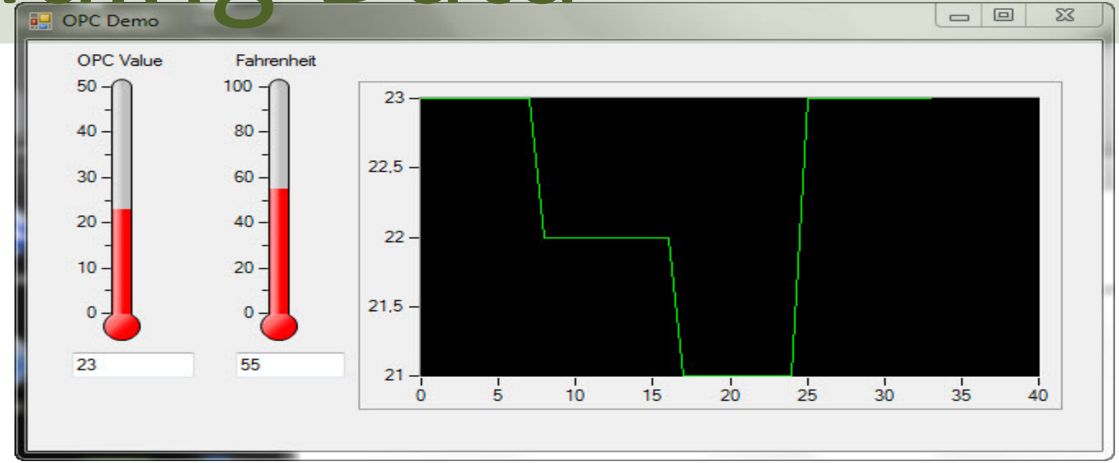
Timer Event:

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

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);

}
```

Example

Name of your WaveformGraph control

Name of the Method to use

Name of the variable with Temperature data

OPC DA in MATLAB

Hans-Petter Halvorsen

Table of Contents

MATLAB OPC DA Read Example 1

```
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)
```

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)
```

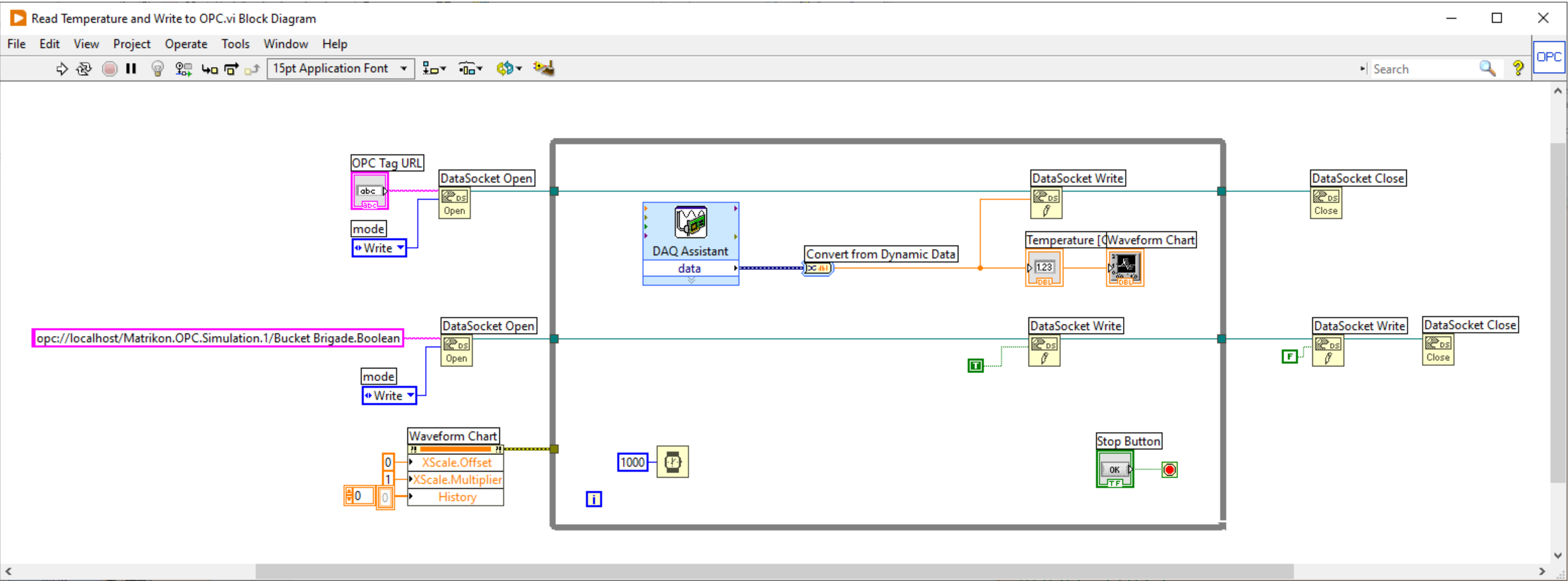
MATLAB OPC DA Read Example 2b

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



Read Temperature and Write to OPC.vi Block Diagram

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!

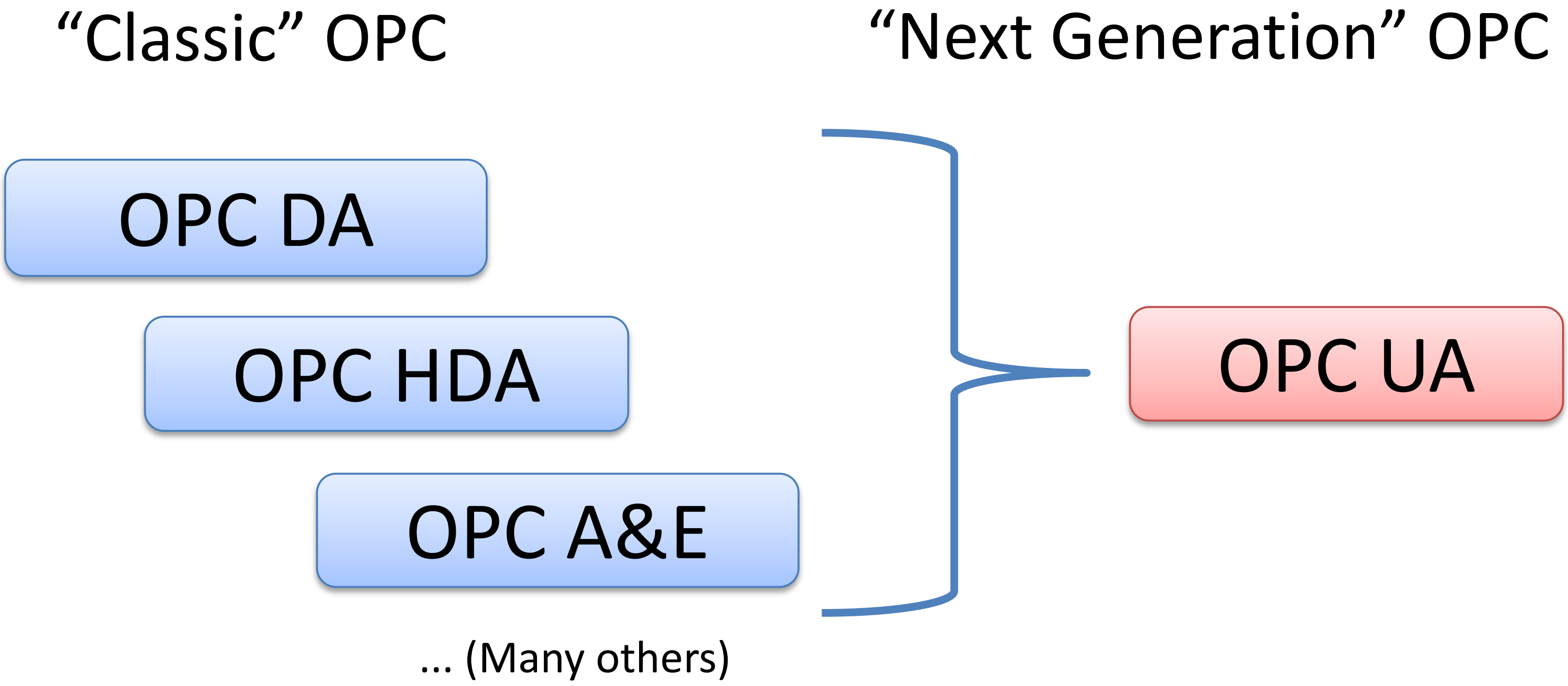
```
clear, clc
% Connect to OPC Server
da = opcda('localhost', 'Matrikon.OPC.Simulation.1');
connect(da);
% Create Group
grp = addgroup(da, 'DemoGroup');
%Add Tags
itmIDs = {'Random.Real8'};
itm = additem(grp, itmIDs)
% Set Properties
logDuration = 60;logRate = 0.2;
numRecords = ceil(logDuration./logRate)
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

Table of Contents

“Next Generation” OPC



“Classic” OPC

“Next Generation” OPC

OPC DA

OPC HDA

OPC A&E

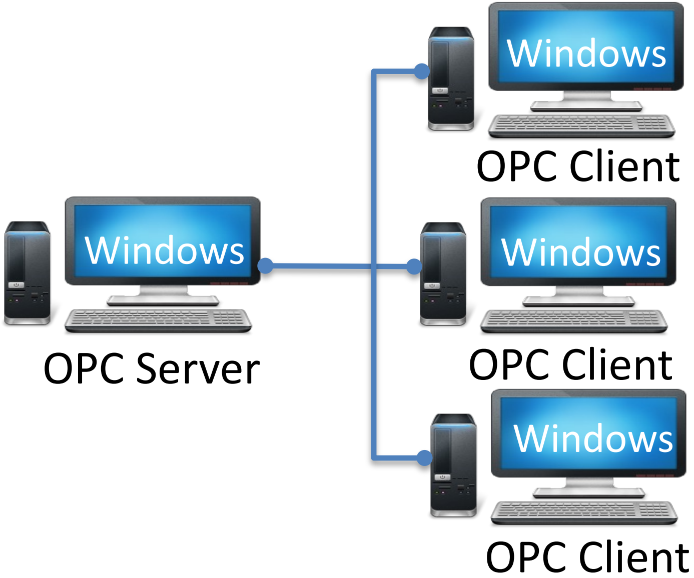
… (Many others)

OPC UA

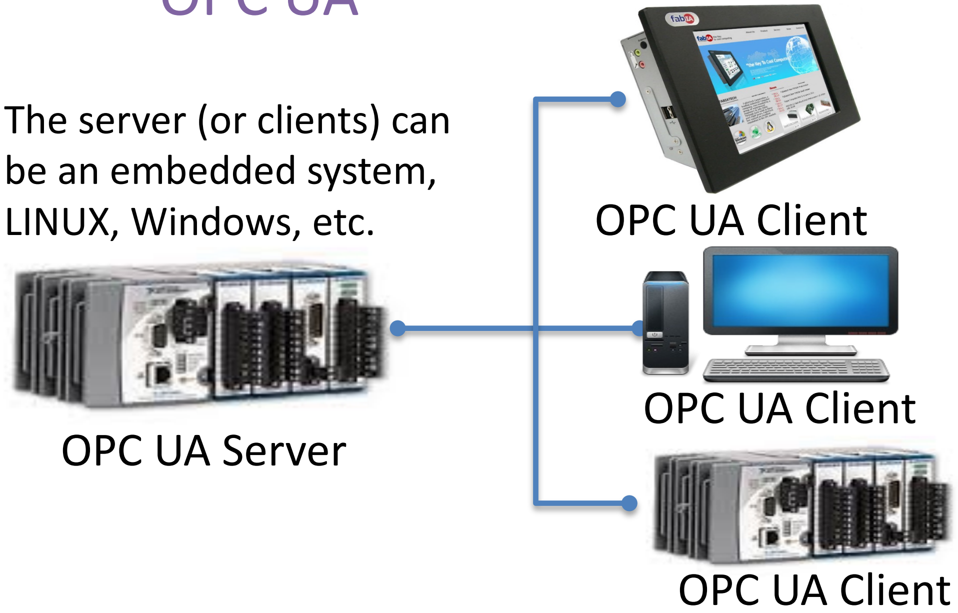
Classic OPC vs. OPC UA

Theory

Classic OPC (DCOM)



OPC UA

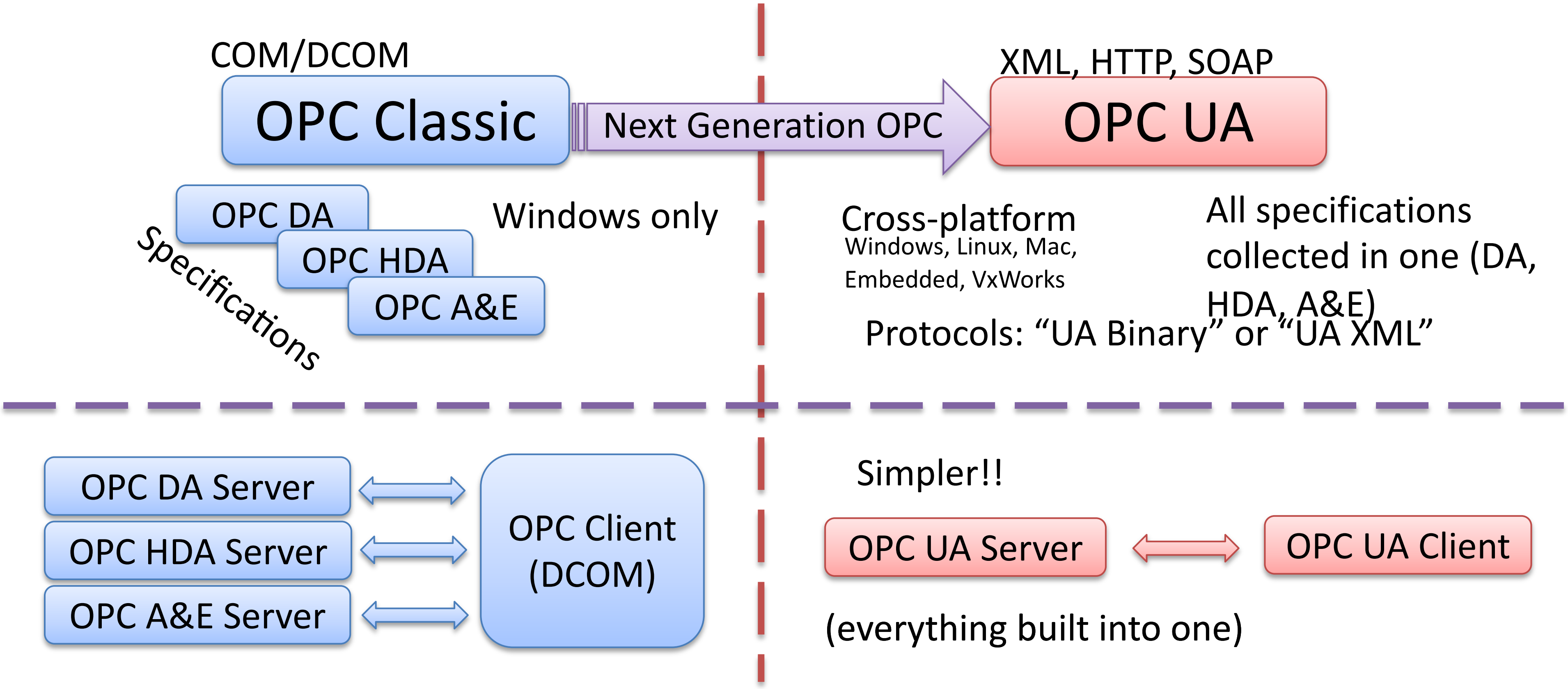

The server (or clients) can be an embedded system, LINUX, Windows, etc.

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

Theory



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!!

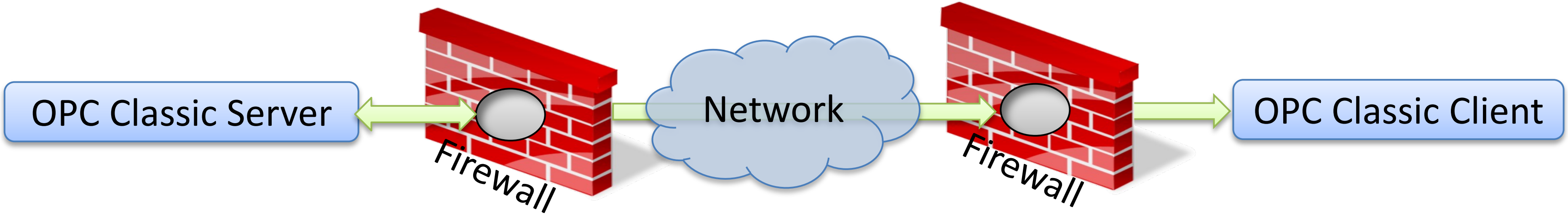
OPC UA Server

OPC UA Client

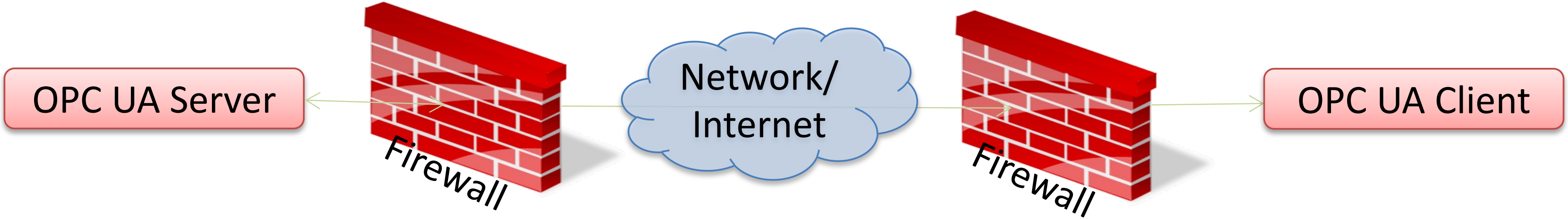
(everything built into one)

Firewall

Theory



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)

Theory

- 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

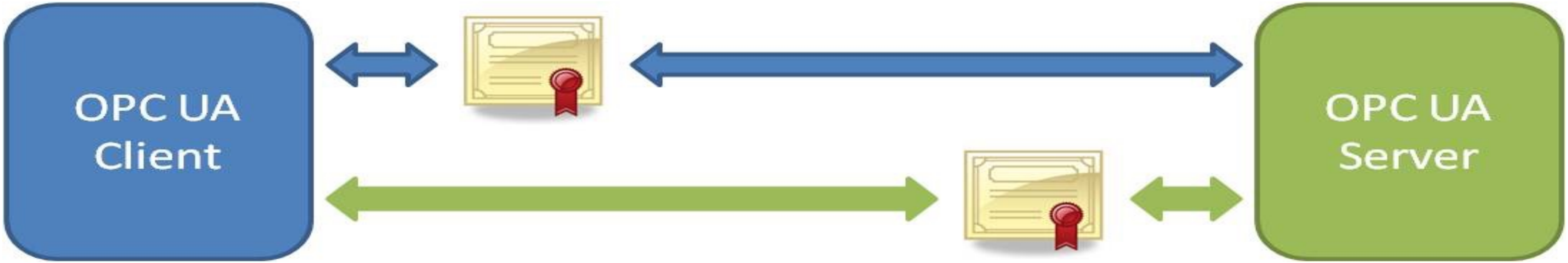
Theory

- 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

Theory

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.

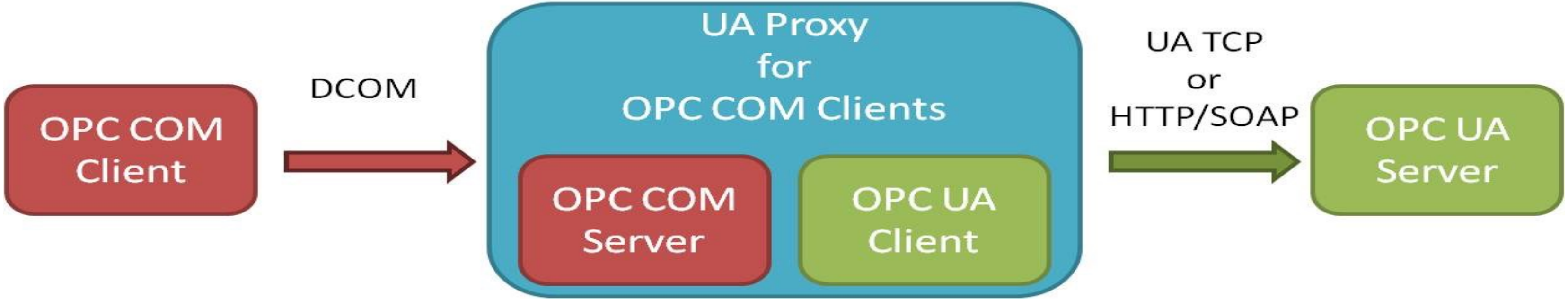
http://www.ni.com/white-paper/13843/en/

Classic OPC and OPC UA Compatibility

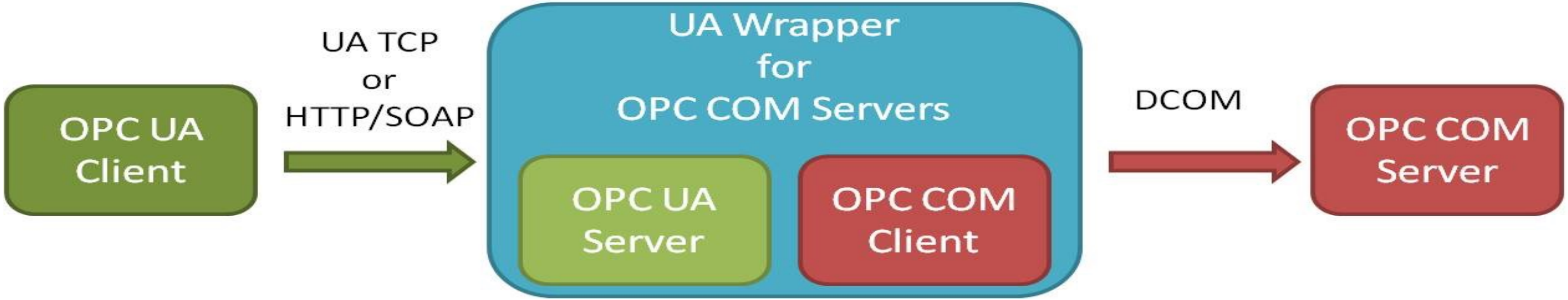
Theory

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/

OPC UA Server Simulator

Free OPC UA Simulation Server from Integration Objects

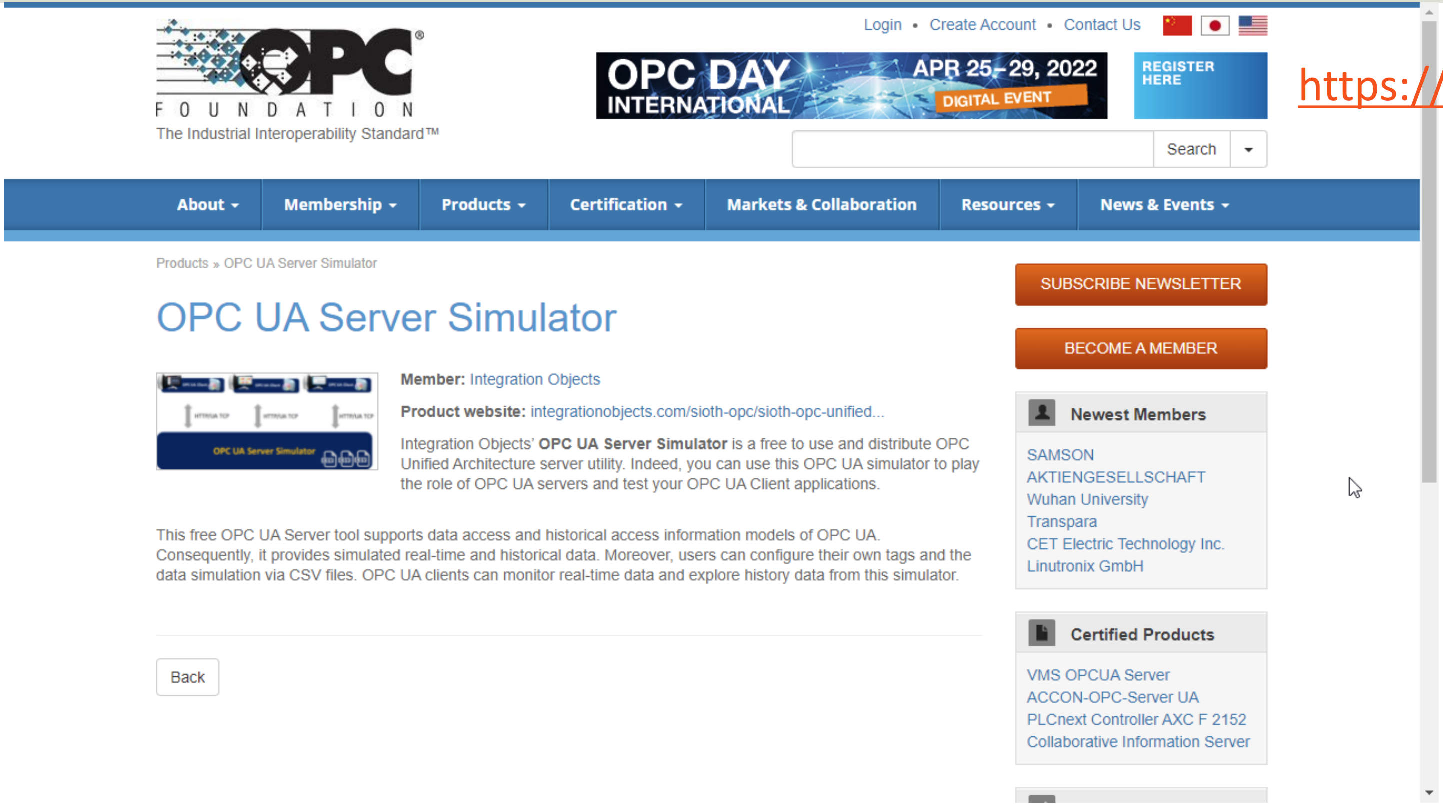
Hans-Petter Halvorsen

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



Products » OPC UA Server Simulator

OPC UA Server Simulator

Member: Integration Objects

Product website: integrationobjects.com/sioth-opc/sioth-opc-unified...

Integration Objects' **OPC UA Server Simulator** is a free to use and distribute OPC Unified Architecture server utility. Indeed, you can use this OPC UA simulator to play the role of OPC UA servers and test your OPC UA Client applications.

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.

Newest Members

SAMSON AKTIENGESELLSCHAFT
Wuhan University
Transpara
CET Electric Technology Inc.
Linutronix GmbH

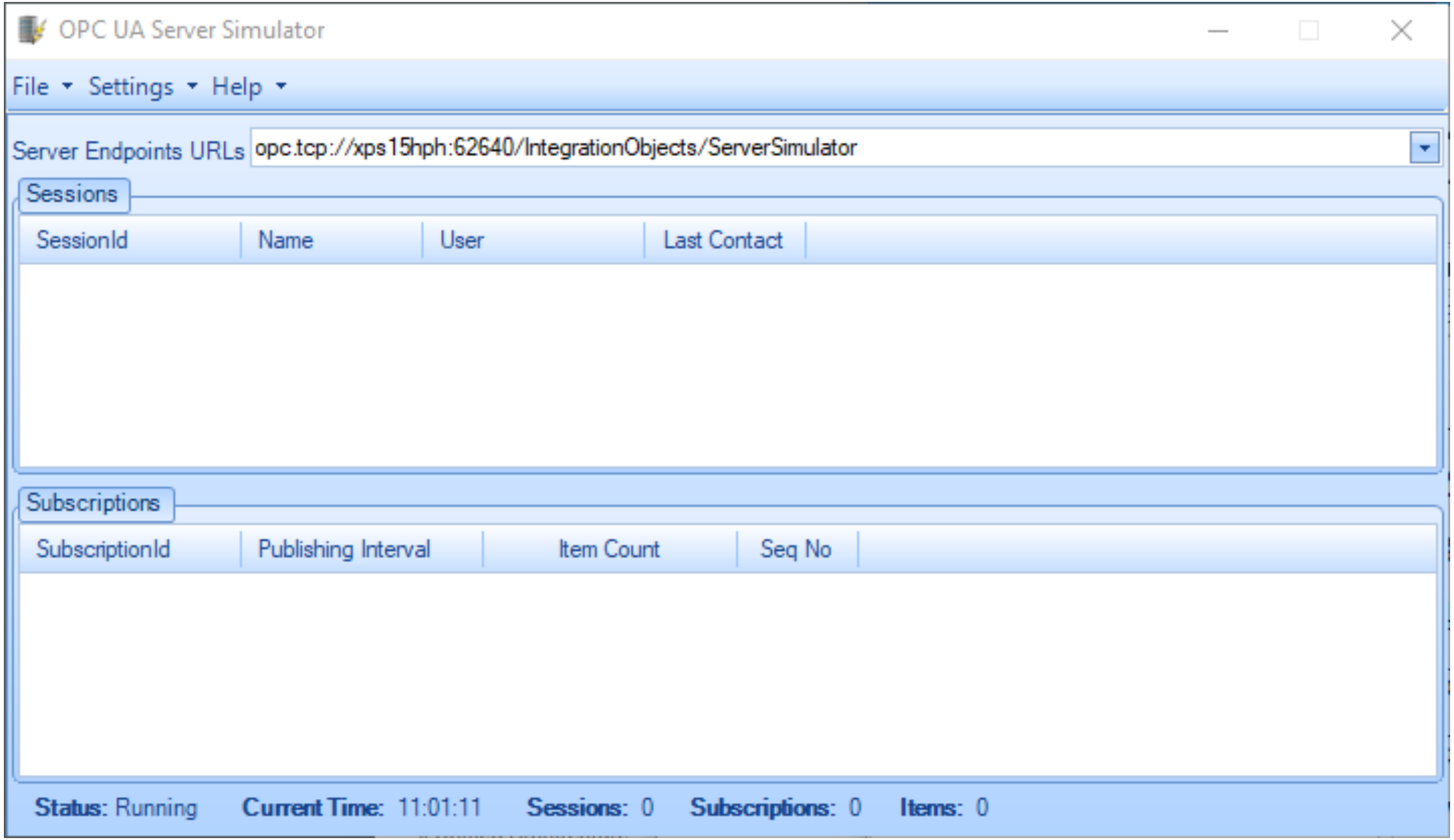
Certified Products

VMS OPCUA Server
ACCON-OPC-Server UA
PLCnext Controller AXC F 2152
Collaborative Information Server

https://opcfoundation.org

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

OPC UA Server Simulator



Server Endpoints URLs opc.tcp://xps15hph:62640/IntegrationObjects/ServerSimulator

Sessions

SessionId	Name	User	Last Contact

Subscriptions

SubscriptionId	Publishing Interval	Item Count	Seq No

Status: Running **Current Time:** 11:01:11 **Sessions:** 0 **Subscriptions:** 0 **Items:** 0

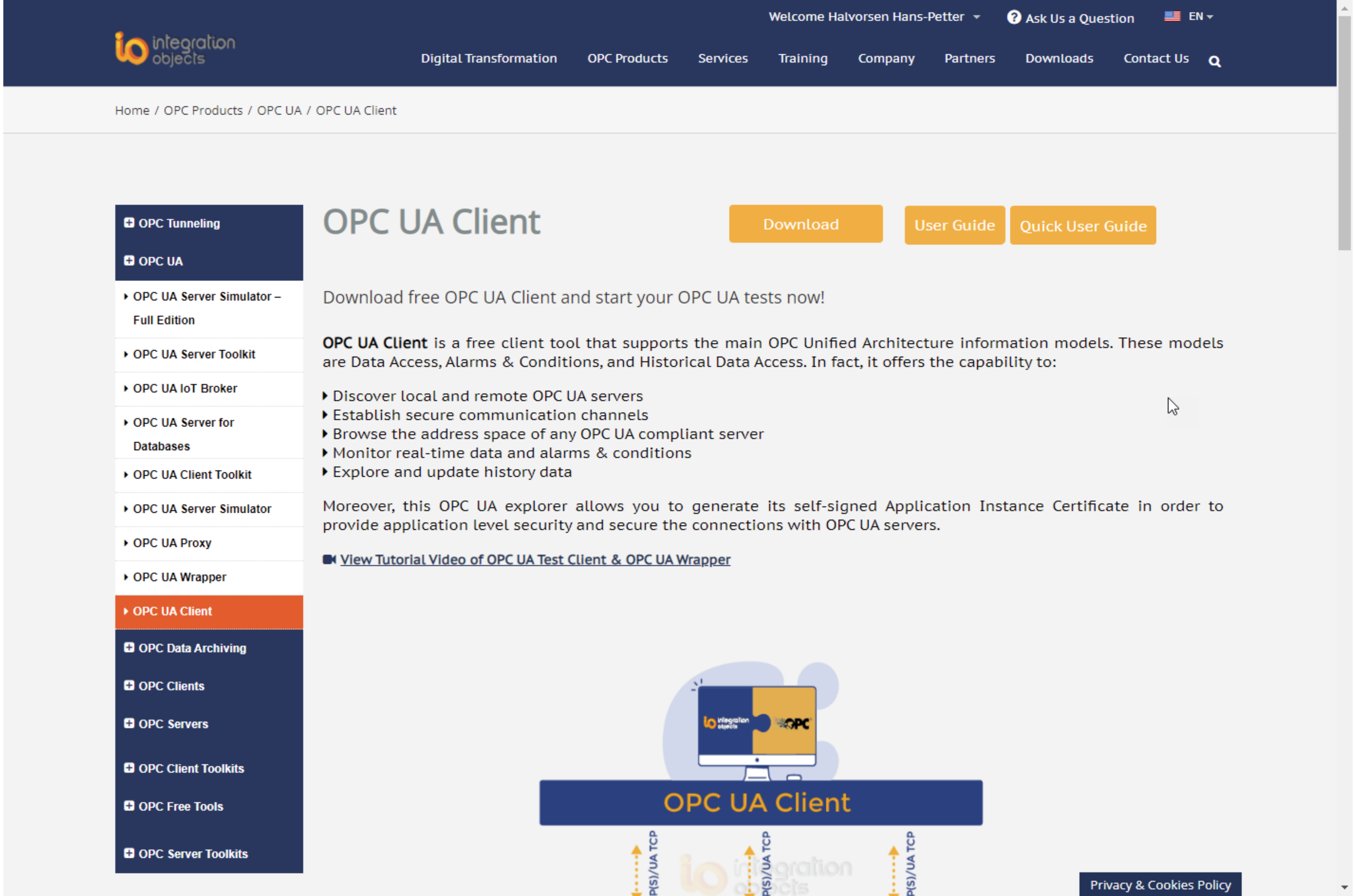
OPC UA Server Simulator

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/



Welcome Halvorsen Hans-Petter ▾ ? Ask Us a Question EN ▾

Digital Transformation | OPC Products | Services | Training | Company | Partners | Downloads | Contact Us

Home / OPC Products / OPC UA / 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!

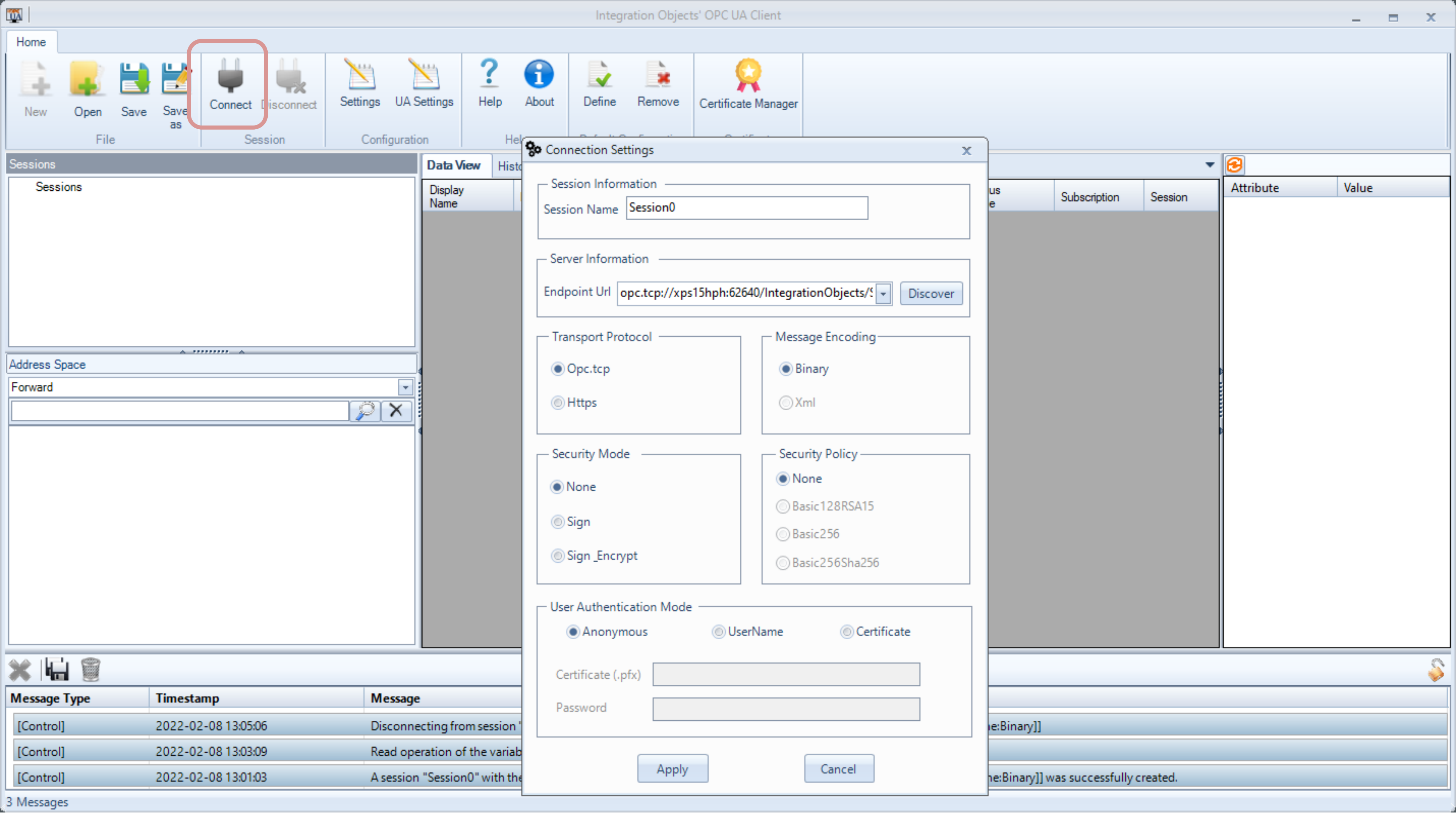
OPC UA Client is a free client tool that supports the main OPC Unified Architecture information models. These models are Data Access, Alarms & Conditions, and Historical Data Access. In fact, it offers the capability to:

- 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

Privacy & Cookies Policy



Connection Settings

Session Information

Session Name: Session0

Server Information

Endpoint Url: opc.tcp://xps15hph:62640/IntegrationObjects/!

Discover

Transport Protocol	Message Encoding
● Opc.tcp	● Binary
○ Https	○ Xml

Security Mode	Security Policy
● None	● None
○ Sign	○ Basic128RSA15
○ Sign _Encrypt	○ Basic256
	○ Basic256Sha256

User Authentication Mode

● Anonymous ○ UserName ○ Certificate

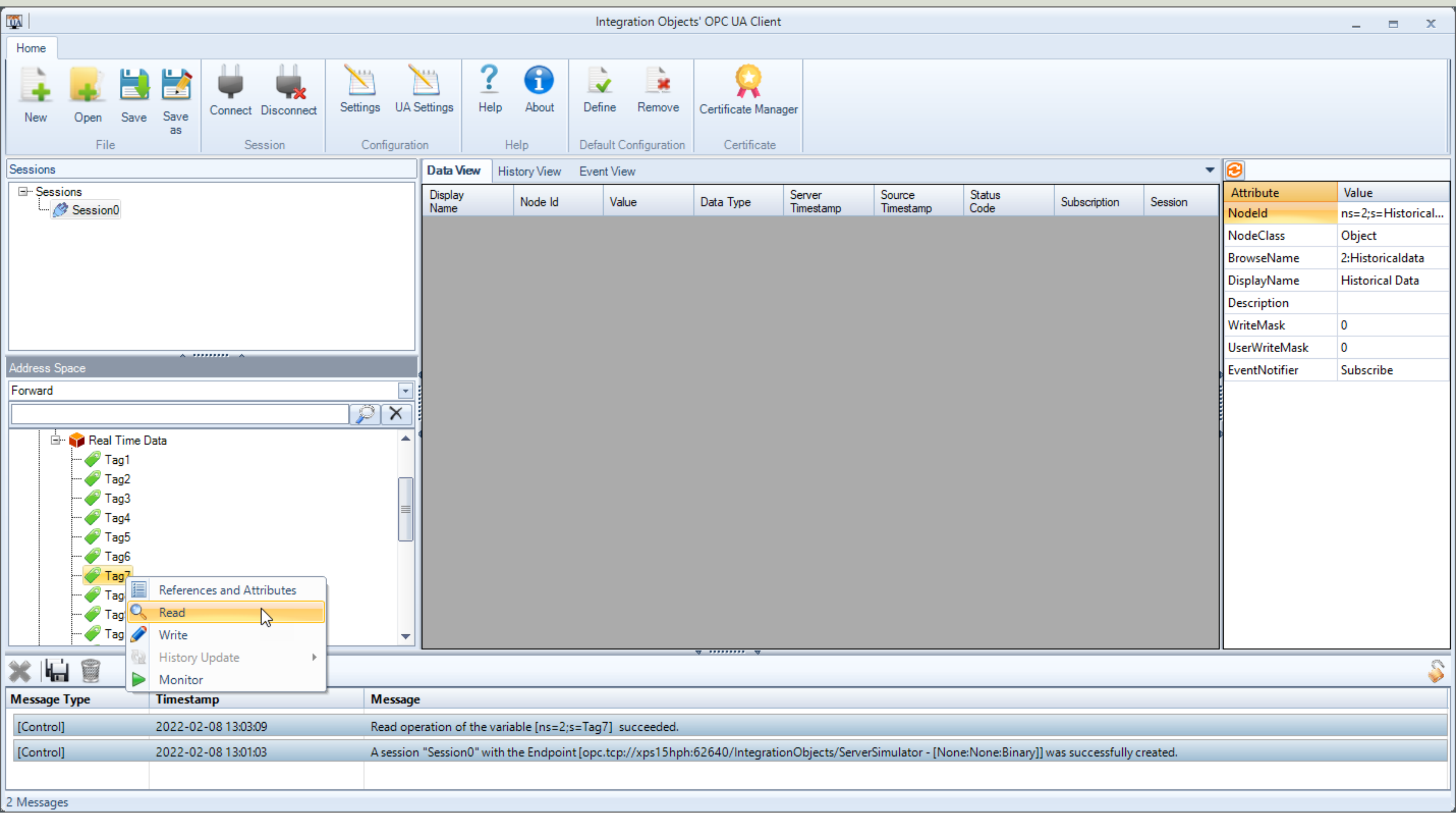
Certificate (.pfx):

Password:

Apply Cancel

Message Type	Timestamp	Message
[Control]	2022-02-08 13:05:06	Disconnecting from session ' … e:Binary]]
[Control]	2022-02-08 13:03:09	Read operation of the variab
[Control]	2022-02-08 13:01:03	A session "Session0" with the … e:Binary]] was successfully created.

3 Messages



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

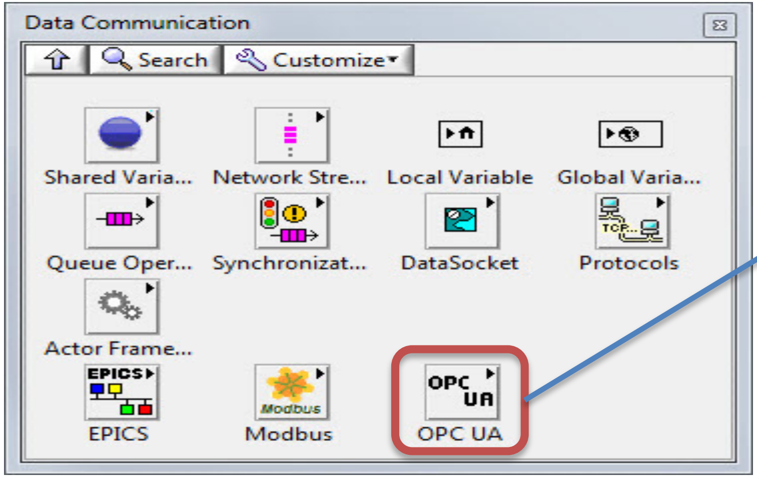
OPC UA in LabVIEW

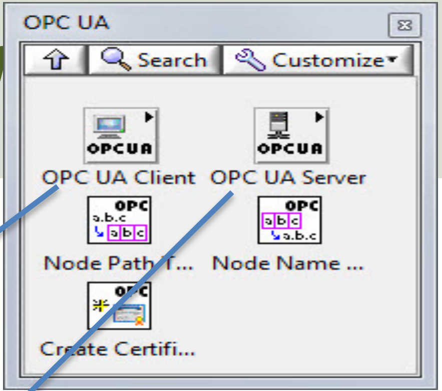
Hans-Petter Halvorsen

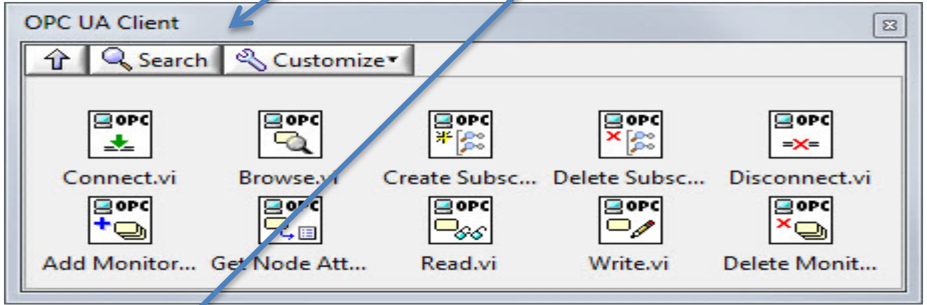
Table of Contents

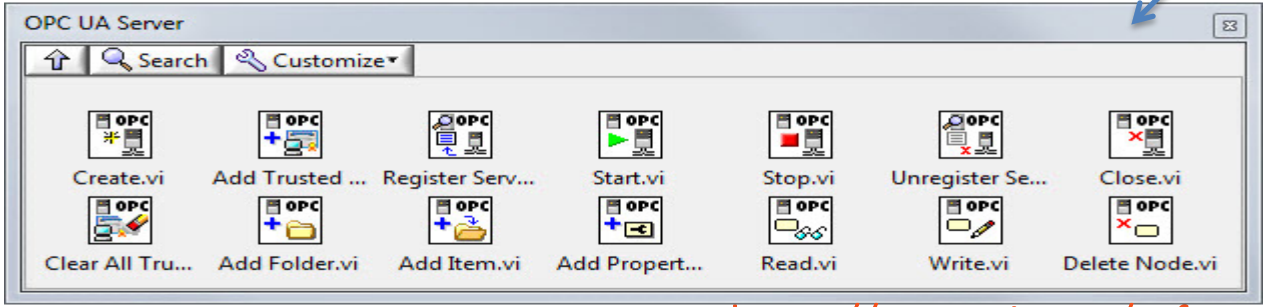
OPC UA in LabVIEW

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





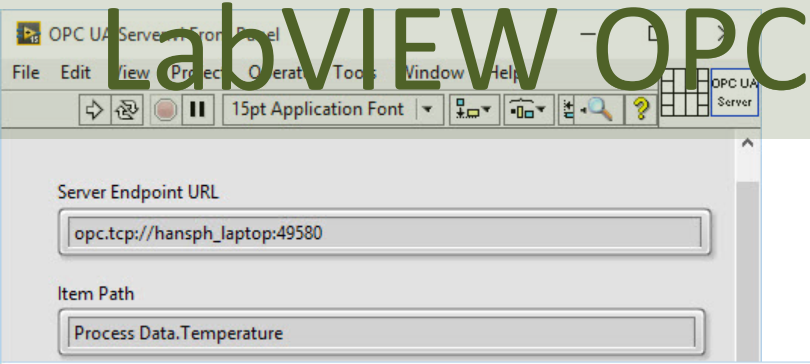




Note! You need to install the **LabVIEW OPC UA Toolkit**

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

LabVIEW OPC UA Server Example

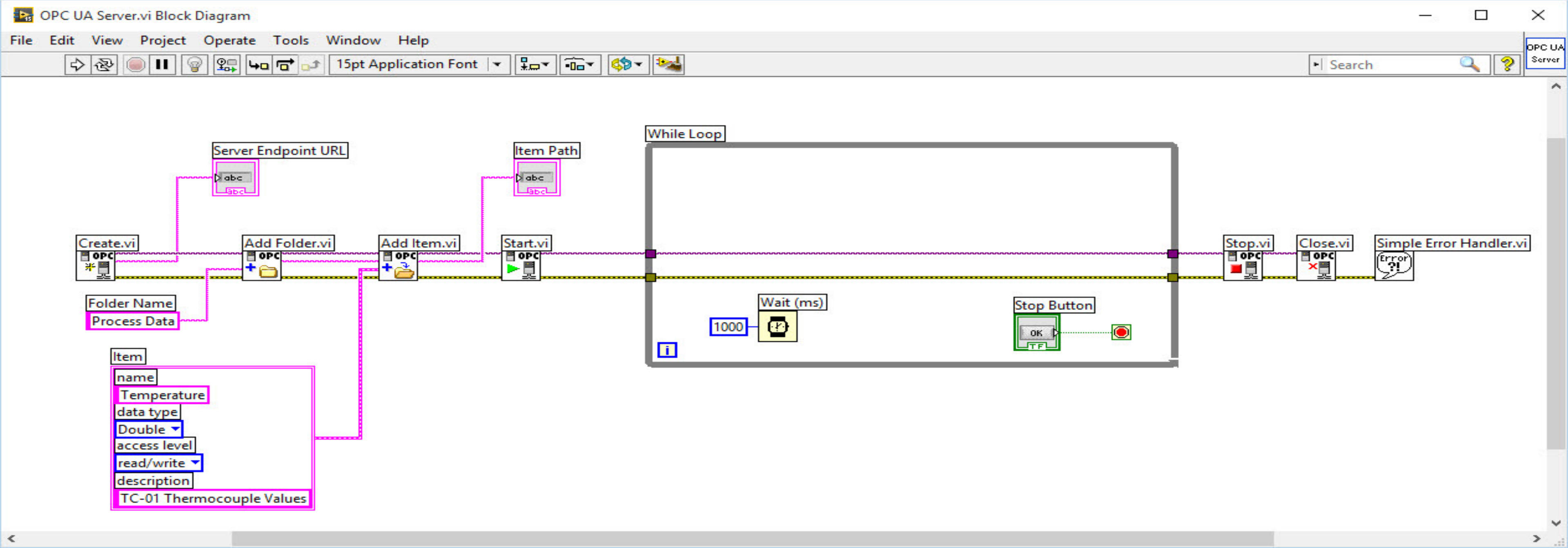


Server Endpoint URL

opc.tcp://hansph_laptop:49580

Item Path

Process Data.Temperature



Folder Name

Process Data

Item

name: Temperature

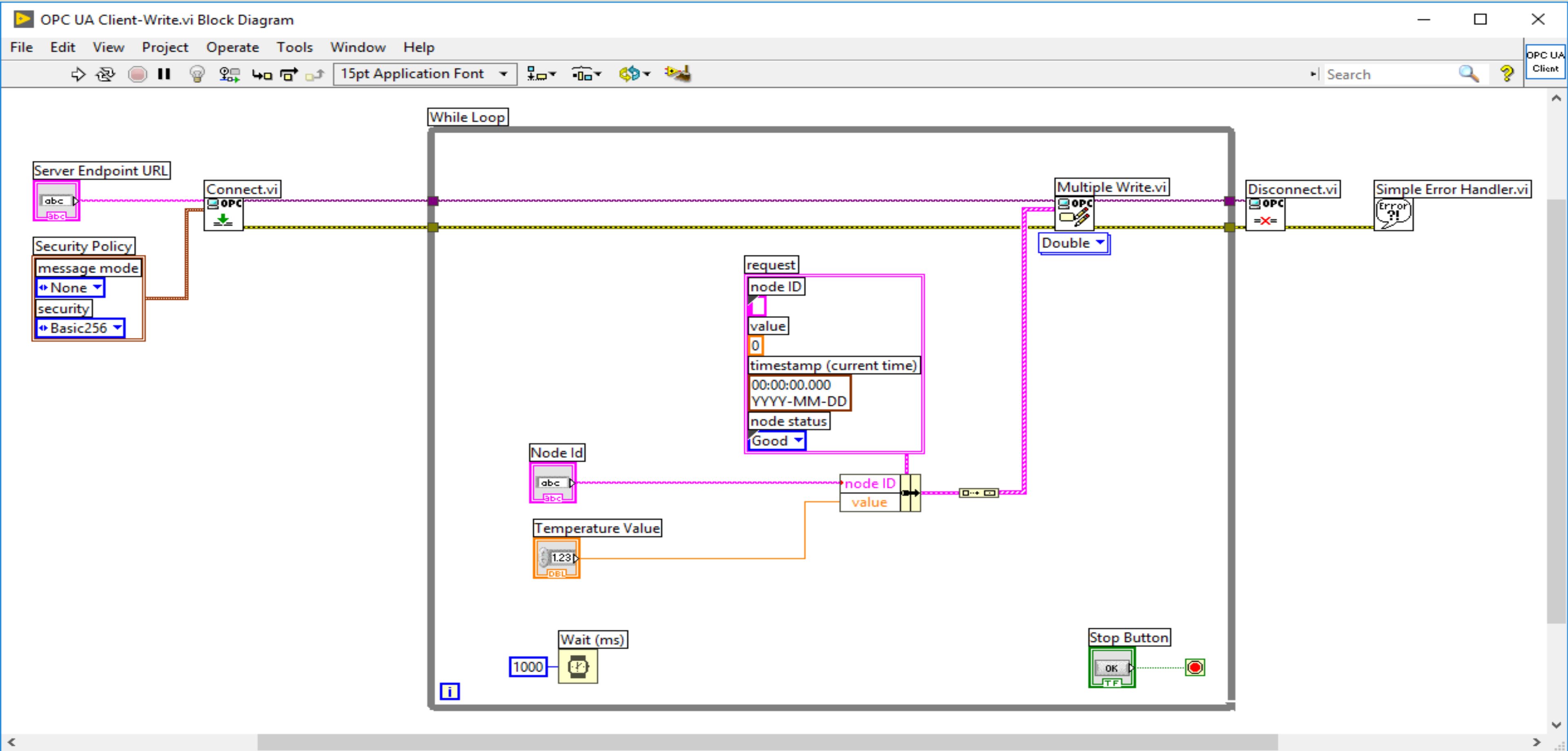
data type: Double

access level: read/write

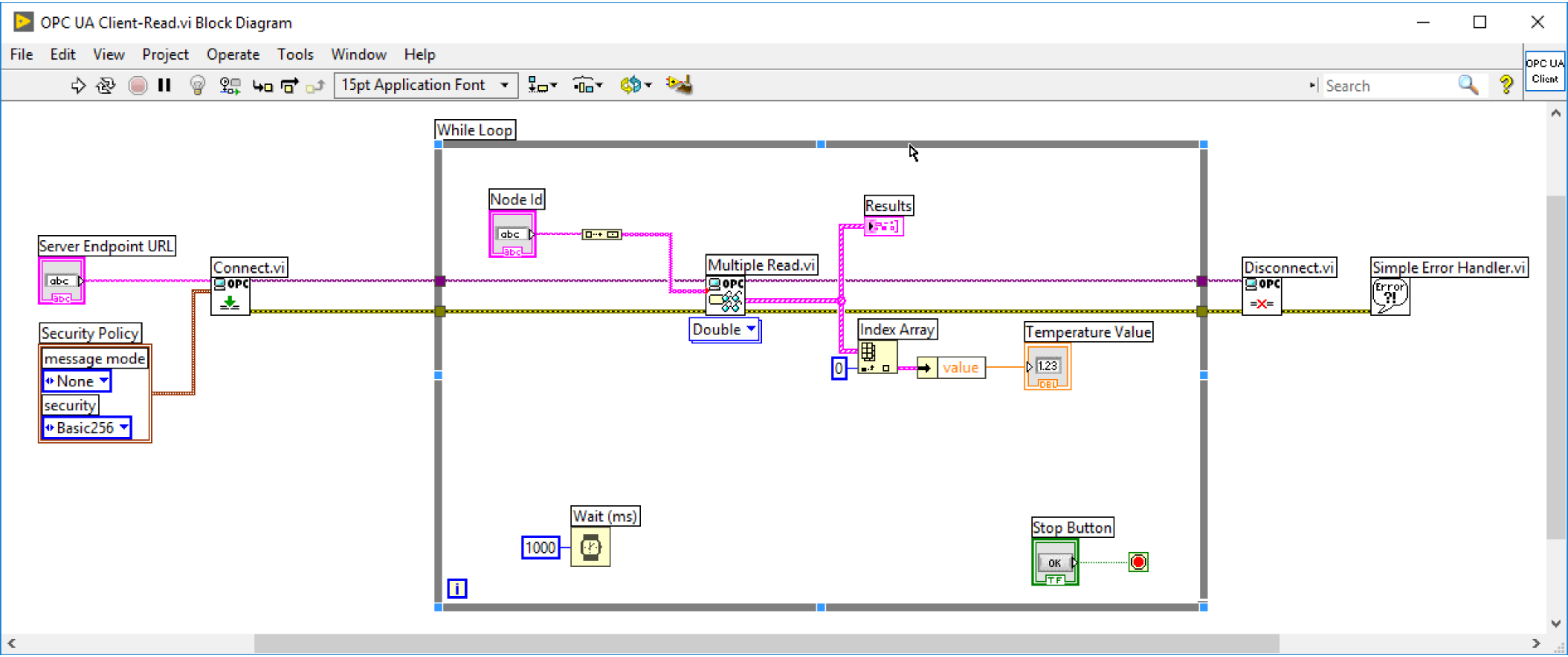
description: TC-01 Thermocouple Values

Wait (ms): 1000

LabVIEW OPC UA Client - Write



LabVIEW OPC UA Client - Read



message mode: None

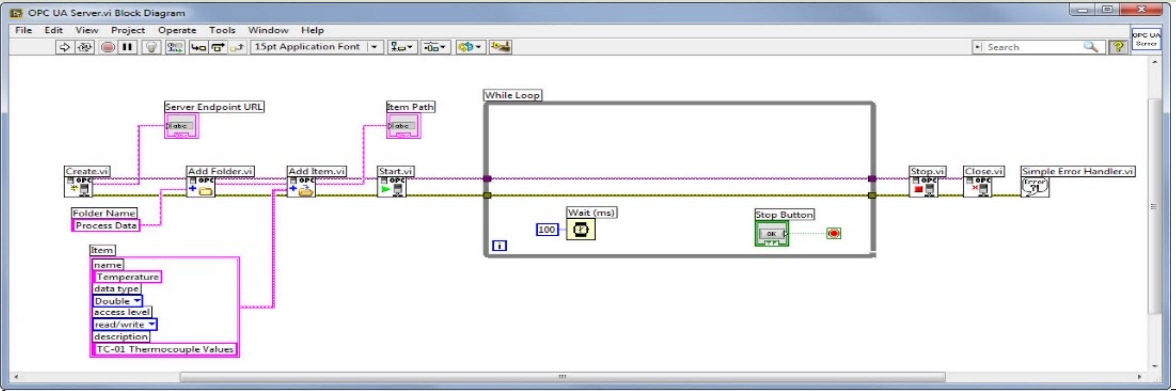
security: Basic256

Double

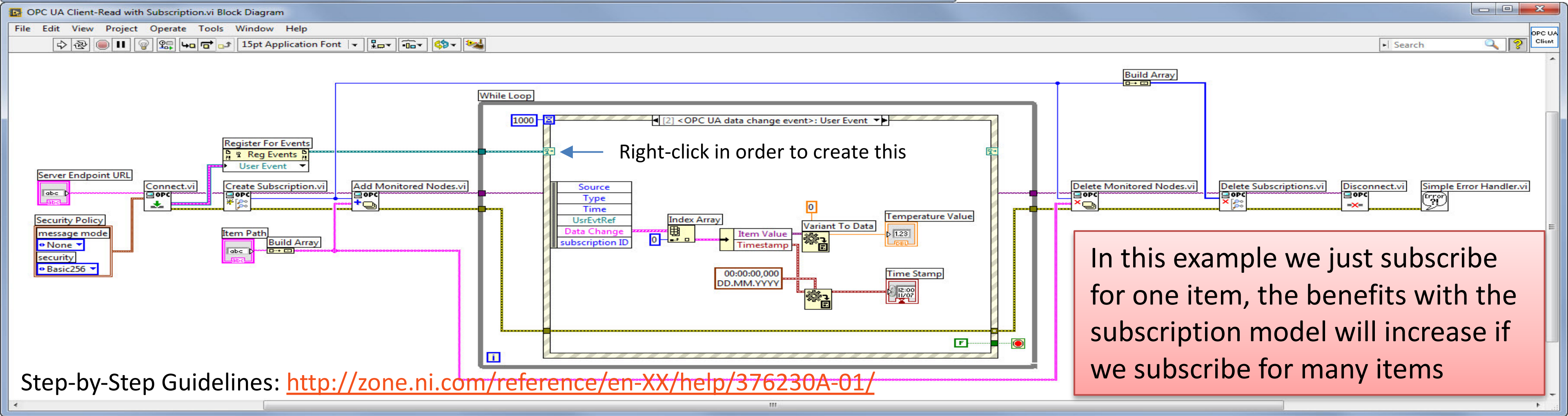
0

1000

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



Right-click in order to create this

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

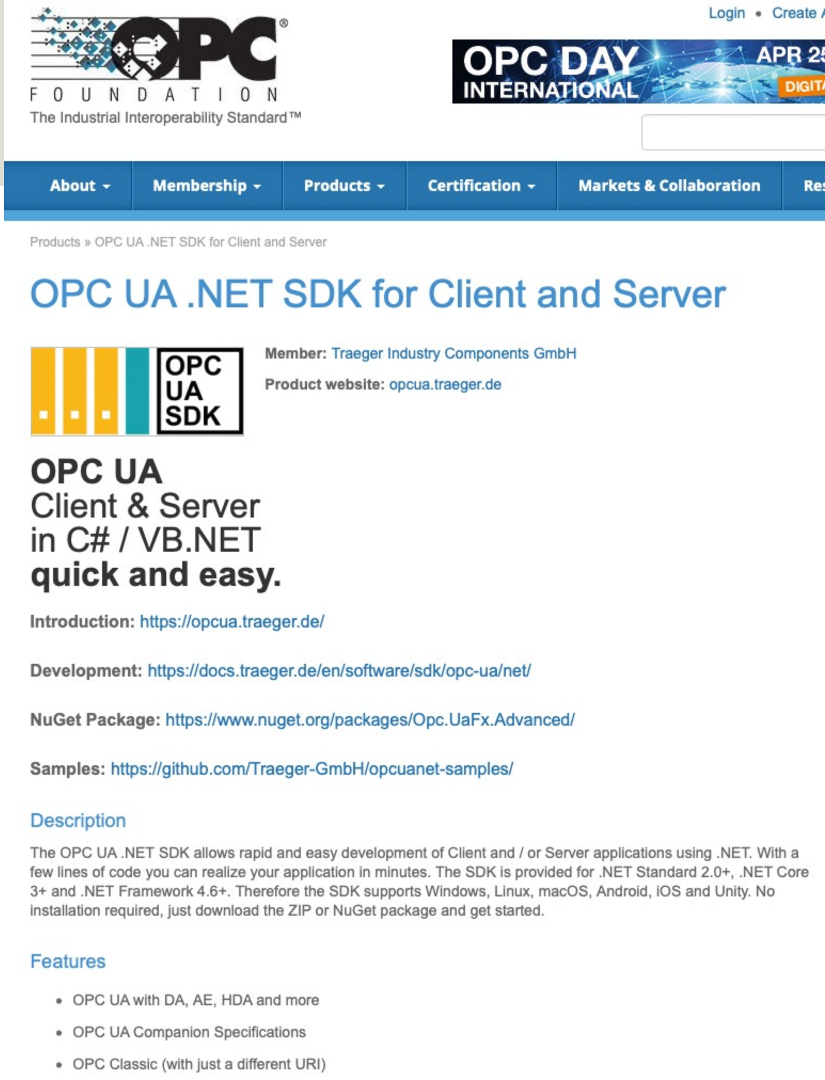
OPC UA in Visual Studio/C#

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



Products » OPC UA .NET SDK for Client and Server

OPC UA .NET SDK for Client and Server

Member: Traeger Industry Components GmbH

Product website: opcua.traeger.de

OPC UA
Client & Server
in C# / VB.NET
quick and easy.

Introduction: https://opcua.traeger.de/

Development: https://docs.traeger.de/en/software/sdk/opc-ua/net/

NuGet Package: https://www.nuget.org/packages/Opc.UaFx.Advanced/

Samples: https://github.com/Traeger-GmbH/opcuanet-samples/

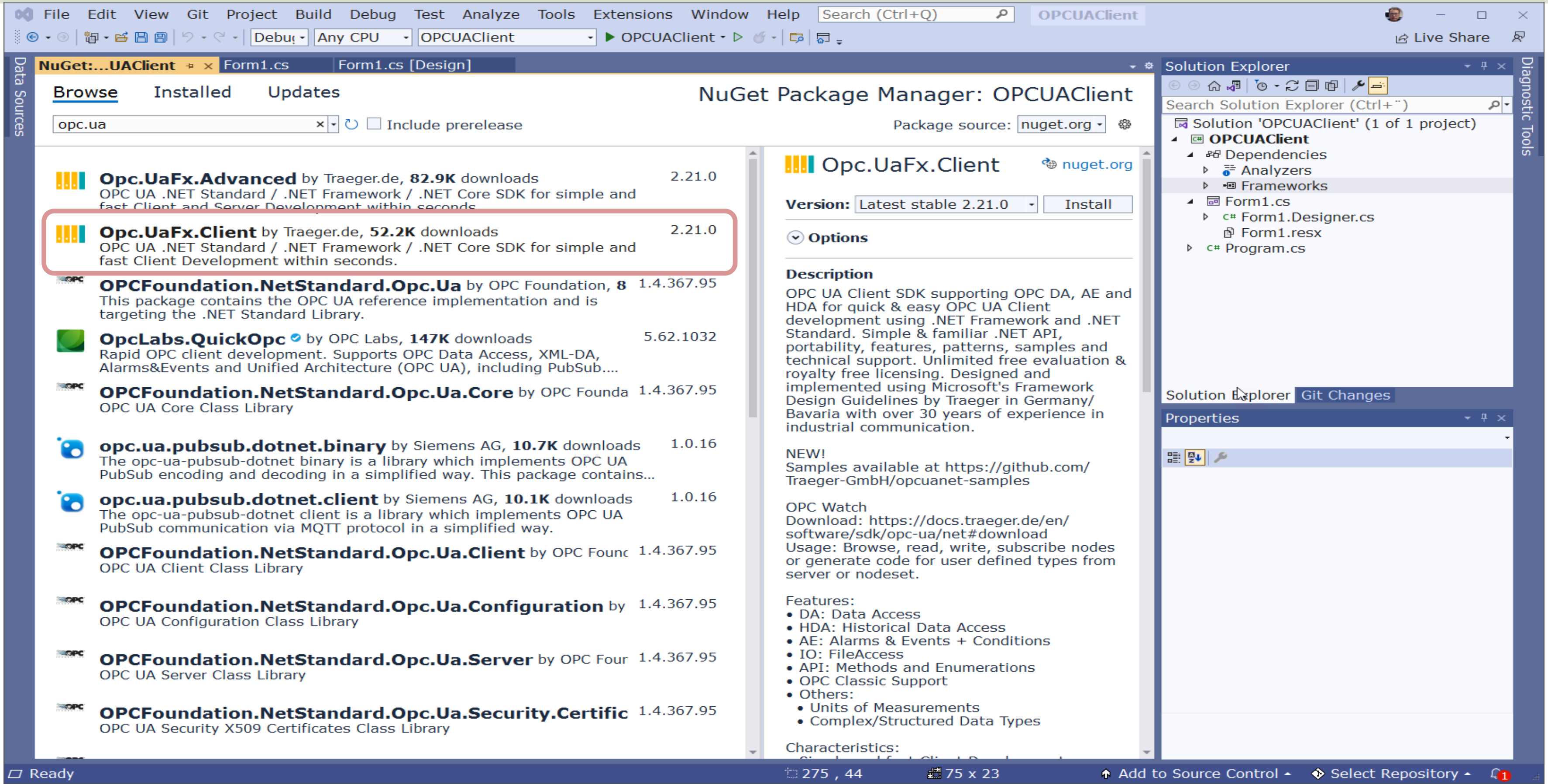
Description

The OPC UA .NET SDK allows rapid and easy development of Client and / or Server applications using .NET. With a few lines of code you can realize your application in minutes. The SDK is provided for .NET Standard 2.0+, .NET Core 3+ and .NET Framework 4.6+. Therefore the SDK supports Windows, Linux, macOS, Android, iOS and Unity. No installation required, just download the ZIP or NuGet package and get started.

Features

- OPC UA with DA, AE, HDA and more
- OPC UA Companion Specifications
- OPC Classic (with just a different URI)

NuGet Package



NuGet Package Manager: OPCUAClient

opc.ua

□ Include prerelease

Package source: nuget.org

Opc.UaFx.Advanced by Traeger.de, **82.9K** downloads — 2.21.0
OPC UA .NET Standard / .NET Framework / .NET Core SDK for simple and fast Client and Server Development within seconds.

Opc.UaFx.Client by Traeger.de, **52.2K** downloads — 2.21.0
OPC UA .NET Standard / .NET Framework / .NET Core SDK for simple and fast Client Development within seconds.

OPCFoundation.NetStandard.Opc.Ua by OPC Foundation, 8 — 1.4.367.95
This package contains the OPC UA reference implementation and is targeting the .NET Standard Library.

OpcLabs.QuickOpc by OPC Labs, **147K** downloads — 5.62.1032
Rapid OPC client development. Supports OPC Data Access, XML-DA, Alarms&Events and Unified Architecture (OPC UA), including PubSub....

OPCFoundation.NetStandard.Opc.Ua.Core by OPC Founda — 1.4.367.95
OPC UA Core Class Library

opc.ua.pubsub.dotnet.binary by Siemens AG, **10.7K** downloads — 1.0.16
The opc-ua-pubsub-dotnet binary is a library which implements OPC UA PubSub encoding and decoding in a simplified way. This package contains...

opc.ua.pubsub.dotnet.client by Siemens AG, **10.1K** downloads — 1.0.16
The opc-ua-pubsub-dotnet client is a library which implements OPC UA PubSub communication via MQTT protocol in a simplified way.

OPCFoundation.NetStandard.Opc.Ua.Client by OPC Founc — 1.4.367.95
OPC UA Client Class Library

OPCFoundation.NetStandard.Opc.Ua.Configuration by — 1.4.367.95
OPC UA Configuration Class Library

OPCFoundation.NetStandard.Opc.Ua.Server by OPC Four — 1.4.367.95
OPC UA Server Class Library

OPCFoundation.NetStandard.Opc.Ua.Security.Certific — 1.4.367.95
OPC UA Security X509 Certificates Class Library

Opc.UaFx.Client — nuget.org

Version: Latest stable 2.21.0 — Install

Options

Description

OPC UA Client SDK supporting OPC DA, AE and HDA for quick & easy OPC UA Client development using .NET Framework and .NET Standard. Simple & familiar .NET API, portability, features, patterns, samples and technical support. Unlimited free evaluation & royalty free licensing. Designed and implemented using Microsoft's Framework Design Guidelines by Traeger in Germany/Bavaria with over 30 years of experience in industrial communication.

NEW!
Samples available at https://github.com/Traeger-GmbH/opcuanet-samples

OPC Watch
Download: https://docs.traeger.de/en/software/sdk/opc-ua/net#download
Usage: Browse, read, write, subscribe nodes or generate code for user defined types from server or nodeset.

Features:
- DA: Data Access
- HDA: Historical Data Access
- AE: Alarms & Events + Conditions
- IO: FileAccess
- API: Methods and Enumerations
- OPC Classic Support
- Others:
  - Units of Measurements
  - Complex/Structured Data Types

Characteristics:

Solution Explorer

Search Solution Explorer (Ctrl+¨)

- Solution 'OPCUAClient' (1 of 1 project)
  - **OPCUAClient**
    - Dependencies
      - Analyzers
      - Frameworks
    - Form1.cs
      - Form1.Designer.cs
      - Form1.resx
    - Program.cs

Solution Explorer | Git Changes

Properties

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();
}
```

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

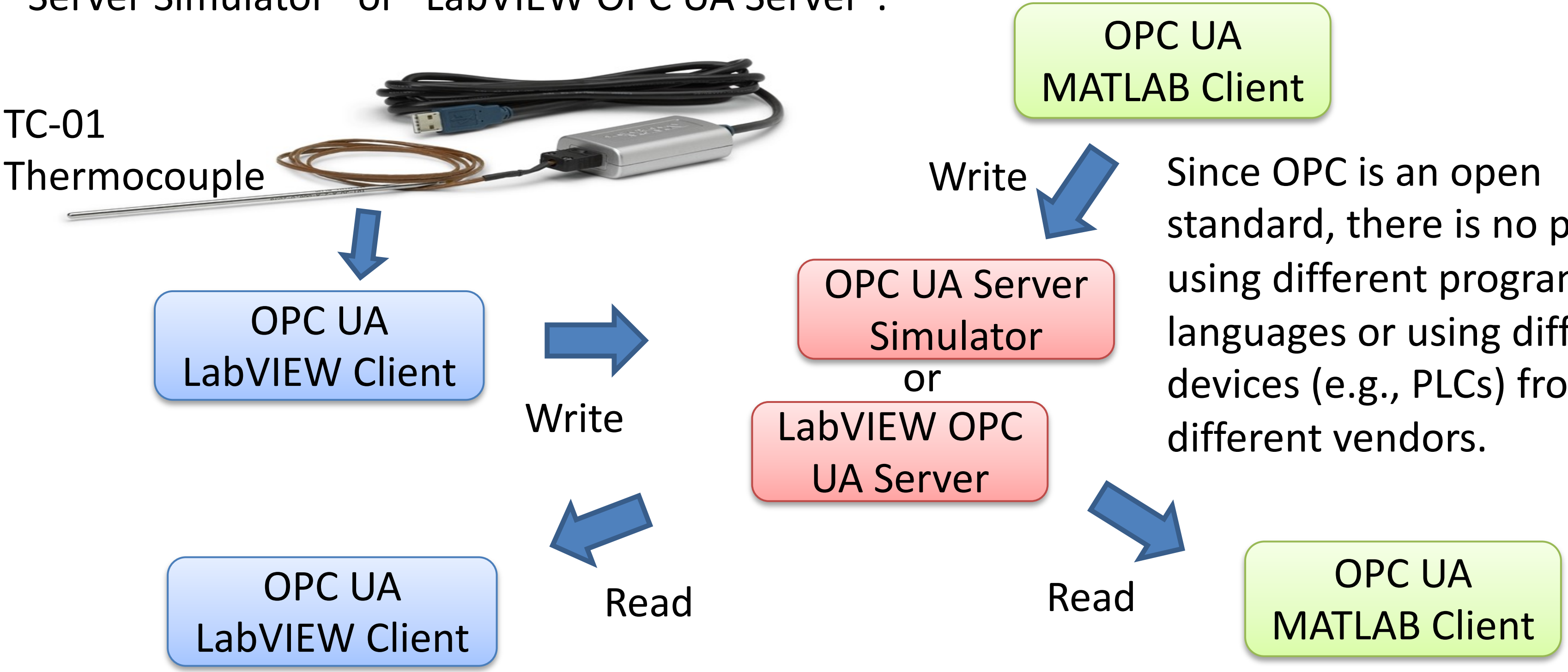
   ```
   disconnect(uaClient)
   ```

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

Read

OPC UA LabVIEW Client

Read

OPC UA MATLAB Client

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

https://www.halvorsen.blog

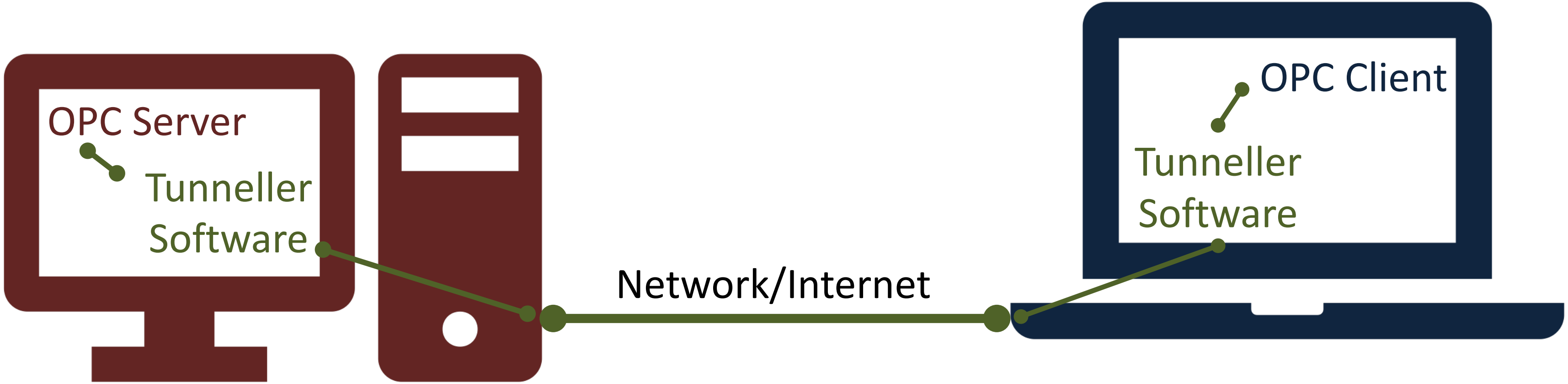
OPC in Network and Tunneling

Hans-Petter Halvorsen

Table of Contents

OPC Tunneller

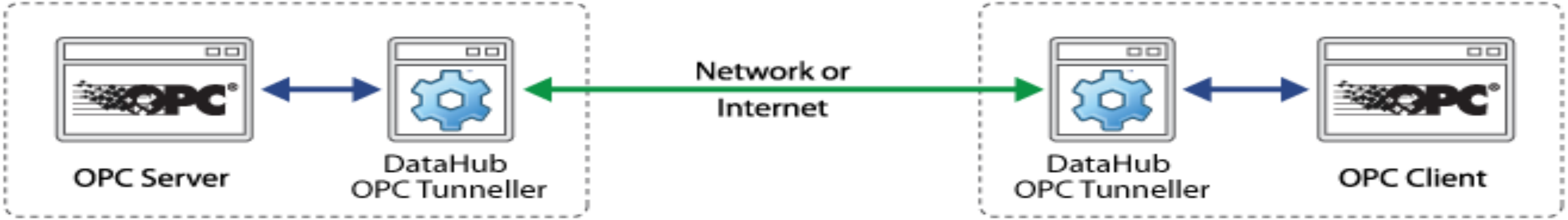
Problem: Sending OPC Data between 2 (or more) Computers in a Network, or even worse, over Internet. OPC DA uses COM/DCOM. This makes it complicated to make it work in a modern Network



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