

<https://www.halvorsen.blog>



LabVIEW in Automation

Datalogging and Monitoring in LabVIEW

Hans-Petter Halvorsen

Contents

- Introduction
- Datalogging and Monitoring Examples
- OPC
 - Matrikon OPC Simulation Server
 - OPC DA
 - OPC UA (The Next Generation OPC)
- Database Systems
 - SQL Server
 - Structured Query Language (SQL)
 - LabVIEW SQL Toolkit
- Delivery

Datalogging and Monitoring

- All Automation Systems or Process Control Systems, SCADA Systems log Data from different Sensors or other Process Information
- These Data are stored into a Database System and/or an OPC Server
- The Data is then Monitored, i.e., showing Plots, Statistics, Alarms, etc.

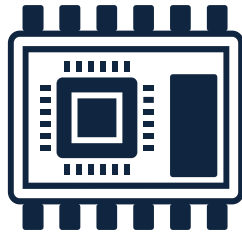
Datalogging and Monitoring

3. Data Analysis and Monitoring

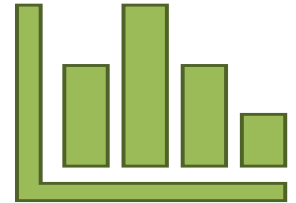
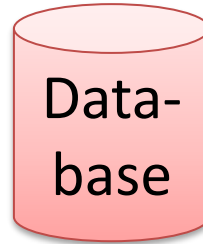
Sensors



DAQ System



Data-
base



Apps

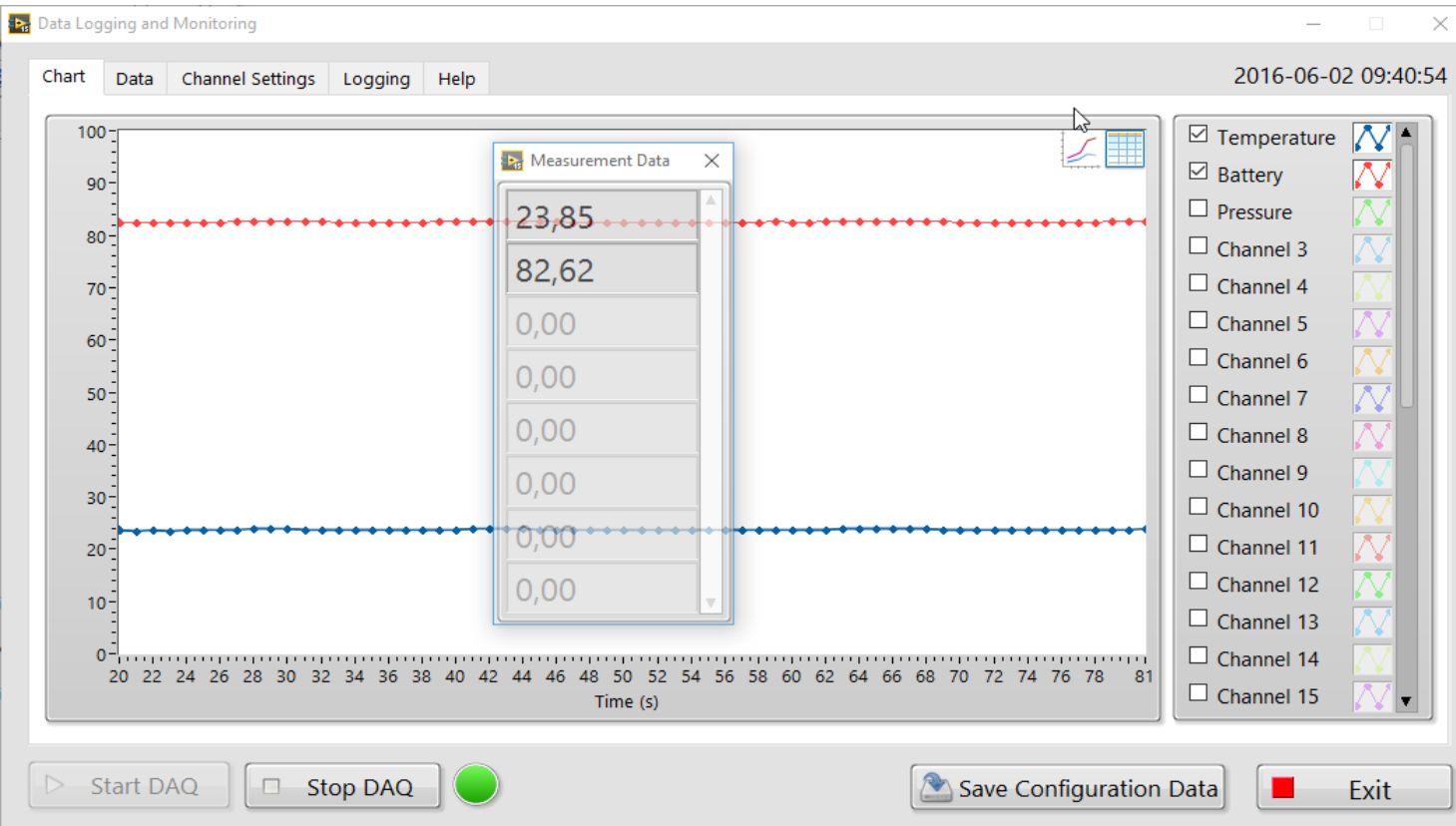


1. Data Collection

2. Data Storage

Datalogging and Monitoring Examples

Below you see an example of a “Datalogging and Monitoring” Application made in LabVIEW:



YouTube Video:

<https://youtu.be/FFnvYuZijel>

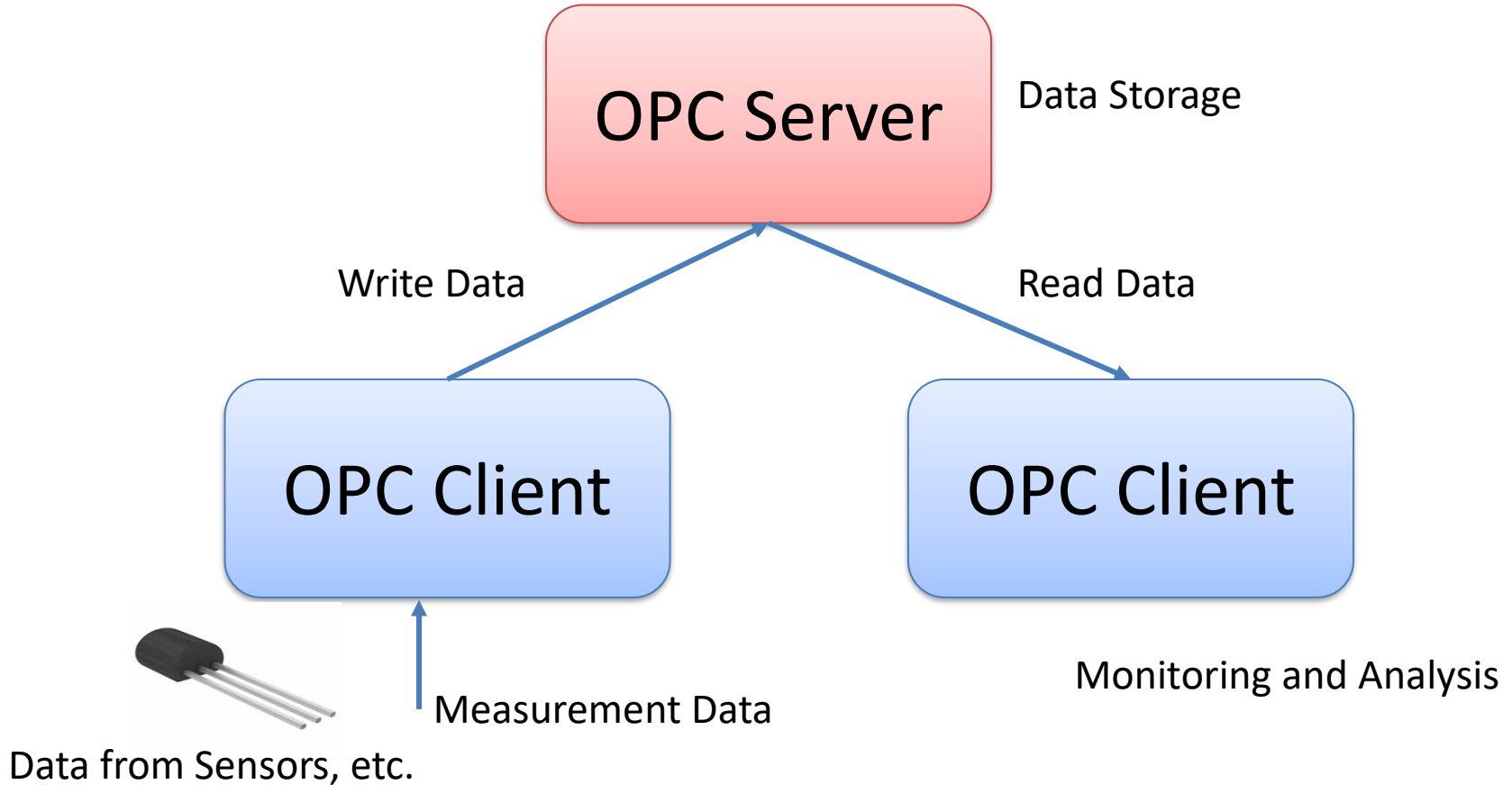


OPC

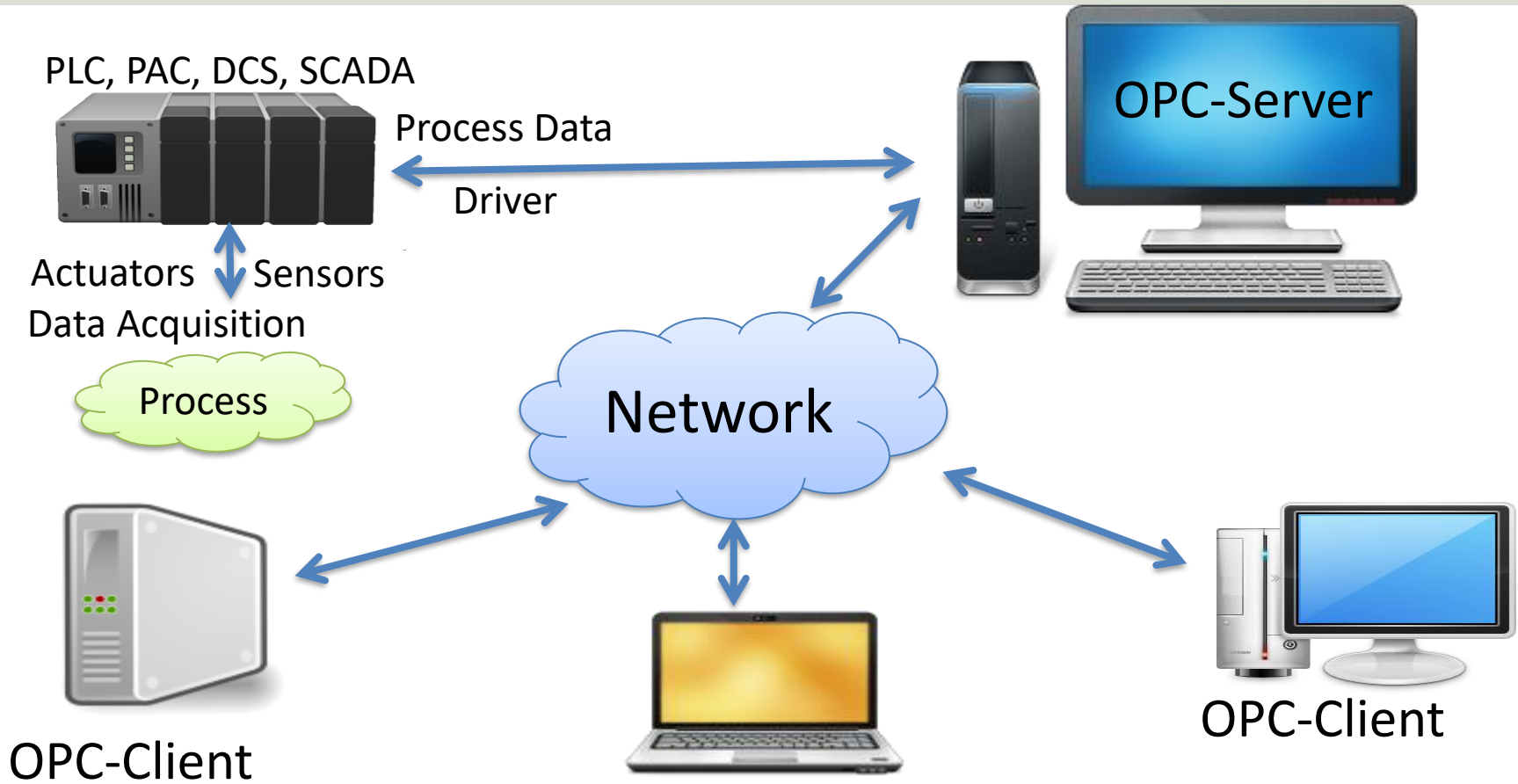
What is OPC?

- 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.

OPC Server and Client(s)



Typical OPC Scenario



OPC Specifications

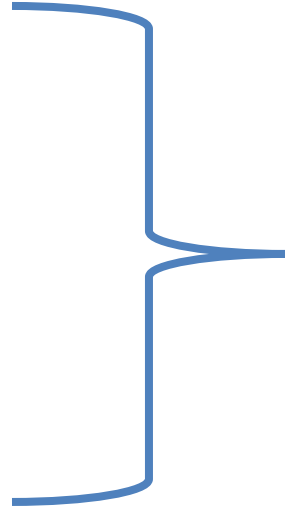
“Classic” OPC

“Next Generation” OPC

OPC DA

OPC HDA

OPC A&E



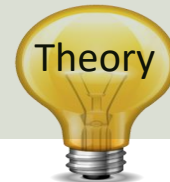
OPC UA

... (Many others)

OPC UA

- UA – Unified Architecture
- The Next Generation OPC
- Cross Platform. “Classic” OPC works only for Windows
- Based on Modern Software/Network Architecture (No DCOM problems!)
- It makes it easier to transmit and receive data in a modern data network/Internet

Next Generation OPC



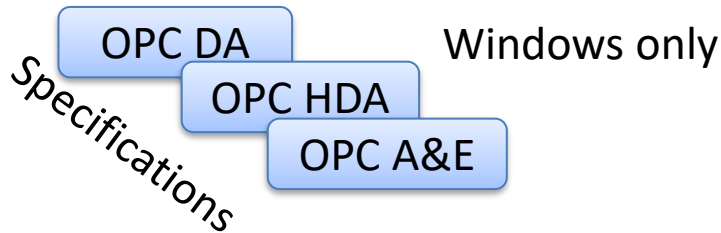
COM/DCOM

OPC Classic

Next Generation OPC

XML, HTTP, SOAP

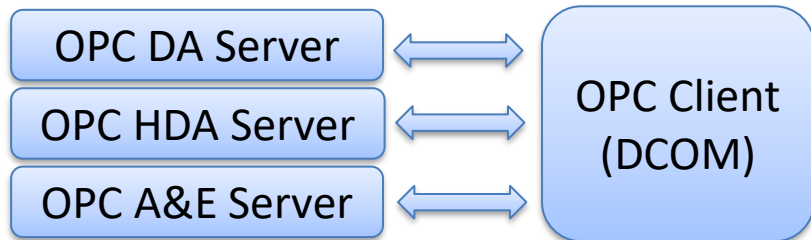
OPC UA



Cross-platform
Windows, Linux, Mac,
Embedded, VxWorks

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

Protocols: "UA Binary" or "UA XML"



Simpler!!

OPC UA Server

OPC UA Client



OPC Software

We will use the following Software:

- Matrikon OPC Simulation Server
 - Free OPC DA Server for Test purposes
- LabVIEW
 - LabVIEW has built-in OPC DA Client
- LabVIEW OPC UA Toolkit
 - Here you can create both OPC UA Servers and OPC UA Clients.
 - If you don't have a valid license, you can use it for free in a limited trial period (some weeks)

Matrikon OPC Simulation Server



MatrikonOPC Explorer - Add Tags



The screenshot shows the MatrikonOPC Explorer software interface. The 'Add Tags' dialog box is open, and the 'Bucket Brigade' folder is selected in the 'Available Items' list. The 'Item ID' field is set to 'Bucket Brigade.Real4'. The 'Data Type' is 'Empty/Default'. The 'Access Path' is empty. The 'Filter' and 'Data Type Filter' are both set to 'Empty/Default'. The 'Write Access' and 'Read Access' checkboxes are checked. The 'Available Items' list shows 'Simulation Items' and 'Bucket Brigade' (circled in red). The 'Available Tags' list shows 'Int1', 'Int2', 'Int4', 'Money', 'Real4', and 'Real8'. The 'Add Tags' button is circled in red. The 'Tags to be added:' list shows 'Bucket Brigade.Real4' (circled in red). The main interface shows the 'Contents of 'Group0'' table with the following data:

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

The 'Finished' text is displayed in the center of the main interface. The 'Server Info' panel shows 'Server: Matrikon.OPC.Simulation.1' and 'Connected: Yes'. The 'Group Info' panel shows 'Group: Group0' and 'Connected (Async I/O): Yes (2.0)'. The status bar shows '1000 ms', '0.00%', and '0.01 Items/Sec'.

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

MatrikonOPC Explorer

The screenshot displays the MatrikonOPC Explorer application window. The left pane shows a tree view of the local network, including 'Localhost \\HANSHA-PC' and 'Matrikon.OPC.Simulation.1'. The main pane shows the 'Contents of 'Group0'' table:

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

A blue callout box with a white border points to the right-click context menu, containing the text: "Right-click in order to Write Data to the OPC Server". The context menu is open over the 'Square Waves.Int4' item in the table, showing options: 'Write Values', 'Deactivate', 'Delete Del', 'Export Items', and 'Properties Alt+Enter'. The 'Server Info' pane at the bottom left shows: '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', and 'Update Local Time: 03/06/2012 10:59:16.300 A'. A small inset image shows a laptop displaying the MatrikonOPC Explorer interface.

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

OPC in LabVIEW

OPC DA in LabVIEW:

- Built-in support using the DataSocket features
- With this feature you can communicate with existing OPC DA Servers like, e.g., the Matrikon OPC Simulation Server

OPC UA in LabVIEW:

- NI OPC UA Toolkit
 - This is an additional Toolkit not part of the the standard LabVIEW software.
 - You need to pay extra for this Toolkit
 - With this Toolkit you can create both OPC UA Servers and OPC UA Clients

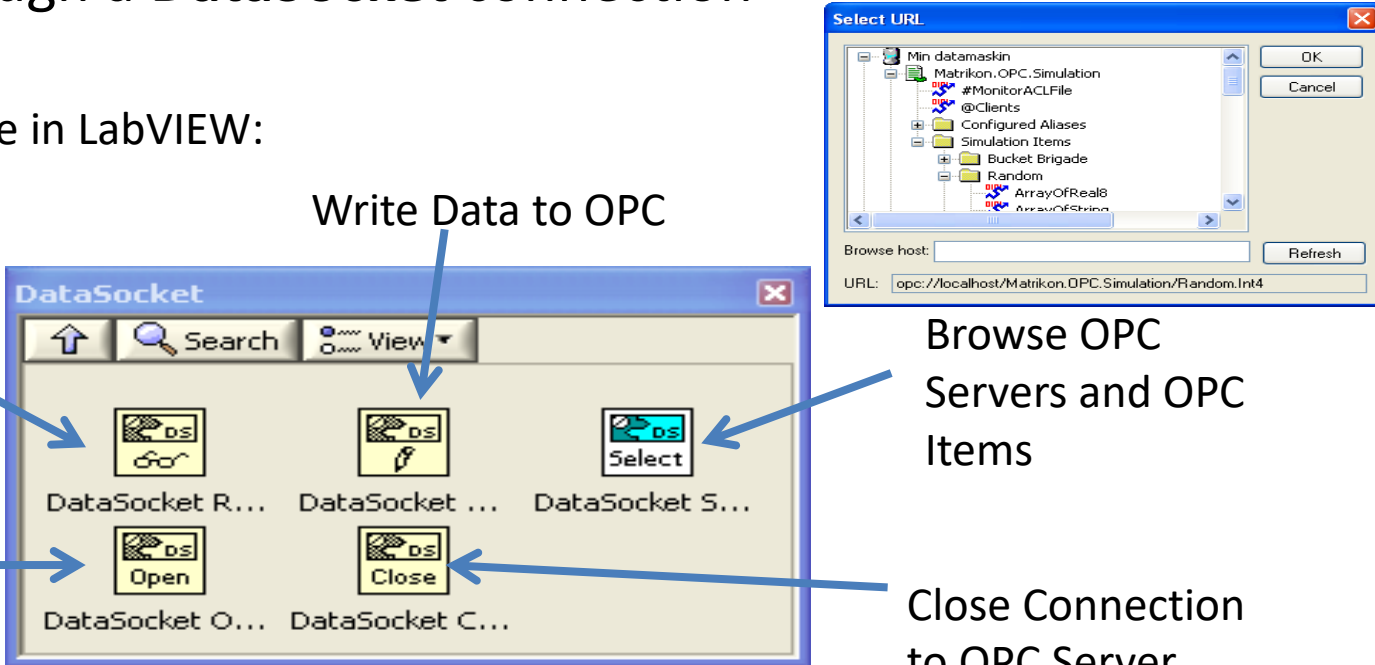


OPC DA

OPC DA in LabVIEW

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

The **DataSocket** palette in LabVIEW:



The image shows the LabVIEW DataSocket palette and a Select URL dialog. The DataSocket palette contains icons for Read Data from OPC, Write Data to OPC, Open Connection to OPC Server, and Close Connection to OPC Server. The Select URL dialog shows a tree view of OPC servers and items, with the URL field set to `opc://localhost/Matrikon.OPC.Simulation/Random.Int4`.

Read Data from OPC

Write Data to OPC

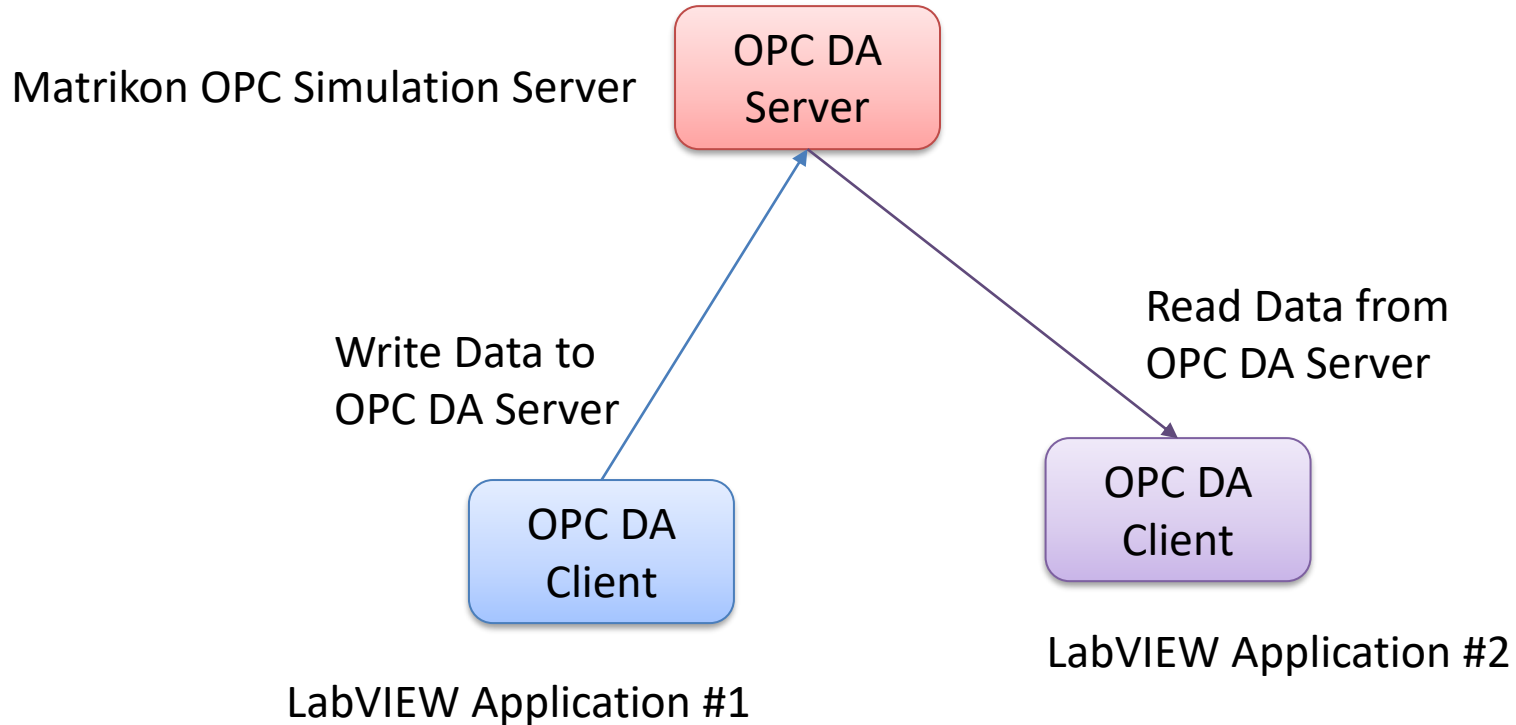
Open Connection to OPC Server

Close Connection to OPC Server

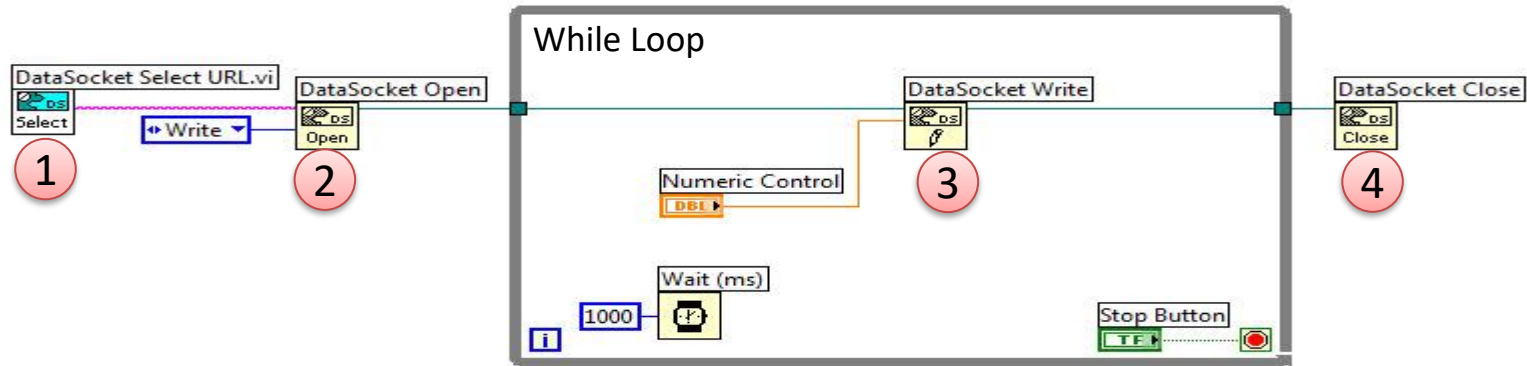
Browse OPC Servers and OPC Items

Select URL dialog showing URL: `opc://localhost/Matrikon.OPC.Simulation/Random.Int4`

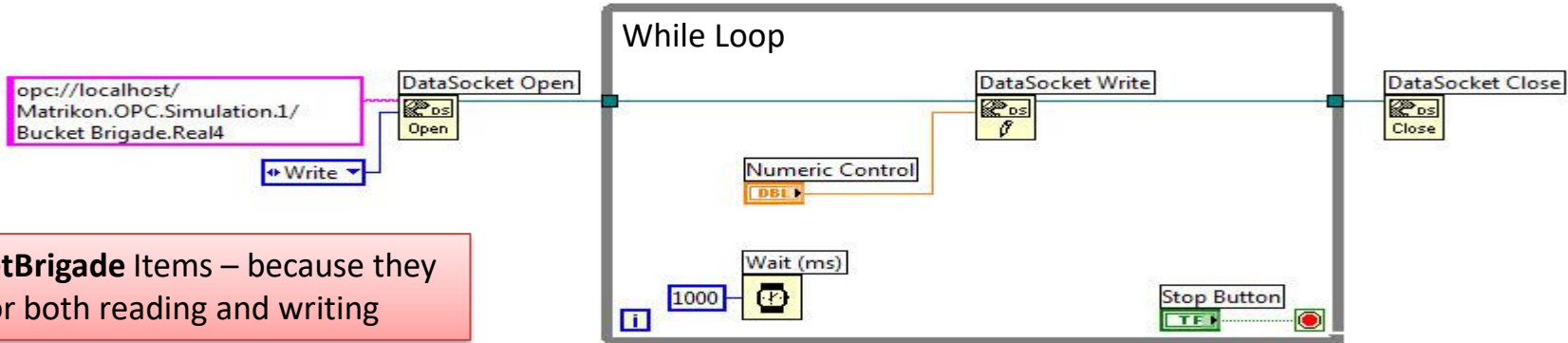
OPC DA LabVIEW Example



Write Data to OPC DA Server



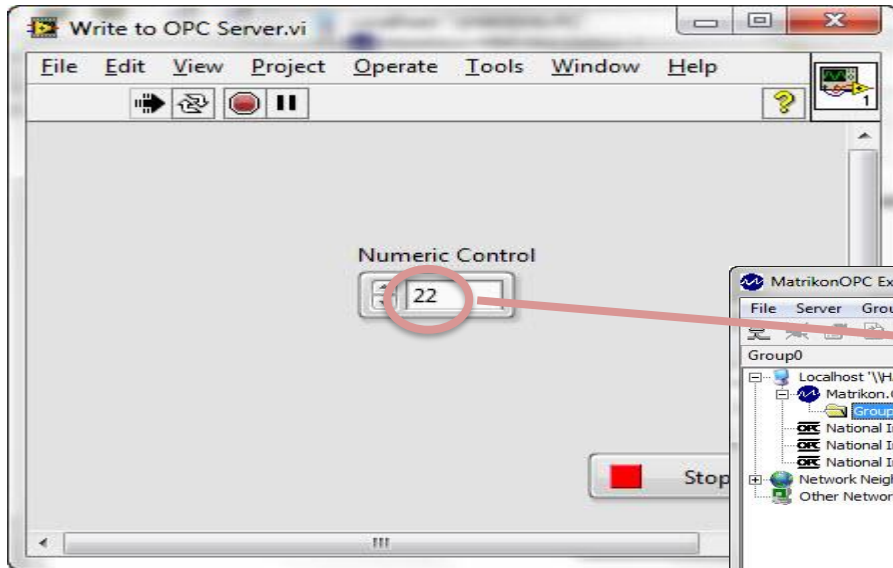
Or specify URL directly:



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

MatrikonOPC Explorer

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



MatrikonOPC Explorer - [Untitled*]

File Server Group Item View Help

Item ID	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 A

Group Info

Group: Group0

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

Active: Yes

Items: 1

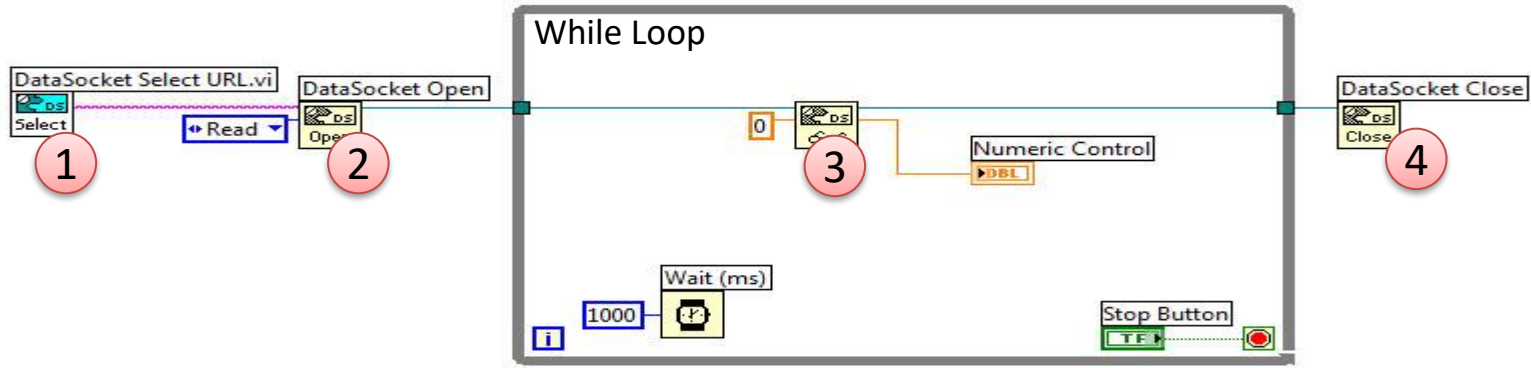
Current Update Rate: 1000 ms

Percent Deadband: 0.00%

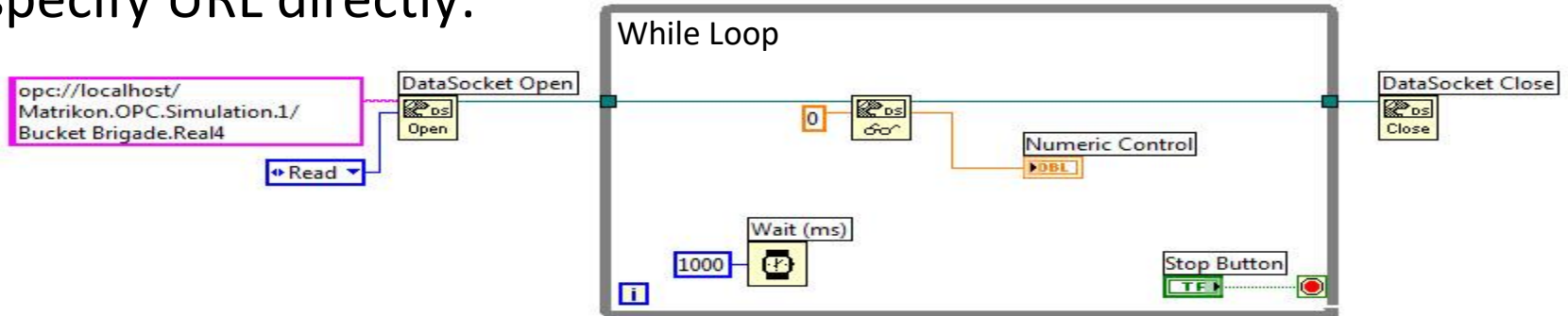
Data Change Rate: 0.01 Items/Sec

MatrikonOPC eLea
Learn on your own time
Click For Details

Read Data from OPC DA Server



Or specify URL directly:





OPC UA

OPC UA Toolkit in LabVIEW

The image displays the LabVIEW Data Communication palette, which is organized into several functional categories. The categories visible include Shared Variable, Network Streams, Local Variable, Global Variable, Queue Operations, Actor Framework, and OPC UA. The OPC UA category is highlighted with a red box and contains two sub-palettes: OPC UA Client and OPC UA Server. The OPC UA Client sub-palette includes tools like Connect.vi, Forward Browse.vi, Create Subscription.vi, Delete Subscriptions.vi, Disconnect.vi, Add Monitored Data Nodes.vi, Get Node Attribute.vi, Multiple Read.vi, Multiple Write.vi, and Delete Monitored... The OPC UA Server sub-palette includes tools like Create.vi, Close.vi, Start.vi, Stop.vi, Add Trusted Clients.vi, Clear All Trusted Clients.vi, Register Server.vi, Unregister Server.vi, Add Folder.vi, Add Item.vi, Add Analog Item.vi, Add Property.vi, Delete Node.vi, Read.vi, Write, Alarms and Conditions, and Historical Access. Red arrows point from the OPC UA Client and Server icons in the main palette to their respective sub-palettes.

Data Communication

↑ Search Customize

Shared Variable

Network Streams

Local Variable

Global Variable

Queue Operations

Actor Framework

OPC UA

OPC UA Client

OPC UA Server

Create Certificate.vi

OPC UA Server

↑ Search Customize

Create.vi

Close.vi

Start.vi

Stop.vi

Add Trusted Clients.vi

Clear All Trusted Clients.vi

Register Server.vi

Unregister Server.vi

Add Folder.vi

Add Item.vi

Add Analog Item.vi

Add Property.vi

Delete Node.vi

OPC UA Client

↑ Search Customize

Connect.vi

Forward Browse.vi

Create Subscription.vi

Delete Subscriptions.vi

Disconnect.vi

Add Monitored Data Nodes.vi

Get Node Attribute.vi

Multiple Read.vi

Multiple Write.vi

Delete Monitored ...

Alarms and Conditions

Historical Access

OPC UA

Example



LabVIEW Application #1

Read Data from OPC UA Server



LabVIEW Application #3

In this Example LabVIEW Application #1, #2 and #3 are on the same computer. Normally they are located on different computers or devices in a Network.

Write Data to OPC UA Server



LabVIEW Application #2

OPC UA Server Example in LabVIEW



OPC UA Server.vi

File Edit View Project Operate Tools Window Help

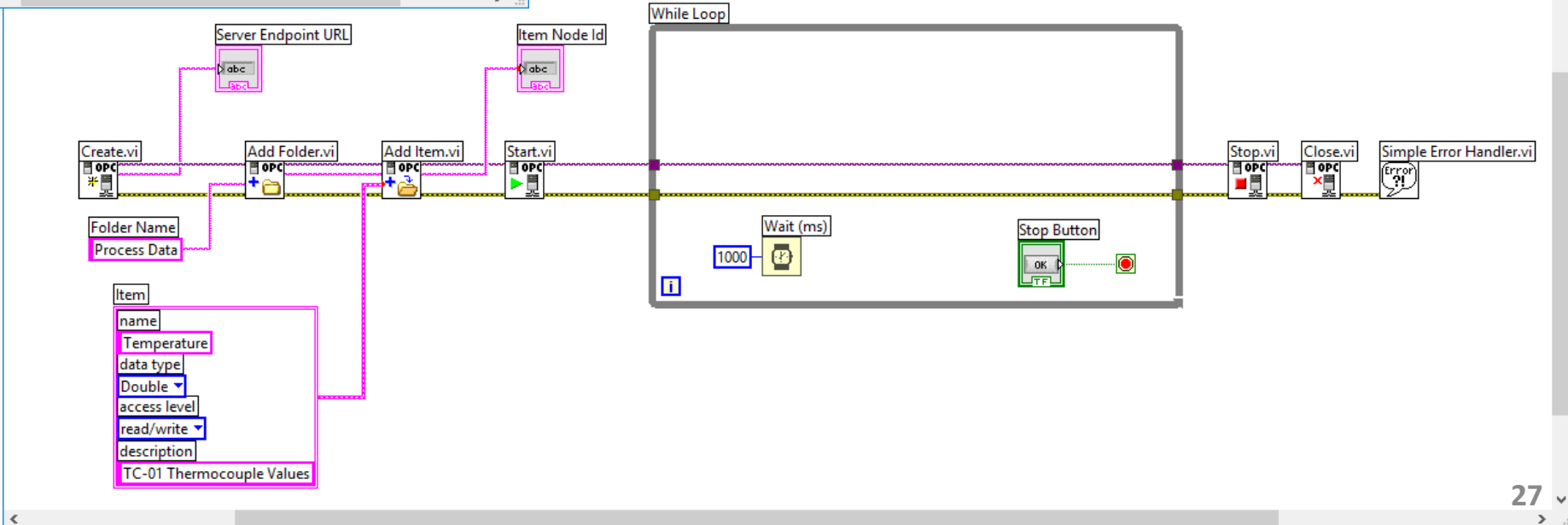
Server Endpoint URL
opc.tcp://XPS15HPH:49580

Item Node Id
ns=2;s=Process Data.Temperature

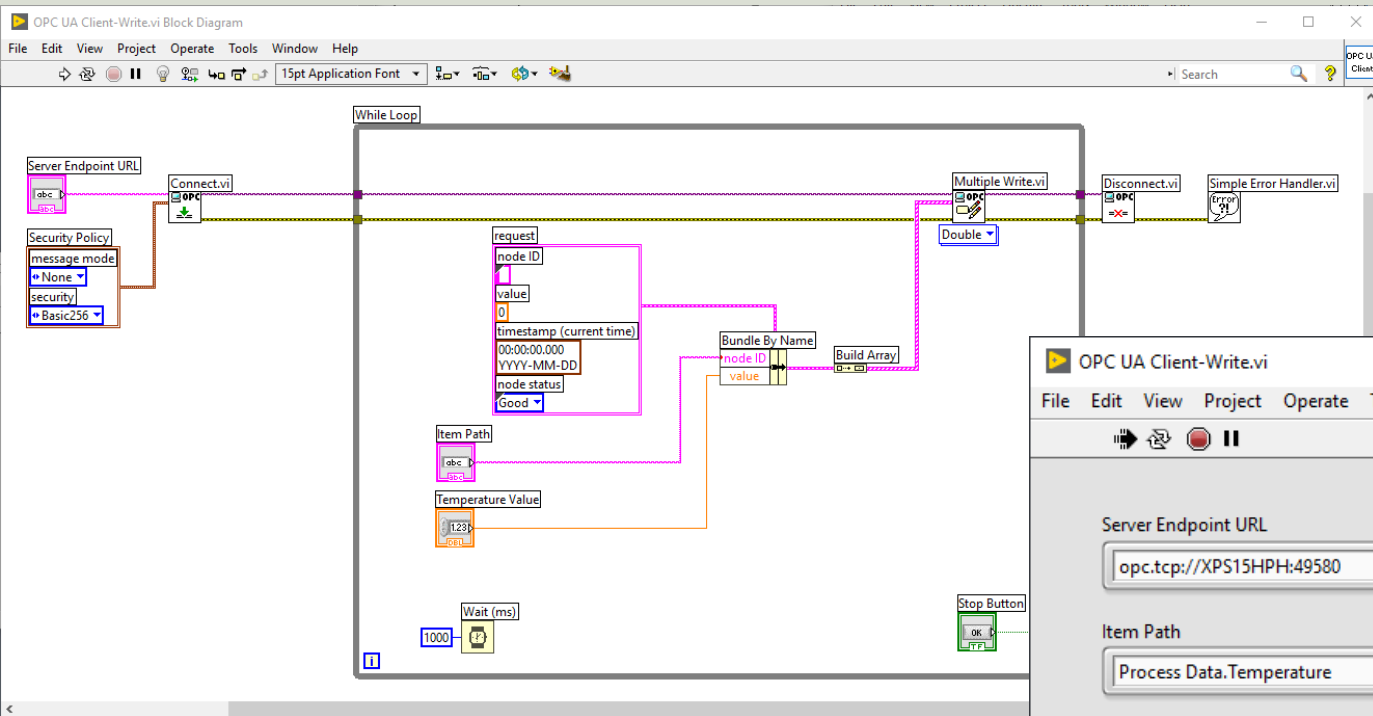
Stop

OPC UA Server

Search



OPC UA Client - Write Data



The front panel of the "OPC UA Client-Write.vi" program shows the following controls:

- Server Endpoint URL:** A text box containing the value "opc.tcp://XPS15HPH:49580".
- Item Path:** A text box containing the value "Process Data.Temperature".
- Temperature Value:** A numeric spinner control with the value "22".
- Stop Button:** A red square button labeled "Stop".

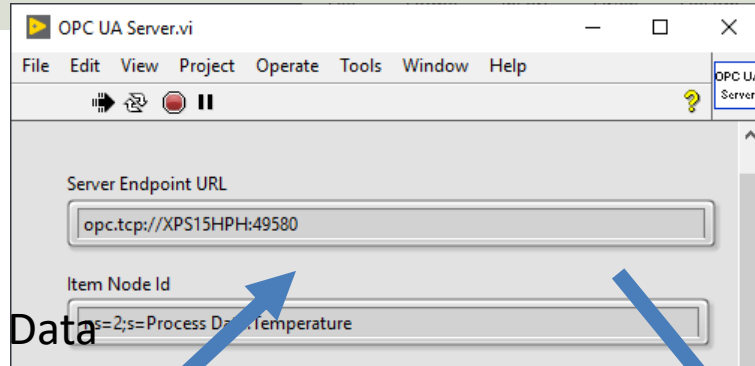
OPC UA Client - Read Data

The image displays the LabVIEW environment for an OPC UA Client application. The main window shows the block diagram of 'OPC UA Client-Read.vi'. The program starts with a 'Server Endpoint URL' control (value: 'abc') and a 'Security Policy' control (message mode: 'None', security: 'Basic256'). These are used in a 'Connect.vi' block. The connection is then used in a 'While Loop' which contains a 'Wait (ms)' block (value: 1000). Inside the loop, an 'Item Path' control (value: 'abc') is used in a 'Build Array' block. The array is then processed by a 'Multiple Read.vi' block, which is configured with a 'Double' data type. The output is processed by an 'Index Array' block (index: 0) and an 'Unbundle By Name' block (output: 'value'). The 'value' is displayed in a 'Temperature Value' control (value: 1.23). The loop ends with a 'Disconnect.vi' block and a 'Simple Error Handler.vi' block.

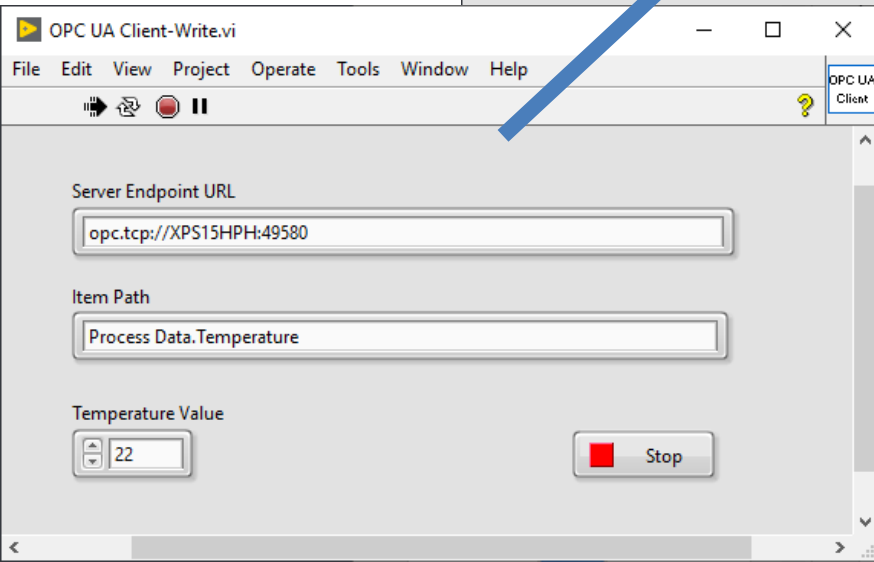
The front panel of the 'OPC UA Client-Read.vi' is shown in the bottom right corner. It features three input controls: 'Server Endpoint URL' (text field: 'opc.tcp://XPS15HPPH:49580'), 'Item Path' (text field: 'Process Data.Temperature'), and 'Temperature Value' (text field: '22'). A 'Stop' button is located at the bottom right of the front panel.

OPC UA in LabVIEW

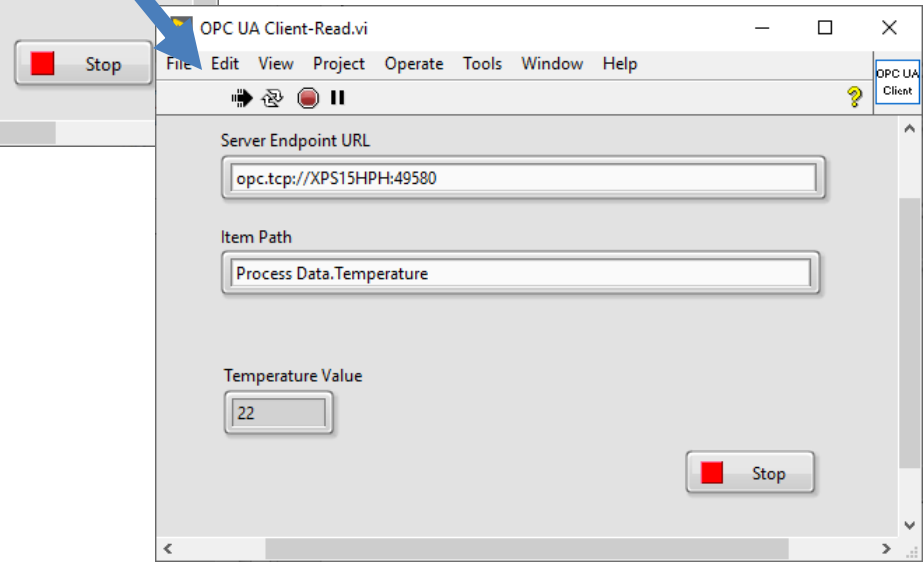
OPC UA Server



OPC UA Client – Write Data



OPC UA Client – Read Data





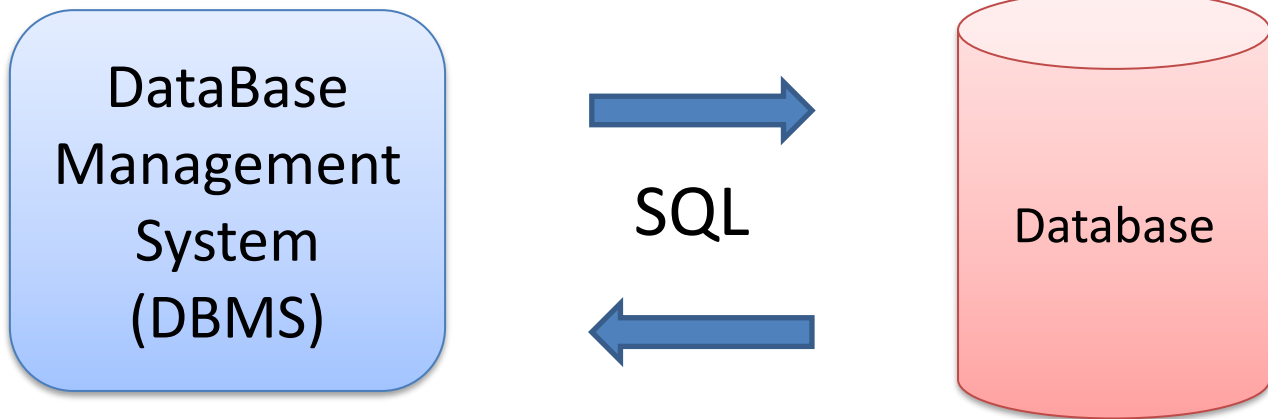
Database Systems

Database Systems

- A Database is a structured way to store lots of information
- The information is stored in different Tables inside the Database System
- We have many different Database Systems today, such as SQL Server, MySQL, PostgreSQL, Oracle, MongoDB, etc.
- We will use SQL Server from Microsoft
- Today “all” Software Systems saves and retrieves data from a Database System
- Examples: Facebook, Bank Systems, Process Control Systems, Web Shops, etc.

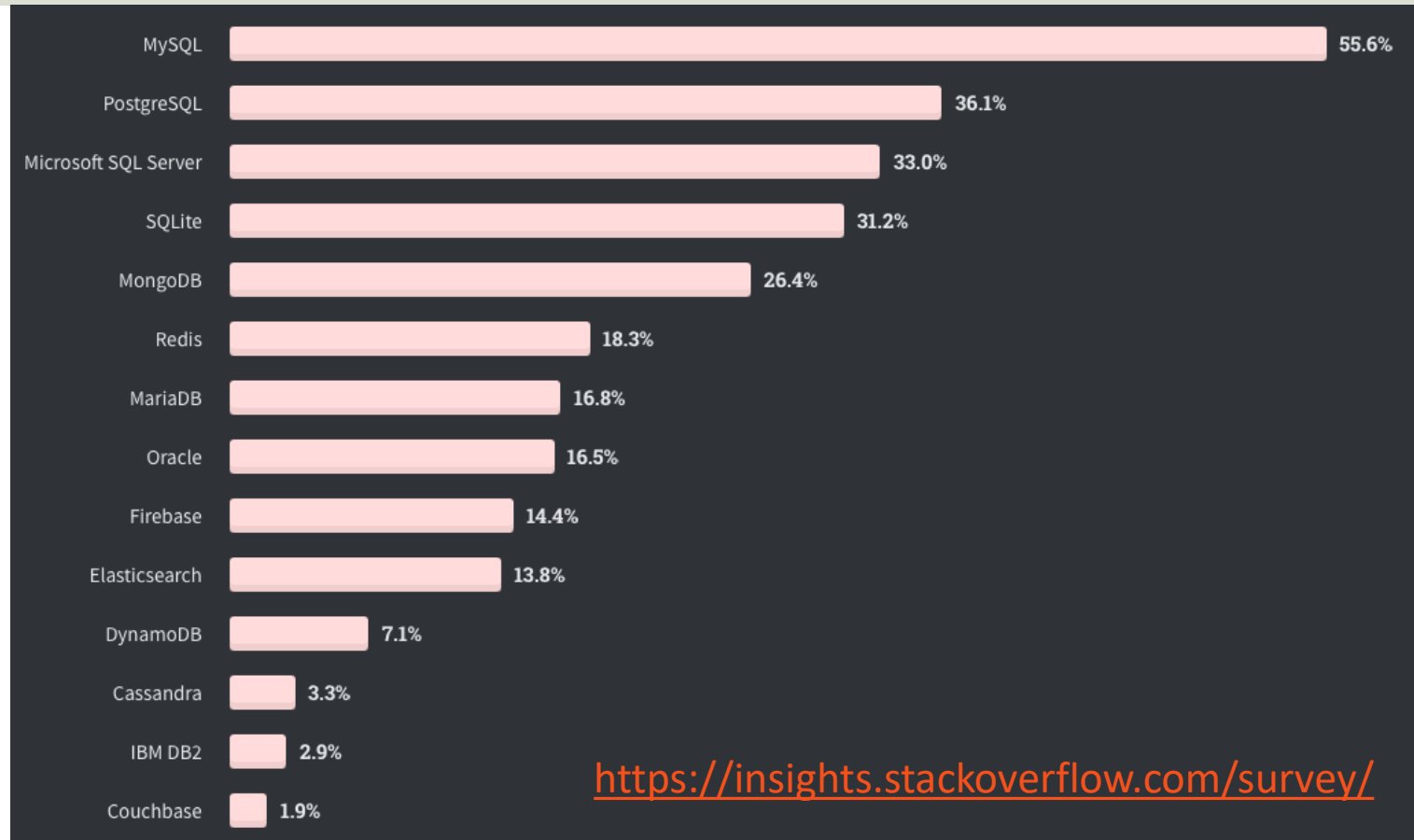
Database Systems

We communicate with the Database using a DataBase Management System (DBMS). We use the Structured Query Language (SQL) in order to communicate with the Database, i.e., Insert Data, Retrieve Data, Update Data and Delete Data from the Database.



SQL – Structured Query Language

Popular Database Systems



<https://insights.stackoverflow.com/survey/>

SQL Server

- SQL Server Express
 - Free version of SQL Server that has all we need for for the exercises in this module
- SQL Server Express consist of 2 parts (separate installation packages):
 - SQL Server Express
 - SQL Server Management Studio (SSMS) – This software can be used to create Databases, create Tables, Insert/Retrieve or Modify Data, etc.

SQL Server Management Studio

The screenshot displays the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left shows a tree view of the 'SCHOOL' database, with 'SCHOOL' and 'dbo.SCHOOL' highlighted. The 'Query Editor' in the center contains the SQL query: `select * from SCHOOL`. The 'Results' pane at the bottom shows the output of the query as a table with 4 rows. The 'Properties' pane on the right shows connection details for the current session.

3 New Query

1 Your SQL Server

2 Your Database

4 Write your Query here

5 The result from your Query

	SchoolId	SchoolName	Description	Address	Phone	PostCode	PostAddress
1	1	TUC	The best school	Telemark	NULL	NULL	NULL
2	2	MIT	OK School	USA	NULL	NULL	NULL
3	3	NTNU	The second best school	Trondheim	NULL	NULL	NULL
4	4	University of Oslo	The third best school	Oslo	NULL	NULL	NULL

Query executed successfully. | PC88235\DEVELOPMENT (10.50 ... | sa (52) | SCHOOL | 00:00:00 | 4 rows

Ready | Ln 1 | Col 21 | Ch 21 | INS

Structured Query Language

- Structured Query Language (SQL) is used to write, read and update data from the Database System
- You can use SQL inside the “SQL Server Management Studio” or inside your LabVIEW Application.

SQL Examples



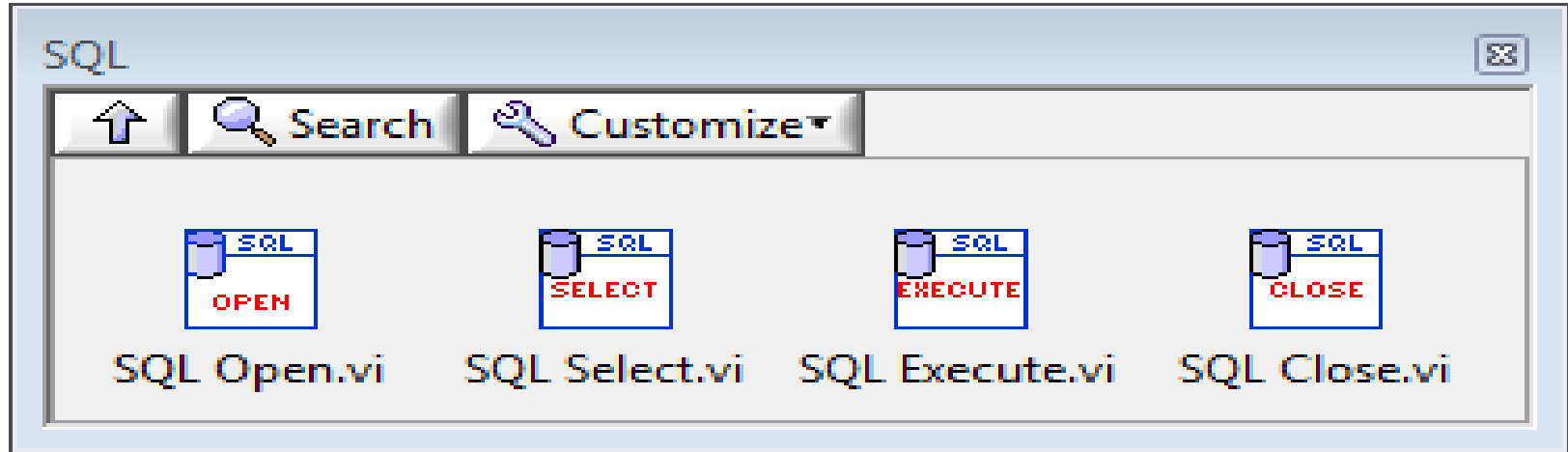
Query Examples:

- **insert** into STUDENT (Name , Number, SchoolId)
values ('John Smith', '100005', 1)
- **select** SchoolId, Name from SCHOOL
- **select** * from SCHOOL where SchoolId > 100
- **update** STUDENT set Name='John Wayne' **where** StudentId=2
- **delete** from STUDENT **where** SchoolId=3

We have 4 different Query Types: **INSERT**, **SELECT**, **UPDATE** and **DELETE**

LabVIEW SQL Toolkit

For Easy Database Communication with LabVIEW



© Hans-Petter Halvorsen

Download for free here:

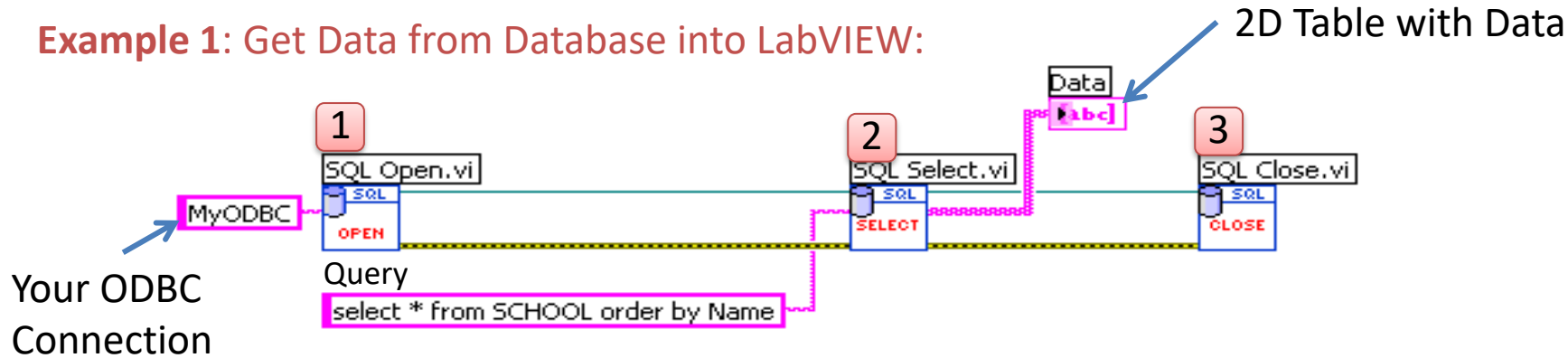
https://www.halvorsen.blog/documents/technology/database/database_labview.php

LabVIEW SQL Toolkit

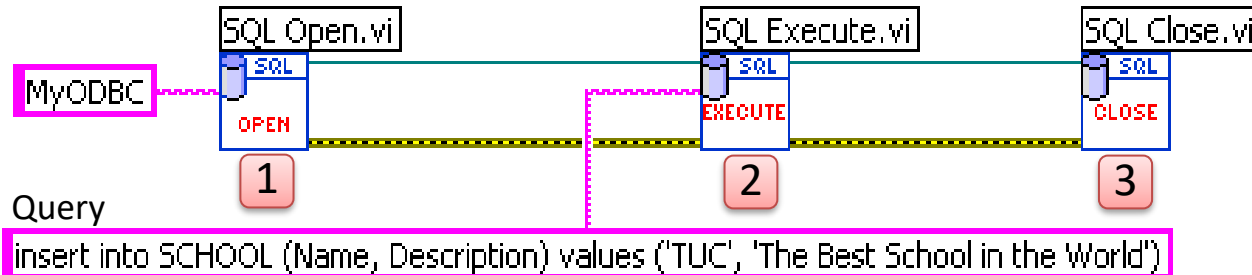


Easy Access to Database Systems from LabVIEW

Example 1: Get Data from Database into LabVIEW:



Example 2: Write Data to Database from LabVIEW:



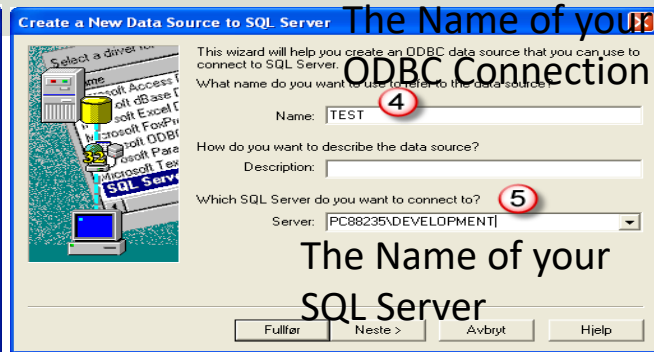
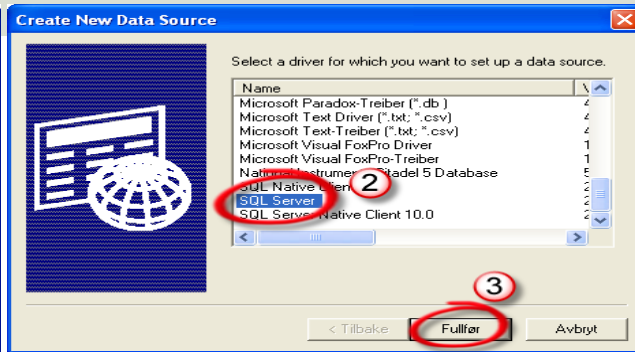
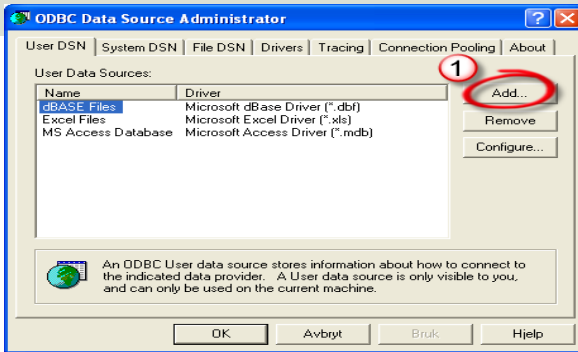
Connect to the Database

- Alt 1: Use ODBC
 - Setup your Database connection using a Wizard (“ODBC Data Source Administrator”)
- Alt 2: Use Connection String directly
 - Alt 2.1: Windows Authentication:
`Data Source=<dbserver>;Initial Catalog=<dbname>;Trusted_Connection=True`
 - Alt 2.2: SQL Server Authentication:
`Data Source=<dbserver>;Initial Catalog=<dbname>;Persist Security Info=True;User ID=sa;Password=<password>`

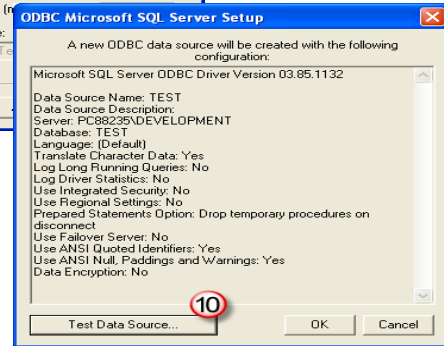
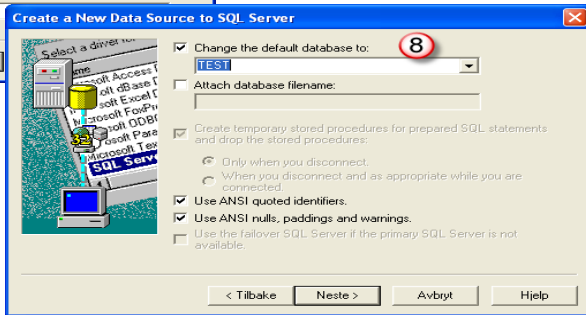
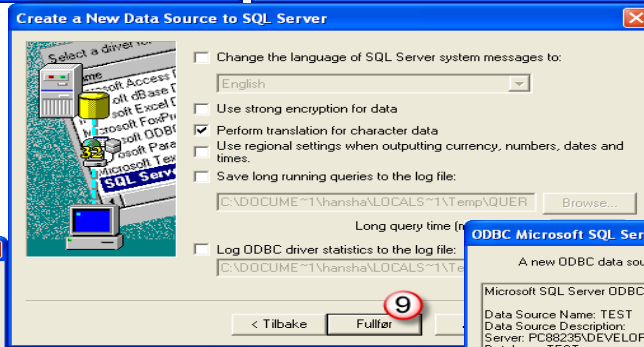
ODBC

The Name of your ODBC Connection

The Name of your SQL Server



Select the Database you are using



Use either Windows or SQL Server authentication (Windows is simplest to use!)

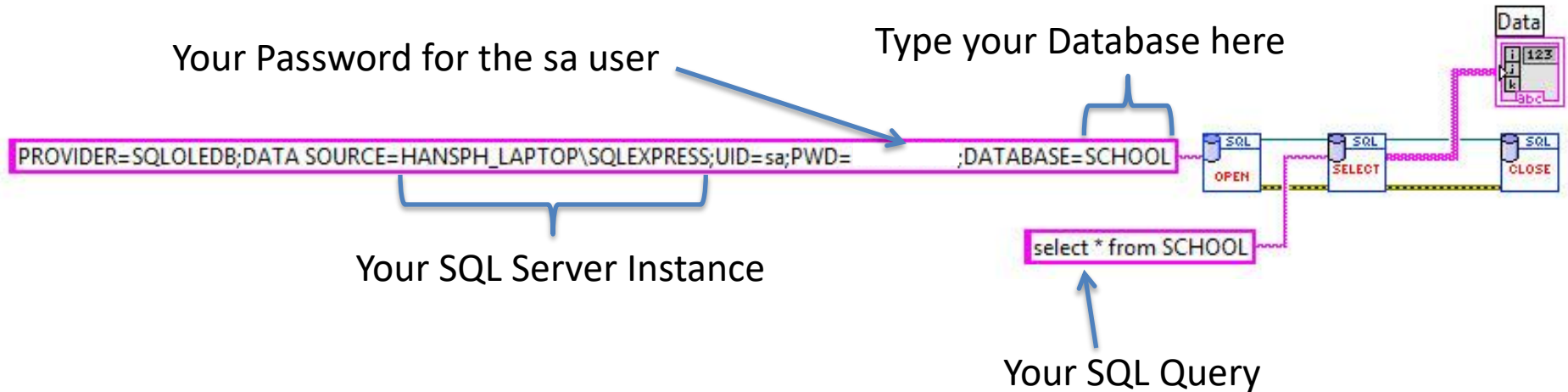
Test your connection to see if its works

Using Connection String



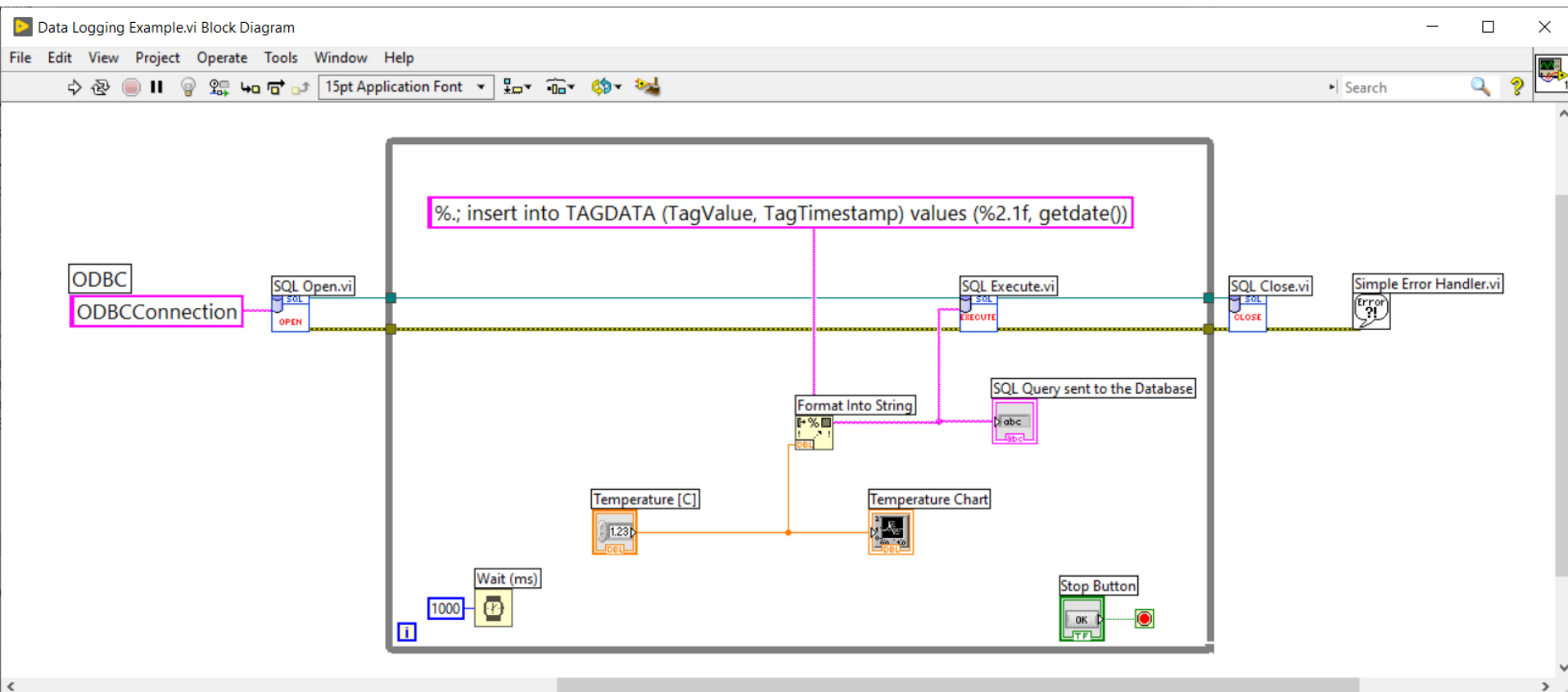
Easy Access to Database Systems from LabVIEW

Alternative Solution: Type in the **Connection String** for your Database

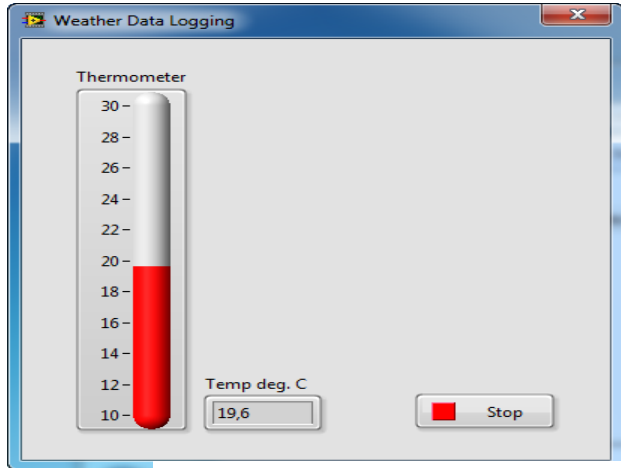


Note! When using this method, you don't need to create an ODBC Connection first!

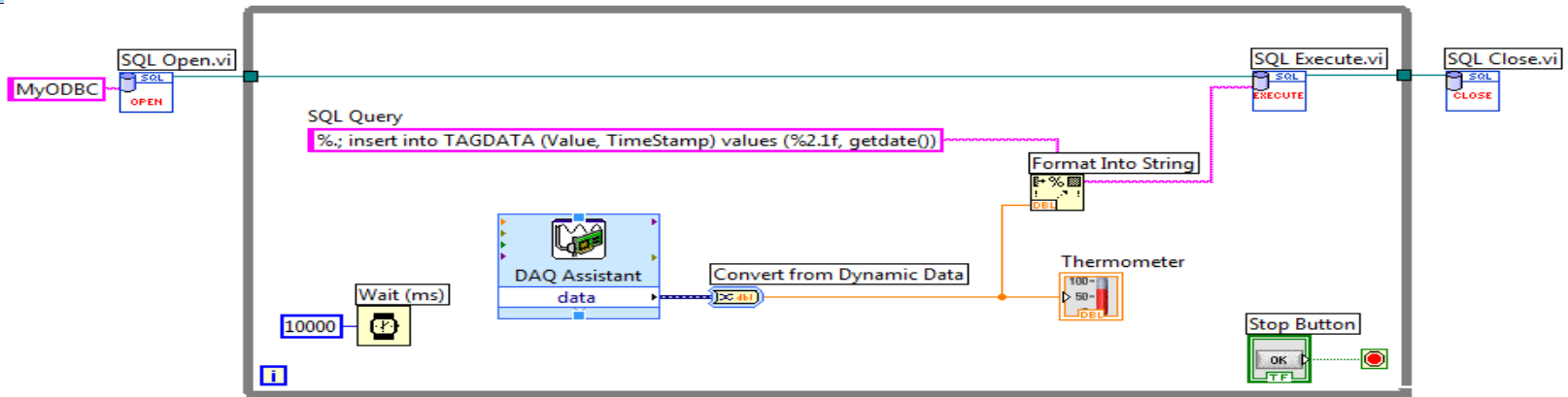
LabVIEW Example



LabVIEW Example



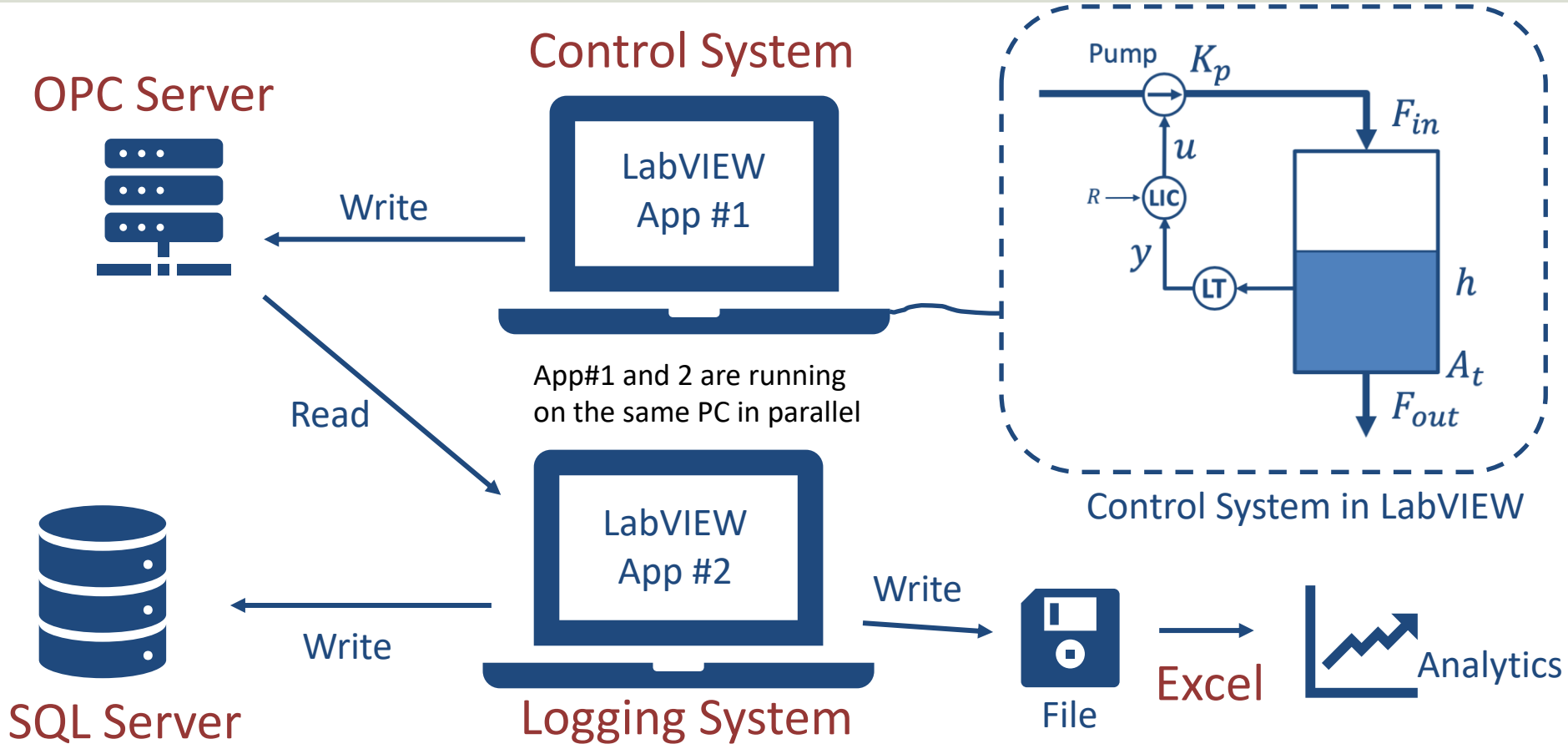
To practice you can use one (or both) of the Temperature Sensors from the previous Module, i.e., read Temperature Data using the USB-6008 DAQ and then Save the Temperature Data to the SQL Server Database





Delivery

System Sketch



System Requirements

- Create a **Datalogging and Monitoring System**
 - **Extend your Level Tank system with Datalogging and Monitoring Features**, i.e., send the process value to an OPC Server and then save it to a Text File and to a Database. Make sure that you have also updated your Level Tank system based on feedback given in previous Modules and include your latest LabVIEW skills.
- You should **create 2 LabVIEW Applications**:
 - **LabVIEW Application #1 - Control System**. Send Data to OPC Server. This means you should update your existing Level Tank application by sending the process data to the OPC Server (it is enough sending the process value, i.e., the level in the water tank). In addition, you should fix your program according to the comments/feedback given from the previous module.
 - **LabVIEW Application #2 – Datalogging System**. Retrieve Data from OPC Server and store the Data in a Text file and a SQL Server Database
- Make sure to create a proper **GUI**. You decide if you want to use OPC DA or OPC UA.
- Make **.exe** files of your LabVIEW Applications.
- You should open the Data from either the Database or the Text File in **MS Excel**. Make a simple plot and do some basic analytics (e.g., find the average, etc.).
- **The code should be well structured and intuitive**. It should contain basic LabVIEW features like While Loop, Case Structure, SubVIs, Arrays, Property Nodes and Clusters, etc. You should use the Project Explorer.
- Please follow the "**LabVIEW Programming Guidelines**"

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

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

