

Background information:

OPC DA - OPC data access

The process data interpreter

Basics

Automation technology most often involves hardware components from many different manufacturers combined in a system. It used to be that every manufacturer went his own way when it comes to passing process data up to the software level. This is true of both the physical communications path and the data format.

To get away from this process data confusion, the original OPC standard was introduced. Leading the way was the OPC Foundation in 1996 as a non-commercial entity. Members of the OPC Foundation are representatives from leading companies in the field of automation.

The goal was to create a globally accepted standard for communication in automation technology.

OPC DA - Classical OPC

OPC DA is not a network protocol as such. It is however an important industry standard and can, depending on the terminal device, also have points of contact with network communication.

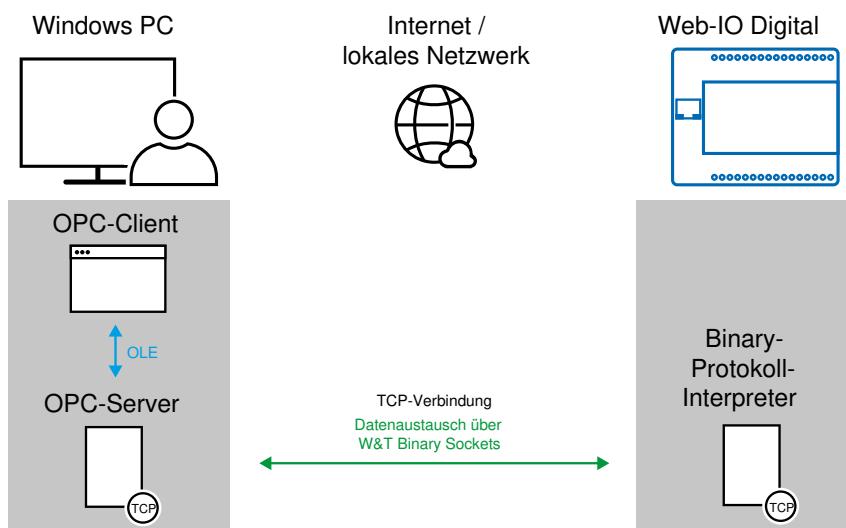
OPC stands for OLE for Process Control, where OLE is the abbreviation for Object Linking and Embedding. The basic idea of OLE is structured embedding of documents from other applications into one's own application, for example inserting an Excel document into a Word file.

Both OLE and OPC were created especially for PCs running Windows operating systems and only work on Microsoft operating systems.

OPC-supported applications do not communicate directly with the accessed terminal devices. Instead an OPC server is installed for the terminal device in question. The OPC server is a software process that runs the manufacturer-specific communication with the terminal device in the background - similar to a hardware driver on the actual PC. The process data made accessible in this way are prepared according to the OPC standard and passed to the application in a standardized form.

The component of the application which communicates with the OPC server is called the OPC client.

The following example shows how access to a Wiesemann & Theis Web-IO 12xDigital using an OPC server takes place:



Access via OPC

The original OPC interface made a distinction from four main tasks:

- Data Access: abbreviated DA, describes the exchange of real-time data via OPC.

- Alarm & Events: abbreviated AE, used for alarm and event handling.
- Historical Data Access: abbreviated HAD, makes stored, historical values and value trends accessible.
- Data Exchange: abbreviated DX, enables OPC servers to exchange data with each other.

OPC treats and process data as single data end points. A data end point can be a measurement value, a process status, a switching state and much more. The individual data end points are referred to as items. The items can be read or written depending on the type.

All items have an Item-ID, a unique address occurring only once in the OPC server. Each item has an undetermined number of properties, such as value, quality, time stamp etc.

The OPC server generally combines the items into groups. The result is a kind of hierarchy (OPC Server > OPC Group > OPC Item).

In order to provide the OPC client with easy access to all available items, many OPC servers permit OPC browsing by the OPC client. The OPC client can query all items in a kind of directory tree structure. Here as an example the structures of the items for a W&T Web-IO 2xDigital and a Web-Thermo-Hygrobarometer.

The screenshot shows a Windows application window titled 'Wiesemann-Theis.Network-IO'. On the left is a tree view of the OPC structure:

- Box1
 - E
 - 0
 - 1
 - A
 - 0
 - 1
 - N
 - 0
 - 1
 - Network
- Box2
 - T
 - 0
 - H
 - 0
 - P
 - 0
- Network

Communication between OPC client and OPC server

The OPC client can select a subset (or all) items from all those offered by the OPC server and for its own part combine them into one or more groups. These groups do not have to be identical to the groups formed by the OPC server. The selected items are then subscribed in groups by the OPC client. This means the OPC client does not have to constantly query the status of the items, but rather is automatically informed by the OPC server when one of the properties of an item changes. In this way the OPC server takes the burden off the OPC client and thereby off the application.

When does it make sense to work with OPC DA?

Whenever you need a flexible application that has to exchange data with the hardware from different OEMs without great effort, then OPC is the ideal solution.

In process control technology applications and process- and measurement data visualization OPC is particularly advantageous.

But even with all the advantages of OPC technology it must also be acknowledged that programming a universal OPC client application is a complex task which demands a high degree of programming competency.

So when the goal is to create a special application for a special terminal device of a manufacturer, it's worth considering whether it would be simpler to take the direct communication path provided by the OEM.

Basics of common industrial protocols

Web-IO - Automation using standard protocols

Integrating signals with Modbus-TCP, OPC UA/DA, REST or MQTT

Data formats and protocols

From traditional data exchange to IoT

Modbus-TCP

Open standard for industrial communication

REST - REpresentational State Transfer

Industrial communication based on standardized HTTP requests

MQTT - Message Query Telemetry Protocol

Data exchange without direct connection

OPC UA - OPC Unified Architecture

OPC support out of the box

Products for industrial applications with standard protocols



**Web-IO 4.0 Digital
2xIn, 2xOut**

Power via PoE also when needed



**Web-IO 4.0 Digital
12xIn, 12xOut**

12x outputs (6-30V),
12x inputs (8-30V)



Other Web-IOs

All W&T Web-IO Digital 24V



We are available to you in person:

Wiesemann & Theis GmbH

Porschestraße 12

42279 Wuppertal

Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)

Fax: +49 202/2680-265

info@wut.de

© Wiesemann & Theis GmbH, subject to mistakes and changes: Since we can make mistakes, none of our statements should be applied without verification. Please let us know of any errors or misunderstandings you find so that we can become aware of and eliminate them.

[Data Privacy](#)