

# lamaPLC Communication: OPC

**Open Platform Communications (OPC)** is a series of standards and specifications for industrial telecommunication. They are based on Object Linking and Embedding (OLE) for process control. An industrial automation task force developed the original standard in 1996 under the name **OLE for Process Control**. OPC specifies the communication of real-time plant data between control devices from different manufacturers.

After the initial release in 1996, the OPC Foundation was created to maintain the standards. Since OPC has been adopted beyond the field of process control, the OPC Foundation changed its name to Open Platform Communications in 2011. The name change reflects the applications of OPC technology for applications in building automation, discrete manufacturing, process control and others. OPC has also grown beyond its original OLE implementation to include other data transportation technologies including Microsoft Corporation's **.NET** Framework, **XML**, and even the OPC Foundation's binary-encoded **TCP** format.

OPC was designed to provide a common bridge for Windows-based software applications and process control hardware. Standards define consistent methods of accessing field data from plant floor devices. This method remains the same regardless of the type and source of data. An OPC Server for one hardware device provides the same methods for an OPC client to access its data as any other OPC Server for any hardware device. The aim was to reduce the amount of duplicated effort required from hardware manufacturers and their software partners, and from the *supervisory control and data acquisition (SCADA)* and other *human-machine interface (HMI)* producers in order to interface the two. Once a hardware manufacturer had developed their OPC Server for the new hardware device, their work was done with regards to allowing any 'top end' to access their device, and once the SCADA producer had developed their OPC client, it allowed access to any hardware with an OPC compliant server.

## OPC Foundation

The OPC Foundation is the organization behind the standard and with 678 members it has a very broad base. Its members include global players in the automation industry. For example: Siemens, Honeywell, Microsoft, Beckhoff, SAP, Yokogawa, ABB, Rockwell, Schneider Electric, Wago, Iconics. All members of the Foundation can be found in the OPC Foundation member list. The association was founded in 1994 and released the first version of OPC in 1996. Since then, it has been working very successfully and actively on the further development and dissemination of the OPC standard.

## OPC UA

**OPC Unified Architecture (OPC UA)** is a machine-to-machine communication protocol used for industrial automation and developed by the OPC Foundation. The OPC UA platform is an platform-independent service-oriented architecture that integrates individual OPC Classic specifications into an extensible framework. OPC UA simplifies industrial connectivity so you can integrate all your devices, automation systems and software applications using a secure and platform-independent standard.



The UA in OPC UA stands for “Unified Architecture” and refers to the latest specification of the standard. It differs from its predecessor in that it is platform-independent, moving away from COM/DCOM to purely binary TCP/IP or alternatively SOAP. In addition to many other improvements, OPC UA also supports a semantic data description.

## OPC UA Server

The OPC Server is the basis of OPC communication. It is a software that implements the OPC standard and thus provides the standardized OPC interfaces to the outside world. Inside, the proprietary communication protocol for controlling the manufacturer is implemented. OPC Servers are provided by different parties.

The basic idea of OPC is that the manufacturer of the hardware provides an OPC Server for his system and therefore allows standardized access. The OPC Server from the manufacturer can be supplied as stand-alone software or as an embedded OPC Server on the device or the machine controller.

## OPC UA Client

The OPC Client is the logical counterpart to the OPC Server. The OPC Server can be connected to the OPC Client and read out the data provided by the Server. Since the OPC Servers implement the predefined interfaces of the OPC standard, each client can access any OPC Server and exchange data with the server in the same way.

Typical OPC clients are applications that depend on data exchange with industrial systems. What happens to the data in the client is very application-specific. Common applications are visualization and SCADA systems (WinCC, InTouch, FAS inMOVE) or MES systems. The OPC Router with its OPC UA Client Plug-in is also a client with a gateway function.

## OPC UA Pub/Sub

The communication participants in OPC UA Pub/Sub are divided into Publisher and Subscriber. Devices and software can communicate with each other via a Broker without having to rely on the 1-to-1 relations of a client/server communication. Therefore, communication within the system can be accelerated and processor capacity can be saved.

## OPC UA Pub/Sub over MQTT

The principle of Pub/Sub is not only used by the OPC UA Pub/Sub Plug-in, but also by the common message protocol MQTT. Both Plug-ins enable the OPC Router to be operated as a Publisher and Subscriber. However, the OPC UA Pub/Sub Plug-in differs from the MQTT Plug-in in that the structure of OPC UA data is already available in the OPC UA Pub/Sub Plug-in. That allows the OPC Router to publish OPC UA data to other devices or systems for further processing, or to subscribe OPC UA data from others. Implementing OPC UA data via MQTT would be much more difficult, since most of the message content in MQTT is described in XML or JSON. ,

## OPC XML

The OPC XML specification was the OPC Foundation's first attempt to achieve platform independence and to break the link to Microsoft technology. The XML format, which was emerging at that time, was used in connection with web technologies. In 2003, the OPC XML specification was released.

## DCOM

The Microsoft technology DCOM is the network version of the COM technology. COM stands for Component Object Model and is the object-oriented technology in the background of the Microsoft operating system. It is used to let different applications work together intelligently. Since it was the leading technology at the time of the development of the OPC standard, it was used as a basis. The OPC Classic Server is a COM component and is connected as such by the client. If an OPC Classic Server is connected via the network, the Distributed COM (DCOM) protocol must be used, which is deeply anchored in the Windows operating system. DCOM has a complex authentication logic and uses multiple and dynamic TCP/IP connections, making it uncontrollable for a firewall and categorically blocked. In practice, therefore, DCOM connections were usually not established.

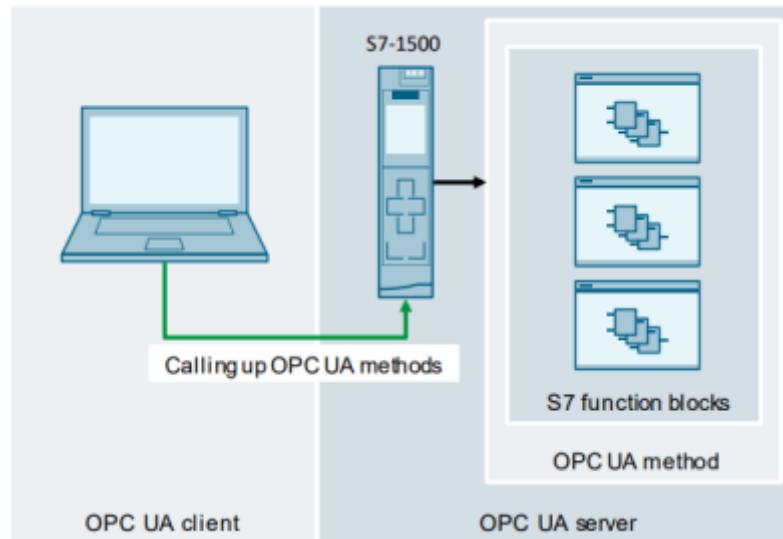
## UaExpert

The UaExpert is a full-featured OPC UA Client demonstrating the capabilities of our C++ OPC UA Client SDK/Toolkit and OPC UA technology in general. The UaExpert is designed as a general purpose test client supporting OPC UA features like DataAccess, Alarms & Conditions, Historical Access and calling of UA Methods.

Download [from here](#).

## Simatic OPC UA

For standardized, communication of an S7 control system that applies to all platforms and manufacturers, Siemens offers you the communication protocol OPC UA. As of firmware version V2.5 and TIA Portal V15.1, the SIMATIC S7-1500 control system can also offer OPC UA methods for the clients via its integrated OPC UA server. With this addition to the functions, you not only have the option of reading or writing OPC UA variables of the control system, you can now also start complex functional sequences via OPC UA. This enables almost complete M2M communication via OPC UA, for the networking of plants or for the control of plants from an ERP / MES level.



## Allowed data types as interface parameters

Approved data types S7 - OPC UA	
Simatic type	OPC UA type
BOOL	Boolean
SINT	SByte
INT	Int16
DINT	Int32
LINT	Int64
USINT	Byte
UINT	UInt16
UDINT	UInt32
ULINT	UInt64
REAL	Float
LREAL	Double
LDT	DateTime
WSTRING	String

You can find [a description of Simatic data types here](#).

## Sources

OPC Router: <https://www.opc-router.com/what-is-opc-ua/>

Siemens:

<https://support.industry.siemens.com/cs/document/109756885/opc-ua-methods-for-the-simatic-s7-1500-opc-ua-server>

## OPC topics on lamaPLC

Page	Date	Tags

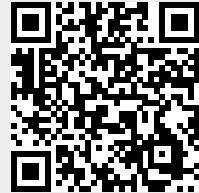
• **lamaPLC Communication:** 2024/11/15 20:33 [communication](#), [opc](#), [scada](#), [ole](#), [net](#), [xml](#), [tcp](#), [hmi](#), [OPC](#) [server](#), [opc ua](#), [opc ua client](#), [mqtt](#), [json](#), [dcom](#), [simatic](#), [erp](#)

[communication](#), [OPC](#), [SCADA](#), [OLE](#), [.NET](#), [XML](#), [TCP](#), [HMI](#), [Server](#), [OPC UA](#), [OPC UA Client](#), [MQTT](#), [XML](#), [JSON](#), [DCOM](#), [Simatic](#), [ERP](#)

This page has been accessed for: Today: 2, Until now: 41

From:

<http://lamaplc.com/> - **lamaPLC**



Permanent link:

[http://lamaplc.com/doku.php?id=com:basic\\_opc](http://lamaplc.com/doku.php?id=com:basic_opc)

Last update: **2024/11/15 20:33**