



CODESYS OPC UA PubSub SL

The library *OPC UA PubSub SL* allows for the exchange of messages via the *Pub/Sub* protocol defined by the *OPC UA Foundation*.

Product description

Communication via the *OPC UA Pub/Sub* protocol provides, in addition to the *Client/Server* communication, also a possibility to exchange data between the subscribers of a network in compliance with the rules of the *OPC UA Foundation*. The structure of the data (*DataSet*) can be freely defined and is agreed in advance between the sender and the receiver. This eliminates the need to transport additional (meta)data. The data transfer takes place with the help of the *OPC UA PubSub SL* library via *UDP/IP* according to the rules defined for *UADP*. A *Publisher* publishes its data to an unknown number of *Subscribers*. So the sender and receiver do not know each other. That is why the number of receivers also does not have any repercussions for the sender. A reply to the sender as to whether its messages have reached the receivers cannot be sent via the protocol due to the nature of the protocol. If necessary, such a reply has to be managed in an application-specific way. The data is transferred in a binary format according to the rules of the *OPC UA Foundation*. The *OPC UA PubSub SL* library takes over the conversion of IEC data types into the corresponding *OPC UA* data types and back. The implementation thereby follows the profiles below:

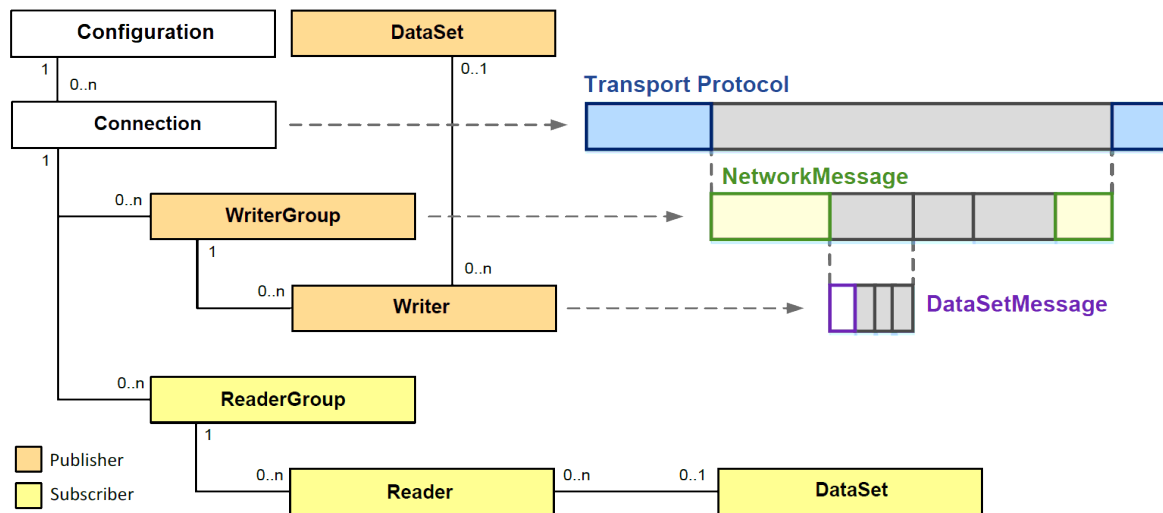
- *Publisher: PubSub Publisher UADP Periodic Fixed Settings*
- *Subscriber: PubSub Subscriber UADP Periodic Fixed Settings*

The message length is limited to 1500 *bytes* (MTU) (*Chunked NetworkMessages not supported*). As long as the rules for a *Time Sensitive Network* is still not available, hard real-time conditions cannot be complied with. However, the implementation of the *OPC UA PubSub SL* library attempts to keep the *jitter* as low as possible.

The following function blocks are included in the library:

- *DataSet*: Function block for the definition of a *DataSetMessage*
- *Configuration*: Function block for the management of shared resources (*Connection, Group, ...*)
- *RootDiagnostics*: Function block for superordinate diagnostic data
- *Connection*: Function block for the management of the connection to *Publisher* and *Subscriber*
- *ConnectionDiagnostics*: Function block for diagnostic data of the connection (*Connection*)
- *ReaderGroup*: Function block for the management of *Reader* function blocks (creation of a *NetworkMessage* from *DataSetMessages*)

- ReaderGroupDiagnostics: Function block for diagnostic data of a ReaderGroup
- Reader: Function block for the management of a DataSet function block (Subscriber)
- ReaderDiagnostics: Function block for diagnostic data of a Reader function block
- WriterGroup: Function block for the management of writer function blocks (creation of a NetworkMessage from DataSetMessages)
- WriterGroupDiagnostics: Function block for diagnostic data of a WriterGroup
- Writer: Function block for the management of a DataSet function block (Publisher)
- WriterDiagnostics: Function block for diagnostic data of a writer function block



Supported functions

- Publish/Subscribe of messages according to *OPC 10000-14: OPC Unified Architecture Part 14: PubSub Release 1.04*
- Maximum size of a *NetworkMessage*: 1500
- The current packet size and *payload* can be configured by means of the configuration of the respective *ReaderGroup/WriterGroup* and *DataSet* function blocks.
- Support of a background task for the respective *Connection* function block
- The following profiles are supported:
 - *Publisher: PubSub Publisher UADP Periodic Fixed Settings*
 - *Subscriber: PubSub Subscriber UADP Periodic Fixed Settings*

Using the sample project

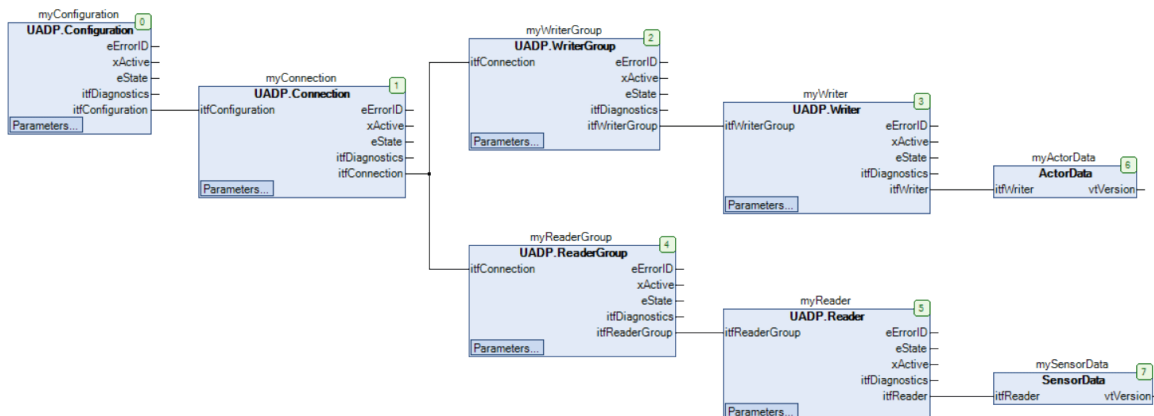
During installation, the sample project *OPC UA PubSub SL Example.project* is installed in the selected target directory. The application *Device_1* demonstrates how to read a *NetworkMessage*. The application *Device_2* shows how messages can be sent by means of the contained function blocks.

Preparatory steps

1. Hardware
 - a. You need two CODESYS PLCs which are connected over a network that supports UDP Multicast.

- b. Adapt the devices in the project to the configuration by updating Device_1 and Device_2 to the used hardware.
2. Customizations in the project
 - a. Set the IP addresses in the GlobalIP GVL to the hardware which you have configured. You find this GVL in the POU area.
 - b. Set the Multicast address to a free address in your network. Note: If you are not in a private network, request that your network administrator assign free address for you.
 - c. Set the port. The default port 4840 is the recommended port for OPC UA PubSub.
3. Log in to both controllers and start the applications.
4. Start the sender (Device_2: PLC_PRG.xEnable := TRUE) and receiver (Device_1: Communication_PRG.xEnable := TRUE).
5. Now on the receiver side you should see how the values of the sine changes. You can manually change the other values in the sender and the see the changes in the receiver.
6. Suggestion for extending the example (adding a variable to the sent DataSet)
 - a. Extend SensorDataSet._aIndex by one entry. Select any data type.
 - b. In SensorDataSet.Init, adapt the version of the DataSet by updating the date entry.
 - c. Create a variable of the selected data type in PLC_PRG of the sender and assign this in txSensorDataSet.PrepareValues.
 - d. Repeat this step on the receiver side in rxSensorDataSet.PrepareValues. Create a new variable of the same type beforehand here as well.
 - e. After a new download, this variable should be transmitted as well.

Example of a typical configuration:



General information

Supplier:

CODESYS GmbH
 Memminger Strasse 151
 87439 Kempten
 Germany

Support:

Technical support is not included with this product. To receive technical support, please purchase a CODESYS Support Ticket.

<https://support.codesys.com>

Item:

CODESYS OPC UA PubSub SL

Item number:

2311000001

Sales / Source of supply:

CODESYS Store
<https://store.codesys.com>

Included in delivery:

CODESYS Package with library, sample project, product data sheet (en, de), and CHM help

System requirements and restrictions

Development system	CODESYS Development System V3.5.16.30 or higher
Runtime	CODESYS Control V3.5.14.0 or higher
Supported platforms and devices	Note: Use the "Device Reader" project for locating the functions supported by the PLC. The Device Reader is available in the CODESYS Store free of charge.
Additional requirements	Depending on the capabilities of the respective runtime system, the messages can be sent via <i>Unicast</i> , <i>Multicast</i> , or <i>Broadcast</i> .
Restrictions	<ul style="list-style-type: none"> • The maximum length of a <i>Networkmessage</i> is 1500 bytes. (<i>Chunked NetworkMessages not supported</i>) • The transmission time point is determined via the task configuration. (The parameter <i>udiPublishingInterval</i> does not have any effect.)
Licensing	



Single Device License: The license can be used on the target device/PLC on which the CODESYS Runtime System is installed.

Licenses are activated on a software-based license container (soft container), which is permanently connected to the controller. Alternatively the license can be stored on a CODESYS Key (USB-Dongle). By replugging the CODESYS Key, the license can be used on any other controller

Note: Without a license, the software runs for 30 minutes demo mode. A manual restart is required after that.

Required accessories

-

Note: Not all CODESYS features are available in all territories. For more information on geographic restrictions, please contact sales@codesys.com.

Note: Technical specifications are subject to change. Errors and omissions excepted. The content of the current online version of this document applies.