Platform MX6 / PI
Software option S118
CODESYS OPC UA PubSub

1 Identification

Identification	
Option identifier	S118
Order number	S-05000319-0000
Short name	CODESYS OPC UA PubSub
Short description	With the help of this software option it is possible to exchange messages via the pub/sub protocol defined by OPC UA Foundation defined Pub/Sub protocol.
Revision identifier document	V1.0

2 System requirements and restrictions

System requirements and restrictions	
Supported platforms or devices	Berghof SPS devices of the MX6 and PI platform (e.g.: MC, CC, DC). Further information regarding availability and compatibility can be found in the product catalog in the options section.
Firmware	MX6-SPS from version 1.27.1, CODESYS from 3.5 SP18 Patch 4 PI-SPS from version 1.3.2, CODESYS from 3.5 SP18 Patch 4
Further requirements	— IP network interface Network access — Publish/Subscribe of messages according to OPC 10000-14: OPC Unified Architecture Part 14: PubSub Release 1.04 — Depending on the equipment of the respective runtime system, the messages can be sent via unicast, multicast or broadcast.
Restrictions	— Maximum size of a NetworkMessage: 1500 (Chunked NetworkMessages not supported) — The current packet size and size of the payload can be configured via the configuration of the respective ReaderGroup/WriterGroup and DataSet blocks. — The transmission time point is determined via the task configuration. (The parameter udiPublishingInterval has no effect).

3 Product description

This software option enables the license the CODESYS library OPC UA PubSub for the device.

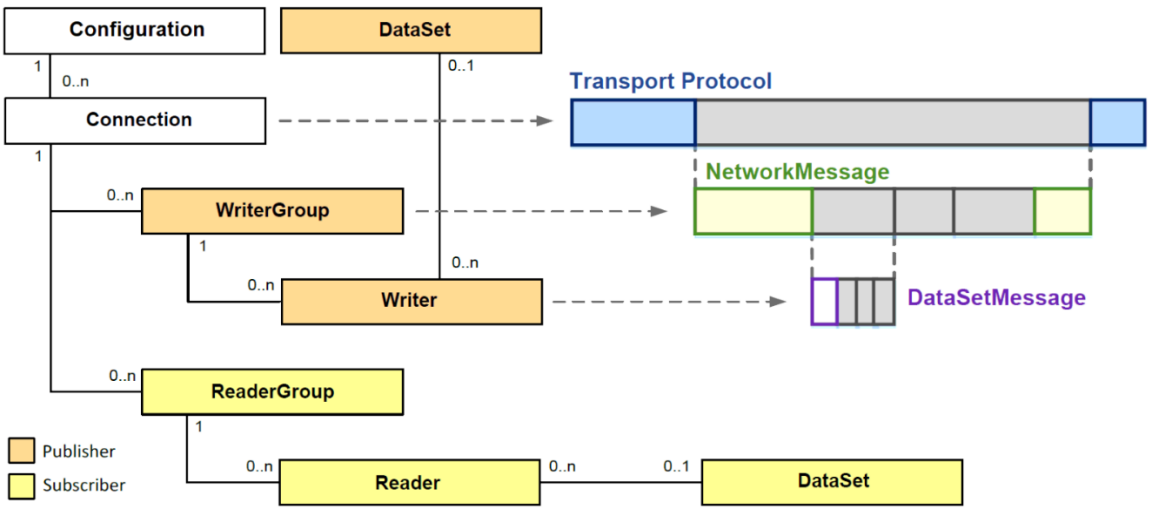
Communication via the OPC UA Pub/Sub protocol provides, in addition to the Client/Server communication, also a possibility to exchange data between the subscribers of a network in compliance with the rules of the OPC UA Foundation. The structure of the data (DataSet) can be freely defined and is agreed 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 realtime 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 (create 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



4 Technical data

Technical data	
Tasking	Support of a background task for the respective connection module
Supported profile	Publisher: PubSub Publisher UADP Periodic Fixed Settings Subscriber: PubSub Subscriber UADP Periodic Fixed Settings

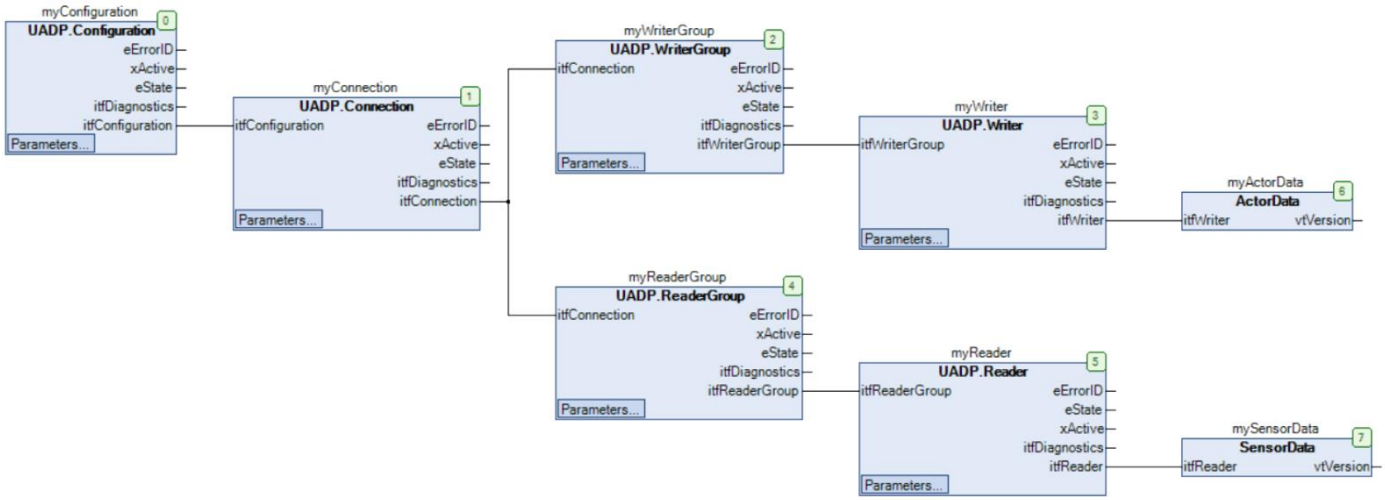
5 Quick Start Guide

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.

5.1 Using the sample project

1. Hardware
   - You need two Berghof SPS, which are connected via a network.
   - 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
   - Set the IP addresses in the GlobalIP GVL to the hardware which you have configured. You find this GVL in the POU area.
   - 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.
   - 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)
   - Extend SensorDataSet._aIndex by one entry. Select any data type
   - In SensorDataSet.Init, adapt the version of the DataSet by updating the date entry
   - Create a variable of the selected data type in PLC_PRG of the sender and assign this in txSensorDataSet.PrepareValues.
   - Repeat this step on the receiver side in rxSensorDataSet.PrepareValues. Create a new variable of the same type beforehand here as well.
   - After a new download, this variable should be transmitted as well.

Example of a typical configuration:



You can reach your contact persons at:

Sales Tean | T +49.7121.894-131 | controls@berghof.com

Berghof Automation GmbH | Arbachtalstraße 26 | 72800 Eningen | www.berghof-automation.com