AC500 PLC

IEC 61850 Library

V3 Example Project Description

AC500 PLC

IEC 61850 Library

V3 Example Project Description

NOTICE

This document contains information about one or more ABB products and may include a description of or a reference to one or more standards that may be generally relevant to the ABB products. The presence of any such description of a standard or reference to a standard is not a representation that all of the ABB products referenced in this document support all of the features of the described or referenced standard. In order to determine the specific features supported by a particular ABB product, the reader should consult the product specifications for the particular ABB product.

ABB may have one or more patents or pending patent applications protecting the intellectual property in the ABB products described in this document.

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

Products described or referenced in this document are designed to be connected and to communicate information and data through network interfaces, which should be connected to a secure network. It is the sole responsibility of the system/product owner to provide and continuously ensure a secure connection between the product and the system network and/or any other networks that may be connected.

The system/product owners must establish and maintain appropriate measures, including, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of antivirus programs, and so on, to protect these products, the network, its system, and interfaces against security breaches, unauthorized access, interference, intrusion, leakage, and/or theft of data or information.

ABB performs functionality testing on the products and updates that we release. However, system/product owners are ultimately responsible for ensuring that any product updates or other major system updates (to include but not limited to code changes, configuration file changes, third-party software updates or patches, hardware change out, and so on) are compatible with the security measures implemented. The system/ product owners must verify that the system and associated products function as expected in the environment in which they are deployed.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license. This product meets the requirements specified in EMC Directive 2014/30/EU and in Low Voltage Directive 2014/35/EU.

A. For customers domiciled outside Germany /

Für Kunden mit Sitz außerhalb Deutschlands

„Warranty, Liability:

The user shall be solely responsible for the use of this products described within this file. ABB shall be under no warranty whatsoever. ABB's liability in connection with application of the products or examples provided or the files included within this products, irrespective of the legal ground, shall be excluded. The exclusion of liability shall not apply in the case of intention or gross negligence. The present declaration shall be governed by and construed in accordance with the laws of Switzerland under exclusion of its conflict of laws rules and of the Vienna Convention on the International Sale of Goods (CISG)."

„Gewährleistung und Haftung:

Der Nutzer ist allein für die Verwendung des in diesem Dokument beschriebenen Produkte und beschriebenen Anwendungsbeispiele verantwortlich.

ABB unterliegt keiner Gewährleistung. Die Haftung von ABB im Zusammenhang mit diesem Anwendungsbeispiel oder den in dieser Datei enthaltenen Dateien - gleich aus welchem Rechtsgrund - ist ausgeschlossen. Dieser Ausschluß gilt nicht im Falle von Vorsatz oder grober Fahrlässigkeit. Diese Erklärung unterliegt Schweizer Recht unter Ausschluß der Verweisungsnormen und des UN-Kaufrechts (CISG)."

B. Nur für Kunden mit Sitz in Deutschland

„Gewährleistung und Haftung:

Die in diesem dokument beschriebenen Anwendungsbeispiele oder enthaltenen Dateien beschreiben eine mögliche Anwendung der AC500 bzw. zeigen eine mögliche Einsatzart. Sie stellen nur Beispiele für Programmierungen dar, sind aber keine fertigen Lösungen. Eine Gewähr kann nicht übernommen werden.

Der Nutzer ist für die ordnungsgemäße, insbesondere vollständige und fehlerfreie Programmierung der Steuerungen selbst verantwortlich. Im Falle der teilweisen oder ganzen Übernahme der Programmierbeispiele können gegen ABB keine Ansprüche geltend gemacht werden.

Die Haftung von ABB, gleich aus welchem Rechtsgrund, im Zusammenhang mit den Anwendungsbeispielen oder den in dieser Datei enthaltenen Beschreibung wird ausgeschlossen. Der Haftungsausschluß gilt jedoch nicht in Fällen des Vorsatzes, der groben Fahrlässigkeit, bei Ansprüchen nach dem Produkthaftungsgesetz, im Falle der Verletzung des Lebens, des Körpers oder der Gesundheit oder bei schuldhafter Verletzung einer wesentlichen Vertragspflicht. Im Falle der Verletzung einer wesentlichen Vertragspflicht ist die Haftung jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht zugleich ein anderer der in Satz 2 dieses Unterabsatzes erwähnten Fälle gegeben ist. Eine Änderung der Beweislast zum Nachteil des Nutzers ist hiermit nicht verbunden.

Es gilt materielles deutsches Recht unter Ausschluß des UN-Kaufrechts."

TRADEMARKS

Release: August 2017

Document number: 3ADR010262, 1, en_US

TABLE OF CONTENTS

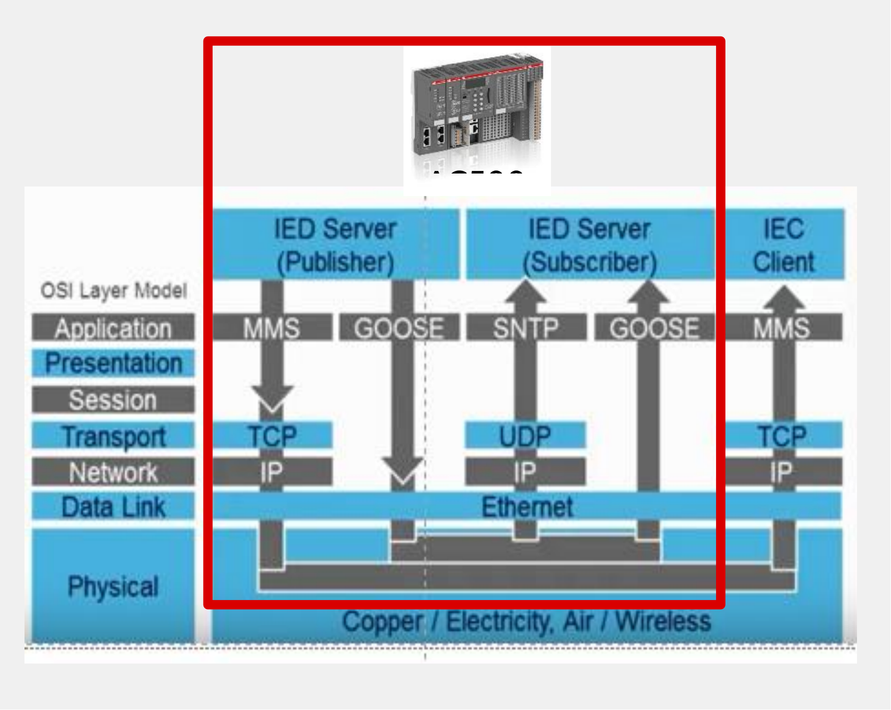
1. INTRODUCTION

1.1 Scope of the document

This document describes the example for IEC 61850 Server 4.0.4 which comes with Automation Builder 2.2.0.

This package allows the AC500 to act as an IED with IEC 61850 Server, Edition 1, providing the following functionality:

- The IEC 61850 Server connects substation automation systems with PLC applications
- AC500 V3 CPU acts as an IED with IEC 61850 Server, Edition 1, allowing communication as MMS Server and GOOSE Publisher and Subscriber



- Automation Builder is used as IED configuration tool for modelling the IEC 61850 data structures and connecting them to the PLC applications
- Support of SCL – Substation Configuration Language to transfer detailed configuration information between different IEDs

This document describes the examples which are part of Automation Builder. The engineering process with Automation Builder and 3rd party IED configuration tools is explained in order to setup the communication with MMS and GOOSE.

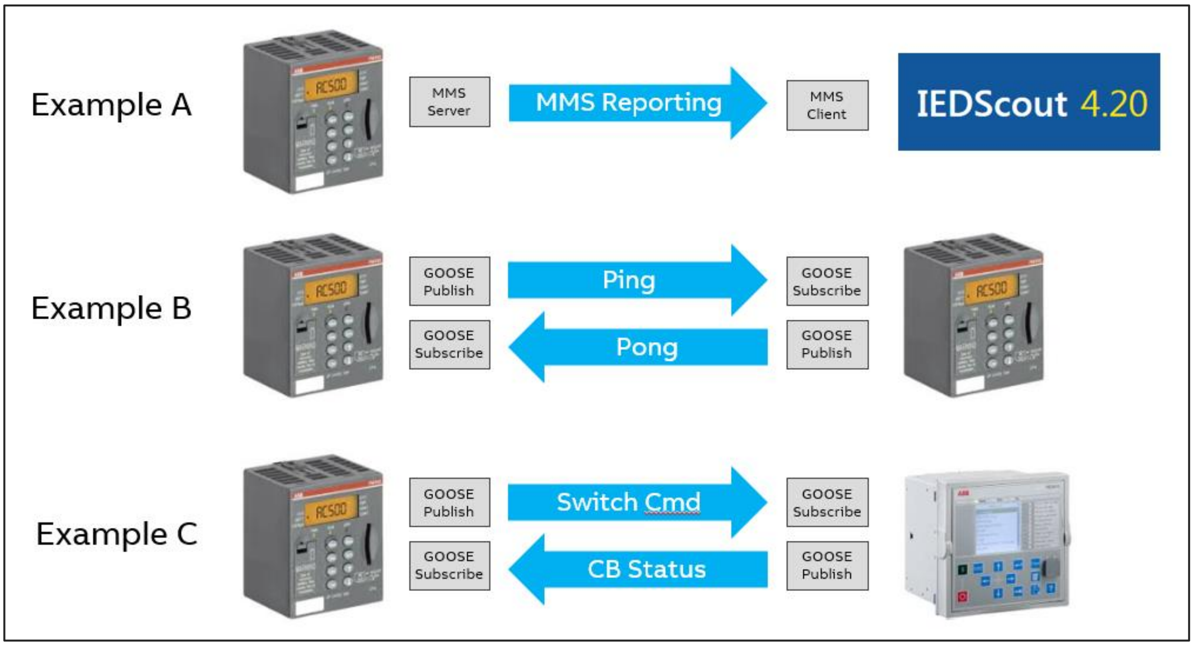
1.2 Overview of the Content

Chapter 2 describes the required hardware and software including the necessary connections for setting up the examples.

Chapter 3 contains the first example A where AC500 is configured as a simple MMS server which can be accessed by any IEC 61850 MMS client, for example IEC Scout from Omicron.

In chapter 4 (Example B) two AC500 are configured as GOOSE Publisher and GOOSE Subscriber in order to exchange data in both directions.

Chapter 5 describes Example C, where AC500 is controlling an IED from ABB, REF615, via GOOSE. A Switch command is sent from AC500 to REF615. The resulting status of the Circuit breaker is reported back from REF615 to AC500.



1.3 Safety Instructions and Preconditions

The user has to read the following instructions and documents before using the libraries:

- All pertinent state, regional, and local safety regulations must be observed when installing and using this product. When functions or devices are used for applications with technical safety requirements, the relevant instructions must be followed.
- Read the complete safety instructions of the user's manuals for the devices you are using, before installation and commissioning.
- Read all safety instructions of the AC500 PLC. See System description AC500 in the online help in Automation Builder
- Read the user Information of the devices and functions you are using, see online help in Automation Builder.

The IEC61850 Library package has been released for the software and firmware versions listed in the Readme file of the package only.

In no event will ABB or its representatives be liable for loss of data, profits, revenue or consequential, incidental or other damage that may result from the use of other versions of product, software or firmware versions. The error-free operation of the High Availability Modbus TCP Library with other devices, software or firmware versions should be possible but cannot be guaranteed and may need adaptations e.g. of example programs.

The user must follow all applicable safety instructions and the guidelines mentioned in the user documents of the ABB products.

Read the complete safety instructions for the AC500 before installation and commissioning.

⚠	CAUTION! Generally, the user in all applications is fully and alone responsible for checking all functions carefully, especially for safe and reliable operation.
☝	Note: The Function Blocks contained in the library can only be executed in RUN mode of the PLC, but not in simulation mode.

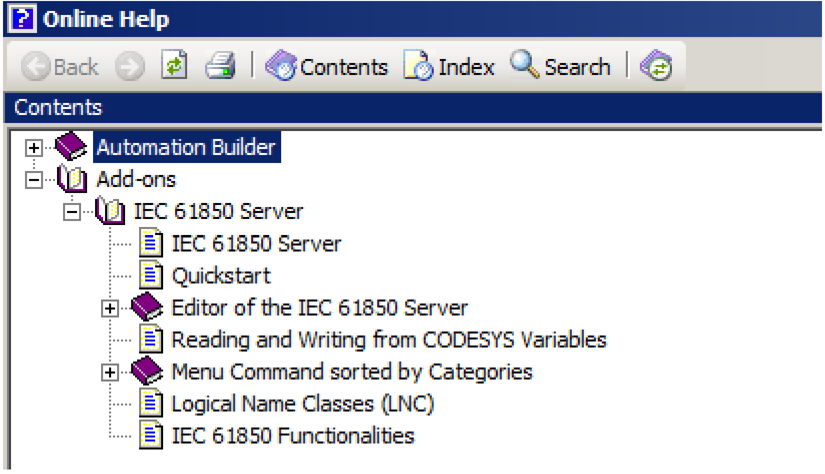
2. OVERVIEW

The IEC 61850 Server is part of Automation Builder 2.1.1 and higher.

2.1 Installation

The IEC61850 Server package can be installed as an option.

General configuration of IEC61850 is described in the online help in the Add-ons area:



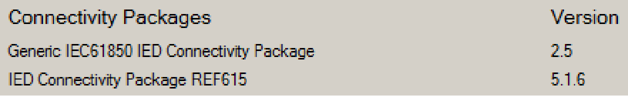
- Automation Builder
- Add-ons
  - IEC 61850 Server
    - IEC 61850 Server
    - Quickstart
    - Editor of the IEC 61850 Server
    - Reading and Writing from CODESYS Variables
    - Menu Command sorted by Categories
    - Logical Name Classes (LNC)
    - IEC 61850 Functionalities

This document provides additional AC500 specific details and examples.

2.2 Hardware and Software requirement

The following table gives an overview of required hardware and software for the different examples.

In the columns A, B and C the required number of items for examples A, B and C is listed, “man” stands for mandatory, “opt” for optionally.

Hardware	**Details**	**A**	**B**	**C**
AC500 V3 CPU	PM56xx, FW 3.2.0 or higher, with IEC61850 runtime license	1	2	1
Ethernet Switch			1	1
IED for bay control	REF615 REF from ABB or Demo Case (DSF615BJ2G) including REF615 plus simulation of a switchgear bay			1
Software on PC	**Details**			
Automation Builder	V2.2.0 or higher	man	man	man
IEC 61850 MMS client	Omicron, IED Scout 4.2.0 or other client	man		
Ethernet sniffer	Wireshark V2.4.2 or higher		opt	
IED engineering tool	PCM600 Version 2.8 including:  Connectivity Packages — Version Generic IEC61850 IED Connectivity Package — 2.5 IED Connectivity Package REF615 — 5.1.6			man

2.3 Limitations

- Edition 1 only
- Only MMS server, no MMS client. Workaround is to use GOOSE like in example C.

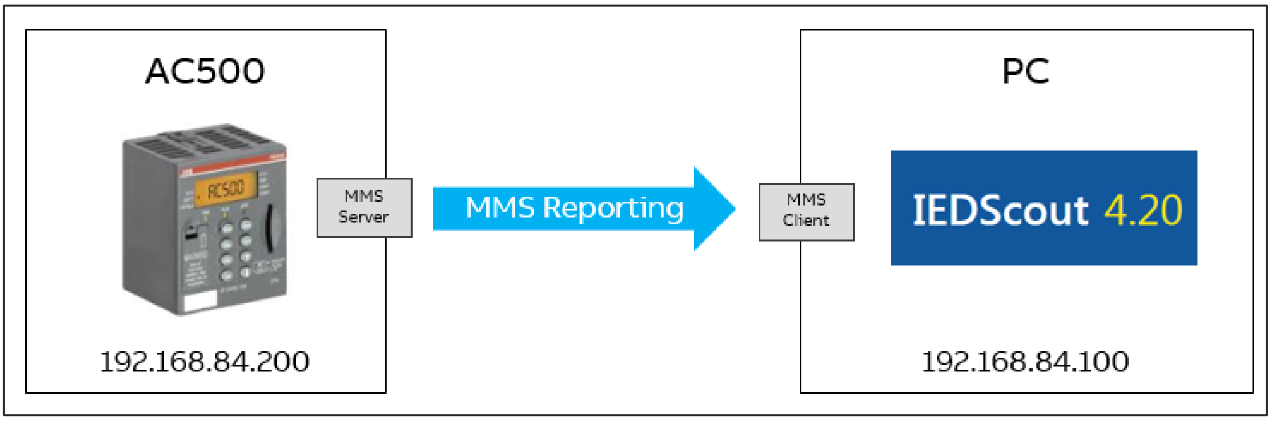
- The functionality behind the Logical Nodes need to be implemented by 61131 code
- Maximum of 5 Client connections per Server
- Maximum of 50 entries per Dataset
- Maximum of 20 DataSets and one Report per DataSet (PUA-167)
- Operation
  - Speed: Max 3000 Byte per cycle. Example: With an IEC61850-cycle time of 2ms it takes at least 10 ms to send 5 reports à 3000 Bytes
- Engineering
  - Not possible to have 2 or more IEC61850 server in one AB project. Workaround: Create 2 or more projects (PUA-172)
  - Only one Logical Node per IEC61850 Server
  - When data objects are inserted the first one has no suffix, e.g. "Ind" instead of "Ind0" (PUA-171)

3. EXAMPLE A: AC500 AS MMS SERVER

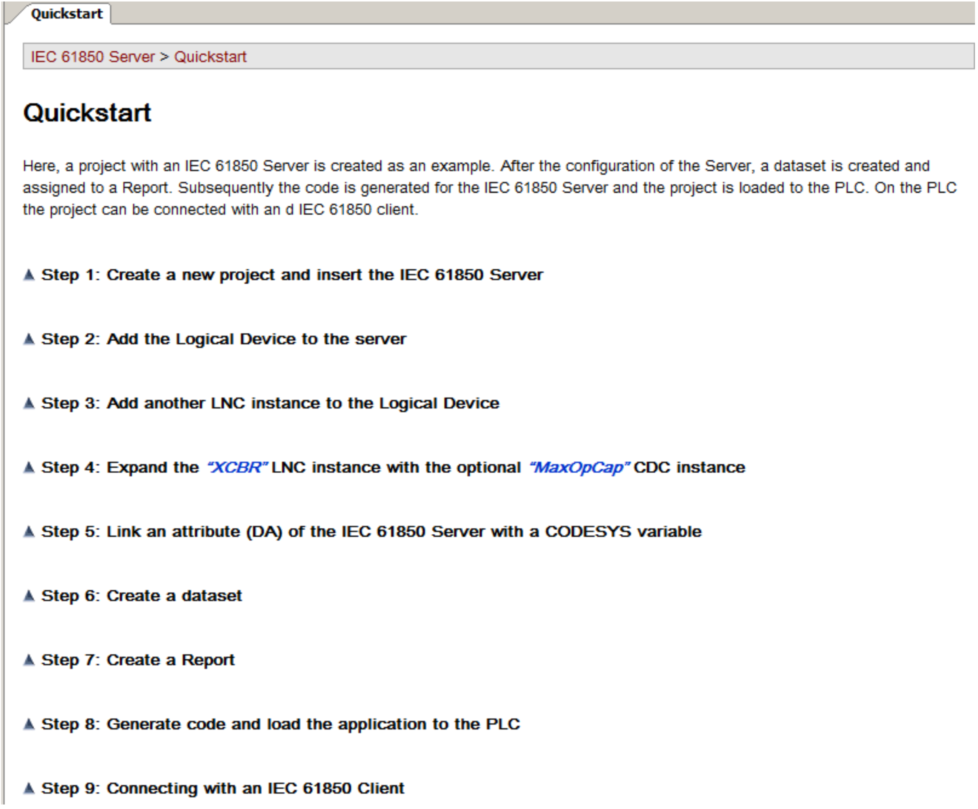
3.1 Basic configuration

Goal of this example is to configure the AC500 as IEC 61850 MMS Server, providing the mandatory Logical nodes and some optional ones in order to publish data from IEC61131 application which can be read by any IEC 61850 MMS client.

Physical connections: Ethernet connection between AC500 (V3) and the PC with Automation Builder and MMS test client, e.g. IED Scout from Omicron.



The engineering steps are according to the Quickstart in the online help:



Quickstart

IEC 61850 Server > Quickstart

Quickstart

Here, a project with an IEC 61850 Server is created as an example. After the configuration of the Server, a dataset is created and assigned to a Report. Subsequently the code is generated for the IEC 61850 Server and the project is loaded to the PLC. On the PLC the project can be connected with an d IEC 61850 client.

▲ Step 1: Create a new project and insert the IEC 61850 Server

▲ Step 2: Add the Logical Device to the server

▲ Step 3: Add another LNC instance to the Logical Device

▲ Step 4: Expand the *"XCBR"* LNC instance with the optional *"MaxOpCap"* CDC instance

▲ Step 5: Link an attribute (DA) of the IEC 61850 Server with a CODESYS variable

▲ Step 6: Create a dataset

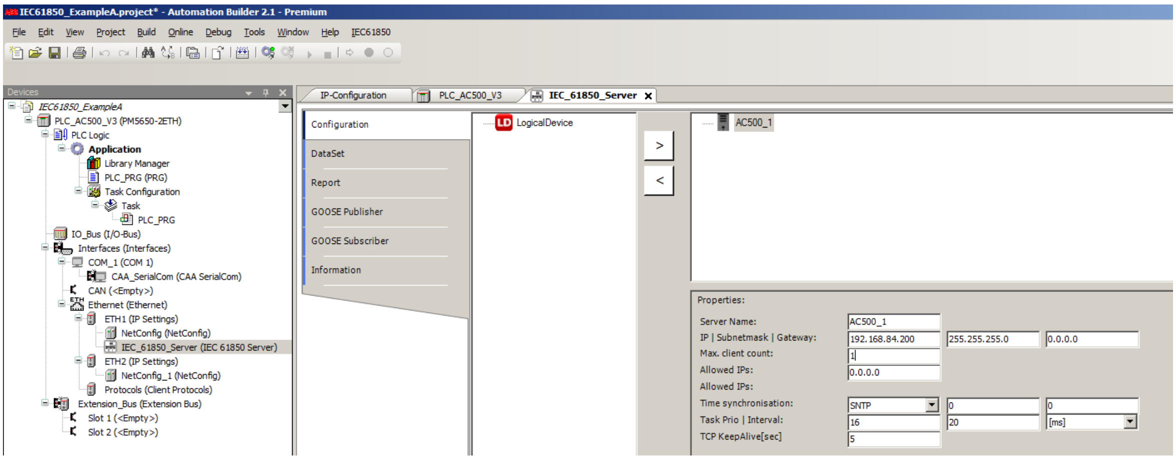
▲ Step 7: Create a Report

▲ Step 8: Generate code and load the application to the PLC

▲ Step 9: Connecting with an IEC 61850 Client

This chapter adds some details and screenshots for this specific example

1. **Create a new project and insert the IEC 61850 Server**
   Create a new project with AC500 V3 PM5650
   Connect to PLC and check if CPU firmware is 3.2.0 or higher.
   Add IEC 61850 Server node below ETH1
   Enter Server name = AC500_1
   Enter IP address of the PLC = 192.168.84.200



Properties:

Server Name: AC500_1

IP | Subnetmask | Gateway: 192.168.84.200 | 255.255.255.0 | 0.0.0.0

Max. client count: 1

Allowed IPs: 0.0.0.0

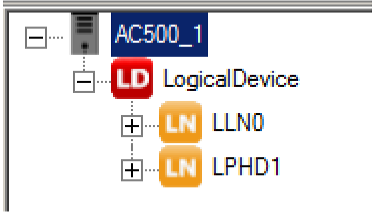
Allowed IPs:

Time synchronisation: SNTP | 0 | 0

Task Prio | Interval: 16 | 20 | [ms]

TCP KeepAlive[sec] 5

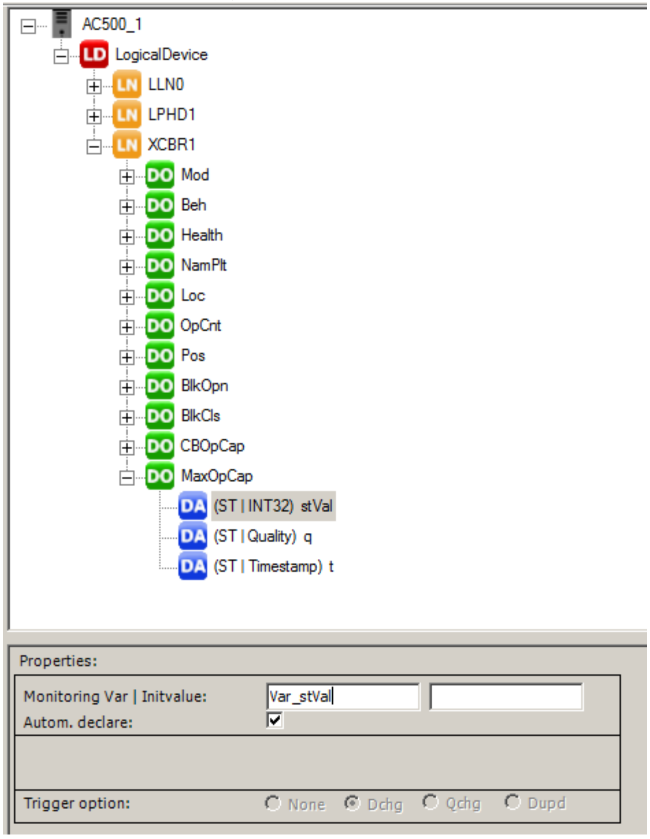
2. **Add the Logical Device to the server**



3. **Add another LNC instance to the Logical Device**

4. **Expand the “XCBR” LNC instance with the optional “MaxOpCap” CDC instance**

5. **Link an attribute (DA) of the IEC 61850 Server with a CODESYS variable**



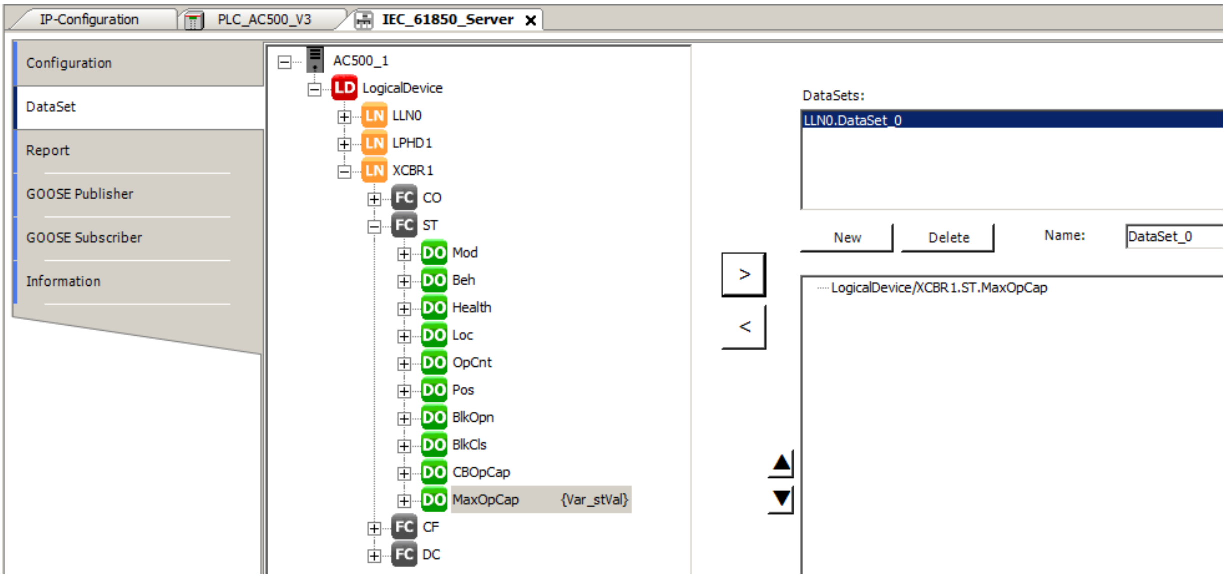
Properties:

Monitoring Var | Initvalue: Var_stVal

Autom. declare: ☑

Trigger option: ○ None ● Dchg ○ Qchg ○ Dupd

6. Create a dataset

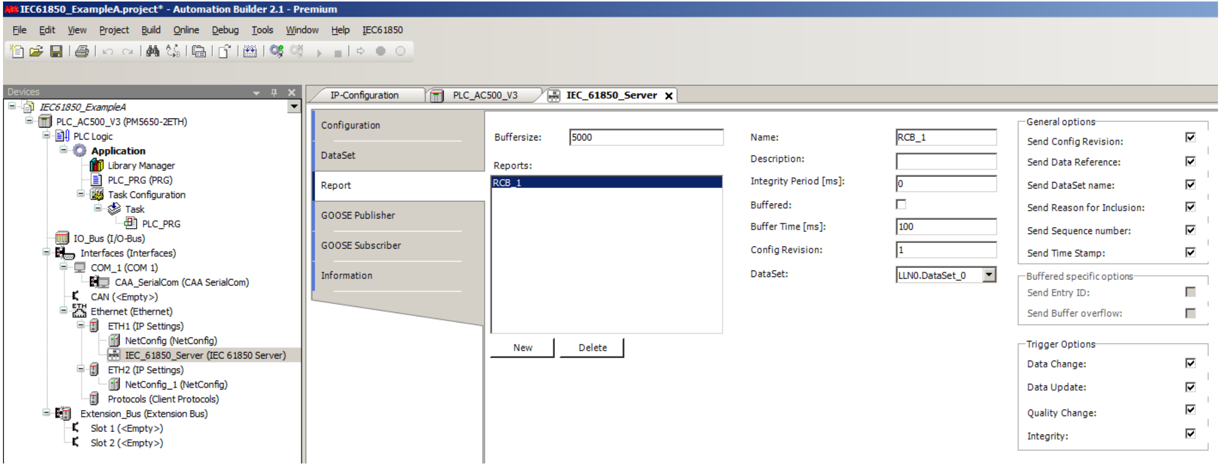


DataSets: LLN0.DataSet_0

Name: DataSet_0

LogicalDevice/XCBR1.ST.MaxOpCap

7. Create a Report



Buffersize: 5000

Reports: RCB_1

Name: RCB_1

Description:

Integrity Period [ms]: 0

Buffered: □

Buffer Time [ms]: 100

Config Revision: 1

DataSet: LLN0.DataSet_0

General options

- Send Config Revision: ☑
- Send Data Reference: ☑
- Send DataSet name: ☑
- Send Reason for Inclusion: ☑
- Send Sequence number: ☑
- Send Time Stamp: ☑

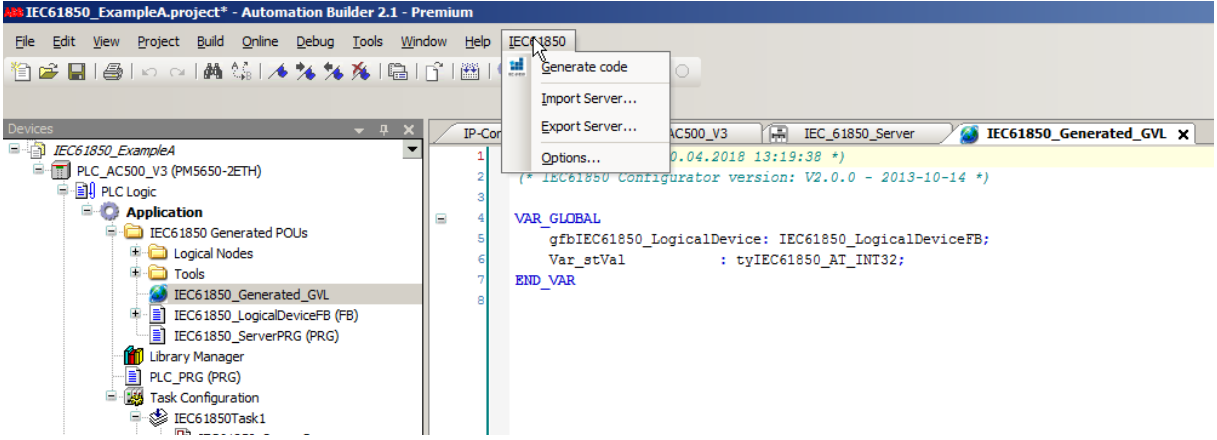
Buffered specific options

- Send Entry ID: □
- Send Buffer overflow: □

Trigger Options

- Data Change: ☑
- Data Update: ☑
- Quality Change: ☑
- Integrity: ☑

8. Generate code, compile, download, RUN



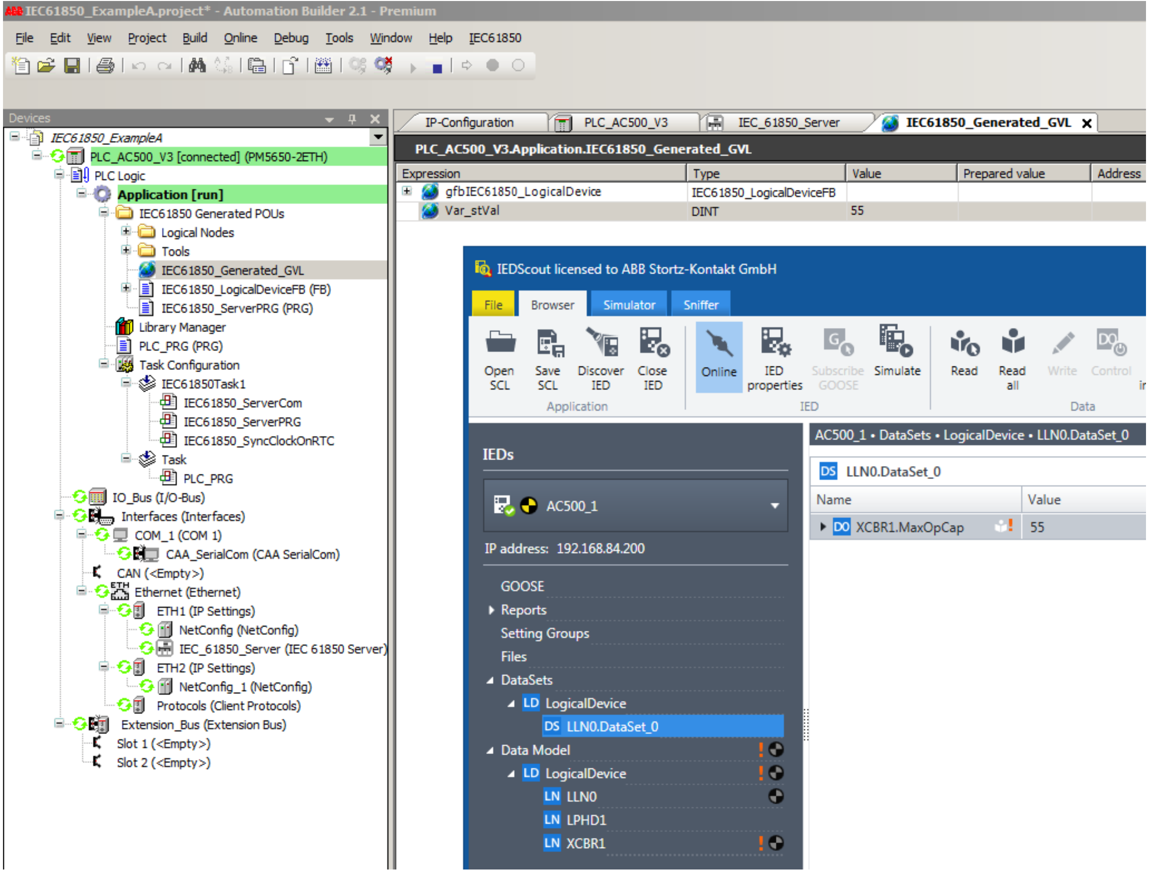
```
0.04.2018 13:19:38 *)
(* IEC61850 Configurator version: V2.0.0 - 2013-10-14 *)

VAR_GLOBAL
    gfbIEC61850_LogicalDevice: IEC61850_LogicalDeviceFB;
    Var_stVal            : tyIEC61850_AT_INT32;
END_VAR
```

9. Connecting with an IEC 61850 Client

This example describes the configuration of the IEDScout from Omicron as MMS client. Of course other IEC 61850 MMS clients can be used in a similar way.

Open the IEDScout and Discover IED by entering the IP address of the AC500: 192.168.84.200. The DataModel of the IED is read automatically and shown in the left part of the window. The DataSet LLN0.DataSet_0 contains the XCBR1.MaxOpCap which can be forced in the Automation Builder, for example to value 55. The same value is shown in the IEC Scout after pressing the button “Read” or “Read all”.



PLC_AC500_V3.Application.IEC61850_Generated_GVL

Expression	Type	Value	Prepared value	Address
gfbIEC61850_LogicalDevice	IEC61850_LogicalDeviceFB			
Var_stVal	DINT	55		

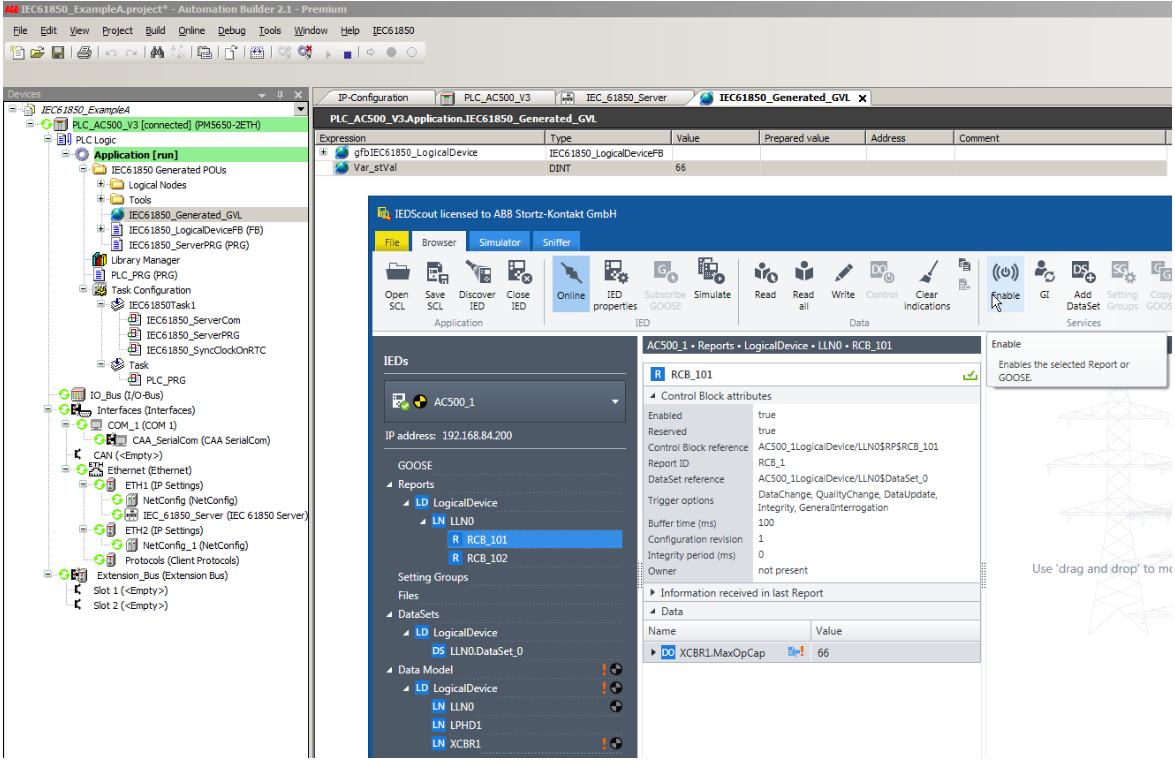
IP address: 192.168.84.200

AC500_1 • DataSets • LogicalDevice • LLN0.DataSet_0

DS LLN0.DataSet_0

Name	Value
DO XCBR1.MaxOpCap	55

For an automatic update of the values the report can be enabled. Afterwards the AC500 actively sends a report when the value has changed, e.g. from 55 to 66:



PLC_AC500_V3.Application.IEC61850_Generated_GVL

Expression	Type	Value	Prepared value	Address	Comment
gfbIEC61850_LogicalDevice	IEC61850_LogicalDeviceFB				
Var_stVal	DINT	66			

IP address: 192.168.84.200

AC500_1 • Reports • LogicalDevice • LLN0 • RCB_101

R RCB_101

Control Block attributes

Enabled	true
Reserved	true
Control Block reference	AC500_1LogicalDevice/LLN0$RP$RCB_101
Report ID	RCB_1
DataSet reference	AC500_1LogicalDevice/LLN0$DataSet_0
Trigger options	DataChange, QualityChange, DataUpdate, Integrity, GeneralInterrogation
Buffer time (ms)	100
Configuration revision	1
Integrity period (ms)	0
Owner	not present

Information received in last Report

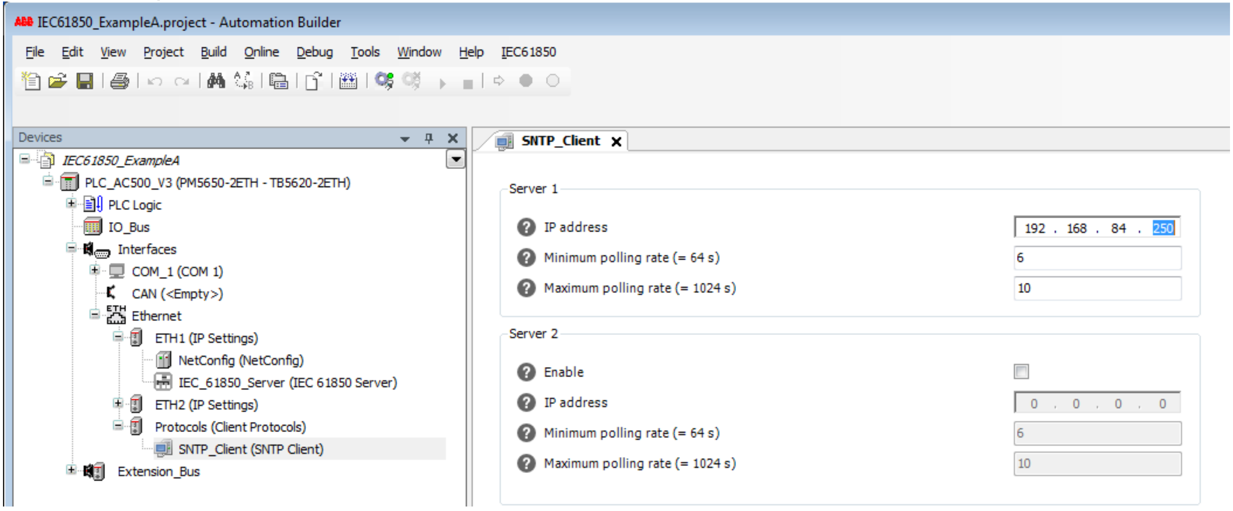
Data

Name	Value
DO XCBR1.MaxOpCap	66

3.2 Time sync

If an SNTP Server (e.g. Meinberg clock) is available in the network the AC500 can be configured as SNTP client in order to receive and provide the correct time information in the MMS and GOOSE reports.

Therefore an SNTP client node must be added and configured (SNTP Server is 192.168.84.250 in this example)


IEC61850_ExampleA.project - Automation Builder
File Edit View Project Build Online Debug Tools Window Help IEC61850
Devices
IEC61850_ExampleA
PLC_AC500_V3 (PM5650-2ETH - TB5620-2ETH)
PLC Logic
IO_Bus
Interfaces
COM_1 (COM 1)
CAN (<Empty>)
Ethernet
ETH1 (IP Settings)
NetConfig (NetConfig)
IEC_61850_Server (IEC 61850 Server)
ETH2 (IP Settings)
Protocols (Client Protocols)
SNTP_Client (SNTP Client)
Extension_Bus
SNTP_Client

Server 1

IP address	192 . 168 . 84 . 250
Minimum polling rate (= 64 s)	6
Maximum polling rate (= 1024 s)	10

Server 2

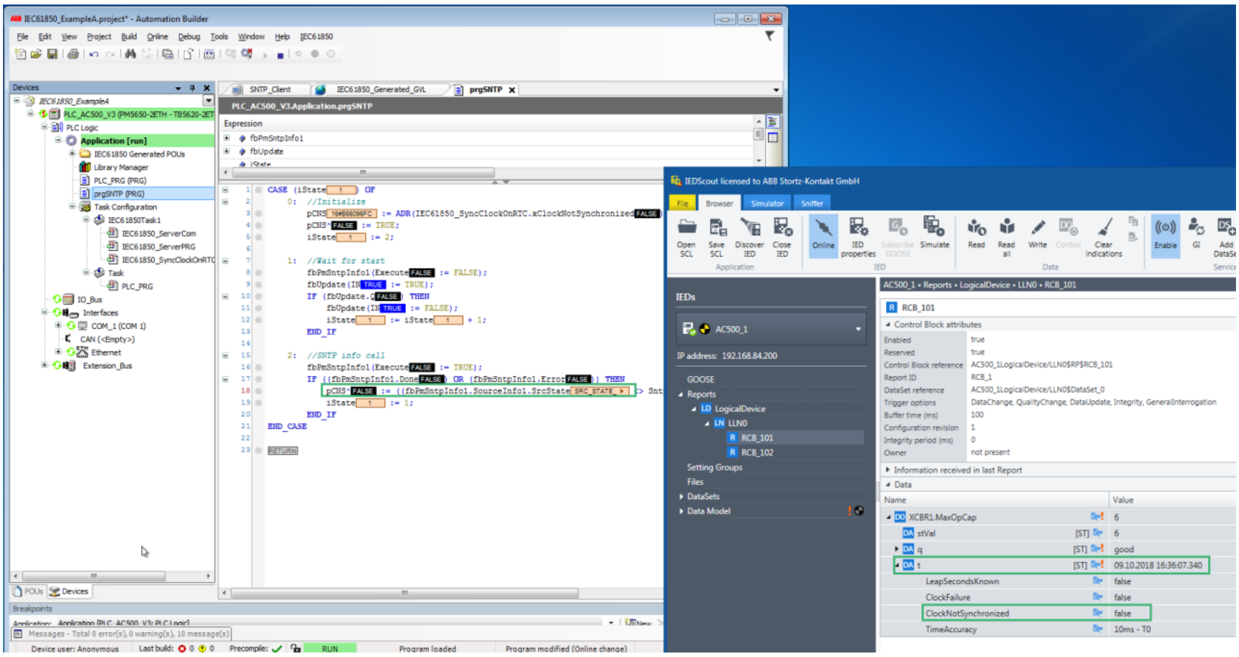
Enable	□
IP address	0 . 0 . 0 . 0
Minimum polling rate (= 64 s)	6
Maximum polling rate (= 1024 s)	10

All parameter of SNPT client are described in the online help → search for SNTP Client Configuration.

With the program "prgSNTP" the SNTP Server is scanned every 30 seconds and the resulting quality bit is written to a pointer to the internal variable xClockNotSynchronized of IEC61850_SyncClockOnRTC

- In step 0 the pointer is defined
- In step 2 the status of SNTP is checked and updated
- In step 1 a timer waits for 30 seconds until it goes back to step 2

If the SNTP time is working correctly the "ClockNotSynchronized" is False



```
CASE (iState) OF
    0:  //Initialize
        pCNS := ADR(IEC61850_SyncClockOnRTC.xClockNotSynchronized)
        pCNS^ := TRUE;
        iState := 2;

    1:  //Wait for start
        fbPmSntpInfo1(Execute := FALSE);
        fbUpdate(IN := TRUE);
        IF (fbUpdate.Q) THEN
            fbUpdate(IN := FALSE);
            iState := iState + 1;
        END_IF

    2:  //SNTP info call
        fbPmSntpInfo1(Execute := TRUE);
        IF ((fbPmSntpInfo1.Done) OR (fbPmSntpInfo1.Error)) THEN
            pCNS^ := ((fbPmSntpInfo1.SourceInfo1.SrcState <> Snt
            iState := 1;
        END_IF
END_CASE

RETURN
```

AC500_1 • Reports • LogicalDevice • LLN0 • RCB_101

RCB_101

Control Block attributes

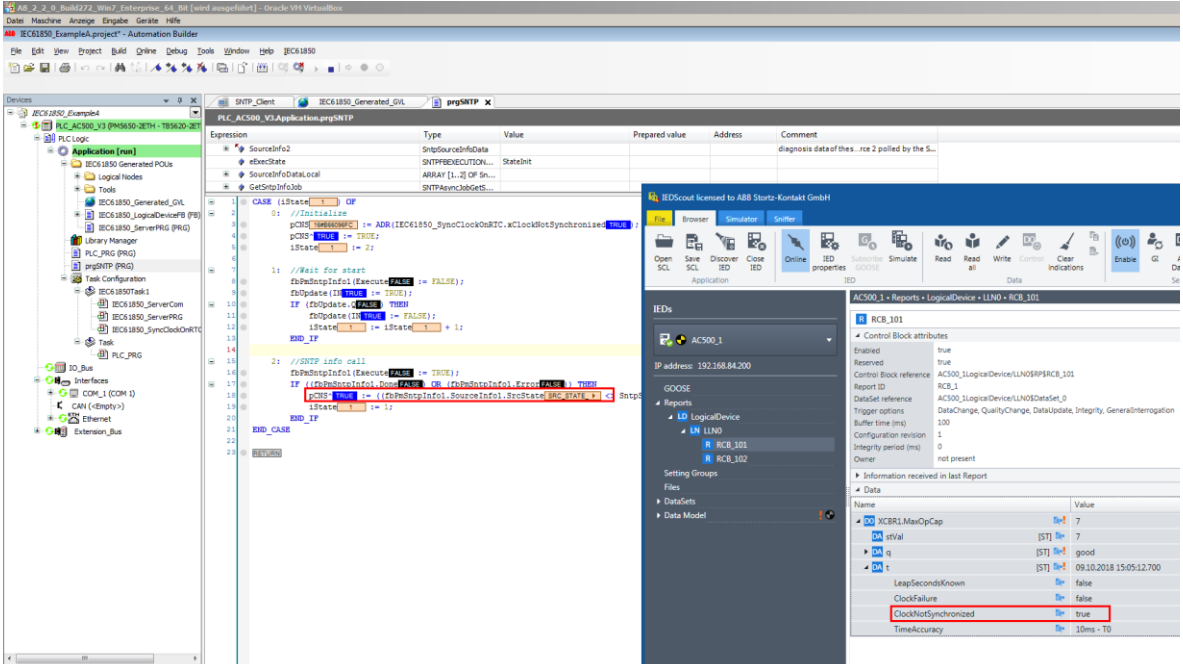
Enabled	true
Reserved	true
Control Block reference	AC500_1LogicalDevice/LLN0$RP$RCB_101
Report ID	RCB_1
DataSet reference	AC500_1LogicalDevice/LLN0$DataSet_0
Trigger options	DataChange, QualityChange, DataUpdate, Integrity, GeneralInterrogation
Buffer time (ms)	100
Configuration revision	1
Integrity period (ms)	0
Owner	not present

Information received in last Report

Data

Name		Value
XCBR1.MaxOpCap		6
stVal	[ST]	6
q	[ST]	good
t	[ST]	09.10.2018 16:36:07.340
LeapSecondsKnown		false
ClockFailure		false
ClockNotSynchronized		false
TimeAccuracy		10ms - T0

If the SNTP Server fails or connection is broken the “ClockNotSynchronized” is True:



PLC_AC500_V3.Application.prgSNTP

Expression	Type	Value	Prepared value	Address	Comment
SourceInfo2	SntpSourceInfoData				diagnosis data of thes…rce 2 polled by the S…
eExecState	SNTPFBEXECUTION…	StateInit			
SourceInfoDataLocal	ARRAY [1..2] OF Sn…				
GetSntpInfoJob	SNTPAsyncJobGetS…				

```
CASE (iState) OF
    0:  //Initialize
        pCNS := ADR(IEC61850_SyncClockOnRTC.xClockNotSynchronized);
        pCNS^ := TRUE;
        iState := 2;

    1:  //Wait for start
        fbPmSntpInfo1(Execute := FALSE);
        fbUpdate(IN := TRUE);
        IF (fbUpdate.Q) THEN
            fbUpdate(IN := FALSE);
            iState := iState + 1;
        END_IF

    2:  //SNTP info call
        fbPmSntpInfo1(Execute := TRUE);
        IF ((fbPmSntpInfo1.Done) OR (fbPmSntpInfo1.Error)) THEN
            pCNS^ := ((fbPmSntpInfo1.SourceInfo1.SrcState <> SntpS
            iState := 1;
        END_IF
END_CASE

RETURN
```

IEDScout licensed to ABB Stortz-Kontakt GmbH

AC500_1 • Reports • LogicalDevice • LLN0 • RCB_101

RCB_101

Control Block attributes

Enabled	true
Reserved	true
Control Block reference	AC500_1LogicalDevice/LLN0$RP$RCB_101
Report ID	RCB_1
DataSet reference	AC500_1LogicalDevice/LLN0$DataSet_0
Trigger options	DataChange, QualityChange, DataUpdate, Integrity, GeneralInterrogation
Buffer time (ms)	100
Configuration revision	1
Integrity period (ms)	0
Owner	not present

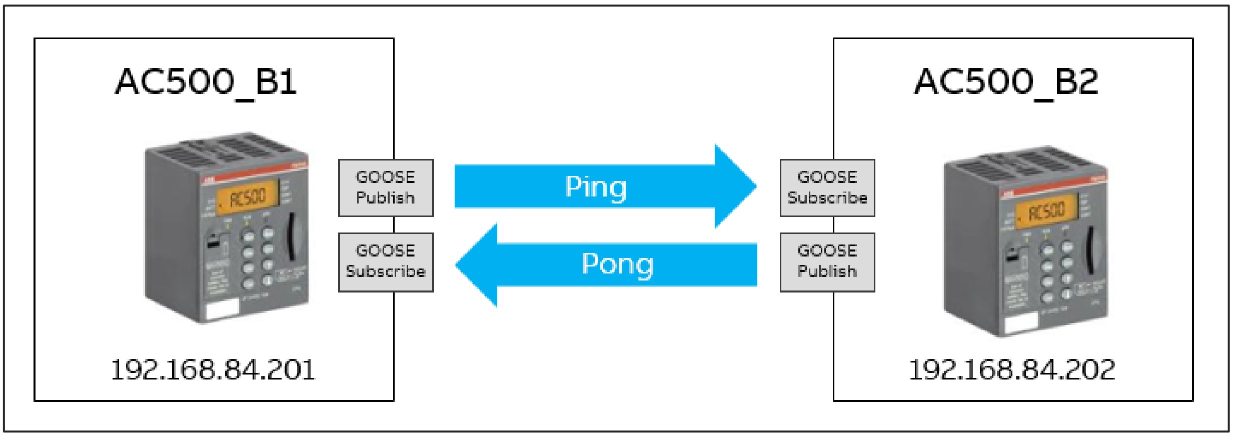
Information received in last Report

Data

Name		Value
XCBR1.MaxOpCap		7
stVal	[ST]	7
q	[ST]	good
t	[ST]	09.10.2018 15:05:12.700
LeapSecondsKnown		false
ClockFailure		false
ClockNotSynchronized		true
TimeAccuracy		10ms - T0

4. EXAMPLE B: TWO AC500 COMMUNICATING VIA GOOSE

Goal of this example is setup a GOOSE communication between 2 AC500 PLCs.
AC500_B1 sends a “Ping” signal as GOOSE publisher. AC500_B2 receives this signal as GOOSE subscriber. The answer “Pong” is sent from AC500_B2 to AC500_B1 in the same way.



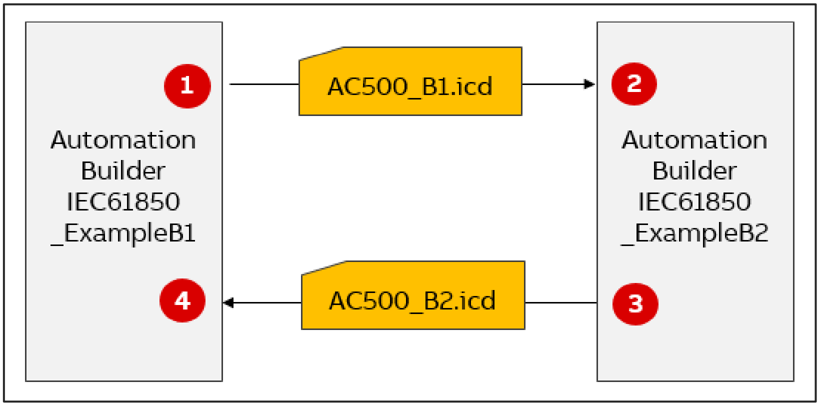
Physical connections: Switch between AC500_B1 and AC500B2 and the PC with Automation Builder and optionally Wireshark for monitoring the Ethernet traffic.

The following sub-chapter show all single engineering steps.

The alternative is to use the existing example projects. In this case the IP addresses and MAC addresses have to be adapted according to the hardware setup.

4.1 Set up GOOSE communication

The engineering is done in two Automation Builder projects, one for each AC500. The exchange of engineering information is done with .icd files according to SCL = Substation Configuration Language:


1
Automation Builder IEC61850 _ExampleB1
4
AC500_B1.icd
AC500_B2.icd
2
Automation Builder IEC61850 _ExampleB2
3

The configuration is done in 4 steps:

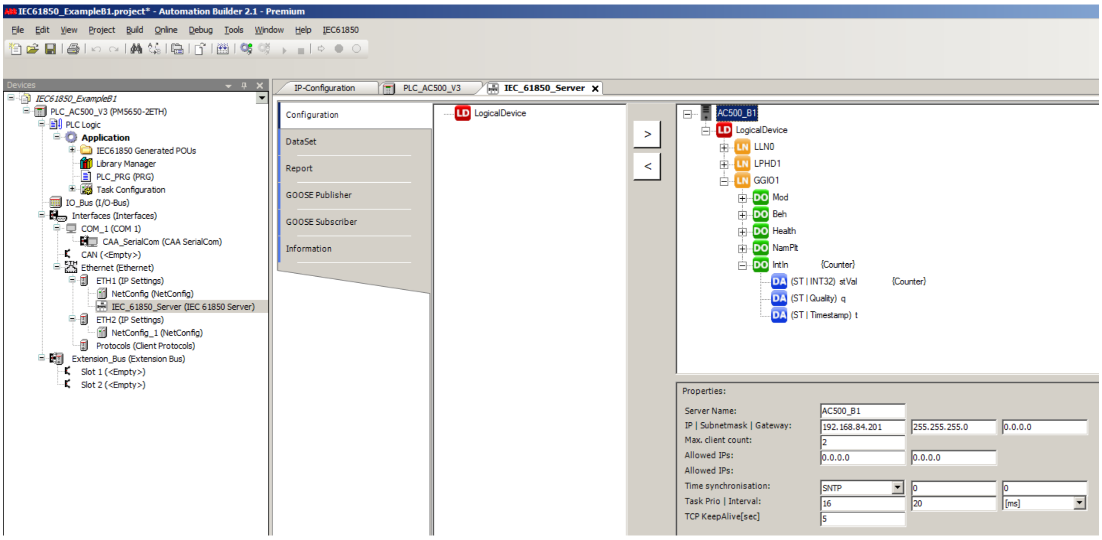
1. GOOSE publisher of AC500_B1, export of AC500_B1.icd
2. GOOSE subscriber of AC500_B2 by importing AC500_B1.icd
3. GOOSE publisher of AC500_B2, export of AC500_B2.icd
4. GOOSE subscriber of AC500_B1 by importing AC500_B2.icd

The details are explained in the following:

1. GOOSE publisher of AC500_B1

a. Create project AC500_B1, insert AC500 V3, insert IEC61850 Server “AC500_B1” with IP address 192.168.84.201, connect to the PLC **AC500_B1**

b. Add Logical Node “GGIO1” and DataAttribute “IntIn” which is connected to 61131-variable “Counter”:

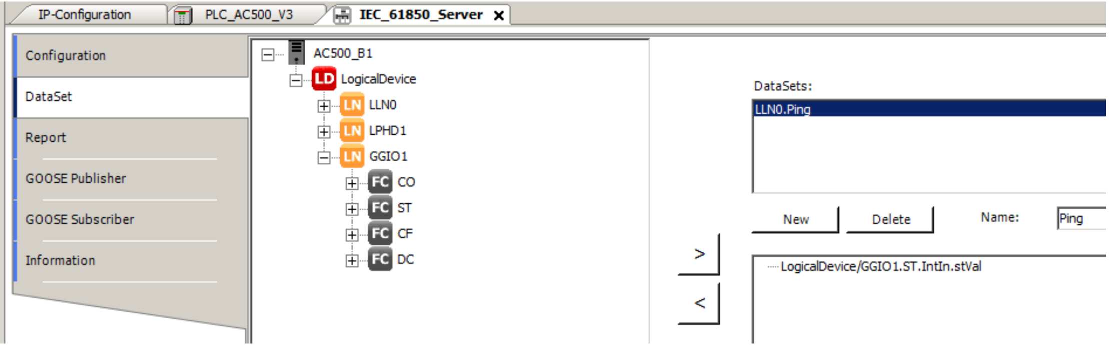


AC500_B1
LogicalDevice
LLN0
LPHD1
GGIO1
Mod
Beh
Health
NamPlt
IntIn {Counter}
(ST | INT32) stVal {Counter}
(ST | Quality) q
(ST | Timestamp) t

Properties:

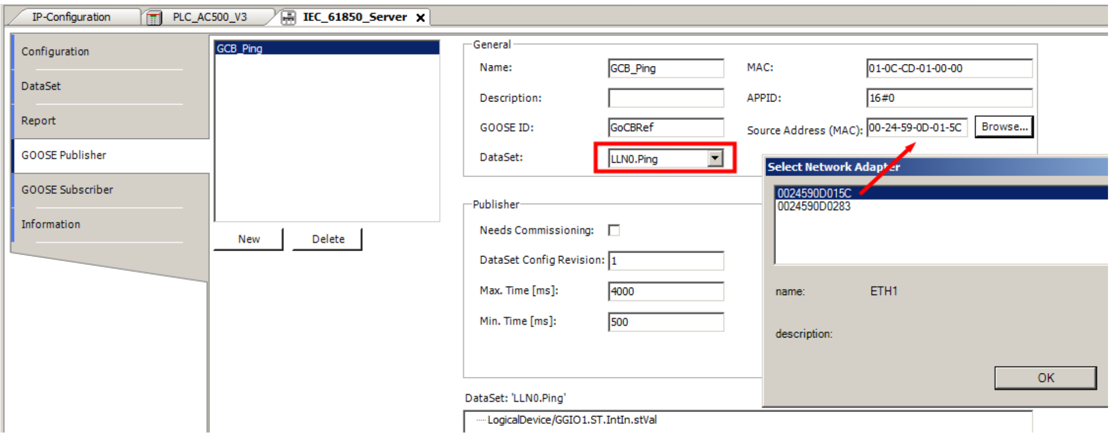
Server Name:	AC500_B1		
IP | Subnetmask | Gateway:	192.168.84.201	255.255.255.0	0.0.0.0
Max. client count:	2		
Allowed IPs:	0.0.0.0	0.0.0.0	
Allowed IPs:			
Time synchronisation:	SNTP	0	0
Task Prio | Interval:	16	20	[ms]
TCP KeepAlive[sec]	5		

c. Create a dataset “Ping” and insert the GGIO1.ST_IntIn.stVal



DataSets:
LLN0.Ping
Name: Ping
LogicalDevice/GGIO1.ST.IntIn.stVal

d. Create a GOOSE Control Block “GCB_Ping”, choose the Dataset “Ping” and select the Source MAC address of ETH1:



GCB_Ping

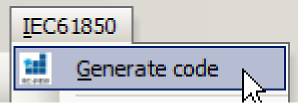
General
Name: GCB_Ping
Description:
GOOSE ID: GoCBRef
DataSet: LLN0.Ping
MAC: 01-0C-CD-01-00-00
APPID: 16#0
Source Address (MAC): 00-24-59-0D-01-5C

Publisher
Needs Commissioning: □
DataSet Config Revision: 1
Max. Time [ms]: 4000
Min. Time [ms]: 500

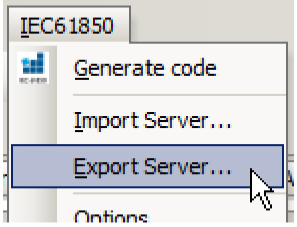
DataSet: 'LLN0.Ping'
LogicalDevice/GGIO1.ST.IntIn.stVal

Select Network Adapter
0024590D015C
0024590D0283
name: ETH1
description:

e. Generate Code

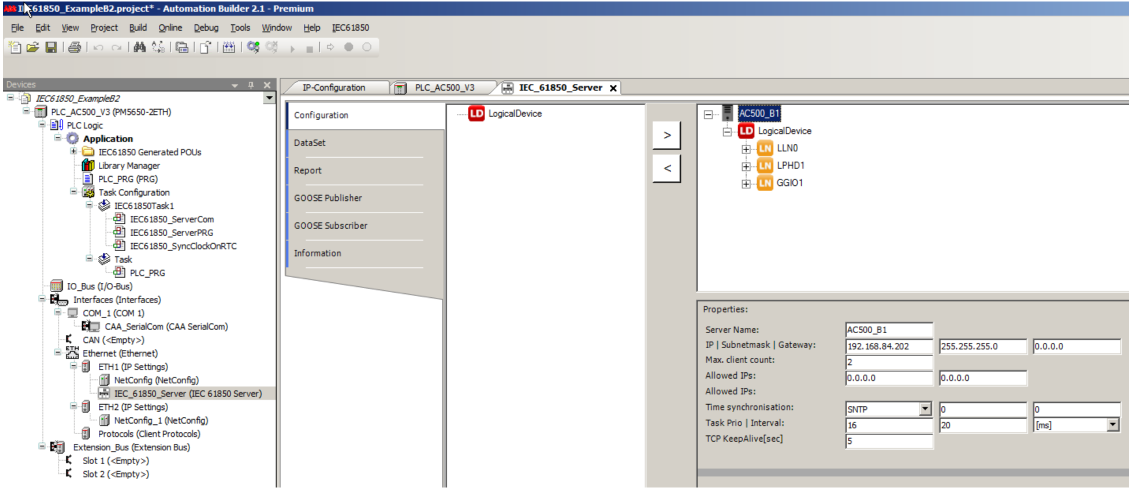


f. Export configuration as AC500_B1.icd



2. GOOSE subscriber of AC500_B2

a. Create project AC500_B2, insert IEC61850 Server “AC500_B2” with IP address 192.168.84.202, connect to **PLC AC500_B2**



Properties:

Server Name: AC500_B1

IP | Subnetmask | Gateway: 192.168.84.202 | 255.255.255.0 | 0.0.0.0

Max. client count: 2

Allowed IPs: 0.0.0.0 | 0.0.0.0

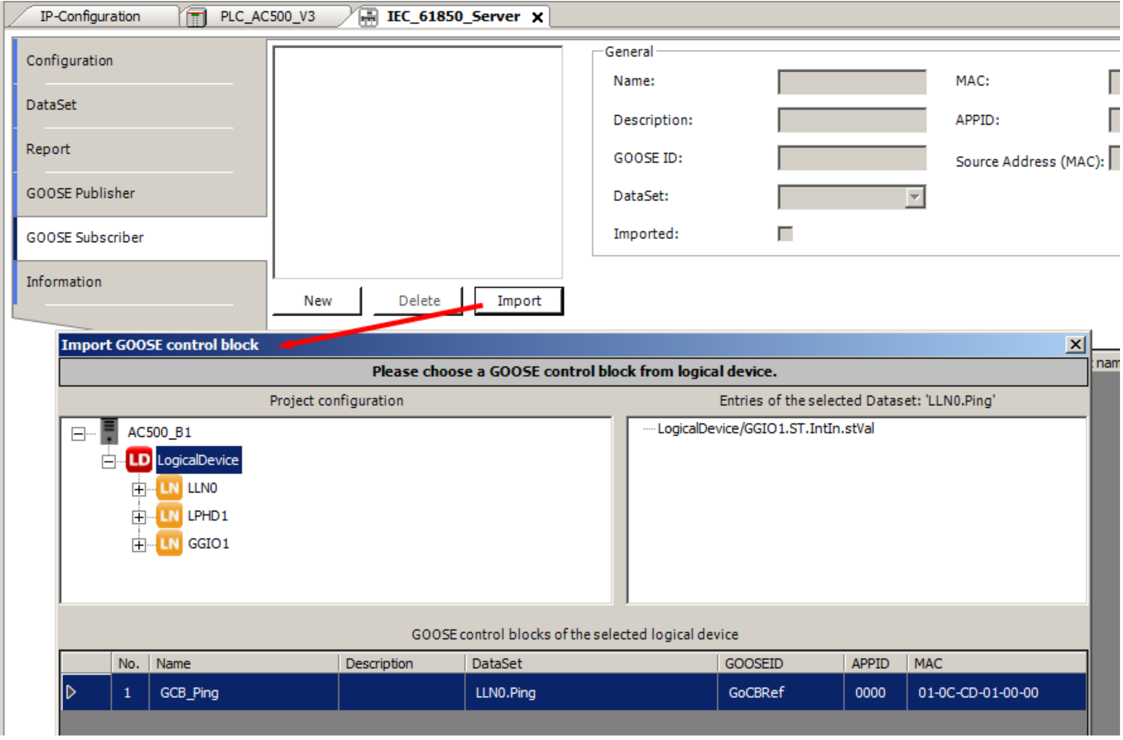
Allowed IPs:

Time synchronisation: SNTP | 0 | 0

Task Prio | Interval: 16 | 20 | [ms]

TCP KeepAlive[sec] 5

b. Configure GOOSE Subscriber by pushing the button “Import” and selecting the AC500_B1.icd. In the “Import GOOSE control block” window the GCG_Ping can be chosen by clicking “OK”



Please choose a GOOSE control block from logical device.

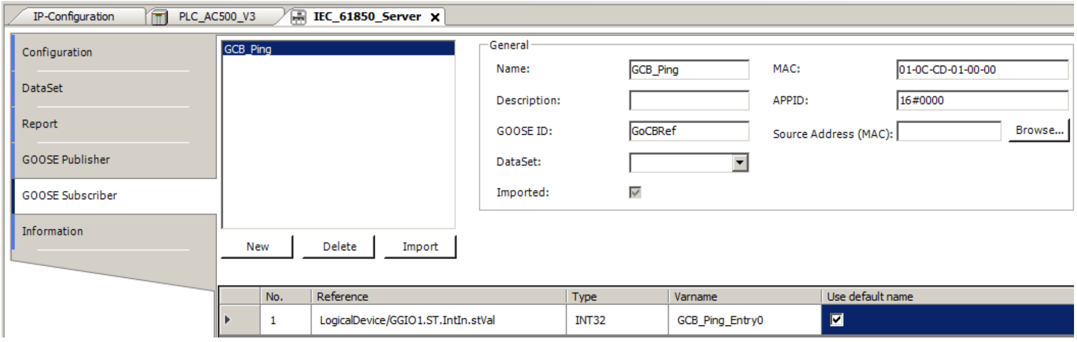
Entries of the selected Dataset: 'LLN0.Ping'

LogicalDevice/GGIO1.ST.IntIn.stVal

GOOSE control blocks of the selected logical device

	No.	Name	Description	DataSet	GOOSEID	APPID	MAC
▷	1	GCB_Ping		LLN0.Ping	GoCBRef	0000	01-0C-CD-01-00-00

c. The signal is automatically added to the list. By choosing the option “Use default name”, the 61131 variable “GCB_Ping_Entry0” is generated automatically.

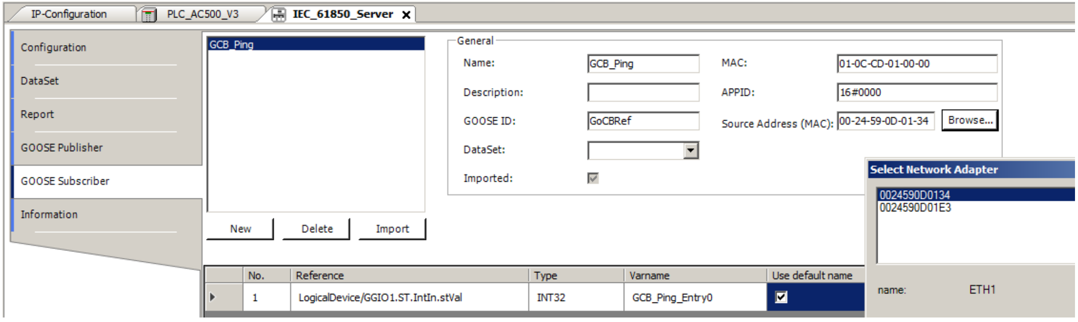


General

Name: GCB_Ping
Description:
GOOSE ID: GoCBRef
DataSet:
Imported: ☑
MAC: 01-0C-CD-01-00-00
APPID: 16#0000
Source Address (MAC):

	No.	Reference	Type	Varname	Use default name
▸	1	LogicalDevice/GGIO1.ST.IntIn.stVal	INT32	GCB_Ping_Entry0	☑

d. Furthermore the Source MAC address has to be selected.
Important: This must be the ETH1 MAC address of **AC500_B2!**



General

Name: GCB_Ping
Description:
GOOSE ID: GoCBRef
DataSet:
Imported: ☑
MAC: 01-0C-CD-01-00-00
APPID: 16#0000
Source Address (MAC): 00-24-59-0D-01-34

Select Network Adapter

0024590D0134
0024590D01E3

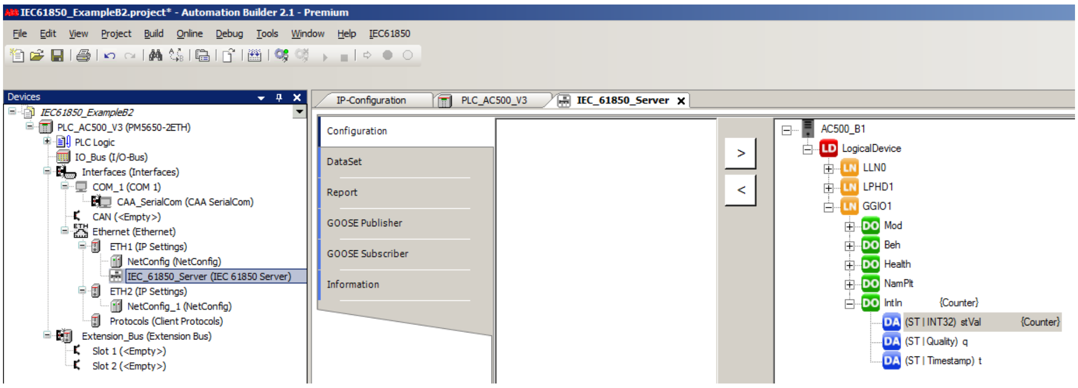
name: ETH1

	No.	Reference	Type	Varname	Use default name
▸	1	LogicalDevice/GGIO1.ST.IntIn.stVal	INT32	GCB_Ping_Entry0	☑

3. GOOSE publisher of AC500_B2
This is similar to step 1, merely “Ping” is substituted by “Pong”:

a. Project “IEC61850_ExampleB2” is already existing from previous step 2.

b. Add Logical Node “GGIO1” and DataAttribute “IntIn” which is connected to 61131-variable “Counter”:



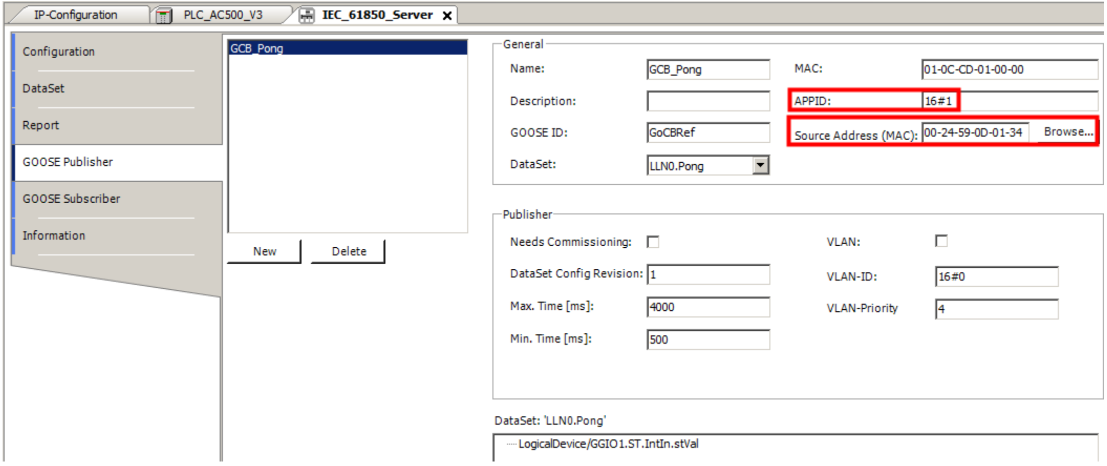
IEC61850_ExampleB2
PLC_AC500_V3 (PM5650-2ETH)
PLC Logic
IO_Bus (I/O-Bus)
Interfaces (Interfaces)
COM_1 (COM 1)
CAA_SerialCom (CAA SerialCom)
CAN (<Empty>)
Ethernet (Ethernet)
ETH1 (IP Settings)
NetConfig (NetConfig)
IEC_61850_Server (IEC 61850 Server)
ETH2 (IP Settings)
NetConfig_1 (NetConfig)
Protocols (Client Protocols)
Extension_Bus (Extension Bus)
Slot 1 (<Empty>)
Slot 2 (<Empty>)

AC500_B1
LD LogicalDevice
LN LLN0
LN LPHD1
LN GGIO1
DO Mod
DO Beh
DO Health
DO NamPlt
DO IntIn {Counter}
DA (ST | INT32) stVal {Counter}
DA (ST | Quality) q
DA (ST | Timestamp) t

c. Create a dataset “Pong” and insert the GGIO1.ST_IntIn.stVal

d. Create a GOOSE Control Block “GCB_Pong”, choose the Dataset “Pong” and select the Source MAC address of ETH1.

Important: Choose a different APPID, e.g. 16#1 (AC500_B1 had 16#0)



General

Name: GCB_Pong

Description:

GOOSE ID: GoCBRef

DataSet: LLN0.Pong

MAC: 01-0C-CD-01-00-00

APPID: 16#1

Source Address (MAC): 00-24-59-0D-01-34

Publisher

Needs Commissioning: □

DataSet Config Revision: 1

Max. Time [ms]: 4000

Min. Time [ms]: 500

VLAN: □

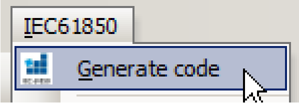
VLAN-ID: 16#0

VLAN-Priority: 4

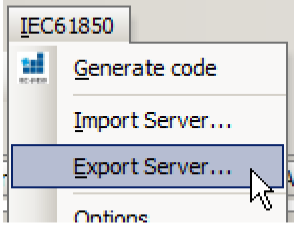
DataSet: 'LLN0.Pong'

LogicalDevice/GGIO1.ST.IntIn.stVal

e. Generate Code



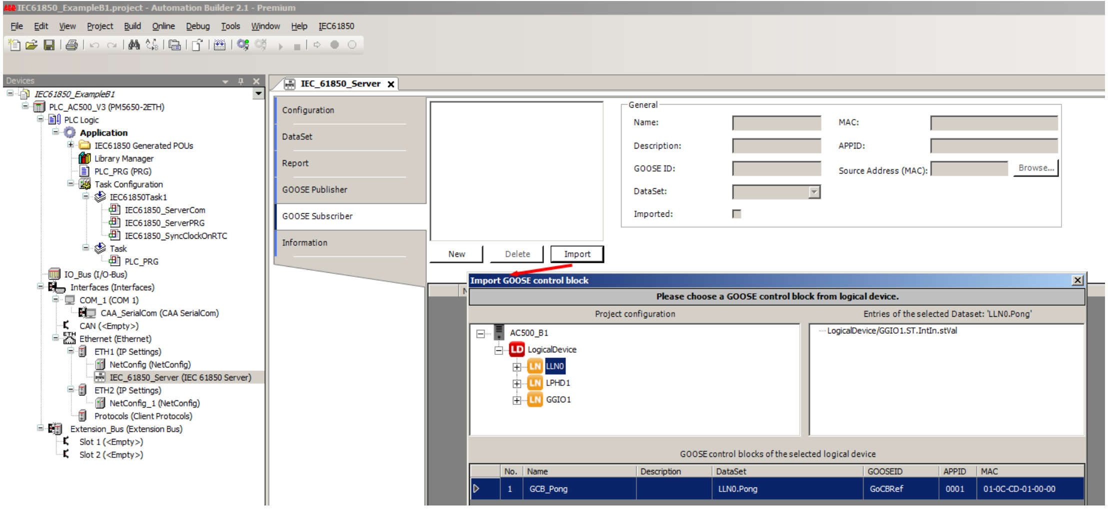
f. Export configuration as AC500_B2.icd



4. GOOSE subscriber of AC500_B2
   This is similar to step 2

   a. Open project IEC61850_ExampleB1

   b. Configure GOOSE Subscriber by pushing the button “Import” and selecting the AC500_B2.icd. In the “Import GOOSE control block” window the GCG_Pong can be chosen by clicking “OK”



General

Name:

Description:

GOOSE ID:

DataSet:

Imported: □

MAC:

APPID:

Source Address (MAC):

Import GOOSE control block

Please choose a GOOSE control block from logical device.

Project configuration

AC500_B1
LD LogicalDevice
LN LLN0
LN LPHD1
LN GGIO1

Entries of the selected Dataset: 'LLN0.Pong'

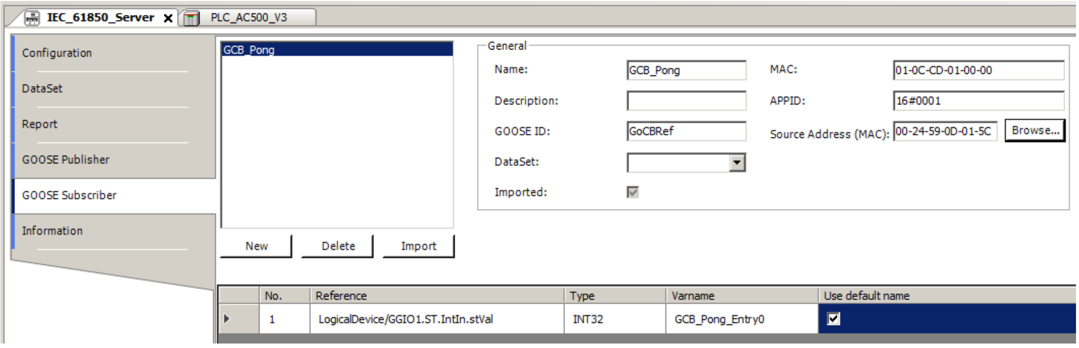
LogicalDevice/GGIO1.ST.IntIn.stVal

GOOSE control blocks of the selected logical device

No.	Name	Description	DataSet	GOOSEID	APPID	MAC
1	GCB_Pong		LLN0.Pong	GoCBRef	0001	01-0C-CD-01-00-00

   c. The signal is automatically added to the list. By choosing the option “Use default name”, the 61131 variable “GCB_Pong_Entry0” is generated automatically.

   d. Furthermore the Source MAC address has to be selected.
   Important: This must be the ETH1 MAC address of **AC500_B1!**



Name: GCB_Pong

Description:

GOOSE ID: GoCBRef

DataSet:

Imported: ☑

MAC: 01-0C-CD-01-00-00

APPID: 16#0001

Source Address (MAC): 00-24-59-0D-01-5C

No.	Reference	Type	Varname	Use default name
1	LogicalDevice/GGIO1.ST.IntIn.stVal	INT32	GCB_Pong_Entry0	☑

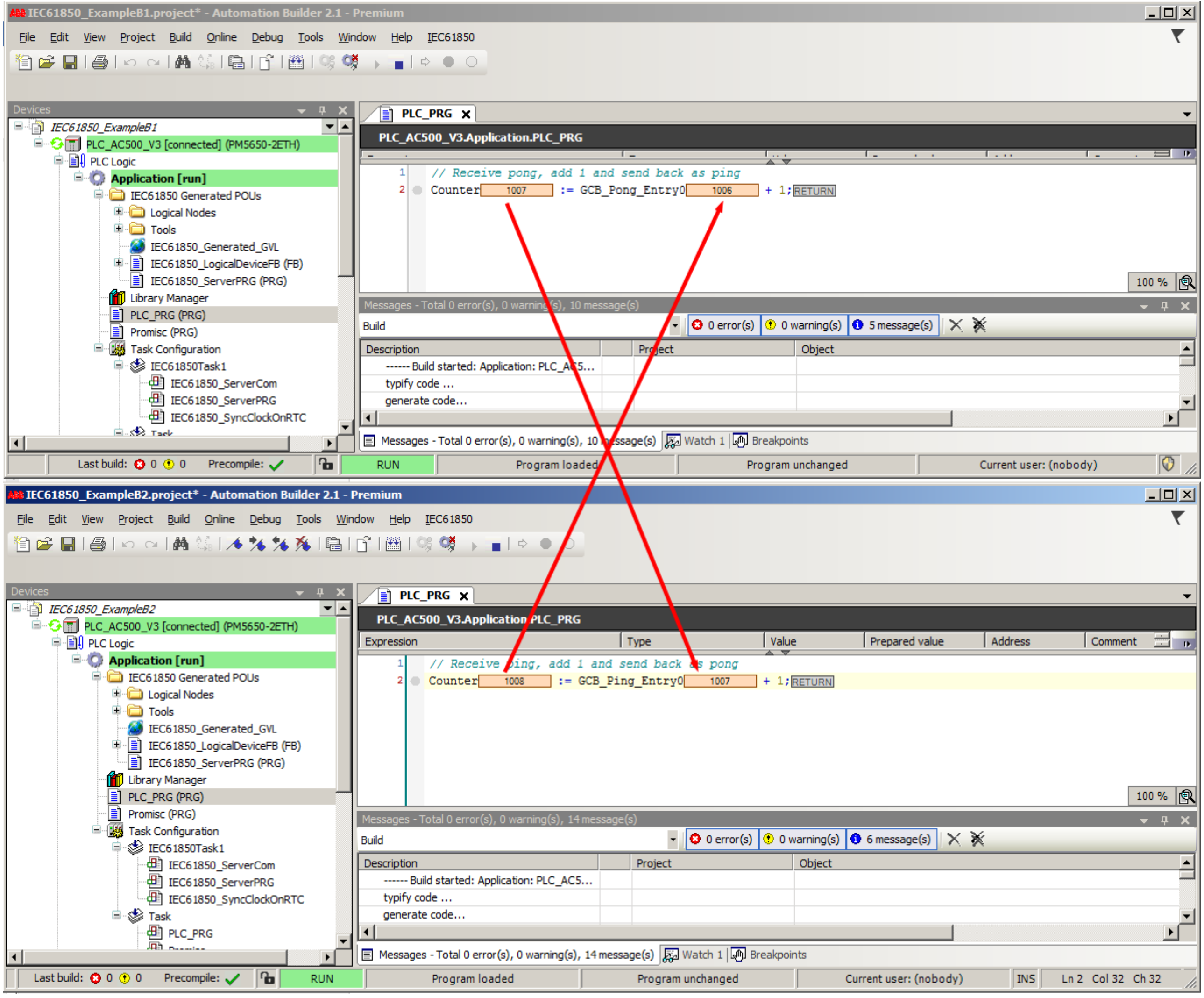
Now the Ping-Pong communication between the two PLCs is configured.

4.2 Close the Ping-Pong loop and run

To close the Ping-Pong loop a 1-line application needs to be created which receives the Ping and writes the Pong (through the variable counter) and vice-versa:

Now both programs can be compiled, downloaded and set to “Run”

The Ping-Pong can be monitored by opening the two projects with 2 Automation Builder:



```
// Receive pong, add 1 and send back as ping
Counter 1007 := GCB_Pong_Entry0 1006 + 1;RETURN
```

Messages - Total 0 error(s), 0 warning(s), 10 message(s)

Build | 0 error(s) | 0 warning(s) | 5 message(s)

Description	Project	Object
------ Build started: Application: PLC_AC5...		
typify code ...		
generate code...		

Last build: 0 0 Precompile: ✓ RUN Program loaded Program unchanged Current user: (nobody)

```
// Receive ping, add 1 and send back as pong
Counter 1008 := GCB_Ping_Entry0 1007 + 1;RETURN
```

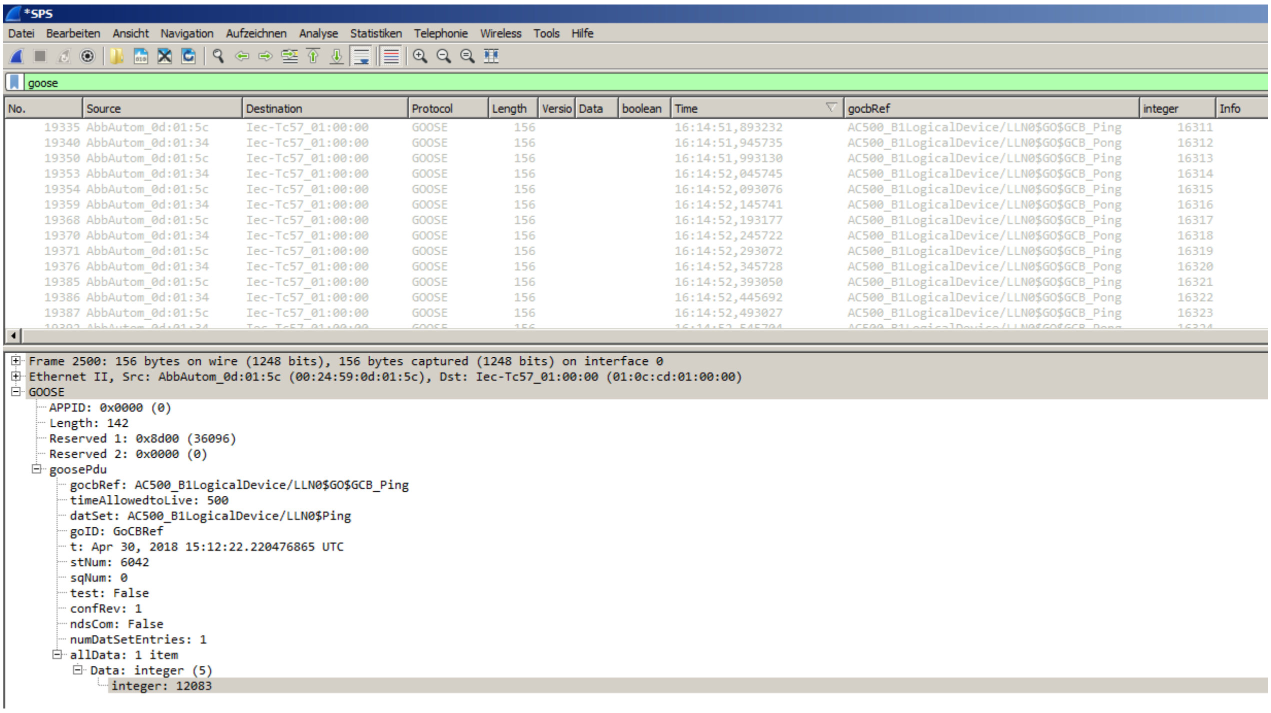
Messages - Total 0 error(s), 0 warning(s), 14 message(s)

Build | 0 error(s) | 0 warning(s) | 6 message(s)

Description	Project	Object
------ Build started: Application: PLC_AC5...		
typify code ...		
generate code...		

Last build: 0 0 Precompile: ✓ RUN Program loaded Program unchanged Current user: (nobody) INS Ln 2 Col 32 Ch 32

By using a sniffer program like Wireshark the GOOSE messages on the Ethernet can be observed after the firewall is configured accordingly:



goose

No.	Source	Destination	Protocol	Length	Versio	Data	boolean	Time	gocbRef	integer	Info
19335	AbbAutom_0d:01:5c	Iec-Tc57_01:00:00	GOOSE	156				16:14:51,893232	AC500_B1LogicalDevice/LLN0$GO$GCB_Ping	16311	
19340	AbbAutom_0d:01:34	Iec-Tc57_01:00:00	GOOSE	156				16:14:51,945735	AC500_B1LogicalDevice/LLN0$GO$GCB_Pong	16312	
19350	AbbAutom_0d:01:5c	Iec-Tc57_01:00:00	GOOSE	156				16:14:51,993130	AC500_B1LogicalDevice/LLN0$GO$GCB_Ping	16313	
19353	AbbAutom_0d:01:34	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,045745	AC500_B1LogicalDevice/LLN0$GO$GCB_Pong	16314	
19354	AbbAutom_0d:01:5c	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,093076	AC500_B1LogicalDevice/LLN0$GO$GCB_Ping	16315	
19359	AbbAutom_0d:01:34	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,145741	AC500_B1LogicalDevice/LLN0$GO$GCB_Pong	16316	
19368	AbbAutom_0d:01:5c	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,193177	AC500_B1LogicalDevice/LLN0$GO$GCB_Ping	16317	
19370	AbbAutom_0d:01:34	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,245722	AC500_B1LogicalDevice/LLN0$GO$GCB_Pong	16318	
19371	AbbAutom_0d:01:5c	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,293072	AC500_B1LogicalDevice/LLN0$GO$GCB_Ping	16319	
19376	AbbAutom_0d:01:34	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,345728	AC500_B1LogicalDevice/LLN0$GO$GCB_Pong	16320	
19385	AbbAutom_0d:01:5c	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,393050	AC500_B1LogicalDevice/LLN0$GO$GCB_Ping	16321	
19386	AbbAutom_0d:01:34	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,445692	AC500_B1LogicalDevice/LLN0$GO$GCB_Pong	16322	
19387	AbbAutom_0d:01:5c	Iec-Tc57_01:00:00	GOOSE	156				16:14:52,493027	AC500_B1LogicalDevice/LLN0$GO$GCB_Ping	16323	

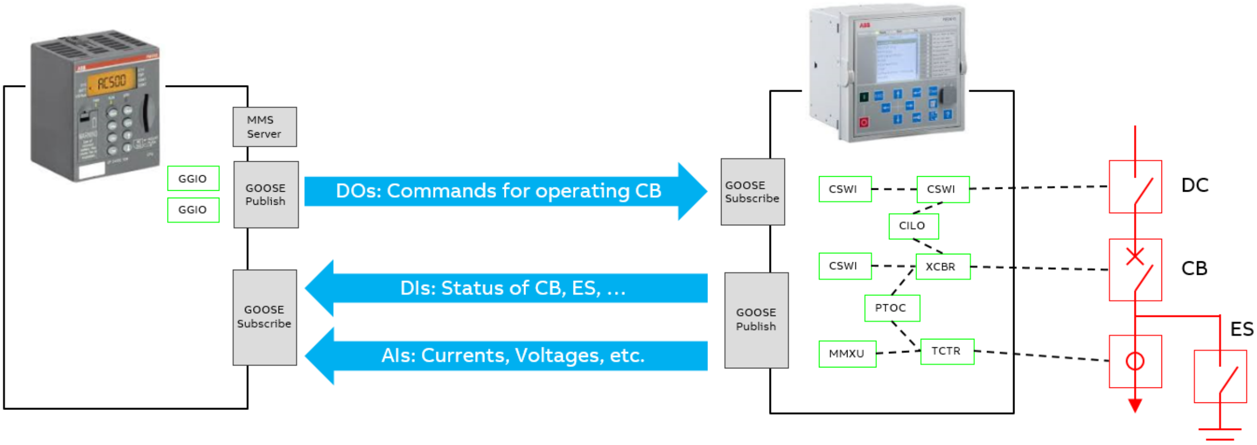
```
⊞ Frame 2500: 156 bytes on wire (1248 bits), 156 bytes captured (1248 bits) on interface 0
⊞ Ethernet II, Src: AbbAutom_0d:01:5c (00:24:59:0d:01:5c), Dst: Iec-Tc57_01:00:00 (01:0c:cd:01:00:00)
⊟ GOOSE
    APPID: 0x0000 (0)
    Length: 142
    Reserved 1: 0x8d00 (36096)
    Reserved 2: 0x0000 (0)
  ⊟ goosePdu
      gocbRef: AC500_B1LogicalDevice/LLN0$GO$GCB_Ping
      timeAllowedtoLive: 500
      datSet: AC500_B1LogicalDevice/LLN0$Ping
      goID: GoCBRef
      t: Apr 30, 2018 15:12:22.220476865 UTC
      stNum: 6042
      sqNum: 0
      test: False
      confRev: 1
      ndsCom: False
      numDatSetEntries: 1
    ⊟ allData: 1 item
      ⊟ Data: integer (5)
          integer: 12083
```

5. EXAMPLE C: AC500 CONTROLLING ABB IED REF615

The example C shows how to configure AC500 for controlling a bay IED like REF615 from ABB.

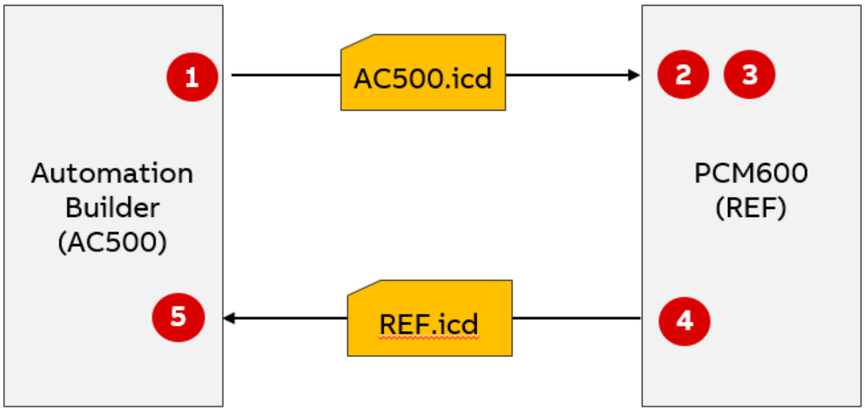
The standard way to do this is with an IEC 61850 MMS client. Since AC500 does not yet have an MMS client the control is done via GOOSE in the following way:

- AC500 sends open and close commands via GOOSE control blocks to the REF615.
- REF615 receives the commands and processes them in the internal circuit breaker block. This needs additional configuration in the REF615 engineering tool PCM600
- Feedback from REF615 including circuit breaker status and other data like currents voltages are sent back to AC500 via GOOSE



Physical connections: Ethernet switch between AC500 (192.168.2.10 on ETH2) and REF615 (192.168.2.9 on RearPort) and the PC with Automation Builder and optionally Wireshark for monitoring the Ethernet traffic.

The engineering is done in 5 steps:



1. Automation Builder: Create switch commands as GOOSE report control block and export AC500.icd
2. PCM600, AC500 part: Import AC500.icd and map to REF615 in order to subscribe the report
3. PCM600, REF615 part: Connect AC500 signals to circuit breaker logic
4. PCM600, IEC 61850 configuration: Create dataset and GOOSE control block with monitoring information and export REF.icd file
5. Automation Builder: Import REF.icd and map variables to 61131 application

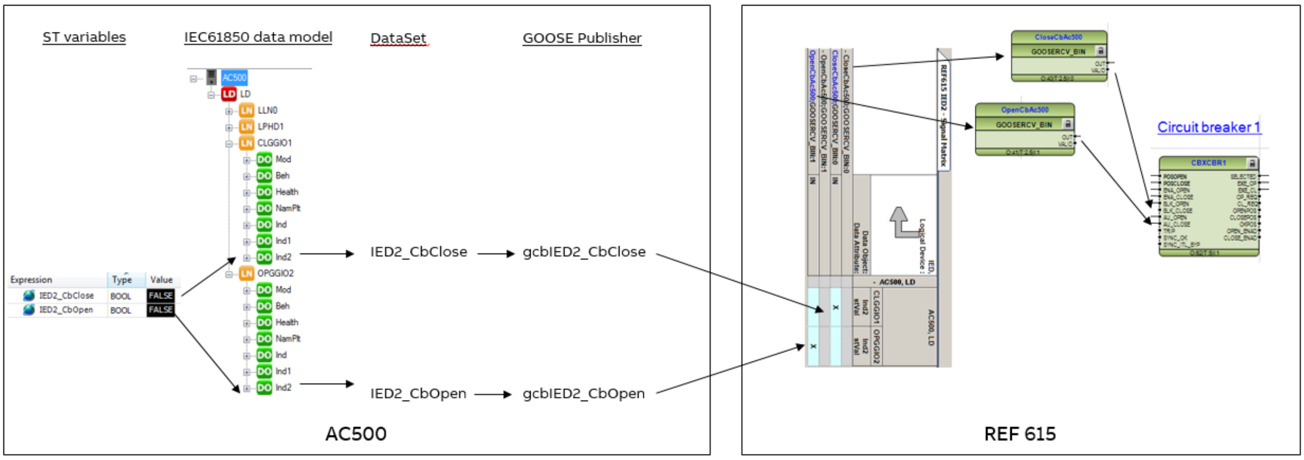
Detailed steps are explained in the following:

5.1 Configure Goose commands from AC500 to REF615

Goal is to send two switch commands from AC500 variables to the REF615 circuit breaker block:

- IED2_CB_Close to input AU_CLOSE of CBXCBR1
- IED2_CB_Open to input AU_OPEN of CBXCBR1

The following picture shows a simplified signal chain over GOOSE:



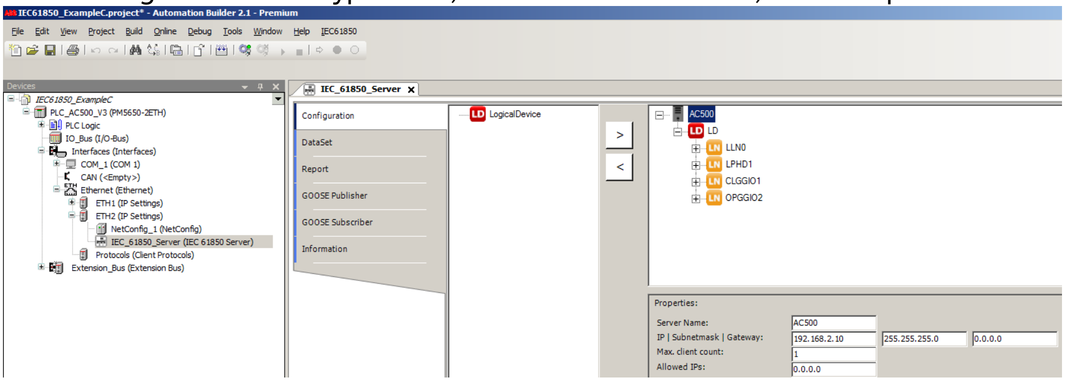
On AC500 side the signals are mapped to IEC 61850 data model (Logical Node GGIO, Data object Ind), packed into datasets and sent as Goose control blocks.

On REF615 side the GOOSE signals are subscribed, mapped to the GOOSE_RCV_BIN blocks and connected to the Auxilliary inputs of the Circuit breaker CBXCBR1.

1. Automation Builder: Create switch commands as GOOSE report control block

   a. Create project AC500_C, insert AC500 V3, insert IEC61850 Server “AC500” with IP address 192.168.2.10, connect to the PLC AC500

   b. Add two logical nodes of type GGIO, one for Close: CLGGIO1, one for Open: OPGGIO2



Properties:

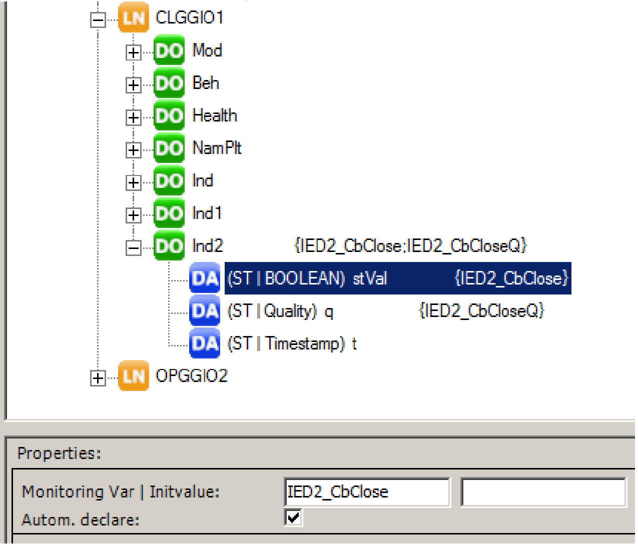
Server Name: AC500

IP | Subnetmask | Gateway: 192.168.2.10 | 255.255.255.0 | 0.0.0.0

Max. client count: 1

Allowed IPs: 0.0.0.0

c. Add DO Ind and map its data attributes stVal and q to local variables:



LN CLGGIO1
- DO Mod
- DO Beh
- DO Health
- DO NamPlt
- DO Ind
- DO Ind1
- DO Ind2 {IED2_CbClose;IED2_CbCloseQ}
  - DA (ST | BOOLEAN) stVal {IED2_CbClose}
  - DA (ST | Quality) q {IED2_CbCloseQ}
  - DA (ST | Timestamp) t

LN OPGGIO2

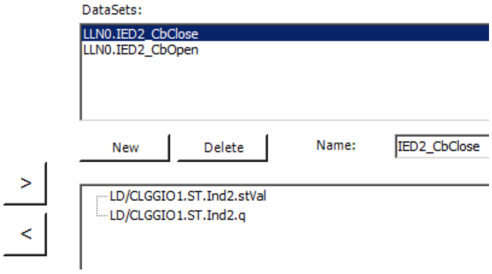
Properties:

Monitoring Var | Initvalue: IED2_CbClose

Autom. declare: ☑

d. Do the same for OPGGIO2

e. Create two datasets:

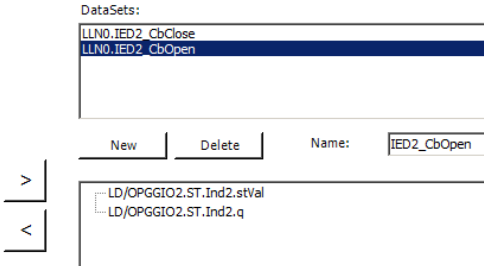


DataSets:

LLN0.IED2_CbClose
LLN0.IED2_CbOpen

Name: IED2_CbClose

LD/CLGGIO1.ST.Ind2.stVal
LD/CLGGIO1.ST.Ind2.q



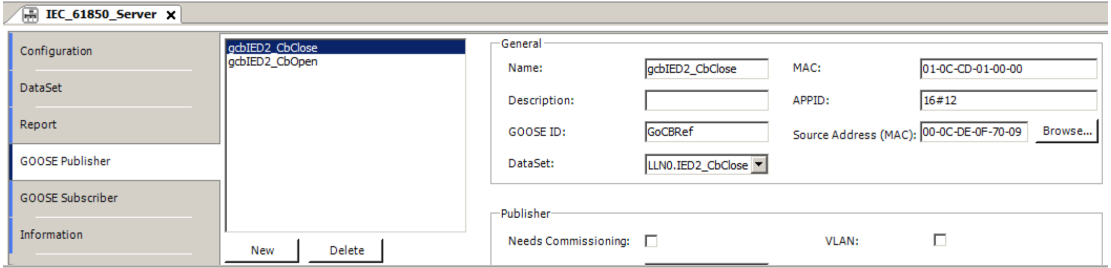
DataSets:

LLN0.IED2_CbClose
LLN0.IED2_CbOpen

Name: IED2_CbOpen

LD/OPGGIO2.ST.Ind2.stVal
LD/OPGGIO2.ST.Ind2.q

f. Create two GOOSE control blocks.
Set the correct Source Address (MAC) of the interface to REF: ETH2 in this example.
Ensure that the APPIDs are unique!



gcbIED2_CbClose
gcbIED2_CbOpen

General

Name: gcbIED2_CbClose

Description:

GOOSE ID: GoCBRef

DataSet: LLN0.IED2_CbClose

MAC: 01-0C-CD-01-00-00

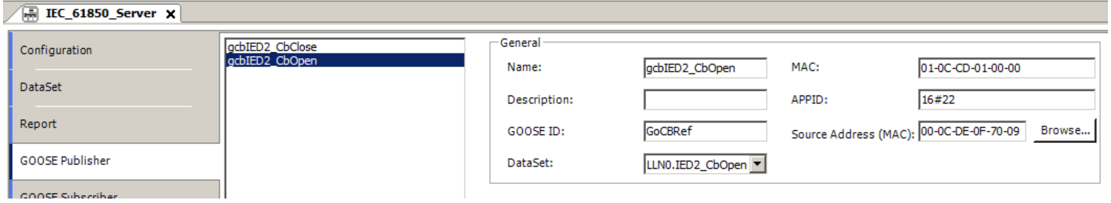
APPID: 16#12

Source Address (MAC): 00-0C-DE-0F-70-09

Publisher

Needs Commissioning: □

VLAN: □



gcbIED2_CbClose
gcbIED2_CbOpen

General

Name: gcbIED2_CbOpen

Description:

GOOSE ID: GoCBRef

DataSet: LLN0.IED2_CbOpen

MAC: 01-0C-CD-01-00-00

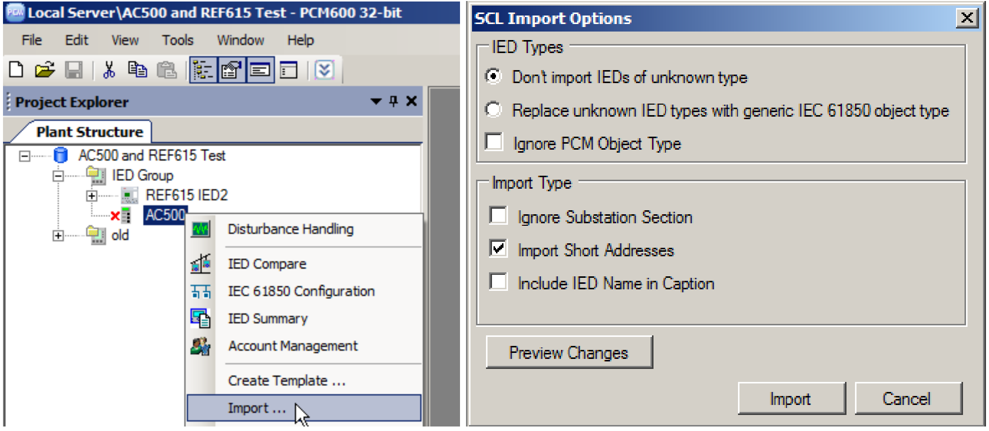
APPID: 16#22

Source Address (MAC): 00-0C-DE-0F-70-09

g. Export AC500.icd

2. PCM600, AC500 part: Import AC500.icd and map to REF615 in order to subscribe the report

   a. PCM600: Create generic IEC 61850 IED, name it “AC500”, Import AC500.icd



SCL Import Options

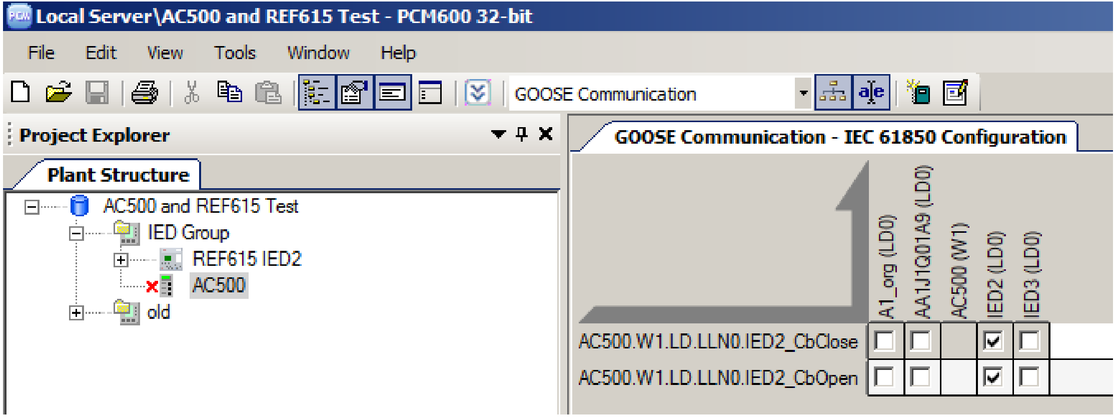
IED Types

● Don't import IEDs of unknown type
○ Replace unknown IED types with generic IEC 61850 object type
□ Ignore PCM Object Type

Import Type

□ Ignore Substation Section
☑ Import Short Addresses
□ Include IED Name in Caption

   b. Open IEC61850 configuration and map the signals from AC500 to the target REF615: IED2 in this example:

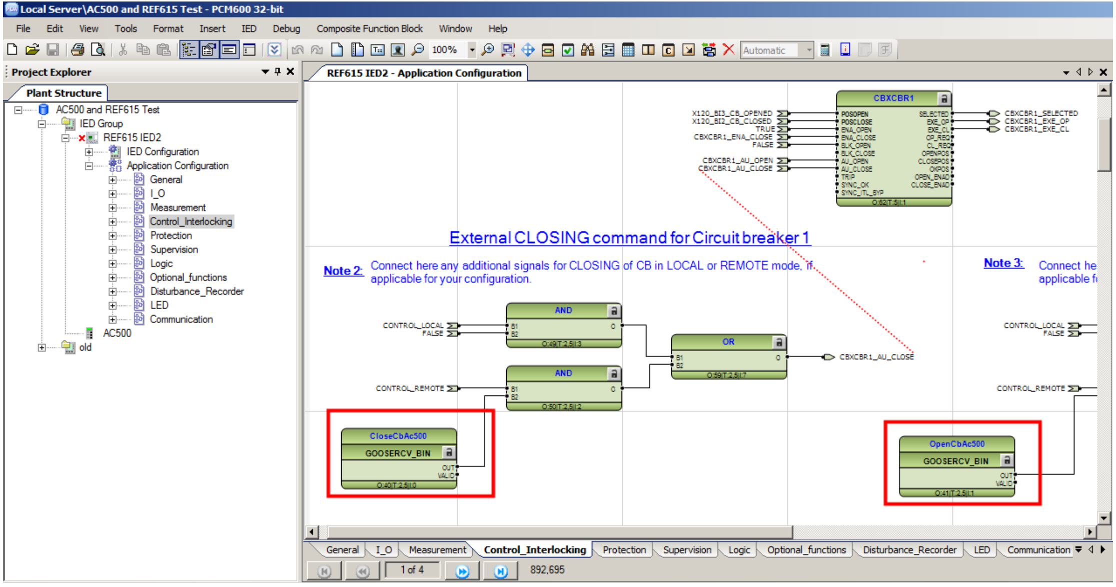


	A1_org (LD0)	AA1J1Q01A9 (LD0)	AC500 (W1)	IED2 (LD0)	IED3 (LD0)
AC500.W1.LD.LLN0.IED2_CbClose	□	□		☑	□
AC500.W1.LD.LLN0.IED2_CbOpen	□	□		☑	□

   Close the IEC 61850 configuration and save changes

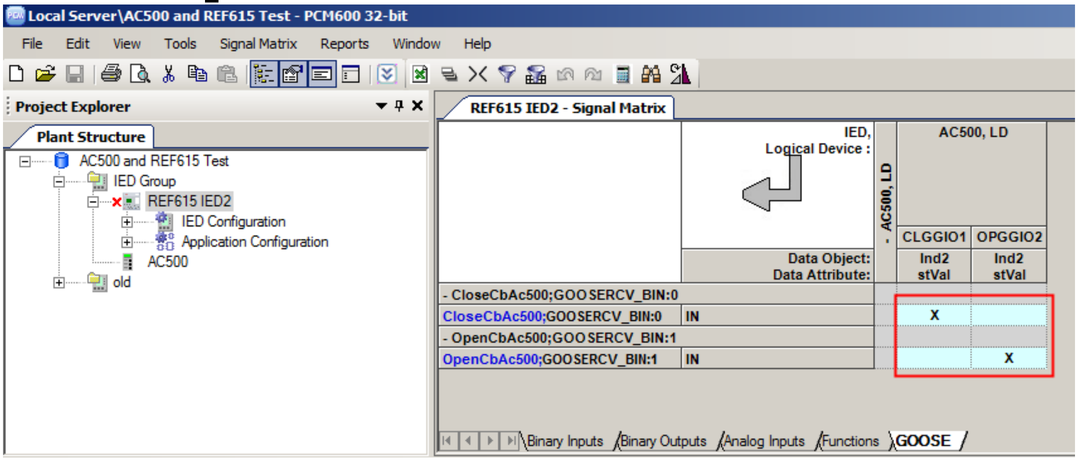
3. PCM600, REF615 part: Connect AC500 signals to circuit breaker logic

   a. Open application configuration of REF615, add two new blocks of the type GOOSERCB_BIN, connect them to the control logic in order to feed the auxiliary inputs of the circuit breaker block CBXCBR1:



   Close the Application configuration and save changes

b. Open the signal matrix tool of REF615 and map the GOOSE signals from AC500 to the new GOOSERCV_BIN blocks.


Local Server\AC500 and REF615 Test - PCM600 32-bit
File Edit View Tools Signal Matrix Reports Window Help
Project Explorer
Plant Structure
AC500 and REF615 Test
IED Group
REF615 IED2
IED Configuration
Application Configuration
AC500
old
REF615 IED2 - Signal Matrix
IED, Logical Device :
AC500, LD
Data Object: Data Attribute:
Binary Inputs
Binary Outputs
Analog Inputs
Functions
GOOSE

		AC500, LD CLGGIO1 Ind2 stVal	AC500, LD OPGGIO2 Ind2 stVal
- CloseCbAc500;GOOSERCV_BIN:0			
CloseCbAc500;GOOSERCV_BIN:0	IN	X	
- OpenCbAc500;GOOSERCV_BIN:1			
OpenCbAc500;GOOSERCV_BIN:1	IN		X

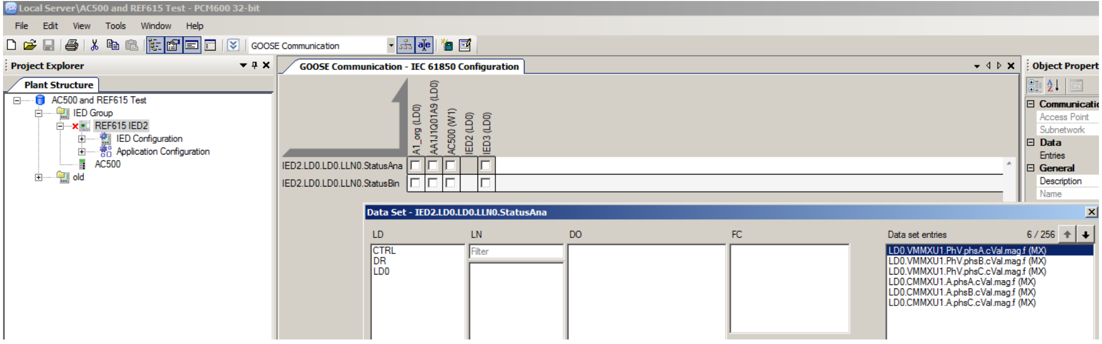
Close the Signal Matrix tool and save changes.

Now the AC500 is ready to send and the REF615 is ready to receive the switch commands over GOOSE.

5.2 Configure Goose feedback from REF615 to AC500

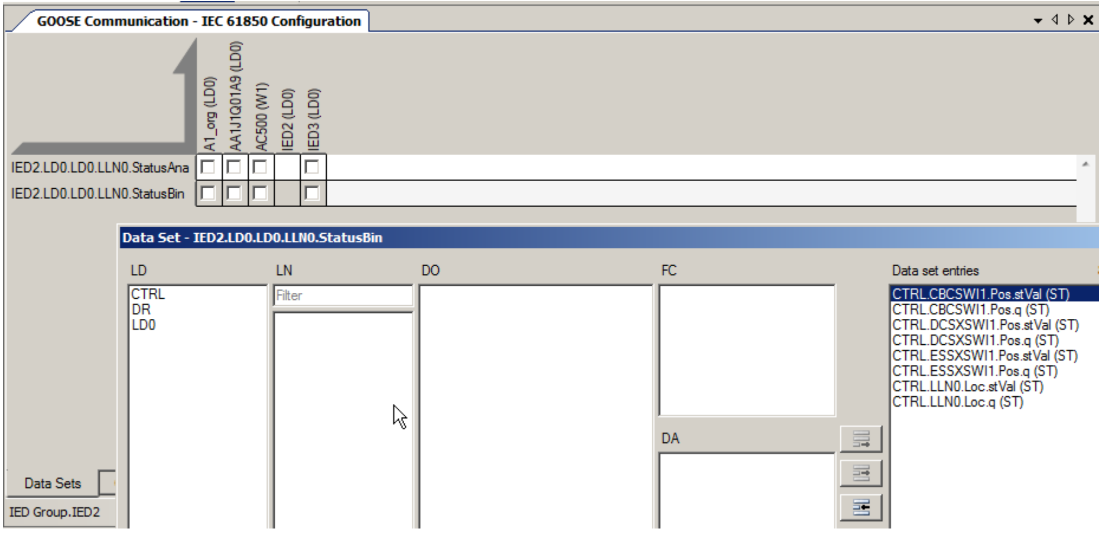
4. PCM600, IEC 61850 configuration: Create dataset and GOOSE control block with monitoring information and export REF.icd file

a. Create new Datasets and fill with data to send from REF to AC500, “StatusAna” for analog values like currents, voltages of all 3 phases:



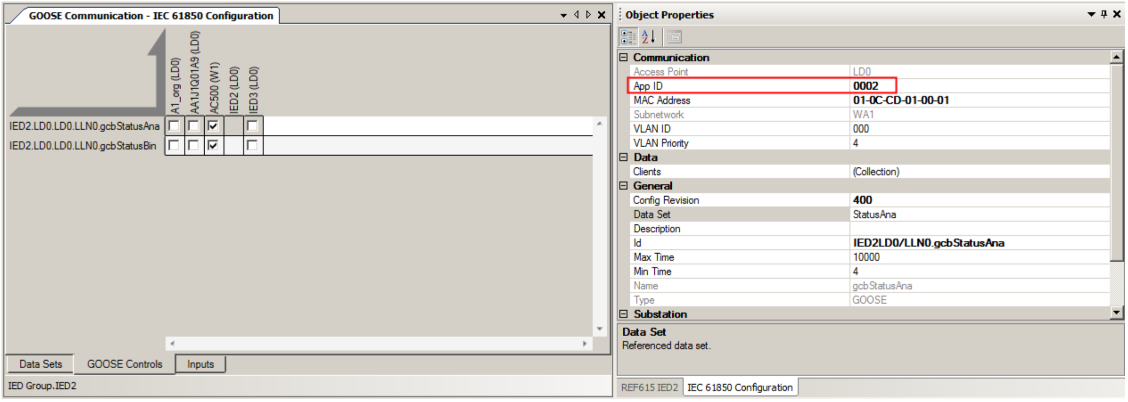
LD0.VMMXU1.PhV.phsA.cVal.mag.f (MX)
LD0.VMMXU1.PhV.phsB.cVal.mag.f (MX)
LD0.VMMXU1.PhV.phsC.cVal.mag.f (MX)
LD0.CMMXU1.A.phsA.cVal.mag.f (MX)
LD0.CMMXU1.A.phsB.cVal.mag.f (MX)
LD0.CMMXU1.A.phsC.cVal.mag.f (MX)

… and “StatusBin” for binary values, like switch position of circuit breaker, disconnector, earthing switch and local/remote status:


GOOSE Communication - IEC 61850 Configuration
A1_org (LD0)
AA1J1Q01A9 (LD0)
AC500 (W1)
IED2 (LD0)
IED3 (LD0)
IED2.LD0.LD0.LLN0.StatusAna
IED2.LD0.LD0.LLN0.StatusBin
Data Set - IED2.LD0.LD0.LLN0.StatusBin
LD
CTRL
DR
LD0
LN
Filter
DO
FC
DA
Data set entries
Data Sets
IED Group.IED2

CTRL.CBCSWI1.Pos.stVal (ST)
CTRL.CBCSWI1.Pos.q (ST)
CTRL.DCSXSWI1.Pos.stVal (ST)
CTRL.DCSXSWI1.Pos.q (ST)
CTRL.ESSXSWI1.Pos.stVal (ST)
CTRL.ESSXSWI1.Pos.q (ST)
CTRL.LLN0.Loc.stVal (ST)
CTRL.LLN0.Loc.q (ST)

b. Create two new GOOSE controls for the new datasets
Ensure that the APPIDs are unique!



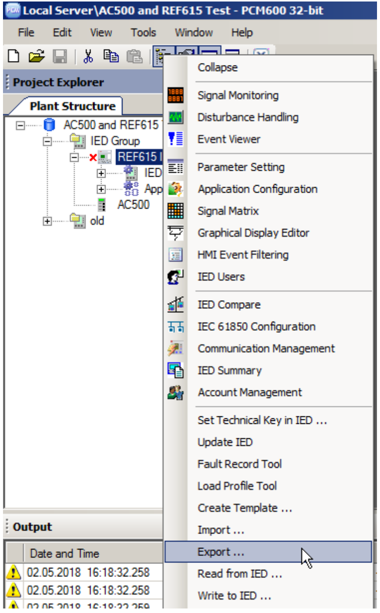
	A1_org (LD0)	AA1J1Q01A9 (LD0)	AC500 (W1)	IED2 (LD0)	IED3 (LD0)
IED2.LD0.LD0.LLN0.gcbStatusAna	□	□	☑		□
IED2.LD0.LD0.LLN0.gcbStatusBin	□	□	☑		□

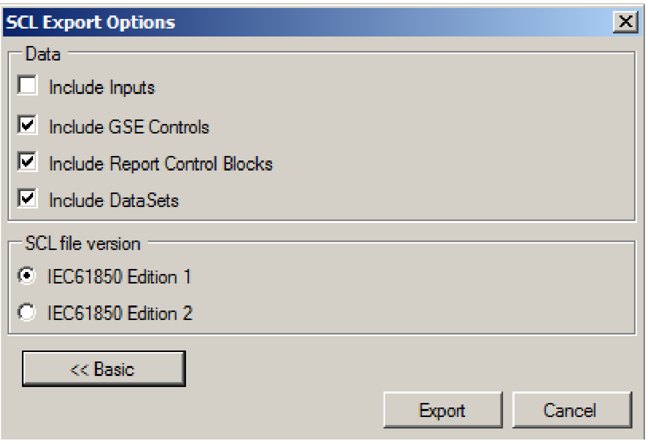
Communication	
Access Point	LD0
App ID	0002
MAC Address	01-0C-CD-01-00-01
Subnetwork	WA1
VLAN ID	000
VLAN Priority	4
Data	
Clients	(Collection)
General	
Config Revision	400
Data Set	StatusAna
Description	
Id	IED2LD0/LLN0.gcbStatusAna
Max Time	10000
Min Time	4
Name	gcbStatusAna
Type	GOOSE
Substation	

Data Set
Referenced data set.

Close the IEC 61850 configuration and save changes

c. Export REF615.icd with default settings:




SCL Export Options

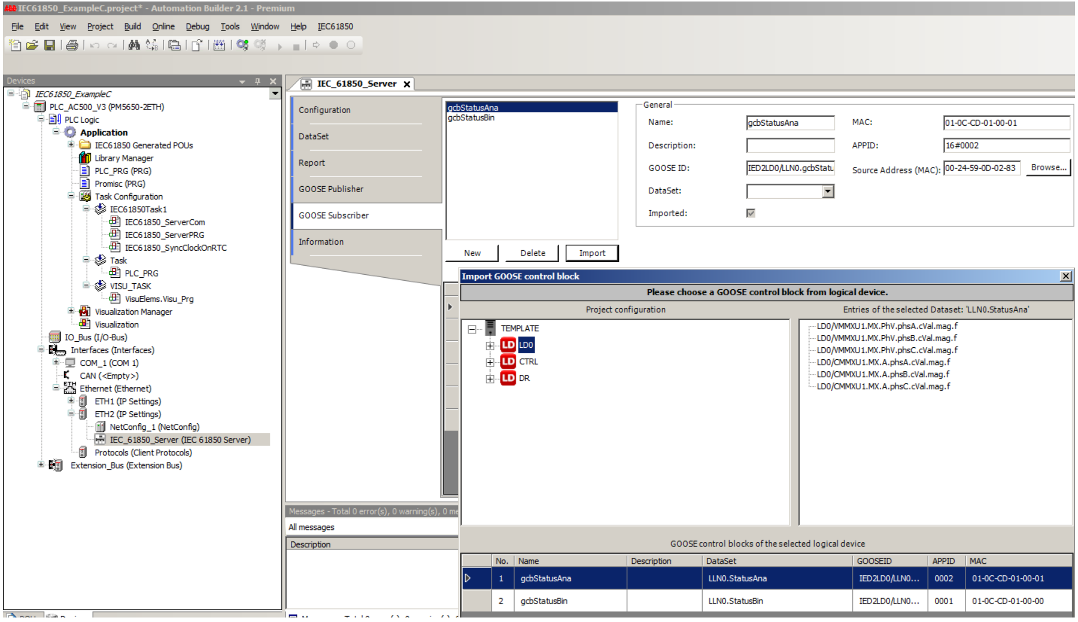
Data
□ Include Inputs
☑ Include GSE Controls
☑ Include Report Control Blocks
☑ Include DataSets

SCL file version
● IEC61850 Edition 1
○ IEC61850 Edition 2

d. Write changes to IED (context menu – Write to IED…)

5. Automation Builder: Import REF615.icd and map variables to 61131 application

   a. Open …ExampleC.project and choose GOOSE Subscriber. Import REF615.icd file and import both datasets from REF615.



IEC61850_ExampleC.project* - Automation Builder 2.1 - Premium
File Edit View Project Build Online Debug Tools Window Help IEC61850
Devices
IEC_61850_Server
Configuration
DataSet
Report
GOOSE Publisher
GOOSE Subscriber
Information
New
Delete
Import
Browse...

gcbStatusAna
gcbStatusBin

General

Name: gcbStatusAna
Description:
GOOSE ID: IED2LD0/LLN0.gcbStatu
DataSet:
Imported: ☑
MAC: 01-0C-CD-01-00-01
APPID: 16#0002
Source Address (MAC): 00-24-59-0D-02-83

Import GOOSE control block

Please choose a GOOSE control block from logical device.

Project configuration: TEMPLATE, LD0, CTRL, DR

Entries of the selected Dataset: 'LLN0.StatusAna'

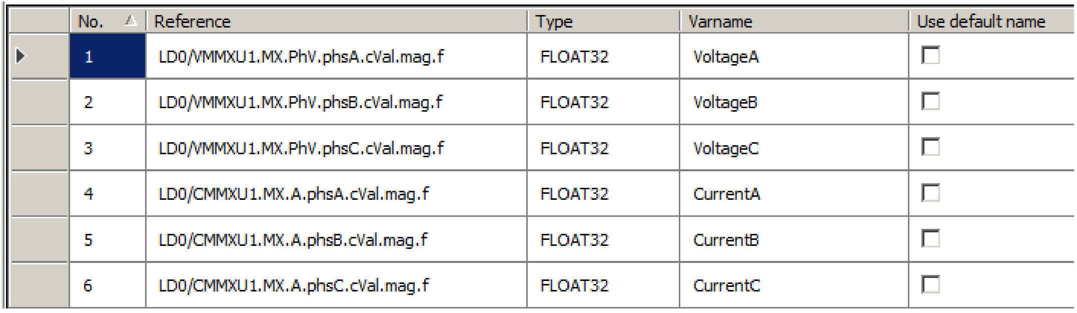
- LD0/VMMXU1.MX.PhV.phsA.cVal.mag.f
- LD0/VMMXU1.MX.PhV.phsB.cVal.mag.f
- LD0/VMMXU1.MX.PhV.phsC.cVal.mag.f
- LD0/CMMXU1.MX.A.phsA.cVal.mag.f
- LD0/CMMXU1.MX.A.phsB.cVal.mag.f
- LD0/CMMXU1.MX.A.phsC.cVal.mag.f

GOOSE control blocks of the selected logical device

No.	Name	Description	DataSet	GOOSEID	APPID	MAC
1	gcbStatusAna		LLN0.StatusAna	IED2LD0/LLN0…	0002	01-0C-CD-01-00-01
2	gcbStatusBin		LLN0.StatusBin	IED2LD0/LLN0…	0001	01-0C-CD-01-00-00

   b. Set the correct Source Address (MAC) of the interface to REF

   c. Assign suitable Varnames:



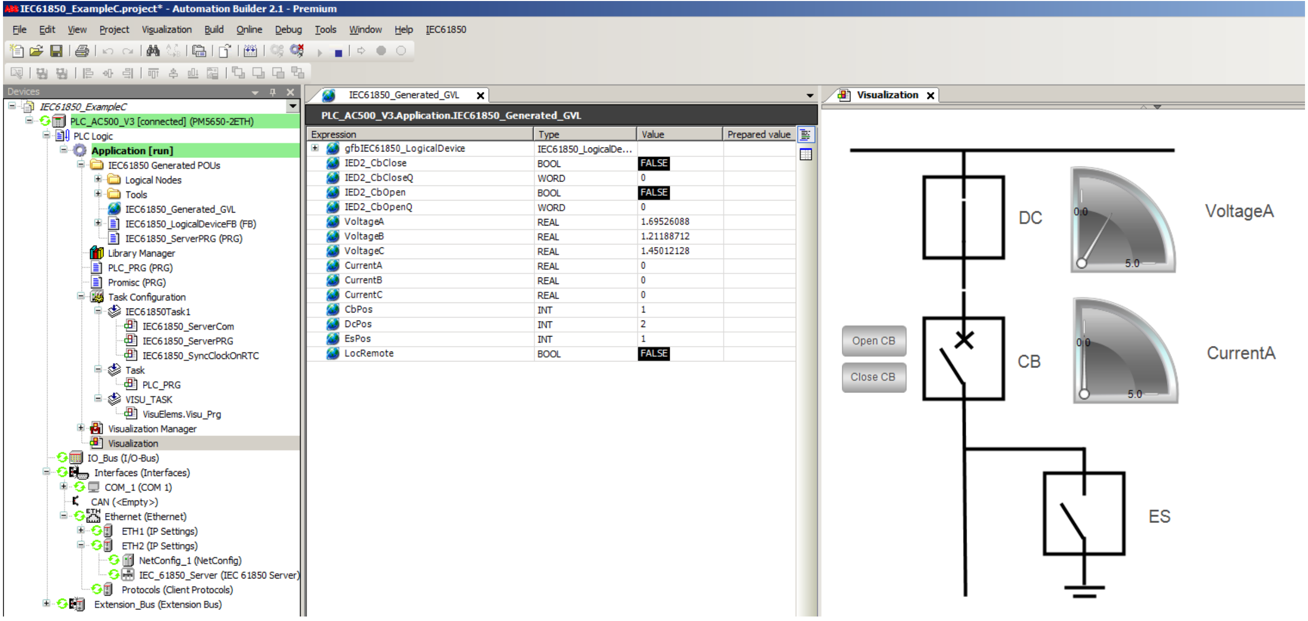
No.	Reference	Type	Varname	Use default name
1	LD0/VMMXU1.MX.PhV.phsA.cVal.mag.f	FLOAT32	VoltageA	☐
2	LD0/VMMXU1.MX.PhV.phsB.cVal.mag.f	FLOAT32	VoltageB	☐
3	LD0/VMMXU1.MX.PhV.phsC.cVal.mag.f	FLOAT32	VoltageC	☐
4	LD0/CMMXU1.MX.A.phsA.cVal.mag.f	FLOAT32	CurrentA	☐
5	LD0/CMMXU1.MX.A.phsB.cVal.mag.f	FLOAT32	CurrentB	☐
6	LD0/CMMXU1.MX.A.phsC.cVal.mag.f	FLOAT32	CurrentC	☐

Now the REF615 is ready to send and the AC500 is ready to receive the feedback signals over GOOSE.

5.3 Run the example

Compile download and set the AC500 to run.

Now the switchgear bay can be controlled with AC500:



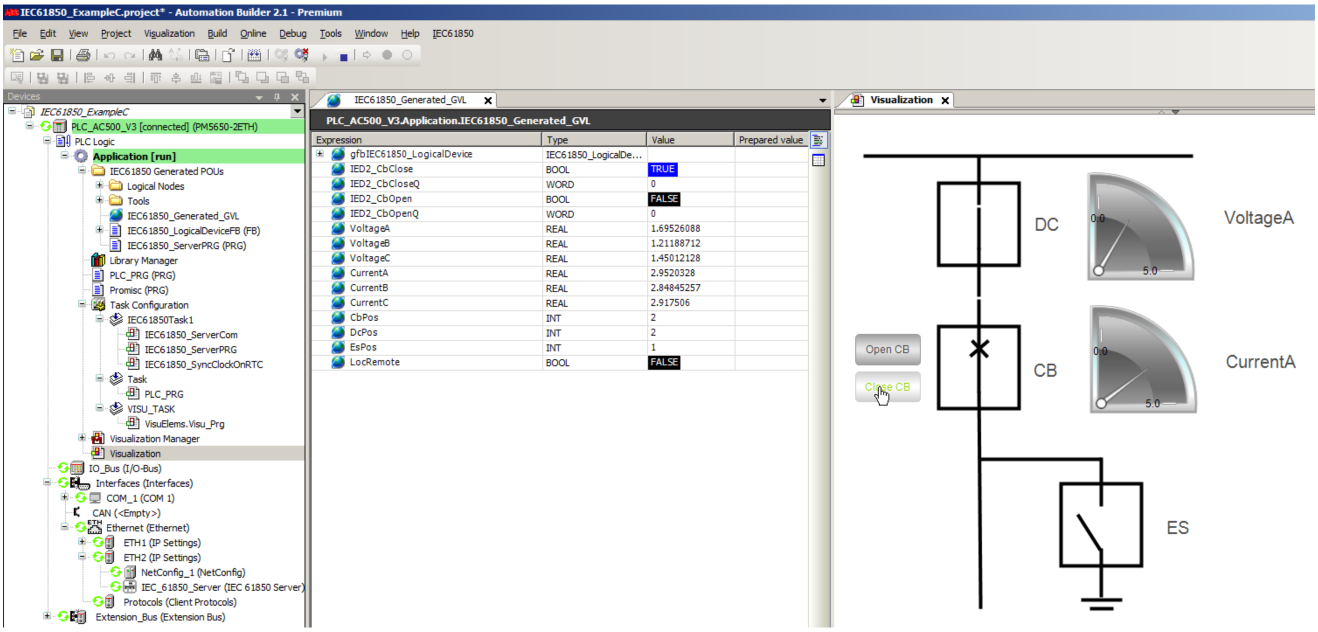
PLC_AC500_V3.Application.IEC61850_Generated_GVL

Expression	Type	Value	Prepared value
gfbIEC61850_LogicalDevice	IEC61850_LogicalDe...		
IED2_CbClose	BOOL	FALSE	
IED2_CbCloseQ	WORD	0	
IED2_CbOpen	BOOL	FALSE	
IED2_CbOpenQ	WORD	0	
VoltageA	REAL	1.69526088	
VoltageB	REAL	1.21188712	
VoltageC	REAL	1.45012128	
CurrentA	REAL	0	
CurrentB	REAL	0	
CurrentC	REAL	0	
CbPos	INT	1	
DcPos	INT	2	
EsPos	INT	1	
LocRemote	BOOL	FALSE	

In the global variables the currents and voltages of all 3 phases are shown. Furthermore the positions of Disconnector (DC), Circuit Breaker (CB) and Earthing Switch (ES) can be monitored: 2 is open, 1 is closed, 0 is intermediate state.

Switch commands can be issued by writing TRUE to IED2_CbClose or IED2_CbOpen.

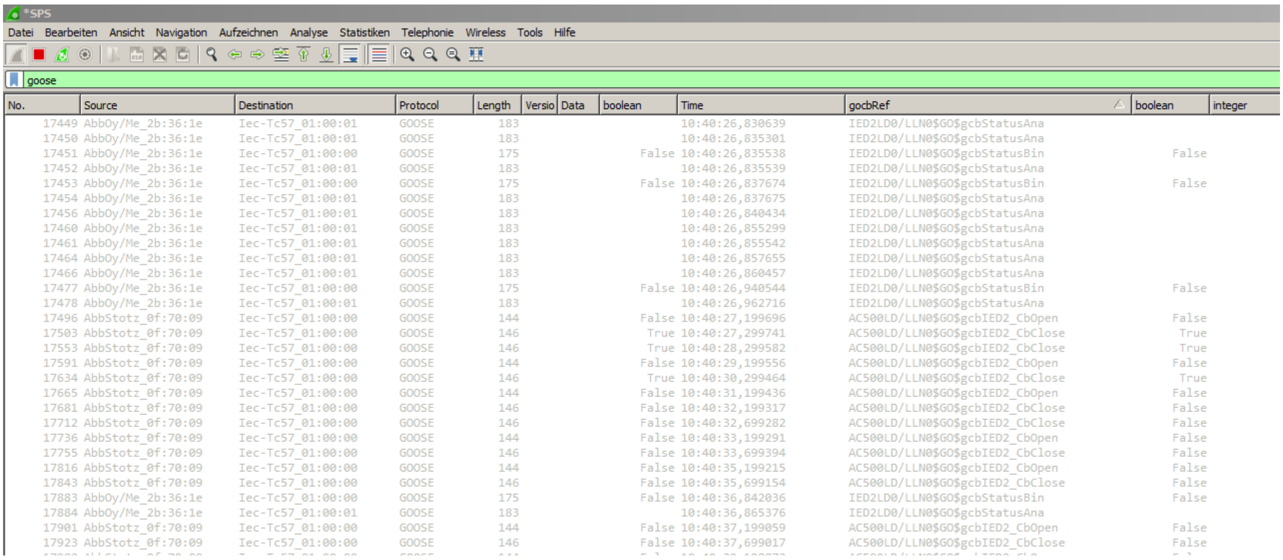
All these variables are connected to a simple visualization which shows the single line of the switchgear bay including buttons for Open and Close the circuit breaker:



PLC_AC500_V3.Application.IEC61850_Generated_GVL

Expression	Type	Value	Prepared value
gfbIEC61850_LogicalDevice	IEC61850_LogicalDe...		
IED2_CbClose	BOOL	TRUE	
IED2_CbCloseQ	WORD	0	
IED2_CbOpen	BOOL	FALSE	
IED2_CbOpenQ	WORD	0	
VoltageA	REAL	1.69526088	
VoltageB	REAL	1.21188712	
VoltageC	REAL	1.45012128	
CurrentA	REAL	2.9520328	
CurrentB	REAL	2.84845257	
CurrentC	REAL	2.917506	
CbPos	INT	2	
DcPos	INT	2	
EsPos	INT	1	
LocRemote	BOOL	FALSE	

The GOOSE traffic between the two IEDs can be monitored with Wireshark:


*SPS
Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telephonie Wireless Tools Hilfe
goose

No.	Source	Destination	Protocol	Length	Versio	Data	boolean	Time	gocbRef	boolean	integer
17449	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,830639	IED2LD0/LLN0$GO$gcbStatusAna		
17450	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,835301	IED2LD0/LLN0$GO$gcbStatusAna		
17451	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:00	GOOSE	175			False	10:40:26,835538	IED2LD0/LLN0$GO$gcbStatusBin	False	
17452	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,835539	IED2LD0/LLN0$GO$gcbStatusAna		
17453	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:00	GOOSE	175			False	10:40:26,837674	IED2LD0/LLN0$GO$gcbStatusBin	False	
17454	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,837675	IED2LD0/LLN0$GO$gcbStatusAna		
17456	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,840434	IED2LD0/LLN0$GO$gcbStatusAna		
17460	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,855299	IED2LD0/LLN0$GO$gcbStatusAna		
17461	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,855542	IED2LD0/LLN0$GO$gcbStatusAna		
17464	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,857655	IED2LD0/LLN0$GO$gcbStatusAna		
17466	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,860457	IED2LD0/LLN0$GO$gcbStatusAna		
17477	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:00	GOOSE	175			False	10:40:26,940544	IED2LD0/LLN0$GO$gcbStatusBin	False	
17478	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:26,962716	IED2LD0/LLN0$GO$gcbStatusAna		
17496	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	144			False	10:40:27,199696	AC500LD/LLN0$GO$gcbIED2_CbOpen	False	
17503	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	146			True	10:40:27,299741	AC500LD/LLN0$GO$gcbIED2_CbClose	True	
17553	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	146			True	10:40:28,299582	AC500LD/LLN0$GO$gcbIED2_CbClose	True	
17591	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	144			False	10:40:29,199556	AC500LD/LLN0$GO$gcbIED2_CbOpen	False	
17634	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	146			True	10:40:30,299464	AC500LD/LLN0$GO$gcbIED2_CbClose	True	
17665	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	144			False	10:40:31,199436	AC500LD/LLN0$GO$gcbIED2_CbOpen	False	
17681	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	146			False	10:40:32,199317	AC500LD/LLN0$GO$gcbIED2_CbClose	False	
17712	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	146			False	10:40:32,699282	AC500LD/LLN0$GO$gcbIED2_CbClose	False	
17736	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	144			False	10:40:33,199291	AC500LD/LLN0$GO$gcbIED2_CbOpen	False	
17755	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	146			False	10:40:33,699394	AC500LD/LLN0$GO$gcbIED2_CbClose	False	
17816	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	144			False	10:40:35,199215	AC500LD/LLN0$GO$gcbIED2_CbOpen	False	
17843	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	146			False	10:40:35,699154	AC500LD/LLN0$GO$gcbIED2_CbClose	False	
17883	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:00	GOOSE	175			False	10:40:36,842036	IED2LD0/LLN0$GO$gcbStatusBin	False	
17884	AbbOy/Me_2b:36:1e	Iec-Tc57_01:00:01	GOOSE	183				10:40:36,865376	IED2LD0/LLN0$GO$gcbStatusAna		
17901	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	144			False	10:40:37,199059	AC500LD/LLN0$GO$gcbIED2_CbOpen	False	
17923	AbbStotz_0f:70:09	Iec-Tc57_01:00:00	GOOSE	146			False	10:40:37,699017	AC500LD/LLN0$GO$gcbIED2_CbClose	False	

6. EXPERT FEATURES

6.1 Bulk data engineering

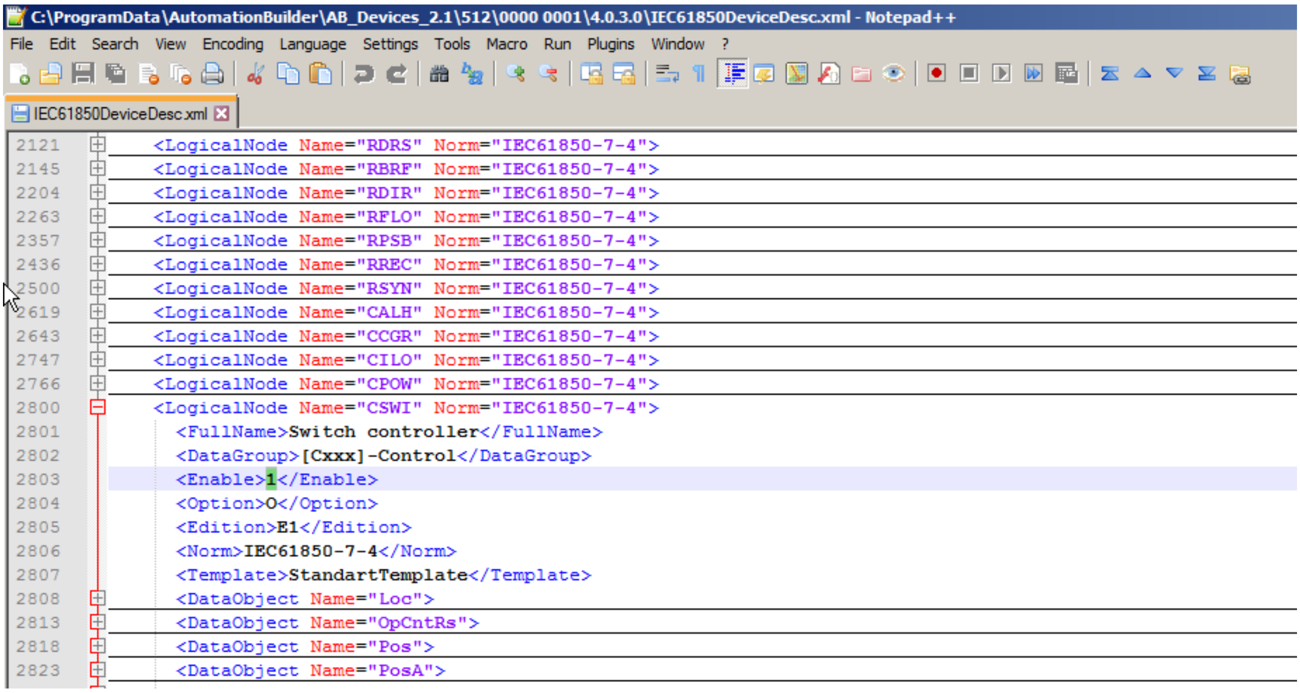
- Create basic IEC 61850 server configuration
- Menu - IEC61850 – Export Server – Save as type XML
- Enhance XML file by copying logical nodes, reports etc.
- Menu – IEC61850 – Import Server, type XML → Configuration is updated

6.2 Add new logical node types

Logical nodes are stored in the device description file:

C:\ProgramData\AutomationBuilder\AB_Devices_2.1\512\0000 0001\4.0.3.0\IEC61850DeviceDesc.xml

This file can be edited in order to enable prepared Logical nodes or to create new ones.



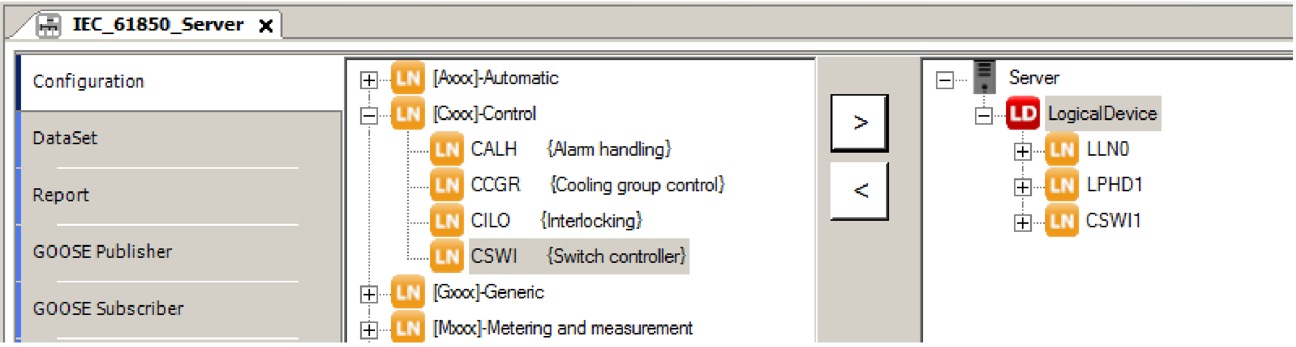
C:\ProgramData\AutomationBuilder\AB_Devices_2.1\512\0000 0001\4.0.3.0\IEC61850DeviceDesc.xml - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
IEC61850DeviceDesc.xml

```
  <LogicalNode Name="RDRS" Norm="IEC61850-7-4">
  <LogicalNode Name="RBRF" Norm="IEC61850-7-4">
  <LogicalNode Name="RDIR" Norm="IEC61850-7-4">
  <LogicalNode Name="RFLO" Norm="IEC61850-7-4">
  <LogicalNode Name="RPSB" Norm="IEC61850-7-4">
  <LogicalNode Name="RREC" Norm="IEC61850-7-4">
  <LogicalNode Name="RSYN" Norm="IEC61850-7-4">
  <LogicalNode Name="CALH" Norm="IEC61850-7-4">
  <LogicalNode Name="CCGR" Norm="IEC61850-7-4">
  <LogicalNode Name="CILO" Norm="IEC61850-7-4">
  <LogicalNode Name="CPOW" Norm="IEC61850-7-4">
  <LogicalNode Name="CSWI" Norm="IEC61850-7-4">
    <FullName>Switch controller</FullName>
    <DataGroup>[Cxxx]-Control</DataGroup>
    <Enable>1</Enable>
    <Option>O</Option>
    <Edition>E1</Edition>
    <Norm>IEC61850-7-4</Norm>
    <Template>StandartTemplate</Template>
    <DataObject Name="Loc">
    <DataObject Name="OpCntRs">
    <DataObject Name="Pos">
    <DataObject Name="PosA">
```

In this example the <Enable> tag of CSWI is set from 0 to 1.

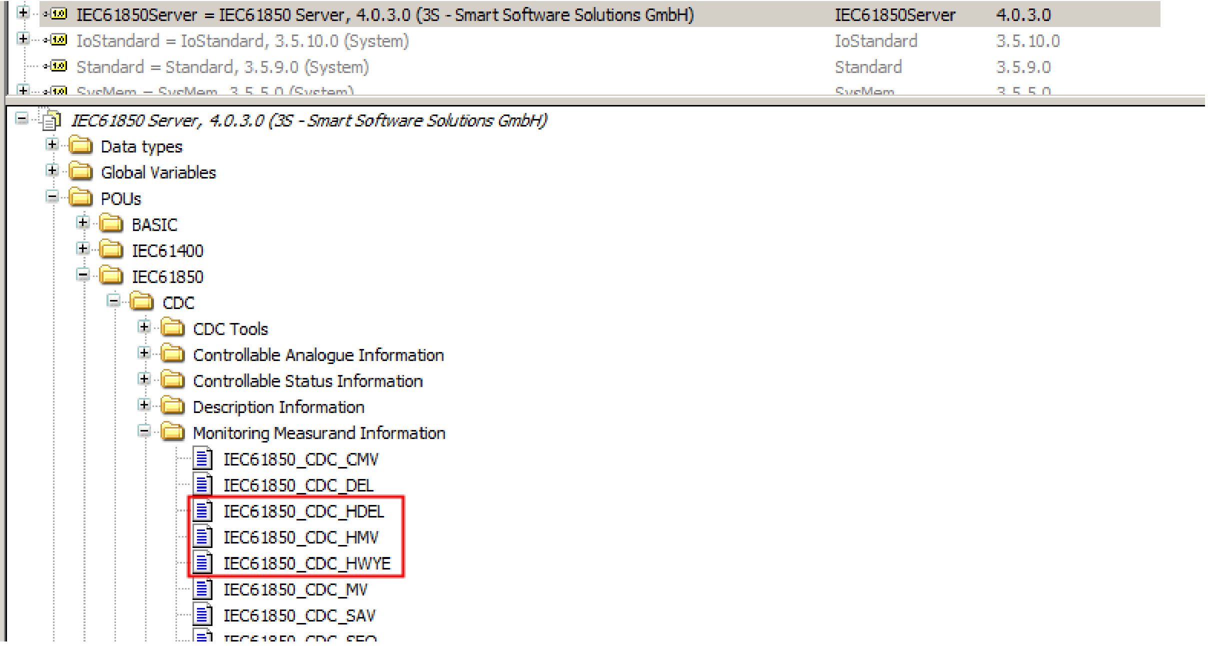
After restart of AB the CSWI can be chosen:



Limitation of Logical nodes which are not enabled by default:

- Their compliance to IEC 61850-7-4 (Ed1) is not fully tested

- Following Common Data Classes are not fully implemented:



```
IEC61850Server = IEC61850 Server, 4.0.3.0 (3S - Smart Software Solutions GmbH)   IEC61850Server   4.0.3.0
IoStandard = IoStandard, 3.5.10.0 (System)                                       IoStandard       3.5.10.0
Standard = Standard, 3.5.9.0 (System)                                            Standard         3.5.9.0
SysMem = SysMem, 3.5.5.0 (System)                                                SysMem           3.5.5.0

IEC61850 Server, 4.0.3.0 (3S - Smart Software Solutions GmbH)
    Data types
    Global Variables
    POUs
        BASIC
        IEC61400
        IEC61850
            CDC
                CDC Tools
                Controllable Analogue Information
                Controllable Status Information
                Description Information
                Monitoring Measurand Information
                    IEC61850_CDC_CMV
                    IEC61850_CDC_DEL
                    IEC61850_CDC_HDEL
                    IEC61850_CDC_HMV
                    IEC61850_CDC_HWYE
                    IEC61850_CDC_MV
                    IEC61850_CDC_SAV
                    IEC61850_CDC_SEQ
```

They are needed by some (not enabled) Logical Nodes, for example MHAI. Enabling such a Logical Node leads to compilation errors.

REVISION HISTORY

Rev.	Page	Change Description	Date / Initial
-r1	all	First version	2018-04-30 IAMF/AC500/Eg
-r2		Chapter 6.3	
-r3		Updated for AB2.2.0 / FW 3.2.0 / IEC61850 Server 4.0.4 Chapter 4.2 deleted (promiscous mode by firmware) Chapter 2.3 (Limitations) updated	2018-10-12 IAMF/AC500/Eg

ABB Automation Products GmbH
Wallstadter Str. 59
68526 Ladenburg, Germany
Phone: +49 62 21 701 1444
Fax : +49 62 21 701 1382
E-Mail: plc.sales@de.abb.com

www.abb.com/plc

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB AG does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.