L-INX/L-GATE

L-INX™ Automation Server

L-GATE™ Universal Gateway

User Manual

LOYTEC electronics GmbH

Contact

LOYTEC electronics GmbH
Blumengasse 35
1170 Vienna
AUSTRIA/EUROPE
support@loytec.com
http://www.loytec.com

Version 5.3

Document № 88073020

Contents

Abbreviations

100Base-T .... 100 Mbps Ethernet network with RJ-45 plug
Aggregation .... Collection of several CEA-709 packets into a single CEA-852 packet
AST .... Alarming, Scheduling, Trending
BACnet .... Building Automation and Control Network
BBMD .... BACnet Broadcast Management Device
BDT .... Broadcast Distribution Table
BOOTP .... Bootstrap Protocol, RFC 1497
CA .... Certification Authority
CEA-709 .... Protocol standard for LONWORKS networks
CEA-852 .... Protocol standard for tunneling CEA-709 packets over IP channels
CN .... Control Network
COV .... change-of-value
CR .... Channel Routing
CS .... Configuration Server that manages CEA-852 IP devices
DA .... Data Access (Web service)
DHCP .... Dynamic Host Configuration Protocol, RFC 2131, RFC 2132
DIF, DIFE .... Data Information Field, Data Information Field Extension
DL .... Data Logger (Web service)
DNS .... Domain Name Server, RFC 1034
DST .... Daylight Saving Time
EEP .... EnOcean Equipment Profile
GMT .... Greenwich Mean Time
IP .... Internet Protocol
IP-852 .... logical IP channel that tunnels CEA-709 packets according CEA-852
LAN .... Local Area Network
LSD Tool .... LOYTEC System Diagnostics Tool
MAC .... Media Access Control
MD5 .... Message Digest 5, a secure hash function, see Internet RFC 1321
M-Bus .... Meter-Bus (Standards EN 13757-2, EN 13757-3)
MIB .... Management Information Base
MS/TP .... Master/Slave Token Passing (this is a BACnet data link layer)
NAT .... Network Address Translation, see Internet RFC 1631
NV .... Network Variable
OPC .... Open Process Control
OPC UA .... OPC Unified Architecture
PEM .... Privacy Enhanced Mail
PLC .... Programmable Logic Controller
RNI .... Remote Network Interface
RSTP .... Rapid Spanning Tree Protocol (Standard IEEE 802.1D-2004)
RTT .... Round-Trip Time

RTU .... Remote Terminal Unit
SCPT .... Standard Configuration Property Type
SSH..................................... Secure Shell
SL .... Send List
SMTP .... Simple Mail Transfer Protocol
SNMP .... Simple Network Management Protocol
SNTP .... Simple Network Time Protocol
SSL .... Secure Socket Layer
STP .... Spanning Tree Protocol (Standard IEEE 802.1D)
TLS .... Transport Layer Security
UCPT.................................. User-defined Configuration Property Type
UI... User Interface
UNVT .... User-defined Network Variable Type
UTC .... Universal Time Coordinated
VIF, VIFE........................... Value Information Field, Value Information Field Extension
WLAN .... Wireless LAN
XML .... eXtensible Markup Language

1 Introduction

1.1 Overview

The L-INX product family consists of high performance, reliable and secure network infrastructure components that implement an embedded automation server. The different models of the L-INX family contain a number of components and network technologies. As protocols on the control network side, the L-INX implements access to BACnet, CEA-709, KNX, Modbus, and M-Bus. The BACnet models are BTL-certified B-BC devices.

The L-GATE product family is intended for use as universal gateways. They provide a similar user interface and set of protocols as the L-INX family. The L-GATE provides data access to a defined set of data points, which are mapped from one control network technology to another control network technology. In particular, the CEA-709/BACnet Gateway (LGATE-900) implements mappings between a set of CEA-709 network variables (NVs) and a set of standard BACnet server objects. Which NVs are mapped to BACnet objects can be configured by an LNS plug-in or stand-alone configuration software. The LGATE-900 is equipped with a 100-BaseT Ethernet port (IP), an FT-10 port (CEA-709), and an RS-485 port (MS/TP). The device is fully compliant with ANSI/CEA-709 and ENV14908, ANSI/ASHRAE-135-2004 and ISO 16484. The LGATE-95X extends the 900 model with a number of additional protocol ports and supports more data points for the gateway.

Data from the supported network technologies are available as data points in the automation server. Those data points are freely configurable via configuration software, which provides a fast and easy way to configure the L-INX and L-GATE using online network scans, import/export features or device templates. Data points between different network technologies can be connected to each other for data transfer between those network technologies (gateway). Data points are also subject to alarming, trending and scheduling (AST) functions of the automation server. The usage of math objects allows basic calculations and the built-in E-mail client allows the L-INX and L-GATE to transmit e-mails on certain conditions. Generated alarms can be configured to send e-mails to predefined addresses. Alarms can also be stored in a historical *alarm log.* Trended data collected by the device and is available in CSV format and through a dedicated Web service.

An embedded OPC server exposes a defined set of data points as OPC tags. It implements the OPC XML-DA standard OPC XML-DA 1.01, which lets OPC clients access the data points via Web services. For secure OPC communication some L-INX models add an OPC UA server. Which native data points are exposed to OPC can also be configured by the configuration software. AST objects such as schedules are exposed as a set of OPC tags. Using the supplied L-WEB designer, users can easily generate a Web-based visualization for the L-INX.

Only the L-INX family contains a freely programmable controller that can operate on all L-INX data points. The controller application is developed using the provided IEC-61131 compliant design tool.

The L-INX and the L-GATE permanently collect statistical information from the attached network channels (OPC connections, FT traffic, MS/TP token passing, Ethernet traffic, etc.). Using this data, the device is able to detect problems on these channels (overload, lost tokens, connection problems, etc.) and warns the system operator via LEDs (see Section 4.4). An intuitive user interface allows fast and easy network troubleshooting without any additional analysis tools or deep system knowledge. For troubleshooting Ethernet protocols a local and remote Wireshark packet capture can be configured (see Section 21.3).

The built-in Web server allows convenient device configuration through a standard Web browser such as the Internet Explorer or Firefox. The Web interface also provides statistics information for system installation and network troubleshooting. Some devices also have an LCD display, which provides a quick way to configure basic settings of the device via a jog dial. Also available on some L-INX models are ports for SD card memory and a USB peripheral bus.

Some L-INX and L-GATE models are also equipped with a 2-port Ethernet Switch/Hub. In switched mode an Ethernet daisy chain can be built, which reduces cabling effort. The two Ethernet connectors can also be configured to work as two isolated IP interfaces. This can be used to safely connect a local building network while keeping it isolated from WAN access, that exposes some aspects using secure services (see Section 5.2.5). By using the external L-WLAN adapter, the device also provides a WLAN interface, which can link to an existing access point, set up its own access point or work in a wireless mesh network (see Section 5.2.7).

The L-INX is used for:

- Exposing data of control network devices from different technologies (CEA-709, BACnet, KNX, Modbus, M-Bus) to data points in the automation server,
- Directly connecting I/Os to data points in the automation server,
- exposing data points to OPC tags,
- visualization of data points with the supplied LOYTEC L-WEB software,
- visualization of data points in an OPC XML-DA, UA SCADA package,
- Room control, plant control running IEC61131 programs on data points,
- automatic meter reading applications via M-Bus and Modbus,
- meter gateway to BACnet using pulse counters or M-Bus and Modbus meters,
- browsing data points on the Web interface or LCD display,
- basic automation functions on data points (alarming, trending, scheduling),
- logging alarms,
- sending e-mails on alarms, trend logs, or scheduled events.

The L-GATE is used for:

- Connecting data points between any supported control networks (CEA-709, BACnet, KNX, Modbus, M-Bus) using the universal gateway function,
- browsing data points on the Web interface or LCD display,
- exposing data points to OPC tags,
- basic automation functions on data points (alarming, trending, scheduling),
- logging alarms,

- sending e-mails on alarms, trend logs, or scheduled events,
- gateway for ANSI/CEA-709 network variables (NVs) and configuration properties (CPs),
- gateway for standard (SNVT, SCPT) and user-defined (UNVT, UCPT) types.

1.2 CEA-709.1

L-INX automation server models that have CEA-709 and all L-GATE models are equipped with an FT port (CEA-709) and a 100Base-T Ethernet port (CEA-852). CEA-709 L-INX models come with a router option or an RNI option. L-INX models with the router option contain a CEA-709 router between the FT and the IP-852 channel, which can be configured like an L-IP. It includes a configuration server (CS) to manage the IP-852 channel. The L-INX models without the router option contain a remote network interface (RNI) instead of the router for remote network access. Please refer to Table 1 to learn, which device models have CEA-709 and the router option.

The CEA-709 L-INX and L-GATE device is fully compliant with ANSI/CEA-709, ANSI/CEA-852-A, EN 14908. The CEA-709 node, that is going to be commissioned in the network, is always connected to the FT port of the device.

The function of the CEA-709 node is to expose CEA-709 network variables (NVs) and configuration properties (CPs) to data points in the automation server or the gateway. The configuration software can be run as LNS plug-in or stand-alone. The CEA-709 data points can be bound in the CEA-709 network as NVs or operated as “external NVs”. External NVs are polled or explicitly written to without allocating static or dynamic NVs on the device. In this case, address information is supplied by the configuration software by importing e.g., a CSV file. User-defined network variable types (UNVTs) can be used as dynamic or external NVs. Configuration properties (CPs) on other devices can be accessed through file transfer. To transfer CPs, the device supports both the LONMARK file transfer and the read memory access method. For CPs, the standard SCPTs and user-defined UCPTs are supported. All those CEA-709 data points can be exposed to the automation server or the gateway.

The CEA-709 L-INX with the router option possesses a router between the CEA-852 interface (IP-852) and the FT interface. The CEA-852 interface can be used to connect the L-INX to an IP-based high-speed backbone. The L-INX’s router can be used as a standard CEA-709 configured router or it can be used as a self-learning plug&play router based on the high-performance, well-proven routing core from our L-Switch plug&play multi-port router devices (“smart switch mode”). The self-learning router doesn’t need a network management tool for configuration but is a true plug&play and easy to use IP infrastructure component. For a detailed description of the CEA-709 router’s usage refer to the L-IP User Manual [1].

The L-GATE and the CEA-709 L-INX without the router option can be configured to run either on the CEA-852 interface (IP-852 mode) or on the FT interface (FT mode). In the FT mode, the device provides a remote network interface (RNI), which appears like a LOYTEC NIC-IP is intended to be used together with the LOYTEC NIC software [3]. The RNI can be utilized for remote access and configuration as well as trouble-shooting with the remote LPA. Please consult our product literature for the LPA-IP to learn more about this IP-based CEA-709 protocol analyzer.

The CEA-709 technology in the L-INX and L-GATE allows for:

- Exposing CEA-709 network variables (NVs) and configuration properties (CPs) as data points to the automation server or gateway,
- supporting standard (SNVT, SCPT) and user-defined (UNVT, UCPT) types,
- scheduling CEA-709 network variables,

- generating alarms over the LONMARK node object,
- CEA-709 PC applications (as a CEA-709 network interface),
- remote LPA functionality,
- communicating on CEA-709 with either FT or IP-852 (IP channel on the Intranet/Internet),
- connecting an FT channel to a high-performance backbone using existing IP infrastructure,
- operating as a configuration server for IP-852 devices with the router option.

1.3 BACnet

L-INX automation server models that have BACnet and all L-GATE models are BTL-certified products that implement the B-BC profile. They are equipped with an MS/TP port and a 100Base-T Ethernet port (BACnet/IP). The MS/TP port supports remote Wireshark packet capture for troubleshooting. BACnet L-INX models with the router option also contain a BACnet router between the MS/TP and the BACnet/IP ports, which can be configured like an LIP-ME201. The router models also include a BACnet broadcast management device (BBMD) to manage BACnet/IP internetworks, which span across IP routers. BACnet models without the router can register as a foreign device (FD) with other BBMDs. The device is fully compliant with ANSI/ASHRAE 135-2010 (1.7) and ISO 16484-5. Please refer to Table 1 to learn, which device models have BACnet and the router option.

The BACnet L-INX and the L-GATE expose BACnet server objects and client mappings to data points of the automation server or the gateway. For client mappings, the BACnet address information is supplied by the configuration software by importing e.g., a CSV file or by performing an online network scan.

The BACnet L-INX and L-GATE models also support the LOYTEC Alarming, Scheduling and Trending (AST) features in native BACnet objects. The device provides BACnet scheduler/calendar objects, which can directly schedule BACnet server objects, remote BACnet objects or non-BACnet registers. For alarm conditions the device supports the intrinsic reporting method of BACnet objects. Trend logs can be uploaded from the device via the native BACnet read range.

The BACnet L-INX with the router option possesses a BACnet router between the BACnet/IP port and the MS/TP port. This router can be operated and configured like the LIP-ME200 from LOYTEC. The BACnet/IP interface can be used to connect the L-INX to an IP-based high-speed backbone. The L-INX also can act as a BBMD for a BACnet/IP network. For a detailed description of the BACnet router's usage refer to the LIP-ME201 User Manual [2].

A BACnet L-INX without the router option and the L-GATE can be configured to run either on the BACnet/IP interface or on the MS/TP interface. In BACnet/IP mode, the L-INX can be configured as a foreign device in another BBMD. The BACnet L-INX without the router option does not provide the BBMD functionality itself. The L-GATE can also be configured to be a BBMD. Please refer to Table 1 to learn, which models can be configured as a BBMD.

The BACnet technology in the L-INX and L-GATE allows for:

- Using the device as a BTL-certified B-BC,
- exposing local BACnet server objects (analog, binary, multi-state) and remote objects (client mappings) to data points,

- scheduling any data point from native BACnet schedule and calendar objects,
- trending any data point to native BACnet trend log objects,
- generating native BACnet alarms on any data point,
- communicating with either MS/TP or BACnet/IP,
- connecting an MS/TP network to a high-performance backbone using existing IP infrastructure,
- operation as a BBMD for a BACnet/IP network with the router option,
- troubleshoot the MS/TP network with Wireshark remote packet capture (see Section 21.3),
- distributing NTP time into the BACnet network as BACnet time master (see Section 18.3.8).

1.4 M-Bus

In addition to the basic network technologies all models except the LGATE-900 support the M-Bus interface according to the standards EN 13757-2 and EN 13757-3. To gain access to the M-Bus network, an external M-Bus interface such as the L-MBUS by LOYTEC must be attached to the device. On devices with a serial port, the M-Bus interface is connected to the serial connector. In this case the user needs to turn M-Bus support on and off via a DIP switch. On devices without a serial port, the L-MBUS interface must be used and is connected to the extension port (EXT).

Through the M-Bus interface the L-INX can be used to scan the attached M-Bus network for devices, pull M-Bus data points into a configuration, connect those data points to other technologies and expose M-Bus data points to the automation server. All AST functions can be used directly on M-Bus data points. Especially trending data and polling for data on M-Bus devices has been optimized for automatic meter reading applications.

For debugging purposes a protocol analyzer is included in the firmware and can be operated via the Web-UI and the configuration software. For more information on how to set up the device for using M-Bus, configuring and using M-Bus data points, refer to Chapter 11.

1.5 Modbus

In addition to the basic network technologies the all models except the LGATE-900 support the Modbus RTU and the Modbus TCP interface. To gain access to the Modbus network, the appropriate interfaces have to be activated either in the Web UI or in the configuration software. Modbus RTU is operated with 8N1. A Modbus port can either be operated as Modbus master or Modbus slave.

On some BACnet L-INX models, the Modbus RTU and BACnet MS/TP protocols share the same port. On those models, Modbus RTU can only be used, if BACnet MS/TP is disabled. Please refer to Table 1 to learn, which BACnet L-INX models have this restriction.

Through the Modbus interface the device can be used to data points to other technologies, and expose Modbus data points to OPC tags. All AST functions can be used directly on Modbus data points. Especially trending data and polling for data on Modbus devices has been optimized for automatic meter reading applications.

For debugging purposes a protocol analyzer is included in the firmware and can be operated via the Web UI and the configuration software. For more information on how to set up the device for using Modbus, configuring and using Modbus data points, refer to Chapter 12.

1.6 KNX

In addition to the basic network technologies, some L-INX and L-GATE models can be connected to KNX networks (see Section 1.9). To gain access to a KNX TP1 network, the LKNX-300 interface has to be attached to the device for TP1 networks. All KNX-capable models support KNXnet/IP directly with their Ethernet interface.

The KNX interface allows creating KNX data points which can be used with the AST functions, the OPC server and also the PLC on the L-INX models. The device configuration can be imported from an ETS database export.

For more information on how to set up the device for using KNX, configuring and using KNX data points, refer to Chapter 13.

1.7 EnOcean

In addition to the basic network technologies, some L-INX and L-GATE models can integrate EnOcean wireless devices (see Section 1.9). To gain access to an EnOcean network, the LENO-800 interface has to be attached via one of the USB ports USB 1 or USB 2.

The EnOcean interface is represented in the Configurator as a technology folder. EnOcean devices are created from device templates and provide data points, which can be used with the AST functions, the OPC server and also the PLC on the L-INX models.

For debugging purposes a protocol analyzer is included in the firmware and can be operated via the Web UI. For more information on how to set up the device for using EnOcean, configuring and using EnOcean data points, refer to Chapter 14.

1.8 L-IOB

The L-INX automation server models allow connecting physical I/Os to the device via the L-IOB I/O modules (see Table 1). On some models those modules can be stacked up directly to the L-INX using the LIOB-Connect feature (see Section 1.9). The connected I/O modules are automatically identified and coupled as data points into the L-INX automation server. All L-INX models also support easy integration of L-IOB I/O modules over FT cabling using the LIOB-FT feature, or over Ethernet using the LIOB-IP feature.

L-IOB modules are available with digital inputs and outputs, analog inputs and outputs and universal inputs that are configurable. Some models are also available with a differential pressure sensor.

The I/O modules can be parameterized over the configuration software or the Web UI. All parameterization data is stored on the L-INX and can be reloaded to the LIOB modules when needed. The exchange of modules is detected automatically. For more information on how to setup and use L-IOB I/Os please refer to the L-IOB user manual [7].

1.9 L-INX and L-GATE Models

This Section provides an overview of the different L-INX and L-GATE models in Table 1 and Table 2. This table identifies the different features of those models. Models that possess a certain feature have a check mark (✓**)** in the respective column. If a feature is not available in the particular model, the column is left blank.

L-INX models with the router option have a CEA-709 router or a BACnet router, respectively. The LINX-151 has both a CEA-709 router between FT and IP and a BACnet router between MS/TP and IP.

Features \ Model	LINX-100	LINX-101	LINX-102	LINX-103	LINX-110	LINX-111	LINX-112	LINX-113	LINX-120	LINX-121	LINX-150	LINX-151
CEA-709 Router		✓		✓		✓		✓		✓		✓
CEA-709 RNI	✓		✓		✓		✓		✓		✓	
CEA-709 (FT)	✓[1]	✓	✓[1]	✓	✓[1]	✓	✓[1]	✓	✓[1]	✓	✓[1]	✓
CEA-852 (IP)	✓[1]	✓	✓[1]	✓	✓[1]	✓	✓[1]	✓	✓[1]	✓	✓[1]	✓
BACnet Router												✓
BACnet MS/TP											✓[2]	✓
BACnet IP											✓[2]	✓
BBMD												✓
Modbus RTU	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓[3]	✓[3]
Modbus IP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
M-Bus	✓	✓	✓[4]	✓[4]	✓	✓	✓[4]	✓[4]	✓[4]	✓[4]	✓[4]	✓[4]
KNX TP1			✓[4]	✓[4]			✓[4]	✓[4]	✓[4]	✓[4]	✓[4]	✓[4]
KNX IP			✓	✓			✓	✓	✓	✓	✓	✓
EnOcean			✓[5]	✓[5]			✓[5]	✓[5]	✓[5]	✓[5]	✓[5]	✓[5]
OPC XML-DA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OPC UA			✓	✓			✓	✓	✓	✓	✓	✓
OPC Client			✓	✓			✓	✓	✓	✓	✓	✓
SNMP			✓	✓			✓	✓	✓	✓	✓	✓
PLC (IEC 61131)					✓	✓	✓	✓	✓	✓	✓	✓
LIOB Connect			✓	✓			✓	✓	✓	✓	✓	✓
LIOB FT + IP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LCD Display			✓	✓			✓	✓	✓	✓	✓	✓
Serial Console	✓	✓			✓	✓						
SD Card									✓	✓	✓	✓
USB			✓	✓			✓	✓	✓	✓	✓	✓
Ethernet Switch/Hub			✓	✓			✓	✓	✓	✓	✓	✓
WLAN			✓[5]	✓[5]			✓[5]	✓[5]	✓[5]	✓[5]	✓[5]	✓[5]
SSH, HTTPS, Firewall			✓	✓			✓	✓	✓	✓	✓	✓

[1] This model can be configured to have either FT or IP active for CEA-709.

[2] This model can be configured to have either MS/TP or IP active for BACnet.

[3] Modbus RTU can only be used, if BACnet MS/TP is not active on this model.

[4] M-Bus and KNX TP1 can be used alternatively only on this model. To operate these protocols an expansion module is needed and must be ordered separately.

[5] To operate these protocols an expansion module is needed and must be ordered separately.

Table 1: Available features in different L-INX and L-GATE models.

On L-INX models without the router option and on all L-GATE models, certain ports can only be used alternatively. On models with CEA-709 this means either as CEA-709 FT or as CEA-852 IP (see note 1 in Table 1 and Table 2). On models with BACnet this means either as BACnet MS/TP or as BACnet IP (see note 2 in Table 1 and Table 2). Some BACnet models have a restriction on Modbus RTU and BACnet MS/TP as they share the same port. On those models Modbus RTU can only be used, if BACnet MS/TP is disabled (see note 3 in Table 1 and Table 2).

The LGATE-951 has two EXT ports and therefore can operate M-Bus and KNX TP1 at the same time, while the LGATE-950 can use etiher M-Bus or KNX TP1.

Model / Features	LINX-200	LINX-201	LINX-202	LINX-203	LINX-210	LINX-211	LINX-212	LINX-213	LINX-220	LINX-212	LGATE-900	LGATE-902	LGATE-950	LGATE-951
CEA-709 Router														
CEA-709 RNI												✓	✓	✓
CEA-709 (FT)											✓[1]	✓[1]	✓[1]	✓[1]
CEA-852 (IP)											✓[1]	✓[1]	✓[1]	✓[1]
BACnet Router		✓		✓		✓		✓		✓				
BACnet MS/TP	✓[2]	✓	✓[2]	✓	✓[2]	✓	✓[2]	✓	✓[2]	✓	✓[2]	✓[2]	✓[2]	✓[2]
BACnet IP	✓[2]	✓	✓[2]	✓	✓[2]	✓	✓[2]	✓	✓[2]	✓	✓[2]	✓[2]	✓[2]	✓[2]
BBMD		✓		✓		✓		✓		✓	✓	✓	✓	✓
Modbus RTU	✓[3]	✓[3]	✓[3]	✓[3]	✓[3]	✓[3]	✓[3]	✓[3]	✓	✓		✓[3]	✓	✓[3]
Modbus IP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
M-Bus	✓	✓	✓[4]	✓[4]	✓	✓	✓[4]	✓[4]	✓[4]	✓[4]		✓[4]	✓[4]	✓
KNX TP1			✓[4]	✓[4]			✓[4]	✓[4]	✓[4]	✓[4]		✓[4]	✓[4]	✓
KNX IP			✓	✓			✓	✓	✓	✓		✓	✓	✓
EnOcean			✓[5]	✓[5]			✓[5]	✓[5]	✓[5]	✓[5]		✓[5]	✓[5]	✓[5]
OPC XML-DA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OPC UA			✓	✓			✓	✓	✓	✓		✓	✓	✓
OPC Client			✓	✓			✓	✓	✓	✓		✓	✓	✓
SNMP			✓	✓			✓	✓	✓	✓		✓	✓	✓
PLC (IEC 61131)					✓	✓	✓	✓	✓	✓				
LIOB Connect			✓	✓			✓	✓	✓	✓				
LIOB FT + IP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
LCD Display			✓	✓			✓	✓	✓	✓		✓	✓	✓
Serial Console	✓	✓			✓	✓					✓			
SD Card									✓	✓			✓	✓
USB			✓	✓			✓	✓	✓	✓		✓	✓	✓
Ethernet Switch/Hub			✓	✓			✓	✓	✓	✓		✓	✓	✓
WLAN			✓[5]	✓[5]			✓[5]	✓[5]	✓[5]	✓[5]		✓[5]	✓[5]	✓[5]
SSH, HTTPS, Firewall			✓	✓			✓	✓	✓	✓		✓	✓	✓

1 This model can be configured to have either FT or IP active for CEA-709.

2 This model can be configured to have either MS/TP or IP active for BACnet.

3 Modbus RTU can only be used, if BACnet MS/TP is not active on this model.

4 M-Bus and KNX TP1 can be used alternatively only on this model. To operate these protocols an expansion module is needed and must be ordered separately.

5 To operate these protocols an expansion module is needed and must be ordered separately.

Table 2: Available features in different L-INX and L-GATE models (continued).

1.10 Scope

This document covers L-INX and L-GATE devices with firmware version 5.3 and the L-INX Configurator version 5.3 and higher. The usage of logiCAD itself is beyond the scope of this manual. Please refer to the logiCAD online help in case of additional questions.

2 What’s New in L-INX/L-GATE

2.1 New in L-INX/L-GATE 5.3.0

This section describes the major changes and new features. For a full list of changes refer to the Readme file.

New L-INX and L-GATE Models

The new L-INX and L-GATE models in the small enclosure are now supported. Equipped with dual Ethernet, a built-in firewall and LCD display, these new models serve as a plug-in replacement for the older devices. Existing data point configurations and device backups can be used without modification. Only logiCAD programs need to be re-compiled for the new hardware resource. In addition, the new models also support L-IOB connect, wireless technologies and KNX.

Extended Support for U.S. Units

An extended support for U.S. units has been built into LOYTEC devices. A device can now be configured to run either in the SI or the U.S. unit system by defining SI and U.S. units per data point as shown in Figure 1. When configured for U.S. units, all data points and L-IOB I/Os process values in the respective U.S. units. This includes the Web UI, the OPC server, the parameter file, global connections, and logiCAD programs. Also the Configurator displays values in U.S. units and conversions are done automatically. A device can even be switched from one unit system to the other. In this case all persistent parameters are reset to their default values in the chosen unit system. A system register shows the currently running unit system. This makes it easy to entirely engineer a device in U.S. units or build devices that can be configured for either European or U.S. markets and meet local requirements at the same time. For more information on units please refer to Section 6.2.13.

Analog Datapoint Max Value [°F]	100	Max range of the value
Analog Datapoint Min Value [°F]	32	Min range of the value
Analog Datapoint Precision	0	Number of significant decimals
Analog Datapoint Resolution [°F]	0	Smallest value increment
Analog Point COV Increment [°F]	5	Change-of-value increment
Unit SI	°C	Data point unit used in the SI system
Unit U.S. (active)	°F	Data point unit used in the U.S. system

Figure 1: Configuration of SI and U.S. units on a data point

OPC Client

L-INX and L-GATE devices are extended by their own OPC client technology. Using the OPC client you are able to integrate other LOYTEC devices simply by importing the data point configuration of their OPC servers. This way an L-DALI device can be added to a L-INX configuration in a few steps. The OPC client can also take advantage of the commission later feature by using the Web UI in order to assign OPC server URLs at the time of device commissioning. Even the base path to imported OPC tags can be replaced. This way sub-configurations can be imported to save OPC client tags and be re-assigned later. Learn more about the OPC client function in Chapter 15.

Forward Delay in Connections

For applications that require staggered start/stop and randomization, the device implements a configurable forward delay in local and global connections. All receive data point items can be configured with a delay. A received value is written to the data point after the delay expires. The delay can be a fixed time or a time interval, in which the actual delay is randomized. With a resolution of 0.1 seconds the total delay can be up to 100 minutes. Learn more about forward delay in Section 6.4.5.

Auto-Generate and Connect

The auto-connect feature has been extended:

- Support %{folder_descr} as a placeholder in auto-generate templates. This evaluates the description property of the parent folder.

- Generate into existing connections. Epecially when auto-generating the same source data points to different technologies, the same connections are used. This makes it easier to manage auto-generated local connections.

Enhanced Structure Support on the Web UI

The display of data point structures has been improved on the Web UI. The data point listing shows a textual short version of the structured value instead of a hexadecimal Byte buffer. Also CEA-709 structures of NVs without sub-data points are displayed and can be edited.

Project Documentation

A new feature on the device is a Web UI for creating and viewing project documentation on the device. The documentation editor requires admin rights and allows storing files on the device or creating documentation links as URLs. Both items can be viewed by guest users. Examples include storing cabling plans as PDF or adding links to a Web site containing manuals, plans or other useful project documentation. Read Section 5.2.30 to learn more about project documentation on the device.

LWEB-802 Application on Device

The L-INX and L-GATE device now come with a pre-installed version of the LWEB-802 application. This enables out-of-the-box L-WEB applications on sites without Internet access. An enhanced Web UI also allows installing a specific version of the LWEB-802 application, which overrides the pre-installed one. As a default, the Internet version is still available. Learn more on how to configure the L-WEB application in Section 5.7.2.

BACnet

All previously extended BACnet features and new features have been BTL-certified in the 5.X firmware series. New features include:

- BACnet object names can be made writeable. With this new option the user can create data point configurations with generic object names and assign location-specific names later in his OWS (see Section 7.3.5).

- BACnet simple value objects are now supported. The user can create Large Analog Value, Signed Integer Value, Unsigned Integer Value, Character String Value and Octet String Value objects as needed. Also auto-connect supports these new object types.
- For Trend Log objects, the properties Notify_Type, Event_Enable, Notification_Class can be pre-configured in the data point configuration (see Section 6.7.4).

User-defined Serial Protocols in logiCAD

A new logiCAD service block has been introduced, which allows implementation of generic serial protocols. The 'SerialComm' service block can receive and transmit strings from and to a serial port using start frame and end frame delimiting characters. A logical bus number is assigned to the service block, which can be selected as a custom serial protocol on the port configuration Web UI as shown in Figure 2. Please read Section 17.6.10 to learn more about using the 'SerialComm' block.

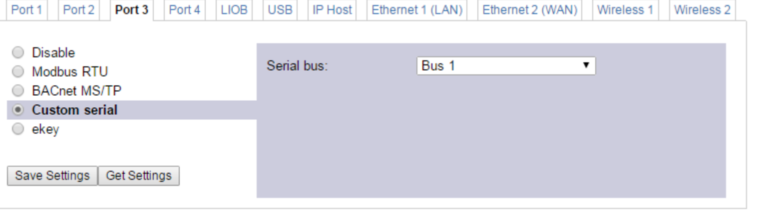


Figure 2: Selecting a custom logiCAD serial protocol.

Data Point Creation from CSV Lists

The Configurator implements a new data point template CSV import feature, which can be used by external tools to generate a list of data points. Each line in this CSV references a data point template file that describes all properties of the data point. The CSV supplies name, description and path of the data point to be created. Additionally, data points can be automatically scheduled and trended. Learn more about data point templates in Section 7.18.

KNX Protocol Analyzer

Devices that support the KNX protocol now have a built-in protocol analyzer. It can be accessed through the KNX statistics Web interface and allows starting, stopping and storing protocol logs on any of the KNX interfaces. Protocol logs can be loaded from the device and stored on the PC either in CSV or in XML format. The XML format can be opened in the ETS protocol viewer. Learn more about the KNX protocol analyzer in Section 13.4.3.

ekey Fingerprint Readers

LOYTEC devices can now be used to build applications with biometric access control (fingerprints). The ekey fingerprint reader devices can be attached to the RS-485 port. Users and fingers can be enrolled and grant or deny access when swiping the finger over a reader. User groups can be implemented and distributed over reader devices. Learn more about ekey fingerprint reader support in Chapter 16.

2.2 New in L-INX/L-GATE 5.1.0

This section describes the major changes and new features. For a full list of changes refer to the Readme file.

Dual-Ethernet with Separate Networks

L-INX and L-GATE models with two Ethernet interfaces can now be configured to work with separate and isolated IP networks. For example, one Ethernet interface can be accessed over HTTPS from a WAN connected to Ethernet 2 while the building network services are running locally on the LAN connected to Ethernet 1. For configuration the device provides separate Ethernet tabs in the port configuration, which allow selecting the offered services on each interface. The example in Figure 3 shows a WAN interface with HTTPS and OPC UA only, while BACnet/IP or CEA-709 over IP are still bound to Ethernet 1 (LAN). Another use case for this feature is operating a gateway between otherwise entirely isolated building networks, e.g. BACnet/IP on Ethernet 1 and KNXnet/IP on Ethernet 2. For more information on how to use multiple Ethernet ports please refer to Section 5.2.5.

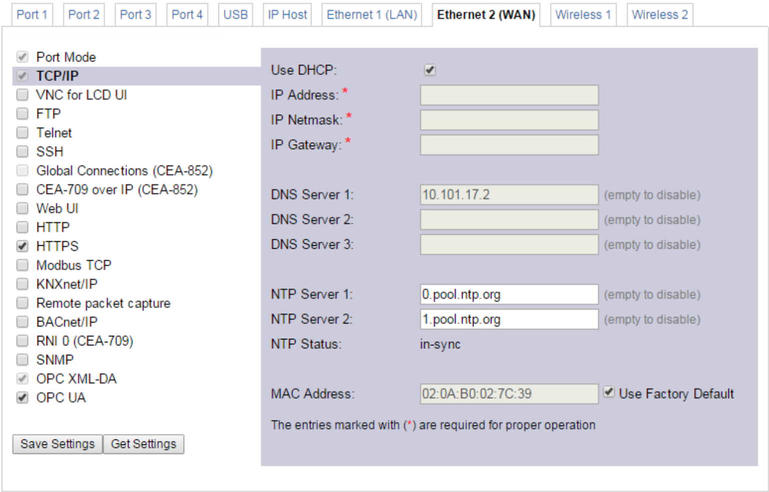


Figure 3: New Ethernet 2 (WAN) tab

WLAN Interface

In combination with the external LWLAN-800 interface, the device provides new interface tabs for wireless IP networks. Similar to the second Ethernet interface, one can choose which protocols are available on the wireless network. The wireless interface can be configured as a WLAN client, access point or mesh node. Using the latter, a wireless mesh network of LOYTEC devices can be built. Please refer to Section 5.2.7 to learn more about the WLAN interface.

EnOcean

The EnOcean technology is now supported by the L-INX and L-GATE devices. Together with the external LENO-800 interface, EnOcean sensors and actuators can be easily integrated. LENO-800 interfaces are available for European, U.S. and Japanese frequency bands. Data point configurations for the EnOcean technology are built using EnOcean device templates. The actual assignment of physical devices is done later in the teach-in process over the Web UI. For more information on the EnOcean technology please refer to Chapter 14.

SNMP

For accessing vital operational data in standard IT equipment, L-INX and L-GATE devices offer an SNMP management base (MIB). All system registers and OPC-exposed data points are available in that MIB. The MIB file can be downloaded from the device and imported in the SNMP management tool. Alarms on the device can be exposed as SNMP traps. For more information on configuring and using SNMP with a LOYTEC device please refer to Section 18.4.

Online Commissioning

The Web interface of the device now provides an online commissioning tool for the respective networking technologies. Using this tool, data point configurations can be created based on placeholder devices, which are marked to be commissioned later. The necessary addressing information can be assigned later on the Web interface by scanning for devices online or be entered manually. Device replacement is also possible in the commissioning Web interface without the need to edit devices in the data point configuration. For more information on the commissioning Web UI please refer to Section 5.4.

Favorites

Enhancements for using favorites include engineering units, active/inactive texts, multi-state texts and value persistency for unlinked favorites. A new feature are structured favorites. For example, the structure of a SNVT_switch may now be created as a favorite. The top-level part of a structured favorite can be linked to a structured data point of the same type. Alternatively, the individual member elements of a structured favorite can be linked to other, individual data points. This way, one is able to create a structured favorite, that can be linked directly to a matching SNVT or to separate BACnet objects. Read more about structured favorites in Section 6.2.11.

Web Interface

The Web interface of the device offers a number of new features:

- A new device info page provides a quick overview of all relevant operational parameters, such as CPU load, active protocols, time synchronization and many more.
- The trend log configuration on the Web UI now also provides a preview tab, which shows a chart of the trend log data. The trend chart allows zoom, scrolling and hiding specific data curves, as shown in Figure 4.

Figure 4: New trend chart on the Web UI

Scheduler

The scheduler objects have been extended by the following new features:

- Color support in BACnet and generic schedulers allows consistent assignment and display of preset colors in L-WEB, L-VIS and the Configurator scheduler UIs.
- Event auto-prune removes passed events, if the capacity of exception events in a scheduler gets low.
- The scheduler default for LONMARK and generic schedulers is extended by a “silent” mode. In this mode the scheduler becomes inactive as soon as the last event is withdrawn. This mode can be used in event-driven scheduler models.
- Generic schedulers allow specifying an existing value preset as the schedule default. LONMARK and BACnet schedulers try identifying a matching preset name from the schedule default value.

5000 BACnet Client Maps and Dynamic Polling

The number of BACnet client mappings on the L-INX has been increased to 5000. This has become possible with the use of strict on-demand dynamic polling on the device. This means any data point configured for polling (client mappings, external NVs, Modbus, etc.) will start polling only when it is actually used, i.e. displayed on L-WEB, the data point Web interface or used by a trend log. This reduces the required network bandwidth.

1000 Trended Data Points in LINX-12x/15x/22x/LGATE-95x

The bigger L-INX and L-GATE devices now allow a total of 1000 trended data points distributed among the 512 trend logs. This limit can be exploited when trending two or more data points in a generic trend log.

Alarm Server Ack-All Data Point

Alarm servers now provide a special ackAll property data point. When writing TRUE on that data point all currently active alarms on the alarm server are acknowledged.

Format Strings in E-Mails

Data point variables used in e-mail templates can now use format strings to specify their numeric appearance in the e-mail text.

Output NVs with Integrated Feedback

Output network variables (NVs) are used to send updates to remote nodes. In order to poll back the actual value of the remote variable, a feedback data point had to be created for the output NV. Now changing the direction on the NV from output to value, an integrated feedback function is activated without the need for an extra data point. This becomes especially useful in data point templates.

Modbus and M-Bus Templates

Modbus and M-Bus device templates offer some new features that boost productivity:

- Templates can now contain definitions of pollgroups and other type resources used by Modbus data points.
- Templates can now store a folder hierarchy.

2.3 New in L-INX/L-GATE 5.0.0

This section describes the major changes and new features. For a full list of changes refer to the Readme file.

Generic Scheduler

Generic schedulers – like generic trends and alarms – can now be created, that are neither CEA-709 nor BACnet objects. Generic schedulers appear next to the generic alarm folder and are ready-to-use on any device. This is beneficial for creating technology-independent applications. Generic schedulers can write to any technology as well as data point favorites and are the ideal solution, if configured via LWEB-900 only. For more information on creating generic schedulers refer to Section 7.12.

Optimum Start for Schedulers

Up to now, optimum start was limited to using a SNVT_tod_event in a CEA-709 scheduler. Now all schedulers (including BACnet and generic schedulers) support timeToNext and nextState data points for implementing optimum start algorithms also for those technologies. Schedulers are extended by property relations, which offer the time to the next state in minutes and the next state in data points. Also, the enable, enableFb (feedback), and presetName information is available as property relations (see Section 6.2.12).

The BACnet scheduler object also has custom properties that expose time to next state and next state to BACnet. If standard BACnet objects are required, the property relations can be linked to any matching BACnet server object as shown in Figure 5 (BV_nextState, AV_timeToNext).

LINX-200 ▸ BACnet Port ▸ Scheduler

Scheduler Name	No.	Direction	OPC	Direction
◢ Schedule_BO_occupied	1	Value	☑	Value
enable	1.1	Value	☑	Value
enableFb	1.2	In	☑	In
nextPresetName	1.3	In	☑	In
nextState -> BACnet Port.Datapoints.BV_nextState	1.4	In	☑	In
presetName	1.5	In	☑	In
timeToNext -> BACnet Port.Datapoints.AV_timeToNext	1.6	In	☑	In
BACnet Port.Datapoints.BO_occupied	1.7			

Figure 5: BACnet scheduler with next state data points.

Favorites

Favorites have been vastly extended to be compatible with all uses that were previously only possible for regular data points. Favorites can now be trended, can have an alarm condition (reporting to a generic alarm server), and be scheduled. Historic filters can be applied to favorites. Favorites can also be used in e-mail templates, math objects and connections like any other data point. Furthermore, it is possible to link favorites to property relations and vice versa, e.g. link a favorite directly to a historic filter data point.

Historic Filters

Historic filters have been extended by a generic delta calculation between any of the defined filter items or the current value. This way, it is no longer necessary to create a separate math object to calculate the consumption of the previous month. Also, a definition for "last day of month" has been added to the monthly period. See Section 6.5.6 for more details.

256 Trend Logs on LINX-10x/11x/20x/21x/LGATE-900

The smaller L-INX and L-GATE devices are now extended from 100 trend logs to 256 trend logs at a total of 256 trended data points. This is beneficial for native BACnet trend logs, which allow only one data point per trend log object. The increase comes at the cost of available log rate, which is reduced from 121,050 to 60,525 records per hour, once more than 100 trend log objects are created. This is, however, still sufficient for 256 energy trends at 1 minute log interval each.

Web Interface

The Web interface of the device offers a number of new features:

- Live update of values in the data point list. This allows monitoring values without repeated pressing of reload. Data point structures can be expanded or collapsed for better overview.
- Breadcrumb navigation has been added to the data point list. This gives faster access to sub-folders.
- A new firmware upgrade menu on the Web interface allows online checking for firmware updates and upgrading by selecting a local firmware file. All this is possible without starting the Configurator.
- The trend overview page displays current trend log states and provides controls for easy trend data upload in CSV format.

Data Point Polling

The receive timeout on input and value data points has been generalized. A receive timeout can now be defined on all technologies for input and value data points. Writing updates to those data points from any source (e.g. a global connection) resets the receive timeout.

Background polling can be enabled in the project settings, which allows a slow polling of all data points even if no pollcycle or dynamic polling has been activated on those data points. For more details refer to Section 6.2.2.

OPC UA Server

The OPC server on the devices, which support security, has been extended by an OPC UA server. This supports the OPC UA binary protocol and exposes the same OPC tags as the well-known OPC XML-DA server. In addition OPC UA offers superior security features as well as slimmer data transfers. For more information on the OPC UA server please refer to Section 10.2.

BACnet

All previously extended BACnet features and new features have been BTL-certified in the 5.X firmware series. New features include:

- Intrinsic alarming for the Accumulator object.
- Option to keep OWS settings in BACnet properties also after a new configuration has been downloaded (see Section 7.3.5).
- BACnet naming rules can now freely define how data point names are constructed out of scanned BACnet information (see Section 7.3.2).

Conversion to Value Data Points

Firmware versions since 4.9.0 support value data points. The default behavior of the Configurator can be controlled in the project settings. This defines whether new value data points or the old "_Read/_Write" data point combinations shall be created. When using templates in L-WEB or L-VIS it is often an all-or-nothing approach, a mixture is not practical.

For making the transition in old projects to value data points, the Configurator now offers a conversion tool. Multi-select old read/write data points and choose the item **Convert to value** from the data point context menu. This converts the selected data points to the new value data points, leaving the IDs, default values and data point usages intact.

Application Objects

Application objects such as math objects, e-mail templates, and alarm logs can now be organized in folders. Copy and paste of application objects between Configurators has been improved. Math objects now allow single constants and single variable assignments, such as "=5" or "=v1". Input variables can be configured to trigger a new calculation or not.

Configurator Usability

The folder tree of the data point manager has been brought to state-of-the-art user concepts such as multi-select, drag-and-drop of folders, moving folders, deleting multiple folders and their contents. A name filter can be applied to quickly find folders in the folder tree.

The property tab has been extended by a property name filter. This makes it easy to find the desired property by typing a sub-string of its name in the filter. For a detailed description on the data point properties refer to Section 7.2.4.

Data point link navigation has been made easy by using a **Go to data point** context menu and speed button everywhere data point references are displayed. A data point usage report dialog shows all references to the selected data point and allows jumping to selected objects.

The new **PLC conflicts** tab shows all PLC write conflicts with other write usage of data points such as math objects, connection receivers, etc., while editing the project. This allows finding problems prior to the configuration download. The conflicts tab also provides easy navigation to data points listed as problematic. For more information refer to Section 17.4.7.

Modbus and M-Bus Templates

Modbus and M-Bus device templates offer some new features that boost productivity:

- Allow setting the device name when storing the template in a file.
- Naming rules for created data points when used on device.
- Configure naming rules for the automatic naming of Modbus devices (e.g. with or without device address).
- Templates include all OPC and PLC settings.
- Imported device templates are also stored in the device configuration. They are now also available after loading a configuration out of the device, not only from the disk file.

3 Quick-Start Guide

This chapter shows step-by-step instructions on how to configure the L-INX for a simple OPC server application with a logiCAD program and using basic I/Os in L-IOB modules. For the L-GATE it gives an example of how to quickly map CEA-709 network variables to BACnet objects in the gateway.

3.1 Hardware Installation

Connect power (12-35 VDC or 12-24 VAC), the CEA-709 and/or MS/TP network, and the Ethernet cable to the ports, which are labeled respectively, as shown in Figure 6. More detailed instructions are shown in Chapter 4.

Important:	*Do not connect terminal 17 with Earth-ground!*

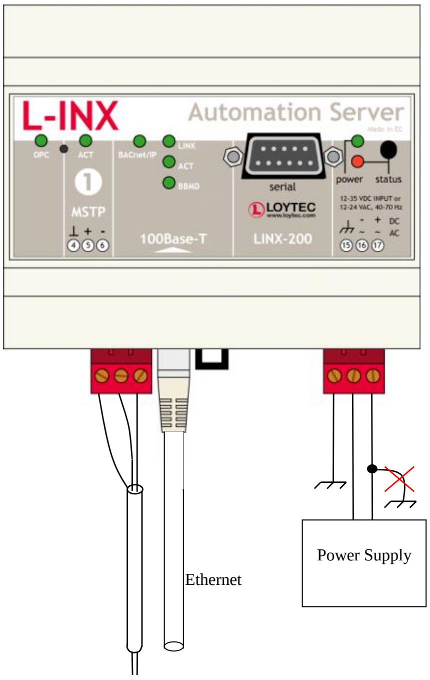


Figure 6: Basic Hardware Installation.

If the L-INX device is connected to a BACnet MS/TP network, the MS/TP network segment must be properly terminated with an LT-04 network terminator connected at each of the two ends of the segment media.

3.2 Configuration of the Device

The device can be configured via a console interface, LCD display or via the Web interface. To configure the device, the following steps have to be performed:

1. Setup IP configuration (see Sections 3.2.1, 3.2.2 and 3.2.3).
2. Setup basic BACnet configuration (see Section 3.2.4).
3. Setup the OPC configuration (see Section 3.3).

Note: *This setup procedure assumes the use of the IP interface.*

3.2.1 IP Configuration on the Console

Use a PC terminal program with the communication settings set to 38,400 bps / 8 data bits / no parity / 1 stop bit / no handshake. To connect COM1 of the PC to the Console on the device, use a standard null-modem cable with full handshaking. Power up the device or press **Return** if the device is already running. The following menu should appear on the terminal:

```
Device Main Menu
================

[1]  Show device information
[2]  Serial firmware upgrade
[3]  System configuration
[5]  IP configuration
[7]  BACnet configuration
[8]  Reset configuration (factory defaults)
[9]  Device statistics

[a]  Data Points

[0]  Reset device

Please choose:
```

Figure 7: Device Main Menu

Select ‘5’ from the device main menu and enter the IP address, netmask, and gateway address. Note that you must use different IP addresses if you are using multiple IP devices in your setup.

```
IP Configuration Menu
=====================

[1]  DHCP                  : disabled
[2]  IP Address            : 192.168.24.99
[3]  IP Netmask            : 255.255.192.0
[4]  IP Gateway            : 192.168.1.1
[5]  Hostname              : test-linx200
[6]  Domainname            : <unset>
[7]  DNS Servers           : 10.101.17.2
[9]  MAC Address           : 00:0A:B0:01:0A:4C (factory default)
[0]  NTP Servers           : <unset> (out-of-sync)
[b]  Link Speed & Duplex   : Auto Detect

[q]  Quit without saving
[x]  Exit and save

Please choose:
```

Figure 8: Enter basic IP settings.

Press ‘x’ to save the IP settings and reset the device with the main menu item ‘0’ in order to let the new IP settings take effect.

Important! ***The default IP address 192.168.1.254 is only set for configuration access. It must be changed in order to make the device functional.***

You should now be able to connect to the device with a Web browser, access it through an OPC XML-DA client or add it to a CEA-852 configuration server as an IP-852 channel member. Also the L-INX Configurator is now able to connect to this device.

3.2.2 IP Configuration via the Web Interface

Optionally to using the console interface one can also use the Web interface to configure the client device. In a Web browser enter the default IP address '192.168.1.254' of the device. Note that if your PC has an IP address in a subnet other than 192.168.1.xxx, please open a command tool and enter the following route command to add a route to the device.

To Add a Route to the Device

1. Windows **START → Run**
2. Enter 'cmd' and click **OK**.
3. In the command window enter the command line

   ```
   route add 192.168.1.254 %COMPUTERNAME%
   ```

   In Windows7 replace %COMPUTERNAME% with the PC's actual IP address.
4. Then open your Web browser and type in the default IP address 192.168.1.254.

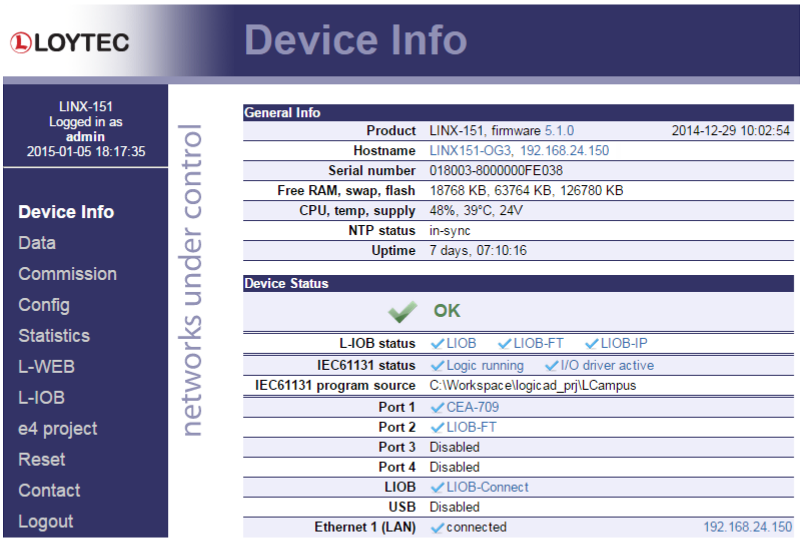


Figure 9: Example Start Screen.

5. Click on **Config** in the left menu. You will be asked to enter the administrator password in order to change the IP settings. Enter 'loytec4u' and select **Login**.

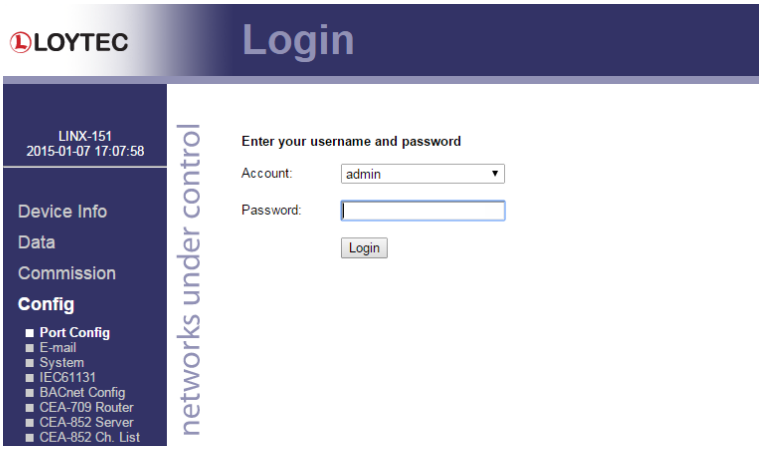


Figure 10: Enter 'loytec4u' as the default administrator password.

6. The Config menu opens. Click on **Port Config** and change to the tab **Ethernet**. The TCP/IP settings are selected as shown in Figure 11. Enter the IP address, the IP netmask, and IP gateway for this device.

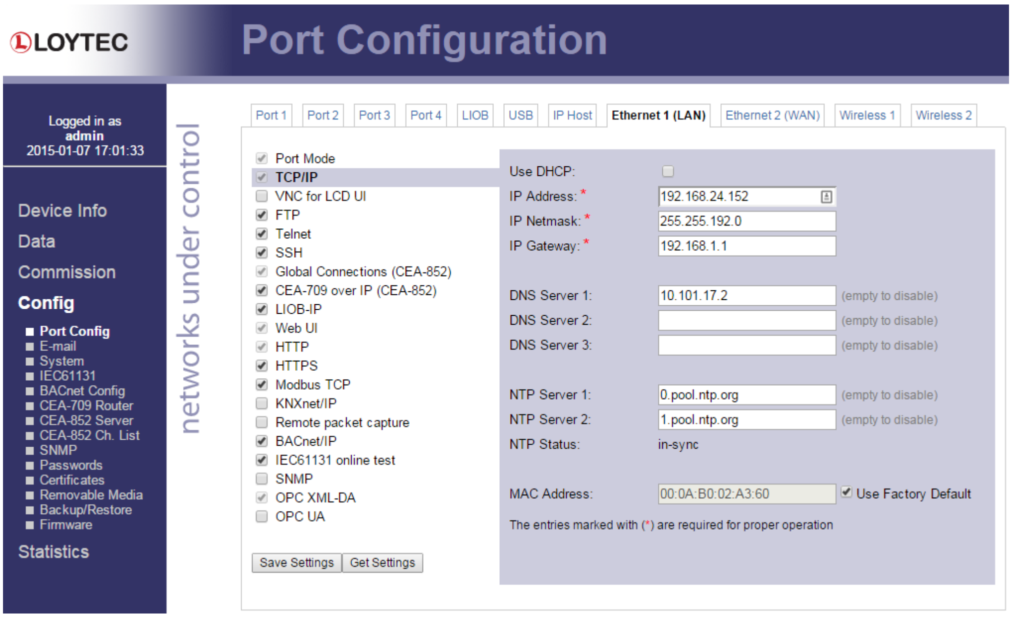


Figure 11: Enter IP address and gateway.

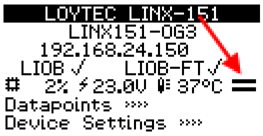
7. Press **Save Settings** and then reset the device by selecting **Reset** in the highlighted text. This changes the IP settings of the device.

3.2.3 IP Configuration via the LCD Display

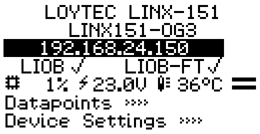
Device models with an LCD display can also be configured to their basic settings through jog dial navigation on the LCD UI. Turn the jog dial to navigate between menu items and press to enter a menu or go into selection mode. When in selection mode turn the jog dial to alter the value and press again to quit the selection. Some input fields provide acceleration. This means turning faster changes the value in larger increments.

To Set the IP Address on the LCD Display

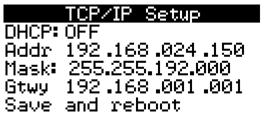
1. On the LCD main screen set the desired language. Navigate to the flag symbol, press the button and choose the desired language.



2. Navigate to the IP address on the main screen and press the button.



3. There navigate to the needed input fields, press and change the value. Press again to set the value. Continue to the next field.



4. Finally navigate to **Save and reboot** and press.

5. Acknowledge the reboot and the device reboots with the new IP address.

3.2.4 BACnet Configuration

To configure the BACnet interface, at least the Device ID and the Device Name must be configured (see Figure 12).

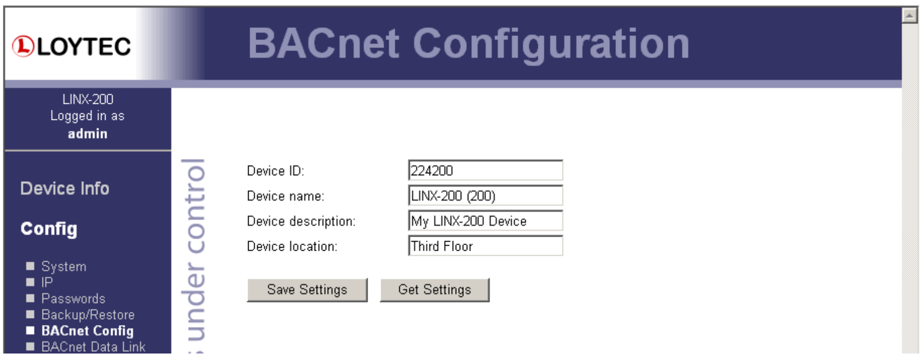


Figure 12: BACnet Device Configuration.

The device ID corresponds to the instance number of the BACnet device object. It must be a unique ID on the BACnet internetwork. Also the Device Name must be a unique name on the BACnet internetwork.

By default the BACnet/IP data link layer is used. If the device shall be used with the BACnet MS/TP data link layer, please refer to Section 5.2.17 for further information.

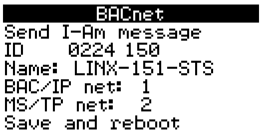
On devices with an LCD display, the BACnet device ID can also be configured over the LCD UI.

To Configure the BACnet Device ID over the LCD Display

1. On the LCD main screen navigate to **Device Settings »»**.

2. Then navigate to the menu **BACnet »»**.

3. In that menu navigate to the **ID** input for entering the device ID. The field is split into two controls, one for the thousands and one for singles, to simplify entering big numbers.



4. After the device ID has been entered the device name is automatically assembled using that device ID, if no other name has been configured on the Web UI.

5. On a BACnet router navigate to the **BAC/IP Net** menu item and enter the BACnet network ID of the BACnet/IP network. Then choose the appropriate **MS/TP Net** number for each available MS/TP port. To disable the router port, scroll down till **off** appears.

6. To let the changes take effect, the device needs to be rebooted. For doing this now you may select the menu item **Save and reboot**.

3.3 Getting Started with the L-INX Configurator

Before setting up a gateway, a working IEC61131 program or creating an L-WEB visualization, the data points of the L-INX automation server or the L-GATE need to be set up. These can be data points of L-IOB I/Os, network variables, BACnet objects, and other available technologies. Before executing the steps below, install the L-INX Configurator Software from the ‘setup.exe’. This file can be downloaded from www.loytec.com.

To Start a Configurator Project

1. Start the L-INX Configurator software by selecting Windows **Start → Programs → LOYTEC LINX Configurator → LOYTEC LINX Configurator**. The application starts up and displays the data point manager screen as shown in Figure 13.

2. When the device is online, connect to the device by clicking on the **Connect to device** speed button as indicated by the red rectangle in Figure 13.

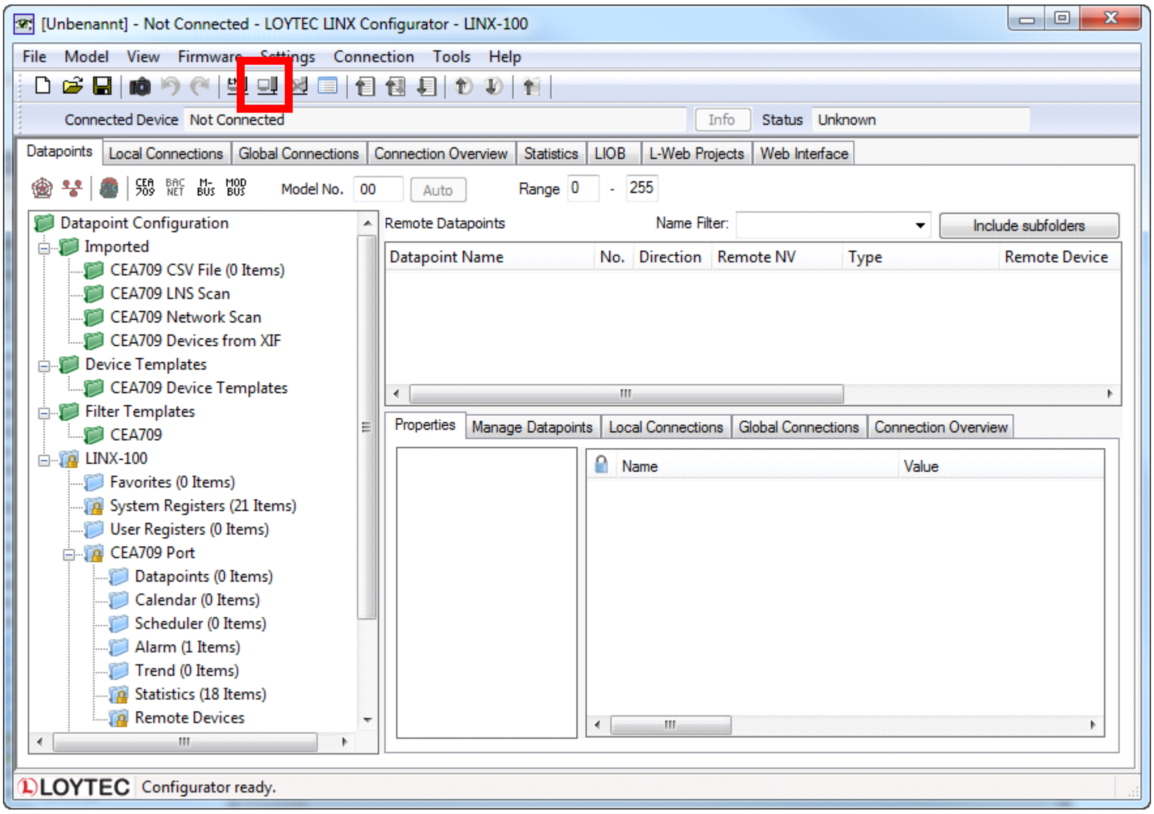


Figure 13: L-INX Configurator main screen.

3. For detailed information on how to create data points out of the network please refer to Section 7.7 for CEA-709 or 7.9 for BACnet.

3.4 Getting Started with the Gateway

The L-GATE and L-INX devices can be used as a gateway between any of the supported network technologies. This example gets you started with the gateway and shows how to map CEA-709 network variables from an LNS database to BACnet. The installer of the L-INX Configurator has already registered the L-INX Configurator as an LNS plug-in. This quick-start assumes LonMaker TE as the LNS tool.

To Configure a Gateway

1. Open a new LNS database or open an existing LNS database. In the latter case you need to register the L-INX Configurator as an LNS plug-in in that database (see Section 7.1.2).

2. Add your LONMARK devices to the LNS database.

3. Add a new device using the device template for the L-GATE (e.g. ‘LOYTEC LGATE-950 FT-10’) or L-INX (e.g. ‘LOYTEC LINX-150 FT-10’) and commission the device.

4. Right-click on the added device template and select **Configure …** This opens the L-INX Configurator as an LNS plug-in (see Section 7.7.1).

5. Click on the **Datapoints** tab of the main window.

6. Click on the button  **Scan channel**. This scans in all NVs on all devices connected to the CEA-709 channel of the device.

7. After the scan has completed, the folder **LNS Database Scan** is populated with the found NVs as shown in Figure 14. For detailed information refer to Section 7.7.2.

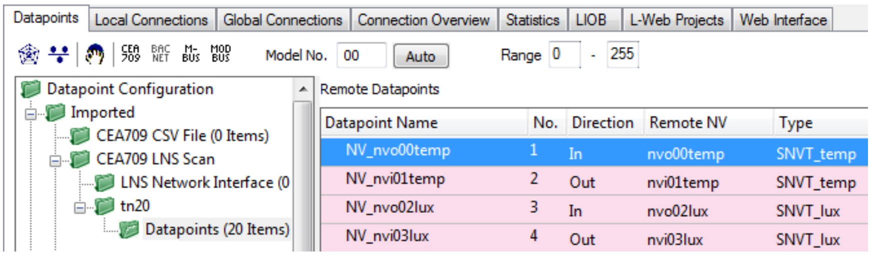


Figure 14: Result of an NV scan in LNS.

8. Select the scanned NVs for use in the gateway and click on the button  **Use on Device** in the tool bar.

9. The data points are created as dynamic NVs under the **Datapoints** folder of the **CEA-709 Port** folder.

10. Select the NVs under the port folder, which shall be mapped to BACnet. Click on the speed button  **Generate and connect selected** in the tool bar.

11. The generated BACnet objects appear in the **Datapoints** folder under the **BACnet Port** folder as shown in Figure 14. These objects are also automatically connected to the respective NVs.

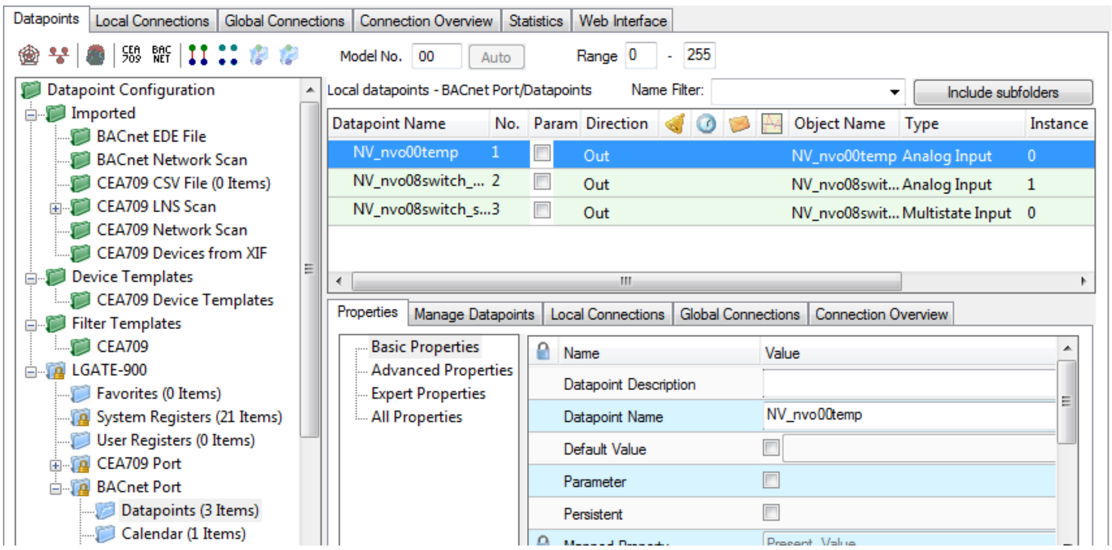


Figure 15: Auto-created BACnet Points in the BACnet Folder

12. Then download the configuration into the device by clicking the **Download Configuration** speed button.

13. After the reboot of the device value changes in the NVs are propagated to the connected BACnet objects and vice versa.

3.5 Configuration of the L-IOB I/O Modules

The L-IOB I/O modules can be attached either directly to the LIOB Connect bus or to the LIOB-FT bus with standard TP/FT-10 wiring rules. Please visit the L-IOB User Manual [7] for detailed hardware installation and terminal configuration instructions.

The L-INX Configurator uses a separate tab to configure the L-IOB devices. The L-IOB device configuration can be done off-line and is shown in the following steps.

To Configure L-IOB I/Os

1. Add L-IOB devices on the **LIOB** tab from the supplied L-IOB templates using the **Add Device(s)** button as shown in Figure 16.

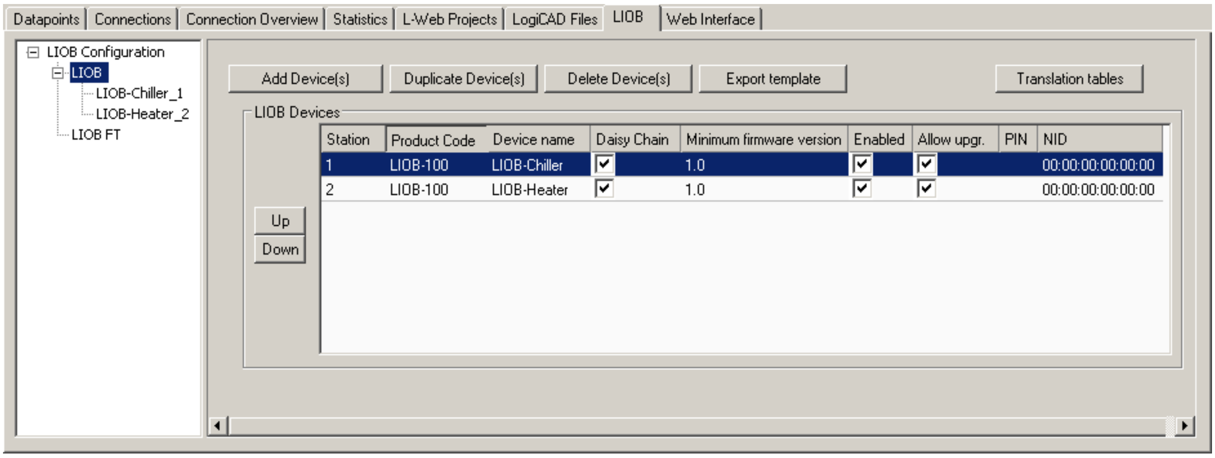


Figure 16: Add L-IOB devices to the LIOB Connect bus.

2. Select a L-IOB device in the tree on the left-hand side and enter names for the terminals by double-clicking into the **Name** column as shown in Figure 17.

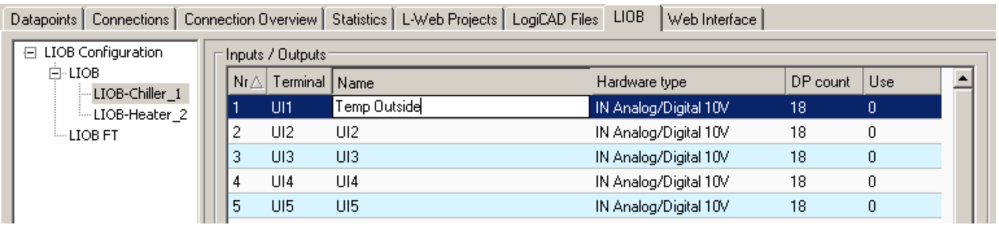


Figure 17: Change L-IOB terminal names for your installation.

3. Select a terminal and change the object parameters to configure this terminal. You can multi-select terminals and change the parameters for all selected terminals.

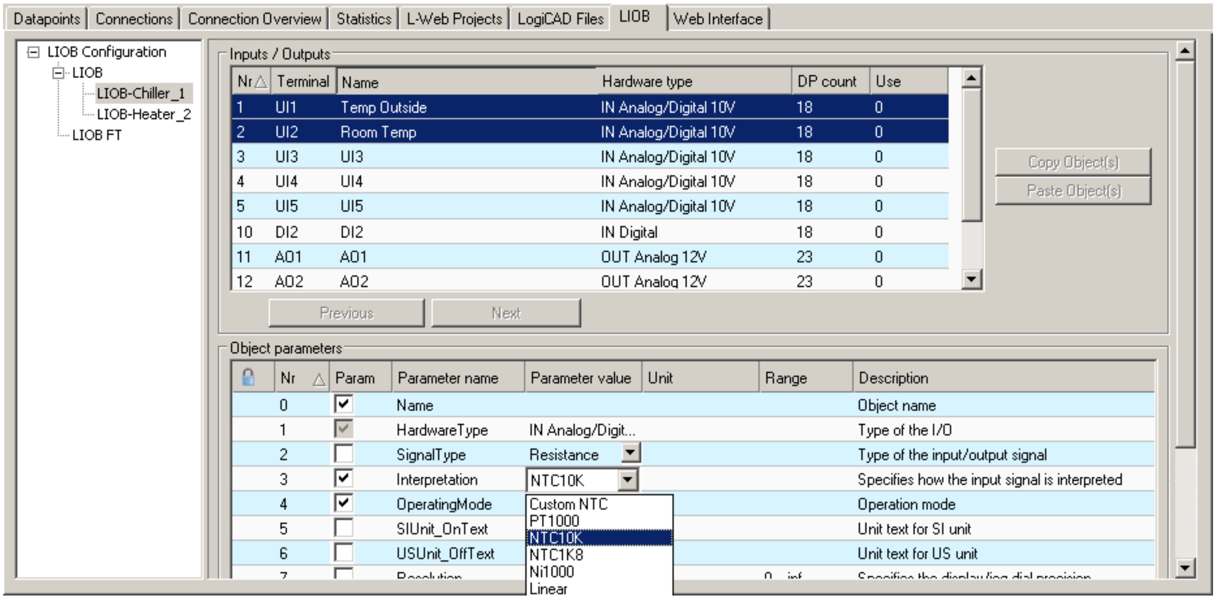


Figure 18: Change L-IOB parameters for the selected terminal(s).

4. On the **Datapoints** tab the data points for the LIOB terminals have been created. These data points can be used, e.g., in the logiCAD IEC61131 program. For terminal inputs the data point L1_x_UIy_**Input_Read** will be used to read an input terminal and for terminal outputs the data point L1_x_DOy_**Output_Write** will be used to set an output terminal.

5. After downloading the L-INX configuration into the L-INX device, the L-IOB input and output terminals can be tested with the L-INX Web UI. An example is shown in Figure 19.

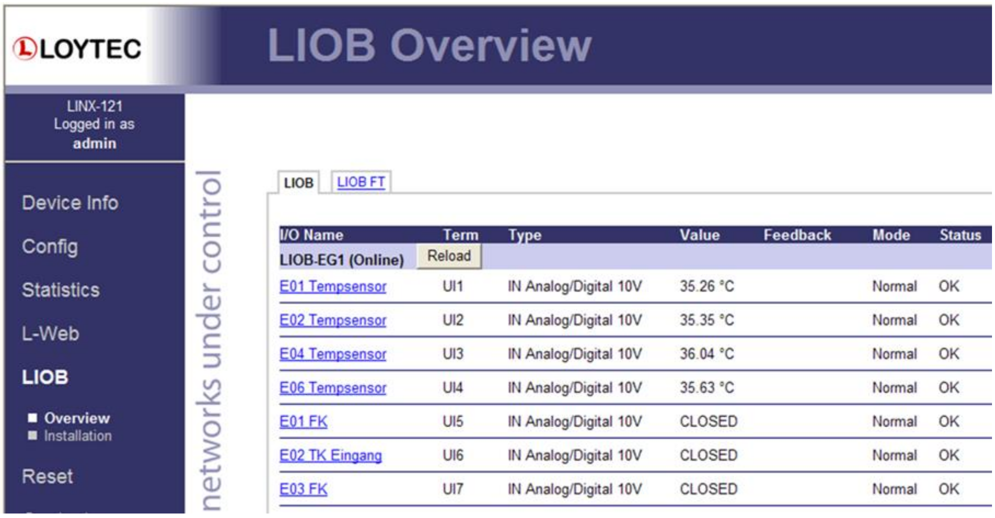


Figure 19: Test L-IOB inputs and outputs on the Web UI.

3.6 Getting started with logiCAD

For developing IEC61131 programs with logiCAD the following components must be installed:

1. L-logiCAD setup package. This package installs the logiCAD software, which is needed to design PLC programs for the L-INX device.

2. L-INX Configurator. This software is required to configure the L-INX device to provide the necessary data points to the PLC and integrate the device into the network.

3. logiCAD license for using logiCAD on the PC. The license is available as a softlock version or as a hardlock version with a USB dongle. On virtual machines it is mandatory to use the hardlock license.

A detailed guide on how to install the software components described above and upgrade of an older license can be found in Section 7.1 and Section 17.2.

To Start a logiCAD Project

1. After installing the necessary software components start logiCAD from the L-INX Configurator by clicking the **Start LogiCAD** speed button.

2. The project wizard starts automatically as shown in Figure 20.

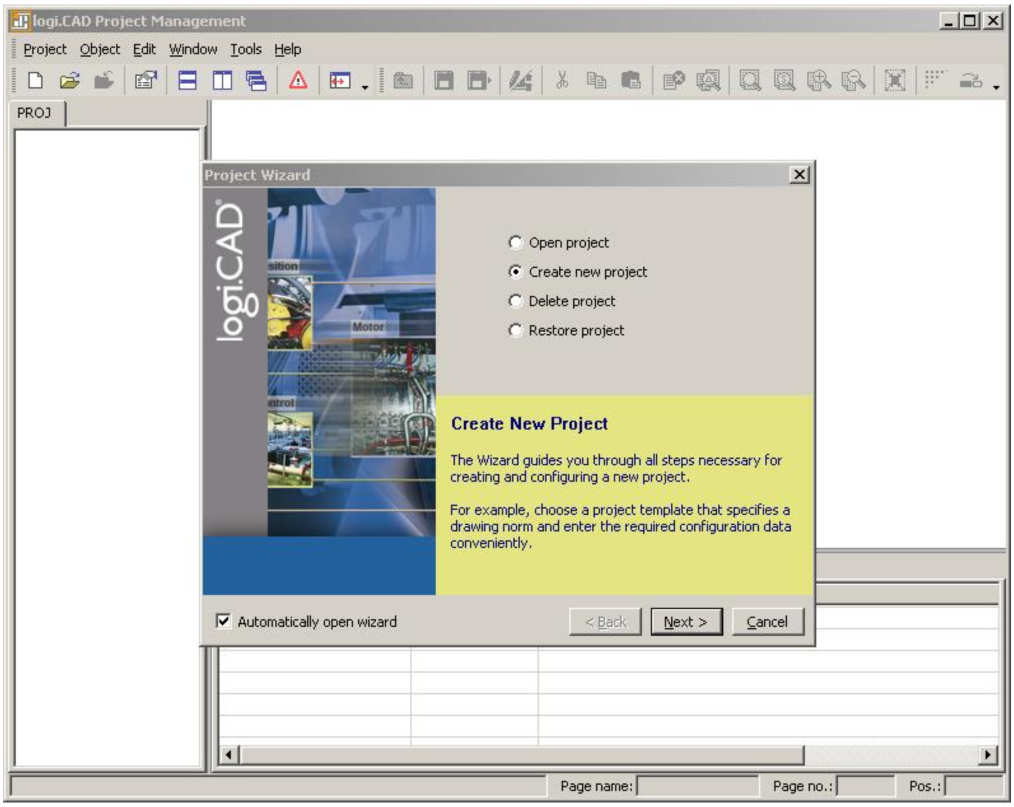


Figure 20: logiCAD project wizard

3. Select Create new project and press **Next**.

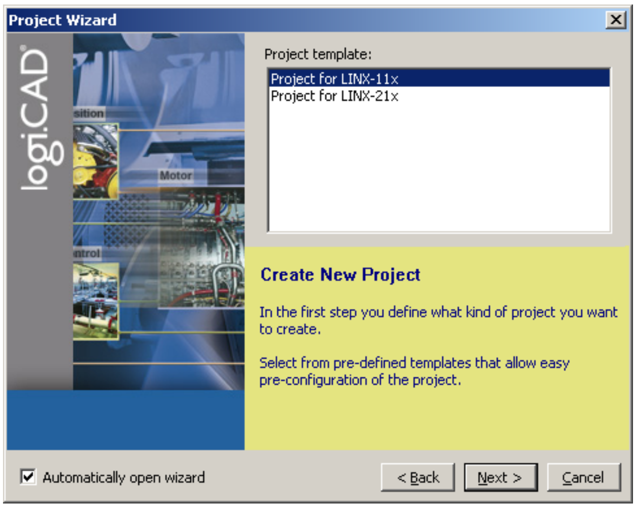


Figure 21: Available project templates

4. Select the project template for the L-INX device (e.g., LINX-11x or LINX-12x).

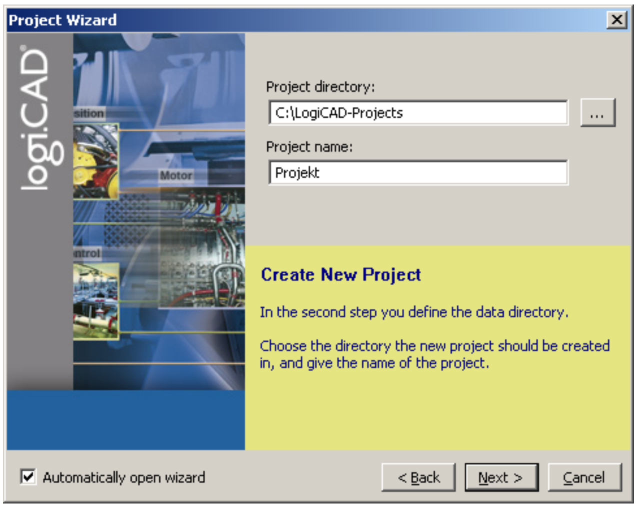


Figure 22: Project name and path

5. Specify the name of the project and the path where to store the project files, see Figure 22.

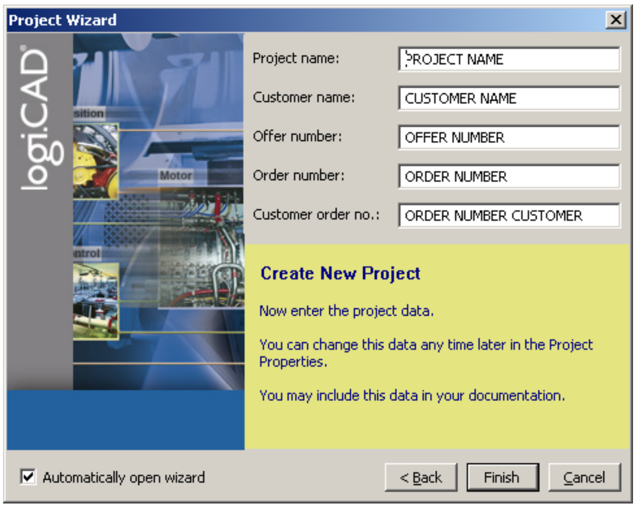


Figure 23: Additional information

6. After specifying additional information the new project is created by pressing the **Finish** button.

7. As shown in Figure 24 below, expand the tree element **Functionplans** and double click **Plan_1** in order to start editing the plan.

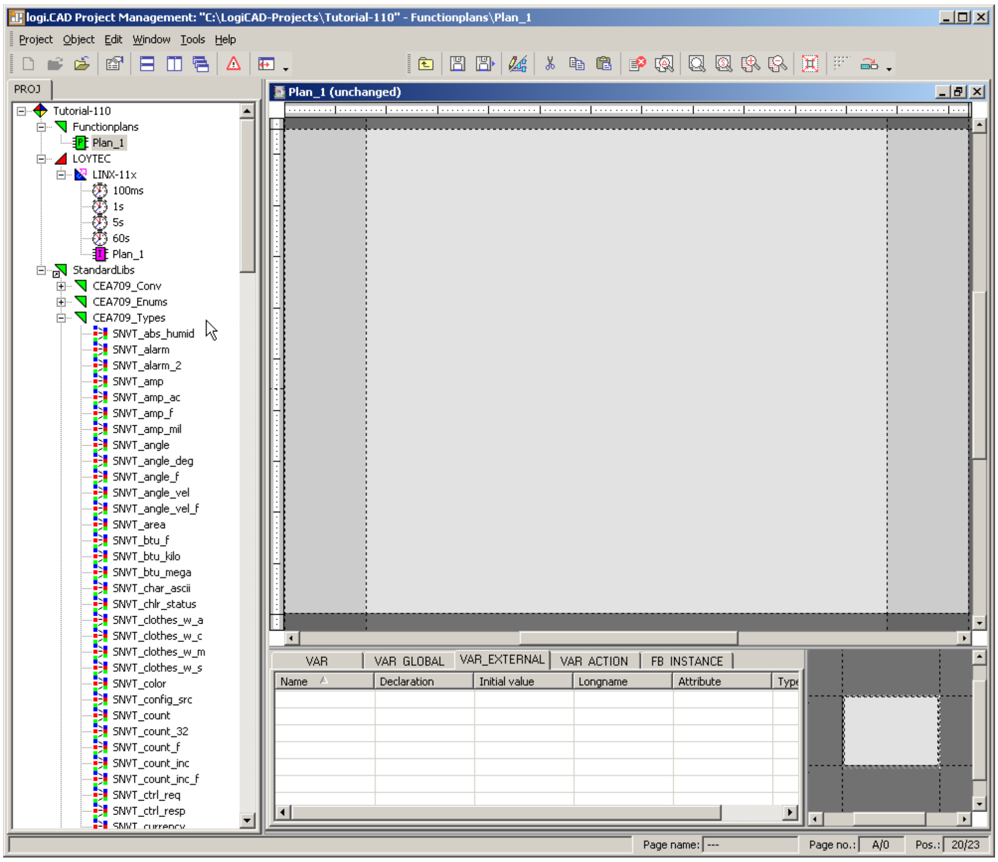
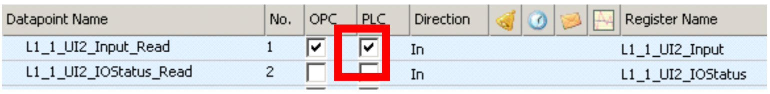


Figure 24: Edit Plan_1

8. In the new LogiCAD project there are no external variables yet. To expose data points from the L-INX to the logic program, activate the **PLC** check box of the corresponding data points in the Configurator, e.g., the L-IOB I/Os of the universal input or any other

data point such as a network variable, user register or BACnet object.



Datapoint Name	No.	OPC	PLC	Direction					Register Name
L1_1_UI2_Input_Read	1	☑	☑	In					L1_1_UI2_Input
L1_1_UI2_IOStatus_Read	2	☐	☐	In					L1_1_UI2_IOStatus

9. When you have completed selecting the PLC data points, click the **Export variables to LogiCAD** speed button while LogiCAD is running.

10. The data points now appear as variables in LogiCAD in a folder under the device folder. The folder is named specific to the technology of the data points, e.g. LIOB for all L-IOB I/Os that are exposed as a PLC variable. An example is shown in Figure 25.

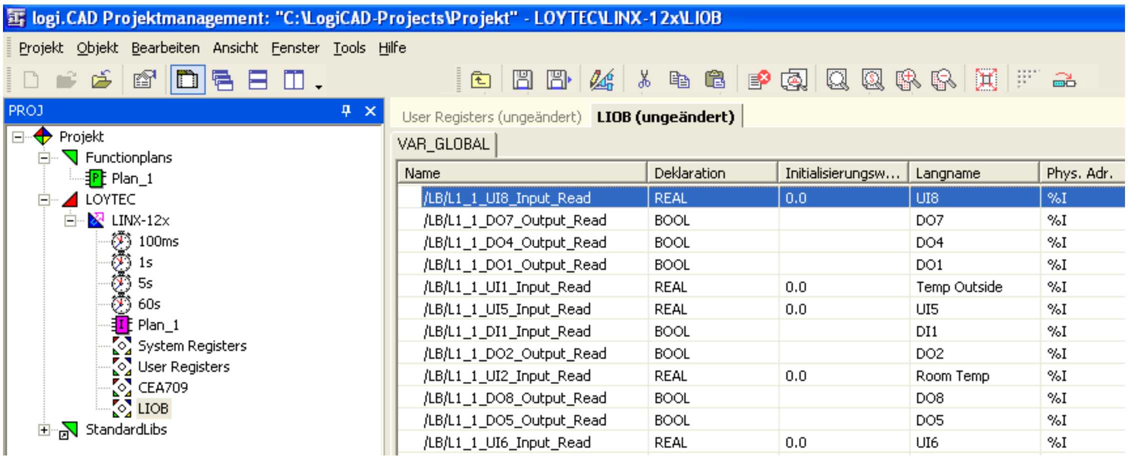


Figure 25: Exposed PLC data points appear in LogiCAD

11. Now the logic can be developed on the function plan.

12. For later debugging, it is good practice to add online test fields to the drawing, to display the current value of the signals during online test. To do this, right-click on the value output of the left function block and select **Create OLT Field** from the context menu, as shown in Figure 26.

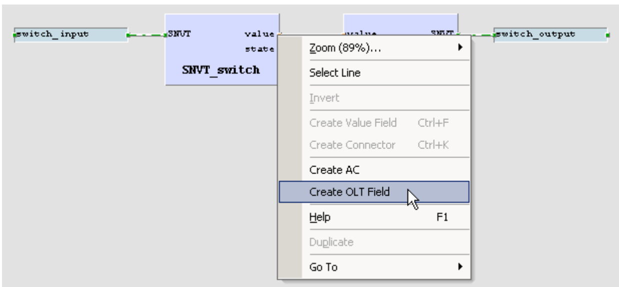


Figure 26: Create online test fields

13. Place the fields above and below the drawing as shown in Figure 27, then press the **Save** button to save your changes.

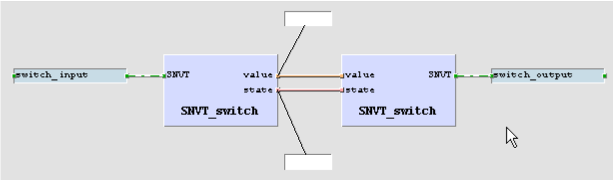


Figure 27: Online Test fields

14. Finally, open the context menu of the **LINX-11x** device again and select **Code Generation**. In the dialog, press the button **Start** to start the code generation process. On success, the code generation window reports Errors=0 and Warnings=1.

15. Close the window by pressing the **OK** button. Now the compiled IEC61131 program can be downloaded to the device. Right-click the tree element **LINX-11x** and select **Download** from the context menu. A connection dialog will appear and ask for the type of connection and additional information.

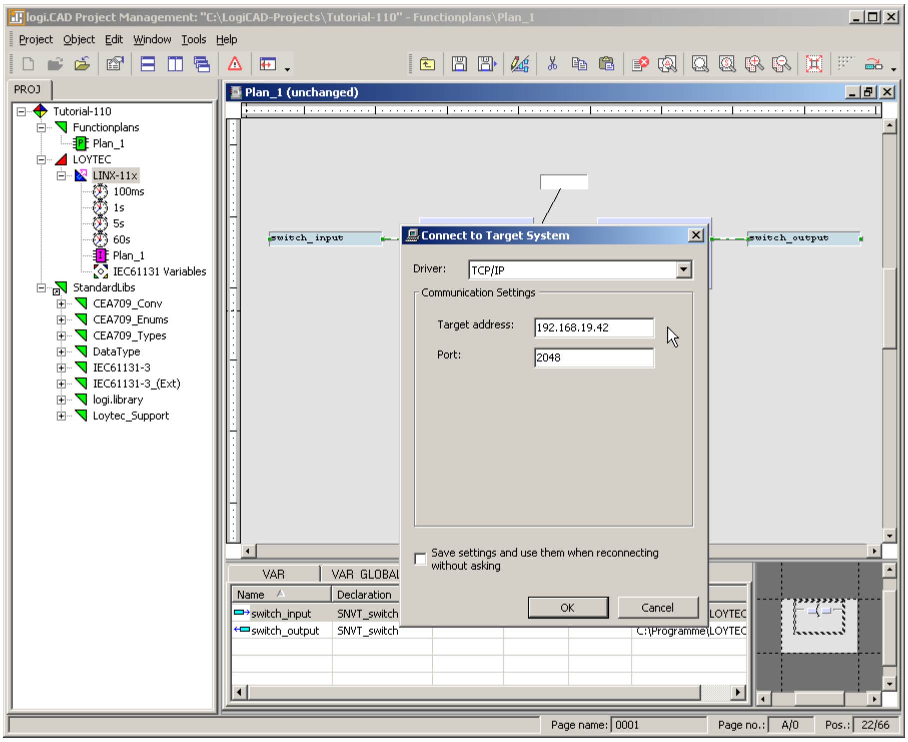


Figure 28: IEC61131 program download

16. Select the TCP/IP communication driver and enter the target address as configured in Section 3.2. Start the download process by pressing the **OK** button.

17. After the download completed check the PLC LED. If it is steady green, the logic is not yet running. Reboot the L-INX to start the IEC61131 program.

3.7 Connect with an OPC XML-DA Client

After the configuration has been downloaded to the L-INX it is ready to serve OPC XML-DA clients. All data points with the **OPC** check box activated will be exposed. Connect to the L-INX using the URL

http://192.168.24.99/DA,

given that 192.168.24.99 is the IP address of the device. Note, that by default, writing to OPC tags needs basic HTTP authentication using the password for the operator user. This is ‘operator’ by default.

3.8 Reset to Factory Defaults

In case the password of the device or the PIN code of the LCD UI has been forgotten you may need to reset the device back to factory defaults to gain access again. On the L-INX models 10X, 11X, 20X, 21X press the service button and power-cycle the device. On the L-INX models with the jog dial press the jog dial and power-cycle the device. Keep the button/jog dial pressed until the port LEDs illuminate orange permanently. Release the button/jog dial within five seconds from that time on to reset the device to factory defaults.

4 Hardware Installation

4.1 Enclosure

4.1.1 LINX-100/101/110/111

The LINX-100/101/110/111 enclosure is 107 mm wide for DIN rail mounting, following DIN 43 880 (see Figure 29).

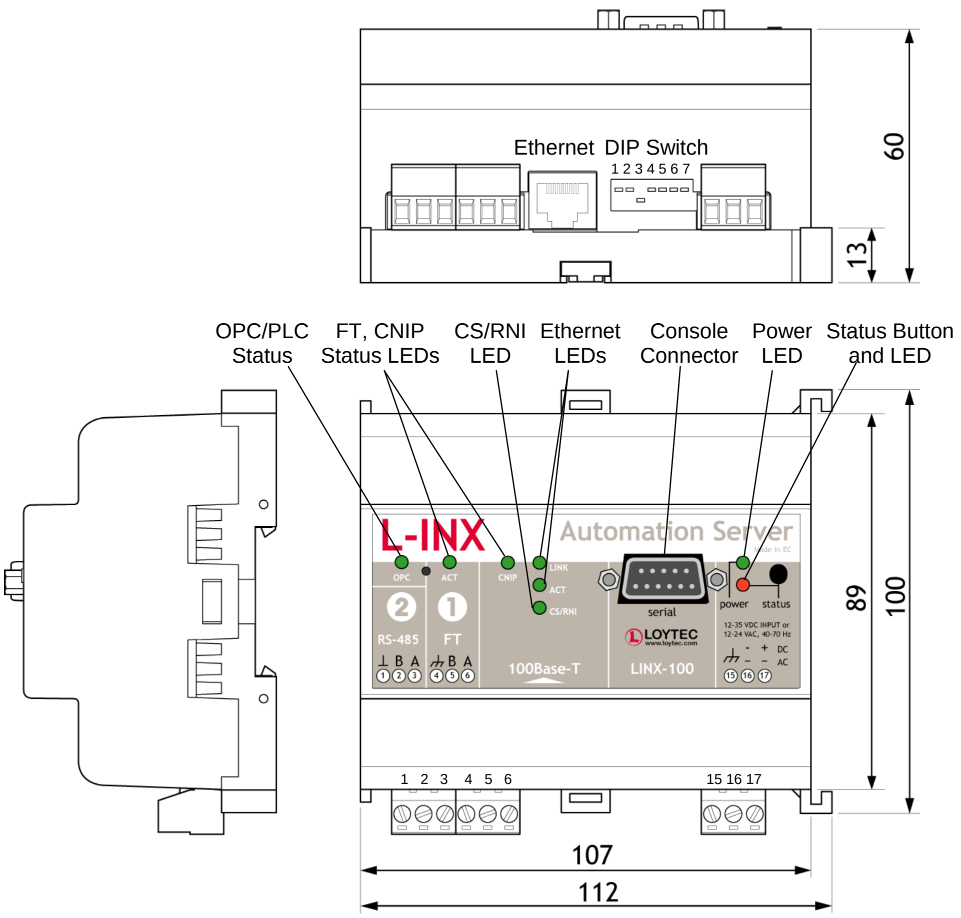


Figure 29: LINX-10X/11X Enclosure (dimensions in mm).

4.1.2 LINX-12X/15X

The LINX-12X/15X enclosure is 159 mm wide for DIN rail mounting, following DIN 43 880 (see Figure 30).

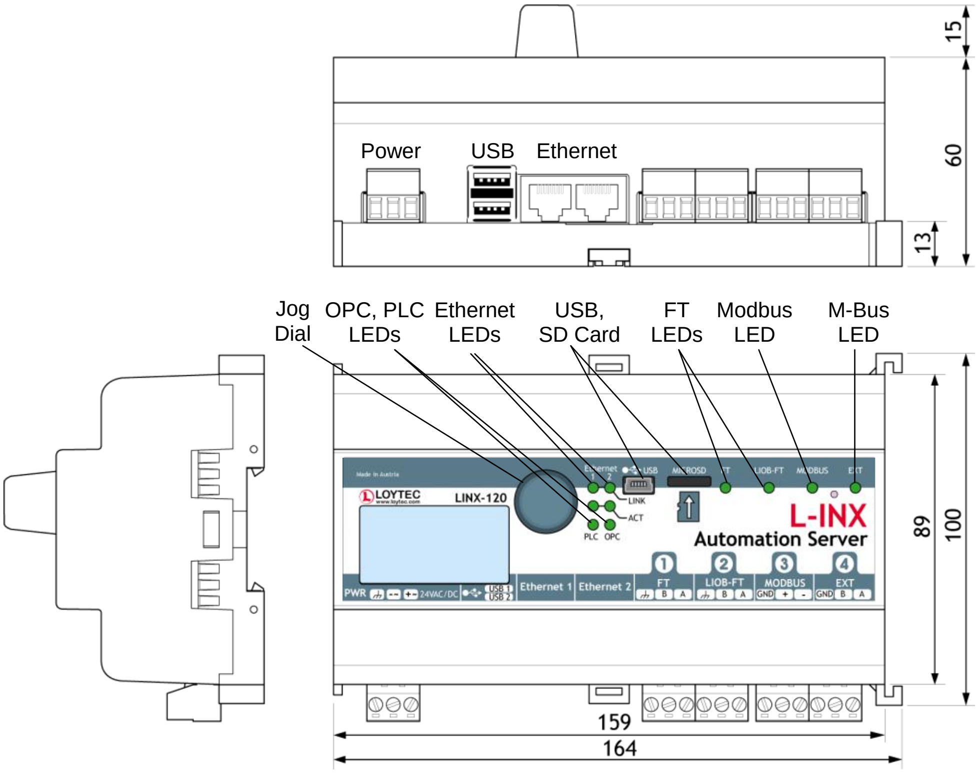


Figure 30: LINX-12X/15X Enclosure (dimensions in mm).

4.1.3 LINX-200/201/210/211

The LINX-200/201/210/211 enclosure is 107 mm wide for DIN rail mounting, following DIN 43 880 (see Figure 31).

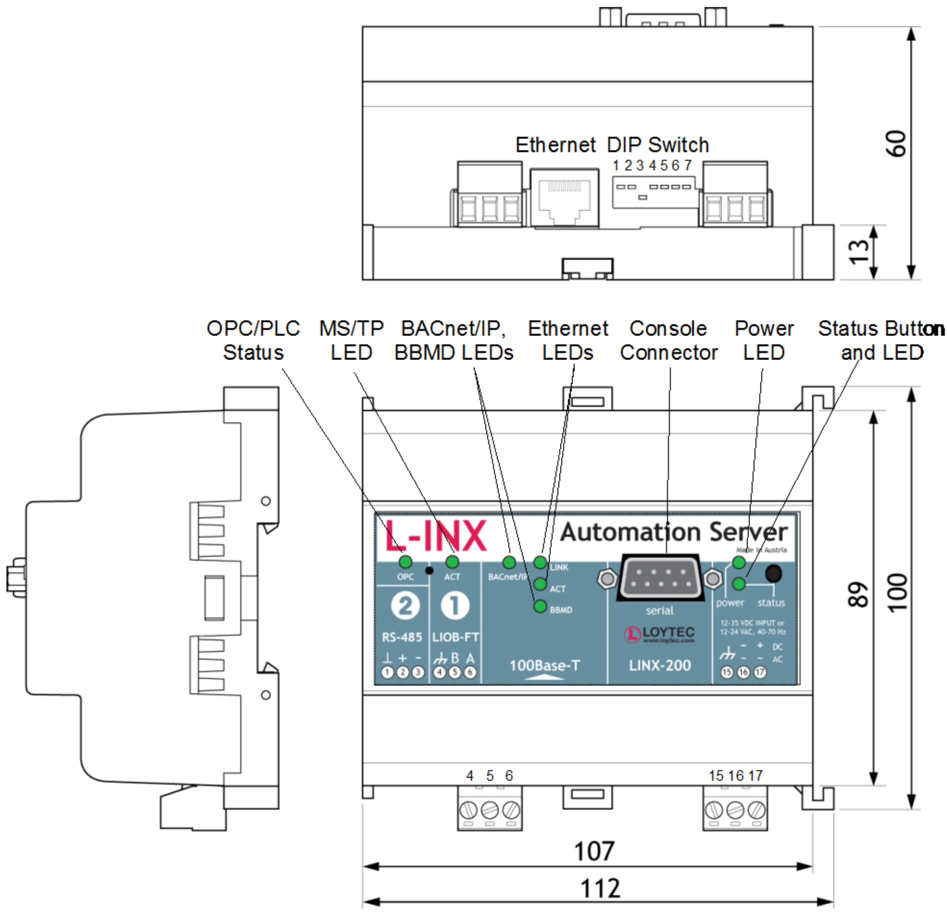


Figure 31: LINX-20X/21X Enclosure (dimensions in mm).

4.1.4 LINX-22X

The LINX-22X enclosure is 159 mm wide for DIN rail mounting, following DIN 43 880 (see Figure 32).

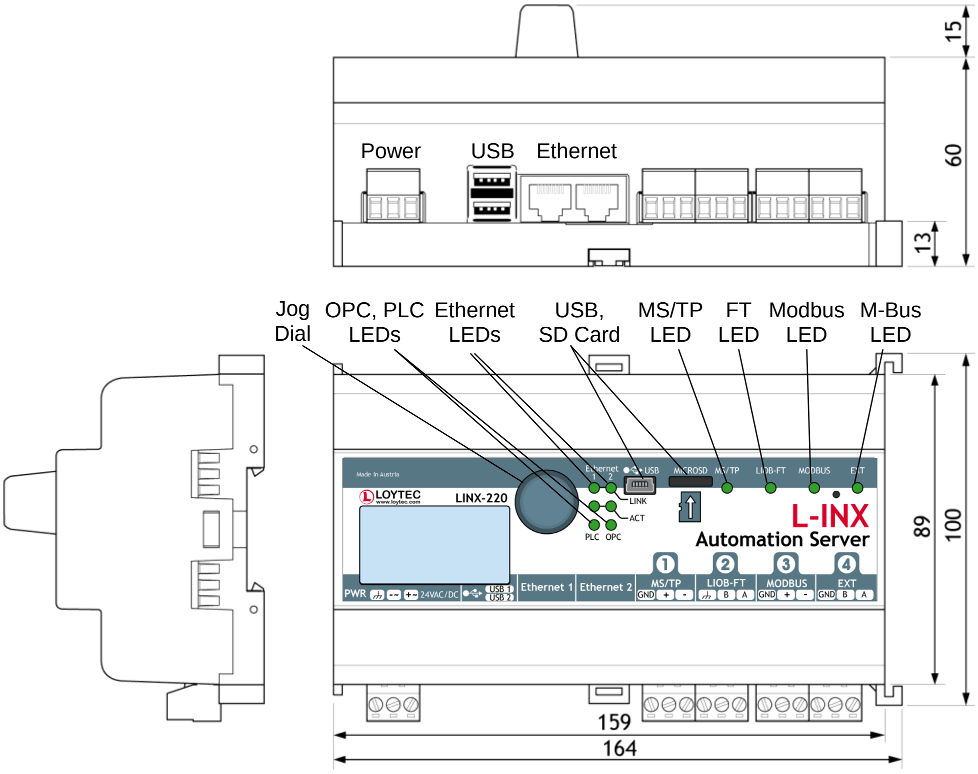


Figure 32: LINX-22X Enclosure (dimensions in mm).

4.1.5 LINX-102/103/112/113/202/203/212/213

The models LINX-102/103/112/113/202/203/212/213 are shipped in the LINX2 enclosure. The LINX2 enclosure is 107 mm wide for DIN rail mounting, following DIN 43 880 (see Figure 33).

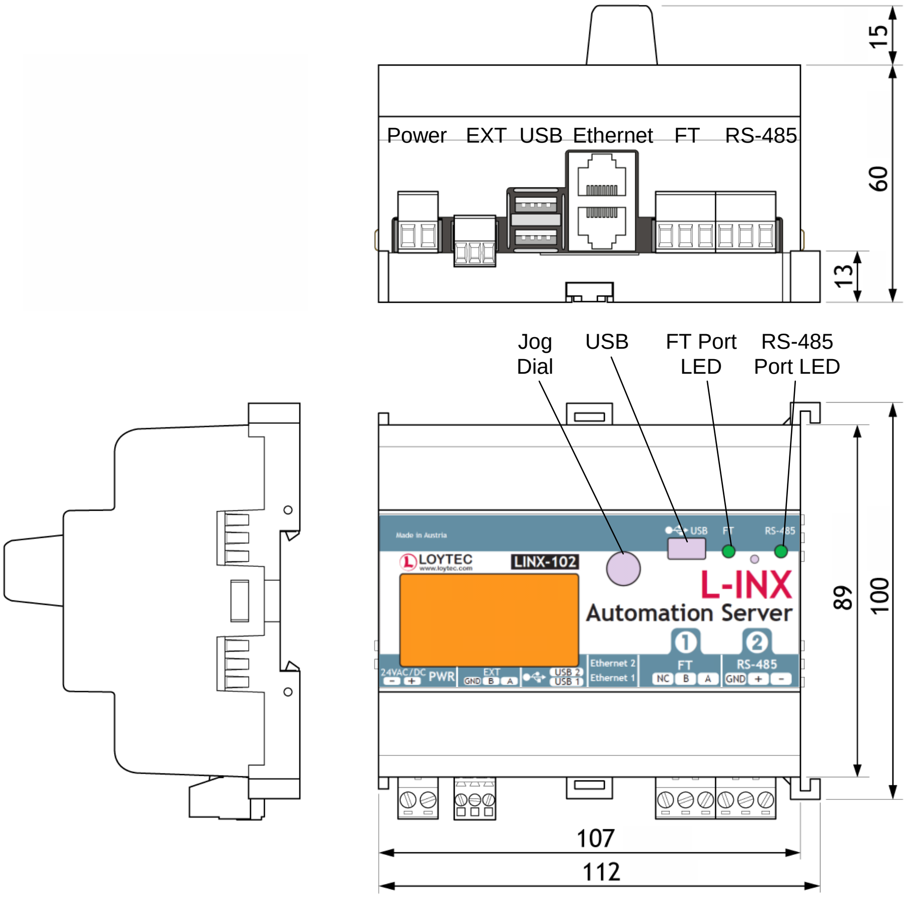


Figure 33: LINX2 Enclosure (dimensions in mm).

4.1.6 LGATE-900

The LGATE-900 enclosure is 107 mm wide for DIN rail mounting, following DIN 43 880 (see Figure 34).

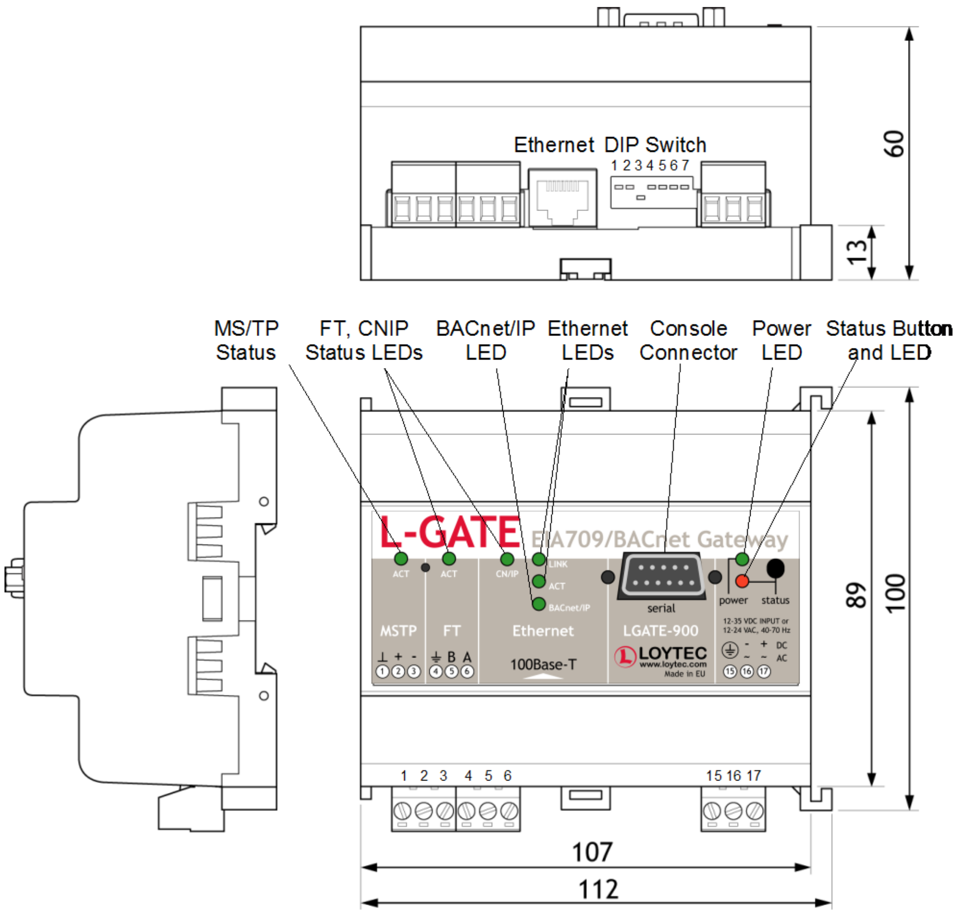


Figure 34: LGATE-900 Enclosure (dimensions in mm)

4.1.7 LGATE-902

The LGATE-902 enclosure is 107 mm wide for DIN rail mounting, following DIN 43 880 (see Figure 35).

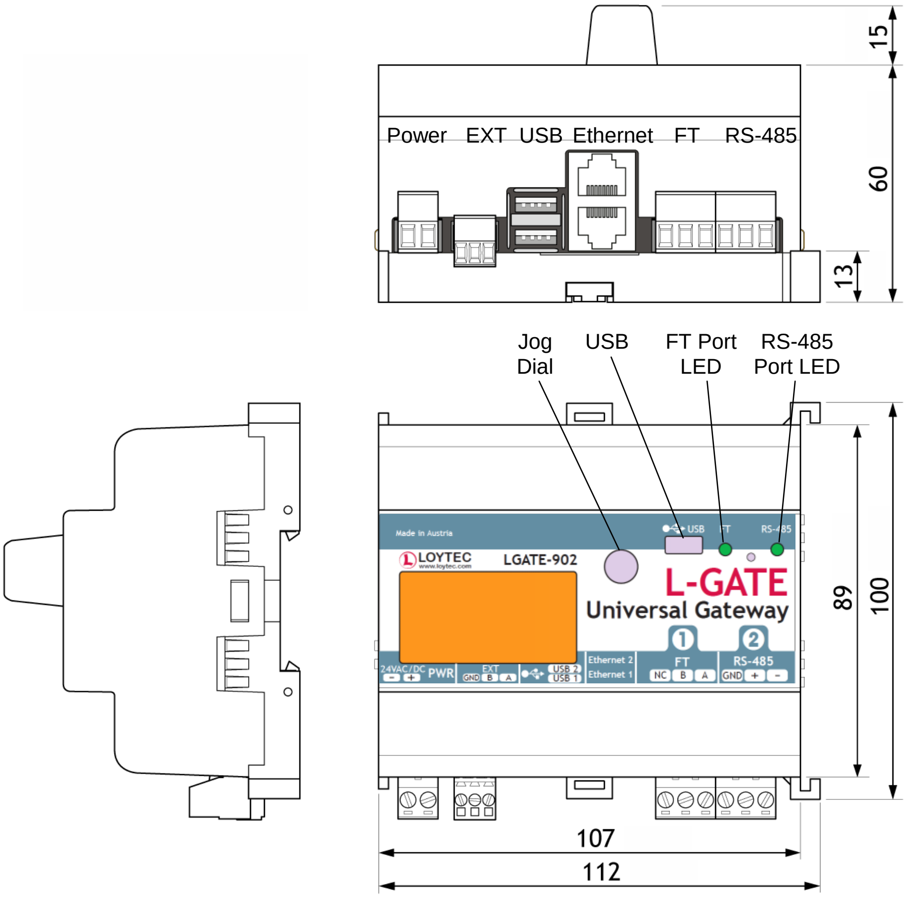


Figure 35: LGATE-902 Enclosure (dimensions in mm).

4.1.8 LGATE-95X

The LGATE-95X enclosure is 159 mm wide for DIN rail mounting, following DIN 43 880 (see Figure 36). The LGATE-951 has a second EXT port on port 3.

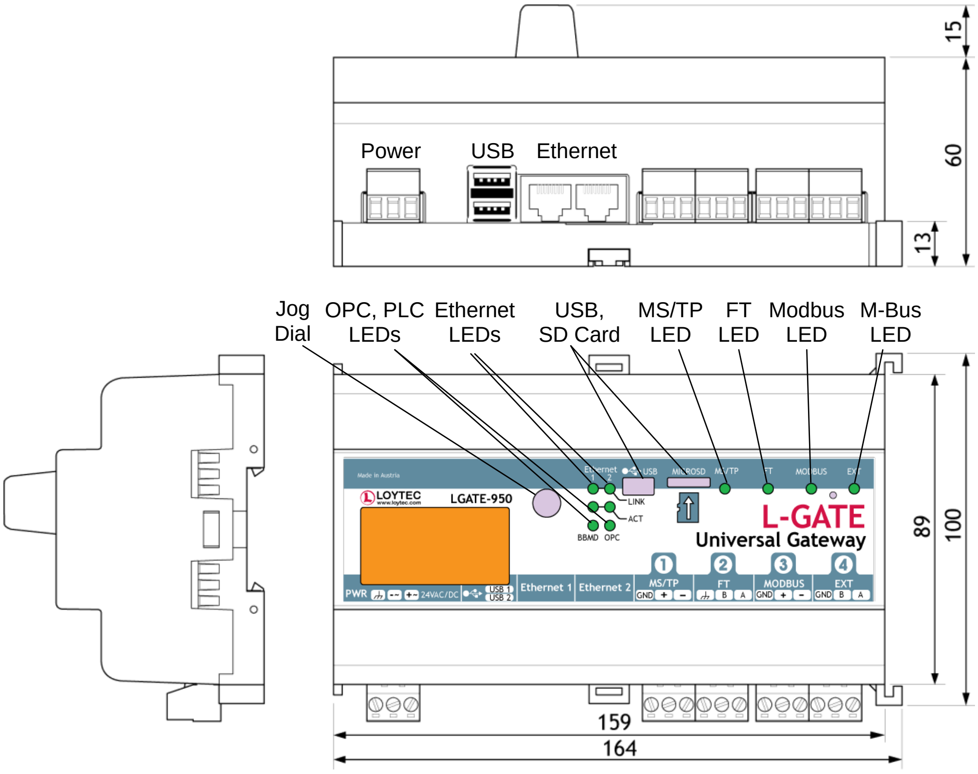


Figure 36: LGATE-950 Enclosure (dimensions in mm).

4.2 Product Label

4.2.1 LINX-10X/11X

The product label on the side of the LINX-10X/11X contains the following information (see Figure 37):

- L-INX order number with bar-code (e.g., LINX-100, LINX-111),
- serial number with bar-code (Ser#),
- unique node ID and virtual ID of each port (NID1, VID1) with bar-code,
- Ethernet MAC ID with bar-code (MAC1).

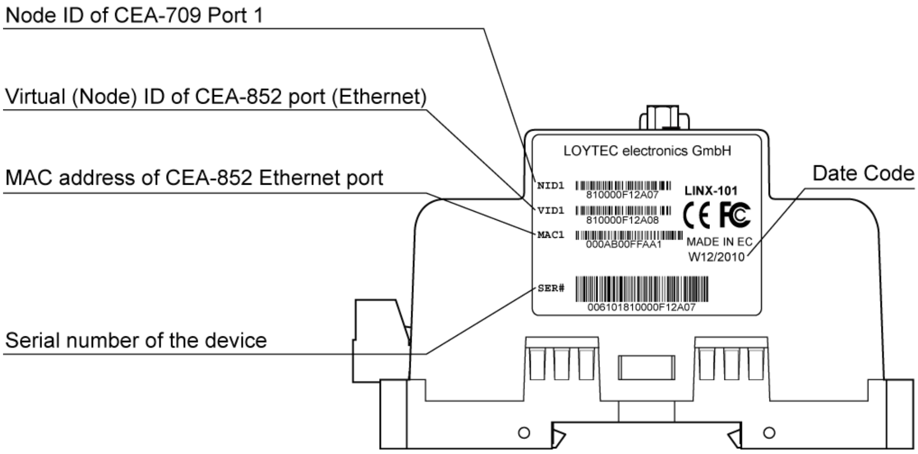


Figure 37: LINX-10X/11X product label.

Unless stated otherwise, all bar codes are encoded using "Code 128". An additional label is also supplied with the L-INX for documentation purposes. A virtual ID (VID) is a Node ID on the IP channel.

4.2.2 LINX-12X/15X

The product label on the side of the LINX-12X/15X contains the following information (see Figure 38):

- L-INX order number with bar-code (e.g., LINX-120, LINX-151),
- serial number with bar-code (Ser#),
- unique node ID and virtual ID of each port (NID1, VID1) with bar-code,
- Ethernet MAC ID with bar-code (MAC1).

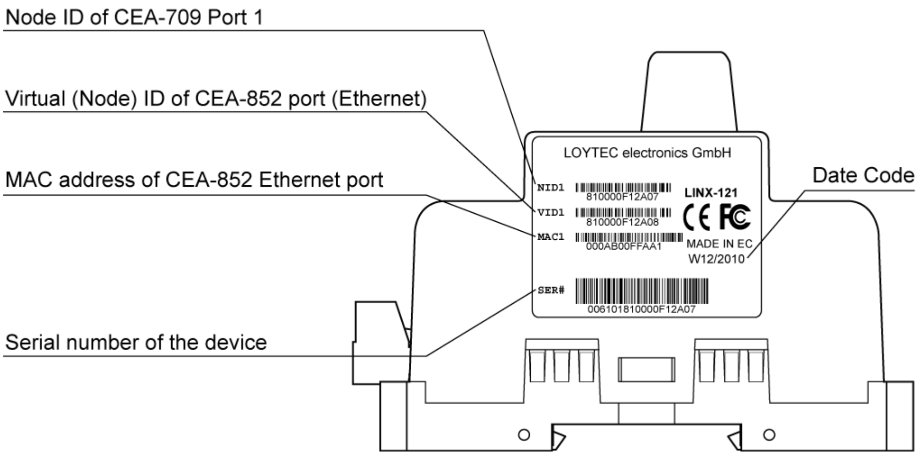


Figure 38: LINX-12X/15X product label.

Unless stated otherwise, all bar codes are encoded using "Code 128". An additional label is also supplied with the L-INX for documentation purposes. A virtual ID (VID) is a Node ID on the IP channel.

4.2.3 LINX-20X/21X

The product label on the side of the LINX-20X/21X contains the following information (see Figure 39):

- L-INX order number with bar-code (e.g., LINX-200, LINX-211),

- Date Code, which defines the production week and year,
- Serial number with bar-code (Ser#),
- MAC address of Ethernet port with bar-code (MAC1).

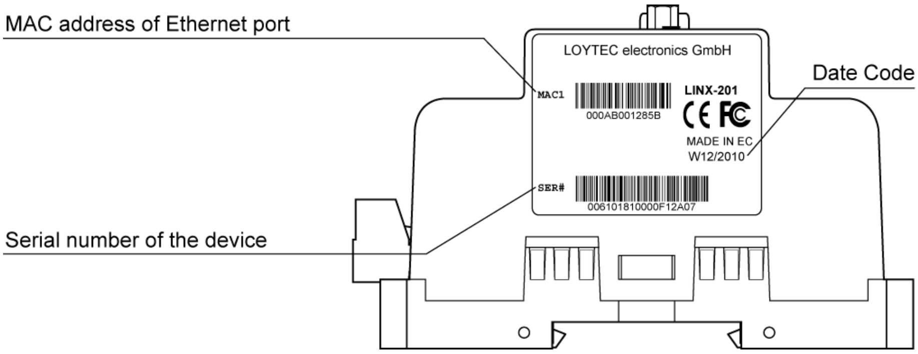


Figure 39: LINX-20X/21X product label.

Unless stated otherwise, all bar codes are encoded using “Code 128”. An additional label is also supplied with the L-INX for documentation purposes.

4.2.4 LINX-22X

The product label on the side of the LINX-22X contains the following information (see Figure 40):

- L-INX order number with bar-code (e.g., LINX-221),
- Date Code, which defines the production week and year,
- Serial number with bar-code (Ser#),
- MAC address of Ethernet port with bar-code (MAC1).

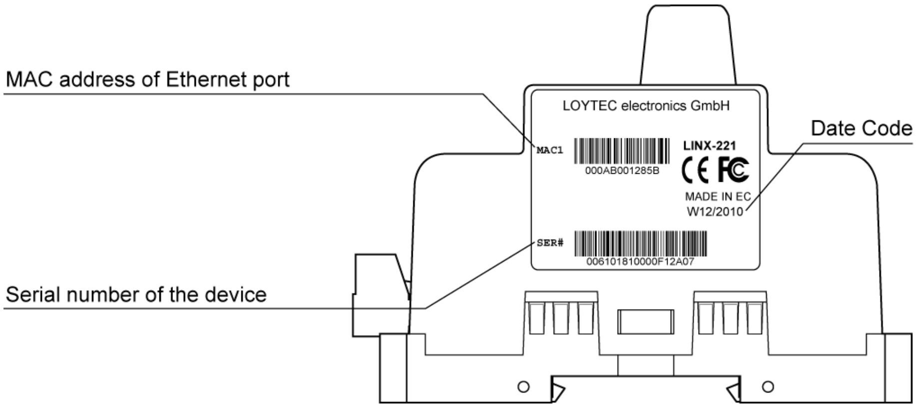


Figure 40: LINX-22X product label.

Unless stated otherwise, all bar codes are encoded using “Code 128”. An additional label is also supplied with the L-INX for documentation purposes.

4.2.5 LINX2

The product label on the side of the LINX2 (for L-INX models 102, 103, 112, 113, 202, 203, 212, 213) contains the following information (see Figure 41):

- LINX2 order number and date code,
- serial number with bar-code (Ser#),

- Ethernet MAC ID with bar-code (MAC1).

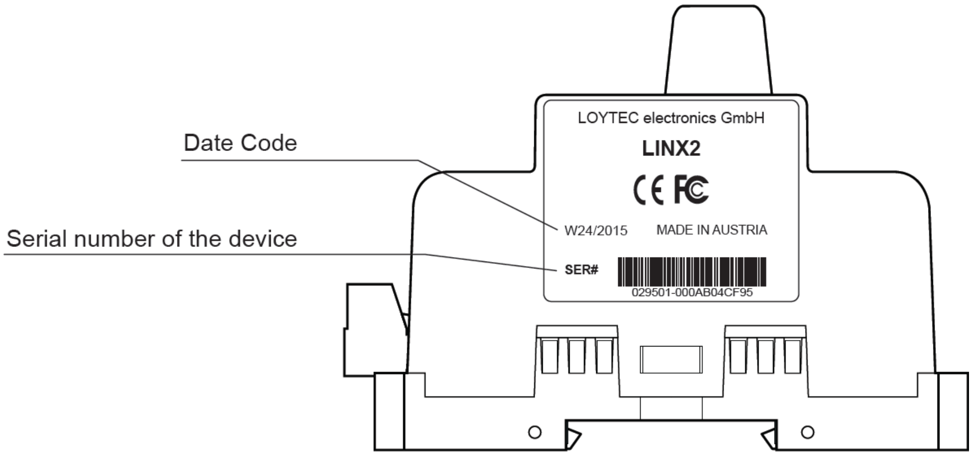


Figure 41: LINX2 product label.

Unless stated otherwise, all bar codes are encoded using “Code 128”. An additional label is also supplied with the device for documentation purposes.

4.2.6 LGATE-900

The product label on the side of the LGATE-900 contains the following information (see Figure 42):

- LGATE-900 order number with bar-code,
- Date Code, which defines the production week and year,
- serial number with bar-code (Ser#),
- unique node ID and virtual ID of each port (NID1, VID1) with bar-code,
- Ethernet MAC ID with bar-code (MAC1).

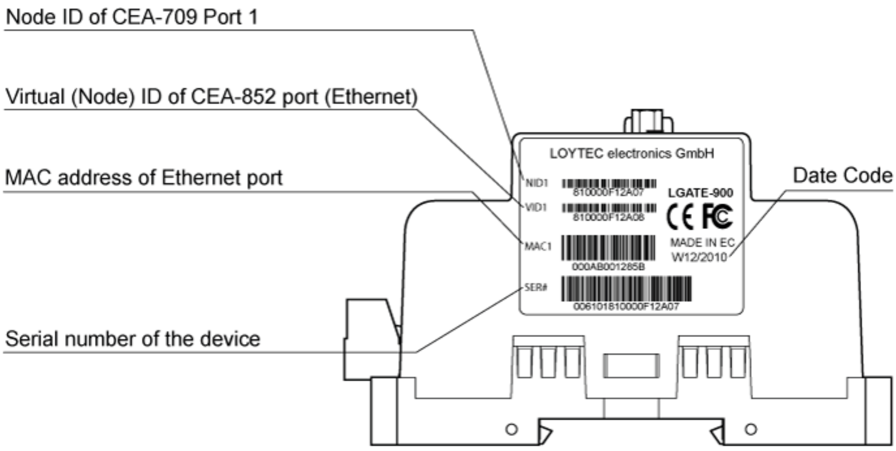


Figure 42: LGATE-900 product label.

Unless stated otherwise, all bar codes are encoded using “Code 128”. An additional label is also supplied with the device for documentation purposes. A virtual ID (VID) is a Node ID on the IP channel.

4.2.7 LGATE-902

The product label on the side of the LGATE-902 contains the following information (see Figure 43):

- LGATE-902 order number and date code,
- serial number with bar-code (Ser#),
- Ethernet MAC ID with bar-code (MAC1).

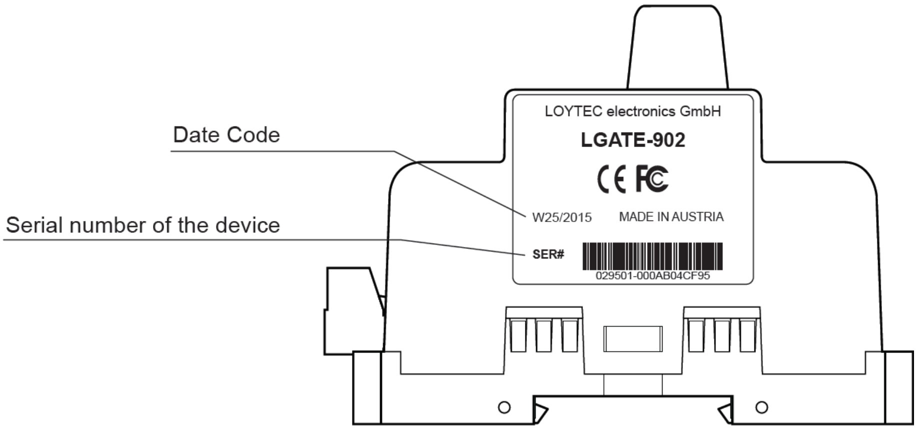


Figure 43: LGATE-902 product label.

Unless stated otherwise, all bar codes are encoded using “Code 128”. An additional label is also supplied with the device for documentation purposes.

4.2.8 LGATE-95X

The product label on the side of the LGATE-95X contains the following information (see Figure 44):

- L-GATE order number with bar-code (e.g., LGATE-950),
- Date Code, which defines the production week and year,
- serial number with bar-code (Ser#),
- unique node ID and virtual ID of each port (NID1, VID1) with bar-code,
- Ethernet MAC ID with bar-code (MAC1).

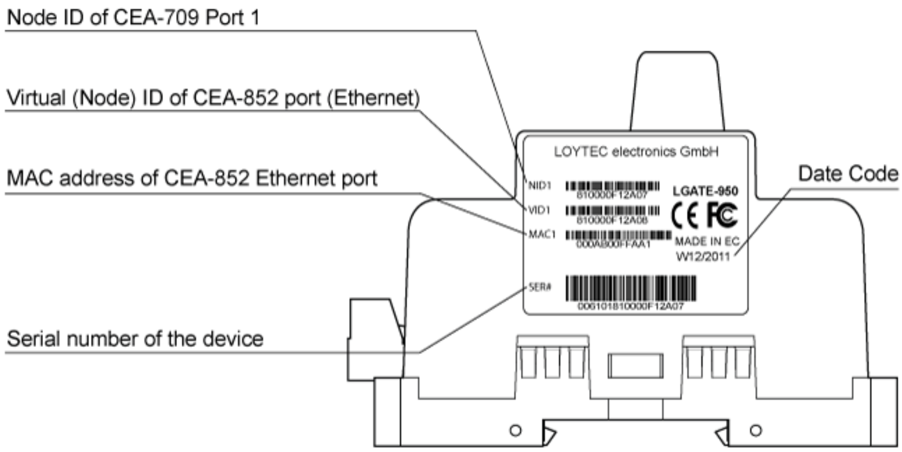


Figure 44: LGATE-95X product label.

Unless stated otherwise, all bar codes are encoded using “Code 128”. An additional label is also supplied with the device for documentation purposes. A virtual ID (VID) is a Node ID on the IP channel.

4.3 Mounting

The device comes prepared for mounting on DIN rails following DIN EN 50 022. The device can be mounted in any position. However, an installation place with proper airflow must be selected to ensure that the device’s temperature does not exceed the specified range (see Chapter 24).

4.4 LED signals

4.4.1 Power LED

The power LED lights up green when power is supplied to the power terminals.

4.4.2 Status LED

This LED is available on L-INX models 10X, 11X, 20X, 21X and on LGATE-900 only. The device is equipped with a red status LED (see Figure 31). This LED is normally off. During boot-up the status LED is used to signal error conditions (red). If the fall-back image is executed the status LED flashes red once every second.

4.4.3 OPC LED

The OPC Server LED illuminates green when at least one OPC client is connected to the OPC server. The LED flickers on OPC XML-DA traffic activity.

4.4.4 PLC LED

The three-color PLC LED indicates the state of the IEC61131 kernel and the IEC61131 program (see Figure 31). Table 3 shows the different LED patterns and their meaning. This LED is not available on the L-INX models 10X, 20X. On L-INX models with a green LED (12X, 15X, 22X) an alternative pattern is used.

Behavior	Description	Comment
GREEN permanent	The IEC61131 program has been stopped	The IEC61131 kernel is running, there is an IEC61131 program, but the program execution was stopped.
GREEN flashing slow at 1 Hz	Normal condition, IEC61131 program running	The IEC61131 kernel and program were successfully loaded. The IEC61131 program is executed.
ORANGE or GREEN flashing fast in an on-on-pause pattern	I/O driver disabled	The I/O driver is disabled, that is that no updates from or the IEC61131 program were handled.
RED or GREEN flashing fast	CPU overload	CPU load exceeded 80%. Modify the PLC program to reduce CPU load in order to guarantee normal system operation (for example, reduce the cycle time of the program).
OFF	No IEC61131 program	No IEC61131 orogram is loaded or a problem occurred while starting the IEC61131 kernel.

Table 3: PLC LED Patterns

4.4.5 FT Activity LED

The FT port on the device has a three-color LED (green, red, and orange, see Figure 31). Table 4 shows different LED patterns of the port and their meaning.

Behavior	Description	Comment
GREEN flashing fast	Traffic	
GREEN flashing at 1Hz	The OPC node or LINX-101's router port is unconfigured	On the LINX-101 this LED only stops flashing if both, node and router, are commissioned.
RED permanent	Port damaged	
RED flashing fast	Traffic with high amount of errors	
RED flashing at 1 Hz (all ports)	Firmware image corrupt	Please upload new firmware.
ORANGE permanent	Port disabled	e.g., using LSD Tool
ORANGE flashing fast	Traffic on port configured as management port	e.g., using LSD Tool

Table 4: CEA-709 Activity LED Patterns.

4.4.6 MSTP Activity LED

The MS/TP port has a three-color MSTP Activity LED (see Figure 31). Table 5 shows the different LED patterns of the port and their meaning. A permanent color reflects a state. Flicker is for 25 ms when there is activity on the MS/TP data link layer.

Behavior	Description	Comment
GREEN permanently, flicker off	Multi-Master, token ok, flicker when traffic	Normal condition on a multi-master MS/TP network.
ORANGE flicker	Sole master, flicker when traffic	Normal condition on a single-master MS/TP network.
RED permanent, flicker GREEN	Token lost state, flicker when transmit attempt	Cable might be broken.
RED flash fast	Transmission or receive errors	This indicates bad cabling.

Table 5: MS/TP Activity LED Patterns.

4.4.7 Modbus LED

The Modbus LED is a three-color LED. Table 6 shows the different LED patterns of the port and their meaning. Flicker is for 25 ms when there is activity on the bus.

Behavior	Description	Comment
GREEN flicker	Traffic, transmission successful	Normal condition.
RED flicker	Traffic, transmission unsusccesful	Device not responding or other network error.

Table 6: Modbus Activity LED Patterns.

4.4.8 EXT LED

The EXT port has a three-color LED (see Figure 31), which displays link and traffic information on the protocol enabled on the port (M-Bus or KNX TP1). Table 7 shows the different LED patterns of the port and their meaning. A permanent color reflects a state. Flicker is for 25 ms when there is activity on the attached network. The LGATE-951 has two EXT ports that conform both to the described behavior.

Behavior	Description	Comment
GREEN permanently, flicker off	LKNX-300 attached, flicker when traffic	Normal condition on a KNX TP1 network.
GREEN flash fast	M-Bus traffic	Normal condition on an M-Bus network.
RED permanent	LKNX-300 not attached, no KNX bus power or port damaged	Check if cabling to LKNX-300 is bad.
RED flash fast	M-Bus transmission or receive errors	This indicates bad cabling. Check if L-MBUS is connected. Check if M-Bus devices are operational.

Table 7: EXT port LED Patterns.

4.4.9 Ethernet Link LED

The Ethernet Link LED lights up green whenever an Ethernet cable is plugged-in and a physical connection with a switch, hub, or PC can be established.

4.4.10 Ethernet Activity LED

The Ethernet Activity LED lights up green for 6 ms whenever a packet is transmitted or received or when a collision is detected on the network cable.

4.4.11 CNIP LED

This LED is available on L-INX models 10X, 11X and on LGATE-900 only. The CNIP LED is a three color LED that indicates different operating states of the device's CEA-852 device.

Green: The CEA-852 device is fully functional and all CEA-852 configuration data (channel routing info, channel membership list, send list) are up-to-date.

Green flicker: If a valid CEA-709 packet is received or transmitted over the IP channel the CNIP LED turns off for 50 ms. Only valid CEA-709 IP packets sent to the IP address of the LINX-10X can be seen. Stale packets or packets not addressed to the device are not seen.

Yellow: Device is functional but some configuration data is not up-to-date (device cannot contact configuration server but has configuration data saved in Flash memory)

Red: Device is non-functional because it was rejected from the CEA-852 IP channel or shut-down itself due to an internal error condition.

Off: Device is non-functional because the CEA-852 device has not started. This can be the case if the device uses DHCP and it has not received a valid IP configuration (address) from the DHCP server.

Flashing red at 1 Hz: Device is non-functional because the CEA-852 device is started but has not been configured. Please add the device to a CEA-852 IP channel (register in configuration server).

Flashing green or orange at 1 Hz: The device's CEA-709 side of the gateway has not been commissioned yet. The color indicates the CEA-852 IP channel status as described above.

4.4.12 CS/RNI LED

This LED is available on L-INX models 10X, 11X only. On the CEA-709 L-INX with the router option this LED indicates the status of the CEA-852 configuration server. If illuminated green, the configuration server is enabled.

On the CEA-709 L-INX without the router option this LED indicates the remote network interface (RNI) status. The LED is dark, if RNI is not supported by this device or the interface is not enabled. The LED is green, if the RNI is currently in use.

4.4.13 BACnet/IP LED

This LED is available on L-INX models 20X, 21X and on LGATE-900 only. The BACnet/IP LED flashes green for 25 ms when BACnet packets are transmitted or received over the BACnet/IP interface.

4.4.14 BBMD LED

This LED is available on L-INX models 201, 211 and on LGATE-95X only. The BBMD LED is permanent green if BBMD is enabled. Otherwise, it is off.

4.4.15 Wink Action

If the CEA-709 device receives a wink command on any of its network ports, it shows a blink pattern on the CNIP and the CEA-709 activity LEDs. The CEA-709 activity and the CNIP LED turn green/orange/red (each 0.15 s). This pattern is repeated six times. After that, the CNIP LED flashes orange six times if the wink command was received on the IP channel or the CEA-709 activity LED flashes orange six times if the wink command was received on the CEA-709 channel. After that the device's LEDs resume their normal behavior.

4.4.16 Network Diagnostics

The CEA-709 device provides simple network diagnostics via its CEA-709 activity LED:

If the LED does not light up at all, this port is not connected to any network segment or the connected network segment currently shows no traffic.

If the LED is flashing green, the network segment connected to this port is ok.

If the LED is flashing red, a potential problem exists on the network segment connected to this port. This state is referred to as overload condition.

A port overload condition occurs if

- the average bandwidth utilization of this port was higher than 70 %, or
- the collision rate was higher than 5 %, or
- more than 15 % CRC errors have occurred on a port with a power-line transceiver, or more than 5 % on a port with a transceiver other than power-line, or
- the device was not able to process all available messages.

For a deeper analysis of the reason for the overload condition, it is recommended to use a protocol analyzer (e.g., LOYTEC's LPA) or a similar tool. The exact reason of the overload condition can also be determined with the LSD Tool.

4.5 Status Button

The L-INX models 10X, 11X, 20X, 21X and the LGATE-900 are equipped with a status button (see Figure 31). When pressing the status button shortly during normal operation of the device, it sends a "Service Pin Message" on the active CEA-709 network port (FT or CEA-852) and a BACnet "I Am" message on all active BACnet data link layers. As an alternative to pressing the status button, a service pin message can be sent via the Web interface (see Section 5.1).

The status button can also be used to switch the device back to factory default state. Press the service button and power-cycle the device. Keep the button pressed until the port LEDs illuminate orange permanently. Release the button within five seconds from that time on to reset the device to factory defaults. Alternatively, the device can be switched back to factory defaults over the console UI (see Section 21.2.2).

4.5.1 Resetting Forwarding Tables

This function is available on CEA-709 models with the router option. In order to reset the forwarding tables of the device's router, the status button needs to be pressed for at least 20 seconds during normal operation of the device. Resetting forwarding tables means:

- Resetting the CEA-709 transceiver to the standard values.
- Setting all ports to unconfigured.
- Clearing the group forwarding, the subnet/node forwarding and the router domain table when used in smart switch mode.
- Clearing the device status and statistic data.
- But **does not** clear the IP address, the CEA-852 configuration settings, and the data point configuration.

All this is done when the button is released. Afterwards a reset is performed to let the changes take effect.

Important: ***If the L-INX is moved from one location to another or if major changes to the configuration of the network are made, it is recommended to reset the configuration to factory defaults.***

Important: ***Wait at least 30 seconds after power-up of the device before pressing the Status Button to ensure that the device has booted properly!***

4.6 LCD Display and Jog Dial

Device models with an LCD display can also be configured to their basic settings through jog dial navigation on the LCD UI. The main page of the LCD UI is shown in Figure 45. It displays the device’s IP address, hostname, CPU load, system temperature and supply voltage. On devices that don’t have Ethernet link LEDs, the LCD display shows the link status as **Eth1+2** or a respective combination thereof.

Below are menu items. Turn the jog dial to navigate between menu items and press to enter a menu or go into selection mode. When in selection mode turn the jog dial to alter the value and press again to quit the selection. The **Datapoints »»** menu allows browsing through the data points on the device.

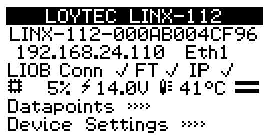


Figure 45: Main Screen of the LCD UI.

The **Device Settings »»** menu allows configuring basic device settings. Navigate to the **Device Management »»** sub-menu, which is displayed in Figure 46.

Figure 46: Device Management Menu on the LCD UI.

This menu gives you the following options for basic device configuration:

- **TCP/IP Setup**: This menu allows configuring the device’s IP address.

- **HTTP Server**: This menu allows to enable/disable the HTTP server and to configure its TCP port.

- **HTTPS Server**: This menu allows to enable/disable the HTTP server, to configure its TCP port and to remove an installed certificate.

- **Date/Time**: This menu allows setting the system time. A time synchronization mechanism can be chosen, and the UTC offset and daylight savings can be defined.

- **Send ID messages**: When selecting this menu, the device sends out service pin, BACnet I-Am, and identification broadcasts for finding the device in the L-Config tool on all applicable ports.

- **Reload config**: By choosing this menu, the device performs a quick restart by reloading its configuration only.

- **Reboot system**: By choosing this menu, the device performs a full reboot.

- **Clear DP config**: By choosing this menu, the user can clear the device's entire data point configuration. This is equivalent to the same Web UI function. The IP address as well as other settings needed to reach the device are not deleted.

- **Reset pers.values**: By choosing this menu, the user can clear all persistent values on the device. They are reset to default values if defined.

- **Factory Defaults**: By choosing this menu, the user can reset the entire device to its factory default. Also IP addresses are cleared.

- **Remote Config**: When enabling this option, the LWEB-822/900 master device manager restores the last saved configuration to the discovered device, if it has no configuration yet. This feature is beneficial when replacing a device.

- **PIN**: Alter the default PIN to any 4-digit number to protect certain operations on the LCD UI. The user will be prompted to enter the PIN on protected areas.

- **Contrast**: This menu allows adjusting the display's contrast.

- **Language**: By choosing this menu, the user can switch between languages on the LCD display.

The **Device Settings »»** menu also allows configuring basic BACnet settings. Navigate to the **BACnet »»** sub-menu, which is displayed in Figure 46.

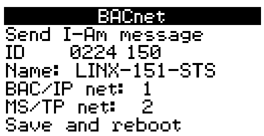


Figure 47: BACnet Menu on the LCD UI.

This menu gives you the following options for basic BACnet configuration:

- **Send I-Am message**: This menu allows sending an I-Am message to the BACnet network.

- **ID**: Use this menu to enter the BACnet device ID. Choose the first four digits then move on the last three digits.

- **BAC/IP Net**: On a BACnet router use this setting to specify the BACnet network number on the BACnet/IP port.

- **MS/TP Net**: On a BACnet router use this setting to specify the BACnet network number on the MS/TP port. If the device has more than one MS/TP port this menu is available for each MS/TP port. To disable the router port, scroll down till **off** appears.

4.7 DIP Switch Settings

The DIP switch assignment for the device is shown in Table 8. Please leave all switches at default state. Note, that the L-INX models 12X, 15X, 22X and the LGATE-95X do not have DIP switches.

DIP Switch #	Function	Factory Default
1	Must be OFF	OFF
2	Must be OFF	OFF
3	Must be ON	ON
4	Must be OFF	OFF
5	Must be OFF	OFF
6	Must be OFF	OFF
7	M-Bus enable	OFF

Table 8: DIP Switch Settings.

4.8 Terminal Layout and Power Supply

4.8.1 LINX-10X/11X

The LINX-10X/11X provides pluggable screw terminals to connect to the network as well as to the power supply. The screw terminals can be used for wires of a maximum thickness of 2.5 mm^2/AWG12. The device can either be DC or AC powered.

Terminal	Function
1	Modbus RS-485 Ground
2	Modbus RS-485 Non-Inverting Input
3	Modbus RS-485 Inverting Input
4	Earth Ground
5, 6	CEA-709 A, B of TP/FT-10 Channel Port
8	Ethernet 100Base-T
15	Earth Ground
16, 17	Power Supply 12 – 35 VDC or 12 – 24 VAC ± 10 % **Do not connect terminal 17 to earth ground!**

Table 9: LINX-10X/11X Terminals.

4.8.2 LINX-20X/21X

The LINX-20X/21X provides pluggable screw terminals to connect to the network as well as to the power supply. The screw terminals can be used for wires of a maximum thickness of 2.5 mm^2/AWG12. The device can either be DC or AC powered.

Terminal	Function
4	BACnet MS/TP / Modbus RS-485 Ground
5	BACnet MS/TP / Modbus RS-485 Non-Inverting Input
6	BACnet MS/TP / Modbus RS-485 Inverting Input
8	Ethernet 100Base-T
15	Earth Ground
16, 17	Power Supply 12 – 35 VDC or 12 – 24 VAC ± 10 % **Do not connect terminal 17 to earth ground!**

Table 10: LINX-20X/21X Terminals.

4.8.3 LGATE-900

The LGATE-900 provides screw terminals to connect to the network as well as to the power supply. The screw terminals can be used for wires of a maximum thickness of 2.5 mm^2/AWG12. The device can either be DC or AC powered.

Terminal	Function
1	BACnet MS/TP Ground
2	BACnet MS/TP Non-Inverting Input
3	BACnet MS/TP Inverting Input
4	Earth Ground
5, 6	CEA-709 A, B of FT-10 Channel Port
8	Ethernet 100BaseT
15	Earth Ground
16, 17	Power Supply 12-35 VDC or 12-24 VAC ± 10% **Do not connect terminal 17 to earth ground!**

Table 11: LGATE-900 Terminals.

4.9 Wiring

4.9.1 LINX-10X/11X

The CEA-709 network segment connected to the LINX-10X/11X needs to be terminated according to the rules found in the specification of the transceiver (see Section 19.1).

Important: ***When using shielded network cables, only one side of the cable should be connected to earth ground. Thus, the shield must be connected to earth ground either at the LINX-10X terminals or somewhere else in the network.***

Important: ***Never connect terminal 17 to earth ground!***

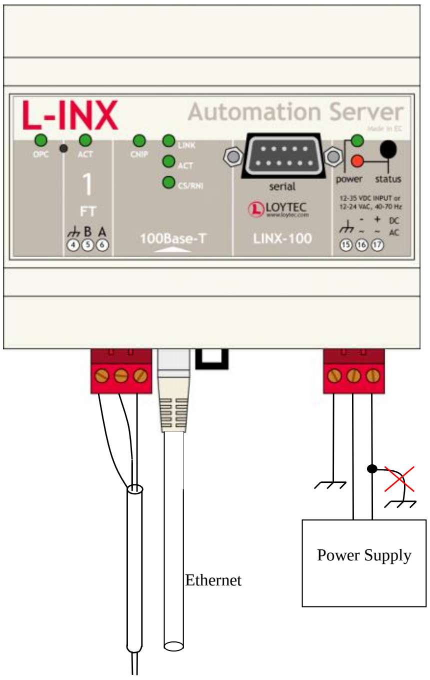
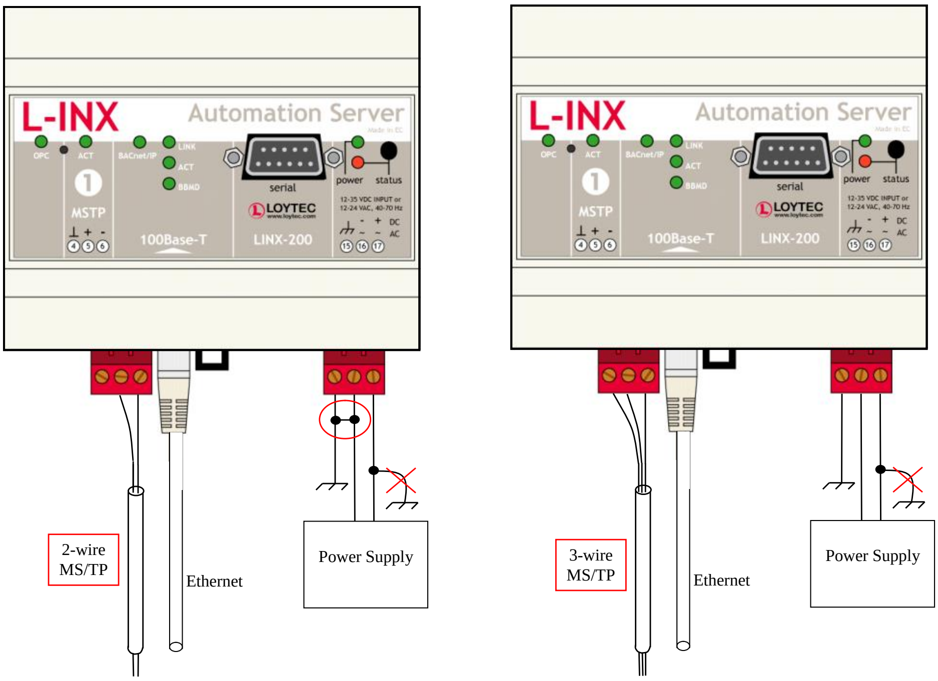


Figure 48: Connecting the LINX-10X/11X.

4.9.2 LINX-20X/21X

If BACnet over MS/TP is enabled, the MS/TP network segment must be properly terminated with an LT-04 network terminator connected at each of the two ends of the segment media.

Important: ***When using 2-wire MS/TP, earth ground must be connected to both terminal 15 and 16 (see Figure 49a). Never connect terminal 17 to earth ground!***

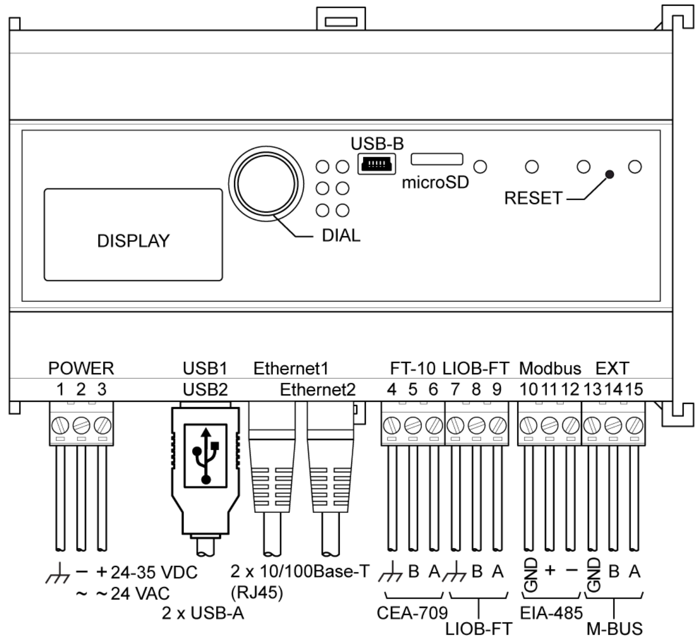


(a) (b)

Figure 49: Connecting the LINX-20X/21X: (a) 2-wire MS/TP, (b) 3-wire MS/TP.

4.9.3 LINX-12X/15X

The terminals and wiring information for the LINX-12X/15X can be seen in Figure 50. The CEA-709 network segment connected to the LINX-10X/11X needs to be terminated according to the rules found in the specification of the transceiver (see Section 19.1).



Connectors		
POWER	1 2 3	Power supply input EARTH GND Power supply input MINUS Can be connected to EARTH GND Power supply input PLUS ⚠ Must NOT be connect to EARTH GND
CEA-709 TP/FT-10	4 5 6	Input EARTH GND Input B Input A
LIOB-FT	7 8 9	Input EARTH GND Input B Input A
Modbus EIA-485	10 11 12	Input GND Input + Input –
EXT M-Bus	13 14 15	Input GND Input B Input A

Connectors		
USB1 USB2 USB-B	USB-A USB-A USB-B	USB Host USB Host PC
Ethernet1 Ethernet2	RJ45 RJ45	10/100Base-T 10/100Base-T
microSD	slot	microSD card max. 2 GB

Buttons	
RESET DIAL	Reset Turn-and-Push Dial

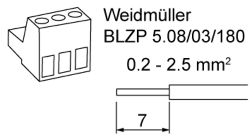
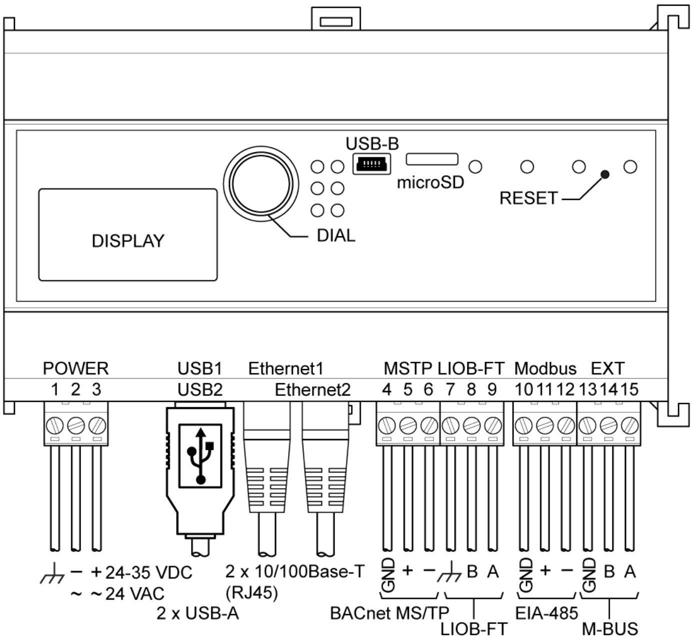


Figure 50: Connecting the LINX-12X/15X.

4.9.4 LINX-22X

The terminals and wiring information for the LINX-22X can be seen in Figure 51. If BACnet over MS/TP is enabled, the MS/TP network segment must be properly terminated with an LT-04 network terminator connected at each of the two ends of the segment media.



Connectors		
POWER	1	Power supply input EARTH GND
	2	Power supply input MINUS Can be connected to EARTH GND
	3	Power supply input PLUS ⚠ Must NOT be connect to EARTH GND
BACnet MS/TP	4	Input GND
	5	Input +
	6	Input –
LIOB-FT	7	Input EARTH GND
	8	Input B
	9	Input A
Modbus EIA-485	10	Input GND
	11	Input +
	12	Input –
EXT M-Bus	13	Input GND
	14	Input B
	15	Input A

Connectors		
USB1	USB-A	USB Host
USB2	USB-A	USB Host
USB-B	USB-B	PC
Ethernet1	RJ45	10/100Base-T
Ethernet2	RJ45	10/100Base-T
microSD	slot	microSD card max. 2 GB

Buttons	
RESET	Reset
DIAL	Turn-and-Push Dial

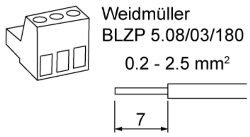
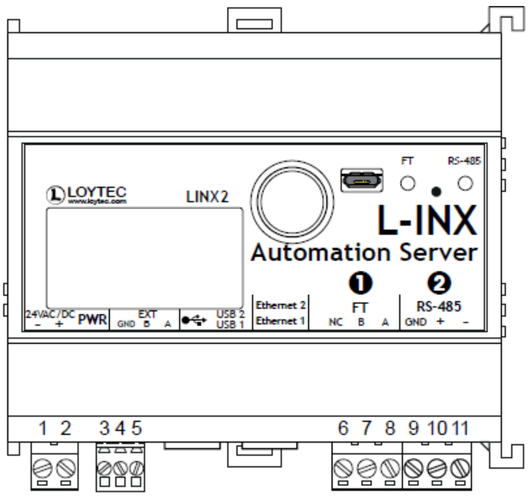


Figure 51: Connecting the LINX-22X.

4.9.5 LINX2

The terminals and wiring information for the LINX2 enclosure (for L-INX models 102, 103, 112, 113, 202, 203, 212, 213) can be seen in Figure 52. The CEA-709 network segment connected to the LINX2 needs to be terminated according to the rules found in the specification of the transceiver (see Section 19.1). If BACnet is configured to run over MS/TP, the MS/TP network segment must be properly terminated with an LT-04 network terminator connected at each of the two ends of the segment media.



Connectors		
POWER	1	Power supply input MINUS
	2	Power supply input PLUS
EXT M-Bus, KNX TP1	3	Input GND
	4	Input B
	5	Input A
TP/FT-10 CEA-709	6	not connected (NC)
	7	Input B
	8	Input A
RS-485 Modbus, BACnet MS/TP	9	Input GND
	10	Input +
	11	Input –

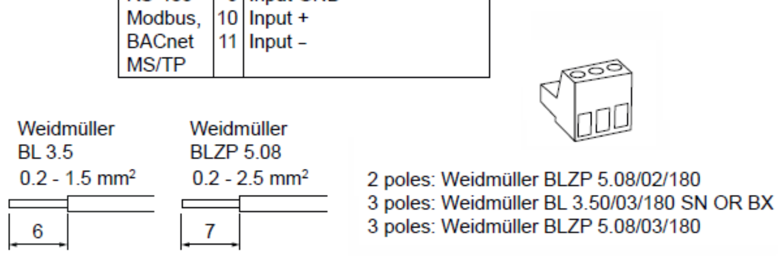


Figure 52: Connecting the LINX2.

4.9.6 LGATE-900

The CEA-709 network segment connected to the LGATE-900 needs to be terminated according to the rules found in the specification of the transceiver (see Section 19.1). If BACnet is configured to run over MS/TP, the MS/TP network segment must be properly terminated with an LT-04 network terminator connected at each of the two ends of the segment media.

Important: ***When using shielded network cables, only one side of the cable should be connected to earth ground. Thus, the shield must be connected to earth ground either at the L-GATE terminals or somewhere else in the network.***

Important: ***When using 2-wire MS/TP, earth ground must be connected to both terminal 15 and 16 (see Figure 49a). Never connect terminal 17 to earth ground!***

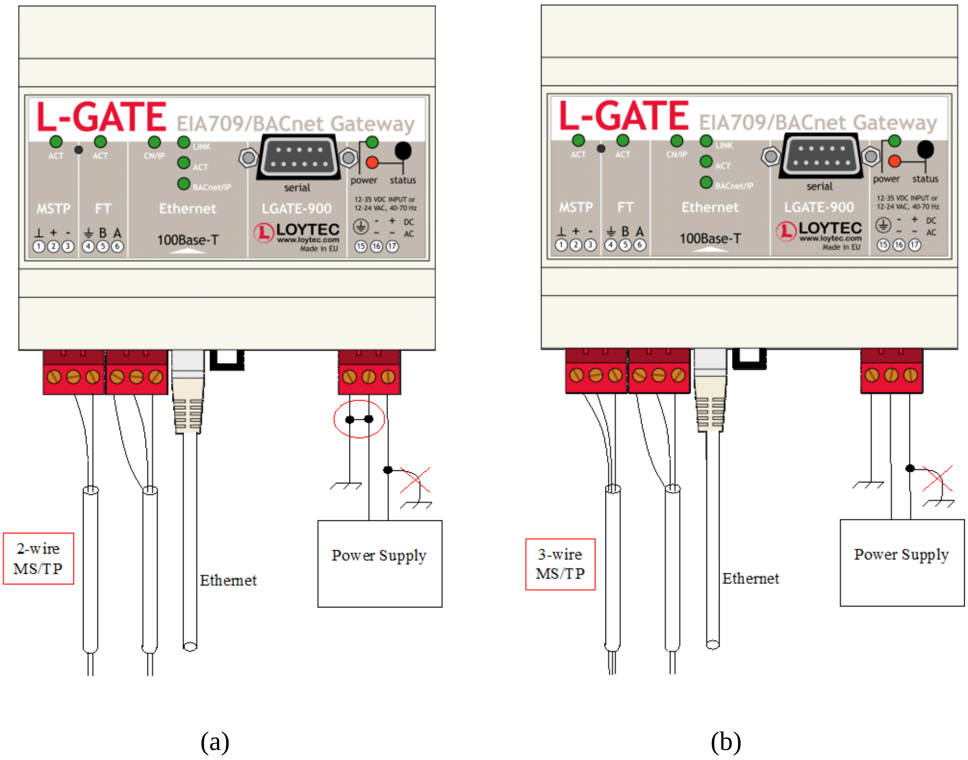
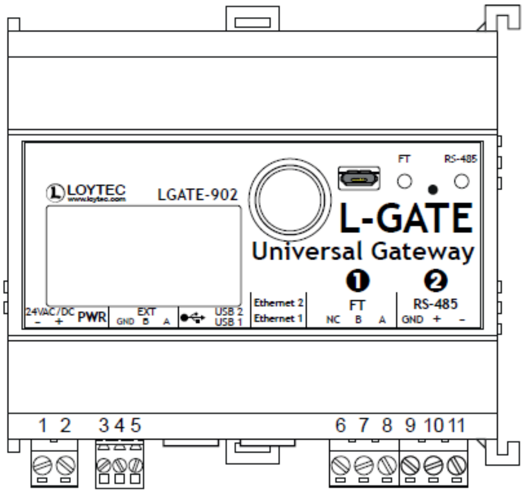


Figure 53: Connecting the LGATE-900: (a) 2-wire MS/TP, (b) 3-wire MS/TP.

4.9.7 LGATE-902

The terminals and wiring information for the LGATE-902 can be seen in Figure 54. The CEA-709 network segment connected to the LGATE-902 needs to be terminated according to the rules found in the specification of the transceiver (see Section 19.1). If BACnet is configured to run over MS/TP, the MS/TP network segment must be properly terminated with an LT-04 network terminator connected at each of the two ends of the segment media.



Connectors		
POWER	1	Power supply input MINUS
	2	Power supply input PLUS
EXT M-Bus, KNX TP1	3 4 5	Input GND Input B Input A
TP/FT-10 CEA-709	6 7 8	not connected (NC) Input B Input A
RS-485 Modbus, BACnet MS/TP	9 10 11	Input GND Input + Input –

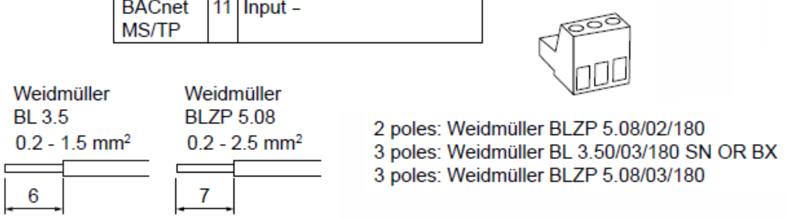
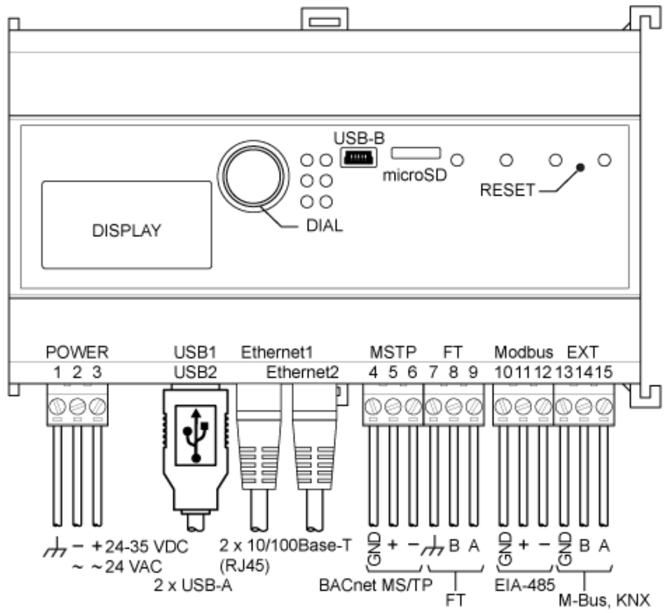


Figure 54: Connecting the LGATE-902.

4.9.8 LGATE-95X

The terminals and wiring information for the LGATE-950 can be seen in Figure 55, for the LGATE-951 in Figure 56. The CEA-709 network segment connected to the LGATE-950 needs to be terminated according to the rules found in the specification of the transceiver (see Section 19.1). If BACnet over MS/TP is enabled, the MS/TP network segment must be properly terminated with an LT-04 network terminator connected at each of the two ends of the segment media.



Connectors		
POWER	1	Power supply input EARTH GND
	2	Power supply input MINUS Can be connected to EARTH GND
	3	Power supply input PLUS ⚠ Must NOT be connect to EARTH GND
BACnet MS/TP	4	Input GND
	5	Input +
	6	Input –
FT	7	Input EARTH GND
	8	Input B
	9	Input A
Modbus EIA-485	10	Input GND
	11	Input +
	12	Input –
EXT M-Bus KNX	13	Input GND
	14	Input B
	15	Input A

Connectors		
USB1	USB-A	USB Host
USB2	USB-A	USB Host
USB-B	USB-B	PC
Ethernet1	RJ45	10/100Base-T
Ethernet2	RJ45	10/100Base-T
microSD	slot	microSD card max. 2 GB

Buttons	
RESET	Reset
DIAL	Turn-and-Push Dial

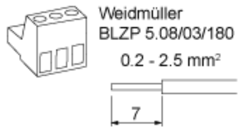
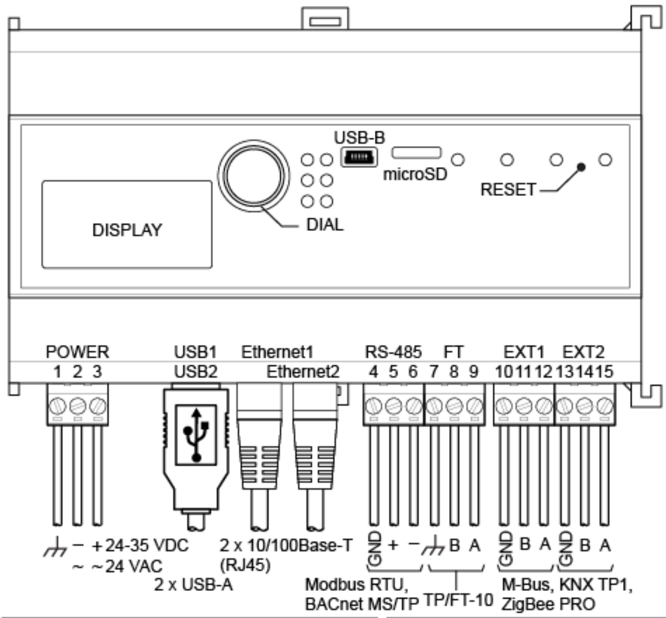


Figure 55: Connecting the LGATE-950.



Connectors		
POWER	1	Power supply input EARTH GND
	2	Power supply input MINUS Can be connected to EARTH GND
	3	Power supply input PLUS ⚠ Must NOT be connect to EARTH GND
Modbus BACnet MS/TP	4	Input GND
	5	Input +
	6	Input –
TP/FT-10	7	Input EARTH GND
	8	Input B
	9	Input A
EXT1 M-Bus, KNX TP1 ZigBee	10	Input GND
	11	Input B
	12	Input A
EXT2 M-Bus KNX TP1 ZigBee	13	Input GND
	14	Input B
	15	Input A

Connectors		
USB1	USB-A	USB Host
USB2	USB-A	USB Host
USB-B	USB-B	PC
Ethernet1	RJ45	10/100Base-T
Ethernet2	RJ45	10/100Base-T
microSD	slot	microSD card max. 2 GB

Buttons	
RESET	Reset
DIAL	Turn-and-Push Dial

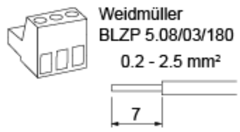


Figure 56: Connecting the LGATE-951.

5 Web Interface

The L-INX/L-GATE comes with a built-in Web server and a Web interface to configure the L-INX and extract statistics information. The Web interface allows configuring the IP settings, CEA-709, CEA-852, BACnet and other configuration settings.

5.1 Device Information and Account Management

In a Web browser, enter the default IP address 192.168.1.254 of the device. Note that if your PC has an IP address in a subnet other than 192.168.1.xxx, you must open a command tool and enter the following route command to add a route to the device.

To Add a Route to the Device

1. Windows **START → Run**
2. Enter ‘cmd’ and click **OK**.
3. In the command window enter the command line
   `route add 192.168.1.254 %COMPUTERNAME%`
   In Windows7 replace %COMPUTERNAME% with the PC's actual IP address.
4. Then open your Web browser and type in the default IP address ‘192.168.1.254’.
5. The device information page should appear as shown in Figure 57.

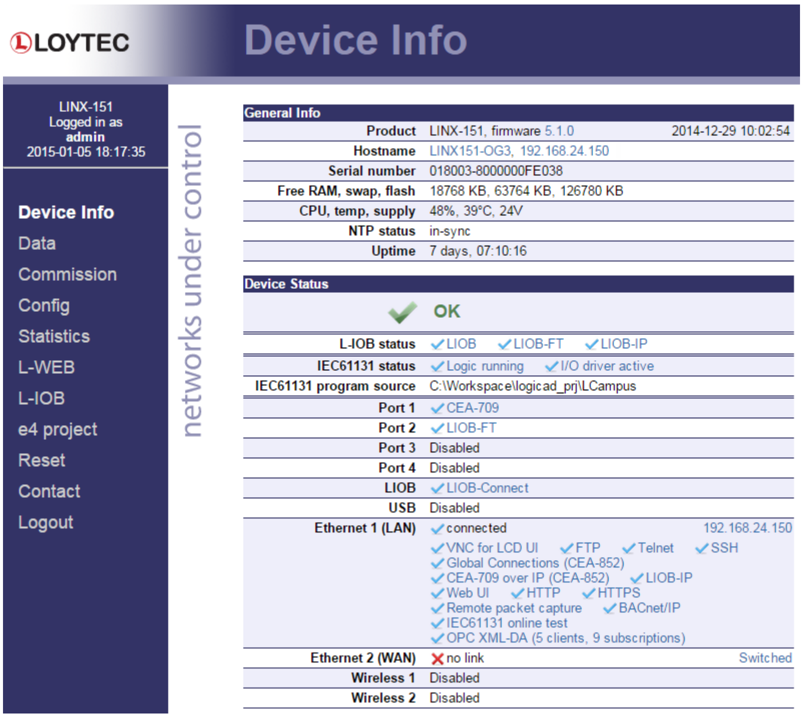


Figure 57: Device Information Page.

The device information page shows some general information about the device in the **General Info** section. This includes the product model and the current firmware version. Below, it shows important operational parameters, such as free memory, CPU load, system temperature and supply voltage, time synchronization status and system uptime.

The **Device Status** section summarizes the status of the various ports and protocols on the device. The summary status is displayed as a green OK checkmark. If any of the interfaces, protocols or operational parameters are non-normal, a warning or error sign is shown instead. L-INX devices show information on the current IEC61131 program. This includes the state of the PLC kernel, the I/O driver and the program source information. Shown below are further a summary on the enabled L-IOB I/O buses and active protocols on the respective ports. All items are links that lead directly to their configuration page.

Below the general status information more specific sections are displayed depending on the model. The **Firmware Info** provides version and build times of the primary and fallback firmware images installed on the device. The **Project Information** area shows details on the currently loaded data point configuration. The **Router Info** and **CEA-709 Application** sections include the unique node IDs (“Neuron IDs”) of the CEA-709 network interfaces. This page can also be used to send the CEA-709 service pin messages. This is a useful feature when commissioning the device, since it is not necessary to be on-site to press the status button.

Click through the menus on the left hand side to become familiar with the different screens. If you click on **Config** in the left menu, you will be asked to enter the administrator password in order to make changes to the settings as shown in Figure 58. Enter the default administrator password ‘loytec4u’ and select **Login**. Note, that firmware versions previous to 4.0 used ‘admin’ as the default password.

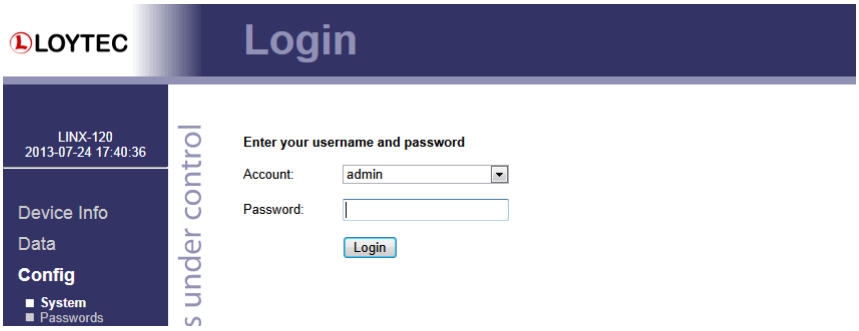


Figure 58: Enter 'loytec4u' as the default administrator password.

The Config menu opens. Click on **Passwords** in the Config menu, which opens the password configuration page as shown in Figure 59. The device has three user accounts: (1) **guest** allows the user to view certain information only, e.g., the device info page. By default the guest user has no password. (2) **operator** is able to read more sensible information such as calendar data. (3) **admin** has full access to the device and can make changes to its configuration. Note that the user accounts are also used to log on to the FTP and Telnet server.

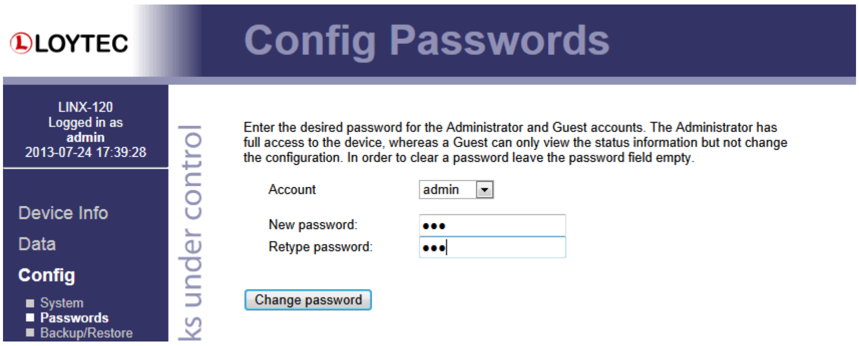


Figure 59: Password Configuration Screen.

Please change the administrator password in order to protect yourself from unwanted configuration changes by anyone else. To do so, select the **admin** account in the drop-down box and enter the new password. If the administrator password is left empty, password protection is turned off and everyone can access the device without entering a password. Click on **Change password** to activate the change.

5.2 Device Configuration

The device configuration pages allow viewing and changing the device settings. Here are some general rules for setting IP addresses, port numbers, and time values:

- An empty IP address field disables the entry.
- An empty port number field sets the default port number.
- An empty time value field disables the time setting.

5.2.1 System Configuration

The system configuration page is shown in Figure 60. This page allows configuring the device's system time and other system settings. The **TCP/IP Configuration** link is a shortcut to the Ethernet port configuration. Follow that link to change the IP settings of the device.

The time sync source can be set to **auto**, **manual**, **NTP**, **BACnet**, or **LonMark**. In the auto mode, the device switches to the first external time source that is discovered. Possible external time sources are NTP, BACnet, LonMark. The option **manual** allows setting the time manually in the fields **Local Time** and **Local Date**. In **manual** mode, the device does not switch to an external time source. Note, that if **NTP** is selected, the NTP servers have to be configured on the IP Configuration page (see Section 5.2.2).

In order to use BACnet as the time source, a BACnet device (time master) must be configured to distribute time synchronization. For doing so, the BACnet address of the devices, which shall be synchronized, must be added to the device object of the BACnet time master (see Section 18.3.8). The device synchronizes automatically as soon as it is contacted by the BACnet time master.

The time zone offset must be defined independently of the time source. It is specified as the offset to GMT in hours and minutes (e.g., Vienna/Austria is +01:00, New York/USA is -06:00). For setting the daylight saving time (DST) predefined choices are offered for Europe and USA/Canada. DST can be switched off completely by choosing **none** or set manually for other regions. In that case, start and end date of DST must be entered in the fields below.

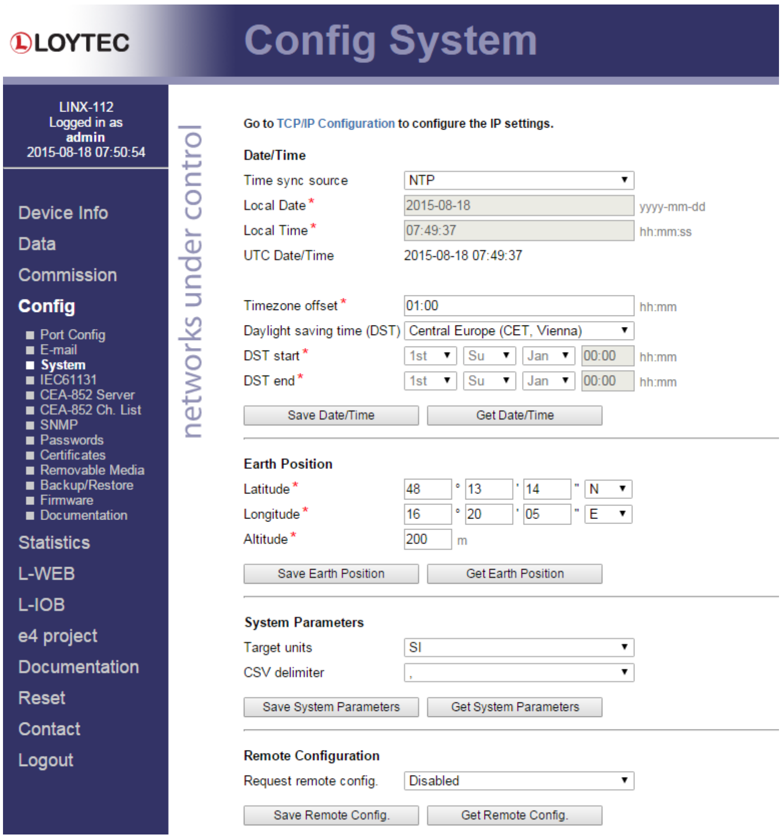


Figure 60: System Configuration Page, e.g., for Vienna, Austria.

The next section on the page allows configuring the device's earth position. This setting defines the longitude, latitude and elevation of the device. The latitude and longitude are entered as degrees, minutes, and seconds. The altitude is entered in meters height above sea level. This setting is used for an astronomical clock. For fixed locations such as a building, the position can be entered on this page. For moving locations, this setting can be updated over the network using the network variable nciEarthPos (see Section 18.2.2) or by writing to the corresponding system register.

The section **System Parameters** allows defining the displayed units and CSV delimiter. The unit system can be either SI or U.S. Depending on the chosen unit system, data point values will be presented in either SI or U.S. units.

Important! ***When changing the unit system, the device needs to be rebooted and will reset all persistent values to their default values converted to the chosen unit system.***

For generating CSV files for trend logs, alarm logs, etc., the delimiter for those CSV files can be configured. This setting can be changed between a comma ',' and a semi-colon ';'. The change takes effect immediately for all files generated by the device.

In **Remote Configuration** it can be configured, whether a replaced device shall automatically request its configuration from an LWEB-900 server. This remote configuration request is sent only, if the device does not have a data point configuration.

5.2.2 Backup and Restore

A configuration backup of the device can be downloaded via the Web interface. Press the backup link as shown in Figure 61 to start the download. The device assembles a single file including all required files. A file requestor dialog allows specifying the location where the backup file shall be stored.

To restore the device settings, simply select a previously generated backup file in the **Restore Configuration** section of the page by clicking the button next to the **Filename** field. Then press the **Restore** button.

The backed up configuration data consists of:

- Device settings (Passwords, IP settings, e-mail config, etc.),
- Data point configuration and persistent values,
- CEA-709 binding information,
- BACnet server objects and client mappings,
- L-IOB configuration and parameters,
- AST settings,
- L-WEB configuration and custom Web pages,
- IEC61131 logic program and retain variables,
- Uploaded documentation and documentation links.

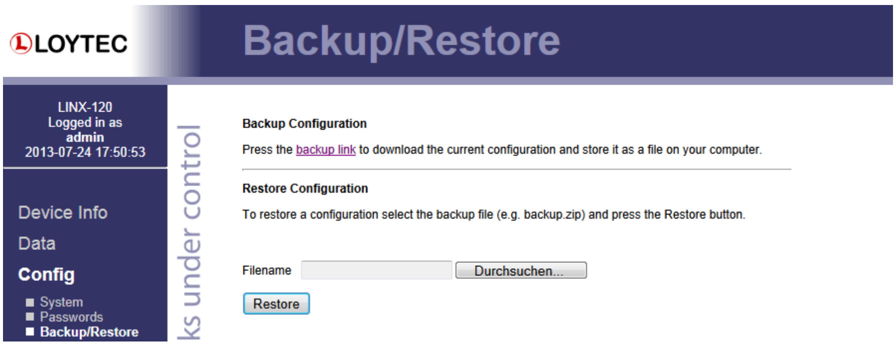


Figure 61: Backup/Restore page.

5.2.3 Port Configuration

This menu allows configuring the device's communications ports. For each communication port, which is available on the device and shown on the label (e.g., Port 1, Port 2 Ethernet), a corresponding configuration tab is provided by the Web UI. An example is shown in Figure 62. Each port tab contains a selection of available communication protocols. By selecting a checkbox or radio button the various protocols can be enabled or disabled on the communication port. Some ports allow exclusive protocol activation only, other ports (e.g., the Ethernet port) allow multiple protocols bound to that port.

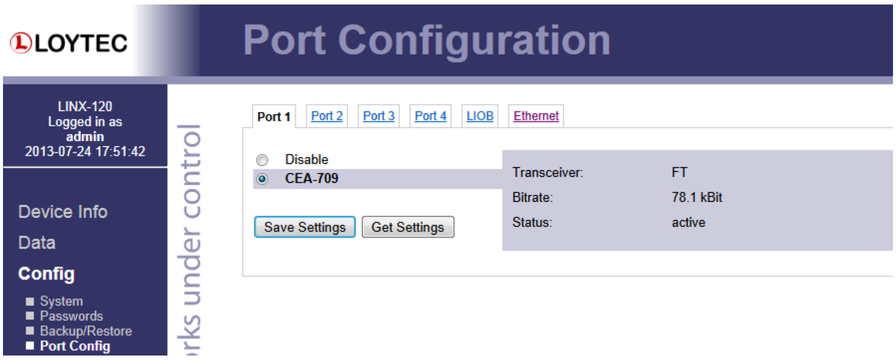


Figure 62: Port Configuration Page.

When selecting a protocol on a communication port, the protocol's communication parameters are displayed in a box on the right-hand side. To save the settings of the currently opened protocol, click the **Save Settings** button. Pressing **Get Settings** retrieves the current settings from the device.

5.2.4 IP Configuration

The TCP/IP configuration is done under the Ethernet port tab as shown in Figure 63. The mandatory IP settings, which are needed to operate an IP interface the device, are marked with a red asterisk (IP address, netmask, gateway). The **Enable DHCP** checkbox switches between manual entry of the IP address, netmask, and gateway address, and automatic configuration from a DHCP server.

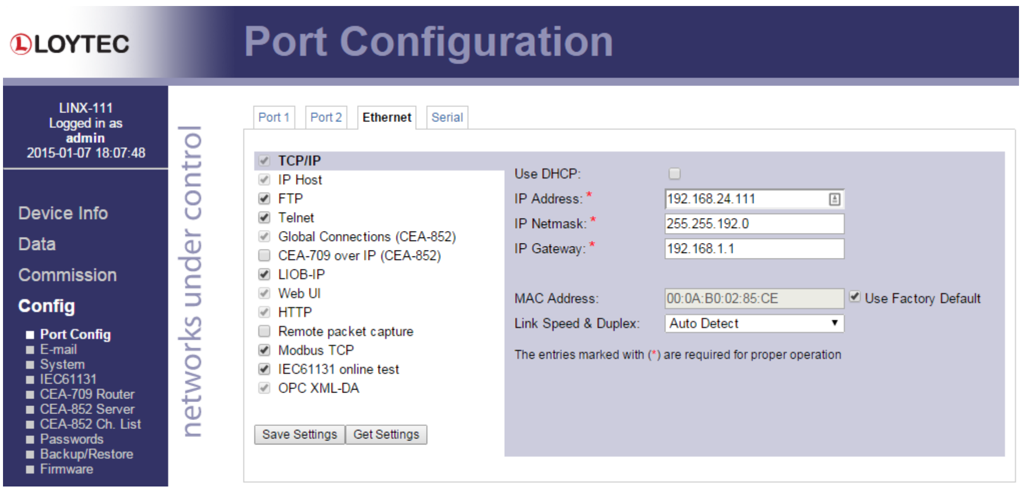


Figure 63: TCP/IP Configuration Page.

The device comes configured with a unique MAC address. This address can be changed in order to clone the MAC address of another device. Please contact your system administrator to avoid MAC address conflicts.

If the device is operated with a 10 Mbit/s-only hub, the link speed should be switched from **Auto Detect** to **10Mbps/Half-Duplex**. With modern 100/10 Mbit/s switches, this setting can be left at its default.

The settings for DNS and NTP servers should be made in the IP host settings (see Section 5.2.6). In case an IP interface runs DHCP, the DNS and NTP addresses supplied by DHCP can be seen here. Models with one Ethernet port only do not have these settings here.

Other standard protocols that are bound to an Ethernet interface are FTP, Telnet, and HTTP (Web server). By deselecting the checkbox, those protocols can be individually disabled. The standard UDP/TCP ports can be changed in the respective protocol settings. An example for the FTP server is shown for FTP in Figure 64. The FTP server is used for instance to update the firmware (see Section 20.1) or to upload a new data point configuration. Note that HTTP for the Web server can only be disabled on the console interface or by using the device configuration of the Configurator.

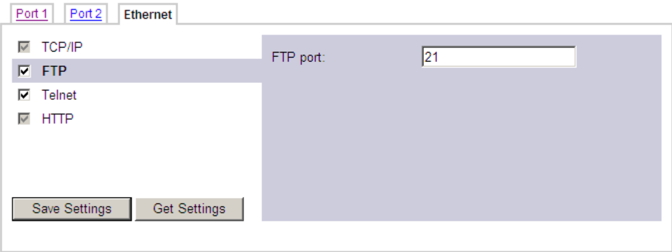


Figure 64: FTP server configuration on the Ethernet port.

5.2.5 Using Multiple IP Ports

On models with multiple IP interfaces, the port configuration provides a separate port tab for each IP port, e.g., **Ethernet 1 (LAN)** and **Ethernet 2 (WAN)**. In the port mode setting these interfaces can be enabled to operate as a separate IP network. As a default only **Ethernet 1 (LAN)** is enabled and configured to be switched with the Ethernet 2 port. To enable **Ethernet 2 (WAN)** as a separate, isolated IP network, choose **Separate network** in the port mode setting as shown in Figure 65 and save settings. A reboot is required to make this change effective.

For each IP interface configured as a separate network, the various standard protocols can be enabled separately. As a default, the secure protocols HTTPS, SSH and OPC UA are enabled on a new separate IP interface. Some protocols can be enabled on multiple interfaces at the same time, others on one interface only. If one of the latter is enabled on a new separate IP interface, a warning will be displayed, stating on which other interface the protocol will now be disabled (e.g., CEA-709 over IP, BACnet/IP, KNXnet/IP).

The separate network mode can be used, if you want to operate an isolated building network on the LAN and expose some aspects outside the building network (denoted as WAN). This configuration can also be used to operate a gateway between two otherwise entirely separate building network domains, e.g. BACnet/IP on the Ethernet 1 port and KNXnet/IP on the Ethernet 2 port. Physically, the two Ethernet ports will be plugged into different Ethernet switches.

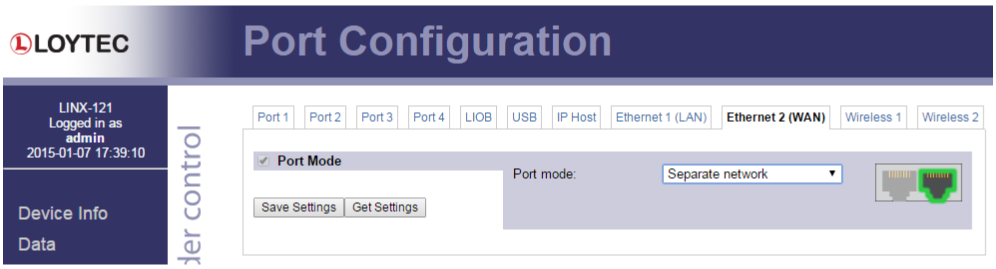


Figure 65: Enable the Ethernet 2 (WAN) interface.

To disable a separate IP interface, choose **Disable** in the port mode setting. This change is effective immediately without a reboot. To configure switch mode again, choose **Switch Ethernet 1+2** in the port mode setting.

5.2.6 IP Host Configuration

The L-INX models, which provide a built-in Ethernet switch/hub possess a separate **IP Host** tab for editing all common host settings as shown in Figure 66. These settings affect all IP interfaces on the entire device. On models with a single Ethernet port, the IP Host settings appear directly on the Ethernet tab.

Hostname and **Domainname** are optional entries and can be left empty. For some DHCP configurations it may be necessary to enter a hostname. Please contact your system administrator on how to configure DHCP to acquire an IP address.

If the device possesses more than one IP interface the **Default Gateway** setting defines the gateway of a given IP interface, which is going to route all non-local network traffic. One of the existing IP interfaces with a separate network must be selected here.

Up to three **DNS Servers** can be defined on this page. These DNS servers will be contacted by all services on any of the IP interfaces for name resolution. In case the DNS servers are supplied by DHCP running one of the IP interfaces, change the setting **Use DNS servers from** to point to that interface.

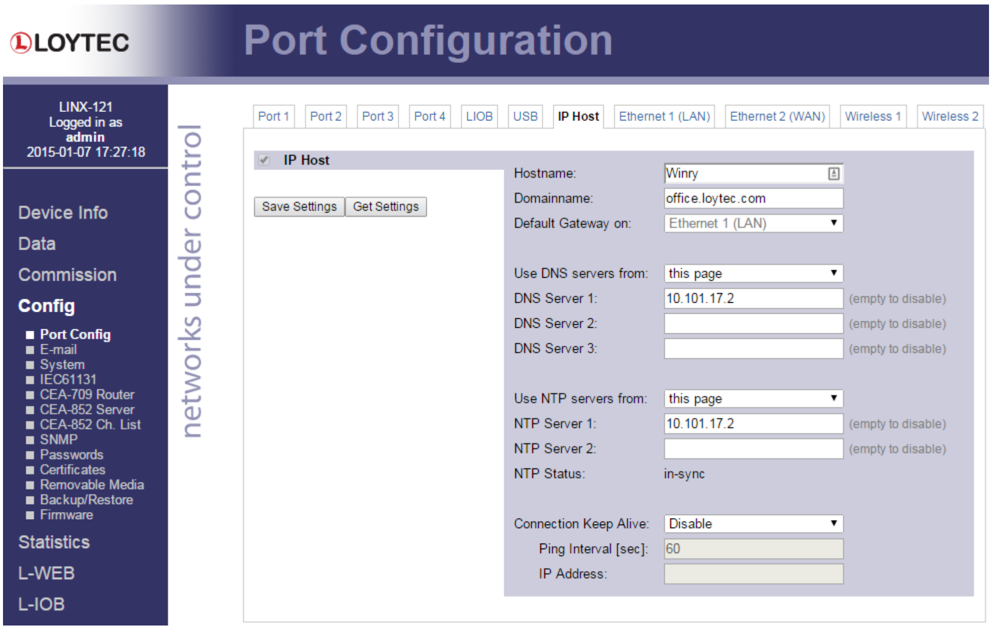


Figure 66: Setting on the IP Host tab.

The device can be configured to synchronize its clock with NTP time. Enter the IP address of a primary and, optionally, a secondary NTP server. The device will use NTP as a time source if the time sync source in the system configuration page is set to **NTP** (see Section 5.2.1). The field **NTP status** below the NTP server settings displays the current NTP synchronization status (**out-of-sync**, or **in-sync**). The settings made here apply to all IP interfaces. In case the NTP servers are supplied by DHCP running one of the IP interfaces, change the setting **Use NTP servers from** to point to that interface.

The **Connection Keep Alive** feature allows the device to automatically ping other devices on the IP network in order to maintain an IP connection that might be automatically disconnected after a specific period of time (e.g. DSL routers automatically disconnect if no activity is detected). When enabled choose one of the options Auto IP or Custom IP.

If auto IP mode is selected and the device has a CEA-852 configuration server, a ping message is sent to all CEA-852 devices in the channel list of the configuration server. If the configuration server is disabled on this device a ping message is sent to the configuration server for the IP-852 channel, if one is known. If custom IP is selected, one specific IP address can be configured as the ping destination.

5.2.7 WLAN Configuration

Devices supporting the LWLAN-800 wireless adapter can be connected to IEEE 802.11 wireless networks. The basic functions available in WLAN operation are described in Section 19.6. Depending on the required wireless modes, the first configuration step is to select the port mode on the **Wireless** tab of the port configuration, as shown in Figure 67.

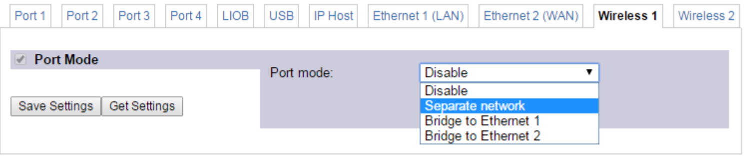


Figure 67: Wireless Port Mode

The following port modes can be selected:

- **Disable**: The wireless interface is disabled.
- **Separate network**: The wireless interface is enabled, but not bridged to any Ethernet interface. It provides its own, isolated services.
- **Bridge to Ethernet 1**: The wireless interface is enabled and bridged to the Ethernet 1 interface.
- **Bridge to Ethernet 2**: The wireless interface is enabled and bridged to the Ethernet 2 interface.

After having selected the port mode, the IP settings have to be set, if the wireless port is configured as a separate network. The wireless interfaces are configured in the same way as Ethernet interfaces described in Section 5.2.5. Depending on the wireless mode, there are some differences:

- **Access point mode (separate network)**: The IP address and netmask are used to define the network in which client get an IP address from the built-in DHCP server. DNS and NTP settings are not needed in this mode.

The wireless client settings are made in the **Wireless** protocol area. This allows setting the **WIRELESS mode** in a drop-down box. The following basic modes are available, which are described below in more detail:

- Client Mode: The WLAN client connects to an existing access point.
- Access Point Mode: The device provides a WLAN access point where a client can connect to the wireless network created by the device.
- Mesh Mode: This mode is used to create an IEEE 802.11s mesh network.

Client Mode. A wireless interface in client mode has the settings shown in Figure 68.

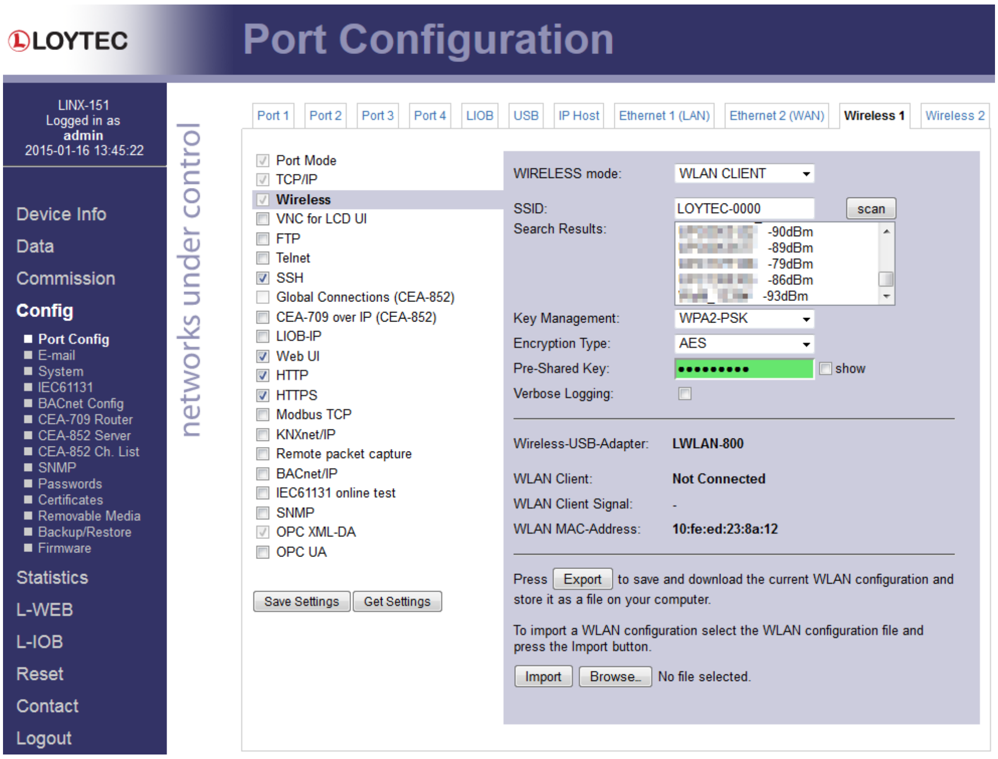


Figure 68: WLAN Client Settings

The following settings are used to configure the wireless client mode:

- **SSID**: This is the service set ID identifying the wireless network to connect to. It can be entered manually, e.g. if the network is hidden, or scanned using the **scan** button. Note that scanning interrupts an active wireless connection, so use this button only when setting up the wireless connection.

- **Search Results**: The search results list contains the discovered SSIDs and signal strenghts. Selecting one of the items copies it into the SSID field.

- **Key Management**: This list selects between NONE (no encryption), WEP, WPA and WPA2 encryption. The recommended setting is WPA2, as WPA and WEP are not considered secure anymore and are provided for backwards compatibility.

- **Pre-Shared Key**: The preshared key is the encryption key for the wireless network. The **show** checkbox shows the PSK in clear text.

- **Verbose Logging**: In case of connection problems, this checkbox can be activated to store wireless connection information in the OS log. It is not recommended to leave this option activated during normal operation.

The page displays the following information:

- **Wireless-USB-Adapter**: The type of the connected wireless adapter.

- **WLAN Client**: Displays whether the interface is connected to a wireless network.

- **WLAN Client Signal**: Displays the signal strenght.

- **WLAN MAC-Address**: Displays the MAC address of the wireless adapter

Access Point Mode. An access point has the settings shown in Figure 69.

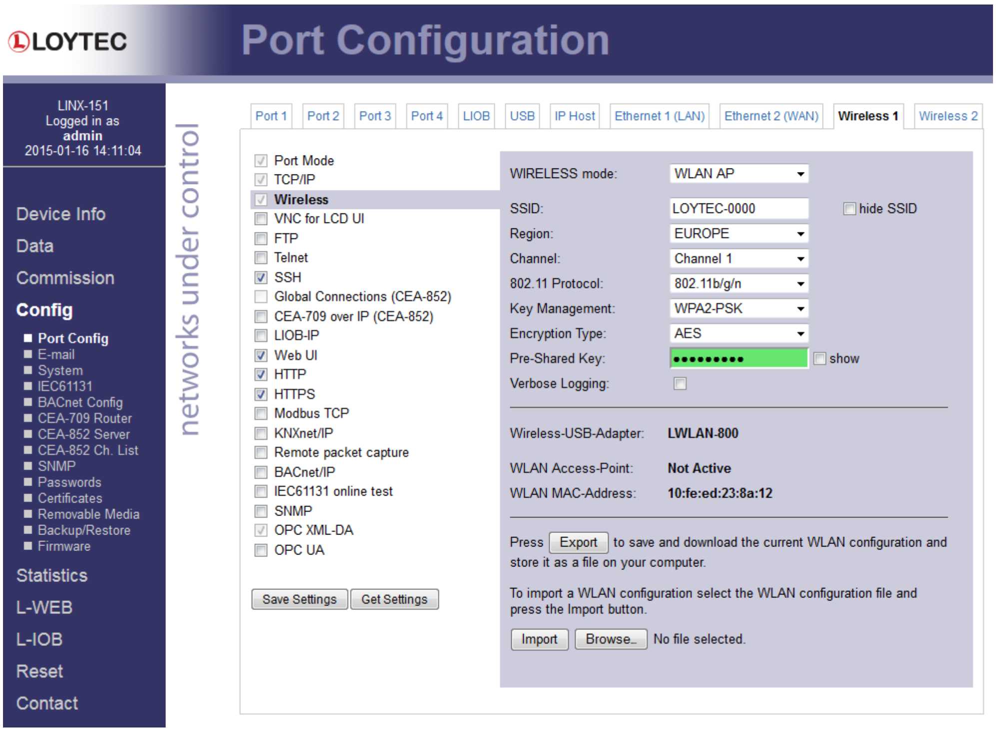


Figure 69: WLAN Access Point Settings

The following settings are used to configure the access point mode:

- **SSID**: This is the service set ID identifying the wireless network provided by this access point. The **hide SSID** checkbox hides the SSID, so that it cannot be scanned. Not that hiding an SSID has more security drawbacks than advantages, so that this setting should be left deactivated.

- **Region**: This defines the region, Europe, North America or Japan, in which this access point is deployed. Note that this settings has to be made correctly to comply with regulatory restrictions. Incorrect settings may cause interference.

- **Channel**: This field selects an available channel. The 2.4 GHz Band provides 13 channels. However these channels overlap and cannot be used without interference. When possible, use channels 1, 6 or 11 to avoid overlapping networks.

- **802.11 Protocol**: This field selects the wireless protocol to use. The default and recommended setting is 802.11b/g/n, which provides all protocols. If there are compatibility issues with some clients, the access point can be restricted to 802.11b/g or 802.11b.

- **Key Management**: This list selects between NONE (no encryption), WEP, WPA and WPA2 encryption. The recommended setting is WPA2, as WPA and WEP are not considered secure anymore and are provided for backwards compatibility.

- **Encryption Type**: This list selects between different encryption options, e.g. AES or TKIP.

- **Pre-Shared Key**: The preshared key is the encryption key for the wireless network. The **show** checkbox shows the PSK in clear text. For a secure network, please use WPA2, AES encryption and a PSK with at least 16 characters.

- **Verbose Logging**: In case of connection problems, this checkbox can be activated to store wireless connection information in the OS log. It is not recommended to leave this option activated during normal operation.

The page displays the following information:

- **Wireless-USB-Adapter**: The type of the connected wireless adapter.
- **WLAN Access-Point**: Displays status of the access point.
- **WLAN MAC-Address**: Displays the MAC address of the wireless adapter.

Mesh Mode. A mesh point or mesh portal has the settings shown in Figure 70.

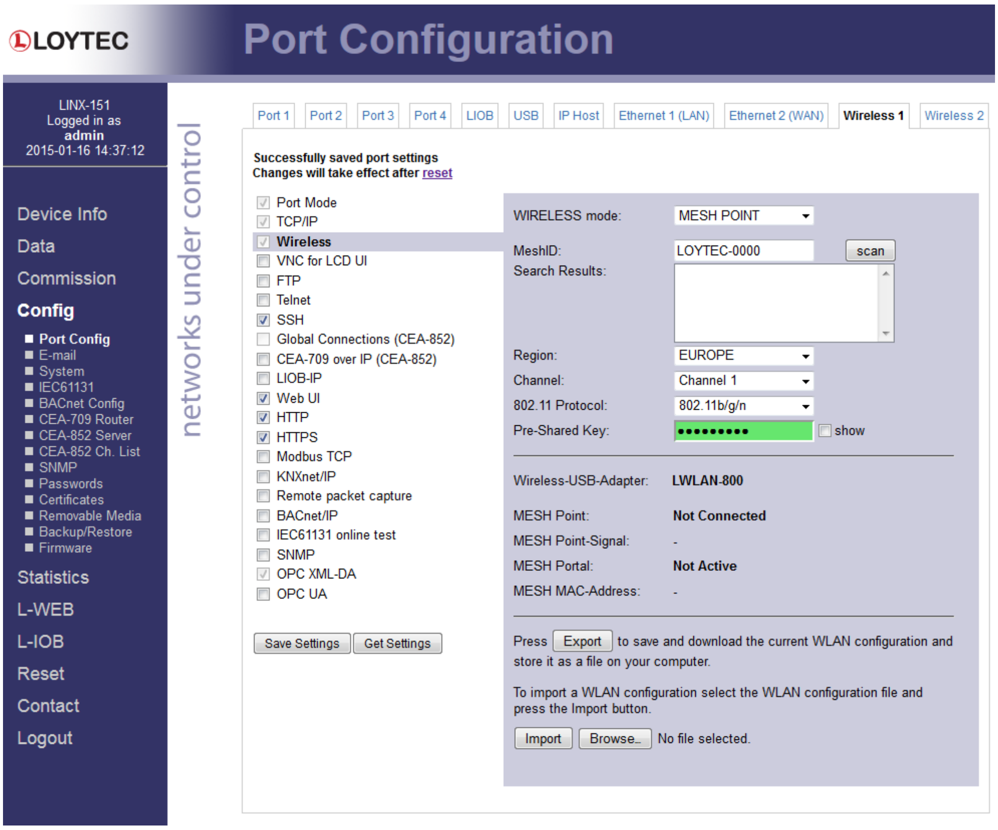


Figure 70: WLAN Mesh Network Settings

The following settings are used to configure the wireless client mode:

- **MeshID**: This is the service set ID identifying the wireless network to connect to. It can be entered manually, e.g. if the network is hidden, or scanned using the **scan** button. Note that scanning interrupts an active wireless connection, so use this button only when setting up the wireless connection.
- **Search Results**: The search results list contains the discovered SSIDs and signal strenghts. Selecting one of the items copies it into the SSID field.
- **Region**: This defines the region, Europe, North America or Japan, in which this access point is deployed. Note that this settings has to be made correctly to comply with regulatory restrictions. Incorrect settings may cause interference.

- **Channel**: This field selects an available channel. The 2.4 GHz Band provides 13 channels. However these channels overlap and cannot be used without interference. When possible, use channels 1, 6 or 11 to avoid overlapping networks. All members of a mesh network have to use the same channel.

- **802.11 Protocol**: This field selects the wireless protocol to use. The default and recommended setting is 802.11b/g/n, which provides all protocols. If there are compatibility issues with some clients, the access point can be restricted to 802.11b/g or 802.11b.

- **Pre-Shared Key**: The preshared key is the encryption key for the wireless network. The **show** checkbox shows the PSK in clear text. A mesh network should be protected by a Mesh ID of at least 16 random characters.

The page displays the following information:

- **Wireless-USB-Adapter**: The type of the connected wireless adapter.

- **MESH Point**: Displays whether the interface is connected to a mesh network..

- **MESH Point Signal**: Displays the signal strenght.

- **MESH Portal**: Indicates whether this is a mesh point or portal.

- **WLAN MAC-Address**: Displays the MAC address of the wireless adapter.

The buttons in the bottom area allow to export and import the wireless configuration. This allows to configure a device and to easily transfer the wireless settings to other devices. The **Export** button allows to save a file containing the wireless settings. The **Import** button imports a wireless configuration which has been selected by the **Browse** button. After changing the wireless settings, you need to click on **Save Settings** and reset the device for applying the settings.

5.2.8 VNC Configuration

LOYTEC devices equipped with an LCD display also provide remote access over Ethernet to the LCD display. The VNC protocol is used for this purpose and the device implements a VNC server for exposing the display. The VNC server is by default disabled on the device. On the PC a VNC client needs to be installed. Using the default settings, the VNC client connects to port 5900 of the device. The password is ‘loytec4u’.

The VNC server can be configured on the **Ethernet** tab of the port configuration. To turn on the VNC server, enable the **VNC for LCD UI** checkbox. The VNC protocol settings are displayed in the settings box on the right-hand side as shown in Figure 71. The **VNC port** and **VNC password** can be changed. As a default, only one VNC client may connect. This limit may be increased in **Max VNC clients**. In order to protect changes made on the LCD UI over VNC with a PIN code, the **Admin PIN code** can be configured. To disable PIN protection, enter ‘0000’.

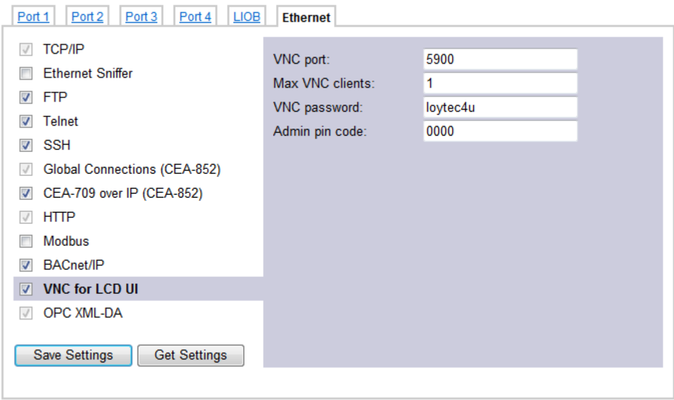


Figure 71: VNC Configuration.

5.2.9 CEA-709 Configuration

The CEA-709 protocol can be enabled on the device's ports Port1, Port2, etc. if available. To enable it, click the **CEA-709** radio button as shown in Figure 72. Note, that depending on the device model, other protocols on the same port will be disabled in this case. The protocol settings box on the right-hand side displays the current transceiver settings.

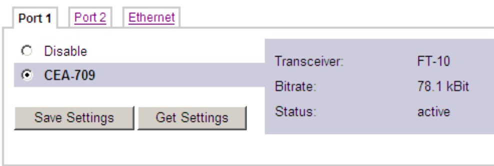


Figure 72: CEA-709 Configuration Page.

5.2.10 CEA-852 Device Configuration

The CEA-852 protocol is only available on the Ethernet port. To enable CEA-852 on the device, select the **CEA-709 over IP (CEA-852)** checkbox on the **Ethernet** tab of the port configuration page. Please note that on device models without a router or a proxy, the CEA-709 protocol on other ports will be disabled (e.g., LINX-100, L-GATE). On devices with multiple IP interfaces, the CEA-852 protocol can be activated only on one of them.

The CEA-852 protocol settings are displayed in the settings box on the right-hand side as shown in Figure 73. Typically, the device is added to an IP channel by entering the relevant information on a configuration server. The configuration server then contacts the CEA-852 device of the L-INX and sends its configuration.

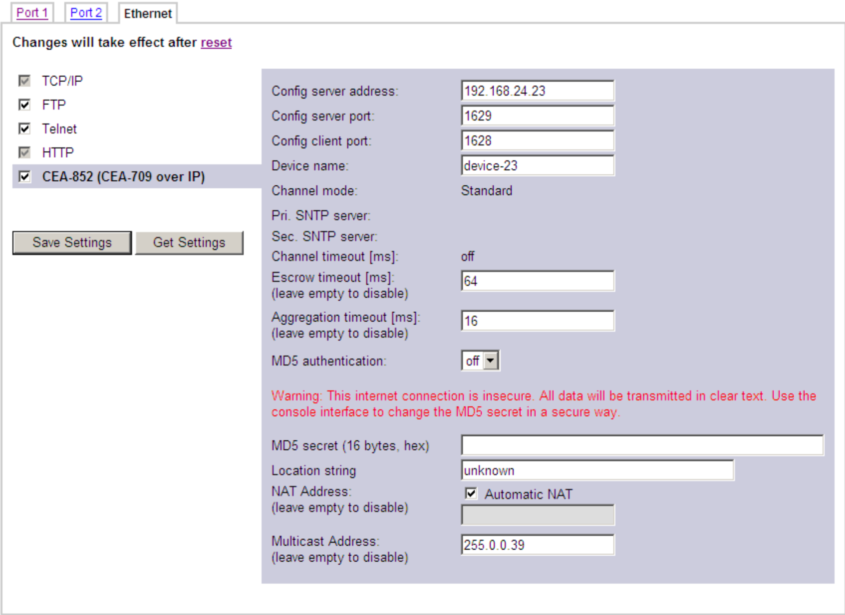


Figure 73: CEA-852 Device Configuration Page.

The field **Config server address** and **Config server port** display the IP address and port of the configuration server, which manages the L-INX and the IP channel. The field **Config client port** represents the IP port of the L-INX's CEA-852 device. This setting should be left at its default (1628) unless there are more than one CEA-852 devices operating behind a single NAT router. Please refer to the L-IP User Manual [1] to learn more about NAT configuration.

In the field **Device name** the user can enter a descriptive name for the L-INX, which will appear in the IP channel to identify this device. You can enter a device name with up to 15 characters. It is recommended to use unique device names throughout the IP channel.

The **Channel mode** field reflects the current channel mode of the CEA-852 device. It is configured by the configuration server. If there are any two devices in the channel which use the same IP address but different ports (e.g., multiple devices behind one NAT router) the channel switches to **Extended NAT mode**. Please refer to the L-IP User Manual [1] to learn more about configuring the Extended NAT mode in the configuration server.

The configuration server sets the **SNTP server** addresses and the **Channel timeout**.

The filed **Escrow timeout** defines how long the CEA-852 device on the L-INX waits for out-of-sequence CEA-852 data packets before they are discarded. Please enter the time in ms or '0' to disable escrowing. The maximum time is 255 ms.

The field **Aggregation timeout** defines the time interval in which multiple CEA-709 packets are combined into a single CEA-852 data packet. Please enter the time in ms or '0' to disable aggregation. The maximum time is 255 ms. Note that disabling aggregation will negatively affect the performance of the CEA-852 device of the LINX.

The field **MD5 authentication** enables or disables MD5 authentication. Note that MD5 authentication cannot be used together with the Echelon's *i*.LON 1000 since the *i*.LON 1000 is not fully compliant with the CEA-852 authentication method. MD5 can be used with the *i*.LON 600. In the following field **MD5 secret** enter the 16-byte MD5 secret. Note that for security purposes the active MD5 secret is not displayed. You may enter the 16 bytes as one string or with spaces between each byte, e.g., 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF.

Also note that entering the MD5 secret on the Web interface may pose a security risk. Since the information is transmitted over the network it can be subject for eavesdroppers on the line. It is recommended to use a cross-over cable.

In the field **Location string** the user can enter a descriptive test which identifies the physical location of the device. A location string can have a maximum length of 255 characters. This is optional and for informational purposes only.

If the CEA-852 device on the L-INX is used behind a NAT router, the public IP address of the NAT router or firewall must be known. To automatically detect the NAT address leave the **Auto-NAT** checkmark enabled.

The **Multicast Address** field allows the user to add the CEA-852 device of the L-INX into a multi-cast group for the CEA-852 IP channel. Enter the channel's IP multi-cast address here. Please contact your system administrator on how to obtain a valid multi-cast address. To learn when it is beneficial to use multi-cast addresses in your channel please refer to the L-IP User Manual [1].

5.2.11 Global Connections Configuration

The CEA-852 device used for global connections (see Section 6.4.4) can be configured on the Ethernet port. The global connections function is always enabled on the CEA-852 device. This is indicated by the checkbox **Global Connections (CEA-852)** on the **Ethernet** tab of the port configuration page as shown in Figure 74. The settings are shared with the **CEA-709 over IP** settings, if that protocol is enabled. Otherwise, the CEA-852 device is configured on this tab as described in Section 5.2.10.

If the user does not want to share the CEA-709 over IP channel with his global connections, the checkbox **Use separate IP channel for global connections** can be activated. In this case, a separate CEA-852 device is configured on this tab as described in Section 5.2.10. Note, that this CEA-852 device will need a different port number, e.g. 1630. The separate CEA-852 device for global connections cannot be added to the local configuration server. In this case, also a separate configuration server (e.g. a LOYTEC L-IP) must be used.

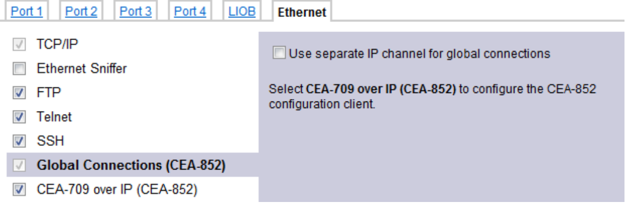


Figure 74: Global Connections Configuration Page.

5.2.12 CEA-709 Router Configuration

This page is only available on CEA-709 L-INX models with the router option (101, 111, 121, 151). The CEA-709 router configuration page allows configuring the built-in router mode. Available modes are **Configured Router** and **Smart Switch**. The L-INX must be rebooted to let the changes on this page take effect.

The configured router mode is the default setting. Choose this setting if you want to use the L-INX as a standard configured CEA-709 router that can be configured in a network management tool such as NL-200 or LonMaker.

The Smart Switch mode lets the device act as a self-learning router like the L-Switch. In this configuration the LINX's router doesn't need to be configured with a network management tool but is completely transparent in the network. Use this operating mode in a plug&play networking environment. The switch mode should only be used in LAN networks. In Smart

Switch mode, this page has two more configuration fields: **Subnet/node learning** and **Group learning**.

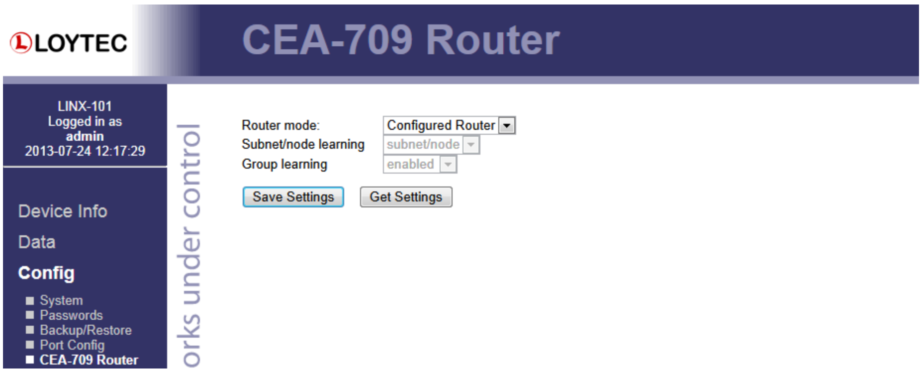


Figure 75: CEA-709 Router Configuration Page.

5.2.13 CEA-852 Server Configuration

This page is available on all L-INX and L-GATE models. On this configuration page the configuration server on the device can be enabled or disabled. In the drop-down box **Config server status** select **enabled** and click on **Save Settings** to activate the configuration server. On devices with multiple IP interfaces choose an IP interface on which the configuration server shall be running. Then the configuration server settings page appears as shown in Figure 76. If the configuration server is enabled the green configuration server LED labeled **CS** will be on, otherwise it will be off.

The configuration server port can be changed in the **Config server port** field. It is recommended to keep the default port setting of 1629. The field **Channel name** is informational only and can consist of up to 15 characters.

The field **Channel members** displays the current number of members on the IP-852 channel. The field **Channel mode** reflects the current channel mode. The L-INX configuration server automatically determines this mode. Depending on if there are any two devices in the channel which use the same IP address but different ports (e.g., multiple CEA-852 devices behind one NAT router). If all IP addresses are unique, the mode is **Standard**, if some are not unique the mode is **Extended NAT mode**. Please refer to Section 8.4.2 to learn more about the implications of this mode.

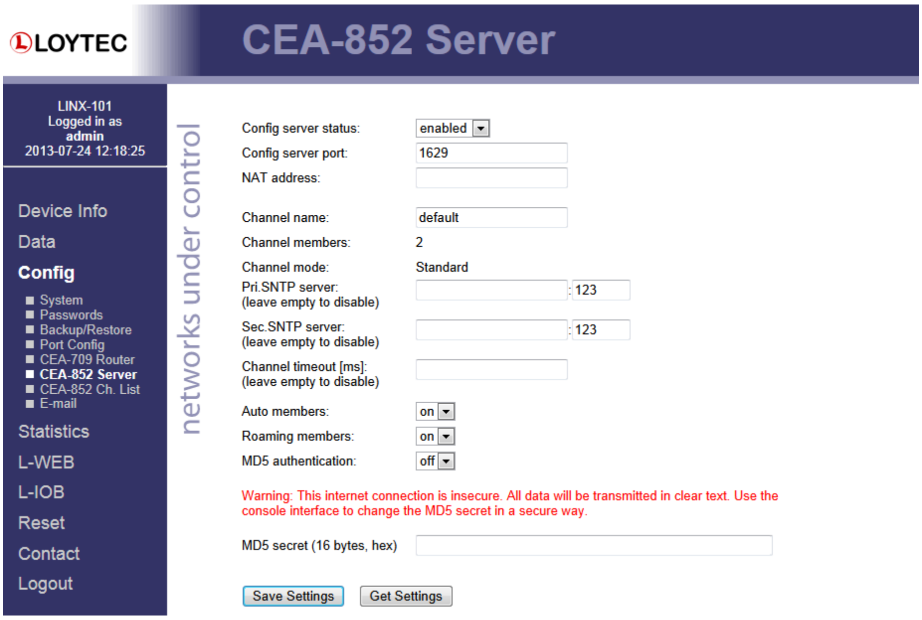


Figure 76: Configuration server settings.

Enter NTP timer server address and ports in the fields **Primary SNTP** and **Secondary SNTP**. The L-INX will synchronize to NTP time if primary or primary and secondary NTP servers are specified. A list of available timeservers can be found at www.ntp.org.

The **Channel timeout** is an IP-852 channel property and indicates how old a packet can be before it is discarded. The channel timeout is set in ms. To disable the channel timeout enter a value of 0. To select the proper value please consult Section 8.7.1. Setting a channel timeout other than 0 requires a valid SNTP server entry on the configuration server.

The **Auto members** option allows members to be automatically added to the channel. If turned on, CEA-852 devices can register on the IP-852 channel without the device being explicitly added on the configuration server. This special feature is useful in combination with the LPA-IP since it can add itself to the configuration server during the debug session. Non-responding auto members are automatically removed from the channel. This feature is turned off by default and must be explicitly turned on. Use this option with care because new CEA-852 devices can add themselves to the channel without knowledge of the system operator. This could cause a potential security hole.

The **Roaming members** option allows tracking CEA-852 devices when their IP address changes. This feature must be turned on if DHCP is used and the DHCP server can assign different IP addresses to the same device (same Neuron-ID). In combination with Auto-NAT the L-INX's router can also be operated behind NAT routers, which change their IP address between connection setups. For more information on this topic refer to Section 8.4.1. The roaming member feature is turned on by default. It is recommended to turn off this feature if DHCP is not used or if the DHCP server always assigns the same IP address to a given MAC address.

Use the drop-down box **MD5 authentication** to enable and disable MD5 authentication. If MD5 authentication is enabled, all devices on the IP-852 channel must have MD5 enabled and must use the same MD5 secret. Note that MD5 authentication cannot be used together with the Echelon's *i*.LON 1000 since the *i*.LON 1000 is not fully compliant with the CEA-852 authentication method. MD5 can be used with the *i*.LON 600. The MD5 secret can be entered over the Web interface. You may enter the 16 bytes as one string or with spaces between each byte, e.g., 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF.

It is recommended, however, to enter the secret locally and not over an Internet connection. It is best to use a cross-over Ethernet cable connected to the PC.

5.2.14 CEA-852 Channel List

This page is available on all L-INX and L-GATE models. If the configuration server is enabled on the L-INX, the CEA-852 device list can be seen in the CEA-852 channel list menu. An example is given in Figure 72.

The **Add Device** button is used to add another CEA-852 device to the IP-852 channel. The **Reload** button updates the Web page and the **Recontact** button contacts all devices to update their status. The **Execute** button executes the option selected in the adjacent drop-down box on the checked members. Each member can be selected for that action in an individual check-box in the **Sel** column. Actions available are: **disable**, **enable**, **delete**, **assign to NAT**, and **remove from NAT**. For more information on the actions on NAT routers refer to Section 8.4.2.

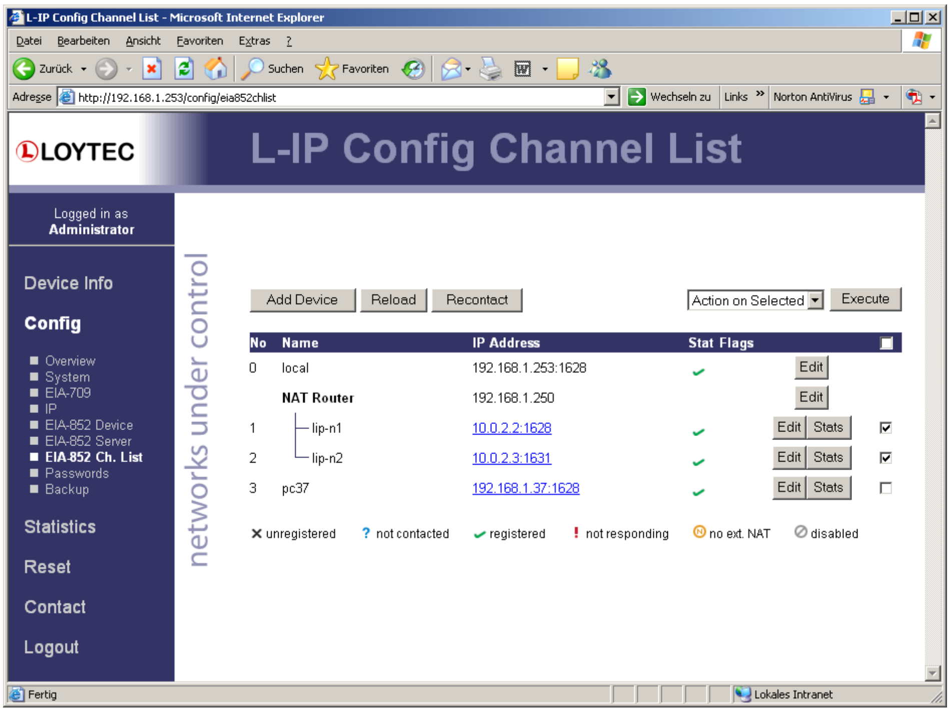


Figure 77: CEA-852 channel membership list.

The device status information is indicated with descriptive icons of different colors. The description for the different status indicators is shown in Table 12. The **Flags** column indicates with an **A** that the device is an auto member.

Click on the **Edit** button to change the device name, IP address, and port number for this device. Click **Edit** on a NAT router to change the NAT router address. The **Stats** button retrieves the statistics summary page from the client device.

Icon	Status	Description
	registered	The CEA-852 device has been successfully registered with the IP-852 channel and is fully functional.
	unregistered	The CEA-852 device has never been registered with the IP-852 channel.
	not contacted	The CEA-852 device has not been contacted since the configuration server has started.
	not responding	The CEA-852 device has been registered but is not responding at the moment.
	disabled	The CEA-852 device has been disabled on the channel (or rejected).
	No extended NAT	The CEA-852 device does not support the extended NAT mode. This device is disabled.

Table 12: Possible Communication Problems in the Configuration Server.

5.2.15 BACnet Configuration

Figure 78 shows the BACnet device configuration page. This configuration page allows setting the **Device ID**, which is the instance part of the Object_Identifier property of the BACnet Device object. The field **Device name** holds the name of the BACnet device object (property Object_Name).

Important! ***The device ID and device name must be unique within the BACnet internetwork.***

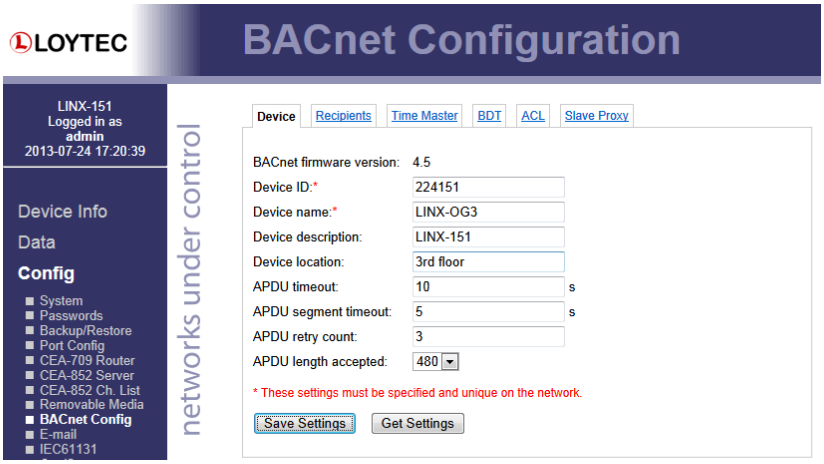


Figure 78: BACnet Device Configuration.

Further, the description and location can be configured. These configuration items correspond to the properties Description, and Location respectively of the BACnet Device object. For tuning BACnet application timing parameters, set **APDU timeout**, **APDU segment timeout**, and **APDU retry count**. The timeout values are entered in seconds allowing decimal notation, e.g. "7.5".

On the settings for BACnet/IP refer to Section 5.2.16. For configuring the MS/TP data link refer to Section 5.2.17.

Note: *If this page displays the message "Device communication is disabled via BACnet network!" the device has been externally disabled. Reboot the device to activate communication again.*

5.2.16 BACnet/IP Configuration

The BACnet/IP protocol is available on the Ethernet port. To enable BACnet/IP on the device, select the BACnet/IP checkbox on the Ethernet tab of the port configuration page. Please note that on device models without a router, the BACnet MS/TP protocol on other ports will be disabled (e.g., LINX-200, L-GATE). On devices with multiple IP interfaces, the BACnet/IP protocol can be activated only on one of them.

The BACnet/IP protocol settings are displayed in the settings box on the right-hand side as shown in Figure 79. On devices with a router (e.g., LINX-201) the **Network Number** of the BACnet/IP port must be configured to operate the built-in router. If the BACnet/IP network uses a non-default UDP port number other than 47808/0xBAC0, enter this port in the **BACnet/IP port** field. Enter '0' in this field for switching back to the default setting.

Important! ***For operating the LINX-151,201,211,221 as a BACnet router between BACnet/IP and MS/TP, the BACnet network numbers for the BACnet/IP and MS/TP ports must be set.***

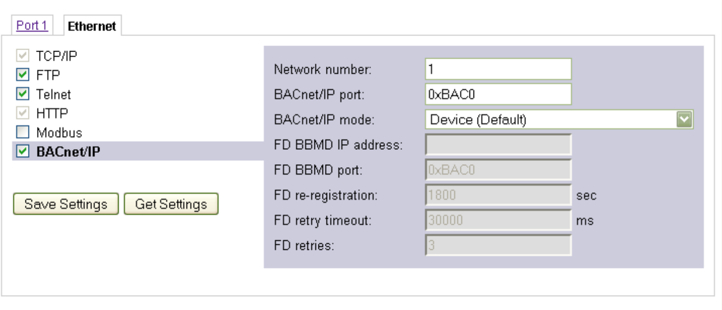


Figure 79: BACnet/IP Configuration.

In the field **BACnet/IP mode** the operation mode of the device is selected:

- **Device** (Default): In this mode the device operates as a regular BACnet/IP device on the local network without other advanced features.
- **Foreign Device** (FD): In this mode, the device registers at an existing BBMD in the BACnet/IP network as a foreign device. It is used, if the device is located as a single BACnet/IP device on a remote IP subnet or behind a NAT router. If operated as a foreign device behind a NAT router, port forwarding to the BACnet/IP port (UDP, default port 0xBAC0) and optionally to the Web server and FTP server port (TCP, default port 80 and 21) must be setup in the NAT router. If foreign device is selected, the following, additional settings must be made:
    - **FD BBMD IP address** and **FD BBMD port**: IP address and port of the remote BBMD the device registers at as a foreign device.
    - **FD re-registration**: A foreign device must periodically re-register at a BBMD. Here you can setup the corresponding interval. The default is 1800 seconds.
    - **FD retry timeout** and **FD retries**: Here you can specify the behavior, if registration does not work instantly. These values should be left at default: 30000ms / 3 retries.
- **Broadcast Management Device** (BBMD): This option is available on the L-INX models with the router (151, 201, 211, 221) and on all L-GATE models. Same as 'Device' but the BBMD function is enabled (see Section 5.2.21). For BBMD-only function, MS/TP can also be disabled (see Section 5.2.17).

On L-INX models with a BACnet router, the BACnet/IP port can be disabled while having MS/TP still enabled. This effectively disables the router, which can be useful for debugging purposes.

5.2.17 MS/TP Configuration

The BACnet MS/TP protocol can be enabled on one of the device's **Port** tabs, if MS/TP is available on the device model. To enable it, click the **BACnet MS/TP** radio button as shown in Figure 80. Note, that depending on the device model, other protocols on the same port will be disabled in this case. On a BACnet L-INX without a BACnet router and on the L-GATE BACnet MS/TP port is not enabled by default. On a BACnet L-INX with a BACnet router the MS/TP port is enabled by default.

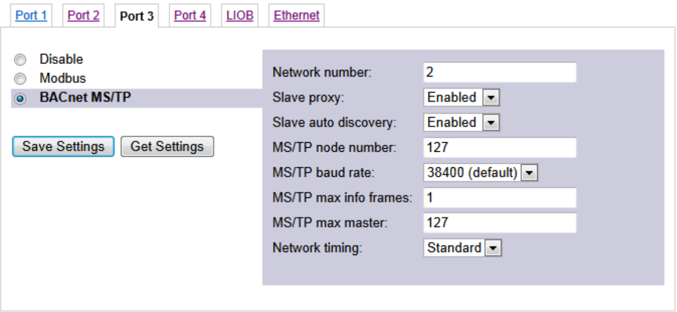


Figure 80: MS/TP Configuration.

The MS/TP protocol settings are displayed in the settings box on the right-hand side as shown in Figure 80. Mandatory settings are the **MS/TP node number** and the **MS/TP baud rate.** The MS/TP node number determines the physical address of the device on the MS/TP channel and must be in the range from '0' to the number configured with the **MS/TP max master** configuration option. It must be unique within the MS/TP channel. The Baud rate on the MS/TP channel can be set to 9600, 19200, 38400, and 76800 Baud.

Important: ***All masters on the MS/TP channel must have the same setting for MS/TP max master. Decreasing the default value 127 of MS/TP max master may reduce latency on the MS/TP bus.***

It is strongly recommended to leave the **MS/TP max info frames** and the **MS/TP max master** configuration options at their default settings. In any case the **MS/TP max master** number must be high enough to include the highest MS/TP node number of all masters on the channel. Slave devices may have a higher MS/TP node number than **MS/TP max master**.

To operate with slow devices on the MS/TP network set the **Network Timing** option to slow. This increases a number of timeouts, which is needed by some devices, but slows down network communication. If communication problems occur in standard mode, try setting the slow mode. For fine-tuning other parameters please refer to Section 19.4.

On L-INX models with a BACnet router (e.g., LINX-201), the **Network Number** of the MS/TP port must be configured to operate the build-in router. On those L-INX models the MS/TP port can also be disabled independently of the BACnet/IP port. On those models the MS/TP slave proxy function can be enabled and disabled. If the slave proxy function is enabled, auto-discovery of MS/TP slaves can be configured. The default setting on a L-INX is slave proxy enabled with auto-discovery on.

5.2.18 BACnet Recipients

BACnet notification class (NC) objects have a recipient list. Other BACnet devices, that shall act as alarm recipients and receive alarm notifications need to be added to the recipient list of the respective notification class. The **Recipients** tab of the **BACnet Config** menu can be used to view currently subscribed recipients as shown in Figure 81. Recipient entries can be modified and deleted from the list. It is also possible to add new recipients to the list with the **Add Recipient** button. This way it is possible to integrate third-party devices as alarm recipients without an OWS.

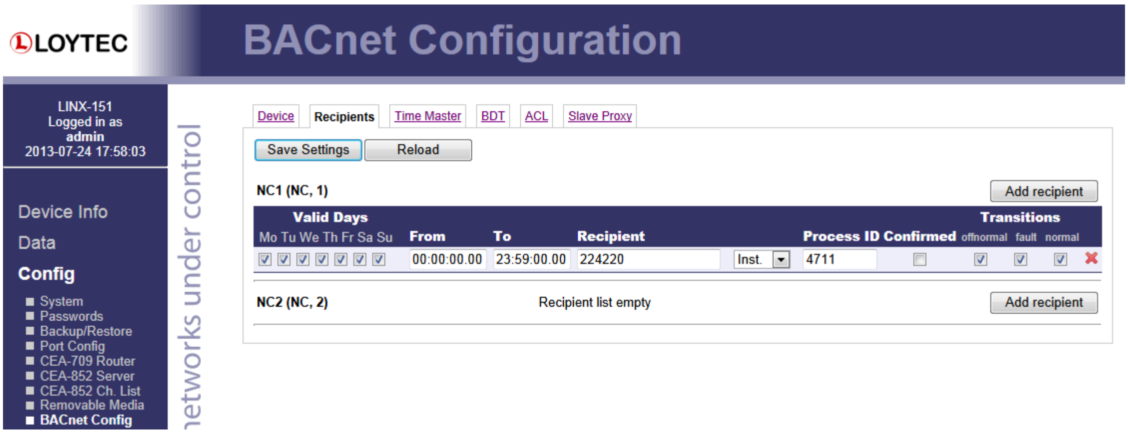


Figure 81: BACnet Recipients Configuration.

5.2.19 BACnet Time Master

The BACnet time master function relies on a list of time recipients. The **Time Master** tab of the **BACnet Config** Web page (see Figure 82) allows adding and removing time recipients of two classes: UTC time sync recipients, and time sync recipients (receiving local time). The time sync interval can also be configured on this tab. See Section 18.3.8 for more information on the settings for time sync interval, interval offset and align intervals.

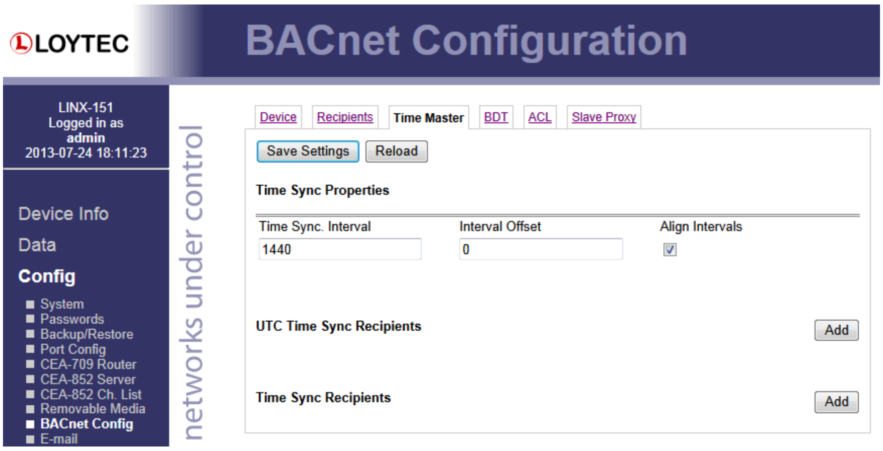


Figure 82: BACnet Time Master Configuration.

5.2.20 BACnet Restart Notifications

The device can be configured to send out a BACnet restart notification every time the device is starting. The list of recipients for this notification can be configured on the **BACnet Restart Notifications** tab. Click the **Add** button for adding a new line to the list. Then choose the recipient type from the drop-down box; it can be etiher a device instance number or a BACnet address. For broadcasting the restart notification, choose **Addr** and type in an asterisk '*' for a global broadcast or prefix it with a destination subnet, e.g. '12:*' as shown in Figure 83. Then click **Save Settings** to store the new recipient.

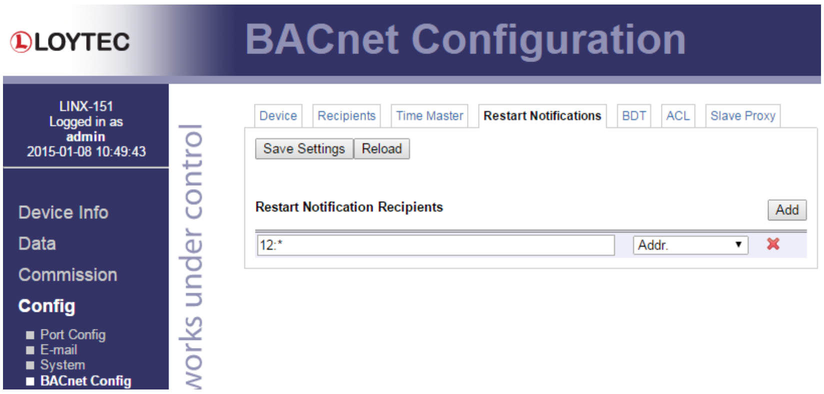


Figure 83: Broadcast BACnet restart notifications to a subnet.

5.2.21 BACnet BDT (Broadcast Distribution Table)

The BBMD function is only available on L-INX models with the BACnet router option (151, 201, 211, 221) and on all L-GATE models. The BBMD function is needed when a BACnet/IP network spans over several IP subnets separated by IP routers. If the device is configured as a BBMD, i.e. the BACnet/IP mode is set to **Broadcast Management Device,** see Section 5.2.15, the BDT (Broadcast Distribution Table) specifies all other BBMDs of the BACnet/IP network. The BDT is shown in Figure 84.

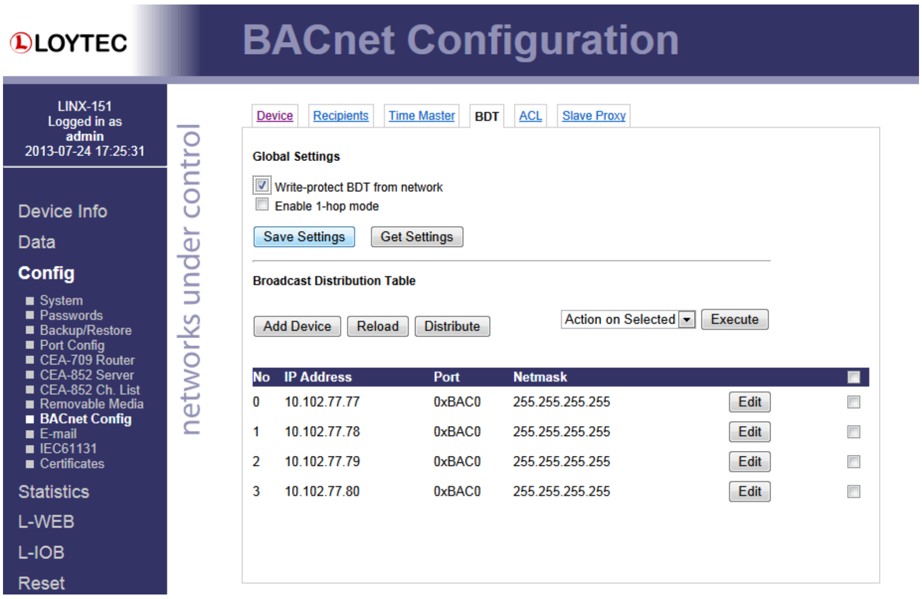


Figure 84: BACnet Broadcast Distribution Table.

By clicking **Add Device** new BBMDs (IP address and port) can be added. With **Action on Selected** and selecting existing entries, certain BBMDs can be deleted again from the table. It is not necessary to reboot the device when changing the table. However, you may want to click **Distribute** in order to propagate the table to all BBMDs in the list.

Note:	*The recommended maximum are 100 BBMD entries in the BDT.*

In the **Global Settings** section of this configuration page the behavior of the BDT can be modified:

- **Write-protect BDT from network**: If this option is enabled, the BBMD will reject any Write-BDT requests from the BACnet network. This option may be useful to protect your BDT tables from malicious access from the network.

- **Enable 1-hop mode**: Normally, the BBMD forwards broadcasts to the designated IP addresses of other BBMDs. This mode is called 2-hop mode. If the IP infrastructure allows sending directed broadcasts to other subnets, the BBMD can be switched to 1-hop mode. In this case, the subnet masks of the destination networks must be configured in the BDT entries.

5.2.22 BACnet ACL (Access Control List)

The device provides a feature in BACnet/IP to filter packets from certain sources on the BACnet/IP network. This feature is based on an access control list (ACL). An example of the ACL configuration is shown in Figure 85.

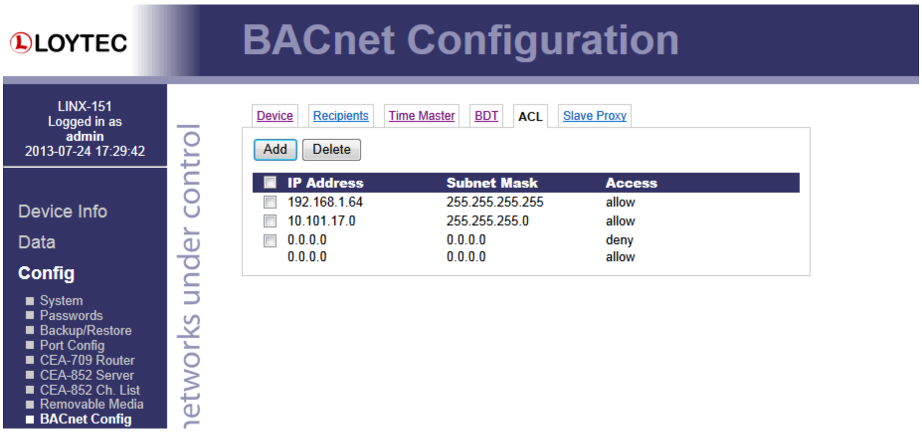


Figure 85: BACnet Access Control List (ACL).

The user can add and delete entries to the ACL. Each entry contains a source specification, which consists of an IP address and an IP mask, and an action (allow or deny). For specifying single hosts use the IP address and the mask '255.255.255.255'. For an address range specify an appropriate mask. For example use '10.101.17.0' and the mask '255.255.255.0' to specify all hosts with IP addresses '10.101.17.xxx'. To specify all IP addresses use '0.0.0.0' and the mask '0.0.0.0'.

The ACL is evaluated from specific host entries down to wider ranges. When adding new entries the ACL is automatically sorted, having the most precise definition at the top and the most general one at the bottom. The default behavior is to allow packets from all IP addresses. This is also the default entry in the ACL.

The example shown in Figure 85 specifies the following behavior for BACnet/IP:

1. Allow packets from the device 192.168.1.64

2. Otherwise allow packets from devices in the network 10.101.17.xxx

3. Otherwise deny packets from all (other) IP addresses. Note, that a "deny" overrules an "allow".

5.2.23 BACnet Slave Proxy

On L-INX models with a BACnet router (e.g., LINX-201), the MS/TP slave proxy function is available. It can be enabled in the MS/TP port configuration settings (see Section 5.2.17). In auto-discovery mode the slave proxy permanently scans the MS/TP channel and automatically detects MS/TP slave devices. On the **Slave Proxy** tab of the **BACnet Config** page the **Slave Address Bindings** list shows all detected slave devices and displays their device instance number and BACnet address (DNET:MAC address) information as shown in Figure 86.

It is also possible to manually add slave address bindings in case MS/TP devices are not detected automatically. For doing so click the **Add** button and enter the device instance number and BACnet address. If not known, leave the DNET part empty and press *Enter*. After adding all manual entries select the **Update DNET** check box and click on **Save Settings**. This updates the current MS/TP DNET number for the manual slave address bindings.

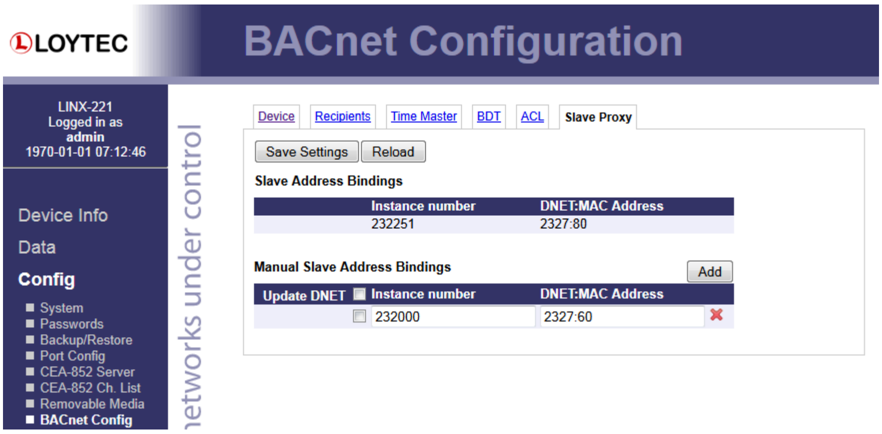


Figure 86: BACnet MS/TP slave proxy address bindings.

5.2.24 E-mail Configuration

The Web interface provides the e-mail configuration page to set up an e-mail account, which is used to send e-mails. The content and time when e-mails are sent is configured through the Configurator software (see Section 7.11). The e-mail configuration page is shown in Figure 87.

In the field for the outgoing e-mail server, enter the SMTP server of your Internet provider. Typically, the SMTP server port can be left at 25. In the field **Source E-mail Address**, enter the e-mail address of the device's e-mail account. In the field **Source E-mail Sender Name** enter a name that the e-mail will display as the source name. Note, that only ASCII characters are allowed in the name. If replies shall be sent to another e-mail address, specify this in the **Reply E-mail Address**.

If the provider's SMTP server requires authentication, enter the required user name and password. Note, that username/password is supported as well as SSL/TLS authentication (e.g., for using Hotmail, gmail, or Yahoo!). For older versions of secure connection check the SMTPS check box.

To verify the e-mail configuration, reboot the device to let the changes take effect and return to the e-mail configuration page. Then press one of the **Send Test E-mail** buttons. Note, that a DNS server must be configured in the IP settings (see Section 5.2.4) to resolve the e-mail server host name. The Web UI displays a warning message at the top of the page, if the DNS configuration is missing.

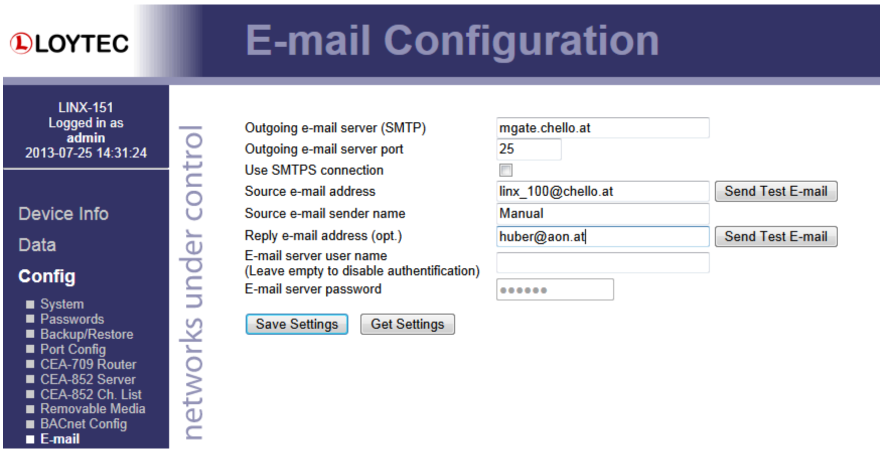


Figure 87: E-mail Configuration Page.

5.2.25 IEC61131 Configuration

On the IEC61131 configuration page the user can download an IEC61131 program as well as controlling the behavior of the I/O driver, see Figure 88. After downloading the IEC61131 program the PLC kernel is restarted with the new program. After it has been restarted check the PLC LED for a successfully started program.

Important! ***Downloading a new IEC61131 program may result in the need for a new data point configuration. Hence, take care about the requirements of the downloaded program and in case of a different set of IEC61131 variables use the L-INX Configurator to adapt the data point configuration. The I/O check feature disables the I/O driver if missing data points are detected.***

The data exchange of the IEC61131 program can be disabled by disabling the I/O driver. As a result the IEC61131 program is not able to receive or send update from/to the appropriate data points. Hence, for debugging purpose, the data point values can be manipulated regardless of the data set from the IEC61131 program. The I/O check feature disables the I/O driver after a reboot, if any variables with missing data points are detected. Deselect the I/O check to disable this feature. Variables, which could not be loaded are listed at the bottom of the page. The information about the running program is also displayed. The IEC61131 program can be removed from the device by clicking the button **Remove**. The device needs to reboot after this action.

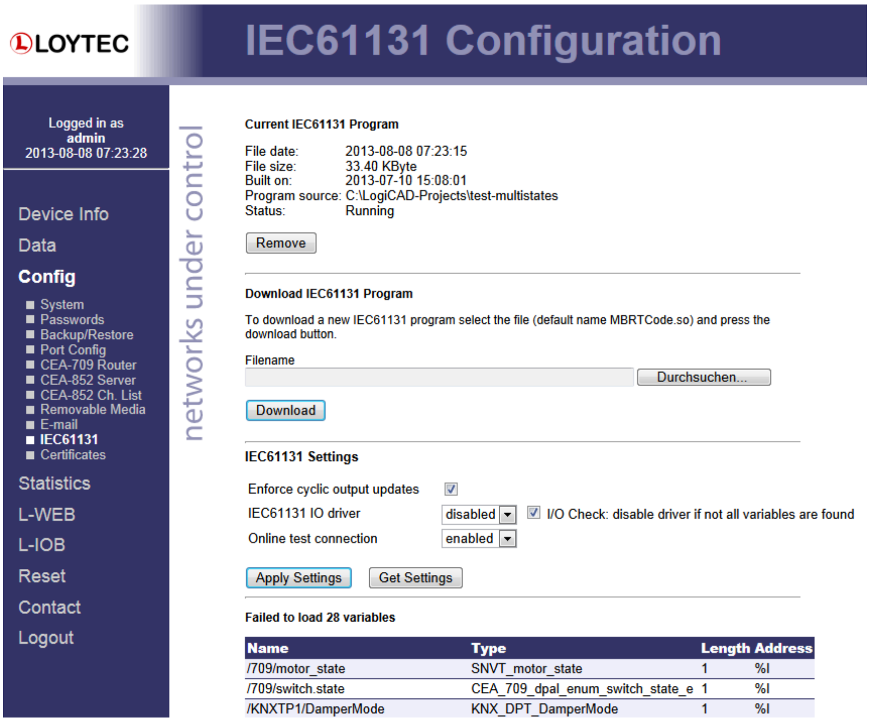


Figure 88: IEC61131 Configuration Page.

The option **Enforce cyclic output updates** makes sure that the output data points will contain the calculated logic value after each cycle. For more information refer to Section 17.4.6.

5.2.26 SSH Server Configuration

Some L-INX and L-GATE models provide an SSH server. SSH allows encryption and authentication. The SSH server settings can be configured in the Ethernet port configuration page.

It is possible to enable or disable the SSH server and to change the TCP port of the SSH server. The default SSH server port is 22. These settings will be active after rebooting.

The SSH configuration page displays the fingerprints of the RSA and DSS host keys. A random DSS key (fixed to 1024 bits) and RSA key (1024 bits) is generated per default. New keys can be created by selecting the required **RSA key size** (1024 or 2048 bits), selecting the **Generate RSA key** and/or **Generate DSS key** check boxes and selecting the **Generate** button. The SSH server will load the new keys after rebooting.

Note that recreating the SSH host keys can take up to a minute to complete. SSH clients which have already accepted the previous host key will refuse to connect to the SSH server until the host key change is confirmed in the client.

5.2.27 Certificate Management

Some L-INX and L-GATE models provide the secure HTTPS and OPC UA in addition to HTTP and OPC XML-DA. It allows for encrypted and authenticated communication.

The HTTPS server settings can be configured in the Ethernet Port Configuration page. It is possible to enable or disable the HTTPS server and to change the TCP port of the HTTPS server. The default HTTPS server port is 443. These settings will be active after rebooting.

When connecting with a web browser to the L-INX you will be warned that the server uses a self-signed certificate. You need to accept the certificate in order to continue. In some browsers this is also called “adding an exception”.

Note that in default configuration, communication is encrypted, but not authenticated, as the default certificate is self-signed. If you operate in a safe environment, no further action has to be taken.

In a hostile environment (e.g. when using over the internet), consider installing a server certificate signed by a certification authority to prevent man-in-the-middle attacks. HTTPS and OPC UA servers use X.509 certificates to authenticate themselves to clients. In order to establish communication, the client has to trust the server certificate. There are two options to accept a server certificate:

- The user manually accepts the certificate.
- The server certificate is provided by a public certification authority (CA).

The L-INX devices are configured with a self-signed certificate, but custom server certificates can be imported in the configuration page. Please follow these steps to install a custom SSL certificate.

1. Go to the **Certificates** configuration page and select the **Create Certificate** tab. Choose the radio button **CA Request** as shown in Figure 89. In **Common Name** provide a valid DNS host name (e.g., linx-g01.acme.com) or the IP address for the device. SSL certificates use host names. Enter organization name, organization unit, city, and state. Then choose the country and click **Create Certificate Request**.

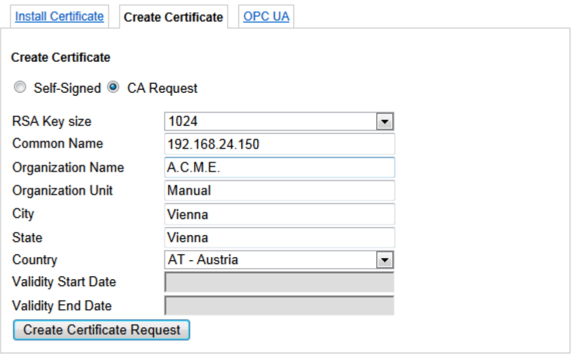


Figure 89: Create a CA certificate request.

2. Copy the X.509 certificate request from the Web page as shown in Figure 90 and follow up with the instructions provided by the certification authority.

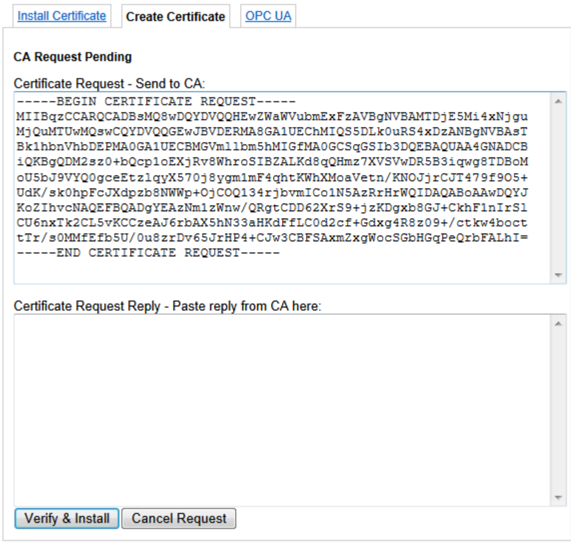


Figure 90: Copy and paste for the X.509 certificate request.

3. Order the certificate. The L-INX requires the certificate to be encoded in PEM format in order to be pasted easily.

4. After receiving the certificate, copy it to the clipboard or a text file. It should look like this:

```
-----BEGIN CERTIFICATE-----
MIICyjCCAjOgAwIQEBBQUAMH4xCzAJBgNV…
… more data follows …
-----END CERTIFICATE-----
```

5. On the tab **Create Certificate** paste the information to the **Certificate Request Reply** text area as shown in Figure 90 and click **Verify & Install**.

6. After next reboot, the server uses the imported certificate, so that the web browsers will indicate the page as trustworthy.

7. Note that SSL certificates have a lifelime, typically 1 or 2 years. You need to repeat these steps to renew your certificates before they expire.

Optionally, a certificate can also be installed from a file. Go to the **Install Certificate** tab as shown in Figure 91 Select the certificate in the **Server certificate** field and its private key in the **Server private key** field. Both can be in PEM or DER (*.der/*.cer) format.

Important! ***You cannot install a Server certificate without its private key!***

If your certification authority uses intermediate certificates, import these **CA certificates** in the CA certificate text field (same format). Press **Save** to import and store the certificates and the server certificate private key. If you want to remove your custom certificate, click on **Reset certificate.**

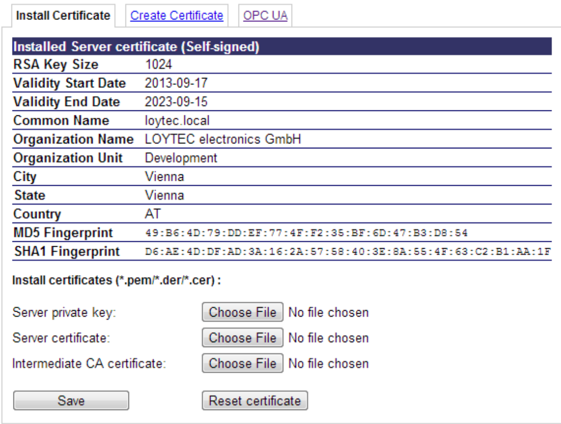


Figure 91: Install a certificate on the Web interface.

5.2.28 Firmware

The firmware page allows upgrading the device's firmware over the Web interface. It offers two options:

- **Web Update**: With Web update the device searches for the latest available firmware on the LOYTEC server. Click on the refresh symbol, if no latest version is displayed. Please note, that the device must have a DNS server configured to find the LOYTEC server. Click on the **Install** button to upgrade your device.

- **Local file**: Update the device from a local disk file. For doing so, choose a .dl file on you hard drive and then click on the **Start Update** button.

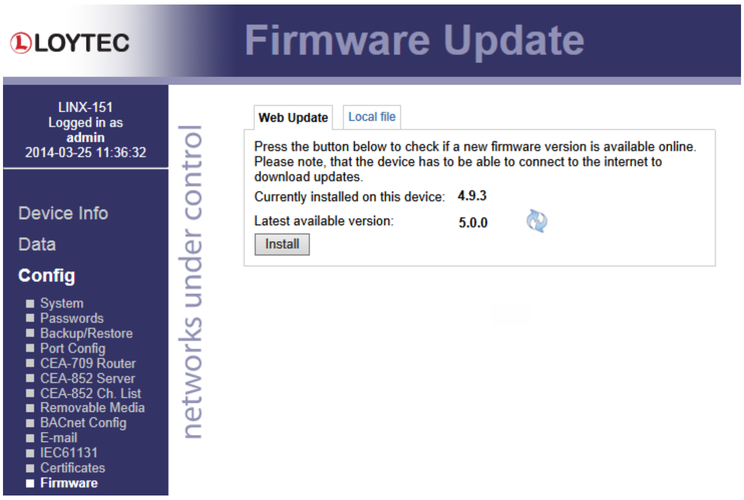


Figure 92: Firmware upgrade over the Web interface.

5.2.29 SNMP

The device has a built-in SNMP server. All system registers and OPC-exposed data points are available as variables in the SNMP management information base (MIB). The MIB definition can be downloaded from the Web interface as shown in Figure 93. One can choose between a text and an XML format, depending on the SNMP tool in use. For more information on SNMP on the device please refer to Section 18.4.

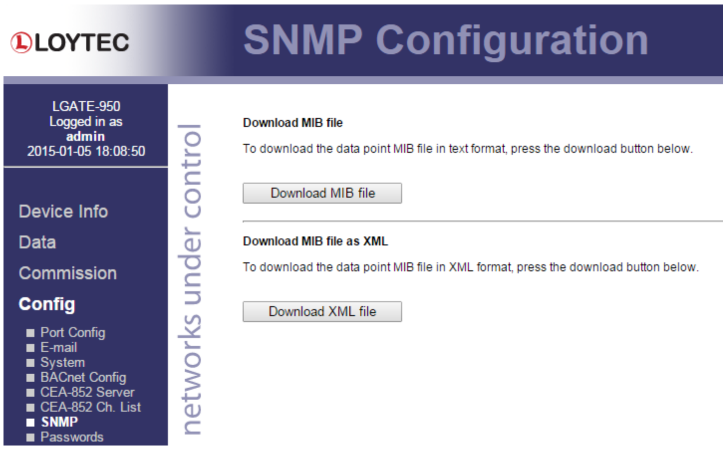


Figure 93: Get the SNMP MIB from the Web interface

5.2.30 Documentation

The **Documentation** page in the **Config** menu allows uploading documentation files or configuring links to external documentation (e.g. Wiring plans, etc.). The documentation configured on this page is accessible via the **Documentation** menu (see Section 5.6).

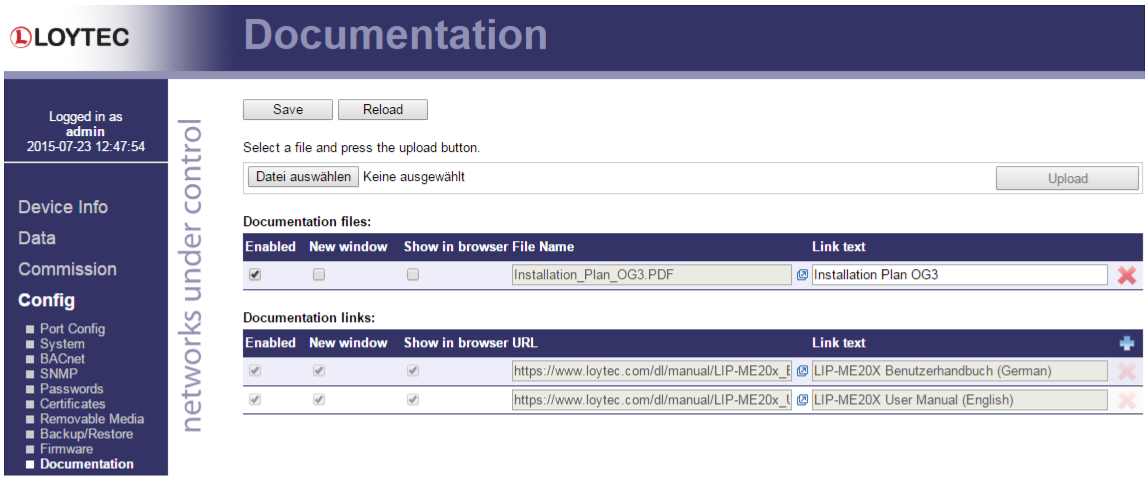


Figure 94: Upload and configure documentation.

To upload a documentation file click on the **Choose File** button. This opens a file dialog. Chose the file to upload. Click on the **Upload** button to start the upload of the selected file. After the upload is completed the file appears in the **Documentation files** section. Enter a link text used to display the uploaded file on the **Documentation** page.

To add a documentation link, click on the symbol in the header row of the **Documentation links** section. Enter the URL and the text used to display the link on the **Documentation** page.

Links and files can be set active and inactive on the **Documentation** page by checking the **Enabled** check box. Inactive entries are not displayed on the **Documentation** page. The check box **New window** determines if the link or file is opened in a new browser tab. If **Show in browser** is checked the browser will try to render the file in the browser, otherwise it will try to download the file. To remove a link or file click on the symbol on the right side of the row. To commit your changes click on the **Save** button.

5.3 Data Management

5.3.1 Data Points

The device's Web interface provides a data point page, which lists all configured data points on the device. An example is shown in Figure 95. The data point page contains a tree view. Clicking on a particular tree item fills the right part of the page with a data point list of that tree level. A breadcrumb navigation header allows navigating back to an arbitrary level in the tree.

The data point list displays the data point name, direction, type, data point state, current value, engineering unit and a description. All values are updated live. Inactive points are displayed in gray. If the data point list does not fit on one page, there are page enumerator links at the bottom. Important data point states and their implications are listed in Table 13. Values can be directly edited in the list where the pencil symbol appears. Data point structures can be expanded or collapsed for better overview. For the structure top a compact list of all member values is displayed including units.

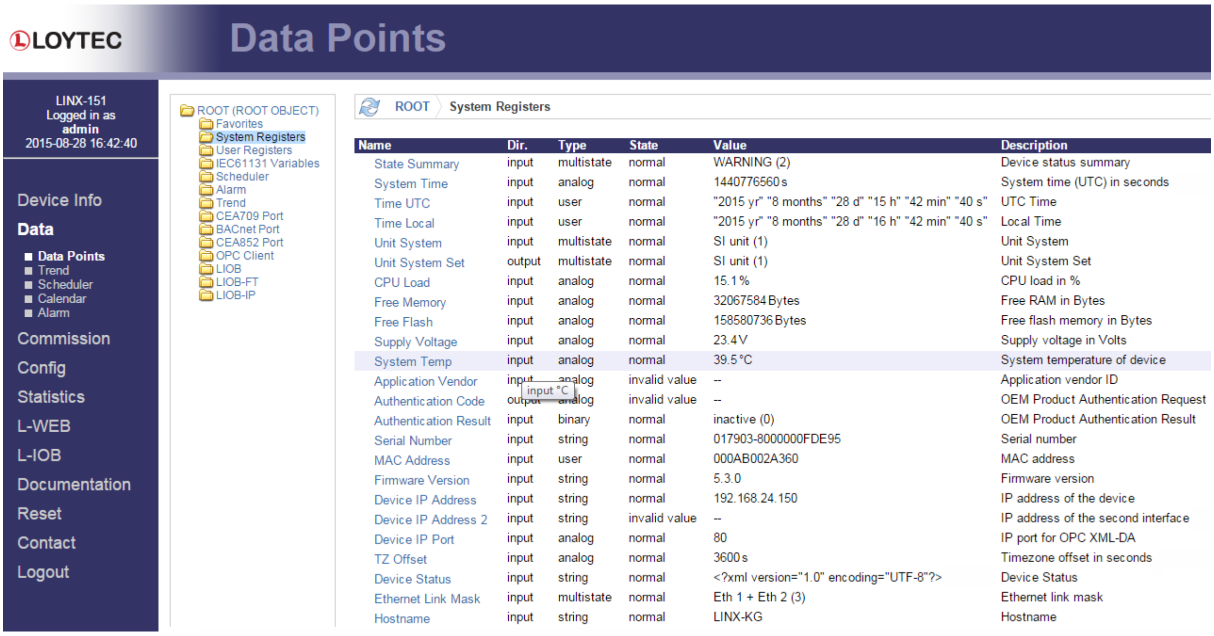
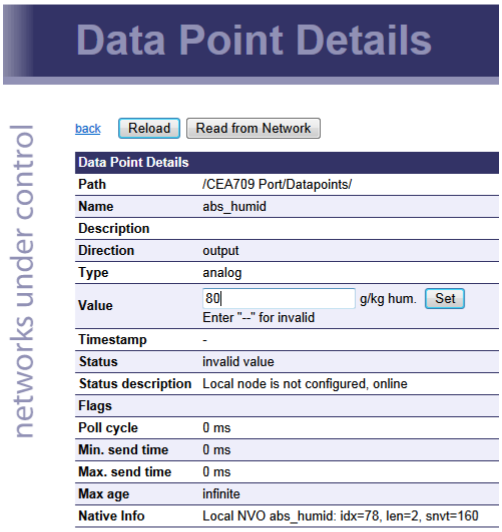


Figure 95: Data point page.

Data Point Status	Description
normal	The data point is in normal operation state and possesses a value.
invalid value	The data point has no valid value.
normal (config)	The data point has a normal value but it is not fully configured on the network (not commissioned, no binding, no client mapping, etc.)
offline	The data point has a value but it is not reflected on the network due to a communication error (e.g., the peer node is not online).
unreliable (offline)	The data point is in normal operation. The value of it, however, is qualified as unreliable because a connected data point is offline. For an output data point it means that the value was fed from a connection, where the source is offline. For an input data point it means that the connected output data point could not send the value to the network.
unreliable (range)	The data point is in normal operation. The value of it, however, is qualified unreliable because the value is an out-of-range value for the connected data point. The value is limited to the supported range.
unreliable	The data point is in normal operation. The value of the data point or a connected data point has been tagged as unreliable over the network. This is the case when the BACnet reliability has been written.
not configured	The data point is mapped to a port, which is not configured (e.g., the port is disabled).
inactive	The data point is inactive and the line is grayed-out. Values can be written but no network communication is triggered. This can be the case, if a data point is not used in the configuration or it is connected to a BACnet server object, which is not present on the device.

Table 13: Data Point States.

The data point names are links. Clicking on such a link opens a detailed page on that data point. If the data point supports it, the user can also enter a new data point value as depicted in Figure 96. For a structured data point the member values are displayed and can be edited in a structure grid. The **Status** field is discussed in Table 13. The field **Status Description** contains a describing text for the data point status. The **Flags**, **Poll cycle**, **Min/Max send time** and **Max age** fields are the common timing parameters for the data point. See Section 6.2.2 for a closer discussion on timing parameters. The **Native Info** field displays detailed information on the underlying technology object.



Data Point Details	
Path	/CEA709 Port/Datapoints/
Name	abs_humid
Description	
Direction	output
Type	analog
Value	80 g/kg hum. Set Enter "--" for invalid
Timestamp	-
Status	invalid value
Status description	Local node is not configured, online
Flags	
Poll cycle	0 ms
Min. send time	0 ms
Max. send time	0 ms
Max age	infinite
Native Info	Local NVO abs_humid: idx=78, len=2, snvt=160

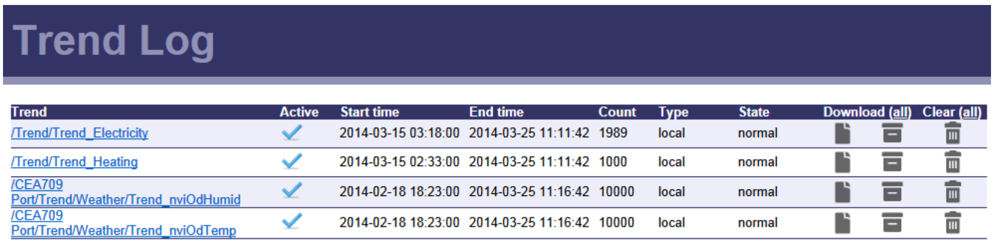
Figure 96: Data point details page.

Clicking on the **Set** button writes the new value to the device's data server. When setting a value, the Web page displays the status of the action:

- **Successfully set value**: The new value has been successfully set in the data point and the update has been sent on the network, if it is a network data point.
- **Could not send value update**: The new value has been set but it has not been sent out on the network. The reason can be that the peer node is currently offline or there is a configuration error. The data point status reflects this error.
- **Could not set value (error code)**: The new value has not been set because of an internal error. Please contact LOYTEC with the error code.

5.3.2 Trend

The Web interface provides a trend log overview page to see all available trend logs and their current state (active, first available date/time, last date/time, number of records). An example is shown in Figure 97. This list allows a convenient upload of single trend data in CSV format by clicking on the respective icons. To upload an archive of all trend data click on the **all** link in the **Download** column heading. It is also possible to purge single or all trend logs directly from that list.



Trend	Active	Start time	End time	Count	Type	State	Download (all)	Clear (all)
/Trend/Trend_Electricity	✓	2014-03-15 03:18:00	2014-03-25 11:11:42	1989	local	normal		
/Trend/Trend_Heating	✓	2014-03-15 02:33:00	2014-03-25 11:11:42	1000	local	normal		
/CEA709 Port/Trend/Weather/Trend_nviOdHumid	✓	2014-02-18 18:23:00	2014-03-25 11:16:42	10000	local	normal		
/CEA709 Port/Trend/Weather/Trend_nviOdTemp	✓	2014-02-18 18:23:00	2014-03-25 11:16:42	10000	local	normal		

Figure 97: Trend log overview on Web UI.

Click on a trend log and re-configure local trend logs at run-time. The changes made to the trend logs take effect immediately without the needs for a reboot of the device. Allocating new trend logs can only be done in the configuration software (see Section 7.14.1). The trend log main page displays all available trend logs. Click on the trend log to be edited. This opens the trend log configuration page. An example is shown in Figure 98.

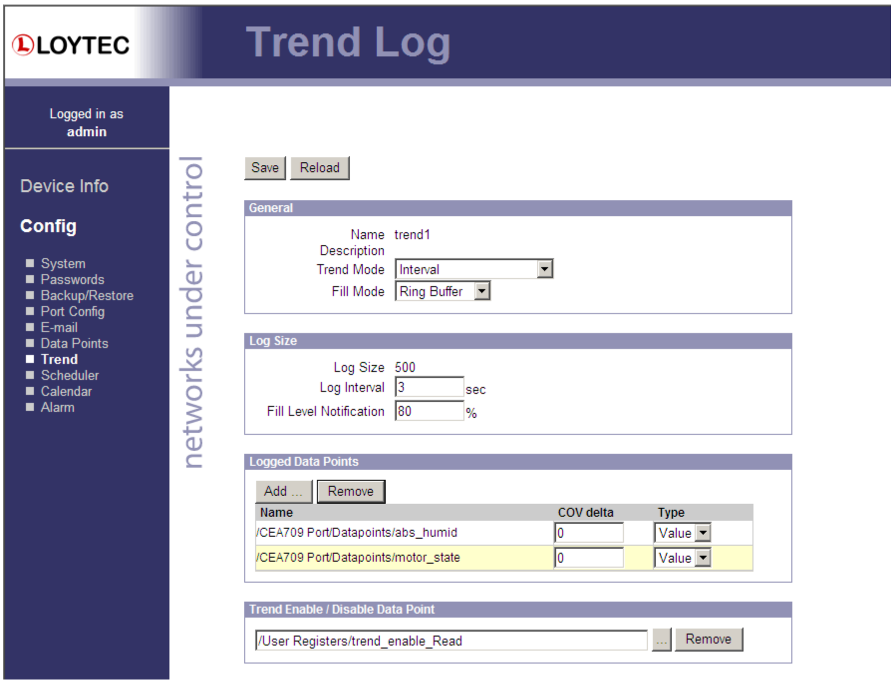


Figure 98: Trend log configuration page.

The user can change the **Trend Mode**, the **Fill Mode**, the **Log Interval** and the **Fill Level Notification**. Furthermore, data points can be added to the trend log by clicking the **Add…** button. A data point selector dialog opens. Click on a data point for adding it. For removing a data point from the trend log, click on it in the **Logged Data Points** list and hit the **Remove** button. Save the changes made by clicking the **Save** button. For more information on how a trend log can be configured please refer to the Configurator Section 7.14.2.

To look at the historical trend data in a chart view select the **Preview** tab as shown in Figure 99. Trend logs with multiple data points are shown with multiple color-coded curves. A legend at the bottom of the page identifies the trended data points. Moving the mouse over the trend chart shows a data curser displaying time stamp and actual value.

Using the chart slider below the trend chart, one can zoom in and out in time as well as shift the time axis. Click into the slider and drag the mouse while keeping the button pressed in order to span a sub-interval, which is displayed in the chart view. Alternatively, select one of the pre-defined sub-intervals (week, day, etc.) and drag the sub-interval along the time axis.

Data points can be deselected in the legend at the bottom of the window. This hides the respective curves in the chart view and may improve visibility for certain detail. Enable the data points again and the curves will re-appear.

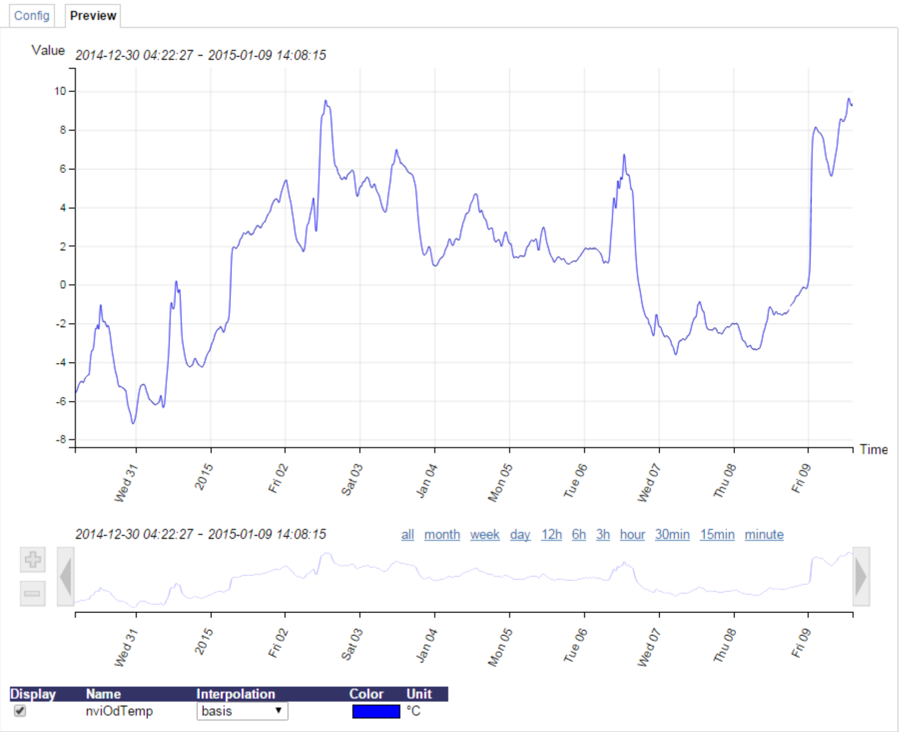


Figure 99: Web UI trend chart preview

5.3.3 Scheduler

The Web interface provides the scheduler page to edit its schedules at run-time, i.e., change the times and values that shall be scheduled. Allocating new schedules and attaching data points to those schedules can only be done in the configuration software (see Section 7.12). The scheduler main page displays all available schedules. Click on the schedule to be edited. This opens the scheduler page. An example is shown in Figure 100.

The **effective period** defines when this schedule shall be in effect. Leave **From** and **To** at '*.*.*' to make this schedule always in-effect. Otherwise select the desired start and/or end dates by clicking the calendar icons. To entirely disable a scheduler de-select the **Enable Schedule** check box.

Schedules are defined per day. On the left-hand side, the weekdays **Monday** through **Sunday** can be selected, or exception days from the calendar, e.g., Holidays. Once a day is selected, the times and values can be defined in the daily planner on the right-hand side. In the example shown in Figure 100, on Monday the value **day** is scheduled at **8:00am**. The same principle applies to **exception days**. **Exception days** override the settings of the normal weekday. Put a check mark on those exception days from the calendar, which shall be used in the schedule. To edit the date ranges of exception days click on the links to the used calendars, e.g., 'calendar' or 'Scheduler_1'. The 'Scheduler_1' is a calendar, which is embedded into the schedule and not accessible by other schedulers. For more information on how to set up schedules and calendars refer to Section 7.12.

To define actual values for the names such as **day** click on the tab **Presets** as shown in Figure 101. To define a new value, click on the button **Add Preset**. This adds a new column. Enter a new preset name (e.g., 'day'). Then enter values for the data points in the **preset** column. The **data point description** column displays the short-hand name defined in the configuration software. This description can also be changed on the Web UI.

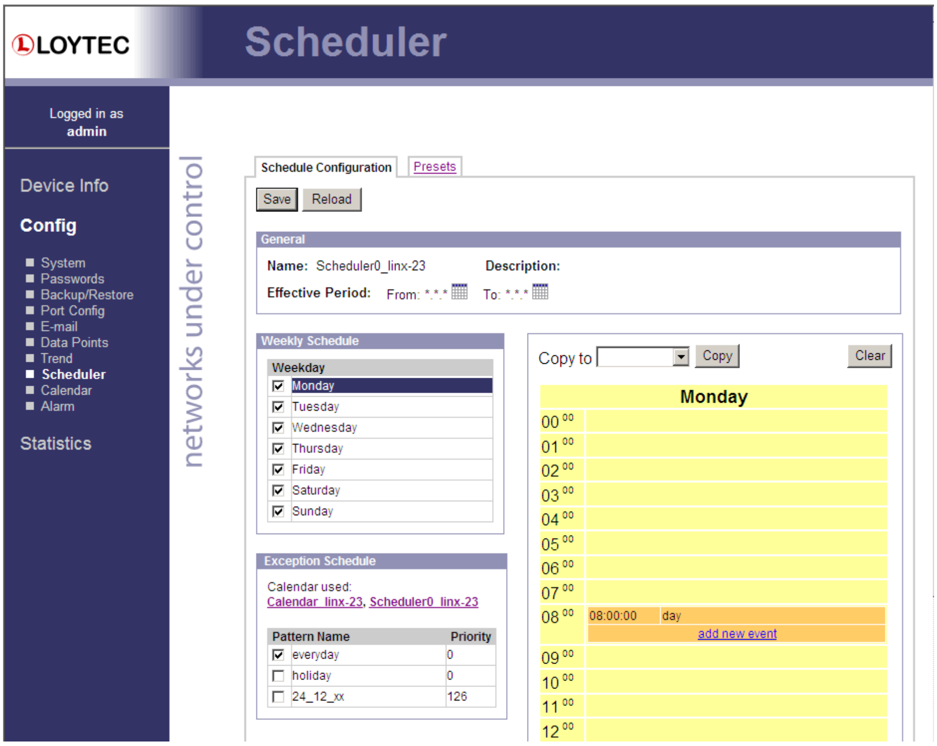


Figure 100: Schedule Configuration Page.

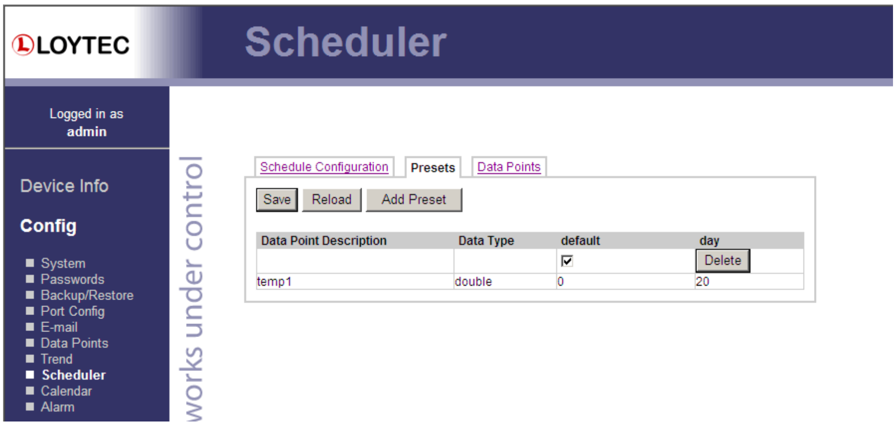


Figure 101: Scheduled Presets Configuration Page.

You can switch back and forth between the two tabs. Once the configuration is complete, click on the **Save** button. This updates the schedule in the device. Any changes made become effective immediately.

Note: *Clicking **Save** may remove any presets which are currently not used in any of the daily schedules. This happens for example in native BACnet schedules, where the underlying network technology cannot store presets individually. Therefore always complete the daily schedules first and then press save as the last step.*

For local schedulers using the CEA-709 network technology, the Web UI also allows to reconfigure the scheduled data points. This change takes effect immediately without a reboot of the device. To add and remove data points to the scheduler, go to the **Data Points** tab. The configuration page is depicted in Figure 102. To add a new data point, click the **Add…** button. To remove a data point, select the data point in the list **Scheduled Data Points** by clicking on it and then press the **Remove** button. Finally, store the changes by clicking the **Save** button. After modifying the scheduled data points, go back to the Presets tab and enter descriptive value label names. For more information on how to configure a scheduler please refer to the Configurator Section 7.12.3.

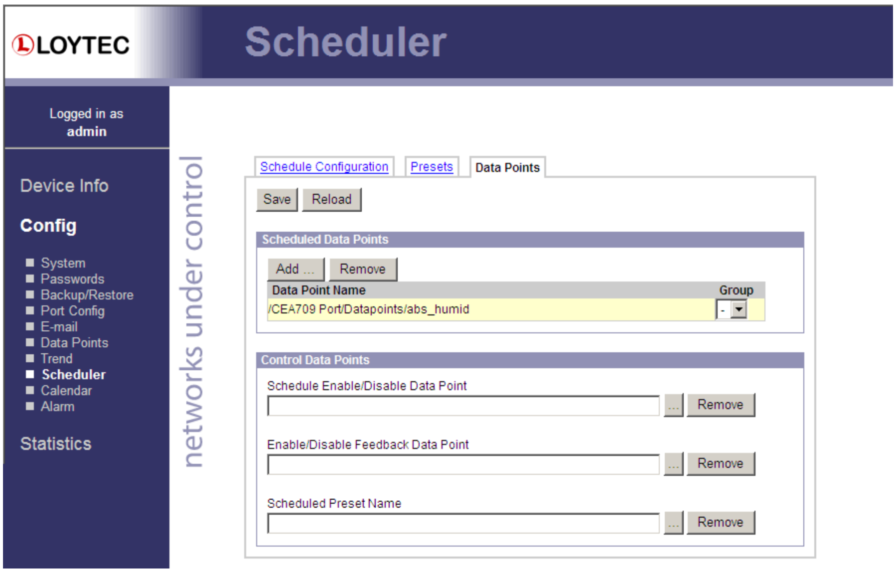


Figure 102: Re-configure scheduled data points on the Web UI.

5.3.4 Calendar

The Web interface provides the calendar page to edit its calendars at run-time, i.e. change the exception days. The calendar main page displays all available calendars. Click on the calendar to be edited. This opens the calendar configuration page. An example is shown in Figure 103.

The **effective period** defines when this calendar shall be in effect. Leave **From** and **To** at '*.*.*' to make this calendar always in-effect. Otherwise enter dates, such as '30.1.2000'.

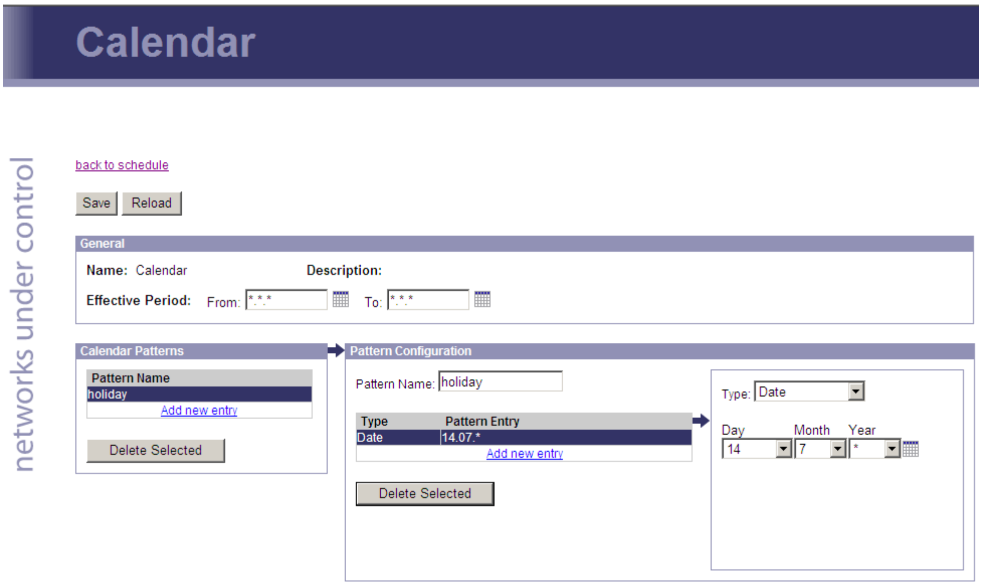


Figure 103: Calendar Configuration Page.

On the remainder of this page, work from left to right. Click on a calendar pattern or create a new calendar pattern by clicking **Add new entry**. A calendar pattern defines a set of pattern entries, which defines the actual dates or date ranges. In the example in Figure 103, the calendar pattern **Holidays** is selected.

In the **Pattern Configuration** box, the calendar pattern's name can be edited if supported by the underlying network technology. Otherwise, an auto-generated name will be assigned and the pattern name box is not shown. Below the pattern name is a list of the individual pattern entries. New entries can be added by clicking **Add new entry**.

Note: *Embedded calendar patterns can only have exactly one entry to define the dates at which the pattern should be in effect. Only calendar patterns in global calendars may consist of multiple entries.*

Existing entries can be selected and edited in the box on the right-hand side. In the example in Figure 103, the date **14.7.*** is selected, which means "The 14.7. of every year". Other entry types such as **Date Range** and **Week-and-Day** can be selected. See Section 6.5.3 for more information about defining exception dates.

5.3.5 Alarm

The Web interface provides the alarm page to view the currently pending alarms of its alarm data points. The alarm main page displays all available alarm data points. Alarm objects which have active alarms are displayed in red. Click on the alarm object to be viewed. This opens the alarm summary page. An example is shown in Figure 104.

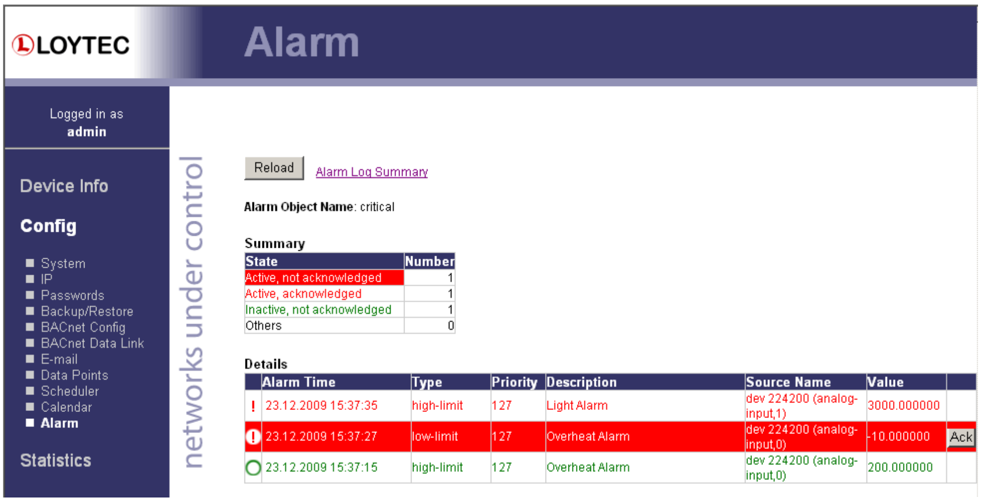


Figure 104: Alarm Summary Page.

Active alarms are highlighted red. Inactive alarms which have not been acknowledged are rendered in green. Alarms that can be acknowledged have an **Ack** button. Press on the **Ack** button to acknowledge the alarm. Depending on the technology, this and older alarm records will be acknowledged. Acknowledged, active alarms are rendered in red. Click on **Reload** to refresh your alarm list.

Inactive alarms that have been acknowledged disappear from the list. To record historical information about those alarms, the alarm log must be used. See Section 5.5.13 for the alarm log Web interface.

5.4 Commission

5.4.1 BACnet

The commissioning Web UI allows assignment of physical devices to existing devices in the data point configuration, that have been created with the commission later option. Under the **Commission** menu choose the BACnet technology to open the BACnet commissioning interface.

The Web page shows a list of all **Devices in configuration**. An example is shown in Figure 105. Each line represents a device and shows the device name, the device **Instance** and the optional BACnet **Address**. The **Static Binding** checkbox defines, whether static device binding is configured for this device and requires a BACnet address. The **Status** column shows their current status. It can be one of the following:

- OK: The device is configured for communication.
- Offline: The device is configured for communication but appears offline.
- Uncommissioned: The device is not yet commissioned.
- Disabled: The device is disabled.

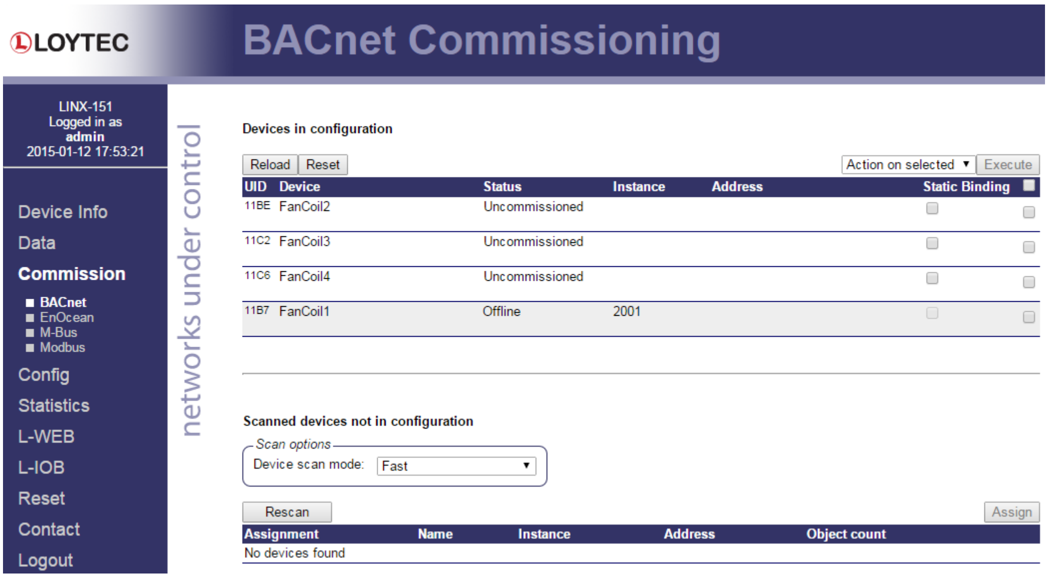


Figure 105: BACnet commissioning Web interface.

In order to execute an action on devices, select the checkbox at the end of the respective lines. Then choose an action in the drop-down **Action on selected** and click the **Execute** button. Actions that can be executed on all devices are enable and disable. A disabled device will stop communication on the network until it is enabled again.

Those devices created as commission later can be assigned to physical devices on the network. The device description displayed beneath the device name can be edited, where the edit symbol appears. The assignment can be done manually by editing the fields in the **Instance** column and **Address** column (for static device binding). It can also be done by executing a network scan. Edit the scan options as appropriate for your BACnet network and click on **Rescan**. The scan progress will be displayed and fill the list for **Scanned devices not in configuration**. An example is shown in Figure 106.

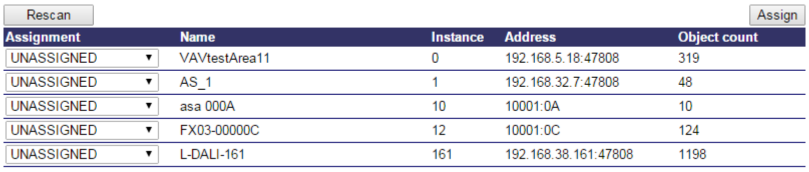


Figure 106: Result of a BACnet scan on the Web interface

To assign a scanned device to an uncommissioned device in the configuration, select the corresponding device name from the drop-down box in the **Assignment** column. Repeat that for all other devices and then click the button **Assign**.

5.5 Device Statistics

The device statistics pages provide advanced statistics information about the CEA-709 device, the CEA-852 device, the BACnet device, the system log, the scheduler, the alarm log and the Ethernet interface.

5.5.1 System Log

The **System Log** page prints all messages stored in the system log of the device. An example is shown in Figure 107. This log data is important for trouble-shooting. It contains log entries for reboots and abnormal operating conditions. Errors and warnings are color-coded in red and yellow. The default log direction is newest entries on top. The direction can be edited by clicking on the arrow ⇑ in the column header.

To save the log click on the **Save System Log** button. When contacting LOYTEC support, have a copy of this log ready.

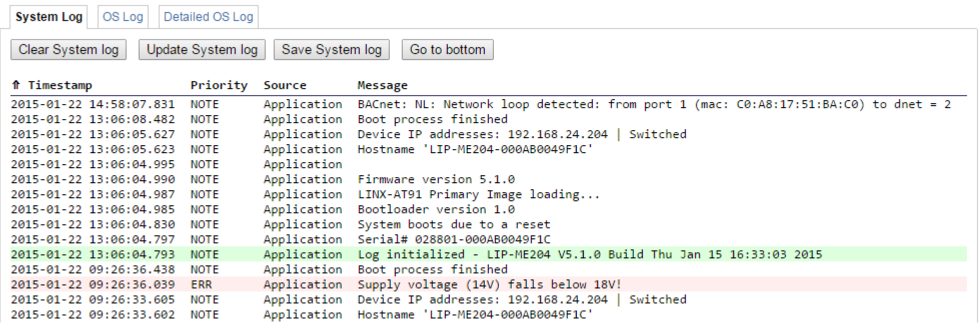


Figure 107: System Log Page.

5.5.2 IP Statistics

Figure 108 shows the IP statistics page. It allows finding possible problems related to the IP communication. Specifically, any detected IP address conflicts are displayed (if the device's IP address conflicts with a different host on the network).

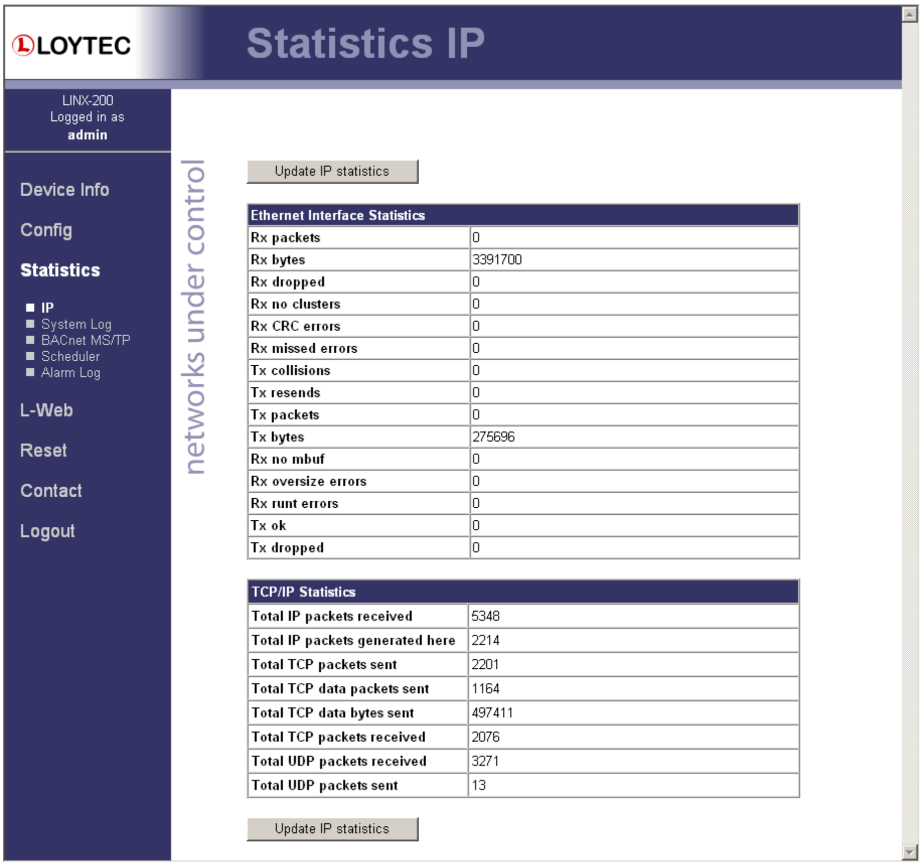


Figure 108: IP Statistics Page.

5.5.3 E-mail

The E-mail statistics page shows information regarding the devices SMTP client (e-mail transmission). This includes information regarding the number of messages queued for transmission (Queued currently/total/max), transmitted messages (delivery successful/failed/failed after retry) and dropped messages (with reason for dropping). Maximum time for gethostbyname shows needed time to resolve DNS names. If the maximum time is high, a problem with DNS servers is likely. Maximum time for SMTP transfer adds up DNS and e-mail transfer time.

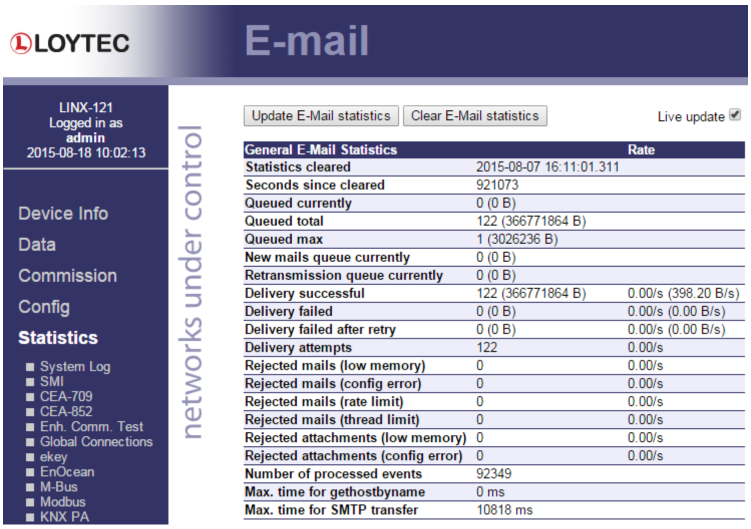


Figure 109: E-mail Statistics.

5.5.4 CEA-852 Statistics

The CEA-852 statistics page displays the statistics data of the CEA-852 device on the device. It is only displayed if the CEA-852 interface is enabled and supported by the device model. The upper part of the CEA-852 statistics page is depicted in Figure 110. To update the statistics data, press the button **Update all CEA-852 statistics**. To reset all statistics counters to zero, click on the button **Clear all CEA-852 statistics**. The field **Date/Time of clear** will reflect the time of the last counter reset.

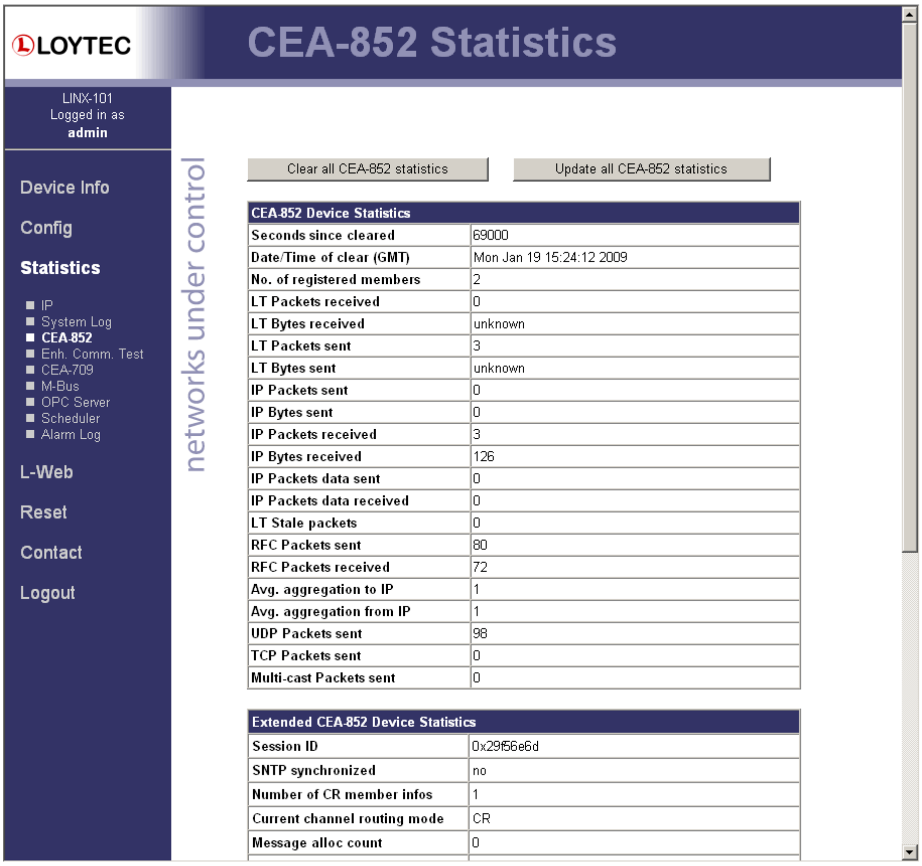


Figure 110: Part of the CEA-852 Statistics Page.

5.5.5 Enhanced Communications Test

The Enhanced Communications Test allows testing the CEA-852 communication path between the CEA-852 device on the L-INX/L-GATE and other CEA-852 devices as well as the configuration server. The test thoroughly diagnoses the paths between individual members of the IP channel and the configuration server in each direction. Port-forwarding problems are recognized. For older devices or devices by other manufacturers, which do not support the enhanced test features, the test passes as soon as a device is reachable, but adds a comment, that the return path could not be tested. A typical output is shown in Figure 111.

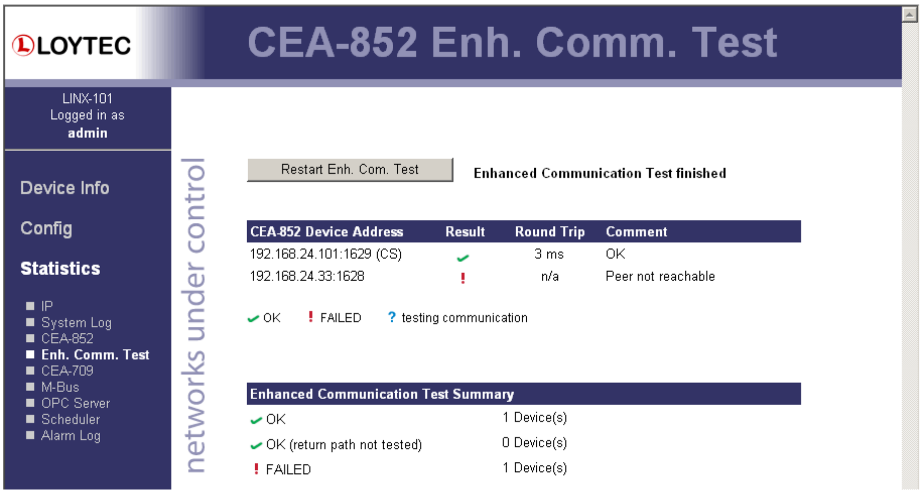


Figure 111: Enhanced Communication Test Output.

The Round Trip Time (RTT) is measured as the time a packet sent to the peer device needs to be routed back to the device. It is a measure for general network delay. If the test to a specific member fails, a text is displayed to describe the possible source of the problem. The reasons for failure are summarized in Table 14.

Text displayed (Web icon)	Meaning
OK, Return path not tested (green checkmark)	Displayed for a device which is reachable but which does not support the feature to test the return path (device sending to this CEA-852 device). Therefore a potential NAT router configuration error cannot be detected. If the tested device is an L-IP, it is recommended to upgrade this L-IP to 3.0 or higher.
Not reachable/not supported (red exclamation)	This is displayed for the CS if it is not reachable or the CS does not support this test. To remove this uncertainty it is recommended to upgrade the L-IP to 3.0 or higher.
Local NAT config. Error (red exclamation)	This is displayed if the CEA-852 device is located behind a NAT router or firewall, and the port-forwarding in the NAT-Router (usually 1628) or the filter table of the firewall is incorrect.
Peer not reachable (red exclamation)	Displayed for a device, if it is not reachable. No RTT is displayed. The device is either not online, not connected to the network, has no IP address, or is not reachable behind its NAT router. Execute this test on the suspicious device to determine any NAT configuration problem.

Table 14: Possible Communication Problems.

5.5.6 Global Connections Statistics

The global connections statistics page shows all currently configured communication groups. For each group the list displays name, address hash, receive, transmit, poll-on-startup status, the most recently communicated value and its timestamp. An example is shown in Figure 112. The receive/transmit/poll-on-startup status displays an ✕ if the direction is configured, but no value was communicated. A green check mark ✓ is shown as soon as a value was received or transmitted, respectively.

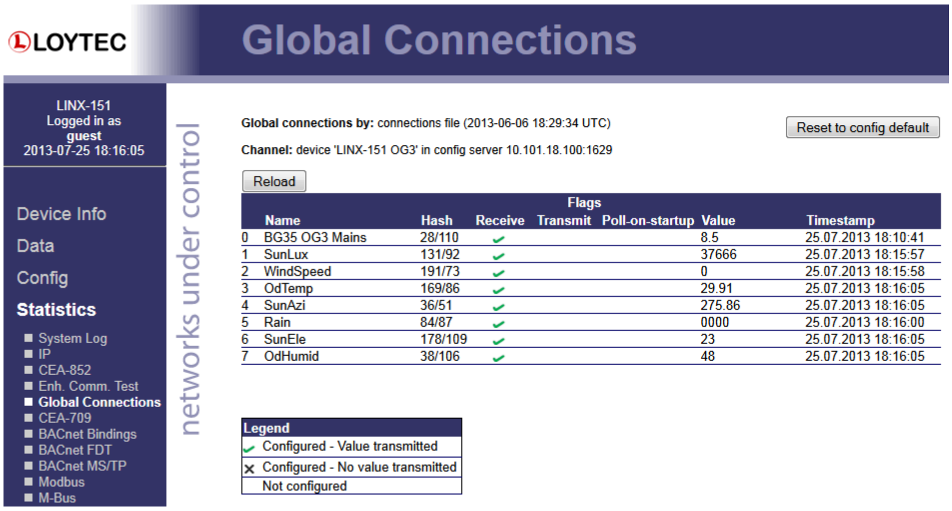


Figure 112: Global Connections Statistics

The **Reload** button refreshes the status. The button **Reset to config default** removes any global connections configured by LWEB-900 at run-time and reverts to the configuration default. A reboot is required in this case.

5.5.7 CEA-709 Statistics

The CEA-709 statistics page displays statistics data of the CEA-709 port on the device as shown in Figure 113. This data can be used to troubleshoot networking problems. To update the data, click on the button **Update CEA-709 statistics**.

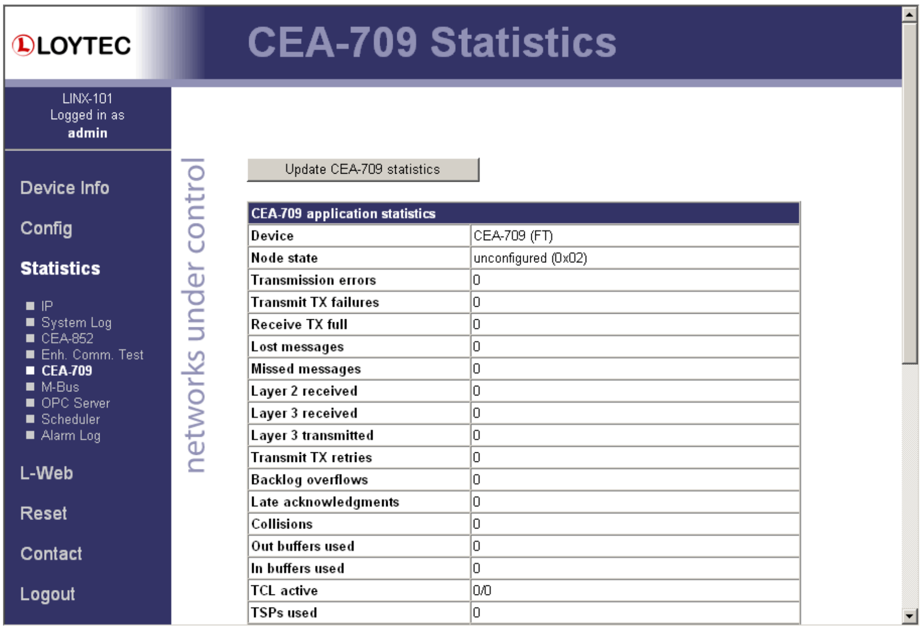


Figure 113: CEA-709 Statistics Page.

5.5.8 OPC XML-DA Server Statistics Page

The OPC XML-DA server statistics page shows statistics data, which contains information on currently and previously connected clients. An example list of OPC clients is shown in Figure 114. Clicking on the **Update OPC XML-DA statistics** button retrieves the current statistics.

The **Summary** table on the top of the page displays the number of currently connected clients. These clients occupy TCP connections. The next line specifies the total number of accepted client connections since the device is running. The figure for rejected connections can be used to detect situations, where too many clients try connecting at the same time.

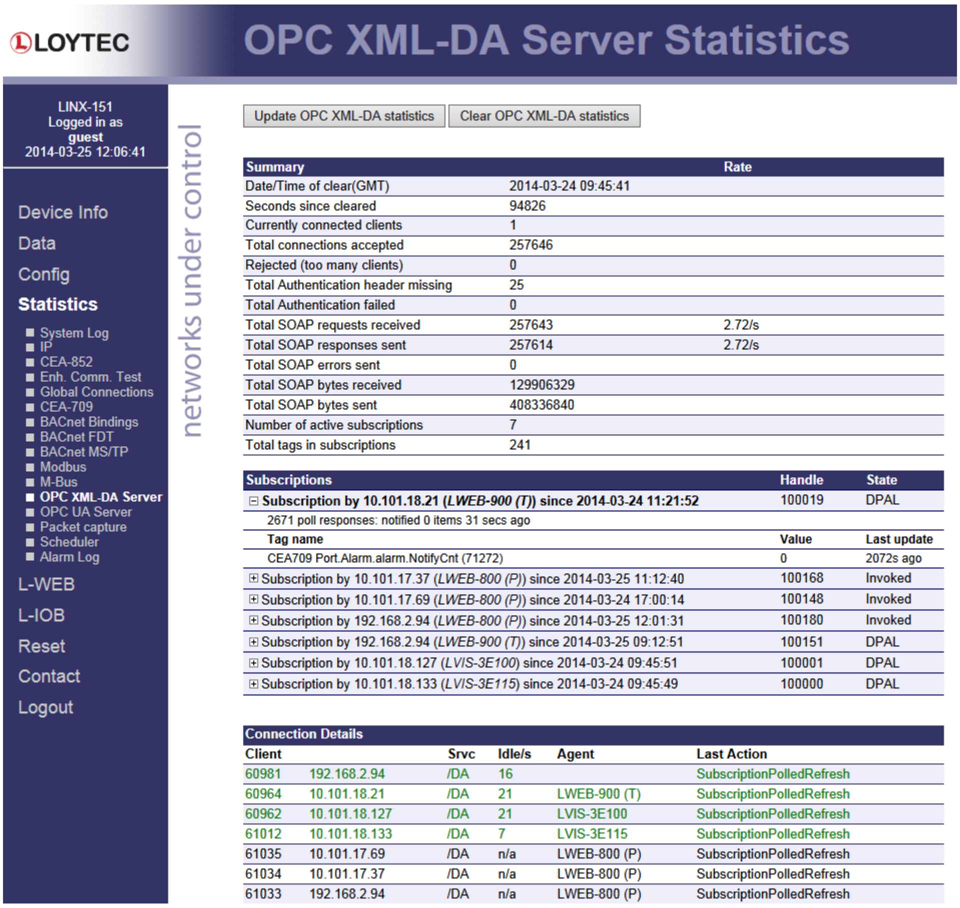


Figure 114: OPC XML-DA Server Statistics Page.

The **Subscriptions** list shows detailed information on OPC items subscribed by OPC XML-DA clients. The subscriptions are sorted by client IP address and can be expanded to show the subscribed items. The **Handle** specifies the server subscription handle and the value represents the last value notified for the respective item in this subscription.

The **Connection Details** list shows more information on the history of client connections. The green lines at the top denote currently active connections. Active connections have an idle time figure specified in seconds. The following lines in black represent a history of the most recent connections. Inactive connections read “n/a” in the **Idle** column.

All lines contain client information, which specifies the client IP address and port of the connected client. The **Srvc** column specifies the type of Web service (Web, DA, and DL). The **Agent** column contains information on the HTTP agent of the client, and the **Last Action** column contains information on the last known Web service SOAP action the client has requested.

5.5.9 BACnet MS/TP Statistics

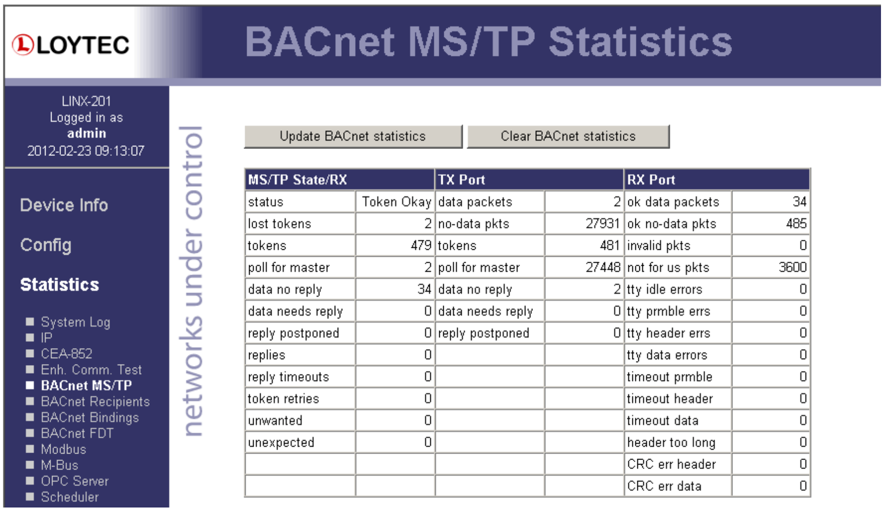
The BACnet MS/TP statistics page is only available, when the MS/TP data link layer is enabled (see Section 5.2.17) and supported by the L-INX model. The three statistics items displayed are: Device Statistics, Bus History, and Token History.

The **MS/TP Device Statistics** (see Figure 115) is split into three major columns, **MS/TP State/RX**, **TX Port**, and **RX Port**. The MS/TP State/RX column contains information related to the status of the MS/TP machine as well as packets received and processed by the MS/TP state machine. The TX Port column counts packets sent by the device according to their types, and the RX Port column tracks packets and errors seen by the MS/TP receive state machine.

The most prominent information in the **MS/TP State/RX** column is the **status** entry which describes the current status of the MS/TP token as perceived by the device. In status **Token Ok**, the token is circulating between the masters. This is the normal state, when multiple masters are on the MS/TP network. The status **Sole Master** is the normal state when the device is the only master on the network. If there are multiple masters on the network, token passing has been interrupted and this state is a hint to a broken cable. In state **Token Lost**, the token is currently not circulating.

While **status** reflects the current state the device is in, the **lost tokens** counter is more indicative for communication problems on the MS/TP network. If it increases, there are cabling, ground, or termination issues.

Note, that the **RX Port** column monitors all packets seen on bus, not only those addressed to the device. Statistics related to received packets that are addressed to the device are tracked in the **MS/TP State/RX** column.



MS/TP State/RX		TX Port		RX Port	
status	Token Okay	data packets	2	ok data packets	34
lost tokens	2	no-data pkts	27931	ok no-data pkts	485
tokens	479	tokens	481	invalid pkts	0
poll for master	2	poll for master	27448	not for us pkts	3600
data no reply	34	data no reply	2	tty idle errors	0
data needs reply	0	data needs reply	0	tty prmble errs	0
reply postponed	0	reply postponed	0	tty header errs	0
replies	0			tty data errors	0
reply timeouts	0			timeout prmble	0
token retries	0			timeout header	0
unwanted	0			timeout data	0
unexpected	0			header too long	0
				CRC err header	0
				CRC err data	0

Figure 115: BACnet MS/TP Device Statistics.

The **MS/TP Bus History** (see Figure 116) presents information related to the MS/TP bus as a whole over the last minute, split into 10 second time slices.

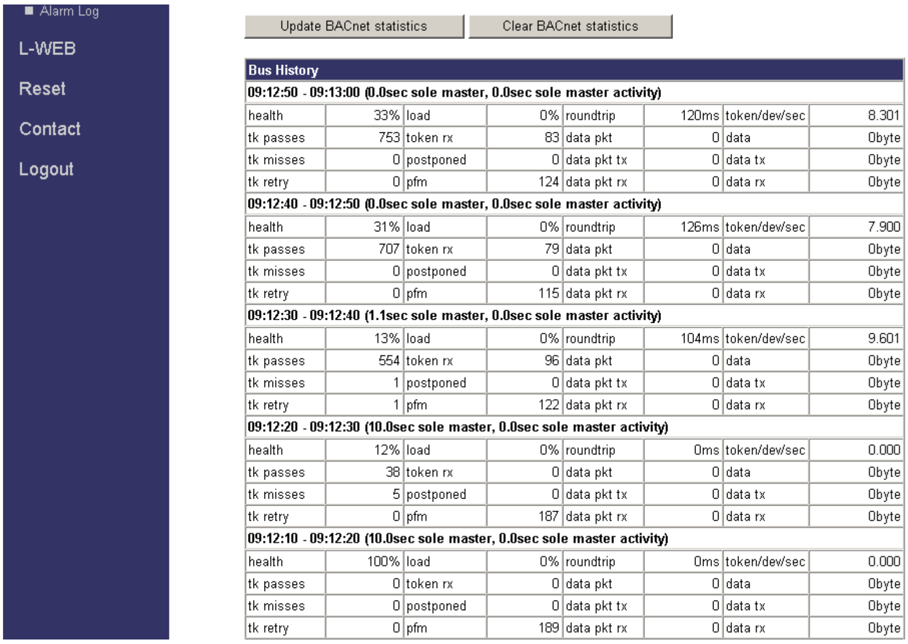
The convenient **health** indicator, a percentage in the range 0 – 100%, gives an overall impression of the communication quality on the bus: The higher the percentage, the better the MS/TP communication between devices on the bus. Reasons for **health** to be low are:

- Superfluous PollForMaster requests (because MS/TP node addresses in use contain gaps or Max_Master of the node with the largest node address is not set to the same value as the node's address),
- token losses,
- reply timeouts,
- slow token passing.

The **load** percentage simply displays how much of the available bandwidth is used for data. Note, however, that actual application data is only a subset of the amount of data taken into account here.

Statistics reflecting the average ability of devices to initiate communication are **roundtrip** and **token/dev/sec**. They give an impression on how long the token requires to circulate once (in milliseconds), and how often a device on the bus receives the token per second.

Other counters of interest are: **tk passes** (the number of times the token was passed), **tk misses** (the number of times the receiver of a token did not continue passing the token), **tk retry** (the number of times passing of token was retried), **postponed** (the number of ReplyPostPostponed packets seen), **pfm** (the number of PollForMaster packets seen), **data pkt**, **data pkt rx**, **data pkt tx** (the number of data packets seen, the number of data packets received and transmitted by the device), ***data*, *data rx*, *data tx*** (the amount of data seen, the amount of data received and transmitted by the device), **token rx** (the number of tokens received by the device).



Bus History							
09:12:50 - 09:13:00 (0.0sec sole master, 0.0sec sole master activity)							
health	33%	load	0%	roundtrip	120ms	token/dev/sec	8.301
tk passes	753	token rx	83	data pkt	0	data	0byte
tk misses	0	postponed	0	data pkt tx	0	data tx	0byte
tk retry	0	pfm	124	data pkt rx	0	data rx	0byte
09:12:40 - 09:12:50 (0.0sec sole master, 0.0sec sole master activity)							
health	31%	load	0%	roundtrip	126ms	token/dev/sec	7.900
tk passes	707	token rx	79	data pkt	0	data	0byte
tk misses	0	postponed	0	data pkt tx	0	data tx	0byte
tk retry	0	pfm	115	data pkt rx	0	data rx	0byte
09:12:30 - 09:12:40 (1.1sec sole master, 0.0sec sole master activity)							
health	13%	load	0%	roundtrip	104ms	token/dev/sec	9.601
tk passes	554	token rx	96	data pkt	0	data	0byte
tk misses	1	postponed	0	data pkt tx	0	data tx	0byte
tk retry	1	pfm	122	data pkt rx	0	data rx	0byte
09:12:20 - 09:12:30 (10.0sec sole master, 0.0sec sole master activity)							
health	12%	load	0%	roundtrip	0ms	token/dev/sec	0.000
tk passes	38	token rx	0	data pkt	0	data	0byte
tk misses	5	postponed	0	data pkt tx	0	data tx	0byte
tk retry	0	pfm	187	data pkt rx	0	data rx	0byte
09:12:10 - 09:12:20 (10.0sec sole master, 0.0sec sole master activity)							
health	100%	load	0%	roundtrip	0ms	token/dev/sec	0.000
tk passes	0	token rx	0	data pkt	0	data	0byte
tk misses	0	postponed	0	data pkt tx	0	data tx	0byte
tk retry	0	pfm	189	data pkt rx	0	data rx	0byte

Figure 116: BACnet MS/TP Bus History.

The **MS/TP Token History** (see Figure 117) shows the most recent token passes on the bus. The syntax used is simple: '`40<15`' means that the node with address '0x15' has passed the token to the node with address '0x40'.

If token losses or token sending retries have been recorded, these are marked by substituting '`.`' for '<'. For example, '`40<15.40<15`' either means that '0x15' retried sending the token to '0x40', or that passing the token to '0x40' failed and '0x15' created a new token and sent it to '0x40'.

Transitions to or from sole master mode can be spotted by looking out for '`XX`', e.g., '`XX<15`' means that after '0x15' received the token, the device entered sole master mode. Finally, based on the recorded token passes, the MS/TP Token History also lists the node addresses of MS/TP masters detected on the bus.

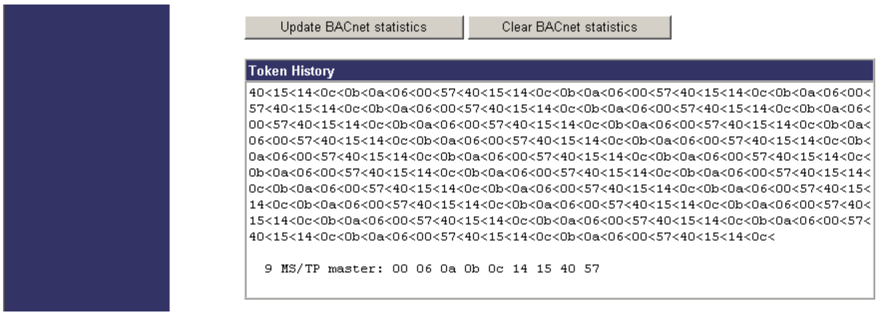


Figure 117: BACnet MS/TP Token History.

5.5.10 BACnet Bindings Statistics

The BACnet bindings statistics page displays a list of all currently active address bindings. This list can be used for troubleshooting to see, which BACnet device instance numbers could be resolved and to what BACnet network number and MAC address. See Figure 118 for an example list. In this case the device instance 224220 has been resolved to the local network and MAC address 192.168.24.220:BAC0.

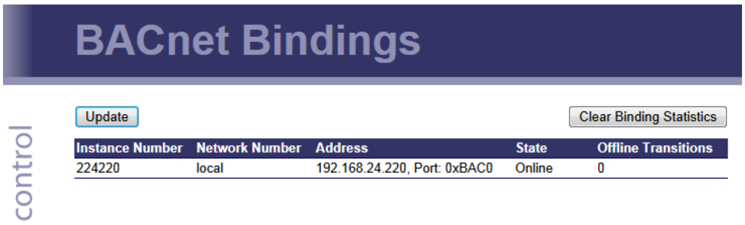


Figure 118: BACnet bindings statistics page.

5.5.11 BACnet FDT Statistics

The BACnet FDT (Foreign Device Table) Statistics page displays a list of all BACnet/IP foreign devices currently registered with the device (see Figure 119). Note, that foreign devices can only register with the device if the latter is configured as BACnet **Broadcast Management Device** (see Section 5.2.16).

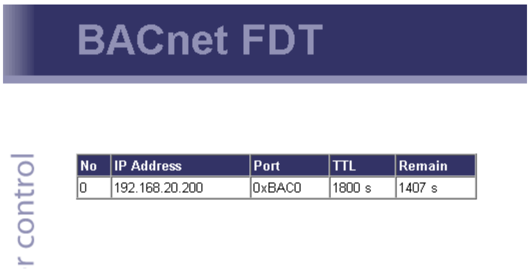


Figure 119: BACnet FDT Statistics page.

5.5.12 Scheduler Statistics Page

The scheduler statistics page provides an overview of what is scheduled at which day and which time. In the **Display Schedules** list, select a single schedule to view its scheduled values and times. Use the multi-select feature to get the overview of more schedules. An example is shown in Figure 120.

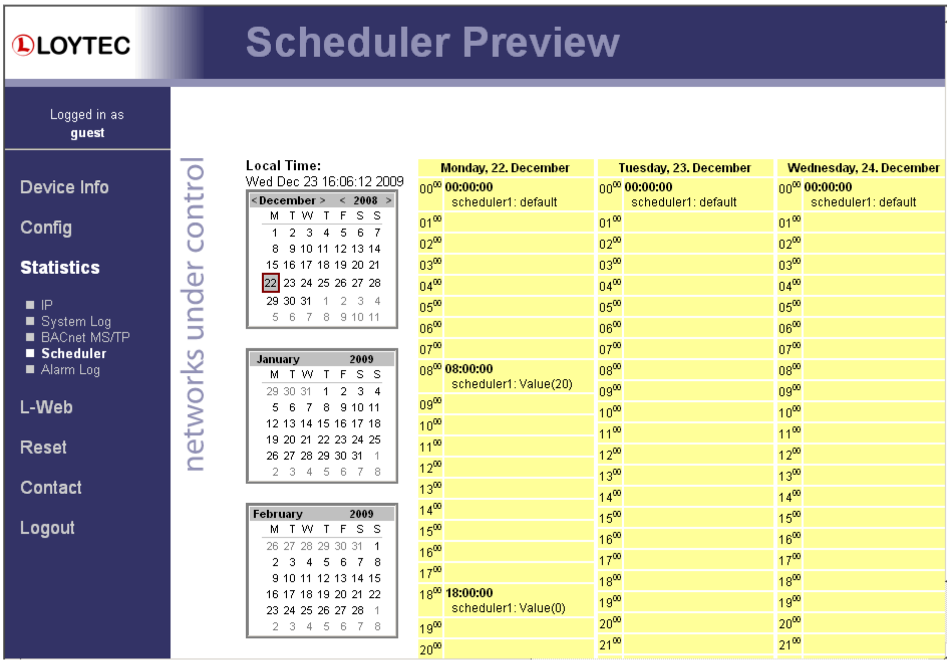
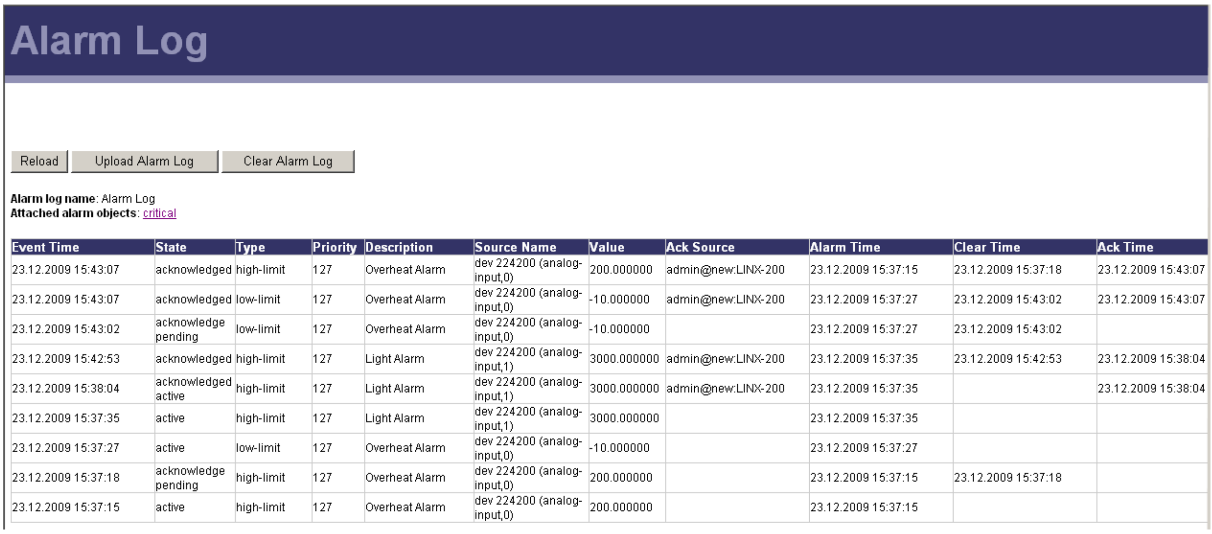


Figure 120: Scheduler Statistics Page.

5.5.13 Alarm Log Page

The alarm log page provides an overview of all alarm logs on the system. Click on one of the links to view a specific alarm log. Each alarm log contains a historical log of alarm transitions. When an inactive and acknowledged alarm disappears from the alarm summary page (live list), the alarm log contains this last transition and maintains it over a reboot. An example is shown in Figure 121.

To refresh the alarm log contents click on the **Reload** button. Currently active alarms cannot be acknowledged in this historical view. Follow the link to the attached alarm objects to get to the respective live lists, where alarms can be acknowledged on the Web interface (see Section 5.3.5).



Event Time	State	Type	Priority	Description	Source Name	Value	Ack Source	Alarm Time	Clear Time	Ack Time
23.12.2009 15:43:07	acknowledged	high-limit	127	Overheat Alarm	dev 224200 (analog-input,0)	200.000000	admin@new:LINX-200	23.12.2009 15:37:15	23.12.2009 15:37:18	23.12.2009 15:43:07
23.12.2009 15:43:07	acknowledged	low-limit	127	Overheat Alarm	dev 224200 (analog-input,0)	-10.000000	admin@new:LINX-200	23.12.2009 15:37:27	23.12.2009 15:43:02	23.12.2009 15:43:07
23.12.2009 15:43:02	acknowledge pending	low-limit	127	Overheat Alarm	dev 224200 (analog-input,0)	-10.000000		23.12.2009 15:37:27	23.12.2009 15:43:02	
23.12.2009 15:42:53	acknowledged	high-limit	127	Light Alarm	dev 224200 (analog-input,1)	3000.000000	admin@new:LINX-200	23.12.2009 15:37:35	23.12.2009 15:42:53	23.12.2009 15:38:04
23.12.2009 15:38:04	acknowledged active	high-limit	127	Light Alarm	dev 224200 (analog-input,1)	3000.000000	admin@new:LINX-200	23.12.2009 15:37:35		23.12.2009 15:38:04
23.12.2009 15:37:35	active	high-limit	127	Light Alarm	dev 224200 (analog-input,1)	3000.000000		23.12.2009 15:37:35		
23.12.2009 15:37:27	active	low-limit	127	Overheat Alarm	dev 224200 (analog-input,0)	-10.000000		23.12.2009 15:37:27		
23.12.2009 15:37:18	acknowledge pending	high-limit	127	Overheat Alarm	dev 224200 (analog-input,0)	200.000000		23.12.2009 15:37:15	23.12.2009 15:37:18	
23.12.2009 15:37:15	active	high-limit	127	Overheat Alarm	dev 224200 (analog-input,0)	200.000000		23.12.2009 15:37:15		

Figure 121: Alarm Log Page.

The alarm log contents can be uploaded from the device in a CSV formatted file. Click on the button **Upload Alarm Log** to upload the current log. To clear the log, press the button

Clear Alarm Log. Please note, that this permanently purges all historical alarm log data of this alarm log.

5.5.14 Packet Capture

The packet capture feature allows configuring and running a local packet capture for the Ethernet and MS/TP ports. Please refer to Section 21.3 for more information on how to set up local capture and configure remote packet capture with Wireshark.

5.6 Documentation

The documentation page allows to access documentation related to the device. See Section 5.2.30 on how to configure documentation links and upload documentation files accessible via this page.

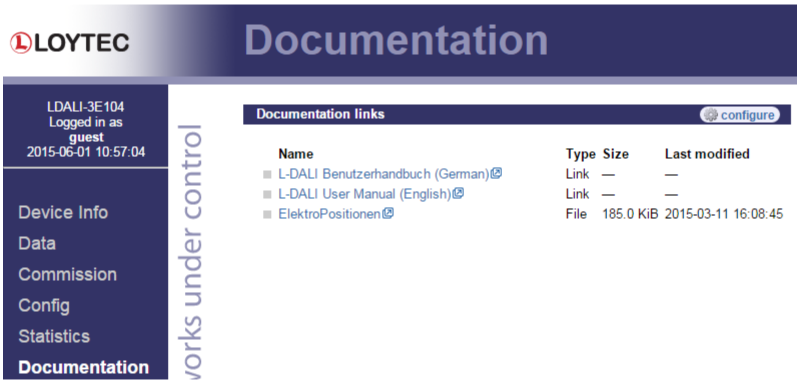


Figure 122: Documentation Page.

Note: *The Documentation page and all files available on it are accessible for all users (incl. Guest).*

5.7 L-WEB

5.7.1 Installation

This configuration page provides a download link to the L-WEB application installer and a listing of L-WEB projects available on the device (see Figure 123). Clicking on **Install** will download the installer for LWEB-803 and start the installation process. Clicking on the Web icon will open the LWEB-802 project in a Web browser. This option is only available, if the L-WEB project has been stored as an lweb2 file. See Section 10.3 for more information on working with the L-WEB visualization.

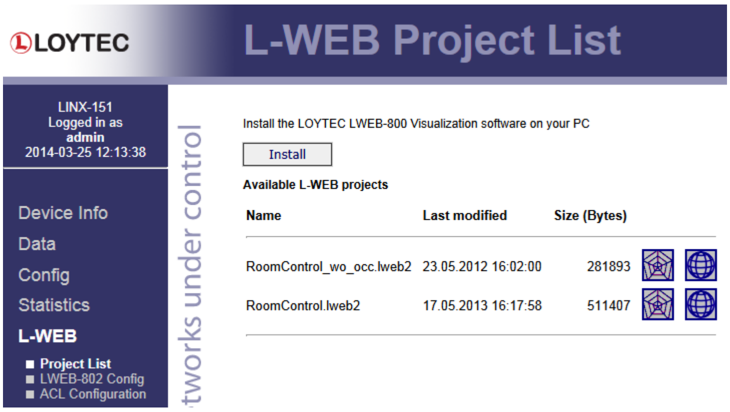


Figure 123: L-WEB Page.

5.7.2 LWEB-802 Config

To operate an LWEB-802 project in a Web browser, the LWEB-802 application must be loaded from a server URL that hosts it. The LWEB-802 configuration page allows specifying the URL used for LWEB-802 projects hosted on the device. The default setting is the LOYTEC Web site, which hosts the official LWEB-802 application release.

For installations that don’t have Internet access or that require a special version of the LWEB-802 application for testing, the application can also be hosted on the Web server of the device. The following options can be configured:

- **LOYTEC Website**: This is the default setting. The URL points to the official LWEB-802 application released on the LOYTEC Web site.
- **Pre-installed on device**: With this setting, a pre-installed local version of the LWEB-802 application is used directly from the device.
- **User-installed on device**: Choose this setting to store a user-supplied version of the LWEB-802 application on the device.
- **Custom URL**: Choose this setting, if the LWEB-802 application has been loaded onto a Web server other than this device. Enter the appropriate URL.

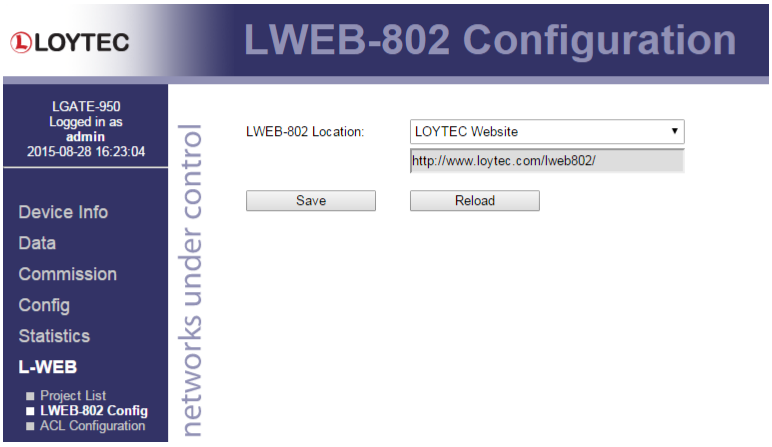


Figure 124: LWEB-802 Configuration Page.

5.7.3 Access Control List

The device provides a security feature to restrict access to the OPC server from the Internet. This feature is based on an access control list (ACL). An example of the ACL configuration is shown in Figure 125.

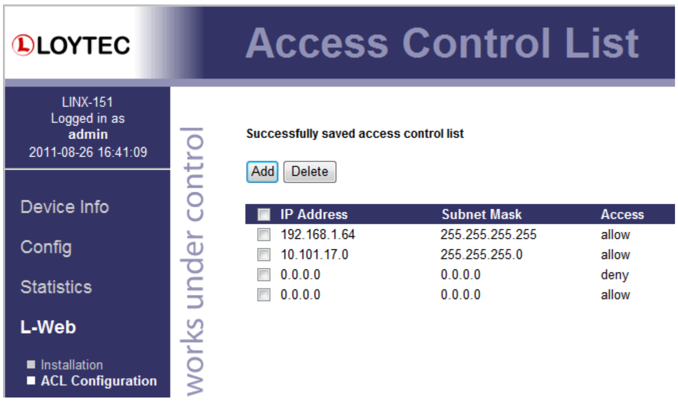


Figure 125: OPC Server Access Control List (ACL).

The user can add and delete entries to the ACL. Each entry contains a source specification, which consists of an IP address and an IP mask, and an action (allow or deny). For specifying single hosts use the IP address and the mask '255.255.255.255'. For an address range specify an appropriate mask. For example use '10.101.17.0' and the mask '255.255.255.0' to specify all hosts with IP addresses '10.101.17.xxx'. To specify all IP addresses use '0.0.0.0' and the mask '0.0.0.0'.

The ACL is evaluated from specific host entries down to wider ranges. When adding new entries the ACL is automatically sorted, having the most precise definition at the top and the most general one at the bottom. The default behavior is to allow packets from all IP addresses. This is also the default entry in the ACL.

The example shown in Figure 125 specifies the following behavior:

1. Allow requests from the device 192.168.1.64
2. Otherwise allow requests from devices in the network 10.101.17.xxx
3. Otherwise deny requests from all (other) IP addresses. Note, that a "deny" overrules an "allow".

5.8 Reset, Contact, Logout

The menu item **Reset** allows the following essential operations:

- Rebooting the device from a remote location.
- Resetting the data point configuration from a remote location. This option clears all data points and the entire port configuration. It leaves the IP settings intact.
- Reverting run-time changes to defaults of the data point configuration. This applies to persistent data points and parameter values.

The **Contact** item provides contact information and a link to the latest user manual and the latest firmware version. The **Logout** item closes the current session.

6 Concepts

6.1 Universal Gateway

The operating principle of a universal gateway is to connect data points of one network technology to data points of another technology. This is the primary function of the L-GATE product family. Gateway function are, however, also available in the L-INX family.

Data points in the CEA-709 network are known as network variables (NVs). For more information on this technology refer to Section 6.6. Data points in the BACnet technology are known as BACnet server objects. They have a specific type (e.g. analog input or binary output) and a set of properties, which describe the data point more closely. The actual value is stored in the “Present_Value”. For more information on this technology refer to Section 6.7.

The typical task in configuring the universal gateway consists of the following steps:

1. Selecting the data points of the network to be mapped (e.g., select the NVs in the CEA-709 network nodes or create new NVs)

2. Select or create matching counterparts of the other technology (e.g., create matching BACnet objects)

3. Create connections between the data points (e.g. connect NVs and BACnet objects).

The connection is the central part of the gateway functionality. It defines, which data points are mapped to which data points. Refer to Section 6.4 about the nature of connections in the device.

6.2 Data Points

6.2.1 Overview

Data points are part of the fundamental device concept to model process data. A data point is the basic input/output element on the device. Each data point has a value, a data type, a direction, and a set of meta-data describing the value in a semantic context. Each data point also has a name and a description. The entire set of data points is organized in a hierarchy using a folder structure. Folders can be created as needed and have a folder name and description.

At the data point level, the specific technological restrictions are abstracted and hidden from the user. Working with different technologies at this level involves common work-flows for all supported technologies.

The direction of a data point is defined as the “network view” of the data flow. This means, an input data point obtains data from the network. An output data point sends data to the network. This is an important convention to remember as different technologies may define other direction semantics. If a data point can both receive and send data on the network, its direction is set to value, indicating no explicit network data flow.

The basic classes of data points are:

- **Analog**: An *analog* data point typically represents a scalar value. The associated data type is a *double precision* machine variable. Meta-data for analog data points include information such as value range, engineering units (SI and U.S.), precision, and resolution.
- **Binary**: A *binary* data point contains a Boolean value. Meta-data for binary data points includes human-readable labels for the Boolean states (i.e., active and inactive texts).
- **Multi-state**: A *multi-state* data point represents a discrete set of states. The associated data type is a signed integer machine variable. Each state is identified by an integer value, the *state ID*. State IDs need not be consecutive. Meta-data of a multi-state data point includes human-readable descriptions for the individual states (state texts) and the number of available states.
- **String**: A *string* data point contains a variable-length string. The associated data type is a character string. International character sets are encoded in UTF-8. A string data point does not include any other meta-data.
- **User**: A *user* data points contains un-interpreted, user-defined data. The data is stored as a byte array. A user data point does not include any other meta-data. This type of data point also serves as a container for otherwise structured data points and represents the entirety of the structure.

6.2.2 Timing Parameters

Apart from the meta-data, data points can be configured with a number of timing parameters. The following properties are available to input or output data points, respectively:

- **Pollcycle** (input, value): The value is given in seconds, which specifies that this data point periodically polls data from the source. This is referred to as static polling.
- **Receive Timeout** (input, value): This is a variation on the poll cycle. When receive timeout is enabled, the data point must receive a value update within the receive timeout period. If it does not receive a value, a technology may actively poll the source. If no value has been received after another period, the data point is set offline and triggers a fault alarm, if configured. Writing data from any source (network technology, connection, logic program) the receive timeout is reset.
- **Poll-on-startup** (input, value): If this flag is set, the data point polls the value from the source when the system starts up. Once the value has been read, no further polls are sent unless a poll cycle has been defined.
- **Minimum Send Time** (output): This is the minimum time that elapses between two consecutive updates. If updates are requested more often, they are postponed and the last value is eventually transmitted after the minimum send time. Use this setting to limit the update rate.
- **Maximum Send Time** (output): This is the maximum time without sending an update. If no updates are requested, the last value is transmitted again after the maximum send time. Use this setting to enable a heart-beat feature.

Dynamic polling is a feature that some network technologies offer. With static polling the pollcycle is used to permanently poll values over the network. This is required for data points that require constant value updates a fixed pollcycle (for example to trend the data). For other data points that do not need permanent value updates, so-called dynamic polling is activated, as soon as the values are needed (for example displayed on the data pont Web UI or in L-WEB). If dynamic polling is active, the data points are polled using the configured pollcycle. When the data is no longer needed, polling stops and no longer puts a burden on the network. The advantage is that a few data points can be refreshed at a higher rate at a time compared to static polling, where all data points must permanently share the available network bandwidth.

Background polling can be enabled in the project settings. With this feature enabled, all input data points, which rely on polling depending on the underlying network technology, are polled one-by-one in a round-robin fashion. This happens even if no pollcycle is set or dynamic polling is activated on those data points. The frequency of the background polling can be defined in the project settings. The default is 60 polls per minute.

6.2.3 Default Values

Default values can be defined for data points when needed. The value of a data point will be set to the defined default value, if no other value source initializes the data point. Default values are beneficial, if certain input data points are not used by the network and need a pre-defined value, e.g., for calculations. Default values are overridden by persistent values or values determined by poll-on-startup.

6.2.4 Persistency

Data point values are by default not persistent. This means that their value is lost after a power-on reset. There exist different strategies for initializing data points with an appropriate value after the device has started.

For input data points, the value can be actively polled from the network when starting up. Use the Poll-on-Startup feature for this behavior. Polling the network values has the advantage that intermediate changes on the network are reflected. An input data point can be made persistent, if the last received value shall be available after a power-on reset before a poll-on-startup completes. This can be beneficial, if the remote device is temporarily offline and the last value is considered usable.

For output data points, the value can be restored after starting up by the application. For example, if the output data point’s value is determined by an input data point and a math object, or the output data point is in a connection with an input, the input can poll its value on startup. If the output data point has no specific other value source, e.g., it is a configuration parameter set by the user, it can be made *persistent*.

To make a data point persistent, enable the Persistent property of the respective data point. The persistency option is only available for the base data point classes analog, binary, multi-state, string and user. More complex objects such as calendars, schedules, etc., have their own data persistency rules. Persistency is also available for unlinked favorites.

For structured data points, only all or none of the structure members can be made persistent. The configuration of the top-level data point, which represents the entire structure, serves as a master switch. Setting the top-level data point to be persistent enables persistency for all sub-data points. Clearing it disables persistency for all sub-data points.

6.2.5 Parameters

A data point can be qualified as a *parameter* data point. This is accomplished in the Configurator software by setting a **Parameter** check box on the data point. Those parameter data points are automatically persistent and will typically have a default value. Their purpose is to store parameterization values, which can be changed from the default value at run time and influence the behavior of the device or the logic running on the device. This way, a number of devices can have the same basic configuration and be adapted by parameter values. Examples are sunblind run times for control logic or descriptive strings for the L-WEB visualization.

The qualified parameter data points are also exported via a parameter file, which contains the entire set of current parameter values including meta-information for external tools to display parameter data in a human-readable way. The LWEB-900 parameter view can process such parameter data points and manage them for a large number of devices. For more information on how to manage parameters on your devices please refer to the LWEB-900 manual [5].

When changing parameters on the device or via the LWEB-900 parameter view, they are out of sync with their default values in the configuration. As a default it assumed that parameters are managed by LWEB-900 and the Configurator does not download and overwrite parameter values to the device.

The project settings can be changed to have the Configurator manage parameters (see Section Figure 126). In this mode the Configurator provides a mechanism to resolve value conflicts and to merge changed parameters back into the configuration. This is accomplished in the parameter merge dialog when uploading or downloading the configuration (see Figure 126). The user can select a resolution in the drop-down box. The arrow indicates in which direction the parameter values shall be copied: Copy value from device to default value, write default value to the device or NONE to leave configuration and value on device separate.

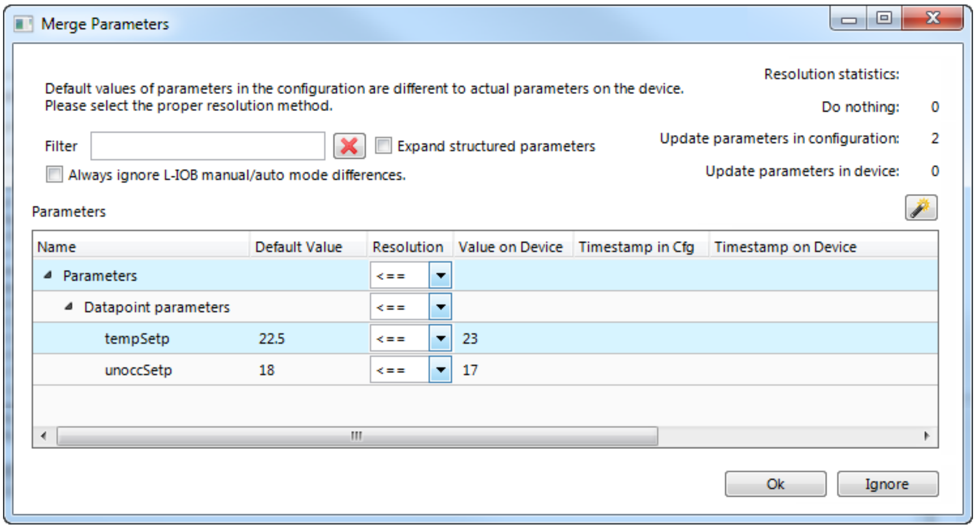


Figure 126: Parameter merge dialog.

When selecting a resolution on single parameters it affects only those parameters. When selecting a resolution on a folder it affects all data points under this folder. Click on **Ignore** to skip the parameter merge process.

L-IOB parameters are not managed by LWEB-900 and the Configurator always tries to merge L-IOB parameters that have been changed on the device. Frequent changes made to manual/auto mode can be ignored by checking **Always ignore L-IOB manual/auto mode differences**.

6.2.6 Behavior on Value Changes

The value of a data point can change, if it is written by the application or over the network. For all data points (input, output and value) the application (connection, user control, etc.) can be notified, when the value is written to. The property **Only notify on COV** defines, whether the notification is done with each write or only if the value changes (change-of-value, COV). If only notify on COV is disabled, writing the same value multiple times will result in multiple notifications.

When the value of an output data point is updated, an update is usually sent out onto the network. The property **Send-On-Delta** decides how the update is reflected on the network. If send-on-delta is inactive, each update of the value is sent, even if the value does not change. If send-on-delta is active, only value changes are sent. The send-on-delta property is only valid for output data points.

For analog data points, the COV or send-on-delta takes an extra argument, which specifies by what amount the value must change to regard it as a change for action. Both, COV and send-on-delta for analog data points check the **Analog Point COV Increment** property. A change is detected, if the value increment is bigger or equal to the specified increment. If the property is '0.0', all updates are reported, even if the value does not change.

Data point usages, such as COV trend logs or math objects may specify their own COV deltas on analog data points. These can be bigger than the data point COV itself, but never smaller.

6.2.7 Custom Scaling

Custom scaling is applied to all analog data points when they communicate values to or from the network. This feature can be used, if a network data point has engineering units not suitable for the application (e.g., grams instead of kilograms). The scaling is linear and applied in the direction from the network to the application as:

$$A = k\,N + d,$$

where N is the network value, k the *custom scaling factor*, d the *custom scaling offset*, and A the application value. When sending a value to the network, the reverse scaling is applied. If this property is enabled, the analog values are pre-scaled from the technology to the data point. The custom scaling is in addition to any technology-specific scaling factors and can be applied regardless of the network technology.

6.2.8 Protected Data Points

Some data points are created automatically depending on the model currently selected. They are protected against manipulation by the user. Therefore they cannot be deleted or moved and their properties cannot be modified. System registers (see Section 6.2.9) fall into this category. In addition some models (e.g. L-DALI) come with a predefined interface which cannot be changed either.

6.2.9 System Registers

The device provides a number of built-in system registers. They are present without a data point configuration. The system registers, such as the System time or the CPU load, can be exposed to the OPC server. By default, all system registers are checked for being exposed to OPC. To reduce the number of needed OPC tags, you may deselect certain system registers, which are not useful in a specific project.

System registers are read-only by default. System register can also serve as a testing setup for the OPC XML-DA communication without a network data point configuration. The *System Time* register is updated every second and may serve for testing subscriptions. The *Authentication Code* register can be used to verify writing to OPC tags.

The available system registers and a short description of their function are listed below:

- **State Summary**: This multi-state register contains one of the following values:
    - **OK** (1): The device is in normal state. All modules are running without problems.
    - **WARNING** (2): Some modules on the device reported a warning. The device may not function as expected.

    - **ERROR** (3): Some modules on the device reported an error. The device is not functioning as excpected.

- **System Time**: This register is an *analog* data point. It supplies the system time of the local clock in UTC as seconds since 1.1.1970. It increments each second. Example: 1302533716.
- **Time UTC**: This register is a structured data point. It supplies the system time as UTC broken down to year, month, day, hour, minutes and seconds.
- **Time Local**: This register is a structured data point. It supplies the system time as local time broken down to year, month, day, hour, minutes and seconds.
- **Unit System**: This register shows the unit system the device is currently running on. It can be either metric (SI) or U.S.
- **Unit System Set**: This register can be written to. It can request a change to another unit system. When changing it, the device needs to be rebooted to let this change become effective. This can be done via the **Command** system register or any other reboot mechanism.
- **CPU Load**: This register is an *analog* data point. It displays the average system CPU load in percent over the last minute. Example: 17 %.
- **Free Memory**: This register is an *analog* data point. It displays the current amount of free RAM memory in Bytes. Example: 20522288 Bytes.
- **Free Flash**: This register is an *analog* data point. It displays the current amount of free memory in Bytes of the Flash storage. Example: 8482688 Bytes.
- **Supply Voltage**: This register is an *analog* data point. It displays the currently measured supply voltage in volts. Example: 15.1 V.
- **System Temp**: This register is an *analog* data point. It displays the currently measured system temperature in degrees Celsius. Example: 39 °C.
- **Application Vendor, Authentication Code,** and **Authentication Result**: These registers can be used to implement an IP protection mechanism for application programs, such as IEC61131 programs. Please refer to Section 17.6.7 for more information.
- **Serial Number**: This register is a *string* data point. It displays the device’s serial number as an ASCII string. Example: “011401-000AB001D1E4”.
- **MAC Address**: This register is a *user* data point. It displays the device’s MAC address as an array of 6 hexadecimal Bytes. Example: 000AB001D1E4.
- **Firmware Version**: This register is a *string* data point. It displays the device’s firmware version as an ASCII string. Example: “4.1.0”.
- **Device IP Address**: This register is a *string* data point. It displays the device’s IP address as an ASCII string. Example: “10.101.18.204”.
- **Device IP Port**: This register is an *analog* data point. It displays device’s HTTP port as an integer value. Example: 80.
- **TZ Offset**: This register is an *analog* data point. It displays the time zone offset relative to UTC in seconds. This means a positive value for a time zone, which lies east of Greenwich. The offset includes daylight savings time. The local time can be derived by adding this register to the system time register. Example: +7200 for GMT+1 (Paris, Berlin, Vienna) including DST.
- **Device Status**: This register is a *string* data point. It contains an XML document with the device status file contents. It is not displayed on the Web UI.
- **Ethernet Link Mask**: This register is a *multistate* data point. It displays the link information of the Ethernet port. Example: “Eth 1”.

- **Hostname**: This register is a *string* data point. It displays the host name, which has been configured in the IP settings. Example: “my_linx”.
- **Position Longitude**: This register is an *analog* data point. It displays the longitude part of the device’s location in degrees. Writing to the corresponding data point **Position Longitude_Set** sets the device’s longitude in degrees. Example: -16.33472.
- **Position Latitude**: This register is an *analog* data point. It displays the latitude part of the device’s location in degrees. Writing to the corresponding data point **Position Latitude_Set** sets the device’s latitude in degrees. Example: 48.22056.
- **Position Altitude**: This register is an *analog* data point. It displays the altitude of the device’s location in meters above sea level. Writing to the corresponding data point **Position Altitude_Set** sets the device’s altitude in meters above sea level. Example: 200 m.
- **Secure Mode**: On models providing a firewall, this binary register enable the firewall to restrict access to the services provided in the **Secure Services** register.
- **Secure Services**: On models providing a firewall, this string register selects the services which should be available when **Secure Mode** is **TRUE**. This registers accepts a space-separated list of service names. If the selected services would make the device unconfigurable, a default configuration with HTTPS and SSH enabled is selected. The available service names are:
    - **HTTP**: Enables access to the configuration pages via HTTP.
    - **HTTPS**: Enables access to the configuration pages via HTTPS.
    - **SSH**: Enables access to the SSH server.
    - **OPC**: Enables access to the OPC XML-DA server.
    - **OPCUA**: Enables access to the OPC UA server.
    - **ICMP**: Allows incoming ICMP packets (recommended).
- **Command**: This multi-state register can be written to any value other than None to execute the selected command:
    - **None** (1): No command is executed. This is the default value.
    - **Warm Reboot** (2): The device immediately performs a warm reboot.
    - **Cold Reboot** (3): The device immediately performs a cold reboot.
    - **Save Parameters** (4): Pending parameter changes are committed to Flash memory.
    - **Reset Parameters** (5): All persistent values are reset to their default values. The device reboots immediately after writing this command.

6.2.10 User Registers

The device can be configured to contain user registers. In contrast to system registers, these are only available as a part of the data point configuration. User registers are data points on the device that do not have a specific technological representation on the control network. Thus, they are not accessible over a specific control network technology.

A register merely serves as a container for intermediate data (e.g., results of math objects, calculation parameters). The register can have the following, basic data types:

- **Double**: A register of base type *double* is represented by an *analog* data point. It can hold any scalar value. No specific scaling factors apply.
- **Signed Integer**: A register of base type *signed integer* is represented by a *multi-state* data point. This register can hold a set of discrete states, each identified by a signed stats ID.

- **Boolean**: A register of base type *Boolean* is represented by a *binary* data point. This register can hold a Boolean value.
- **String**: A register of base type *string* is represented by a *string* data point. This register can hold a variable-length character string in UTF-8 format.
- **Variant**: A register of base type *variant* is represented by a *user* data point. This register can hold any user-defined data of up to a specified length of Bytes. This length is defined when creating the register and cannot be changed at run time.

Since a register has no network direction, it can be written and read. Therefore, it is created as a value data point by default. It is also possible to create two data points for each register, one for writing the register (output) and one for reading the register (input). In this case a suffix is added to the register name to identify the respective data point. For example, the register *MyValue* will have two data points generated for: *MyValue_Read* and *MyValue_Write*.

6.2.11 Structures

Complex data belonging semantically together may be structured. The data point model allows mapping structure types onto user-defined data points of *variant* type. This can be necessary, if a network technology carries such structured data or if a user-defined register shall provide structured data for access through a single data point. In any case, the structure is modeled as a top-level data point and a hierarchy of sub-data points representing the structure members.

The top-level data point is a user data point of variant data type. It contains the image of the entire structure as a Byte array. Each structure field is then modeled as a sub-data point of the appropriate class (e.g. analog, binary, or multi-state). A structure field may itself be a structure going down one level in the hierarchy of sub-data points.

An example is shown in Figure 127. In this case a user register of two Bytes is bound to a structure type mapping the two bytes on analog data points. The two sub-data points *byte_0* and *byte_1*.

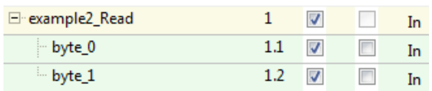


Figure 127: Example of a structured data point.

The structure types are available in a type repository with the Configurator. This repository is divided into scopes. Within each scope a type has a unique name. When selecting a type, the scope and the type name needs to be specified.

Favorites can also be structured. A structured favorite can be created by dragging a structured data point into the favorites folder. As a default, the structure top is linked to the structure top of the target data point while all sub-element are linked to their respective target sub-elements. It is also possible to unlink the structure top and link its sub-elements to different individual data points. When entirely unlinked, the structured favorite behaves like a structured user register.

6.2.12 Property Relations

A data point possesses a number of properties, which influence the behavior and appearance of the data point. Examples are data point name, poll cycle or alarm limits. Most of those properties are determined by the configuration and are static during operation of the device. Some of those properties, however, shall get a default value from the configuration and be modified during run-time. Modification may be carried out by the user by setting the property value over the Web UI, by L-WEB over the Web service or by a PLC program.

In some cases property values shall also be updated by other data points, e.g. a user register or a technology data point. In this case the data point property is linked to another data point following a given, semantic relation. This is modeled as a *property relation*. Property relations appear as data point links with the respective property names underneath their governing data point. An example is shown in Figure 128. They are marked with a link symbol . When hovering with the mouse over the link symbol, a bubble help appears describing the property relation.

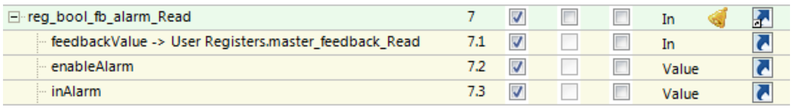


Figure 128: Example of property relations.

The property relations can be accessed like regular sub-data points from the Web UI, by L-WEB over the OPC web service or by a PLC program. For this usage, no linkage against other data points is necessary. Property relations may, however, also be linked to other data points, e.g. 'feedbackValue' in Figure 128. In this case the linked data point is used as the related property. The user may right-click on a linked property relation and choose **Go to related data point** from the context menu. For mass engineering property relation links to other data points refer to Section 7.2.8.

The following properties are available as property relations:

- **feedbackValue:** This property relation is used for feedback alarm conditions. The data point value is compared against the feedback value. An alarm is generated, if these values differ (by a certain amount). It exists only, if an alarm condition has been created.
- **enableAlarm:** This property relation is used to enable or disable alarm generation on the data point. It exists only, if an alarm condition has been created.
- **inAlarm:** This property relation is TRUE, if the data point is in an alarm. It exists only, if an alarm condition has been created.
- **ackPend:** This property relation is TRUE, if the data point's alarm needs acknowledgement. It exists only, if an alarm condition has been created.
- **highLimit:** This property relation defines the high limit for analog alarms. It exists only, if an alarm condition has been created.
- **lowLimit:** This property relation defines the low limit for analog alarms. It exists only, if an alarm condition has been created.
- **deadband:** This property relation defines the dead band for analog alarms. It exists only, if an alarm condition has been created.
- **nativeAlarm:** This property relation links to a technology data point, which is required for alarms reported to another technology. It exists only, if an alarm condition has been created and alarms are reported to the given technology (e.g. BACnet). This property relation cannot be modified by the user.
- **reportTo:** This property relation exists only in generic alarm servers. It may be linked to a technology alarm server to report alarms to that network technology.
- **totalActive, totalUnacked, totalAcked:** These property relations exist only in alarm servers. They contain counters for active unacknowledged, inactive unacknowledged, active acknowledged alarm records of the alarm server, respectively.
- **ackAll:** This property relation exists only in alarm servers. When writing TRUE, all alarms on that alarm server are acknowledged.
- **historicFilter:** This property relation exists for data points that have at least one historic filter assigned (see Section 6.5.6).

- **enable:** For Schedulers can be enabled or disabled by the use of this property relation. If the enable data point has been defined in the scheduler's configuration, this property relation is linked to that data point.
- **enableFb:** This property relation shows the enable state of the scheduler. If the enable feedback data point has been defined in the scheduler's configuration, this property relation is linked to that data point.
- **presetName:** This property relation of type string shows the preset name of the currently scheduled value. If the preset name data point has been defined in the scheduler's configuration, this property relation is linked to that data point.
- **timeToNext:** This analog property relation contains the number of minutes till the next scheduled state changes. It can be used to implement an optimum start algorithm.
- **nextState:** This property relation contains the next scheduled state. It can be used to implement an optimum start algorithm.
- **nextPresetName:** This property relation contains the preset name of the next scheduled state, if such name exists. Otherwise it stays at invalid value.

6.2.13 Convertible Engineering Units

Analog data points possess one or more unit properties that define the physical unit of the underlying scalar value, e.g., "°C". The engineering unit is displayed as a human-readable text to the user. The text can be freely entered by the user to describe the nature of the scalar value. The Configurator matches this text against its database of known engineering units. If it can identify the unit, it is denoted as a *convertible unit* with a green checkmark .

Convertible units are linked to additional meta-information in the metric (SI) or U.S. unit system. For each data point the Configurator offers unit representations in the respective unit system. Data points will provide values in that unit system. These may be different from values transported over an underlying network. For those technologies the Configurator defines a fixed *network unit*.

Important properties of convertible units when used are:

- Unit representations can be configured for the metric (SI) and U.S. unit system. The device can be configured to run in either unit system and provides data point values in the respective unit. Automatic conversion from network units to the chosen unit system is performed.

Important!	***When changing the unit system, the device needs to be rebooted and will reset all persistent values to their default values converted to the chosen unit system.***

- Automatic unit conversion in local connections is performed, if data points with compatible convertible units are connected (e.g. '1000 W' are converted to '1 kW'). No custom scaling is required.
- Auto-generated data points in connections are created such that they have a best-matching unit in their target technology (e.g., the best-matching SNVT is created out of a BACnet data point of a certain engineering unit).

For example, a data point has a fixed network unit in °C. As it is a convertible unit, the user can define a representation for that data point in the metric (SI) system (°C) and one in the U.S. system (°F). Depending on the selected unit system, the received value on the network is converted either to °C or °F. All data point values on the device are processed in the selected unit system, including the Web UI, OPC server, parameter file, global connections, and programmable logic. For more information on how to start a project in SI or U.S. units, please refer to Section 7.6.7.

6.3 Math Objects

6.3.1 General Properties

Math objects are advanced application objects that can execute mathematical operations on data points. A math object takes a number of input data points (variables v_1, v_2, …, v_n) and calculates a result value according to a specified formula. The result is written to a set of output data points. The formula is calculated each time one of the input data points updated its value. The formula is only evaluated if all of the input data points have a valid value (i.e., don't show the *invalid value* status).

Using the assigned variable names, immediate values, parenthesis, operators and function names, the user can enter a formula in the usual way (infix notation). Apart from the functions in the next Section, the short-hand operators `+`, `-`, `/`, `*`, `%`, `AND`, `OR`, `XOR`, `^`, `&`, `|`, `=`, `!=`, `<`, `>`, `<=`, `>=` can be used directly. Further, it is possible to use parenthesis to define the precedence of the operations.

Example: `(v1 + v2) * sqrt(pow(v3,0.1))`

Note: *As usual practice in programming languages, the comma is used to separate arguments in expressions and the **decimal point** is used in decimal values. The expression `sum(4,5)*2` evaluates to 18, while `sum(4.5)*2` evaluates to 9.*

As you enter the formula, it will be parsed and the resulting sequence of calculations will be displayed in a list at the right of the property page. This list shows your formula in reverse polish notation (RPN), also known as postfix notation, as used by many scientific pocket calculators.

6.3.2 Usage Hints

A few functions end with a … (three dots) in the argument list. This means that they accept a variable number of arguments. When used in the formula, they will fetch all available values from the stack and then calculate the result, which will be put back on the stack and be the only value on the stack, since all other values were used as input to the function.

This behavior causes some limits in how these functions may be used. You are on the safe side, if you use such a function only as the outermost function (infix), or as the last function on the stack (postfix) for example:

```
sum(v1, v2, exp(v3, -1))
```

Or the postfix equivalent: `v1, v2, v3, -1, exp, sum`

If you have to use it as an argument to another function, it may only be the first argument; otherwise the formula cannot be processed by the math object, which internally uses an RPN machine, with precompiled instructions for optimal performance. Example:

`add(avg(v1, v2, v3), 5)` or `avg(v1, v2, v3)+5` will work.

`add(5, avg(v1, v2, v3))` or `5 + avg(v1, v2, v3)` will NOT work.

Another property of those functions is that they ignore input values, which have the invalid value. Therefore, assuming v1=5, v2=invalid, v3=3 the calculation `add(v1,v2,v3)` evaluates to 8 while `v1+v2+v3` returns invalid. This can be used to purposely allow inputs in the calculation that have no value.

To limit the number of re-calculations, the data point option **Only Notify on COV** should normally be checked on all connected input data points. This avoids recalculating the formula and writing a value to the output data point when it is already clear that the result will be the same, because the input value did not change. The same option can also be checked for the output data point to avoid unnecessary writes to the output data point, in case the inputs changed but the result of the formula is still the same.

6.3.3 Function List

The currently supported math function calls are listed in Table 15.

Function	**Return Value**
add(v1,v2)	v1 + v2
sub(v1,v2)	v1 - v2
mul(v1,v2)	v1 * v2
div(v,d)	v / d
mod(v,m)	Returns the remainder of dividing v by m, where v and m should be integer values. Fractional values will be rounded to the nearest integer automatically
max(v1,...)	Returns the maximum of all values on the value stack
min(v1,...)	Returns the minimum of all values on the value stack
avg(v1,...)	Returns the arithmetic mean value of all values on the stack
log(v)	Returns the natural logarithm of v
log2(v)	Returns the base 2 logarithm of v
log10(v)	Returns the base 10 logarithm of v
exp(v)	Returns the value of e (the base of natural logarithms) raised to the power of v
exp2(v)	Returns the value of 2 raised to the power of v
exp10(v)	Returns the value of 10 raised to the power of v
sqrt(v)	Returns the non-negative square root of v
pow(v,exp)	Returns the value of v raised to the power of exp
round(v)	Round v to the nearest integer
floor(v)	Round v down to the nearest integer
ceil(v)	Round v up to the nearest integer
sum(v1,...)	Returns the sum of all values on the stack
and(b1,b2)	logical AND of the Boolean values b1 and b2 (b1&&b2)
or(b1,b2)	logical OR of the Boolean values b1 and b2 (b1||b2)
xor(b1,b2)	logical exclusive OR of the values b1 and b2 (b1^b2)
not(b)	logical inverse of the Boolean value b (!b)
lt(v1,v2)	returns 1 if v1 is lower than v2, else returns 0 (v1 < v2)
le(v1,v2)	returns 1 if v1 is lower or equal v2, else 0 (v1 <= v2)
eq(v1,v2)	returns 1 if v1 equals v2, else 0 (v1 = v2)
ge(v1,v2)	returns 1 if v1 is greater or equal v2, else 0 (v1 >= v2)
gt(v1,v2)	returns 1 if v1 is greater than v2, else 0 (v1 > v2)
if(b,vt,vf)	returns vt if b is true, else returns vf (b ? vt : vf)
encode(b1,...)	Reads all values from the stack, converts them to Boolean values and encodes them into an integer value, where the first value is used as the LSB and the last value as the MSB.
sin(v1)	Returns the sine of v1, where v1 is given in radians
cos(v1)	Returns the cosine of v1, where v1 is given in radians
tan(v1)	Returns the tangent of v1, where v1 is given in radians
sinh(v1)	Returns the hyperbolic sine of v1, which is defined mathematically as (exp(v1) - exp(-v1)) / 2
cosh(v1)	Returns the hyperbolic cosine of v1, which is defined mathematically as (exp(v1) + exp(-v1)) / 2
tanh(v1)	Returns the hyperbolic tangent of v1, which is defined mathematically as sinh(v1) / cosh(v1)
asin(v1)	Returns the arc sine of v1; that is the value whose sine is v1
acos(v1)	Returns the arc cosine of v1; that is the value (in radians) whose cosine is v1
atan(v1)	Returns the arc tangent of v1; that is the value (in radians) whose tangent is v1
asinh(v1)	Returns the inverse hyperbolic sine of v1; that is the value whose hyperbolic sine is v1
acosh(v1)	Returns the inverse hyperbolic cosine of v1; that is the value whose hyperbolic cosine is v1
atanh(v1)	Returns the inverse hyperbolic tangent of v1; that is the value whose hyperbolic tangent is v1
gamma(v1)	Returns the value of the Gamma function for the argument v1. The Gamma function is defined

Function	Return Value
	by Gamma(x) = integral from 0 to infinity of t^(x-1) e^-t dt. It is defined for every real number except for no positive integers. For nonnegative integral m one has Gamma(m+1) = m! and, more generally, for all x: Gamma(x+1) = x * Gamma(x) For x < 0.5 one can use Gamma(x) * Gamma(1-x) = PI/sin(PI*x)
abs(v1)	computes the absolute value of the argument v1

Table 15: Available math functions.

6.4 Connections

6.4.1 Local Connections

With the use of connections data points can interact with each other. Connections specify which data points exchange values with each other. Various types of connections – from “1:*n*” to “*m*:*n”* connections – are supported. Data points added to a connection specify whether they feed a value into the connection (send) or they receive a value from the connection (receive).

This means, the following connections are possible:

- 1 input data point is connected and writes to *n* output data points,
- *m* input data points are connected and write to 1 output data point,
- *m* input data points are connected and write to *n* output data point.

The most common connection will be the 1:1 connection. This is the type of connection that is auto-generated by the Configurator software. Other types must be created manually or by a template in the Configurator.

In the 1:*n* connection the input value is distributed to all *n* output data points. In the *m*:1 connection, the most current input value is written to the output data point. When polling the output data point in poll-through mode (maximum cache age is set on the output), the value from the first input data point is polled. The same holds true for a *m*:*n* connection. The default data flow of data points in a connection is a result of the data point direction. This can be overridden by a custom setting (i.e. an output data point can be configured as an input to the connection).

Connections can connect data points of different technologies with each other (also mixed among the target data points). When connecting data points of different classes the exchanged values need to be converted. The connection inherits the type of the first data point class. If data points of a different class are added to this connection, an *adaptor* needs to be defined. For example an analog value connection has a multi-state output data point. Adaptors can be saved in a library and re-used later for similar conversions.

The following conversions apply:

- **Analog to Analog**: The value range is capped on the output data points. This means, if the input value in the hub does not fit into the range of an output data point, the value is capped to the biggest or smallest allowed value. If the input and output data points both have convertible units the value is converted. The user can also specify a simple math formula as an adaptor. In this case no implicit unit conversion is performed.
- **Binary/Multi-state to Analog**: The Boolean or state value is directly converted to an analog value (e.g. state ID ‘4’ is written as ‘4.0’) as a default. The user should specify an adaptor to map the Boolean or state value to designated analog values.

- **Analog to Binary/Multi-state**: As a default the analog value is converted to the next Boolean or state value (e.g. '1.2' is written as state ID '2'). The user should specify an adaptor with its own translation of value ranges to state values.
- **Multi-state to Multi-state**: Multi-state data points that have different state maps lead to a conversion of their state values. The state maps of inputs and outputs are ordered by state ID in ascending order. The state value of the input is then ranked as the n-th state and propagated over the connection. For example, the input state ID '1' is the 2nd state and the output's 2nd state has the state ID '0'. If the output data point has less states than the input, the output state is limited to its highest state ID. The user should specify an adaptor that defines which input state maps to which output state.
- **Binary to Binary**: Binary data points can be connected without conversion.
- **String to String**: String data points can only be connected to string data points.
- **User to User**: User data points can only be connected to user data points. If the length is different, only valid bytes are written or excess bytes are truncated, respectively.
- **SNVT_switch to Analog/Binary/Multi-state**: The user data point of a SNVT_switch can be connected to analog, binary, and multi-state data points.
- **Analog/Binary/multi-state to SNVT_switch**: Analog, binary, and multi-state data points can be directly connected to a SNVT_switch user data point.

6.4.2 Multi-Slot Connections

Connections between structured data points often need to connect each structure member separately. To increase the overview in the project on the involved, single connections, a *multi-slot connection* can be created for local connections. This is a connection with several slots for transporting separated values over the connection. Each slot has a number and a name and can connect two or more data points. Data points added to other slots do not share their values across slots. One can think of such a connection as a cable with many wires. An example is shown in Figure 129 (a). The data point 'IN A' sends its value to 'OUT X' but not 'OUT Y'.

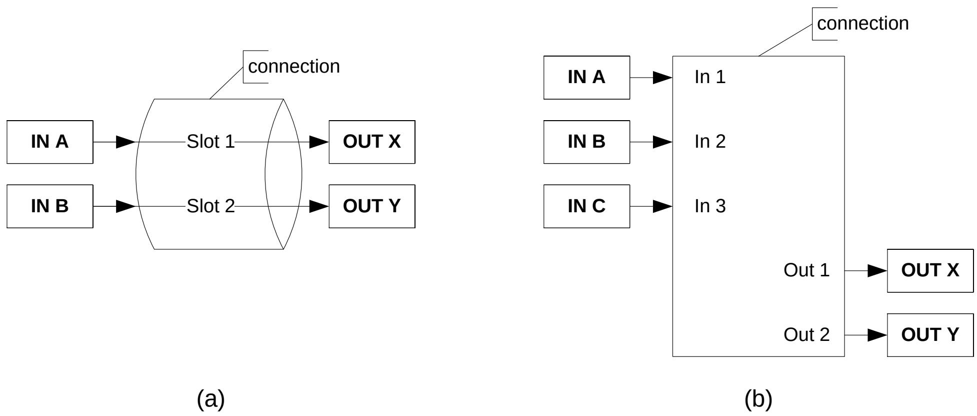


Figure 129: Multi-slot connection (a) and multi-slot with math block adaptor (b).

Some gateway applications also require a functional mapping between different data point structures in one connection. A multi-slot connection can be used with a math block adaptor to accomplish this task. A math block has *n* inputs and *m* outputs. The multi-slot connection has a slot for each input and output, which can be connected to the respective data points as depicted in Figure 129 (b). For this multi-slot connection the math block adaptor defines a fixed layout of the slots; no more slots can be added to this connection.

In a math block adaptor with *n* inputs $v_1, v_2, \dots, v_n$ each output o_i is calculated as a formula depending on all inputs $o_i = f_i(v_1, v_2, \dots, v_n)$. Each output has two math formulae following the same format as used in math objects (see Section 6.3):

- Output value formula: This formula calculates the output value as a function of all input values.
- Output enable formula: This formula calculates an output enable (result > 0 is enable) for the output. If the output is enabled, the output value will be written to the output. If the output is disabled, the calculated output value is not written to the output.

In addition, each input slot can be configured whether it shall trigger the calculation or not. Normally, any change in any input triggers the calculation of all outputs.

6.4.3 Automatic Generation and Templates

In a gateway application the systems engineer has a typical workflow: He will be confronted with some network equipment of one technology that needs to be exposed to another network technology. The task of generating the counterparts of data points in another technology and connecting them is covered by the *smart auto-generate and connect* method. The existing data points are called *sources* and the generated data points are called *targets*.

In principle, the Configurator supports auto-generate for all source technologies but generation is limited to select target technologies. Depending on availability on the device model, the following technologies can be target for auto-generation:

- CEA-709 (static NVs),
- BACnet (server objects),
- Registers,
- Modbus (slave registers).

The target data point is generated with opposite direction and of the same class as the source data point. Depending on the target technology, however, certain restrictions apply on what can be generated. Typical issues are engineering units, state maps and data point structures. The folder structure of the source data points is replicated for the target data points.

For example, when generating matching counter parts to NVs, there are two types of NVs to be considered: Simple NVs that hold only one value (scalar or enumeration), and structured NVs, that consist of a number of fields. For simple NVs only one BACnet object per NV is generated. For structured NVs, one BACnet object is generated for each structure member. This method is called structure flattening. Some target technologies do support structures and no flattening is applied. When generating an analog target, a data point with the best-matching engineering unit is created. If the target allows arbitrary engineering units this will be the same as the source engineering unit. If the target has only a limited number of engineering units, the technology object with the best-matching unit is created. Multi-state target data points are created with an equal number of states and compatible state IDs. For example the CEA-709 state IDs are sorted and renumbered to start at '1' in BACnet (i.e., a '-1' of MOTOR_NUL in CEA-709 maps to a '1' of MOTOR_NUL in BACnet). This is necessary as the SNVT states '-1' and '0' cannot be represented in BACnet as a raw value, because allowed BACnet multi-states start at 1.

The Configurator provides a preview dialog that shows, which target data points will be created. Thus, the implicit generation rules are visible to the user. If the target technology provides several options on what to generate, the user can change the default in this dialog.

The setting is stored in the project and will be applied again with the next generation. The project settings also provide defaults for auto-generation. How exactly data points are created depends on the target technology. Refer to the technology sections for more information how data points are used in connections.

For more advanced connection tasks that involve specific adaptors *auto-generate templates* must be used. An auto-generate template contains the source data point, the desired target data point and the local connection with all appropriate adaptors. There are two types of auto-generate templates:

- **Simple auto-generate template**. This template contains exactly one source data point (scalar or structured). It may contain one or more target data points, which will be generated. This template can be applied on any selection of single source data points. If the type of the source data point matches the one in the template, this auto-generate template can be selected to generate the target data points. This template type can be used to generate special target objects for certain scalar source data points using adaptors. It can also be used to connect structure elements of the source to structure elements of a target using a math block adaptor.

- **Complex auto-generate template**. This template contains more than one source data points. This type must be used, if two or more sources shall generate the targets in a specific way. Since no single source data points can be matched in this case, the source data points which belong together must be grouped under a folder. Math block adaptors can be used with complex auto-generate templates.

Auto-generate templates can use configurable placeholders for data point name, data point description, server object name, server object description. These placeholders are evaluated when the template is applied and new data point instances are created. The available placeholders are listed in Table 16.

Placeholder	Meaning
`%{name}`	In simple auto-generate templates this expands to the source data point name.
`%{descr}`	In simple auto-generate templates this expands to the source data point description.
`%{native_name}`	In simple auto-generate templates this expands to the native name (e.g. register name, NV programmatic name, server object name) of the source data point. If no such native name exists, the data point name is used instead.
`%{native_descr}`	In simple auto-generate templates this expands to the native object description (e.g. server object description) of the source data point. If no such native description exists, the data point description is used instead.
`%{path}`	This placeholder expands to the source data point/folder path. The path extends up until the respective data point folder root folder. Example: The source data point is located in 'CEA-709 Port.Datapoints.Floor1.Room202'. The path expands to 'Floor1.Room202'.
`%{folder_descr}`	This placeholder expands to the folder description of data point's parent folder. Folder descriptions are copied from source to generated folders.

Table 16: Placeholders in auto-generate templates.

6.4.4 Global Connections

Global connections provide the same notions as local connections but extend beyond the scope of one device. A global connection establishes a data cloud with a system-wide name. Data points added to a global connection can send data into that connection or receive data from the connection. The data is transferred over an IP-based network. All data is automatically matched by the global connection name. This makes global connections especially useful to provide certain global data in a system, without knowing who will be reading that data. Examples are weather station data, wind alarms or global on/off.

Global connections cannot use adaptors for conversions as in local connections. If conversions are needed, an intermediate register data point must be used to receive/send data from/to the global connection. The adaptor needs to be installed with a local connection between the register and the data point, which requires the conversion.

The system in which the data cloud of a global connection is established is defined by an IP-852 channel. This channel is not related to the CEA-709 technology; it is purely used to define the set of devices exchanging data through global connections. It can, however, coexist with an IP-852 channel for CEA-709. The configuration of the IP-852 channel is done by adding devices to a configuration server. This is described in closer detail in Section 8.3.

A global connection has the following properties:

- **Max Send Time**: This timing parameter of the global connection specifies a time in seconds, in which a value update is transmitted into the connection, even if not value has changed. This is typically used for heartbeat functions.
- **Min Send Time**: This timing parameter of the global connection specifies a time in seconds, for which transmissions will be delayed after sending out a value into the connection. This setting can be used to limit the transmission rate to the connection.

The following properties are derived from the data points in a global connection:

- **Receive Timeout**: A data point with a receive timeout will be put into the state offline, if it does not receive a value within the specified period of time (see Section 6.2.2). This also applies to values received from the global connection.
- **Poll on startup**: If a data point in the global connection has the poll on startup feature enabled (see Section 6.2.2), an initial value update will be triggered for the global connection.

How a global connection is created and configured in the Configurator software is described in Section 7.10.7. Note, that the number of configurable global connections on a device is limited (see Section 24.2).

6.4.5 Forward Delay

Connections can be used to implement stagger delay with randomization. This is beneficial for applications with load shedding. All receive items in a connection, both local and global, can be configured with a *delay*. The delay can be specified as a constand delay in seconds, or as an interval, in which the actual delay will be randomized.

These are example delay settings:

- 0s: No delay is imposed on received updates.
- 10s: Each received update on the receive data point will be delayed for 10 seconds.
- 10-60s: Each received update is delayed randomly between 10s and 60s.

If updates are generated faster than they are forwarded because of the delay, the last update overwrites any pending updates. No queueing is implemented for delayed updates over connections.

6.5 AST Features

6.5.1 Alarming

The alarming architecture comprises a number of entities. Objects that monitor values of data points and generate alarms depending on an *alarm condition* are called *alarm sources*. The alarms are reported to an *alarm server* on the same device. The alarm server maintains a list of alarm records, called the *alarm summary*. The alarm server is the interface to access the local alarms.

Generic alarm servers provide the maximum set of alarming features and can be accessed over L-WEB (via the Web service) or the Web UI. Data points of all network technologies can be alarmed through generic alarm servers. Technology alarm servers can be used to expose access to the alarms to network technologies that support it. Generic alarm servers can be configured to report their generic alarms to technology alarm servers. For example, a generic alarm server may report its alarms to both CEA-709 and BACnet alarm servers.

An alarm record contains the information about a specific alarm. This includes information about the alarm time, the source of the alarm (i.e., which data point caused the alarm), an alarm message, an alarm value, an alarm type, an alarm priority, and an alarm state. An alarm record undergoes a number of state changes during its life-cycle. When the alarm occurs, it is *active*. At this point the alarm time, alarm message, alarm value is notified using the alarm priority. When the alarm condition subsides, the alarm becomes *inactive*. At this point the clear time and the clear message is notified using the normal priority. The priority levels are configurable on the alarm server, where 0 is the highest and 255 is the lowest priority.

Alarm transitions (to an alarm state, to the normal state) can be acknowledged by an operator. Which of those transitions requires an acknowledgement is configurable on the alarm server. If an active alarm is acknowledged it becomes *active acknowledged*. Active alarms can also become inactive, but an acknowledgement is still required. Then they become *ack-pending*. When an alarm is inactive and was acknowledged it finally disappears from the alarm summary.

An alarm state can be of different alarm types. The alarm type specifies the class of the alarm. The following alarm types exist:

- **Off-Normal Alarm**: This alarm type is a generic alarm class that applies to binary and multi-state alarm conditions. It indicates that the alarmed data point is on an off-normal operating condition that triggered the alarm. An alarm value is supplied. In technology alarm servers, restrictions may apply.
- **High/Low Limit Alarm**: This alarm type is typical for analog alarm conditions. It applies when the alarmed value is over or under the defined alarm limits. An alarm value is supplied. In technology alarm servers, restrictions may apply.
- **Fault Alarm**: This alarm type is indicating that the monitored data point is in a fault state. This is different from off-normal or high/low limit alarms. The value of the data point is within the specifications of the alarm condition but the data point itself is considered faulty. This can stem from an unreliable value or an offline value, i.e., if the data point is offline. No alarm value is supplied.

Alarms may be generated from a given data point value (alarm value or value range) or by comparing a data point command value with a feedback value (feedback alarm). When defining a feedback alarm, the alarmed data point represents the command value and has a ‘feedbackValue’ property relation (see Section 6.2.12). This property relation can be linked to another data point, which effectively provides the feedback value.

Alarmed data points also possess other property relations. The ‘enableAlarm’ property relation can be used to disable or enable alarm conditions when linked to a data point. The property relations ‘highLimit’, ‘lowLimit’, ‘deadband’ can be used to modify analog alarm conditions. The property relations ‘inAlarm’ and ‘ackPend’ are TRUE if a data point is in an alarm state or needs acknowledgement, respectively.

When a data point is alarmed by a generic alarm server, which reports to a technology that requires a dedicated technology data point (e.g., an alarm for a user register is reported to BACnet), the required data point is automatically created and linked via the ‘nativeAlarm’ property relation.

Alarm server objects possess property relations that provide a counter value of active unacknowledged, active acknowledged, and inactive unacknowledged alarms. These property relations may be linked to other data points that can be used to process this information.

Other devices can access the alarm information through a technology alarm server or the Web service. These devices are *alarm clients*. They register with the alarm server and get notified about changes to the alarm summary. Alarm clients can be used to display the current alarm summary and to acknowledge alarm transitions. Depending on the underlying technology, some restrictions may apply to the available alarm information and acknowledgement behavior. Refer to the technology sections for more information.

6.5.2 Historical Alarm Log

The alarm summary of the alarm objects contains a live list of currently active and acknowledge-pending alarms. As soon as an alarm becomes inactive and has been acknowledged, it disappears from the alarm summary. To store a historical log of alarm transitions an *alarm log* is utilized. An alarm log can log transitions of one or more alarm objects.

The alarm log is always local and stored as a file on the device. The size of an alarm log is configurable. The alarm log operates as a ring buffer. As soon as its size limit is reached, the oldest alarm log records are overwritten by newer alarm transitions. The alarm log is available on the Web UI or can be uploaded from the device as a CSV file. The CSV file can also be used as an e-mail attachment.

6.5.3 Scheduling

Schedulers are objects that schedule values of data points on a timely basis. A scheduler object is configured by which data points it shall schedule. This configuration is done by the system engineer once, when the system is designed. The configuration of the times and values that shall be scheduled is not part of that initial configuration and may be changed later. This distinction has to be kept in mind.

A scheduler object sets its data points to predefined values at specified times. The function of the scheduler is state-based. This means, that after a given time, the scheduler maintains this state. It can re-transmit the scheduled values as appropriate (e.g., when rebooting). The predefined values are called *value presets*. A value preset contains one or more values under a single label (e.g., “occupied” schedules the values { 20.0, TRUE, 400 }). Each preset can also be configured with a display color.

Which value preset is scheduled at what time is defined by a *scheduled event*. The event defines the starting time, value preset and end time in a 24-hour period. Events can be one-time events or recurring events. A schedule typically consists of a number of recurring and one-time events, for instance one event for the weekdays Monday through Sunday. See Figure 130 for an example of a schedule in a given calendar week.

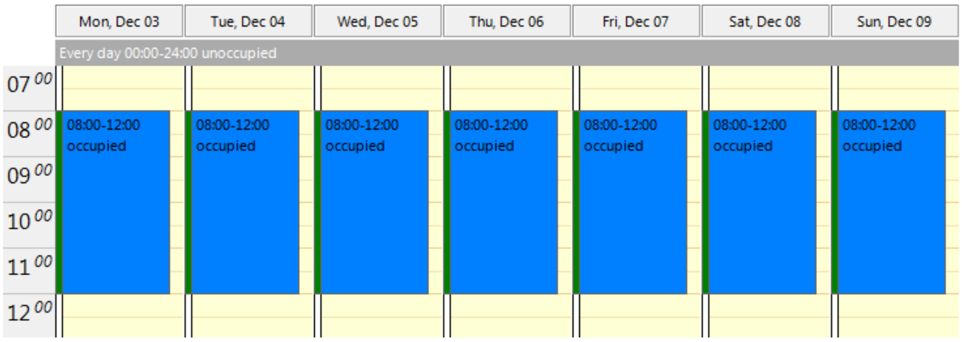


Figure 130: Example of a recurring event in a schedule.

How scheduled events are recurring can be defined by choosing the appropriate event type:

- **One-time**: This event occurs exactly on one defined date.
- **Daily**: This event occurs every day, starting at a given date and ending at a given date.
- **Weekly**: This event occurs every week on the specified weekday.
- **Monthly**: This event occurs every month following a date range or a defined rule (e.g. every last Friday).
- **Yearly**: This event occurs every year following a date range or a specific day every year.
- **Default**: This is a special event. The selected preset value will be in effect 00:00 to 24:00 hours every day if no other event occurs.
- **Calendar**: For some tasks the regular recurrence such as on weekdays is not sufficient. This can be implemented by defining events based on a *calendar*. For instance, there may be a calendar for holidays. The calendar contains a number of *calendar patterns*. Each calendar pattern describes a pattern of dates on which an event shall occur, e.g., *Holidays*.

One can define a set of scheduled events that are recurring differently. For example one event is defined for regular workdays (Monday through Friday). Another event is defined based on the holidays calendar pattern. This will lead to overlapping events between workday and holiday for those weekdays, which are holidays.

The resolution of this overlap is simple: Each event is configured with a *priority*. Should an overlap occur, the event with the higher priority will be in effect (e.g., Dec 25th in Holidays overrides the regular workday event). An example is shown in Figure 131. The detailed view shows the two overlapping events and the preview shows the effective schedule. Note, if two events with the same priority exist, it is not defined, which one is in effect. Therefore, always use distinct priorities.

Priorities are numbers, but some priorities have been pre-assigned, e.g. highest, override, normal, low. Please also refer to the technology-specific limitations described in Section 7.12 to learn about special behavior of the respective networking technology.

If no event is in effect at a given time, the *schedule default* becomes effective. This can be defined to be any of the defined presets. In Figure 131 the scheduler will write out “unoccupied” after 12:30 as no other event exists. A special schedule default is *silent*. With the silent default the scheduler will be inactive, if no event is in effect. This means it will not update its scheduled data points until the next scheduled event, not even at midnight. Thus, using the silent default one can build an event-based scheduler.

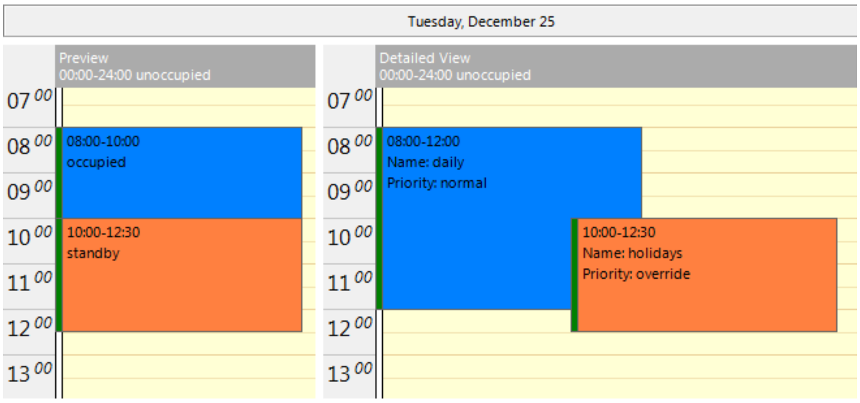


Figure 131: Example with overlapping events and different priorities.

The configuration of calendar-based recurrence is done by calendar patterns in the calendar. Each calendar pattern contains a number of pattern entries. These entries can define the following:

- A single date: This defines a single date. Wildcards may be used in the year to specify Dec 25th of every year.
- A date range: This defines a range. Starting with a start date and ending with the end date. No wildcards should be used.
- A Week-and-Day definition: This defines dates based on a week, such as every 1st Friday in a month, every Monday, every last Wednesday of a month.

A schedule defines at which time instants certain states of the scheduled data points are maintained. The *next-state* feature allows looking up to 48 hours ahead into the future and predicts when the next scheduled state change will occur. There are two data points involved: the timeToNext is a counter in minutes to the next scheduled event, and the nextState data point is the state of the next scheduled event. This information can be used by controllers for optimum start algorithms (e.g., pre-heat a room for the scheduled occupancy state). Use the SNVT_tod_event in CEA-709 to accomplish this task. With generic schedulers and BACnet schedulers use the scheduler's property relations timeToNext and nextState (see Figure 132).

Datapoint Name	No.	Direction	OPC	Use	ID
calendar	1	Value	☑	1	1039
◢ Schedule_reg1	2	Value	☑	0	1060
enable -> User Registers.sched_enable	2.1	Value	☑	0	106F
enableFb	2.2	In	☑	0	1070
nextPresetName	2.3	In	☑	0	1074
nextState	2.4	In	☑	0	1073
presetName	2.5	In	☑	0	1071
timeToNext	2.6	In	☑	0	1072
User Registers.reg1	2.7				105F

Figure 132: Property relations of a scheduler object.

When a scheduler is executing the schedule on the local device, it is called a *local scheduler*. Such a scheduler is configured to schedule data points and later its daily schedules can be modified. When accessing the daily schedules of a scheduler, which executes on a remote device, the object is called a *remote scheduler*. A remote scheduler has the same interface to the user to modify daily schedules. A remote scheduler object can be used as a user-interface for schedulers that execute on different devices.

6.5.4 Trending

Trending refers to the ability to log historical values of data points over time. A trend log object is responsible for this task. The generic trend log object provides the maximum set of features and can be accessed by L-WEB and the Web service. It can be configured to record historical data of any data point on the device. Log records are generated either in fixed time intervals, on change-of-value (COV) conditions, or when a trigger is activated. The fixed intervals can be optionally aligned to the wall time (e.g., to the top of the hour). After a reboot the recording is resumed at the aligned intervals. Trend log objects can trend either local or remote data points. Technology trend log objects can be used to record historical values of the respective technology data points and expose them to network technologies that support it. These historic logs are separate from the generic trend logs and certain restrictions of the technology may apply.

The trend data is stored in a binary format on the device. The capacity of a given trend log is configured. The trend log can be operated in one of two modes: In *linear mode* the trend file fills up until it reaches its capacity. It then stops logging. In *ring buffer* mode the oldest log records are overwritten when the capacity is reached.

Devices with SD cards also allow backups of the trend logs on external Flash storage. This backup can be triggered by the user over the LCD display or be triggered by certain actions. The trend data is stored in CSV format under a folder identifying the device by serial number and the trends sub-directory, e.g. '016101-8000000DEA51/trends'. The SD card can be used on different devices. In this case different device directories will be created. The trend backup files can be opened directly on a PC. The backup on external storage can be enabled individually per trend log.

A fill-level action can be activated, whenever the trend log has logged a percentage of its log size with new log records. A fill-level condition of 70% on a trend log with 1000 items capacity will activate the fill-level trigger every 700 logged records. This trigger can be used to send E-Mails or backup trend data on external storage if available.

Trended data points can be logged as their actual values at given time instants or as an aggregated value over the defined log interval. Aggregation can be calculated as minimum, maximum, or average. Aggregation can be beneficial, if the trended value changes more frequently than the selected log interval. Using aggregation, the log interval can be chosen to limit the amount of logged data while preserving information of the trended value.

For technology trend log objects, certain restrictions apply as to how many data points can be trended in one trend log and which trend modes are available. Refer to the technology sections for more information.

6.5.5 E-mail

The e-mail function can be combined with the other AST features. The format of an e-mail is defined through *e-mail templates*. An e-mail template defines the recipients, the e-mail text, value parameters inserted into the text and triggers, which invoke the transmission of an e-mail. An e-mail template can also specify one or more files to be sent along as an attachment.

The e-mail text content can contain text and configurable placeholders. The placeholders expand to their content when the e-mail is transmitted. Placeholders can also be used in other text fields, such the Subject field. The placeholders available for e-mail templates are listed in Table 17.

Placeholder	Meaning
%{v*n*}	This placeholder expands to the content of a data point variable defined in the e-mail template. The n refers to the n-th data point variable in the list. The data point variable list specifies this index.
%{v*n*.src_name}	If the data point variable is an alarm, this placeholder expands to the source name of the alarmed data point, for which a new alarm is reported.
%{v*n*. al_descr}	If the data point variable is an alarm, this placeholder expands to the message of the reported alarm. For a to-alarm transition it contains the alarm message, for a to-normal transition it contains the clear message, for a to-fault transition it contains the fault message.
%{v*n*. al_type}	If the data point variable is an alarm, this placeholder expands to the alarm type of the reported alarm.
%{v*n*. al_state}	If the data point variable is an alarm, this placeholder expands to the state of the reported alarm.
%{v*n*. al_tm}	If the data point variable is an alarm, this placeholder expands to the alarm time of the reported alarm.
%{v*n*. cl_tm}	If the data point variable is an alarm, this placeholder expands to the clear time of the reported alarm.
%{v*n*. ack_tm}	If the data point variable is an alarm, this placeholder expands to the acknowledge time of the reported alarm.
%{v*n*. ack_src}	If the data point variable is an alarm, this placeholder expands to the acknowledge source text of the reported alarm. If the alarm has not been acknowledge, this is empty.
%{v*n*. al_val}	If the data point variable is an alarm, this placeholder expands to the value which triggered the alarm (alarm value).
%{mailid}	This placeholder expands to the mail ID used for the transmitted message. This mail ID is different for each message.
%{timestamp}	This placeholder expands to the mail timestamp seen in the transmitted message.

Table 17: Placeholders in e-mail templates.

A prerequisite to sending e-mails is the configuration of an e-mail account on the device. This can be done on the Web UI (see Section 5.2.24). It is recommended to use the e-mail server of your Internet provider. For public mailers, enable the required authentication. SSL/TLS e-mail authentication is supported for using Hotmail, gmail or Yahoo!.

The amount of generated e-mails can be limited using a rate limit algorithm. The transmission of e-mails can be disabled altogether by using a special data point. That data point can be scheduled or driven over the network.

If an e-mail cannot be sent (e.g. the mail server is not reachable), the mail delivery is retried up to 24 times every 30 minutes.

6.5.6 Historic Filters

For certain applications historic values of a given base data point, both recent and far into the past, can be of interest. This can be accomplished with *historic filters*. Historic filters allow processing historic values of the base data point according to a filter function. One or more such functions can be defined per base data point. The result of the historic filter is written to *historicFilter* property relations. For each historic filter function a time period can be defined at which the base value is sampled, e.g., every first of the month at midnight, and how many samples ago. Historic filters can be created for any analog, binary, or multi-state data point. It is not necessary to create a trend log.

The following sampling periods can be defined:

- Value every *x* minutes aligned to full hour (*x* = 1, 2, 5, 10, 15, 20, 30 min), 0 or 1 samples ago,

- Hourly value at full hour, 0..24 samples ago,
- Daily value at HH:MM:SS of the day, 0..60 samples ago,
- Weekly value at HH:MM:SS on weekday (Mon..Sun), 0..10 samples ago,
- Monthly value at HH:MM:SS on day of month (1..31, last), 0..24 samples ago,
- Yearly value at HH:MM:SS on DD/MM of the year, 0..5 samples ago.

By using historic filter data points it is possible to implement numerous calculations on historic values of the base data point. For example it is possible to create two filter data points with a daily sampling period recording the energy consumption at midnight, one holding the most current sample (today at midnight) and the other the previous sample (yesterday day at midnight). This is shown in Figure 133.

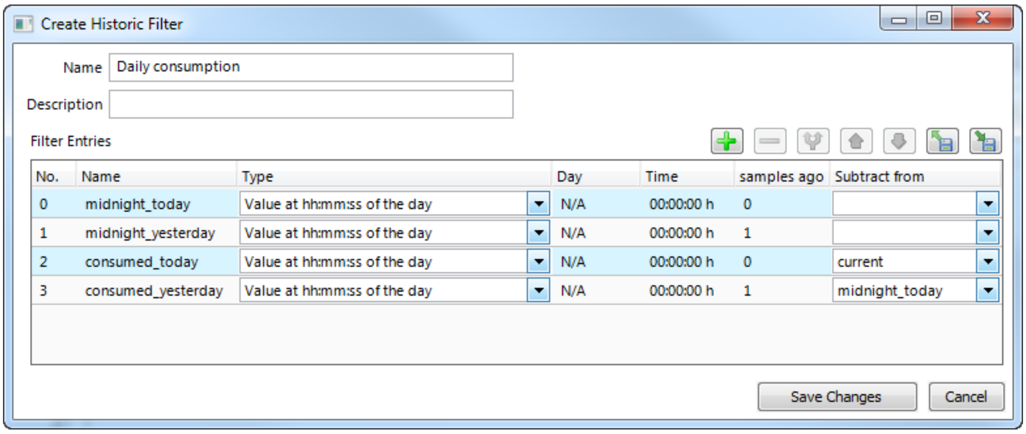


Figure 133: Example historic filters for daily consumption.

For calculating the difference between the current value and any historic value, the filter definition can be configured in a delta mode. This is a shortcut to creating a math object subtracting the historic filter data point value from the current value of the underlying data point or the value of another filter item. The example shows two results: Filter item ‘2’ yields the consumption to-the-hour of the current day (subtract value at midnight from current value). Filter item ‘3’ yields yesterday’s consumption (subtract the value of midnight yesterday from midnight today). The resulting values are available in data points, which can be visualized or trended.

The historic filters definitions are managed by historic filter resources. These are templates and stored in the project resources. They can be applied to data points. When editing an historic filter template, all existing historic filter relations are updated accordingly. For more information on how to configure historic filters in the Configurator please refer to Section 7.17.

6.6 CEA-709 Technology

6.6.1 CEA-709 L-INX Device

The CEA-709 L-INX implements a LONMARK device which exposes network variables (NVs) and configuration properties (CPs) from the CEA-709 network to data points in the automation server.

The L-INX has one physical FT port and one IP-852 port, which is accessible over Ethernet. On the L-INX with the RNI option, the automation server node is internally connected either to the FT port or to the IP-852 port. Which one is used can be configured in the CEA-709 port configuration (see Section 5.2.9). The schematic is shown in Figure 134 (a). If configured for the FT channel, the L-INX provides an RNI for remote access to the FT channel. The RNI can be used to commission nodes and trouble-shoot communications on the FT channel.

The L-INX with the CEA-709 router connects the FT port and the IP-852 port. On this L-INX the automation server node is always internally connected to the FT port. The schematic is shown in Figure 134 (b).

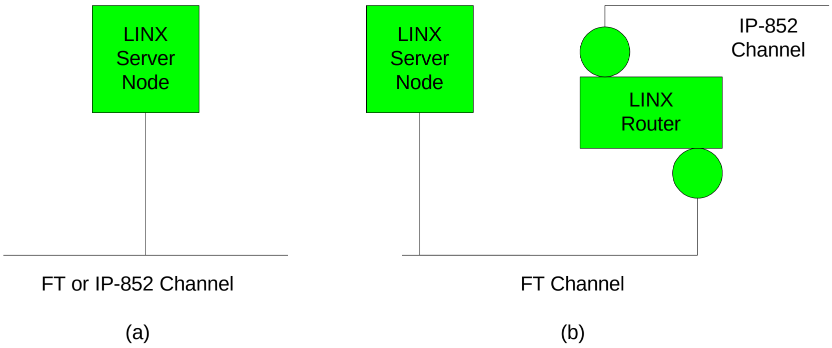


Figure 134: (a) LonMark node on L-INX with RNI, (b) LonMark node and router on the L-INX with router option.

If the automation server shall only expose network variables from the local FT channel and there is no IP-852 backbone, then the router is not needed. In this case, the user needs to commission only the L-INX server node. To attach the FT channel to an IP-852 backbone, the CEA-709 router in the L-INX needs to be commissioned. See Chapter 8 for more information on the built-in router and configuration server.

6.6.2 CEA-709 L-GATE Device

The L-GATE implements a LONMARK device which exposes network variables (NVs) and configuration properties (CPs) from the CEA-709 network for connections to other network technologies.

The L-GATE has one physical FT port and one IP-852 port, which is accessible over Ethernet. The gateway node is internally connected either to the FT port or to the IP-852 port. Which one is used can be configured in the CEA-709 port configuration (see Section 5.2.9). The schematic is shown in Figure 134. If configured for the FT channel, the LGATE-950 provides an RNI for remote access to the FT channel. The RNI can be used to commission nodes and trouble-shoot communications on the FT channel.

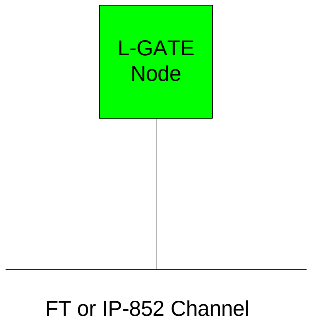


Figure 135: LONMARK node of the L-GATE.

6.6.3 CEA-709 Data Points

Data points in the CEA-709 network are known as network variables (NVs). They have a direction, a name, and a type, known as the standard network variable type (SNVT) or user-defined network variable type (UNVT). In addition to NVs, also configuration properties (CPs) in the CEA-709 network can be accessed as data points. Both standard CP types (SCPTs) and user-defined CP types (UCPTs) are supported.

The typical procedure in configuring the device consists of the following steps:

1. Select the data points of the network to be used on the device (e.g., select the NVs in the CEA-709 network nodes).

2. Create necessary registers, math objects.

3. Select those data points on the device, which shall be exposed as OPC tags or as PLC variables.

The CEA-709 NVs on the device can be created in three different ways:

- **Static NV**: For each selected NV on the network there is a static NV created on the device. This NV can be bound to the NV on the network. Note that adding static NVs to the device results in a change to the default XIF file. The device is assigned a new “model number” to reflect this change (see Section 6.6.4). Static NVs are the way to use NVs in non-LNS systems, where bindings are used for the NVs instead of polling.
- **Dynamic NV**: For each selected NV on the network there is a dynamic NV created on the device. Compared to static NVs, dynamic NVs do not change the XIF interface of the device. The dynamic NVs are created by the network management tool. Currently, only LNS-based tools can manage dynamic NVs. As for static NVs, with dynamic NVs it is possible to use bindings instead of polling.
- **External NV**: The selected NVs on the network are treated as external NVs to the device. The device doesn’t create any NVs on the device, but instead uses polling to read from those NVs and explicit updates to write to the NVs. Therefore, no bindings are necessary for external NVs. For input data points using external NVs however, a poll cycle must be configured. If not configured explicitly, a default poll cycle of 60 sec. is chosen. The default poll cycle can be changed in the project settings menu. Note, that the receive timeout option cannot be used with external NVs.

Based on the NV the data point is derived from, the following kinds of data points are created:

- Simple NVs that hold only one scalar value, e.g., SNVT_amp: Those kinds on NVs are represented as analog data points. The data points holds the current value, NV scaling factors are applied.

- Simple NVs based on an enumeration, e.g., SNVT_date_day: Enumeration types result in multi-state data points. They represent the state of the NV.
- Structured NVs that consists of a number of fields, e.g., SNVT_switch: All structured NVs are represented as user point. That is, the data point is structured similar to the NV it is based on. Beneath the user data point, the individual structure fields are presented as “sub-data points”.

For more information on the different types of network variables and their implications please refer to the application note in Section 22.2. For CPs the allocation type “File” is used.

6.6.4 Static Interface Changes

The device can be configured to use static NVs. Unlike dynamic NVs, static NVs cannot be created in the network management tool. They are part of the static interface and are usually compiled into the device. When static NVs are used, the device changes its static interface and boots with a new one.

Each time the static interface of the device changes (i.e., static NVs are added, deleted, or modified), the model number is changed. The model number is the last byte of the program ID. Thus, a change in the static interface results in a change of the program ID and a new device template needs to be created in the network management tool. A new device template usually means that the device has to be deleted and added again in the database. All bindings and dynamic NVs have to be created again for the new device.

When the Configurator software is connected via LNS, it supports the process of changing the device template for the new static interface. It automatically upgrades the device template of the L-INX device in the LNS database and restores the previous bindings and dynamic NVs. If the L-INX is not configured with an LNS-based tool, this support is not available. The new static interface is only available in a new XIF file or by uploading the new device template into the database. For more information on the static interface and device templates please refer to the application note in Section 22.2.

Important! ***It is not recommended to mix manually created, dynamic NVs outside the virtual function block and static NVs. In this case, the Configurator is not able to restore all manually created dynamic NVs.***

The Configurator supports model number ranges for different projects. By assigning a model number range to a configuration, the Configurator can draw a new model number from within the specified range. This feature is useful, if different device classes shall be deployed and the model numbers need to be coordinated between installers. In this case, the installers can agree on ranges they are free to use. The model number range can be set on the data points tab as shown in Figure 136.

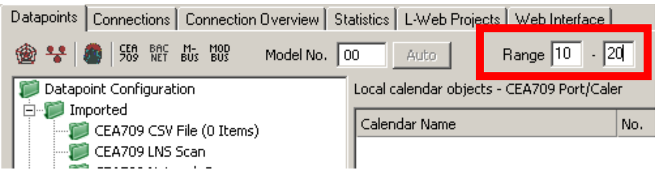
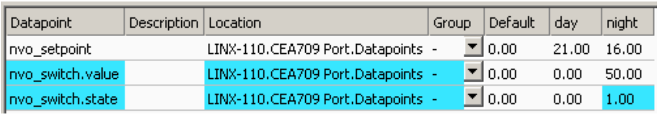


Figure 136: Setting a model number range in for a project.

6.6.5 Limitations for Local CEA-709 Schedulers

CEA-709 schedulers and the CEA-709 calendar adhere to the LONMARK standard objects. For CEA-709, certain restrictions exist that need to be kept in mind. Attached data points can either represent an entire NV or individual elements of a structured NV. CEA-709 schedulers may have several different groups of data points attached, i.e., the value preset may consist of more than one element. For example, a CEA-709 scheduler might schedule a SNVT_temp and a SNVT_switch and have 3 elements in each value preset as depicted in Figure 137.



Datapoint	Description	Location	Group	Default	day	night
nvo_setpoint		LINX-110.CEA709 Port.Datapoints	-	0.00	21.00	16.00
nvo_switch.value		LINX-110.CEA709 Port.Datapoints	-	0.00	0.00	50.00
nvo_switch.state		LINX-110.CEA709 Port.Datapoints	-	0.00	0.00	1.00

Figure 137: Example value presets in CEA-709 schedulers.

Priorities of exception days in a CEA-709 scheduler range from 0 (the highest) to 126 (the lowest). The value 127 is reserved as a default for weekdays.

Further, the implementation as LONMARK standard objects requires the use of configuration properties. If the number of CEA-709 schedulers or their capacities for daily schedules and value presets is changed, the resource and static interface of the CEA-709 port changes. The resources reserved for LONMARK calendar and scheduler objects can be changed in the project settings (see Section 7.3.4). When downloading a project, the software verifies if sufficient resources have been configured. If it detects a problem, the user is notified to update the project settings. The Auto-Set feature automatically selects the right amount of resources.

6.6.6 Limitations for CEA-709 Alarm Servers

Local CEA-709 alarming supports only one alarm server object. This alarm server object is represented by the device's LONMARK node object and facilitates the SNVT_alarm2 output network variable. Acknowledging alarms in the alarm server is adhering to the LONMARK specification and relies on the RQ_CLEAR_ALARM mechanism.

6.6.7 Limitations for Local CEA-709 Trends

Local CEA-709 trend objects support trending multiple data points in all trend modes, interval, COV, and trigger, including aligned intervals. The enable data point is also supported. All data points can be NVs, registers or of any other technology. There is no LONMARK object linked to the trend object. Consequently, trend data cannot be accessed over a LONMARK mechanism.

6.6.8 Dynamic Polling in CEA-709

External input NVs in CEA-709 rely on polling. Static polling can be configured by setting the pollcycle (see Section 6.2.2). In addition, this technology also supports dynamic polling. If the data point Web UI or L-WEB requires a refresh on those NVs, which are otherwise not used, polling is activated at the configured pollcycle. If those data points go out of scope, the polling on the external input NVs is stopped. If no polling is needed at all, the pollcycle setting can be left at zero.

Dynamic polling has no effect on static or dynamic input NVs. These NVs are supposed to have bindings and rely on update events. If static polling is configured via the pollcycle, no change of the pollcycle is made at run-time.

6.6.9 CEA-709 Data Points in Connections

All types of CEA-709 data points can be used in local and global connections. All CEA-709 data points can be sources for auto-generating target technology data points. There is a special treatment of the SNVT_switch, which will generate a binary data point in the target technology.

CEA-709 can also be a technology target for auto-generation (see Section 6.4.3). In this case static NVs with SNVTs are created only. If the source is an input, an NVO will be generated, otherwise an NVI. If the source data point is analog, the SNVT is chosen with the best-matching engineering unit. In many cases there exist SNVTs in fixed-point and floating point versions. The default can be edited in the CEA-709 project settings. If the source is a multi-state data point, a SNVT_count is generated and the source state map is used. If the source is a binary data point, a SNVT_switch is created. In the auto-generate preview the user can review and change the desired SNVTs individually before generating the data points.

6.7 BACnet Technology

6.7.1 BACnet Data Points

Data points in the BACnet technology are known as BACnet objects. They have a specific type (e.g. analog input or binary output) and a set of properties, which describe the data point more closely. The actual value is stored in the “Present_Value”.

On the device, there exist two classes of BACnet data points:

- **BACnet server objects** (SO): These BACnet objects configured by the Configurator software to be allocated *locally* on the device. These objects can be accessed by the BACnet building control system or operating workstations. They support COV subscriptions to deliver value changes in an event-driven way. Local server objects can be created as AI, AO, AV, BI, BO, BV, MI, MO, MV, Accumulator, Pulse Converter, Loop, Large Analog Value, Integer Value, Positive Integer Value, Character String Value, Octet String Value objects.

- **BACnet client mappings** (CM): For certain applications, it is necessary that the device acts as a BACnet client. This functionality can be configured by activating a *client mapping*. Client mappings can be of the type *Poll*, *COV*, *Write*, *Auto* or *Value*. This specifies how the BACnet client accesses other BACnet objects on the BACnet network. The *Auto* method determines the best way (poll, COV, or write) to talk with other server objects. *Poll* is used for objects that need to read data from other BACnet objects in a periodic manner. *COV* is used to subscribe for COV at other BACnet objects in order to get updates in an event-driven fashion. *Write* is used to send updates to other BACnet objects. *Value* refers to a combined read and write client mapping. When writing a value to this client mapping, the value is written to the remote BACnet object. As soon as the Present_Value of the remote BACnet object is updated, the value is transferred back.

The direction of BACnet server objects deserves a closer look. The direction specified for data points in the Configurator software always refers to the network view of the communication. The definition of input and output objects in BACnet, however, refers to the process view, which is opposite to the network. Therefore, a BACnet analog input (AI) object is modeled as an analog output data point. The direction of client mappings naturally refers to the network communication. Therefore, a write client mapping is represented as an analog output data point.

In BACnet commandable objects can be written with values at a certain priority. The value with the highest priority is in effect. When revoking a written value, the NULL value is written. This takes back the value. When all written values are withdrawn, the Relinquish_Default value is in effect.

For BACnet server objects the write priority defines which priority is written to the commandable server object. It is possible to create additional *priority write* data points, which can be configured to write at other priority levels. This may be necessary, if two parts of an application are required to write with two different priorities. To know, which priority slots are used in a commandable object, additional *priority read* data points can be added. They reflect the value on a given priority slot.

The default value feature of a data point is mapped to the Relinquish_Default property for commandable objects. For BACnet objects, which are not commandable, the Present_Value is initialized with the specified default value.

Analog BACnet objects have no fixed network unit. Depending on the chosen unit system, the analog BACnet object will be created with the specified metric (SI) or U.S. unit in the Engineering_Unit property. This means that the BACnet server object changes its representation on the BACnet network (i.e., engineering unit and value), when the unit system is changed on the device.

6.7.2 BACnet Alarming

BACnet alarming on the device is based on the *intrinsic reporting* mechanism. Currently, algorithmic reporting is not supported. Alarm conditions can only be applied to data points, which map to BACnet server objects. If defined, the intrinsic reporting properties of the underlying BACnet objects are enabled. Alarm conditions can be specified for analog input, output, value objects (AI, AO, AV), for binary input, output, value objects (BI, BO, BV), and for multi-state input, output, value objects (MSI, MSO, MSV). With BACnet intrinsic reporting alarm conditions on binary output (BO) and multi-state output (MSO) can only be feedback alarms.. These restrictions do not apply, if the alarm condition reports to a generic alarm server.

Alarm servers in the BACnet technology are mapped to BACnet Notification Class (NC) objects. Each alarm server is mapped to one NC. The notification class number can be configured in the object instance number property of the alarm server object.

Remote alarms in the BACnet technology refer to a remote NC object. When the device starts up, the remote alarm object reads out the current alarm state of the remote NC and reporting objects. To get notified about alarm transitions during run-time, the device registers in the Recipient_List of the remote NC object.

Some BACnet devices do not send a usable text in their alarm notification messages. For those devices the alarm client provides the option **Ignore alarm message text**. If this option is enabled, the alarm client ignores the message text of an alarm notification and reads the description property of the alarmed object instead.

6.7.3 BACnet Schedulers and Calendars

BACnet schedulers and the BACnet calendar adhere to the standard schedule and calendar object in BACnet. For each scheduler a BACnet Schedule object is created. The calendar deserves a closer look. For each calendar pattern a BACnet Calendar object is created. The visible calendar on the Web UI is therefore a collection of BACnet calendar objects. Each calendar pattern therefore is associated with a BACnet object instance number. The calendar pattern “Holidays” is for example visible as CAL,1 on the BACnet port.

The BACnet schedule object allows only objects of one selected data type to be scheduled. Therefore, schedulers on BACnet can only schedule one class of data points (e.g., only one group of analog data points). As a consequence, the value preset in BACnet always has only one element. The name of the value preset is not stored in BACnet. It is not accessible over the BACnet network, either. Therefore, a default name is created, such as ‘22 °C’ for an analog value. An example of two scheduled BACnet objects is shown in Figure 138. With the extended BACnet features enabled in the project settings, a preset label can be assigned to a specific scheduled value. For example the value ’16 °C’ can be assigned ‘night’. Click in the column header and type the desired text.

Datapoint	Description	Location	Group	Default	22 °C	night
bac_temp1	temp	BACnet Port.Datapoints	1	0.00	22.00	16.00
bac_temp2	temp	BACnet Port.Datapoints	1	0.00	22.00	16.00

Figure 138: Example value presets in BACnet schedulers.

Priorities of exception days in a BACnet scheduler range from 1 (the highest) to 16 (the lowest). Weekdays in BACnet have no priority.

Changing the number of calendar patterns in a BACnet calendar can only be done through the configuration software and not during run-time. The individual calendar pattern entries in the calendar patterns can be changed at run-time. Therefore, it is advisable to reserve a suitable number of calendar patterns in a BACnet calendar and leave them empty if not needed immediately.

6.7.4 BACnet Trend Logs

Trending in the BACnet technology is based on the BACnet TrendLog object. A number of restrictions apply to trend log objects in BACnet. Trend log objects must be created by the Configurator software. These objects are accessible over the BACnet network for other BACnet devices and operator workstations (OWS). All configuration properties can be modified by the Configurator software as well as an OWS. The number of trend log objects cannot be changed at run-time. Therefore, if it is intended that an OWS configures the trend logs, a suitable number of empty trend log objects (i.e., without attached data points) must be created in the Configurator software.

In BACnet trend logs, only one data point can be trended per trend log object. The trended data point can be either a local BACnet server object or a remote BACnet object accessed through a client mapping, showing the referenced property for trending to the OWS. Data points of other technologies and the min/max/avg algorithms can be trended as generic data points without having a BACnet property reference.

BACnet trend logs support interval, COV and trigger-mode logs, aligned intervals are available in interval mode. The setting linear and ring-buffer logging is mapped to the Stop_When_Full property of the underlying BACnet trend log object. This setting in the Configurator software is a default and can be overridden by writing to the Stop_When_Full property by the OWS. The trend log object adheres to BACnet revision 12.

If an enable data point is configured by the Configurator software, the Log_Enable property is written with the value of that data point. If no enable data point is configured, the Log_Enable is TRUE as a default and can be modified over the network.

The fill-level action is mapped to generating a buffer event notification in the BACnet trend log object. The fill-level trigger can still be used for e-mails even if no notification class is configured in the BACnet trend log object. The fill-level percentage maps to the Notification_Threshold property. The percentage setting in the Configurator software is a default and can be changed by the OWS over the network.

The Notify_Type and Notification_Class BACnet properties are usually written by the OWS. If they shall be defined by the data point configuration, configure the **Notify Type** and **Notification Class** data point properties on the trend log object to non-default values. When doing so, the OWS can no longer change them permanently.

The BACnet technology also supports *remote trend logs*. A remote trend log is basically a BACnet trend log client, which accesses trend data on another device. The remote trend can load the trend data from the remote device and supply it to L-WEB or the trend CSV files.

6.7.5 Dynamic Polling in BACnet

Reading client mappings in BACnet rely either on COV or on polling. Static polling can be configured as a fallback, if COV is not supported, by setting the pollcycle (see Section 6.2.2). Data points which are not used by other objects do not subscribe via COV or perform polling. In addition, this technology also supports dynamic polling. If the data point Web UI or L-WEB requires a refresh on those client mappings, COV subscriptions are made or polling is activated at the configured pollcycle. If those data points go out of scope, the polling on the remote BACnet object stops and COV is unsubscribed.

If no static polling is needed at all, the pollcycle setting can be left at zero in the client mapping. In this case, only a COV subscription is made, if the device supports COV. If COV is not supported, polling is only started, as soon as the values are required. This is especially important on MS/TP channels with devices, that do not support COV.

6.7.6 BACnet Data Points in Connections

BACnet data points can be used in local and global connections. In BACnet commandable objects can be written with values at a certain priority. The value with the highest priority is in effect. When revoking a written value, the NULL value is written. This takes back the value. When all written values are withdrawn, the Relinquish_Default value is in effect. In other technologies there is no notion of taking a value back. To model this behavior, a distinctive *invalid* value can be written to such a data point. For those data points that do not an intrinsic invalid value, it can be specified when editing the data point. To make a BACnet object convey that invalid value to the opposite side, enable the property **Relinquish to Invalid**.

BACnet data points can be auto-generated from other data point sources (see Section 6.4.3). Only BACnet server objects can be generated and the connected value is reflected in the Present_Value property. Which type of BACnet object is created depends on the type of the source data point or of the source structure member. For analog sources, analog objects are created. The best-matching BACnet engineering unit is chosen. Other properties of analog objects are copied from the source data point, including min and max present value. Multi-state objects are created for source enumeration types. Which state IDs exist is documented in the BACnet multi-state texts array. This information is copied from the source and made compatible with BACnet restrictions by renumbering state IDs.

Normally BACnet AI, BI, MI are created out of input source data points and AO, BO, MO out of output source data points. The BACnet project settings allow changing this default to BACnet value objects AV, BV, MI. In the auto-generate preview the user can review and change those object types individually before generating the data points.

6.8 IEC61131 Variables

IEC61131 variables are used to exchange data with the IEC61131 program. These variables are represented in the data point configuration as register data points and can be connected to other data points, e.g. to CEA-709 NV points, via data point connections.

In contrast to CEA-709 or BACnet variables, IEC61131 variables are always represented as single data point. In case of scalar values (representing CEA-709 scalar or enumeration types) one of the following basic data types might be used:

- **Double**: A register of base type *double* is represented by an *analog* data point. It can hold any scalar value. No specific scaling factors apply.
- **Signed Integer**: A register of base type *signed integer* is represented by a *multi-state* data point. This register can hold a set of discrete states, each identified by a signed stats ID.
- **Boolean**: A register of base type *boolean* is represented by a *binary* data point. This register can hold a Boolean value.

Structured IEC61131 variables, representing for example structured NVs, or customer defined IEC61131 structures, are stored as user type:

- **User**: A *user* data point contains un-interpreted, user-defined data. The data is stored as a byte array. A user data point does not include any other meta-data. This type of data point also serves as a container for otherwise structured data points and represents the entire data structure. User data points can only be connected to other user data points of the same data length.

6.9 Storage of User Documentation

The LINX-12x/15x/22x and LGATE-950 devices provide a storage area for saving user files. This storage can be used for placing project documentation files on the device. To transfer files to the device, a standard FTP client can be used. Log in as 'admin' or 'operator' user in the FTP client. Use the passwords defined for these users on the device. After the login, the FTP client is connected to the home directory. Files can be stored in this directory.

Note, that the storage capacity in the home directory is limited. It is recommended to use an external storage card, such as an SD card or USB memory stick. If doing so, change to the following directories to store files on external storage:

- SD card: Change to '/media/sdcard0'.
- USB stick: Change to '/media/usbmem0'.

6.10 Regular Expressions

Some features of the Configurator use regular expressions to perform complex operations on text. These apply to data point filters, naming rules and folder copy and rename. A regular expression is a pattern that describes a set of strings. It is applied to an input text and performs pattern-matching by evaluating the expression string consisting of literal characters to match and meta-characters. Literals match themselves (**`abc`** matches exactly 'abc') whereas meta-characters may match one or more characters of the input text. The available metacharacters are listed in Table 18. Regular expressions can handle abbreviations, such as **`\d`** instead of **`[0-9]`**. The abbreviations provided are listed in Table 19.

Meta-Character	Meaning
`.`	Matches any single character.
[ ]	Indicates a character class. Matches any character inside the brackets (for example, **`[abc]`** matches ‘a’, ‘b’, and ‘c’).
`^`	If this meta-character occurs at the start of a character class, it negates the character class. A negated character class matches any character except those inside the brackets (for example, **`[^abc]`** matches all characters except ‘a’, ‘b’, and ‘c’). If **`^`** is at the beginning of the regular expression, it matches the beginning of the input (for example, **`^[abc]`** will only match input that begins with ‘a’, ‘b’, or ‘c’).
`-`	In a character class, **`-`** indicates a range of characters (for example, **`[0-9]`** matches any of the digits ‘0’ through ‘9’).
`?`	Indicates that the preceding expression is optional: it matches once or not at all (for example, **`[0-9][0-9]?`** matches ‘2’ and ‘12’).
`+`	Indicates that the preceding expression matches one or more times (for example, **`[0-9]+`** matches ‘1’, ‘13’, ‘666’, and so on).
`*`	Indicates that the preceding expression matches zero or more times.
`??`, **`+?`**, **`*?`**	Non-greedy versions of **`?`**, **`+`**, and **`*`**. These match as little as possible, unlike the greedy versions which match as much as possible. Example: given the input ‘<abc><def>’, **`<.*?>`** matches ‘<abc>’ while **`<.*>`** matches ‘<abc><def>’.
`( )`	Grouping operator. Example: **`([0-9]+,)*[0-9]+`** matches a list of numbers separated by commas (such as ‘1’ or ‘1,23,456’).
`{ }`	Indicates a match group. The actual text in the input that matches the expression inside the braces can be retrieved through the sequence **`\0`**, **`\1`**, etc.
`\`	Escape character: interpret the next character literally (for example, **`[0-9]+`** matches one or more digits, but **`[0-9]\+`** matches a digit followed by a plus character). Also used for abbreviations (such as **`\a`** for any alphanumeric character; see Table 19 below). If **`\`** is followed by a number *n*, it matches the *n*-th match group (starting from 0). Example: **`<{.*?}>.*?</\0>`** matches ‘<head>Contents</head>’.
`$`	At the end of a regular expression, this character matches the end of the input. Example: **`[0-9]$`** matches a digit at the end of the input.
`|`	Alternation operator: separates two expressions, exactly one of which matches (for example, **`T|the`** matches ‘The’ or ‘the’).
`!`	Negation operator: the expression following **`!`** does not match the input. Example: **`a!b`** matches ‘a’ not followed by ‘b’.

Table 18: Metacharacters in Regular Expressions.

Abbreviation	Matches
`\a`	Any alphanumeric character: **`([a-zA-Z0-9])`**
`\b`	White space (blank): **`([ \\t])`**
`\c`	Any alphabetic character: **`([a-zA-Z])`**
`\d`	Any decimal digit: **`([0-9])`**
`\h`	Any hexadecimal digit: **`([0-9a-fA-F])`**
`\n`	Newline: **`(\r|(\r?\n))`**
`\q`	A quoted string: **`(\"[^\"]*\")|(\'[^\']*\')`**
`\w`	A simple word: **`([a-zA-Z]+)`**
`\z`	An integer: **`([0-9]+)`**

Table 19: Abbreviations for regular expressions

When performing a replace operation on an input text, match groups are used in the output template. Match groups are delimited by curly braces containing a matching pattern. As an example the regular expression **`{[0-9]?[0-9]}:{[0-9][0-9]}`** contains two match groups. The first match group matches any combination of one or two digits. The second matches any two-digit combination. To make the replacement effective, the entire regular expression has to match. In the example the regular expression matches any one or two-digit combination followed by a ':' and by any two-digit combination.

Then the output is assembled by specifying an output template. In that output template, both literal text and references to the match groups can be specified. The first match group is denoted by **`\0`**, the second by **`\1`** and so on. Using the output template **`ref\0-\1`** on the example expression, the following input texts will produce these results:

- "ab1:22c" matches, replacement is "ref1-22",
- "foo22:11bar" matches, replacement is "ref22-11",
- "ab22:1c" does not match, no replacement results in "ab22:1c".

7 The L-INX Configurator

This Chapter gives step-by-step instructions on how to commission the L-INX and L-GATE, create a data point configuration with network variables, BACnet objects, how to expose those data points to the automation server, and how to maps data points to other technologies in the universal gateway.

7.1 Installation

7.1.1 Software Installation

The L-INX Configurator must be used to setup the data point configuration of the L-INX automation server and the L-GATE universal gateway. The Configurator is installed as a plug-in tool for all LNS-based network management tools as well as a stand-alone tool (for systems without LNS).

System requirements:

- LNS 3.1 SP8 U1, LNS 3.2 TE SP5, OpenLNS (for LNS mode),
- Windows Vista, Windows 7, Windows 8 (64 bit) or Windows Server 2003 (32 bit), Windows Server 2008, Windows Server 2012,
- Internet Explorer 9 or higher.

The L-INX Configurator can be downloaded from the LOYTEC Web site http://www.loytec.com. When asked for the type of installation, there are two options to choose from. Select **Typical** to install the required program files. Select **Full** to install the LONMARK resource files along with the software. This option is useful, when the system does not have the newest resource files.

7.1.2 Registration as an LNS Plug-In

If the CEA-709 L-INX or the L-GATE shall be configured using LNS-based tools (e.g., NL200 or LonMaker), the L-INX Configurator needs to be registered as an LNS plug-in. In the following, the process is described for LonMaker TE. Otherwise, please refer to the documentation of your network management tool on how to register an LNS plug-in.

To Register in LonMaker TE

1. Open LonMaker and create a new network.

2. Click **Next** until the plug-in registration tab appears in the Network Wizard. Select the **LOYTEC LINX Configurator (Version X.Y)** from the list of **Not Registered** (see Figure 139).

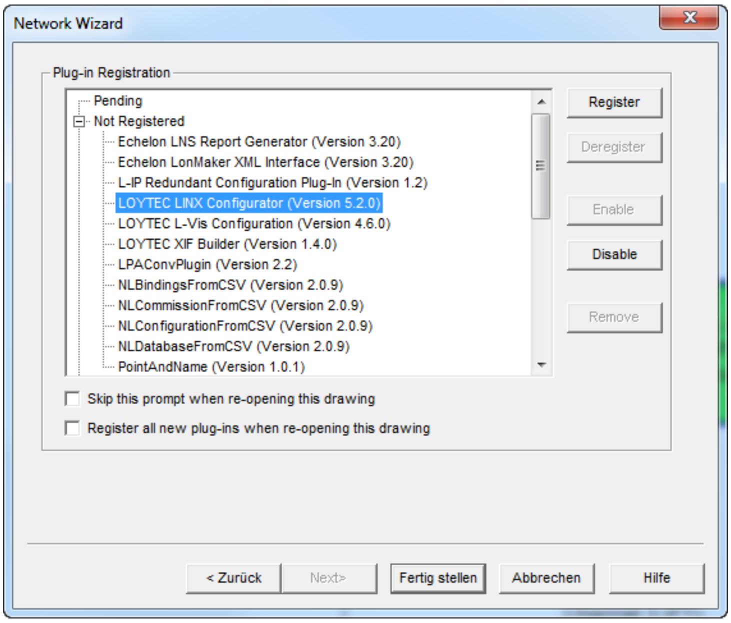


Figure 139: Select the Plug-in to be registered.

3. Click **Register**. The Configurator now appears in the **Pending** list.

4. Click **Finish** to complete the registration.

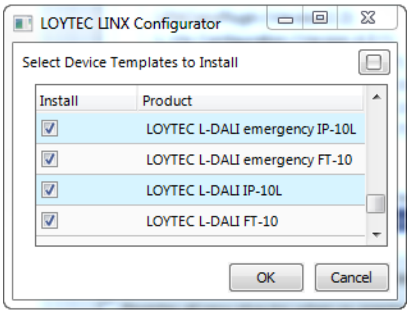


Figure 140: Select device templates for installation.

5. A dialog appears to optionally select the device templates to be installed. Deselect unneeded device templates to speed up registration. Click **OK** to continue.

6. The selected device templates are added automatically and XIF files are copied into the LNS import directory.

Note: *If you are using multiple databases (projects) make sure you have registered the plug-in in each project.*

7. Under LonMaker → Network Properties → Plug-In Registration make sure that the LOYTEC LINX Configurator (Version X.Y) shows up under Already Registered.

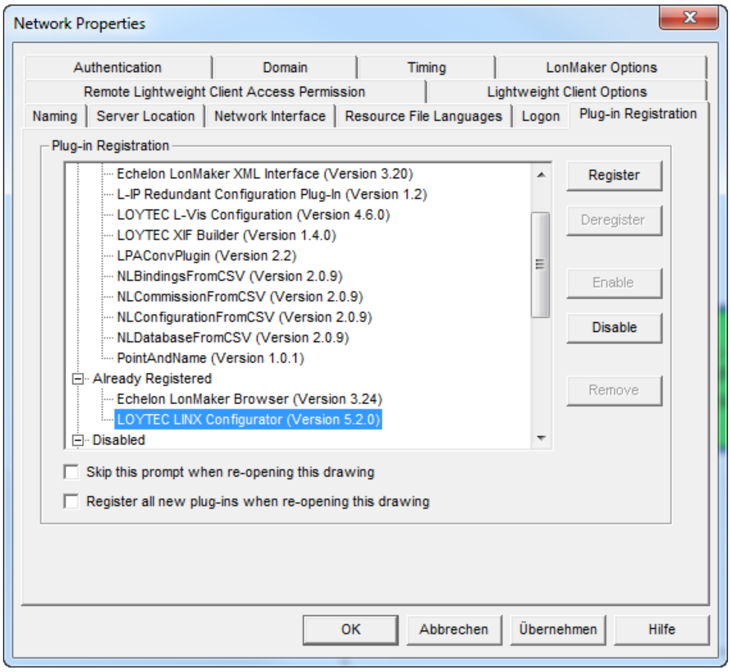


Figure 141: Check that the L-INX Configurator is properly registered.

7.1.3 CEA-709 Operating Modes

The Configurator can be used in on-line, off-line, and stand-alone mode. On-line and off-line mode refers to the 2 operating modes of your LNS network management software.

- **On-line Mode**: This is the preferred method to use the Configurator. The network management tool is attached to the network and all network changes are directly propagated into the network. This mode must be used to add the device, commission the device, extract the port interface definition, and download the configuration into the device.
- **Off-line Mode**: In off-line mode, the network management software is not attached to the network or the device is not attached to the network, respectively. This mode can be used to add the device using the device templates, create the port interface definition and to make the internal connections.
- **Stand-alone Mode**: The Configurator can also be executed as a stand-alone program. This mode is useful for the engineer who doesn’t want to start the configuration software as a plug-in from within network management software (e.g., NL-220, LonMaker or Alex). Instead the engineer can work directly with the device when online or engineer it offline.

7.2 Data Point Manager

The Configurator uses a central concept to manage data points. The data point manager is located on the **Datapoints** tab as shown in Figure 142. It is used to select, create, edit and delete data points. The dialog is divided into three sections:

- The folder list (number 1 in Figure 142),
- The data point list (number 2 in Figure 142),
- And a property view (number 3 in Figure 142).

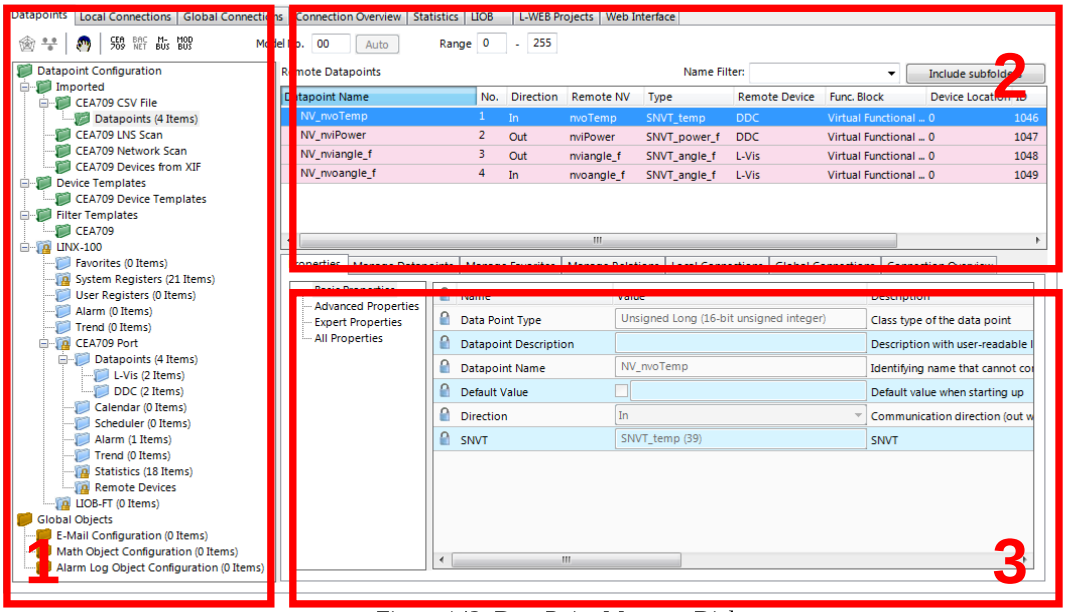


Figure 142: Data Point Manager Dialog.

7.2.1 Folder List

At the left is a list of folders which is used to sort the available data objects by their category. There are a number of predefined folders available:

- **Imported**: This folder has a number of sub-folders for different import methods:
  - **CEA-709 CSV File:** This folder is used to display data points imported from CSV files.
  - **CEA-709 LNS Scan**: This folder is used to hold data retrieved from a network database scan.
  - **CEA-709 Network Scan**: This folder holds NVs scanned online from an attached CEA-709 network.
  - **CEA-709 Devices from XIF**: This folder holds sub-folders and NVs created from XIF files.
  - **BACnet Network Scan**: This folder is used to display data points retrieved by an online scan of the BACnet network.
  - **BACnet EDE File**: This folder is used to display data points imported from an EDE file.

  Data objects in the import folder are not stored on the device when the project is downloaded. They represent data objects which are available on remote devices and are shown here as templates to create suitable data objects for use on the device by selecting the **Use on Device** option.

- **Filter Templates**: This folder contains the created data point templates. They contain a set of properties, which are applied to data points, when they are created on the device. There is a sub-folder for filter templates specific to different technologies, e.g. CEA-709.
- **LINX-XXX**, **LGATE-XXX**: This is the device folder of the L-INX or the L-GATE. It contains all the necessary data points which constitute to the device's configuration. These data points are created on the device when the configuration is downloaded. The following subfolders may be present depending on the particular model:
  - **Favorites**: This folder contains freely configurable symbolic links to data points, which may reside anywhere in the folder structure. This folder

represents a way to assemble an alternate logical view to the data point hierarchy. This folder is also available on the Web UI or the LCD UI.

- **System Registers**: This folder contains system registers, which provide information on the device itself.
- **User Registers**: This folder holds user-definable registers. These registers are not visible on the underlying network and are intended for internal usage.
- **Scheduler**: This folder contains generic scheduler and calendar objects. These provide technology-independent scheduling functionality. Any data point can be alarmed using a generic scheduler object.
- **Alarm**: This folder contains generic alarm servers. These provide technology-independent alarms. Any data point can be alarmed using a generic alarm server.
- **Trend**: This folder contains generic trend log objects. These can record historical values for any data points.
- **CEA-709 Port**: This folder contains data points, schedulers, calendars, trend logs, statistics, and remote data points of the CEA-709 network technology. See Section 7.2.2.
- **BACnet Port**: This folder contains data points, schedulers, calendars, trend logs, statistics, and remote data points of the BACnet network technology. See Section 7.2.2.

- **Global Objects**: This top-level folder contains sub-folders that organize specific application objects that operate on data points.
  - **E-mail Configuration**: This folder contains e-mail templates. An e-mail template defines the destination address and text body of an e-mail, which is triggered by data points and may contain data point values or file attachments. To create an e-mail template, select the folder and use the context menu.
  - **Math Objects Configuration**: This folder contains math objects. Math objects are used to perform a predefined calculation on a number of input data points and write the result to a defined set of output data points. Each math object contains one formula. To create a math object, select the folder and use the context menu.
  - **Alarm Log Configuration**: This folder contains the alarm log objects. Each alarm log object creates a historical log of alarm transitions of one or more alarm objects (alarm server or client). To create an alarm log, select the folder and use the context menu.

Using the context menu on a folder, sub-folders may be created to organize the available objects. If new objects are created automatically, they are usually placed in the base folder and can then be moved by the user to any of his sub-folders. Note, that the folder structure described above cannot be changed by adding or deleting folders at that level.

The context menu also allows editing folder properties. Choose **Properties …** from the context menu to open a property editor. Change name and description there.

7.2.2 Network Port Folders

Each network port folder on the device has the same structure of sub folders. These sub folders are:

- **Datapoints**: This folder holds all data points, which are allocated on the network port. To create a data point, select the folder and use the context menu.
- **Calendar**: This folder is used to hold a locally available calendar object with its calendar patterns (definitions of day classes like holiday, maintenance day, and so on).

Current devices allow one local calendar object. To create a calendar, select the folder and use the context menu.

- **Scheduler**: This folder is used for local scheduler objects. Each of these objects will map to a local scheduler on the device's network port. Configuring schedules through these objects actually transfers *schedule configuration data* to the underlying scheduler objects on the network port. To create a scheduler, select the folder and use the context menu.
- **Alarm**: This folder is used for local alarm server objects. Each of these alarm server objects represent an alarm class, which other objects can report alarms to. Other devices can use the alarm server object to get notified about alarms. To create an alarm server object, select the folder and use the context menu.
- **Trend**: This folder is used for local trend log objects. Each of these objects will be able to trend data points over time and store a local trend log file. To create a trend log object, select the folder and use the context menu.
- **Statistics**: This folder contains registers, which provide communication statistics specific to the network port.
- **Remote Devices**: This folder is used to collect all remote calendars, schedulers, trend logs, and alarm client objects, which were created from network scan data. For each remote device, a subfolder will be created where the objects referencing this device are collected.

7.2.3 Data Point List

At the top right, a list of all data objects which are available in the selected folder is shown. From this list, objects may be selected (including multi-select) in order to modify some of their properties. Click on the **Include Subfolders** button to show all data points of the selected data point folder and all its sub-folders. This can be a convenient way for multi-select across folders.

To filter for data point names, enter a search text into the **Datapoint Name Filter** text box and hit *Enter*. A drop-down list holds the previously used filters available. Filters can also specify name patterns for sub-data points using a dot. Typing the first dot expands all filtered data points to the first sub-level. Continuing typing after the dot then filters on names of those sub-data points. For example, typing "sw.val" selects all data points having "sw" in their name, then expanding to their first sub-level and filtering all sub-data points on that level having "val" in the name. For complex name filters, regular expressions may be used (see Section 6.10).

The list order can be changed manually by drag-and-drop. Select one or more data points and drag them to the desired position in the data point list. The data points will get a new order number.

The list can be sorted by clicking on one of the column headers. For example, clicking on the **Direction** column header will sort the list by direction. Other columns display **Datapoint Name**, **NV name**, **SNVT**, **Object Name**, object **Type** and **Instance** number, allocation (**Alloc**) of server object (SO) and/or client mapping (CM), number of attached **Client Maps**, and the data point unique **ID**. To apply the current sort order as the new data point order on the device, right-click on the column header and select **Renumber Datapoints**. Alternatively, select from the menu **Tools → Renumber Datapoints.**

The **OPC** column provides check boxes for each data point. If checked, the respective data point is exposed to OPC on the device. Deselect the check box, if a data points shall not be exposed to OPC. Note, that deselected data points do not add to the OPC tag limit.

The **Param** column provides check boxes for each data point. If checked, the respective data point is exposed to the parameter file. The **PLC in** and **PLC out** columns provide check-boxes, which define if data points are visible inside the IEC61131 PLC program as input and output variables.

New objects may be created in the selected folder by selecting **New Datapoint** command in the context menu of the data point list. A plus ⊞ sign in the list indicates that the data point contains sub-points. Clicking on the plus ⊞ sign expands the view.

If an entry in the data point list denotes a reference to another data point (e.g., a scheduled value reference beneath a scheduler object), right-click on that item and choose **Go to related data point** in the context menu. This will navigate your selection to the referenced data point.

For the alarming, scheduling, trending (AST) features, there are columns, which display icons for data points that are attached to an AST function. See Table 20 for details.

Icon	Data Point Usage
	Data point is scheduled
	Data point has an active alarm condition
	Data point has an inactive alarm condition.
	Data point has sub-data point with alarm condition.
	Data point is a trigger for e-mails.
	Data point used for trending.
	Data point is a link.
	Data point has sub-data points, which are links.

Table 20: Icons for used data points in the data point list view.

The data points in this list are color coded to make general information visible to the user at-a-glance. The color coding is described in Table 21.

Color	Coded Information
ao1 (yellow)	Data point is user-created and can be downloaded to the device.
MAC (blue)	Data point is fixed on the device and cannot be edited, e.g. system registers.
dark red	Data point is user-created and its technology object (e.g. dynamic NV) has been deleted. The data point, however, is still used in the configuration. It will not work on the device until the technology object is created again.

Table 21: Color coding used for data points in the data point list view.

7.2.4 Property View

When one or multiple data points are selected, the available properties are displayed in the property view. Properties which are read-only are marked with a lock sign. When applying multi-select, only those properties common to all selected data points are displayed. According to the frequency of usage, different views for the properties are provided. **Basic Properties** list the most frequently used ones. **All Properties** is a list of all available properties for the data point. Depending on the network technology and data point class, different properties may exist.The **Name Filter** in the properties tab allows quick filtering of properties that have a matching text in their name. For example, type “OPC” to filter the OPC Tag property.

Data point properties common to all technologies:

- **Datapoint Name**: This is the technology-independent data point name. This name may be longer than and different to the name of the native communication object (i.e., network variable). Data point names must be unique within a given folder. The maximum length is limited to 64 ASCII characters.
- **Datapoint Path**: This informational property specifies the entire path of the data point within the data point hierarchy. The maximum length is limited to 64 ASCII characters.
- **Datapoint Description**: This is a human-readable description of the data point. There are no special restrictions for a description.
- **OPC Tag**: If enabled, the data point will be exposed to OPC. If not enabled, this data point does not contribute to the limit of OPC tags.
- **Parameter**: If enabled, the data point will be exposed to the parameter file. Those data points are visible to the LWEB-900 parameter view [5]. A parameter data point is also persistent. See Section 6.2.5.
- **PLC in - Logic Variable**: If enabled, the data point will be usable in the IEC61131 PLC program as an input variable. This is only available on the L-INX.
- **PLC out - Logic Variable**: If enabled, the data point will be usable in the IEC61131 PLC program as an output variable. This is only available on the L-INX.
- **Use Pollcycle value as**: For input data points, this property defines whether the input shall use a receive timeout or be constantly polling. See Section 6.2.2.
- **Poll on Startup**: For input data points this property defines, whether the data point shall be polled once at start-up. Poll-on-startup can be enabled independently of the poll cycle. See Section 6.2.2.
- **Pollcycle**: For input data points, this property defines the poll cycle in seconds. Set this property to 0 to disable polling. See Section 6.2.2.
- **Receive Timeout**: For input data points, this property defines the receive timeout in seconds. Set this property to 0 to disable polling. See Section 6.2.2.
- **Min Send**: For output data points, this property defines the min send time in seconds. See Section 6.2.2.
- **Max Send**: For output data points, this property defines the max send time in seconds. See Section 6.2.2.
- **Send-on-delta**: For output data points this property defines, if value updates shall be sent only once they meet the COV condition of the data point. For analog data points the analog COV increment is used. If not checked, updates are always transmitted according to min and max send times. See Section 6.2.7.
- **Use Linear Scaling**: If this property is enabled, the analog values are pre-scaled from the technology to the data point. This scaling is in addition to any technology-specific scaling factors. If enabled, the properties **Custom Scaling Factor** and **Custom Scaling Offset** accept the scaling factors. See Section 6.2.7.
- **Custom Scaling Factor, Custom Scaling Offset**: These properties only exist, if linear scaling is enabled. They apply to analog data points only. See Section 6.2.7.
- **Only notify on COV**: This property assists for binary and multi-state input data points. It defines, if a data point shall trigger an update only when the value changes or on every write. If this is enabled, consecutive writes with the same value do not trigger an update. If you want to convey every write, disable COV on the data point.
- **Persistent**: This property defines, if the last written value shall be stored as a persistent value. Persistent data points restore that value after a restart from the persistent storage. See Section 6.2.4.

- **Default Value**: This property defines a default value (see Section 6.2.3). Enter a default value to enable this feature in the data point. Delete the value entirely to remove the default value. If no default value is defined, this property reads “N/A”. The default is no default value.
- **Historic Filter**: This property allows defining historic filters for the scalar data point. See Section 6.5.6.
- **Point Type**: This is the base data point type, e.g., “Analog Datapoint”.
- **Direction**: This is the data point direction. Use input, output or value as directions.
- **Network Unit**: For analog data points this property contains the definition of an engineering unit of the scalar value as represented on the network, e.g., “°C”. A human-readable text for the engineering unit is displayed and can be entered. If the unit is known by the Configurator as a convertible unit, it is displayed with a green checkmark (see Section 6.2.13).
- **Unit SI**: If the data point has a convertible unit, a unit representation in the metric (SI) system can be chosen. If the SI unit system is active, all values are converted to this unit, e.g. °C. For a non-convertible network unit this option is not available.
- **Unit U.S.**: If the data point has a convertible unit, a unit representation in the U.S. system can be chosen. If the U.S. unit system is active, all values are converted to this unit, e.g. °F. For a non-convertible network unit this option is not available.
- **Analog Datapoint Max Value**: For analog data points this property contains the upper limit of the supported value range. Note that this does not define an alarm limit.
- **Analog Datapoint Min Value**: For analog data points this property contains the lower limit of the supported value range. Note that this does not define an alarm limit.
- **Analog Datapoint Precision**: For analog data points this property defines the number of decimals. ‘0’ specifies an integer value. Display units may use this to format the floating point value accordingly.
- **Analog Datapoint Resolution**: For analog data points this property defines the smallest possible value increment.
- **Analog Point COV Increment**: This property is valid for analog input data points. It specifies by which amount the value needs to change, before an update is generated. If every write shall generate an update even when the value does not change, specify 0 as the COV increment. If any value change shall generate an update, delete the value, which results in **Any**.
- **Active Text**: For binary data points this property defines a human-readable text for the active state (true).
- **Inactive Text**: For binary data points this property defines a human-readable text for the inactive state (false).
- **Current State Map**: For multi-state data points this property defines the multi-state map. It must be set to a valid multi-state map or it points to User/UndefinedStates. Click on to assign a state map.
- **State Count**: For multi-state data points this property defines the number of discrete states.
- **State Text**: For multi-state data points this property defines a human-readable state label for each state.

7.2.5 Tracking Data Point Usage

Data points can be used by other objects, such as connections, schedulers, math objects and many more. To get an overview on data point usage, the data point list contains the column

Use. This column provides a usage counter of the individual data points. If usage is ‘0’ the data point is not used by any other object.

If the usage count is larger than zero, choose the item **Show Usage …** from the data point context menu. This opens up a report window showing the objects referring to the selected data point. An example is shown in Figure 143.

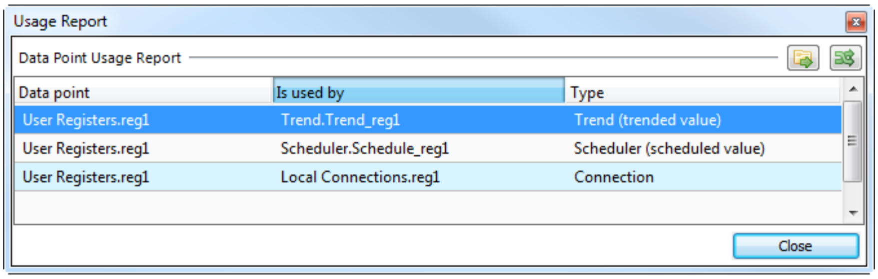


Figure 143: Data point usage report.

Each line reveals an object using the data point. Select a line and click on the **Go to data point** button . This will navigate yours selection to the reported object.

7.2.6 Managing Multistate Maps

Multistate data points have a descriptive set of state texts for their state IDs. To manage those state IDs and state texts among many multistate data points, they refer to *multistate maps*. Some technologies have a fixed set of such multistate maps others have freely configurable multistate maps (e.g, user registers). Editing a multistate map affects all multistate data points, which are using that particular map. It is not necessary to edit each data point individually. For managing multistate maps go to the menu **Tools → Manage State Maps …** .

To Edit a Multistate Map

1. Click on the **Configure** button in the **Current State Map** property of a multistate data point. This opens the multistate map manager as shown in Figure 144.

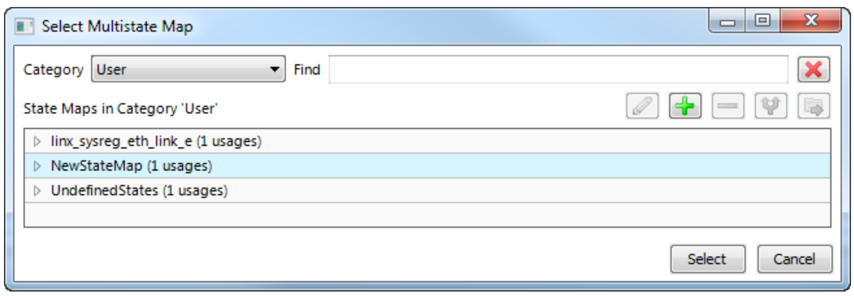
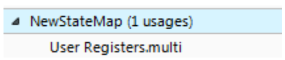


Figure 144: Select multistate maps in the multistate map manager.

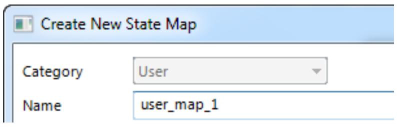
2. Select a **Category** and an existing state map in the **State Maps** list and click on **Select**. Maps that are fixed and cannot be changed are marked with a lock symbol .

3. Expand a state map line to see where this state map is used. Select the usage and click the **Go to Data Point** button . This navigates to the data point.

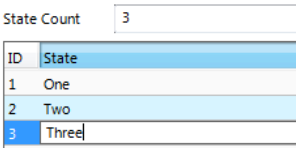


4. If a new multistate map shall be created, click the **Create State Map** button .

5. In the **Create New State Map** dialog enter a state map Name.



6. Then enter the desired number of states and edit the state texts as needed and click into the list of states. Edit state IDs and texts to your needs. Pressing Enter goes to the next line. Finally click the **Save** button.

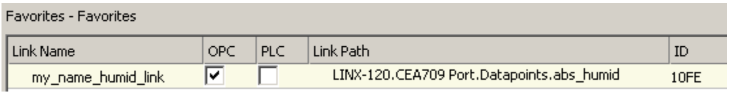


7. Select the newly created multistate map and click the **Select** button. The selected map is now assigned to the data point.

7.2.7 Organizing Favorites

There is a special **Favorites** top-level folder in the device data point folder hierarchy. This folder contains freely configurable symbolic links to data points, which may actually reside anywhere in the folder structure. This folder represents a way to assemble an alternate logical view to the data point hierarchy.

To configure favorites, select any data point from any location in the data point folder hierarchy and drag it onto the favorites folder. This will create a data point link, which is displayed in the data point list:



The **link name** can be edited to something different that the original data point name. The contents of this folder are also available on the Web UI or the LCD UI. The link names are displayed there. The data point links can also be individually exposed to the OPC server or PLC program notwithstanding if their original source is exposed or not.

Furthermore, the user can create sub-folders in the favorites folder and beneath to build a hierarchy of data point links. Folder links are, however, not allowed.

For editing links of favorites for a large number of data points, the **Manage Favorites** tab in the property view area of the data point manager provides a fast solution. The data point tree is replicated in that tab as shown in Figure 145. Select a folder and optionally click on the button to include favorites in sub-folders. Enter a filter expression to **Filter**, which applies to the favorite name. For example, enter ‘room1’ to display all favorites that contain this as a sub-string.

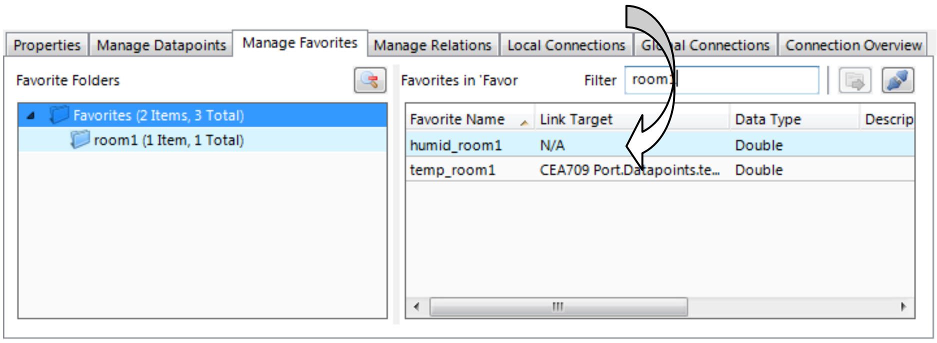


Figure 145: Manage favorites tab.

To link favorites to other data points using the manage favorites tab, navigate to the desired folder in the main folder tree of the data point manager and select the data point to be linked. Drag the data point onto the **Link Target** column. For detaching links, use multi-select on the desired favorites and click the detach button . On linked favorites you may click on the button , which jumps to the linked data point in the data point list.

7.2.8 Managing Property Relations

Property relations can be linked to data points in various user dialogs (e.g. alarm condition dialog). For editing links of property relations for a large number of data points, the **Manage Relations** tab in the property view area of the data point manager provides a fast solution. The data point tree is replicated in that tab as shown in Figure 146. Select a folder and optionally click on the button to include property relations of data points in sub-folders. Enter a filter expression to **Filter**, which applies to both the data point name and relation type. For example, enter ‘feedback’ to display all feedback value property relations.

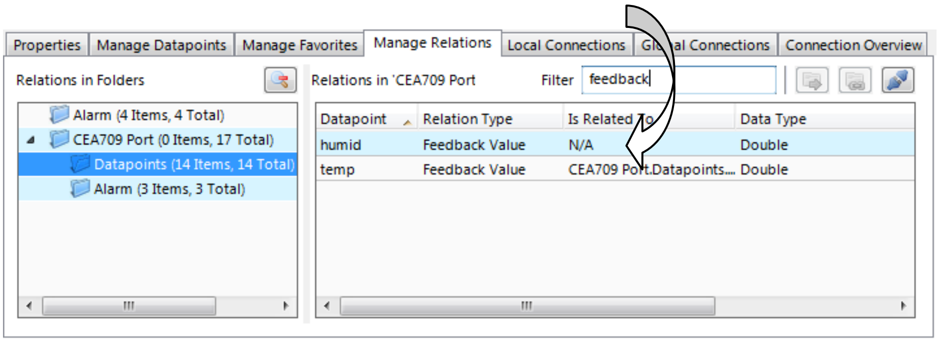


Figure 146: Manage relations tab.

To link property relations to other data points using the manage relations tab, navigate to the desired folder in the main folder tree of the data point manager and select the data point to be linked. Drag the data point onto the **Is related To** column. For detaching links, use multi-select on the desired property relations and click the detach button . On linked property relations you may click on the button , which jumps to the linked data point in the data point list.

7.2.9 CEA-709 Properties

Apart from the common data point properties discussed in Section 7.2.4 the data points of the CEA-709 technology have additional properties. Depending on whether a NV is local or external (remote), the properties may vary.

- **NV Allocation**: This property defines how a data point shall be allocated on the device. Choices are “Static NV”, “Dynamic NV”, and “External NV”. If the allocation type cannot be changed, this property is locked.

- **SNVT**: This property defines the SNVT of the NV, e.g., “lux (79)”.
- **Invalid Value**: This property defines the “invalid value” for the NV. If set, this specific value will be interpreted as “invalid” in the data point. If known by the SNVT, the invalid value is filled in. Otherwise, the user can specify an invalid value.
- **CEA-709 Mapping Information**: This information is derived from the SNVT. It defines how the NV contents are mapped to the data point.
- **NV Scaling A, B, C**: These are the scaling factors known from the SNVT table. The scaling factors are applied to translate a raw NV value into the scalar representation of the data point.
- **Data Type**: This is the basic NV data type. This is usually filled in from the SNVT definition.
- **Local NV Member Index**: This property specifies the NV member index within a given functional block. This must be a unique index in the functional block, which identifies the NV after other NVs have been added or removed from the interface.
- **Local/Remote NV Index**: This property specifies the NV index. For local, static NVs this is the NV index of the static NV. For external NVs, this is the NV index of the NV on the remote device.
- **Local/Remote NV Name**: This property specifies the programmatic name of the NV. For local, static NVs this is the programmatic name of the static NV. For external NVs, this is the programmatic name of the NV on the remote device.
- **Local/Remote Functional Block**: This property specifies the programmatic name of the NV. For local, static NVs, one of the reserved functional blocks can be selected.
- **Local/Remote NV Flags**: This property specifies the NV flags. For local (static or dynamic) NVs, the flags can be configured. For external NVs, these flags are only informational.
- **Remove NV Information**: For external NVs, this property contains the information on the remote device and the NV selector on that device.
- **Remote Device ID**: For external NVs, this property contains information on the remote device by listing the program ID and location string.
- **Remote Device Address**: For external NVs, this property contains the CEA-709 network addressing information to access the node, i.e., subnet, node, and NID.
- **Retry Count**: For external NVs, this property defines the retry count. The default is 3.
- **Repeat Timer**: For external NVs, this property defines the repeat timer in milliseconds. The default is 96 ms.
- **Transmit Timer**: For external NVs, this property defines the transmit timer in milliseconds. The default is 768 ms.
- **LNS Network Path**: If available from an LNS scan, this property specifies the LNS network path of the device where the given NV exists.
- **LNS Channel Name**: If available from an LNS scan, this property specifies the LNS channel name of the device where the given NV exists.

7.2.10 BACnet Properties

Apart from the common data point properties discussed in Section 7.2.4 the data points of the BACnet technology have additional properties. Depending on whether a server object and/or a client mapping exists, the properties may vary.

- **Engineering Units**: For analog BACnet server objects, this property defines the engineering units from the BACnet standard. One of those units can be chosen from a drop-down box, if this property is not locked. Depending on the active unit system, the BACnet engineering units property is either metric (SI) or U.S.

- **Server Object Type**: This property defines the BACnet object type of the underlying BACnet server object. It can be changed within the class, i.e., for an analog data point, the server object type analog input, analog output, or analog value can be chosen.
- **Commandable**: This property defines, if the underlying BACnet server object is commandable. For BACnet value objects (AV, BV, MSV) this property can be edited to create commandable or non-commandable BACnet value objects.
- **Relinquish to invalid value**: This property defines whether the data point maintains the Relinquish_Default value, if the priority array is empty or assumes the invalid value. By default, this property is false and the Relinquish_Default value is used. Setting this property to true can be beneficial when used in a connection to withdraw a value in another technology.
- **Server Object Name**: This property defines the object name of the underlying BACnet server object. It must be unique among all server objects. It can be up to 64 characters and is read-only on the BACnet network by default. If the option to keep OWS values is enabled in the BACnet project settings (See Section 7.3.5), this property is writeable and the server object name is only initialized by this property. Any changes made by the OWS will be kept also after a new configuration has been downloaded.
- **Server Object Instance No**: This property defines the object instance number of the underlying BACnet server object.
- **Server Object Description**: This property defines the object description of the underlying BACnet server object. It can be left blank.
- **Server Object Device Type**: This property defines the object device type of the underlying BACnet server object. It can be left blank.
- **Server Accumulator Offset Correction**: This property is available for accumulator objects only. It is activated as a default. It ensures that the written data point value is corrected in the Present_Value as soon as the Value_Set property is written and requests a certain Present_Value. That correction offset is then added to the written data point value, e.g., when received over a connection.
- **Get Active Priority**: Set this check box to let the data point reflect the active priority of the local or remote BACnet object. The priority is a number between 1 and 16. This property is only applicable for commandable BACnet objects.
- **Allocate Server Object**: This Boolean property defines, if a server object shall be allocated for the data point. This option is useful, when a local server object shall be allocated for a client mapping.
- **Allocate Client Mapping**: This Boolean property defines, if a client mapping shall be allocated for the data point. This option is always set, if at least on client mapping is attached.
- **Client Map Count**: This property defines the number of client mappings attached to a data point. A data point can have one read client map or *n* write client mappings.
- **Client Map [n]**: This is a list of client mappings. The property shows a summary of the client mapping parameters.
- **Client Confirmed COV**: This Boolean property defines, if a client map subscribes with the confirmed COV service. If not enabled, the unconfirmed COV is used.
- **Client Map Type**: This property determines the type. It can be one of the following: Poll, COV, Auto, Write, or Value (see Section 6.7.1).
- **Client Write Priority**: For a write or value client map, this property defines which priority is used for writing.

- **Remote Instance Number**: This property specifies the object instance number of the remote server object. The object type cannot be edited.
- **Value Read Mode**: For value client mappings, this property defines the read mode: Poll, COV, or Auto.
- **BACnet Notify Type**: When set to 'default' the Notify_Type property is to be written by the OWS. Change this to 'Event' or 'Alarm' to set the property via the data point configuration.
- **BACnet Notification Class**: If notify type is non-default, this property is enabled. It sets the Notification_Class via the data point configuration.
- **BACnet Event Enable (To**-XXX): If notify type is non-default, this property is enabled. It sets the Event_Enable property via the data point configuration.

7.3 Project Settings

The project settings allow defining certain default behavior and default settings used throughout the project. To access the project settings go to the menu **Settings → Project Settings…**. This opens the project settings dialog, which provides several tabs as described in the following sections.

7.3.1 General

The general tab of the project settings as shown in Figure 147 contains settings independent of the technology port. The settings are:

- **Project Name**: This setting allows entering a descriptive name for the project.
- **Parameters managed by**: This setting defines, which instance is managing data point parameters. As a default, parameters are assumed to be managed by LWEB-900 and the Configurator does not download parameter values to the device. If changed to parameters managed by Configurator, the Configurator downloads parameter values and merges parameter changes back into the configuration. This setting does not affect L-IOB parameters, which are always managed by the Configurator. For more information on data point parameters refer to Section 6.2.5.
- **L-IOB: Always ignore L-IOB manual/auto differences**: Set this check box to ignore differences in the manual mode parameter of L-IOB I/Os.
- **Device Configuration Download Default**: This group of settings defines, how the download of device configuration parameters shall proceed. If **Download only data point configuration** is selected, the configuration download does not download anything else than the data point configuration. If **Ask** is selected, the download will pop up a dialog in which the user can choose what to download. If **Download specific** is selected, the project settings of this dialog determine what is downloaded onto the device. The individual items to download are selected by individual checkboxes below the download option. As a default, the configuration download includes the schedules and calendar patterns as well as L-WEB projects.
- **Background polling**: Activate this option to permanently poll data points, even if they have no pollcycle assigned. The data points are polled one-by-one. The poll rate can be configured. This setting is not active by default. For more information on background polling refer to Section 6.2.2.
- **Automatically structure imported data points for faster OPC browsing**: This option enables the automatic generation of sub-folders when using data points on the device. A sub-folder is created for each scanned or imported device. This allows OPC clients to browse the OPC tags in a hierarchical way.

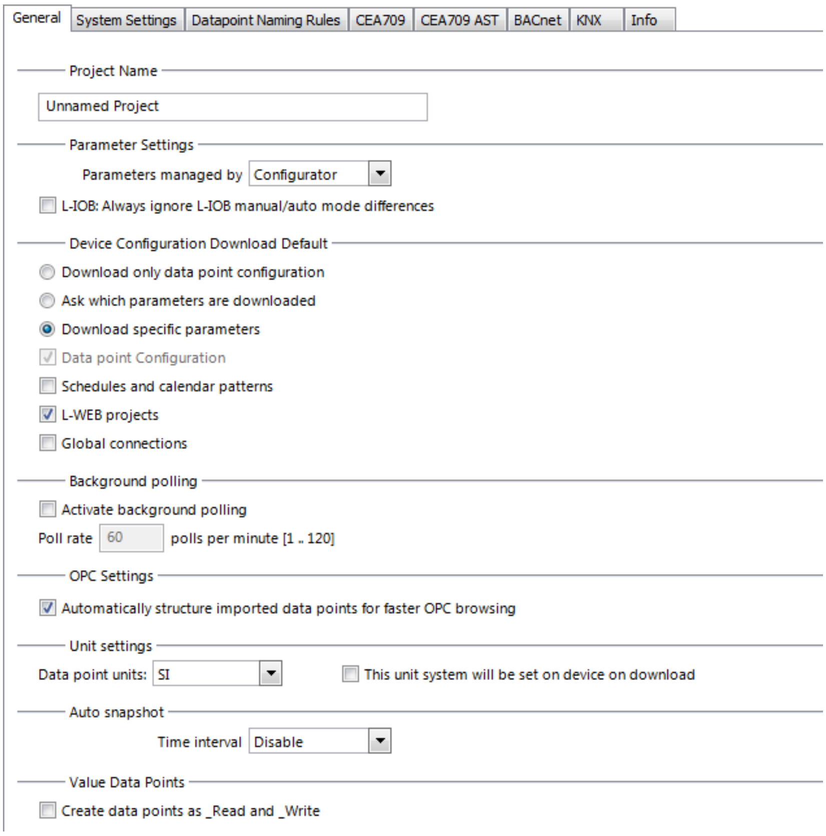


Figure 147: General Project Settings.

- **Data Point Units**: This setting defines which unit system is active in the Configurator. Depending on the active unit system, SI units are displayed or U.S. units, as defined per data point. This setting, however, influences only the display in the Configurator. If the device shall be configured to run on this same unit system, set the check box **This unit system will be set on device on download**. In this case each configuration download will ensure the device runs on the selected unit system.

Important! ***If this effectively changes the unit system on the device, all persistent values are reset to their default values converted to the respective unit system.***

- **Auto snapshot**: This setting is off by default. It can be enabled in the drop-down box by selecting a time interval, which is used by the Configurator to produce configuration snapshots. The user can jump back and forth between those snapshots. Snapshots can also be made explicitly by clicking the snapshot button in the tool bar.

- **Value Data Points**: Select this option to create data points in the old style with "_Read" (input) and "_Write" (output) data points. Old configurations will have this option set to continue creating data points using the same style. Newly created configurations should not use this setting.

7.3.2 Data Point Naming Rules

The data point naming rules tab allows specifying how data point names are automatically derived from imported or scanned objects on the network. For each technology a rule can be defined, how individual components shall assemble the desired data point name. The technology is chosen in the tree view. The name components are added by clicking on . Then click on the name component and choose which content shall be assigned from a drop-down menu. To remove a component, click on it and choose **Remove** from the drop-down menu. An example rule is shown in Figure 148.

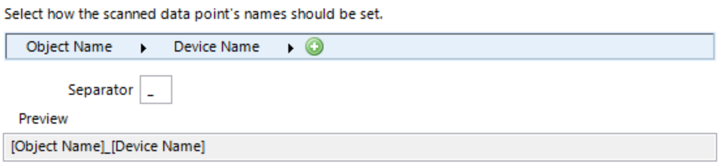


Figure 148: Adding components to naming rules

A **Separator** character can be defined, which is inserted between the data point name components. As a default the underline '_' is configured. The Preview shows how names would look like with the chosen separator and name components.

The CEA-709 data point naming rules (see Figure 149) work on scanned network variables. The preview shows how names would look like, when the check marks are modified. Note, that changing the name components does not change already scanned NVs; this setting affects only new scan results.

- **Use programmatic name, Use display name**: This option decides how the data point name is extracted from the NV. The programmatic name is the NV name from the XIF file and is limited to 16 characters. The display name may be extracted from LNS, which allows displaying a different, longer name than the programmatic name.

- **Add Subsystem Name via Filter**: This line can be edited by adding LNS sub-system components to the data point name. Click on to add another component. The first subsystem1 is the one containing the device, the second subsystem2 the one containing subsystem1, and so on. Click on the arrow symbol for editing a filter expression, which defines how the sub-system text is transformed to the name component, e.g. copy the last three characters. Examples for regular expressions can be inserted by clicking the question mark symbol and selecting a transformation. In the preview section the filter expression can be tested against an example text. For more information on using regular expressions please refer to Section 6.10.

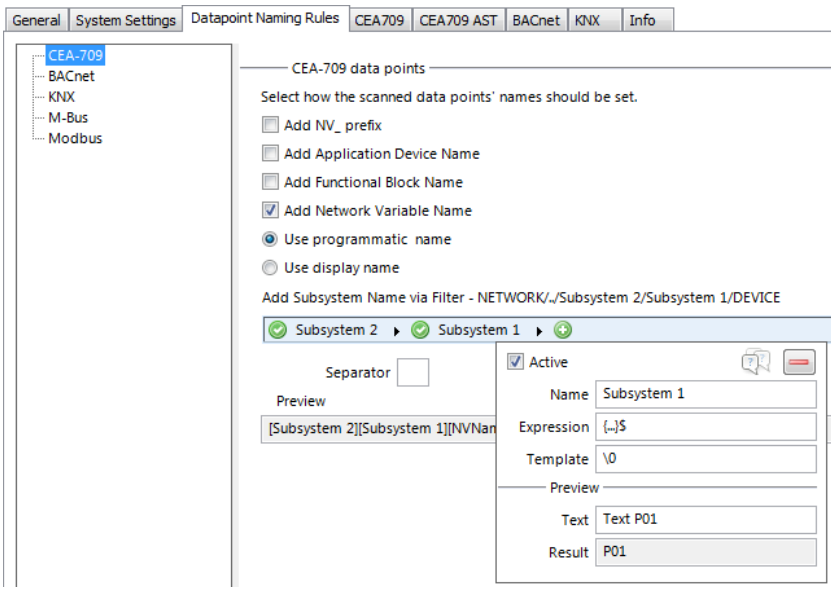


Figure 149: Data Point Naming Rules Project Settings.

7.3.3 CEA-709 Settings

The CEA-709 configuration tab as shown in Figure 150 allows configuring properties of the device's CEA-709 port. The options are:

- **Enable Legacy Network Management Mode**: This group box contains check boxes for each CEA-709 port of the device. Put a check mark on the port, if this port shall be operated in the legacy network management mode. In that mode, the port does not use the extended command set (ECS) of network management commands. This can be necessary to operate the device with some network management tools that do not support the ECS.
- **Default Pollcycle for External NVs**: When using external NVs, this poll cycle is set as a default for input data points. The poll cycle can be edited individually in the properties view of the data point manager.
- **Use state-member of SNVT_switch as**: This setting defines how the state member of the SNVT_switch shall be mapped to a data point. Depending on how the data point shall be used, it can be binary or multi-state. The multi-state setting allows setting the UNSET state explicitly. As a binary point the UNSET state is implicitly chosen, if the value is invalid.
- **Omit unused child data points of UNVT/UCPT structures**: This setting defines, that if set, also unsued sub-data points of user-defined structure types are not downloaded onto the device. This option can reduce the total amount of data points in the configuration. As a default it is not enabled to allow full structure information after an upload to the PC even if the user does not have the original resource files installed.
- **Prefer floating point SNVTs when auto-generating**: If enabled, this setting defines, that auto-generated static NVs prefer flaoting point SNVTs over fixed-point types, e.g. a SNVT_temp_f will be generated instead of a SNVT_temp.
- **Configuration Download**: This group box contains self-configuration settings for the CEA-709 ports. This is necessary if the device shall be used without being commissioned by a network management tool. Set the check mark and enter the CEA-709 domain and subnet/node information. If operated in self-configured mode, the CEA-709 network can be scanned using the network scan (see Section 7.7.4) and external NVs can be used on the device. Note, that the domain must match the nodes' domain on the network and the subnet/node address must not be used by another device.

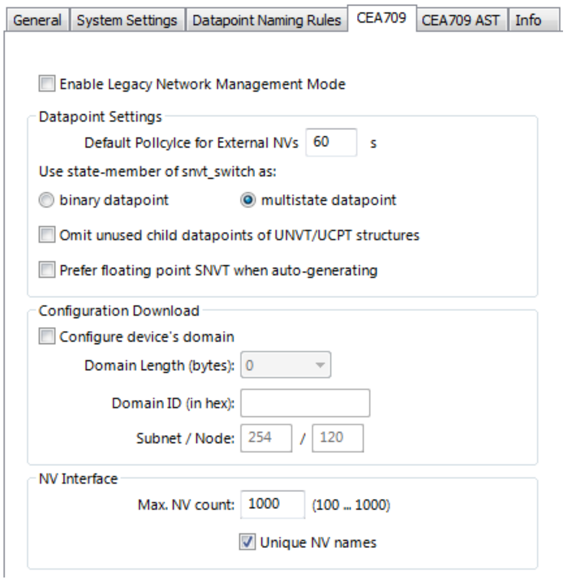


Figure 150: CEA-709 Project Settings.

- **Max NV count**: On models that support it, the maximum number of NVs in the static interface can be increased from the default number. Note that doing so changes the static interface of the device (see Section 6.6.4).

- **Unique NV names**: In the default setting the programmatic names of static NVs must be unique on the interface. When removing this check box, this restriction is relaxed to be unique per function block. Note, that once disabling this option, it cannot be enabled anymore.

7.3.4 AST Settings

For CEA-709 devices, the use of alarming, scheduling, and trending (AST) features requires additional resources (functional objects and NVs). This changes the static interface. Since the number of used resources also influences the performance, the CEA-709 AST tab allows configuring those resources for the project. In this tab, the required number of scheduler units that may be instantiated and their capacity may be configured (how many time/value entries, value templates, bytes per value template, etc.). It contains the following options and settings, which are relevant to calendar and scheduler functionality of the device:

- **Enable Calendar Object:** This checkbox enables a LONMARK compliant calendar object on the device. It is automatically enabled together with local schedulers, since the two are always used together.
- **Enable Scheduler Objects:** This checkbox enables local LONMARK compliant scheduler objects on the device. Checking this box will automatically enable the calendar as well.
- **Enable Remote AST Objects:** This checkbox enables the functional object for NVs, which are used to access remote AST objects. If this box is checked, the *Clients* functional block is included in the static interface.
- **Enable AST v2:** This checkbox enables the AST interface version 2 for local CEA-709 schedulers on the device. This interface is not compatible with older devices. The new interface provides access to the value label descriptions in schedule presets for remote schedulers.
- **Number of calendar patterns:** Specifies the maximum number of different exception schedules (day classes like holiday, maintenance day) supported by this calendar object.
- **Total number of date entries:** Specifies the maximum number of date definitions which may be stored by the calendar. This is the sum of all date definitions from all calendar entries. A date definition is for example a single date, a date range, or a week and day pattern.
- **Number of local schedulers:** This is the number of local scheduler objects which should be available on the device. Each local scheduler data point created in the data point manager will connect to one of these scheduler objects. There may be more scheduler objects available on the device than are actually used at a certain time. It is a good idea to have some spare scheduler objects ready, in case another scheduler is needed.
- **Number of daily schedules:** This is the maximum number of schedules supported by each scheduler object. This number must at least be 7, since a scheduler always needs to provide one schedule for each day of the week (default weekly schedule). For each special day defined by the calendar or embedded exception day, an additional daily schedule is required to support it.
- **Entries in Time/Value table:** This is the total number of entries in each scheduler defining a value template that should apply on a specific day starting at a specific time (the time table).
- **Number of value templates:** This is the maximum number of value templates supported by each scheduler.
- **Data size per value template:** This specifies the buffer size reserved to hold the data for each value template. More data points or bigger data structures require a bigger value buffer.

- **Max. number of data point maps:** Specifies the maximum number of individual data points that this scheduler is able to control.
- **AST Configuration Size:** This number in Bytes is calculated from the scheduler settings above and represents the total size of the LONMARK configuration properties file stored on the device. While certain settings can be freely edited within the given limits, the resulting configuration size is also limited.

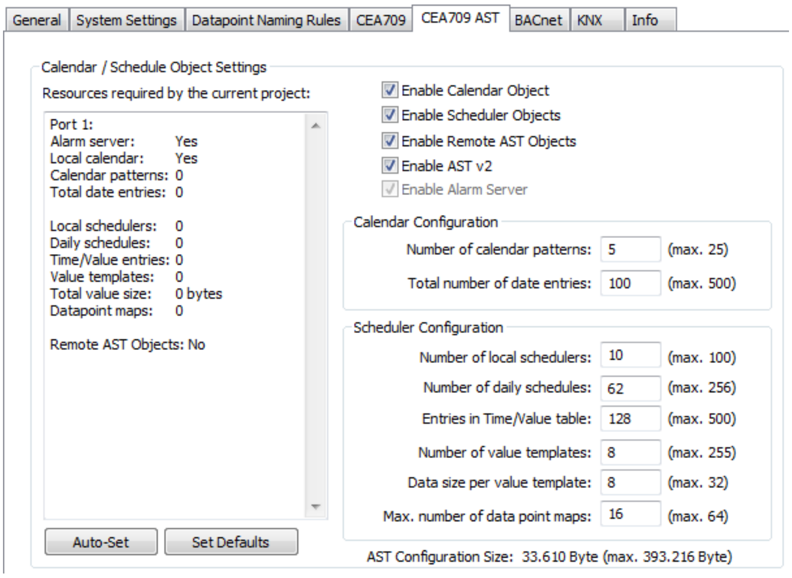


Figure 151: CEA-709 AST Project Settings.

As can be seen from the above list, it is not easy to configure a LONMARK scheduler object. There are many technical parameters which need to be set and which require some knowledge of how these scheduler objects work internally. Therefore, the configuration software provides the following mechanisms to help in choosing the right settings:

- **Resources required by the current project:** The absolute minimum settings required by the current project are shown in a table at the left side of the window. This data may be used to fill in the values at the right side, but some additional resources should be planned to allow for configuration changes which need more resources.
- **Auto-Set:** This button may be used to let the configuration software decide on the best settings to use, based on the current project. Since the current projects resource usage is taken as a starting point, all schedulers and calendar patterns in the project should first be configured before this button is used.
- **Set Defaults:** This button will choose standard values for all settings. In most cases, these settings will provide more resources than necessary.

7.3.5 BACnet Settings

The BACnet configuration tab as shown in Figure 152 allows configuring properties of the device's BACnet port. The options are:

- **Enable Unsolicited COV**: Put a check mark on this option to enable COV-U on the BACnet port. When active, the device sends unsolicited COV broadcast on all BACnet objects, when their value changes in accordance to the respective COV rules.
- **Always create value objects on auto-create**: If activated, the auto-create BACnet points function of the configuration software creates commandable value objects (AV, BV, MV) instead of output objects (AO, BO, MO) and non-commandable value objects (AV, BV, MV) instead of input objects (AI, BI, MI). This feature can be activated if the regular input/output model is not desired.

- **Use 255.255.255.255 for global broadcast**: This setting overrides the standard behavior of BACnet to send broadcasts as global IP broadcasts. This can solve scanning problems with some BACnet devices.
- **Enable periodic I-Am broadcast**: This setting enables the periodic transmission of I-Am broadcasts. Specify the interval in seconds. If disabled, the device sends an I-Am only when starting up. This is the default behavior of BACnet devices.
- **Support proprietary properties**: Check this option if a scan on a remote device shall find proprietary properties in addition to the standard properties of supported objects.
- **Enable extended BACnet features**: Check this option to enable additional properties in BACnet server objects. This affects Elapsed_Active_Time, etc. properties in binary objects, custom properties in scheduler objects (value labels).
- **Keep OWS values in device**: Check this option, if BACnet properties changed by the OWS shall be maintained in the device even after a new configuration download. Without this option, a new configuration will overwrite any changes made by the OWS with the values defines in the configuration (e.g., high and low limits of alarm conditions). The default is to overwrite with configuration values. If this option is set, BACnet object names are made writeable by the OWS as well.
- **String encoding**: This setting defines, how strings in BACnet objects are encoded. By default it is ASCII/UTF-8, which is compatible with most BACnet software. To support characters of Western European languages, choose ISO-8859-1. To support Unicode character sets (e.g., Japanese) select UCS-2.
- **Default Poll cycle, Default COV Expiry, Default Write Priority**: This setting defines the default values that are used when creating new client mappings. Changing this option does not affect already existing client mappings. The default write priority is also used when writing to commandable server objects.
- **Preallocated Calendar Objects**: This setting defines how many BACnet calendar objects shall be created as a default. These are filled up with calendar patterns as they are defined.
- **Native L-IOB objects start with instance nr**: This setting defines, which BACnet object instance numbers shall be assigned to native L-IOB BACnet objects starting with the defined instance number. This enables you to keep native BACnet objects for L-IOB I/Os in a separate instance number range than regular BACnet objects.

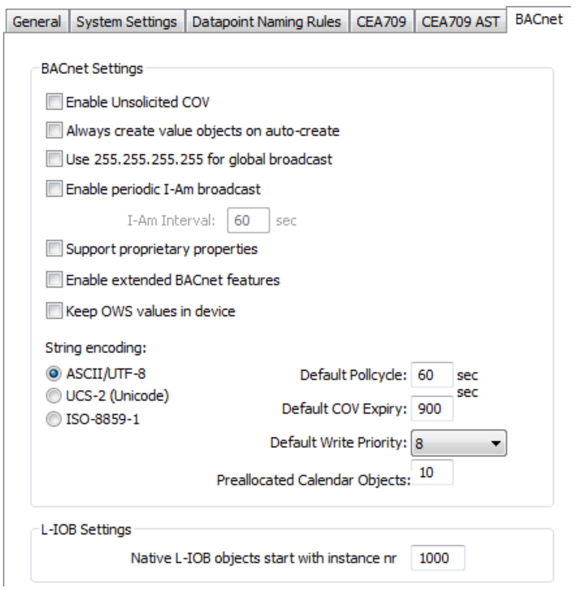


Figure 152: BACnet Project Settings.

7.3.6 System Settings

This tab is shown in Figure 153. It is available only with the newest firmware version and can be used to configure the device through the Configurator. In the configuration tree on the left-hand side the user can select certain groups of settings, e.g. Web server settings. The dialog displays the settings of the selected group in the dialog area. The structure is similar to the menu structure on the Web UI.

Under the port configuration tree, the user can enable or disable communication protocols on the device’s ports. Enabled protocols are marked with a checkmark. Click on the checkmark and toggle it. Note, that depending on the device model communication protocols on other ports may be disabled.

The IP address settings cannot be changed in this dialog. The FTP server can not be disabled in this dialog, either. This ensures that the Configurator can maintain connection to the device.

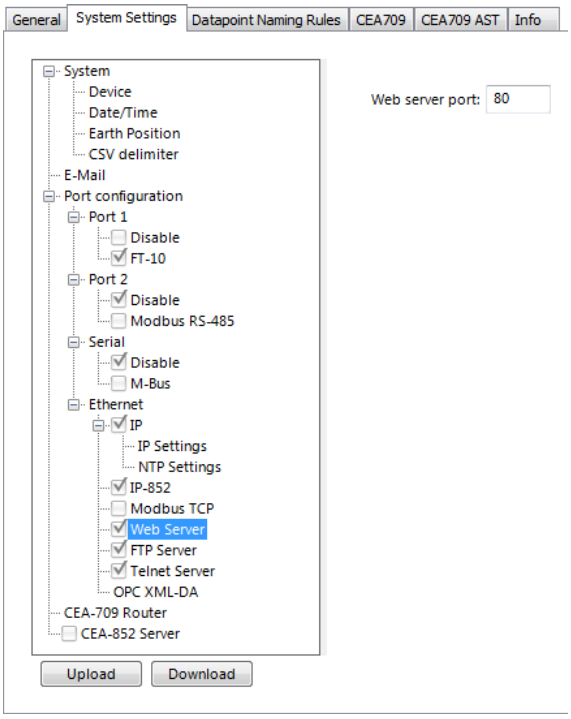


Figure 153: Device Configuration Settings

The **Upload** button can be used to get the current device settings from the device and display them in this dialog. The **Download** button can be used to explicitly transfer the settings from this dialog onto the device. Those changes will be visible immediately on the Web UI but take effect only after a reboot of the device.

Important! ***After downloading the device settings from this dialog the changes will be visible immediately on the Web UI but the device needs to be rebooted to let the changes take effect.***

7.3.7 Project Information

This tab is shown in Figure 154. It provides fields to enter additional information about the project such as author name and a reference field. The comments text area allows entering free text describing the project. For instance it can be used to document a revision history. The fields **Last Saved** and **Configurator Version** are filled in when saving the project.

When creating a new project without having it saved for the first time, those fields are empty.

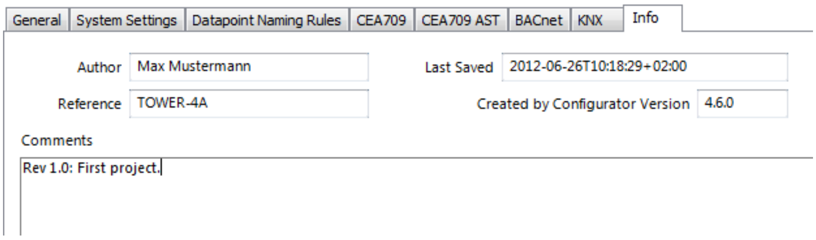


Figure 154: Info tab in project setting.

7.4 Basic Workflows

7.4.1 Replace a Device

A device can be replaced in the network by another unit. This might be necessary if a hardware defect occurs. First of all, the replacement device needs to be configured with the appropriate IP settings. The remainder of this section focuses on restoring the device configuration from a backup file. The work flow is depicted in Figure 155.

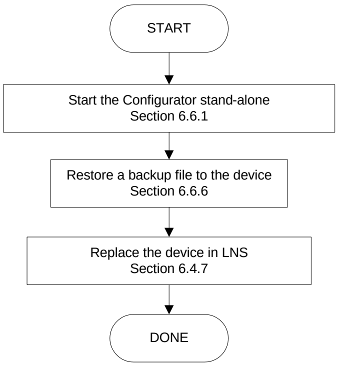


Figure 155: Basic work flow to configure a replacement device.

Start the Configurator software stand-alone and connect via the FTP method (see Section 7.6.1). Then restore the device configuration from the backup file, which has been created when the original device has been configured or modified (see Section 7.6.6). After the restore all data points, dynamic NVs and bindings, BACnet server objects and client mappings are restored. The device is again configured online and fully functional in the network.

If using an LNS-based tool, the device needs to be replaced in that tool at some later point in time (see Section 7.4.3) as the NID has changed. If you are not using LNS, then refer to your network management tool’s reference manual on how to replace a device.

7.4.2 Adding the Device to LNS

To configure a device in your LonMaker drawing, the device needs to be added to the LNS database and commissioned. This Section refers to LonMaker TE and describes how to add a device to your database. The example discusses a LINX-10X but it is general to all CEA-709 L-INX and L-GATE models

To Add a Device to LonMaker TE

1. In your LonMaker drawing, drag a device stencil into the drawing. Enter an appropriate name as shown in Figure 156.

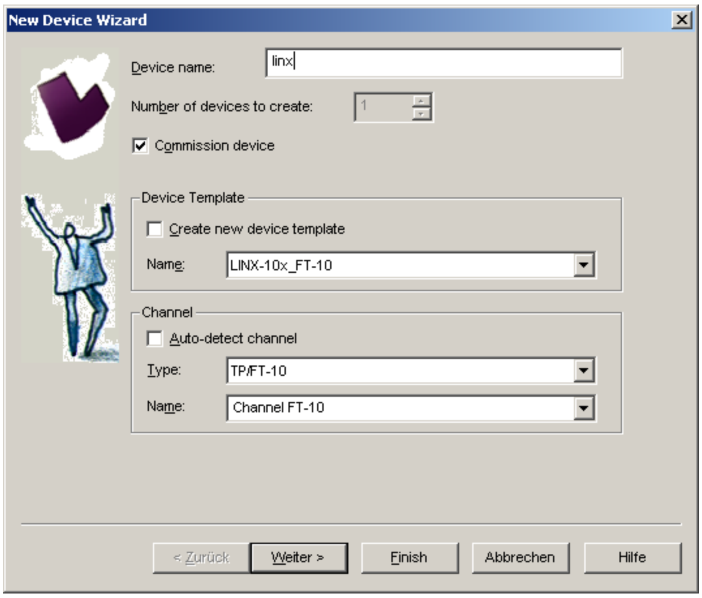


Figure 156: Create a new device in the drawing.

2. Select **Commission Device** if the LINX-10X is already connected to the network.

3. In the **Device Template** group box select the existing device template of the LINX. Select “LINX-xxx_FT-10”, if the L-INX is configured to use the FT-10 interface, or “LINX-xxx_IP-10L”, if the L-INX is configured to be on the IP channel. For information on how to configure which port to use, refer to Section 5.2.9 for the Web UI. Note that for the LINX-15x the LINX-12x XIF has to be used.

4. Select the channel, which the device is connected to and click **Next**.

5. The following dialog shown in Figure 157 appears, click **Next**.

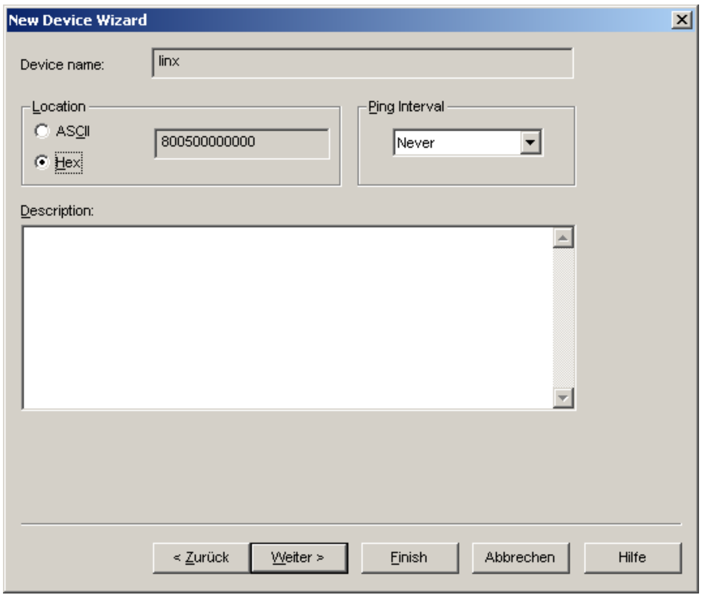


Figure 157: Leave defaults for Location.

6. Check Service Pin as the device identification method as shown in Figure 158 and click **Next**.

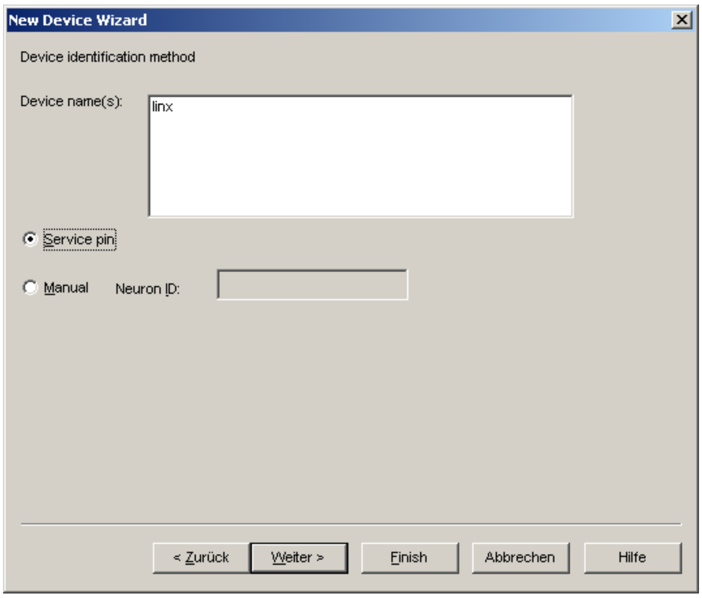


Figure 158: Use Service Pin.

7. Click **Next** in the following screens until you get to the final dialog shown in Figure 159.

8. If the device is already on-net, select **Online**.

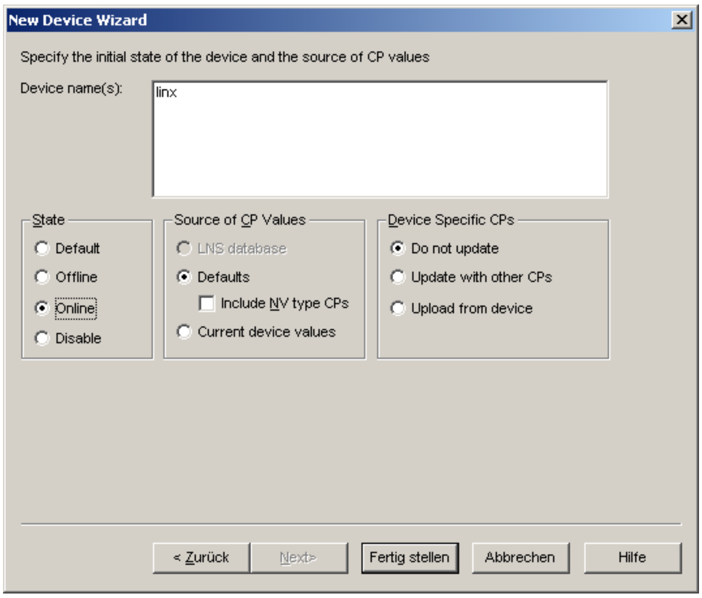
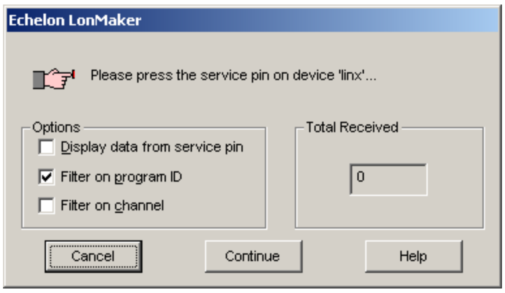


Figure 159: Final dialog.

9. Click **Finish**. A dialog will prompt to press the service pin.



10. Finally, you should get the device added to your drawing as depicted in Figure 160.

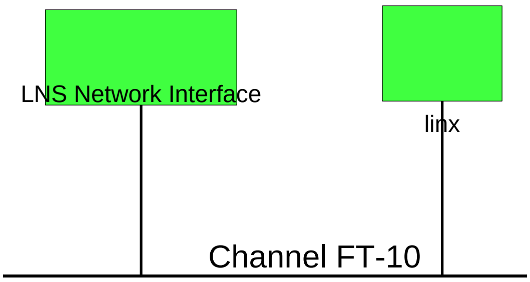


Figure 160: The L-INX has been added to the drawing.

7.4.3 Replace a Device in LNS

This Section describes how to replace a device in your LNS database. The example discusses a LINX-10X but is general to all CEA-709 L-INX and L-GATE models. The description refers to LonMaker TE. Let’s assume there is a device ‘linx’ in the LNS database as shown in Figure 161.

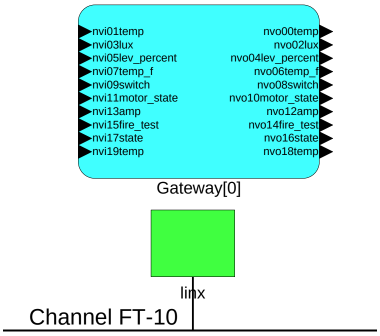


Figure 161: LonMaker drawing with one L-INX.

To Replace a Device in LonMaker TE

1. Select the device and right-click on the device shape.
2. Select **Commissioning → Replace…**. This opens the LonMaker Replace Device Wizard as shown in Figure 162.

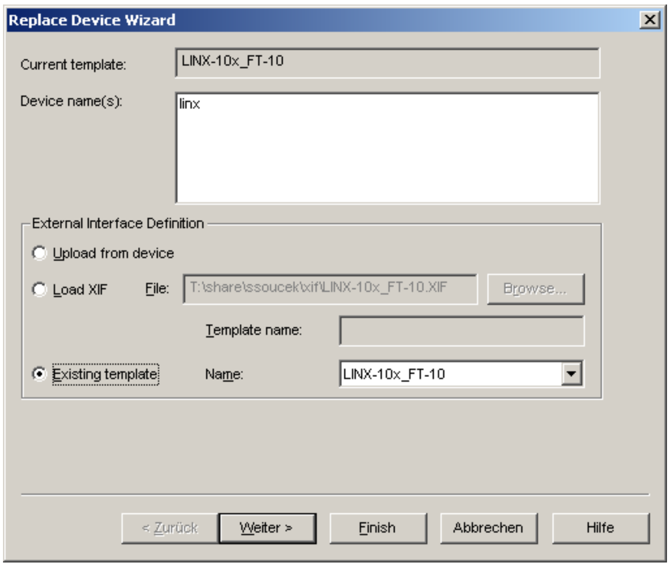


Figure 162: LonMaker replace device wizard.

3. Choose the existing device template and click **Next**.
4. In the following window shown in Figure 163 click **Next**.

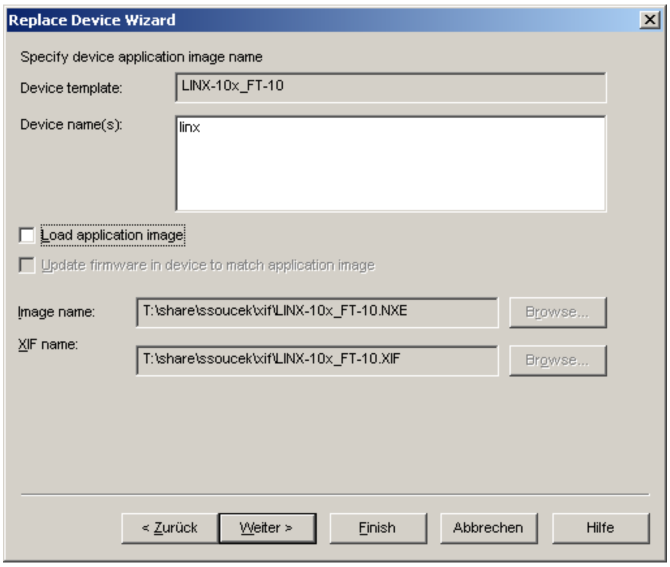


Figure 163: Click Next without loading an application image.

5. Then select **Online** as shown in Figure 164 and click **Next**.

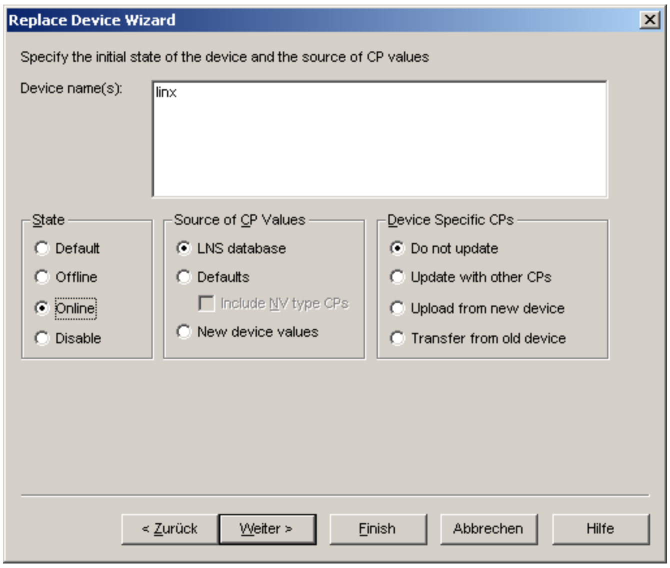


Figure 164: Select online state.

6. Select the **Service pin** method and click on **Finish** as shown in Figure 165.

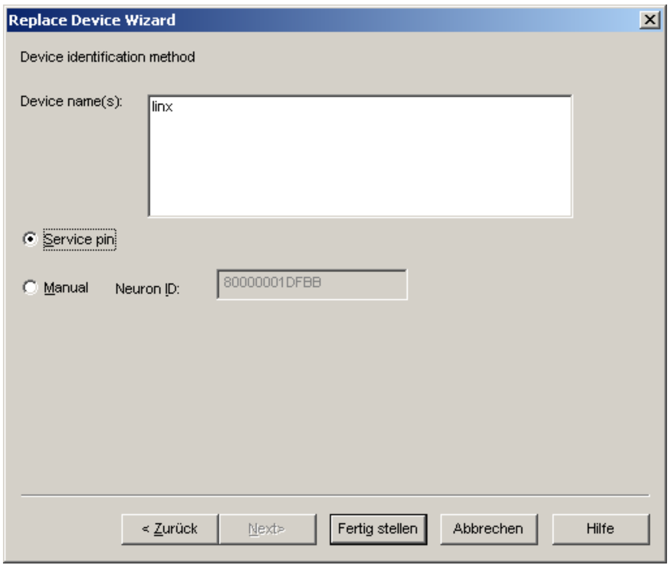


Figure 165: Select Service Pin and click Finish.

7. Then the service pin requestor opens as shown in Figure 166. Press the service pin on the replacement device on the correct port. You can also send the service pin using the Web interface (see Section 5.1).

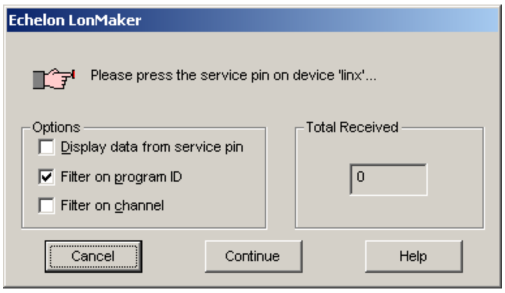


Figure 166: Wait for the service pin from the device.

8. After the service pin has been received, LonMaker commissions the replacement device, creates the dynamic NVs again (if any), and installs the bindings.

7.5 Model-Specific Workflows

7.5.1 L-INX Workflows for CEA-709

This section discusses a number of work flows for configuring a CEA-709 L-INX in different use cases in addition to the simple use case in the quick-start scenario (see Chapter 3). The description is intended to be high-level and is depicted in flow diagrams. The individual steps refer to later sections, which describe each step in more detail. In principle, the L-INX Configurator supports the following use cases:

- Network Management Tool based on LNS 3.x (see Section 7.5.1.2)
- Non-LNS 3.x network management tool with polling (see Section 7.5.1.3)
- Non-LNS 3.x network management tool with bindings (see Section 7.5.1.4)

7.5.1.1 Involved Configuration Files

In the configuration process, there are a number of files involved:

- XIF file: This is the standard file format to exchange the static interface of a device. This file can be used to create a device in the database without having the L-INX or L-GATE on-line. There exists a standard XIF file for the FT port (e.g., LINX-10x_FT-10.xif) and one for the IP-852 port (e.g., LINX-10x_IP-10L.xif). For the LINX-15x model use the LINX-12x XIF files.
- L-INX Configurator project file: This file contains all ports, data points, and connections of a project. These files end with ".linx0", ".linx1", ".linx2", or ".gtw". It stores all relevant configuration data and is intended to be saved on a PC to backup the device's data point configuration.

7.5.1.2 Configure with LNS

The flow diagram in Figure 167 shows the steps that need to be followed in order to configure the L-INX in a network with LNS 3.x. In this scenario, the L-INX will use dynamic NVs and bindings.

First, the L-INX device must be added to LNS (see Section 7.4.2). Then the L-INX Configurator must be started in plug-in mode to configure the device (see Section 7.7.1). In the Configurator, scan for the data points in the LNS database (see Section 7.7.2). Select the data points that the L-INX shall expose (see Section 7.7.5). Finally, the configuration needs to be downloaded to the L-INX via LNS (see Section 7.7.9). It is recommended to backup the device configuration to a file for being able to replace the device in the network (see Section 7.6.6).

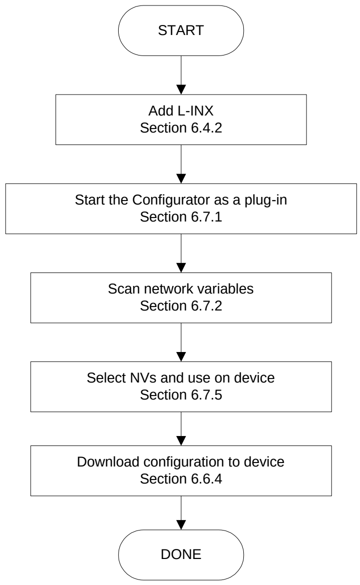


Figure 167: Basic design-flow with LNS.

To add more NVs when all bindings are in place and the L-INX is already being used, simply repeat the steps described above. The Configurator software will back up the bindings, create or delete the dynamic NVs, and re-create the bindings again.

7.5.1.3 Configure without LNS

The flow diagram in Figure 168 shows the steps that need to be followed in order to configure the device without LNS 3.x. In this scenario the device will use external NVs and polling. The advantage of this solution is that no bindings in the non-LNS tool (or self-binding nodes) need to be changed. This comes at the cost of a constant network load caused by polling.

Start the Configurator in stand-alone mode and connect to the device via the FTP method (see Section 7.6.1). If changing an existing configuration, upload the current configuration from the device (see Section 7.6.2). In the Configurator, import data points from a CSV import file (see Section 7.7.3) or from an XIF file (see Section 7.8.1). If the other devices are already connected to the network you may also scan them online (see Section 7.7.4). Select the data points that the device shall expose (see Section 7.7.5). Alternatively, you can create external NVs manually (see Section 7.7.8). Finally, the configuration needs to be downloaded to the device (see Section 7.6.4). It is recommended to backup the device configuration to a file for being able to replace the device in the network (see Section 7.6.6).

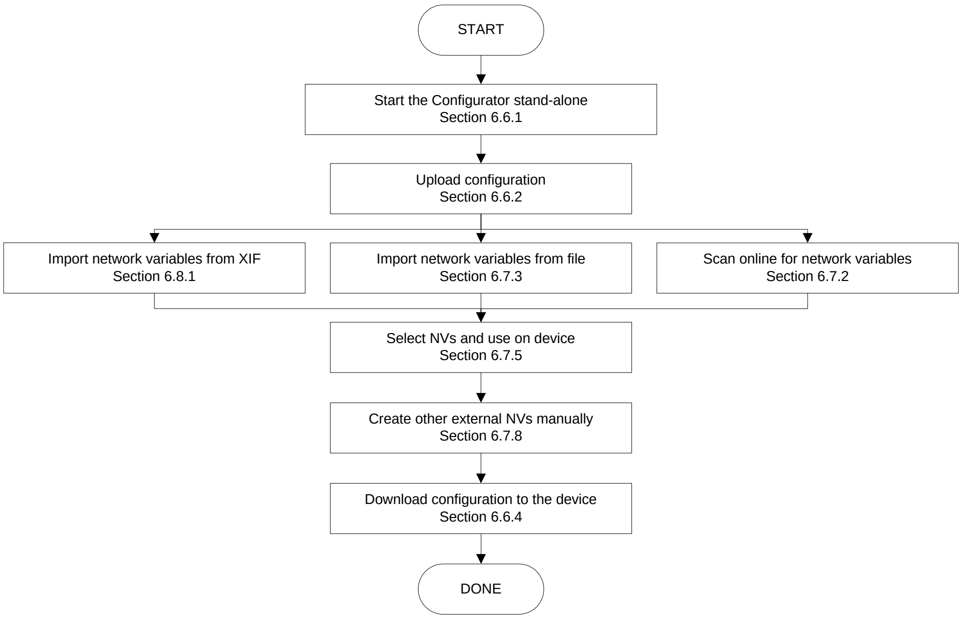


Figure 168: Basic design-flow without LNS.

7.5.1.4 Configure without LNS Using Bindings

The flow diagram in Figure 169 shows the steps that need to be followed in order to configure the device without LNS 3.x. In this scenario the device will use static NVs and bindings. The advantage of this solution is that the network load is minimized. However, the non-LNS management tool must create bindings for the device and update an existing network.

Start the Configurator in stand-alone mode and connect to the device via the FTP method (see Section 7.6.1). In the Configurator import data points from a CSV import file (see Section 7.7.3) or from an XIF file (see Section 7.8.1). If the other devices are already connected to the network you may also scan them (see Section 7.7.4). Select the data points that the device shall expose (see Section 7.7.5). For the NVs used on the device select the “static NV” allocation type (see Section 7.7.6). Alternatively, you can create static NVs manually (see Section 7.7.7).

For network management tools, which do not support the ECS (enhanced command set) network management commands, the legacy network management mode must be configured (see Section 7.7.10). Please contact the tool’s vendor for information whether ECS is supported or not.

Download the configuration onto the device (see Section 7.6.4). Finally, export a XIF file (see Section 7.7.11). It is recommended to backup the device configuration to a file for being able to replace the device in the network (see Section 7.6.6).

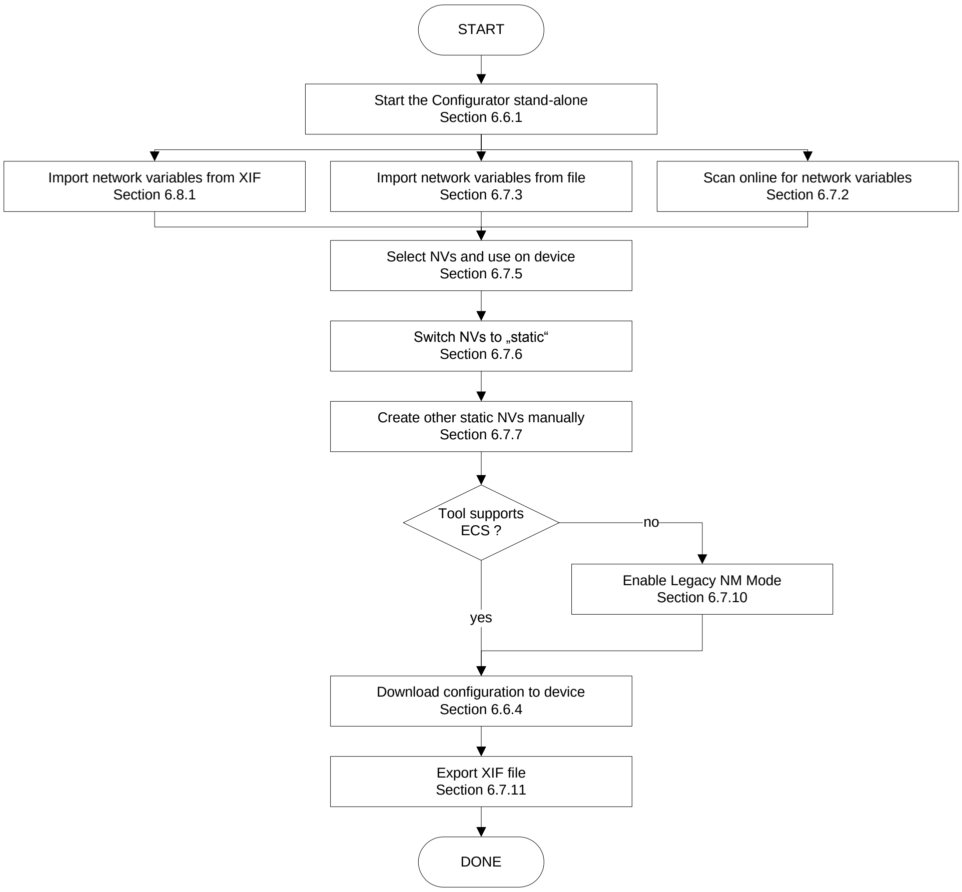


Figure 169: Basic design-flow without LNS using bindings.

To use the device in the non-LNS management tool, commission the device using the exported XIF file and create the bindings.

When changing a running device configuration with existing bindings, it is recommended to create additional data points as external NVs with polling as described in Section 7.5.1.3. Otherwise, depending on the third-party tool, a new XIF file may be required to be exported for replacing the device in the non-LNS tool. In this case the user would need to create all bindings again from scratch (see Section 6.6.4).

7.5.2 L-INX Workflows for BACnet

This section discusses a number of work flows for configuring a BACnet L-INX in different use cases in addition to the simple use case in the quick-start scenario (see Chapter 3). The description is intended to be high-level and is depicted in flow diagrams. The individual steps refer to later sections, which describe each step in more detail.

7.5.2.1 Involved Configuration Files

In the configuration process, there are a number of files involved:

- L-INX Configurator project file: This file contains all ports, data points, and connections of a project. These files end with “.linx0”, “.linx1”, or “.linx2”. It stores all relevant configuration data and is intended to be saved on a PC to backup the L-INX’s data point configuration.
- EDE file: When engineering offline, the Configurator can import remote BACnet data points via an EDE file. Out of this information client mappings are created.

7.5.2.2 Engineer Online

The flow diagram in Figure 170 shows the steps on how to configure the BACnet port when being on-line. In this case, the device must be present in the BACnet network and configured with an IP address. The user can connect to the device and scan for existing BACnet devices and objects on the network.

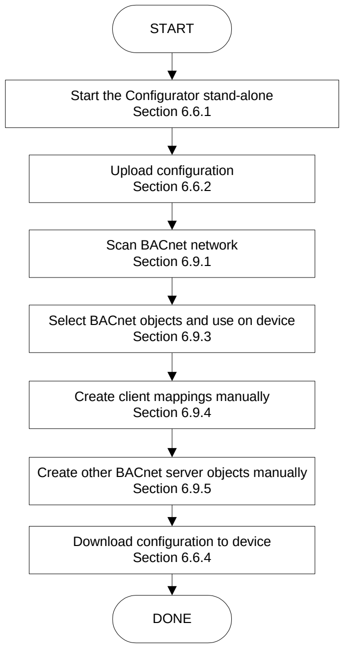


Figure 170: Basic work flow to engineer on-line.

Start the Configurator in stand-alone mode and connect to the device via the FTP method (see Section 7.6.1). If changing an existing configuration, upload the current configuration from the device (see Section 7.6.2). In the Configurator, start an on-line network scan to discover devices and BACnet objects (see Section 7.9.1). Select the data points that the device shall expose (see Section 7.9.3). Alternatively, you can create client mappings (see Section 7.9.4) and local BACnet server objects (see Section 7.9.5) manually. Finally, the configuration needs to be downloaded to the device (see Section 7.6.4). It is recommended to backup the device configuration to a file for being able to replace the device in the network (see Section 7.6.6).

7.5.2.3 Engineer Offline

The flow diagram in Figure 171 shows the steps on how to configure the BACnet port when being off-line. In this case, the device doesn't need to be present in the BACnet network. The systems integrator can engineer the BACnet port and download the configuration at a later point in time.

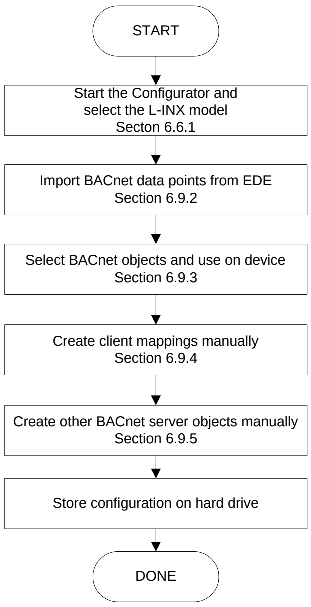


Figure 171: Basic work flow to engineer off-line.

Start the Configurator in stand-alone mode and select the appropriate L-INX model in the menu **Model** (see Section 7.6.1). In the Configurator, import external BACnet data points from an EDE file (see Section 7.9.2). Select the data points that the device shall expose (see Section 7.9.3). Alternatively, you can create client mappings (see Section 7.9.4) and local BACnet server objects (see Section 7.9.5) manually. When finished store the configuration on the hard drive and download later to the device (see Section 7.6.4).

7.5.3 L-GATE Workflows

This section discusses a number of work flows for configuring the L-GATE in different use cases in addition to the simple use case in the quick-start scenario (see Section 3.4). The description is intended to be high-level and is depicted in a flow diagram. The individual steps refer to later Sections, which describe each step in more detail. In principle, the Configurator supports the following use cases:

- Network Management Tool based on LNS 3.x (see Section 7.5.3.2)
- Non-LNS 3.x network management tool with polling (see Section 7.5.3.3)
- Non-LNS 3.x network management tool with bindings (see Section 7.5.3.4)

7.5.3.1 Involved Configuration Files

In the configuration process, there are a number of files involved:

- XIF file: This is the standard file format to exchange the static interface of a device. This file can be used to create a device in the database without having the L-GATE on-line. There exists a standard XIF file for the FT port (LGATE-900 FT-10.xif) and one for the IP-852 port (LGATE-900 IP-10L.xif).
- Configurator project file for L-GATE: This file contains all ports, data points, and connections of a project. These files end with ".gtw". It stores all relevant configuration data and is intended to be saved on a PC to backup the L-GATE's data point configuration.

7.5.3.2 Configure with LNS

The flow diagram in Figure 172 shows the steps that need to be followed in order to configure the L-GATE in a network with LNS 3.x. In this scenario the L-GATE will use dynamic NVs and bindings.

First, the L-GATE device must be added to LNS (see Section 7.4.2). Then the Configurator must be started in plug-in mode to configure the L-GATE (see Section 7.7.1). In the Configurator scan for the data points in the LNS database (see Section 7.7.2). Select the NVs that the L-GATE shall expose to BACnet (see Section 7.7.5). Generate BACnet objects and connections from the used NVs (see Section 7.10.8). Finally, the configuration needs to be downloaded onto the L-GATE (see Section 7.7.9). It is recommended to save the complete configuration to a disk file for being able to replace an L-GATE in the network.

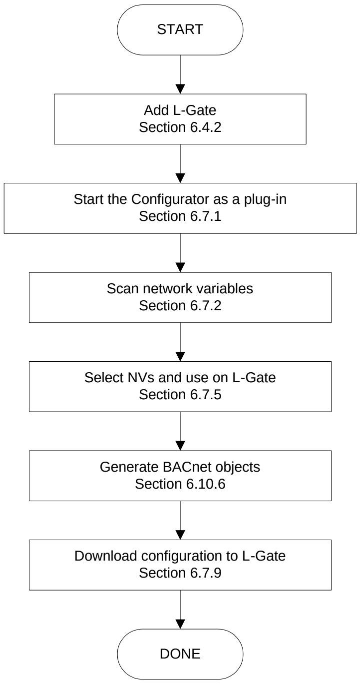


Figure 172: Basic L-GATE workflow with LNS.

To add more NVs when all bindings are in place and the L-GATE is being used simply repeat the steps described above. The Configurator software will back up the bindings, create or delete the dynamic NVs, and re-create the bindings again.

7.5.3.3 Configure without LNS

The flow diagram in Figure 173 shows the steps that need to be followed in order to configure the L-GATE without LNS 3.x. In this scenario the L-GATE will use external NVs and polling. The advantage of this solution is that no bindings in the non-LNS tool (or self-binding nodes) need to be changed. This comes at the cost of a constant network load caused by polling.

Start the Configurator in stand-alone mode and connect to the L-GATE over IP (see Section 7.6.1). If changing an existing configuration upload the current configuration from the L-GATE (see Section 7.6.2). In the Configurator import data points from a CSV import file (see Section 7.7.3) or scan an CEA-709 network online (see Section 7.7.4). Select the NVs that the L-GATE shall expose to BACnet (see Section 7.7.5). Alternatively, you can create external NVs manually (see Section 7.7.8). Generate BACnet objects and connections from the used NVs (see Section 7.10.8). Finally, the configuration needs to be downloaded onto the L-GATE (see Section 7.6.4). It is recommended to save the complete configuration to a disk file for being able to replace an L-GATE in the network.

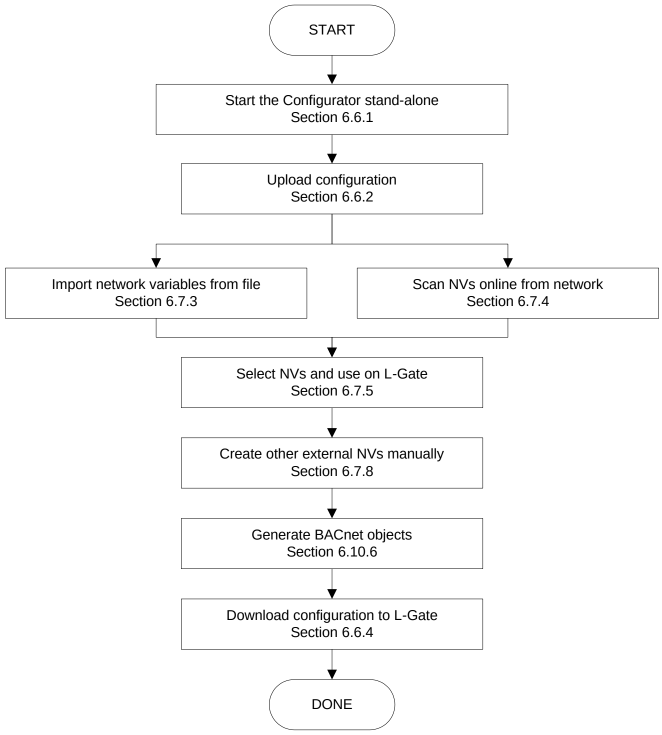


Figure 173: Basic L-GATE workflow without LNS.

7.5.3.4 Configure without LNS Using Bindings

The flow diagram in Figure 174 shows the steps that need to be followed in order to configure the L-GATE without LNS 3.x. In this scenario the L-GATE will use static NVs and bindings. The advantage of this solution is that the network load is minimized. However, the non-LNS management tool must create bindings for the L-GATE and update an existing network.

Start the Configurator in stand-alone mode and connect to the L-GATE via the FTP method (see Section 7.6.1). In the Configurator import data points from a CSV import file (see Section 7.7.3) or scan a CEA-709 network online (see Section 7.7.4). Select the NVs that the L-GATE shall expose to BACnet (see Section 7.7.5). For the NVs used on the L-GATE select the “static NV” allocation type (see Section 7.7.6). Alternatively, you can create static NVs manually (see Section 7.7.7).

For network management tools, which do not support the ECS (enhanced command set) network management commands, the legacy network management mode must be configured (see Section 7.7.10). Please contact the tool’s vendor for information whether ECS is supported or not.

Generate BACnet objects and connections from the used NVs (see Section 7.10.8). Download the configuration onto the L-GATE (see Section 7.6.4). Finally, export a XIF file (see Section 7.7.11). It is recommended to save the complete configuration to a disk file for being able to replace an L-GATE in the network.

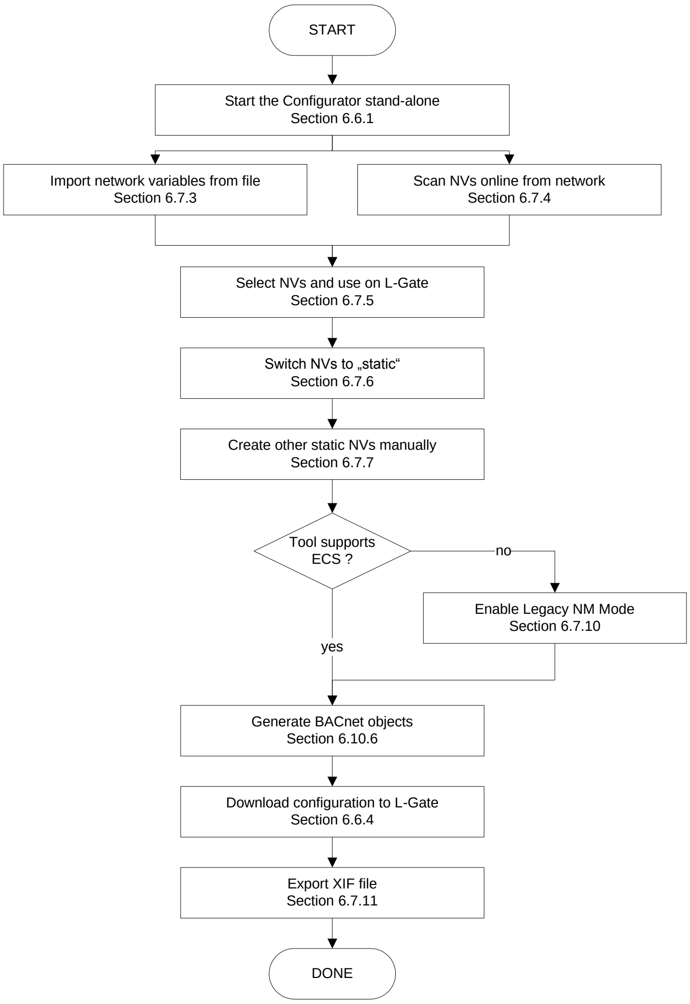


Figure 174: Basic L-GATE workflow without LNS using bindings.

To use the L-GATE in the non-LNS management tool, commission the L-GATE using the exported XIF file and create the bindings.

When changing a running L-GATE configuration with existing bindings, it is recommended to create additional data points as external NVs with polling as described in Section 7.5.3.3. Otherwise, a new XIF file needs to be exported and replacing the L-GATE in the non-LNS tool requires the user to create all bindings again from scratch (see Section 6.6.4).

7.6 Using the L-INX Configurator

7.6.1 Starting Stand-Alone

Go to the Windows **Start** menu, select **Programs**, **LOYTEC LINX Configurator** and then click on **LOYTEC LINX Configurator**. This starts the Configurator and the main window with the data point tab is displayed.

If the L-INX is not yet connected to the network, go to the **Model** menu and select the L-INX model to be configured. If the L-INX is already connected to the network it is recommended to connect the Configurator online to the L-INX.

To Connect to a L-INX Stand-Alone

1. Select the direct connection method by clicking on the **Connect to device** button

   in the tool bar of the main window. The connect dialog as shown in Figure 175 opens containing the managed device connection templates.

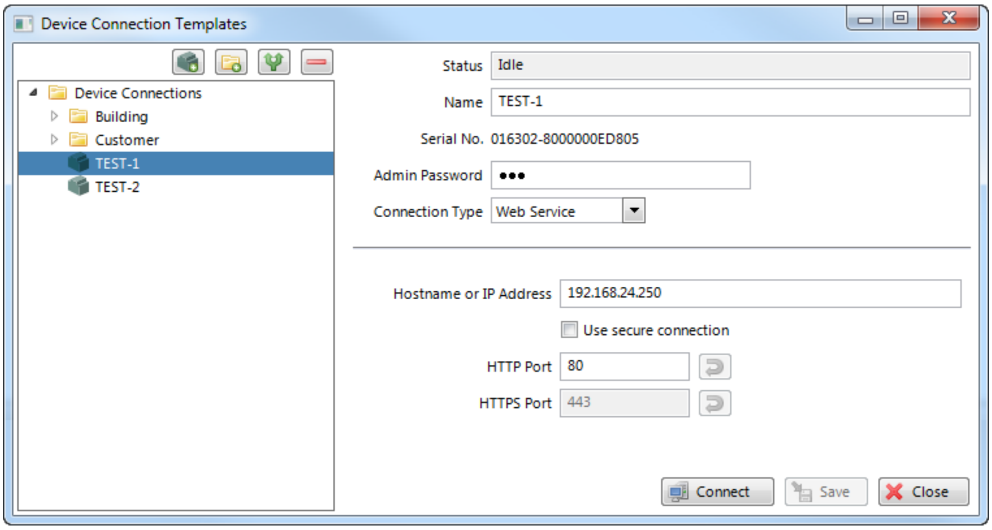


Figure 175: Connection dialog.

2. To add a new device connection, click on the **New Device Connection** button or select an existing connection in the tree on the left-hand side and click on the **Duplicate** button.

3. Enter a name for the connection.

4. Enter the admin password. The default password is ‘loytec4u’ (older firmware versions used ‘admin’).

5. Choose a **Connection Type**. Possible options are:

   a. **Web service** (recommended): Firewall-friendly connection using HTTP or HTTPS.

   b. **TCP/IP**: This uses the IP protocols FTP and Telnet or SSH to connect to the device.

   c. **CEA-709 (NIC)**: Connection via a LOYTEC CEA-709 network interface.

6. For IP-based connection methods enter the host name or IP address of the device. Optionally, click on **Use secure connection** to enable encrypted SSH or HTTPS access to the device.

7. If your device is located behind a NAT router or firewall, you may change the FTP, Telnet, SSH, HTTP and HTTPS ports to your needs for accessing the device.

8. If the CEA-709 connection method is used, enter the CEA-709 address information and choose a network interface.

9. Click on **Save** to store that connection.

Note: *If you connect without having the connection settings saved, a dialog asks whether to use the changed settings temporarily for this connection only. In this case the existing connection is not altered.*

---

10. Organize device connections into folders. To add a new folder, click on the **New Folder** button  and enter a folder name. Drag the new device connection onto the new folder.

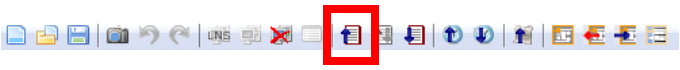
11. Click on **Connect**. This establishes the connection to the device.

7.6.2 Uploading the Configuration

To get the current data point configuration of the device, the configuration needs to be uploaded. This will upload the entire configuration from the device, including data points, NVs, registers, client mappings, schedules, etc.

To Upload a Configuration

1. Click on the **Upload Configuration** button



in the tool bar. The configuration upload dialog opens up as shown in Figure 176.

2. If the check box **Automatically sync local dynamic NVs** is enabled on a CEA-709 device, the Configurator uploads any manually created dynamic NVs and merges them into the data point configuration.

3. Click on the button **Start** to start the transfer. This will upload the configuration of all ports.

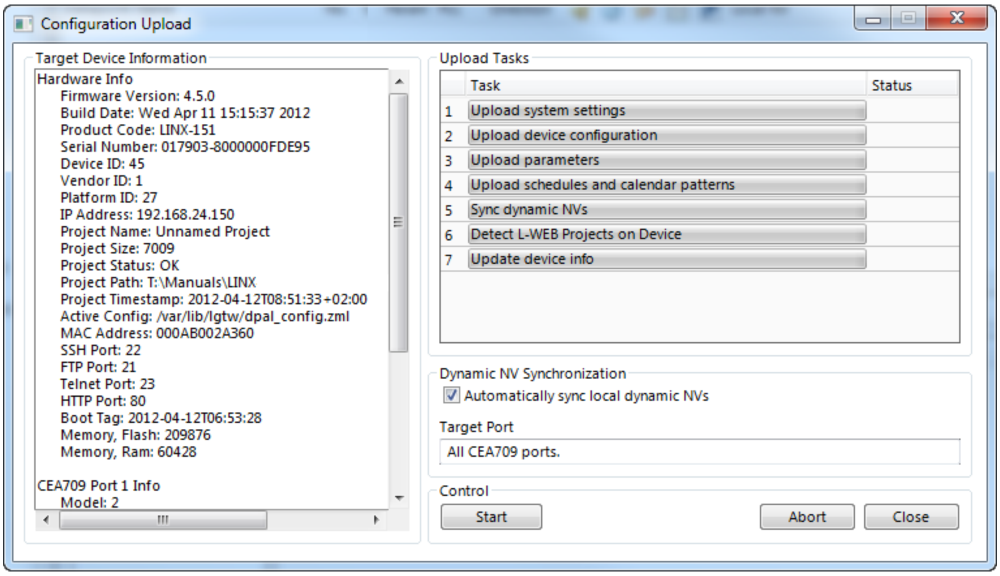


Figure 176: Configuration upload dialog.

4. When the project settings of the configuration being uploaded specify to ask, which specific parameters shall be uploaded, check the needed information and proceed.

5. If parameters are selected to be uploaded and their values have changed on the device, the parameter merge dialog appears. For resolving the reported conflicts refer to Section 6.2.5.

6. If dynamic NVs were synchronized, click on **Finish**.

7.6.3 Create User Registers

User registers are data points on the device that do not have a representation on the network. Thus, they are not accessible over a specific technology. A register merely serves as a container for intermediate data (e.g., results of math objects). Since a register has no network direction, it can be written and read. Therefore, it is created as a value data point. For back-ward compatibility, it is still possible to generate two data points for each register, one for writing the register (output) and one for reading the register (input).

To Create a User Register

1. Select the **User Registers** folder under the device folder.

2. Right-click in the data point list and select **New Datapoint…** in the context menu. This opens the register creation dialog as shown in Figure 177.

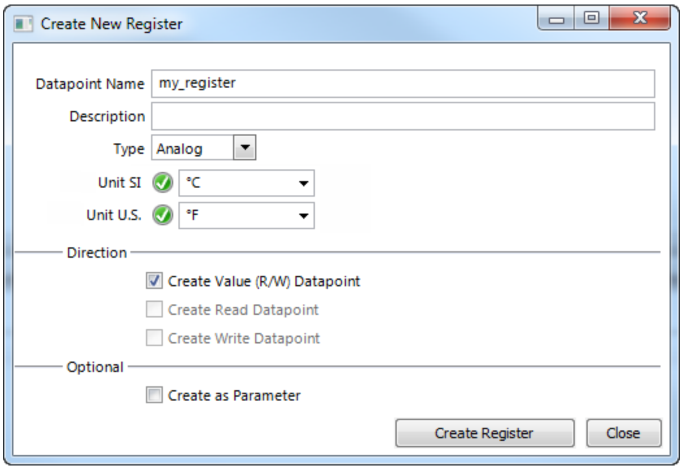


Figure 177: Create a user register.

3. Enter a **Datapoint Name** for the register. You may leave the **Register Name** blank to give the underlying register the same name as the data point.

4. For analog registers choose appropriate units for the metric (SI) and U.S. unit systems. To specify a non-convertible unit type in the desired unit as a free text.

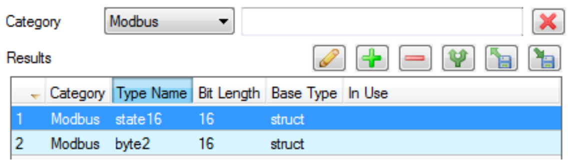
5. The standard setting is a value data point for read/write. Optionally, deselect the value option and select the read data point or write data point check box. This is necessary when configuring for an older firmware version.

6. Select **Create as Parameter** if needed. In this case, the register will be a persistent parameter. It can be done later in the data point properties also.

7. Select a **Type**. Available are “Analog”, “Binary”, “Multistate”, “String”, “Variant” or “User”.

8. When selecting **User**, a register with a user-defined structure is going to be created. Click on ... next to **Custom Type** in order to select a structure type.

9. In the dialog **Select Custom Type** choose the type **Category** in the drop-down list and the desired user type. Then click the button **Select**.



10. Click Create Register.

11. If needed create additional registers from the dialog. Finally, click **Cancel** to exit the dialog.

7.6.4 Configuration Download

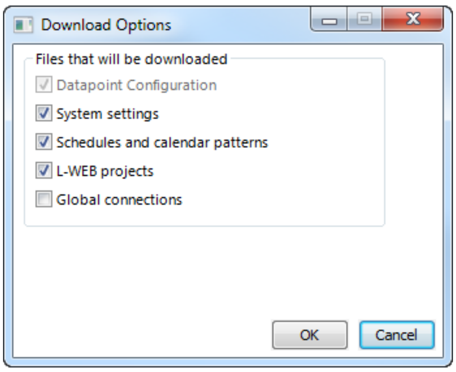
After the data points have been configured, the configuration needs to be downloaded to the device. For doing so, the device must be online. If the device is not yet connected to the network, the configuration can be saved to a project file on the local hard drive.

To Download a Configuration

1. Click on the **Download Configuration** speed button

in the tool bar of the main window. This will open the configuration download dialog as shown in Figure 178.

2. If the project settings specify to ask (see Section 7.3.1), which specific parameters shall be downloaded, check all that apply and click **Ok**.



3. Click **Start** to start the download. Each of the actions is displayed in the **Task List** section of the dialog. The current progress is indicated by the progress bar below.

4. If parameters are selected to be downloaded and their values have changed on the device, the parameter merge dialog appears. For resolving the reported conflicts refer to Section 6.2.5.

5. When the download process has finished, a notification window appears, which has to be acknowledged by clicking **OK**.

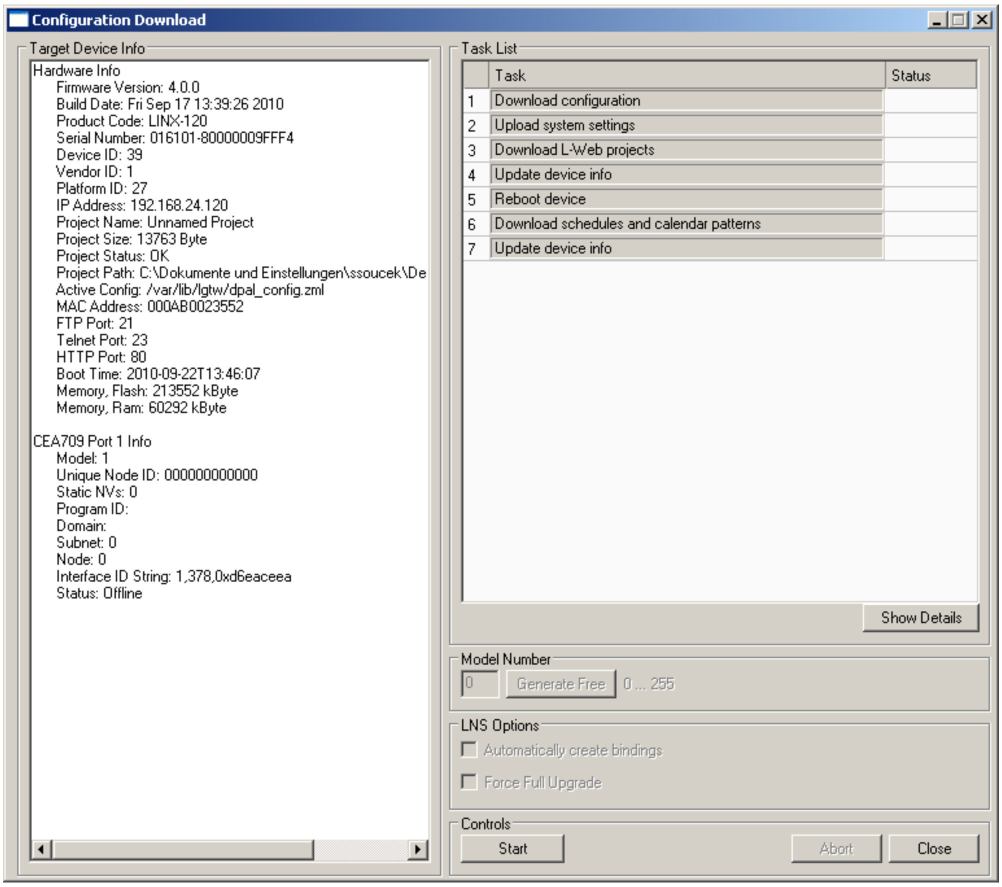


Figure 178: Configuration Download Dialog.

7.6.5 Upload the System Log

The system log on the device contains important log messages. Log messages are generated for important operational states (e.g., last boot time, last shutdown reason) or errors at run-time. This file is important for trouble-shooting and is available on the Web UI (see Section 5.5.1). The file can also be uploaded from the device with the Configurator.

To Upload the System Log

1. Connect to the device (see Section 7.6.1).

2. Click on the **Upload system log** button

in the tool bar. The upload system log dialog as shown in Figure 179 opens showing the upload progress.

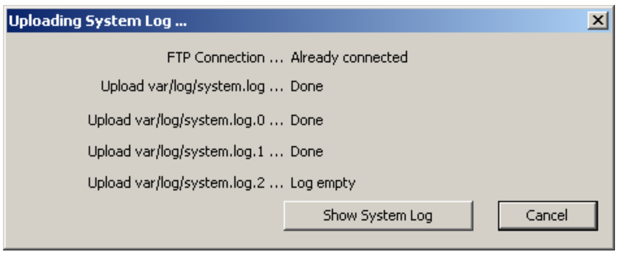


Figure 179: Upload system log dialog.

3. When the upload is finished, click on **Show System Log**. The system log window appears as shown in Figure 180.

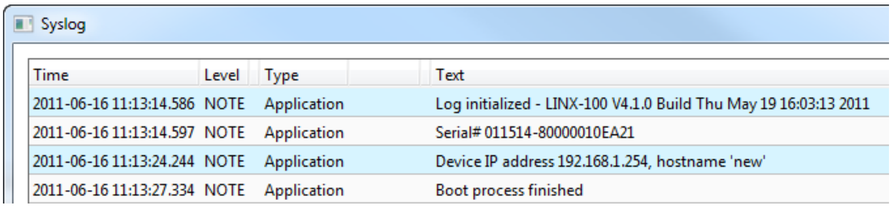


Figure 180: System log window.

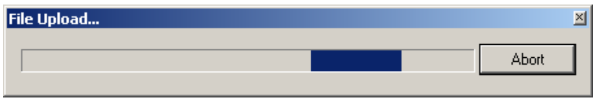
4. Click on **Save** to store the system log into a file on your local hard drive.

7.6.6 Backup and Restore

The Configurator provides a backup and restore function for the connected device. It is highly recommended to create a device backup once the device configuration has been completed. This backup can be used in the case a device needs to be replaced in the network.

To Create a Backup

1. Connect to the device (see Section 7.6.1).
2. Choose the menu Tools → Backup Device Configuration ….
3. A file requestor opens. Choose a location to store the ZIP file of the device backup. The suggested file name contains device IP address and creation date.
4. Click on **Save**. The backup is being uploaded.



To Restore a Backup

1. Choose the menu Tools → Restore Device Configuration ….
2. In the file requestor choose a backup ZIP file and click **Open**.
3. The Configurator restores and reboots the device. The process is complete when the device has finished rebooting.

7.6.7 Create Projects for SI and U.S. Units

Project engineering in the Configurator can be started using the unit system of choice, either metric (SI) or U.S. From that point on, all values in the Configurator are displayed in the chosen unit system.

For analog data points with a convertible network unit, the Configurator provides two unit representations, one to be used in the metric (SI) system, and one used in the U.S. system. The value is converted from the network unit to the unit of the chosen unit system. When creating analog data points, both unit representations are chosen from the network unit as appropriate. One can adapt these settings to the project’s needs.

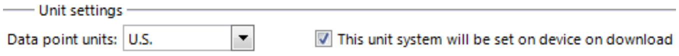
Since both unit representations can be specified, a project can be engineered to run in a metric (SI) and U.S. environment. Even when using a fixed data point configuration on the device, it can be switched to the desired unit system. All values on the device are processed in that chosen unit system, including Web UI, OPC server, parameter file, global connections, and programmable logic.

To Start a Project In U.S. Units

1. In the Configurator menu go to **Settings → Project settings …**

2. Click on the tab **General**.

3. Under **Unit setting** choose **U.S.** in the drop-down box. The check box **This unit system will be set on device on download** is set as a default. This ensures the device will run on the chosen unit system.

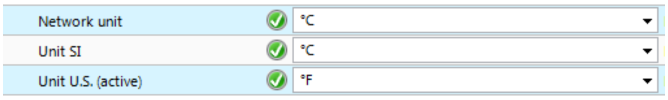


4. Click **OK.**

To Create Data Points in U.S. Units

1. Create a data point and choose the appropriate U.S. unit.

2. On existing analog data points, choose the appropriate U.S. unit in the data point properties. Compatible units to the network unit are shown first. The active unit system is indicated.



3. Enter all analog values such as default, min/max values, etc. in U.S. units. The unit is indicated next to the data point property.

To Create Projects For Both SI and U.S. Units

1. On the **General** tab of the **Project Settings** dialog, deselect the check box **This unit system will be set on device on download**. Choose your unit system of preference to be used in the Configurator for the project design.

2. Create data points and choose both appropriate units for metric (SI) and U.S. unit systems.

3. In program logic evaluate the system register **Unit System** in order to select appropriate sets of coefficients for regulators.

4. Download the data point configuration to the device. The device can be switched to the desired unit system using the Web UI or the system register **Unit System Set**.

Important!	***When changing the unit system, the device needs to be rebooted and will reset all persistent values to their default values converted to the chosen unit system.***

7.7 CEA-709 Configuration

7.7.1 Starting as an LNS Plug-In

In LonMaker the plug-in is started by right-clicking on the L-INX device shape or the Gateway/PLC functional block and selecting **Configure…** from the pop-up window.

In NL220, the Plug-in is started by right clicking on the L-INX node, then selecting the Option **LOYTEC LINX Configurator** in the **PlugIns** sub menu.

In Alex, the Plug-in is started by right clicking on the L-INX device and selecting the **LOYTEC LINX Configurator** in the **Starte PlugIn** sub menu.

A window similar to what is shown in Figure 181 should appear.

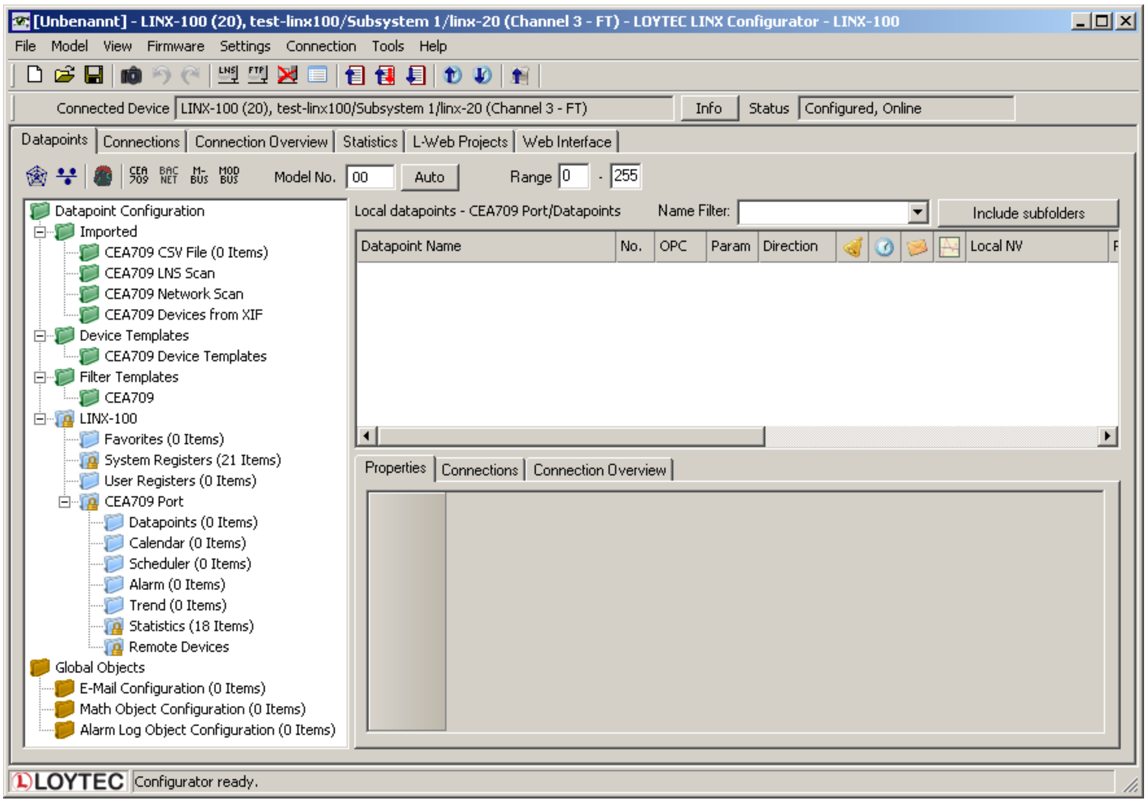


Figure 181: L-INX Configurator main window.

7.7.2 Scanning for Network Variables

When the L-INX Configurator is connected to an LNS database, network variables can be scanned from that data base.

To scan network variables from the LNS database

1. Click on the **Datapoints** tab of the main window.

2. Click on the button  **Scan channel**. This scans in all NVs on all devices connected to the CEA-709 channel of the device.

3. After the scan has completed, the folder **LNS Database Scan** is populated with the found NVs. Data point names for those NVs are automatically generated, following the data point naming rules defined in the project settings (see Section 7.3.2). By default

the name is generated from node name, object name, and NV name. These names are ensured to be unique by adding a counter for multiple occurrences of the same name.

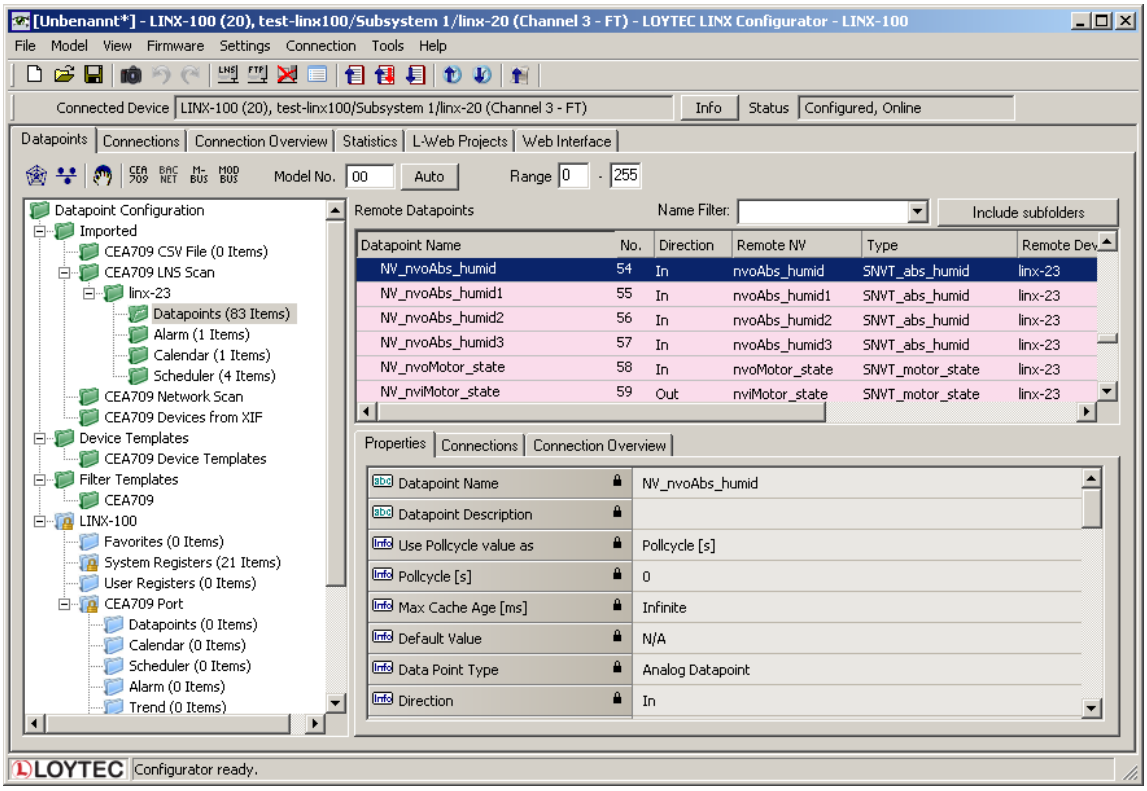


Figure 182: Scanned NVs in the LNS Database Scan Folder.

Figure 182 shows an example result of the database scan. The list can be sorted by each column. Selecting a line will display a number of associated properties in the property view below. Multiple items can be selected by using the <Ctrl> key and clicking with the mouse. All items can be selected by pressing <Ctrl-A>.

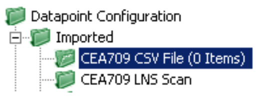
7.7.3 Importing Network Variables

Without LNS, the tool cannot connect to an LNS database, where it scans for network variables (NVs). Therefore, the list of NVs to be used on the device has to be available in a CSV file. This file can be produced by external software or created by hand. The CSV format for importing NVs is defined in Section 18.2.1.

To Import NVs from a File

1. Click on the **Datapoints** tab of the main window.

2. Select the import folder **CEA709 CSV File.**



3. Right-click and select **Import File**. In the following file selector dialog, choose the CSV import file and click **OK**.

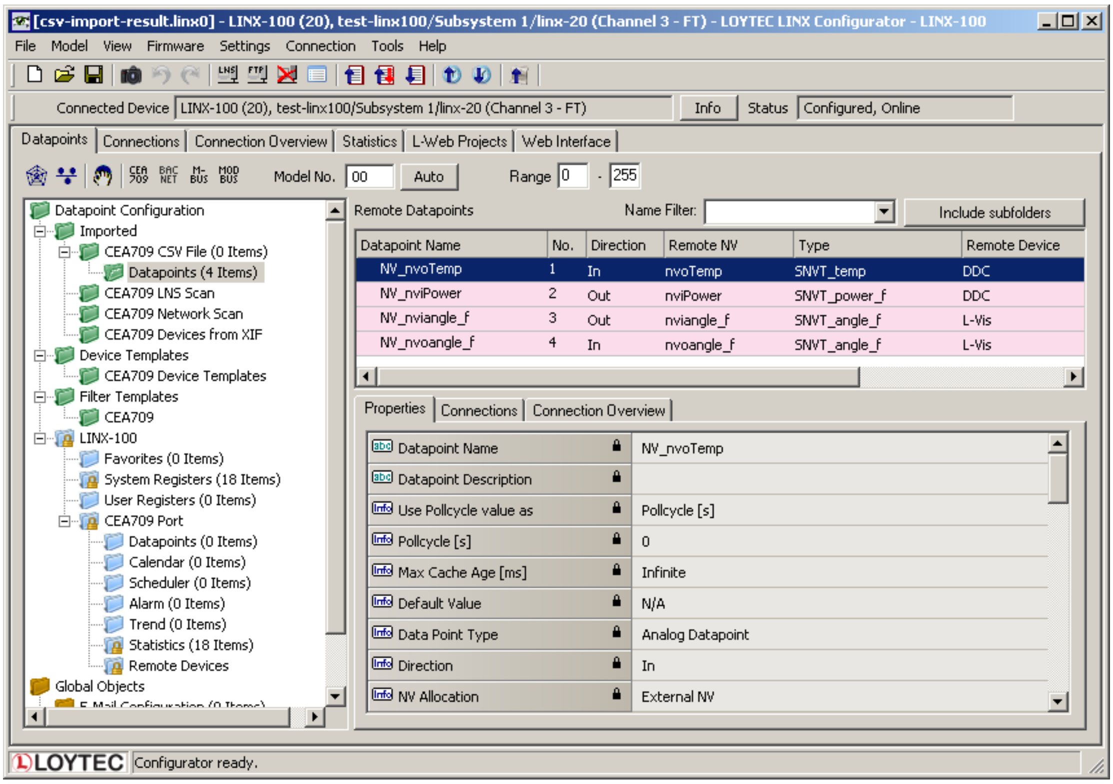


Figure 183: Imported NVs.

4. Now the CSV File folder is populated with the imported NVs as shown in Figure 183.

The list can be sorted by each column. Selecting a line will display a number of associated properties in the property view below. Multiple items can be selected by using the <Ctrl> key and clicking with the mouse. All items can be selected by pressing <Ctrl-A>.

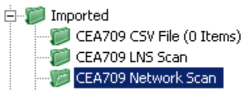
7.7.4 Scanning NVs online from the Network

LOYTEC devices also support an online network scan on the CEA-709 network. In this scan, the device searches for other devices on the CEA-709 network and pulls in NV information of these devices. These NVs can then be used instead of importing them from a CSV file.

To scan NVs online off the CEA-709 network

1. Click on the **Datapoints** tab of the main window.

2. Select the folder **CEA709 Network Scan**.



3. Right-click on that folder and select **Scan CEA709/852 Network…**. This opens the **CEA709 Management** dialog as shown in Figure 184.

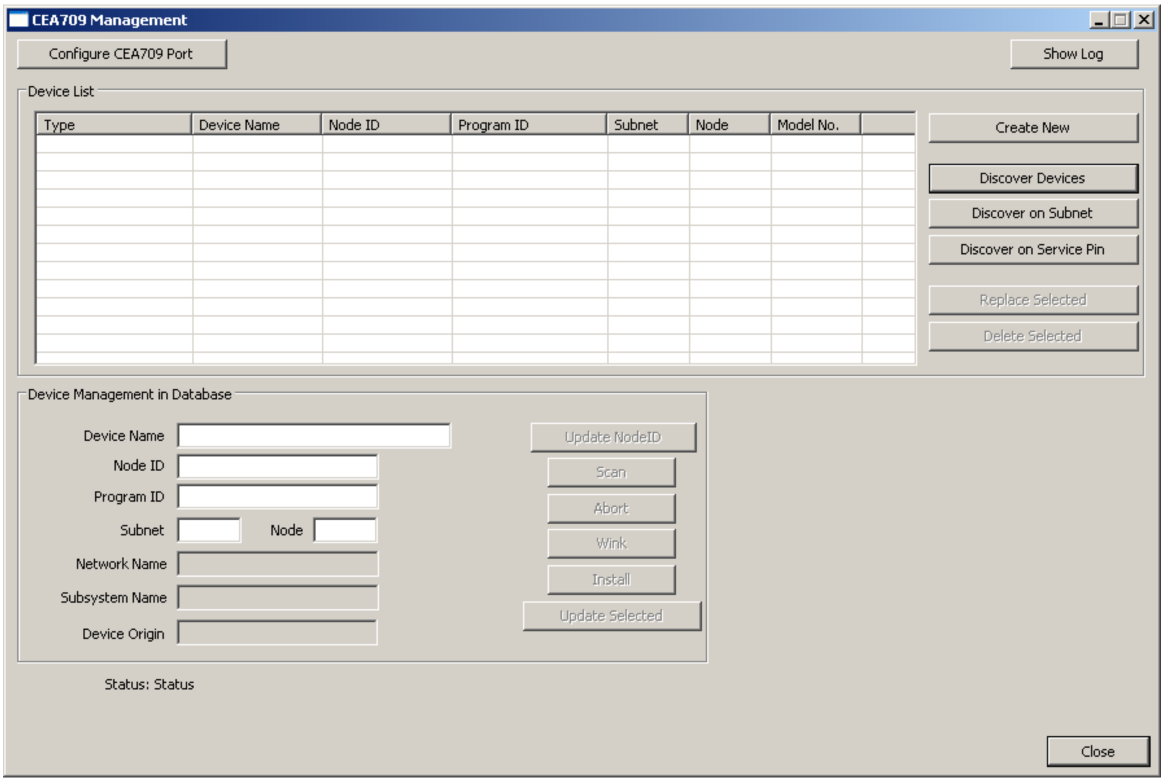


Figure 184: CEA-709 network scan dialog.

4. If the device has not been installed with a network management tool (e.g. LNS-based tool), press the **Configure CEA709 Port** button. This opens the device install dialog as shown in Figure 185.

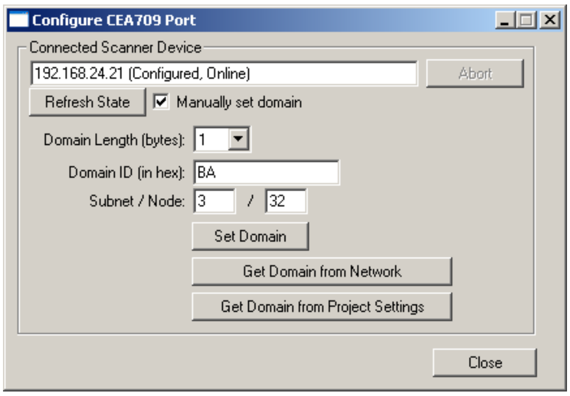


Figure 185: Configure CEA-709 port dialog.

5. Select the **Manually set domain** check-box and click the **Set** button. This sets the device configured, online to start the scan. Then click **Close**.

Note: *You need to set the same domain as the devices to be scanned. Click **Get Domain from Network** and press a service pin on some other, already installed device to retrieve the domain information before setting the device online.*

6. Click on the button **Discover Devices**. This starts a network scan. The results are put in the device list box.

7. Alternatively, click the button **Discover on Service Pin**. Then press the service pin of a particular device on the network. This device will be added to the device list.

8. Select a device in the device list. To give the device a usable name, enter the name in the **Device Name** field and click on the **Update Selected** button.

9. Then click the button **Scan**. This scans the NVs on the selected device and adds them to the CEA709/852 Network Scan folder as a separate sub-folder for the device as shown in Figure 186.

Tip! *If you are not sure which device you have selected, click on* ***Wink****. The selected device will execute its wink sequence.*

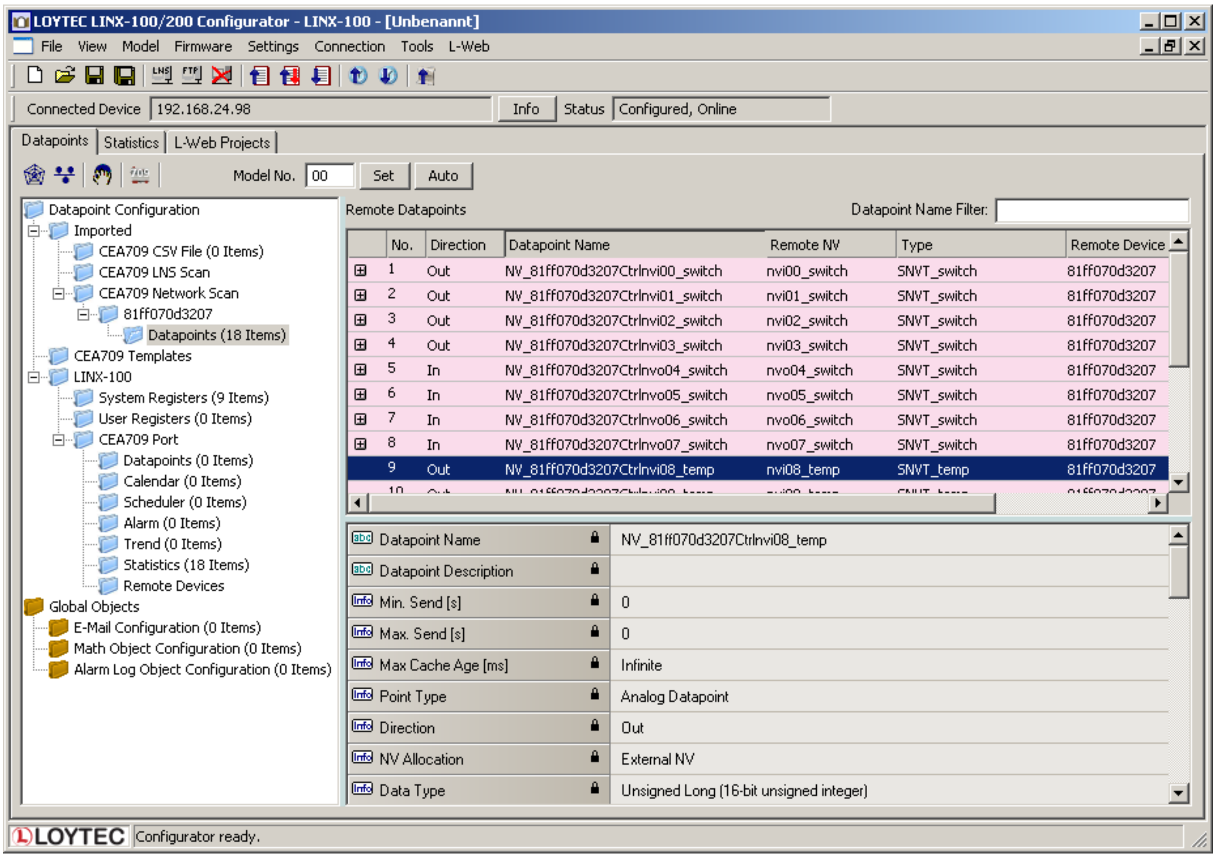


Figure 186: CEA-709 network scan results.

10. Click **Close** when all devices needed have been scanned.

7.7.5 Select and Use Network Variables

Data points in the **CEA709 LNS Scan** folder, the **CEA709 Network Scan** folder, the **CEA709 CSV File** folder, or in the **CEA709 Devices from XIF** folder can be selected for use on the device. Select those NVs, which shall be used on the device.

To Use NVs on the Device

1. Go to any of the CEA709 LNS Scan, CEA709 Network Scan, CEA709 CSV File, or the CEA709 Devices from XIF folder.

2. Use the multi-select feature by holding the *Shift* or *Ctrl* keys pressed.

3. Click on the button  **Use on Device** in the tool bar.

4. This creates data points in the CEA709 Port folder of the device. All data points in that folder will actually be created on the device after downloading the configuration.

Tip! *Data points can be edited by selecting a single point or using multi-select. The available properties to be edited are displayed in the property view below.*

7.7.6 Change the NV Allocation

After selecting the **Use on device** action on scanned or imported NVs, they are assigned a default NV allocation in the CEA709 port folder. This default allocation can be changed, e.g., for imported NVs when they shall be allocated as static NVs on the device.

To Change the NV Allocation Type

1. In the data point view, select the NVs in the CEA709 port folder, for which the NV allocation shall be changed.

Tip! *By using Ctrl-A all NVs can be selected.*

2. Select the **NV allocation** property as indicated by the red rectangle in Figure 187.

3. To make the data points static NVs on the device, select **Static NV** in the **Basic Properties** section.

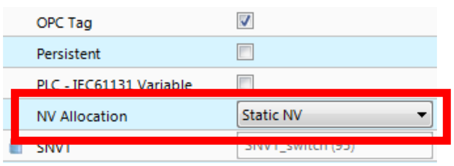


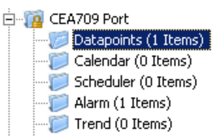
Figure 187: Change the NV allocation type.

7.7.7 Create Static NVs

The LOYTEC device can be configured to change its static interface and boot with a new one. Apart from creating static NVs from scanned or imported data points, static NVs can also be created manually in the CEA709 port folder.

To Create Static NVs Manually

1. Select the **Datapoints** folder under the CEA-709 port folder.



2. Right-click in the data point list and select **New Datapoint…** in the context menu. This opens the NV creation dialog as shown in Figure 188.

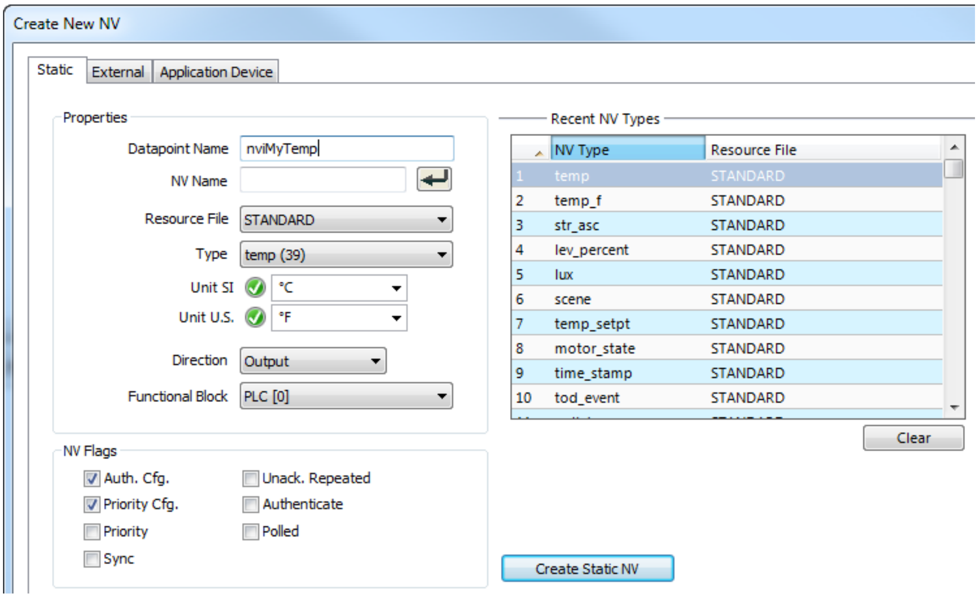


Figure 188: Create a static NV manually.

3. Enter a data point name and a programmatic name. The programmatic name is the name of the static NV which is being created.

4. Select a resource file. To create a SNVT, let the STANDARD resource file be selected.

5. Select a SNVT and a direction. If a non-standard resource file has been selected, choose from one of the UNVTs.

Tip! *Recently created SNVTs are available in the **Recent NV Types** list. Click on one to set the NV type without scrolling through the drop-down box.*

6. The chosen SNVT implies a specific network unit. Compatible units for the metric (SI) and U.S. systems are chosen. Adapt these to your needs.

7. Choose a functional block, where this static NV shall be located in.

8. Click **Create Static NV**. The static NV is created and appears in the data point list.

9. Note, that the static interface of the device will change as soon as static NVs are added or modified in the data point manager. This change is reflected in a new model number, which the device will receive after the configuration download (see Section 6.6.4). Also note that the manually created static NVs are not bound automatically by the Configurator. They simply appear on the device and need to be bound in the network management tool.

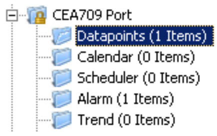
10. Click **Close**.

7.7.8 Create External NVs

External NVs are not actually allocated NVs on the device as NVs. Instead, the device uses polling to read data from and explicit updates to write data to external NVs. Since external NVs do not affect the static NV interface of the device, they can be used to extend the interface configuration at run-time, when no LNS with dynamic NVs is available.

To Create an External NV manually

1. Select the **Datapoints** folder under the CEA-709 port folder.



2. Right-click in the data point list and select **New Datapoint…** in the context menu. This opens the NV creation dialog.

3. Click on the tab **External** as shown in Figure 189.

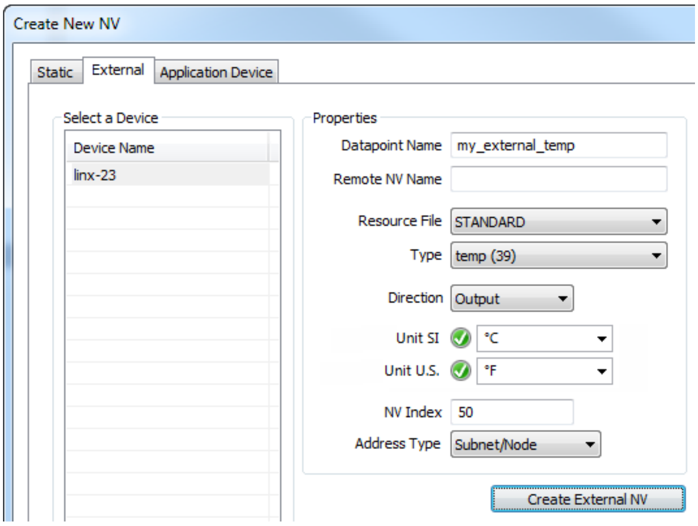


Figure 189: Create a new external NV.

4. Select the device in the box **Select a Device** on the left-hand side.

5. Enter the properties of the external NV on that device, starting with the local data point name, the remote programmatic NV name, the NV type (SNVT) and direction. Note, that the direction is the direction of the external NV on the device. Therefore, the remote output NV nvo00_temp becomes an input on the device. Also enter the NV index in decimal notation. This is the index under which the NV is found on the remote device. Choose the preferred addressing mode, e.g., Subnet/Node.

6. Click **Create External NV** to add this NV to the data point list.

7. The external NV now appears in the data point list. For external NVs, which are inputs to the device, adapt the poll cycle property to your needs.

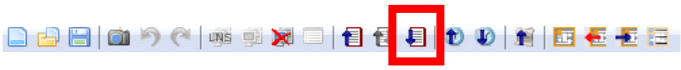
7.7.9 Configuration Download over LNS

After the data points have been configured, the configuration needs to be downloaded to the device. For doing so, the device must be online. If the device is not yet connected to the network, the configuration can be saved to a project file on the local hard drive.

If the Configurator is connected to a CEA-709 device via LNS and the device uses static or dynamic NVs, it can automatically generate Bindings in the LNS database. This behavior can be influenced in the download dialog. The download process can also manage the device template upgrade automatically in LNS, if the static interface changes (see Section 6.6.4).

To Download a Configuration

1. In the main connections window, click on the **Download Configuration** speed button



in the tool bar of the main connections window. This will open the configuration download dialog as shown in Figure 178.

2. If no bindings shall be generated, deselect the **Automatically create bindings** checkbox indicated by the red circle in Figure 190.

3. If the static NV interface has been changed, a new model number for the device needs to be selected. This is necessary, as the static network interface of the device changes on the CEA-709 network. The Configurator automatically selects a usable value, which can be overridden in the field **Model Number** marked by the blue rectangle in Figure 178.

4. Click **Start** to start the download. Each of the actions is displayed in the **Task List** section of the dialog. The current progress is indicated by the progress bar below.

5. When the download process has finished, a notification window appears, which has to be acknowledged by clicking **OK**.

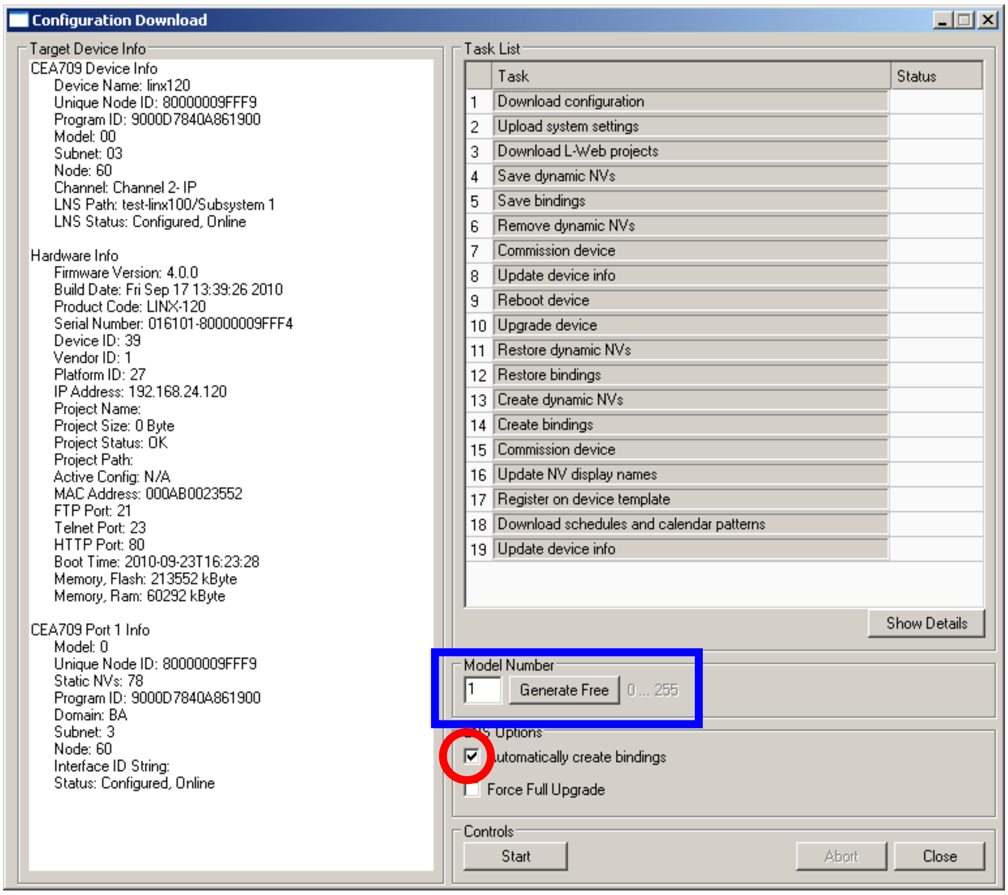


Figure 190: Configuration Download Dialog via LNS.

Note, that after the download is complete, the interface changes become active on the device (i.e., the static NV interface has changed). Refresh the network management tool to synchronize the tool with the changes to the LNS database made by the Configurator (e.g., use the menu “LonMaker|Refresh” in LonMaker or hit *F5* in NL-220).

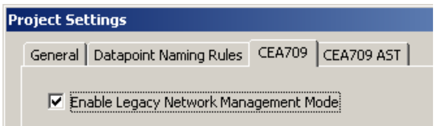
Normally, the Configurator software optimizes the download process by not executing certain LNS operations, if not necessary. For example, only those bindings and dynamic NVs are deleted and re-created, which correspond to real changes in the interface. The user can check the **Force Full Upgrade** option to clean and re-do all steps.

7.7.10 Enable Legacy NM Mode

For network management tools, which do not support the ECS (enhanced command set) network management commands, the legacy network management mode must be configured. Please contact the tool’s vendor for information whether ECS is supported or not. Note, that changing to legacy network management mode changes the static interface of the device.

To Enable Legacy NM Mode

1. In the Configurator menu go to **Settings → Project settings …**
2. Click on the tab **CEA709**.
3. Put a check mark in Enable Legacy Network Management Mode.



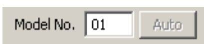
4. Click **OK.**
5. Download the configuration to activate the change.

7.7.11 Build XIF for Port Interface

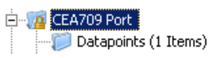
When using static NVs on the device, the Configurator can export a new XIF file for the changed static interface. Before exporting the XIF for the interface it is recommended to download the configuration into the device. In this case, the Configurator can verify that the model number of the port is correct.

To Create a XIF File

1. Make sure the **Model No** will match the final model number of the port. If not, enter an appropriate model number in the toolbar of the **Datapoints** tab.



2. Select the **CEA-709 Port** folder



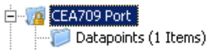
3. Right-click on that folder and in the context menu select **Build XIF …**.
4. This opens a file requestor where the XIF file name needs to be entered. Select a useful name to identify the device, e.g., as “LINX-10X_1.xif”.

7.7.12 Upload Dynamic NVs from Device

In LNS-based tools it is possible to create dynamic NVs on the device manually. This is a possible workflow to engineer the NV interface of the device in the LNS database. To use those manually created dynamic NVs, the Configurator must synchronize its dynamic NV information with the CEA-709 port.

To Upload Dynamic NVs

1. Select the **CEA709 Port** folder.



2. Right-click and select **Sync Dynamic NVs** in the context menu. The Configurator then loads any new dynamic NVs, which have been created but are not yet represented by data points on the CEA-709 port. The process completes when the dialog shown in Figure 191 appears.

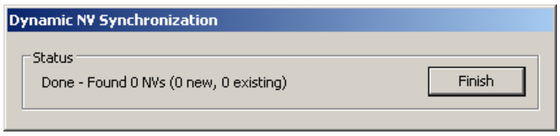


Figure 191: Synchronizing dynamic NVs from the device.

3. Click on **Finish**. The new dynamic NVs now appear in the data point list and can be edited and used on the device.

7.8 Advanced CEA-709 Configuration

7.8.1 Import Devices from XIF Templates

When working entirely without LNS, nodes on the network can be engineered via importing device templates from a XIF file. The Configurator provides a XIF device template import feature. Having devices imported from a XIF file is similar to have devices scanned online from the CEA-709 network, only their actual node IDs are unknown.

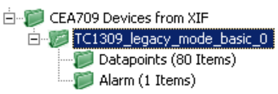
To Import from a XIF Template

1. Select the folder **CEA709 Devices from XIF**.

2. Right-click on the folder and select **Create device(s) from XIF file…** from the context menu.

3. In the file open dialog select a XIF file to import and click **Ok**.

4. The imported data points appear as a device sub-folder of the **CEA709 Devices from XIF** folder named after the XIF file name.



5. In that folder select those data points, which shall be used on the device and use them on the device as described in Section 7.7.5.

6. Repeat the import of XIF files for as many nodes as needed. The same XIF can be imported more than one time, resulting in multiple nodes of the same type in the **CEA709 Devices from XIF** folder.

7.8.2 Install Unconfigured Devices

CEA-709 devices must be installed by a network management tool (e.g., LNS-based tool) to be available for communication. Devices can be imported from a CEA-709 network scan or from a XIF file. If no network management tool is available, the CEA-709 device manager must be used to install the unconfigured devices. To install a device the following steps need to be done:

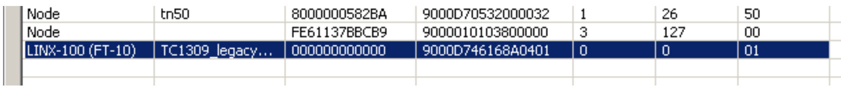
- The imported devices must be assigned to actual devices on the network. This is done by setting a node ID that corresponds to a node on the network.

- The domain information must be written to the device and it must be set configured, online to be ready for data communication.

To Install Devices

1. Open the CEA-709 management dialog by clicking on the **Manage CEA-709 Devices** speed button.

2. If devices have been imported via a XIF file, they do not have a node ID (all zero). To assign the physical node to the device, select the imported device.



3. Click the **Update NodeID** button and press the service button on the network node. The node ID will be filled in to the selected device. Alternatively the node ID can also be entered manually.

4. After node IDs have been assigned to all unassigned devices, select the device(s) to install in the **Device List** of the CEA-709 management dialog. Multi-select of devices is possible.

5. Click the **Install** button. This opens the **Install Devices** dialog as shown in Figure 192.

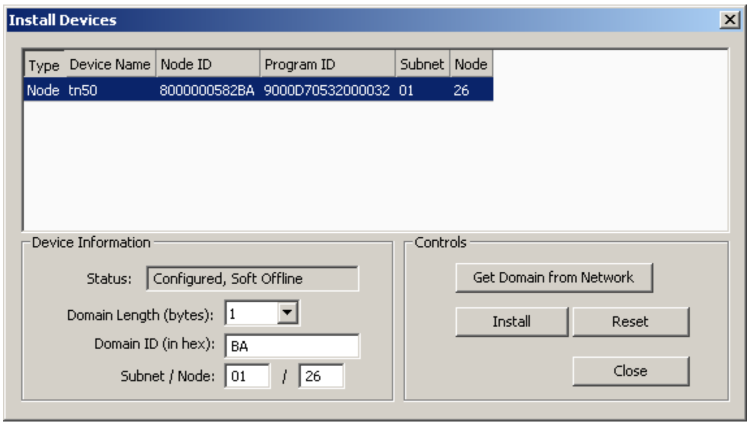


Figure 192: Install devices dialog.

6. Select the device to be installed.

7. Enter the domain information or click **Get Domain from Network** and press a service pin.

8. Enter a subnet and node address and click **Install**.

9. Some nodes won't be operable on the new settings until they are reset. Click the **Reset** button to reset the selected node.

10. Repeat this step for other unconfigured devices on the network.

7.8.3 Using Feedback Data Points

Feedback data points allow reading back the value written out over an output data point. In LONMARK systems getting a feedback value is normally accomplished by creating a dedicated feedback NV on the device, which can be bound back to the devices that are interested in the currently active value on an output.

Some nodes, however, do not possess such feedback NVs for certain functions. To support getting feedback values on such nodes, the Configurator can create feedback data points based on existing output data points. This is especially interesting for bound output NVs (static and dynamic alike). The corresponding feedback data point is an input, which uses the original output NV for polling the target NV. Once the binding is changed the new target is polled. No additional input NV needs to be created for the feedback value, if the feedback data point feature is used. Alternatively, the output data point can be switched to a value data point with an integrated feedback function without the need for an extra feedback data point.

To Create a Feedback Data Point

1. Select an output data point in the data point list of the **CEA-709 Port** folder, e.g. 'nvoHumid101'.

2. Right-click and choose **Create Feedback-Point** from the context menu.

3. A new input data point is created, having '_fb' appended to the original name, e.g., 'nvoHumid101_fb'. Note, that the feedback data point maps to the same NV index as the original output data point.

4. Choose an appropriate poll cycle in the data point properties for the feedback data point.

To Create an Integrated Feedback

1. Select an output network variable in the data point list of the **CEA-709 Port** folder, e.g. ‘nvoHumid101’.

2. In the data point properties tab change the direction from ‘output’ to ‘value’.

3. Choose an appropriate poll cycle in the data point properties for the value data point.

7.8.4 Working with Configuration Properties

Configuration properties (CPs) are supported by the LNS network scan and the online network scan. They can be selected and used on the device in a similar way as NVs. There is a notable difference to NVs: CPs are part of files on the remote nodes. Reading and writing CPs on the device results in a file transfer.

The device supports both, the LONMARK file transfer and the simpler direct memory read/write method. In both cases however, one has to keep in mind that a file transfer incurs more overhead than a simple NV read/write. Therefore, polling CPs should be done at a much slower rate than polling NVs (e.g., every 10 minutes).

Another aspect is how CPs are handled by network management tools. Formerly, those tools were the only instance that could modify CPs in devices. Therefore, most tools do not automatically read back CPs from the devices when browsing them. This can result in inconsistencies between the actual CP contents on the device and their copy in the network management tool. It is recommended to synchronize the CPs from the device into the LNS database before editing and writing them back.

To Synchronize CPs in NL220

1. Double-click on the device object in the device tree

2. Press the **Upload** button on the Configuration tab of the device properties (see Figure 193).

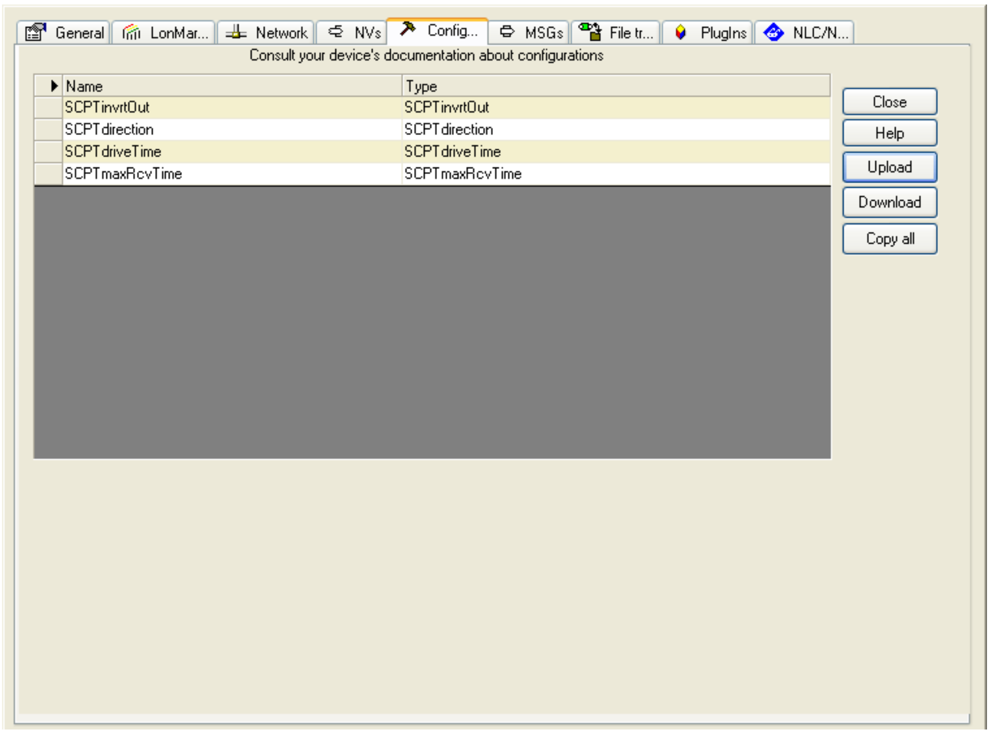


Figure 193: Configuration Tab for Configuration Properties in NL220.

To Synchronize CPs in LonMaker TE

1. Right-click on a device object and select **Commissioning → Resync CPs…** from the context menu.

2. This opens the dialog shown in Figure 194.

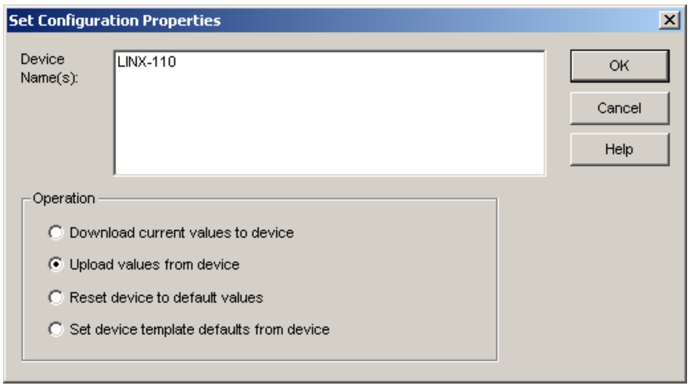


Figure 194: Set Configuration Properties in LonMaker TE.

3. In this dialog select the radio button **Upload values from device** in the **Operation** group box. To use the current settings of the device as default values for new devices, select **Set device template defaults from device**.

4. Execute the operation by clicking the **OK** button.

7.8.5 Working with UNVTs, UCPTs

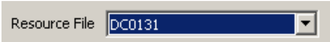
This device supports user-defined type, including user-defined network variable types (UNVTs) and user-defined configuration property types (UCPTs). In order to interpret the contents of user-defined types, the *device resource files* supplied by the vendor must be added to the resource catalog on your PC.

Once the resource files are installed, the CEA-709 network scan and the LNS scan will display the user-defined types from the resource files. Those data points can be used on the device like regular, standard-type data points. Also manual creation of UNVTs can be performed.

To Manually Create a Static UNVT

1. Perform the steps to manually create a static NV as described in Section 7.7.7.

2. When the **Create New NV** dialog appears, change the resource file from 'STANDARD' in the **Resource File** drop-down box to the desired, user-defined resource file



3. Then select the desired UNVT from the **Type** drop-down list below. This list will display the types of the selected resource file only.

4. Click **Create Static NV** to create the UNVT on the device.

7.8.6 Configure User-Defined Function Blocks

As a default the device comes with 8 LONMARK function blocks in which the user can create NVs. They have a pre-defined name ('Gateway' or 'PLC' depending on the device model). For complex applications it may, however, be desirable to change those function blocks to more meaningful names in order to group NVs in a better way. Note, that this change will also change the static interface of the device. This makes a new model number necessary (see Section 6.6.4).

To Configure Function Blocks

1. Select the menu **Tools → Manage LonMark Objects …** . The dialog Manage LonMark Objects as shown in Figure 195 appears.

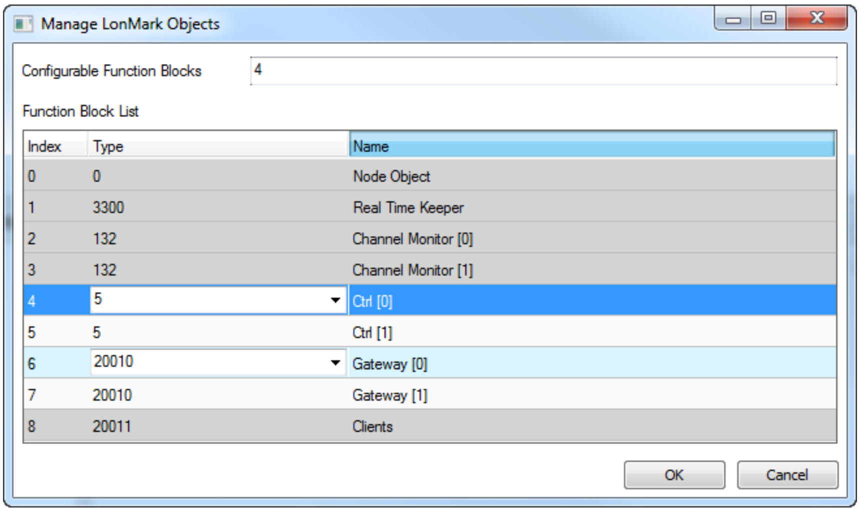


Figure 195: Manage LONMARK Objects.

2. Edit the field **Configurable Function Blocks** to the number of needed function blocks.

3. Select a **Type** from the drop-down box for your function block. This drop-down box is available at the top of a function block array. A change applies to the entire array.

4. To break up the pre-defined array, double-click on a name in the **Name** column and edit it to something different. Typing the same name for consecutive function blocks will create a new array.

5. To use an object type outside the scope of standard types, just type in the number instead of choosing from the drop-down.

7.9 BACnet Configuration

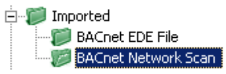
7.9.1 Scan for BACnet Objects

LOYTEC devices also support an online network scan on the BACnet network. In this scan the device searches for other devices on the BACnet network and pulls in the BACnet object information of these devices. These BACnet objects can then be used on the device as the basis for client mapping.

To Scan for BACnet Objects

1. Go to the **Datapoints** tab.

2. Select the folder BACnet Network Scan



3. Right-click on that folder and select **Scan BACnet Network…**. This opens the BACnet Network Scan dialog as shown in Figure 196.

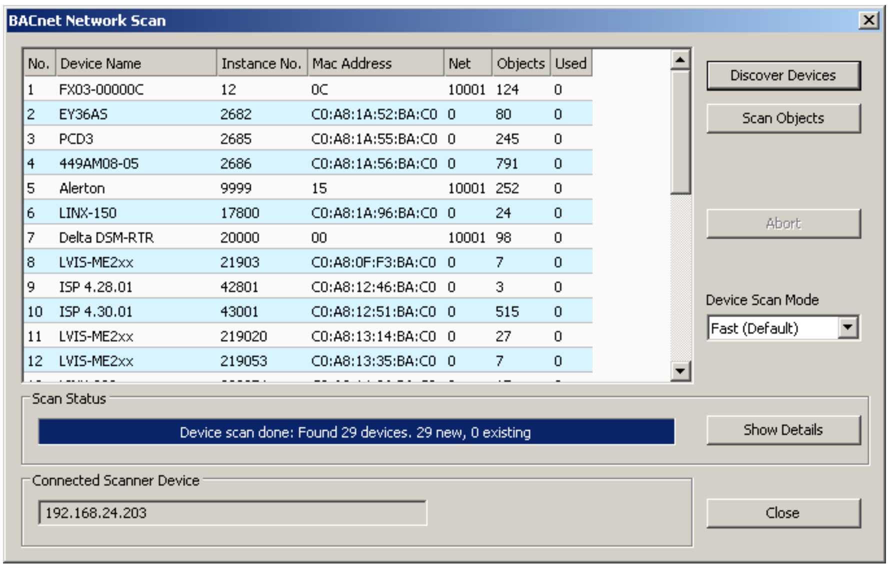


Figure 196: BACnet network scan dialog.

4. Click on the button **Discover Devices**. This starts a network scan. The results are put in the device list box. A progress bar below indicates how many devices are being scanned.

5. Select a device in the device list and click the button **Scan Objects**. This scans the BACnet objects on the selected device and adds them to the **BACnet Network Scan** folder as a separate sub-folder for the device.

6. If the scan does not give the expected results, change the **Device Scan Mode** to normal or slow and try again. With this setting the scanner uses simpler but slower protocol features.

7. Click **Close** when all devices needed have been scanned.

Note: *If proprietary properties access on a remote device is required, support for proprietary properties must be enabled in the BACnet tab of the Project Settings dialog (see Section 7.3.5).*

7.9.2 Import from EDE File

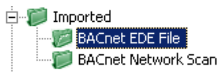
If the device is engineered offline or some of the required BACnet devices are not yet online in the network, the engineering process can be done by importing a device and object list from a set of EDE files. These objects also appear in the import folder and can be later used on the device as client mappings.

There are a set of EDE files. Select the main EDE file, e.g. *device.csv*. The EDE import will also search for the other components, which must be named *device-states.csv*. Which components are expected, please refer to Section 18.3.12. Example EDE files can be found in the ‘examples’ directory of the LOYTEC Configurator software installation directory.

To Import BACnet Objects from an EDE File

1. Go to the **Datapoints** tab.

2. Select the folder **BACnet EDE File**



3. Right-click and select **Import File**. In the following file selector dialog, choose the EDE import file and click **OK**.

4. Now the **BACnet EDE File** folder is populated with the imported BACnet objects.

7.9.3 Use Imported BACnet Objects

After BACnet objects have been imported (with a network scan or by importing from an EDE file) the user can select the BACnet objects that the device shall access. When executing the **Use on device** the configuration software allocates client mappings on the device. These client mappings will read or write values from the BACnet objects in the network.

In an additional step, there can be also server objects allocated on the device. These server objects can be created automatically from converting a client mapping to a server object. This is usually done, if the imported BACnet objects shall also be directly modified over the BACnet network on the device itself.

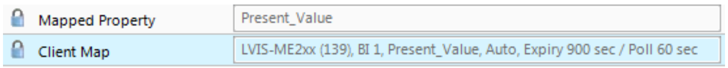
To Use Imported BACnet Objects on the Device

1. Open the data point manager dialog and select the desired BACnet objects in one of the import folders.

2. Use the multi-select feature by holding the *Shift* or *Ctrl* keys pressed.

3. Click on the button **Use on Device** in the tool bar.

4. This creates data points in a remote device sub-folder of the BACnet Port/Datapoints folder. All data points in that folder will be created as client mappings. No server object is created automatically in this case.



5. To also create server objects select the data points in question using the multi-select feature. Then activate the property **Allocate Server Object** in the section **Advanced**.

6. For editing the client mapping, you may multi-select client map data points and edit the corresponding data point properties **Client Confirmed COV**, **Client COV Expiry**, **Client Map Type**, **Client Write Priority**, **Remote Instance Number**.

7.9.4 Create a Client Mapping

The client mapping information can also be created manually. Usually, this is done to create client data points without importing information from EDE or scanning online.

To Create a Client Mapping

1. Select the **Datapoints** folder under the **BACnet Port** folder.

2. Right-click in the data point list and select **New Datapoint…** in the context menu. This opens the **Create New BACnet Point** dialog as shown in Figure 197.

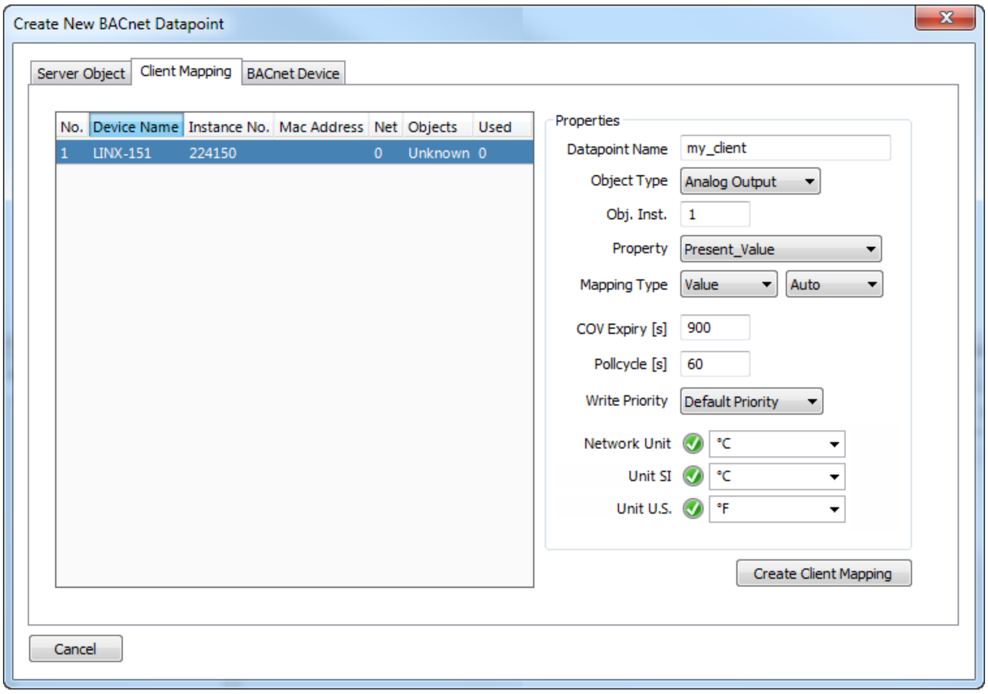


Figure 197: Create Client Mapping Dialog.

3. Select the tab **Client Mapping**.

4. Choose a target device in the list of known devices. Enter a **Data Point Name**, choose an **Object Type**, and edit the target object instance number. Then select the **Mapping Type**. For read client mappings edit the **COV expiry** or **Pollcycle** setting. For write

client maps edit the **Write Priority**. For value client maps edit both. When finished click **Create Client Mapping**.

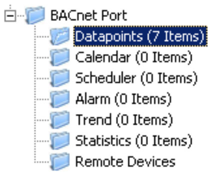
5. For an analog client mapping define a **Network Unit**. This is the engineering unit of the remote object. Optionally, define a unit representation of the remote scalar value for the metric (SI) and U.S. unit system on the device.

7.9.5 Create Server Object

On the BACnet port server objects can also be created manually. These BACnet objects are visible on the BACnet network and can be modified by other devices. They appear as data points in the **BACnet/Datapoints** folder.

To Create Server Objects Manually

1. Select the **Datapoints** folder under the **BACnet Port** folder.



2. Right-click in the data point list and select **New Datapoint…** in the context menu. This opens the **Create New BACnet Point** dialog as shown in Figure 198.

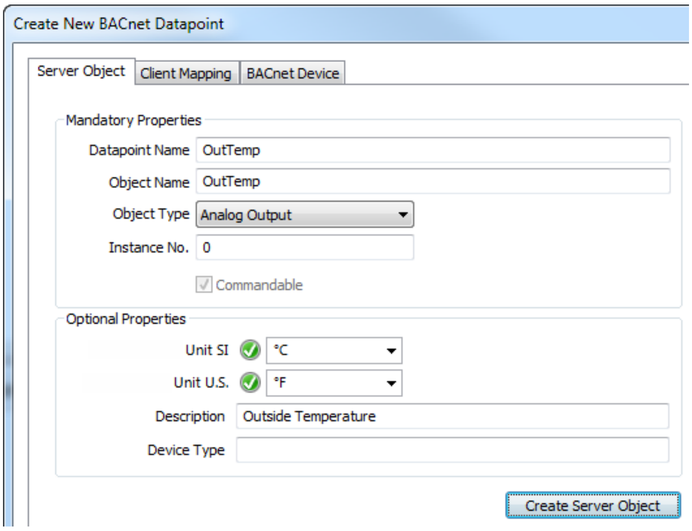


Figure 198: Create a Server Object manually.

3. In the **Mandatory Properties** enter a **Datapoint Name** and an **Object Type**. Optionally, update the **Instance No** and select the **Commandable** check box for value objects, if the value object shall be commandable from the network.

4. In the **Optional Properties** you may select **Unit SI** and **Unit U.S.** for analog objects. BACnet objects have no fixed network unit. Depending on the chosen unit system, the analog BACnet object will be created with the specified metric (SI) or U.S. unit in the engineering unit property.

5. For all object types you can enter the **Description**. The **Device Type** can be left empty. For multi-state objects you have to select a multi-state map.

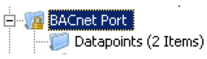
6. Click **Create Server Object**. The BACnet data point is created and appears in the data point list.

7.9.6 Export Server Objects to an EDE File

When engineering offline it can be beneficial to hand out the server object configuration of the device to other parties electronically. For doing so you may export the server object configuration to a set of EDE files. The set of EDE files consist of the main EDE file, e.g. *myDevice.csv*. This file contains the list of all objects and refers to state texts that are exported to a second file named *myDevice-states.csv*. For which components are exported in an EDE file, please refer to Section 18.3.12.

To Export an EDE File

1. Select the **BACnet Port** folder.



2. Right-click and select **Export EDE …** in the context menu. This opens the EDE export dialog to enter the EDE header information as shown in Figure 199.

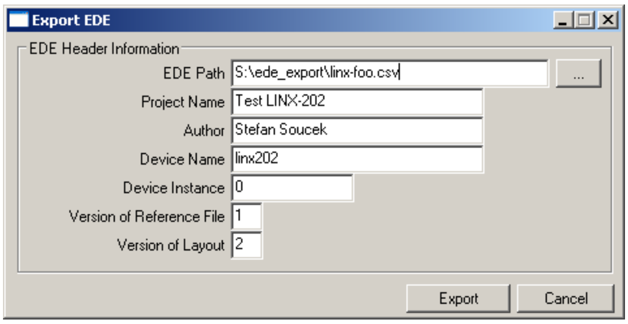


Figure 199: EDE Export Dialog.

3. Click the  button to select the EDE file export location.

4. Specify the **Device Name** and **Device Instance**. The device instance will be used by other tools to configure their BACnet clients for accessing the exported device.

5. Optionally fill in project name, author to document that information in the EDE file.

6. Click **Export**.

7.9.7 Import Server Objects from an EDE File

It is also possible to import a BACnet server object interface from EDE files. In this use case, the device is configured to resemble a the device of the EDE file. If conflicts in instance numbers or object names arise with already existing server objects, the imported objects are re-assigned.

There are a set of EDE files. Select the main EDE file, e.g. *device.csv*. The EDE import will also search for the other components, which must be named *device-states.csv*. Which components are expected, please refer to Section 18.3.12. Example EDE files can be found in the ‘examples’ directory of the LOYTEC Configurator software installation directory.

To Import BACnet Server Objects from an EDE File

1. Select the folder **BACnet Port**

2. Right-click and select **Import Server Objects from EDE…**. In the following file selector dialog, choose the EDE import file and click **OK**.

3. Now a folder for the device in the EDE file is generated and a report is displayed, informing about the imported objects and possible reassignments.

7.9.8 Map other Properties than Present_Value

When creating a BACnet server object, the Present_Value property is mapped by the created data point. That means writing and reading on the data point reads or writes the Present_Value. If other properties shall be accessed, they must be added to the BACnet server object's data point.

To Add other BACnet Properties

1. Select the BACnet server object for adding properties.

2. Right-click on the data point and select **Add/Remove BACnet properties …** from the context menu. The dialog appears as shown in Figure 200.

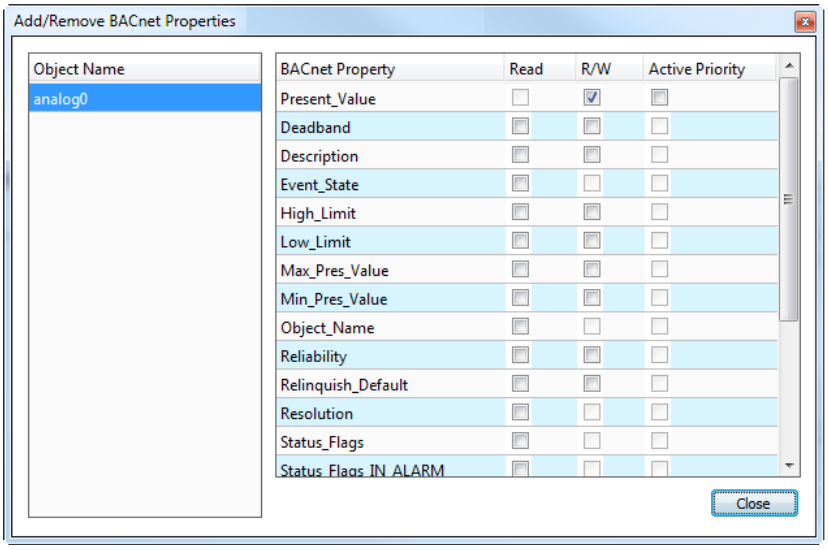
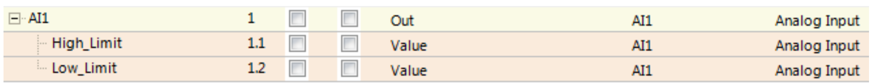


Figure 200: Dialog for adding/removing BACnet properties.

3. Check the additional properties. Checking the **Read** box will add an input data point, checking the **R/W** box will add a value data point.

4. Click **Close**. The selected data point can now be expanded with the plus icon and show its additional properties as sub-data points.



5. To remove properties perform the same steps and uncheck the corresponding check-boxes. Alternatively, select the property (or more) and press the *Delete* key.

7.9.9 Enable International Character Support

By default BACnet objects on the device contain ASCII strings in properties such as object name, description, active/inactive text, state texts. This is the setting most third-party tools are interoperable with. To support international character sets, the device can be configured to expose strings as ISO-8895-1 (for most Western European languages) or UCS-2 (for Unicode character sets such as Japanese).

To Enable International Character Support

1. In the Configurator software menu go to **Settings → Project settings …**. This opens the **Project Settings** dialog (see also Section 7.3.5).

2. Click on the tab **BACnet**.

3. Put a check mark either on **ASCII/UTF-8** (default), **UCS-2** (Unicode, e.g., for Japanese), or **ISO-8859-1** (for Western European languages).

4. Click **OK.**

5. Download the configuration to activate the change.

7.9.10 Read the Active Priority

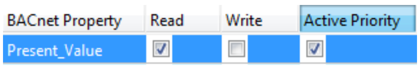
In BACnet objects that possess a priority array the effective value depends on the used write priority slots. A special data point on the device can be added that allows reading out the active priority of such an object, giving a value between 1 and 16.

To Read the Active Priority of a Local Object

1. Select the BACnet server object for adding properties.

2. Right-click on the data point and select **Add/Remove BACnet properties …** from the context menu. The dialog for mapping BACnet properties to data points appears.

3. For the Present_Value select the additional box Active Priority.



To Read the Active Priority of a Feedback Value

1. Select a write client mapping.

2. Right-click on the data point and select **Create Priority Feedback Point** from the context menu.

3. A new data point is created, which is a feedback client mapping that reads the active priority out of the remote object.

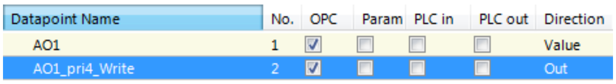
7.9.11 Write and Read with Priority

In BACnet objects that possess a priority array the effective value depends on the used write priority slots. As default, data points for those objects have the direction input, which is intended to read the resulting value. Additional priority write output data points can be added for writing to the BACnet object. For those data points, a write priority between 1 and 16 can be defined. The default write priority is defined by the project settings.

To read back the value at a given priority slot, additional priority read input data points can be added. They reflect the value at the configured priority slot between 1 and 16. If the priority slot is NULL the data point stays at invalid value.

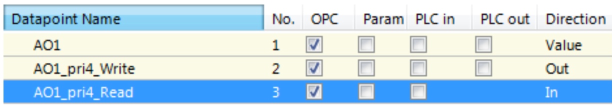
To Create a Priority Write Data Point

1. Select the commandable BACnet server object.
2. Right-click and select **Create Priority Write Point…** from the context menu.
3. A dialog prompts for the write priority. Note that the write priority can also be changed later.
4. The new priority write data point appears below the original BACnet server object data point.



To Create a Priority Read Data Point

1. Select the commandable BACnet server object.
2. Right-click and select **Create Priority Read Point…** from the context menu.
3. A dialog prompts for the read priority. Note that the read priority can also be changed later.
4. The new priority read data point appears below the original BACnet server object data point.

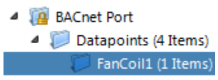


7.9.12 Duplicate BACnet Devices with Data Points

When importing BACnet devices via network scan or EDE file, the resulting client mappings are used on the device. For each BACnet device a sub-folder is created which organizes the client mapping data points for that device. The BACnet device itself appears in the BACnet device manager. After editing the client mapping data points as appropriate, entire device folders can serve as templates for duplication. The created copies of the data points are pointing to an unassigned device, which can be commissioned later on the Web interface.

To Duplicate BACnet Devices

1. Select a folder created for a scanned/imported BACnet device.



2. Right-click and choose **Duplicate** in the context menu.
3. The **Duplicate data points and set naming rules** dialog opens as shown in Figure 201.

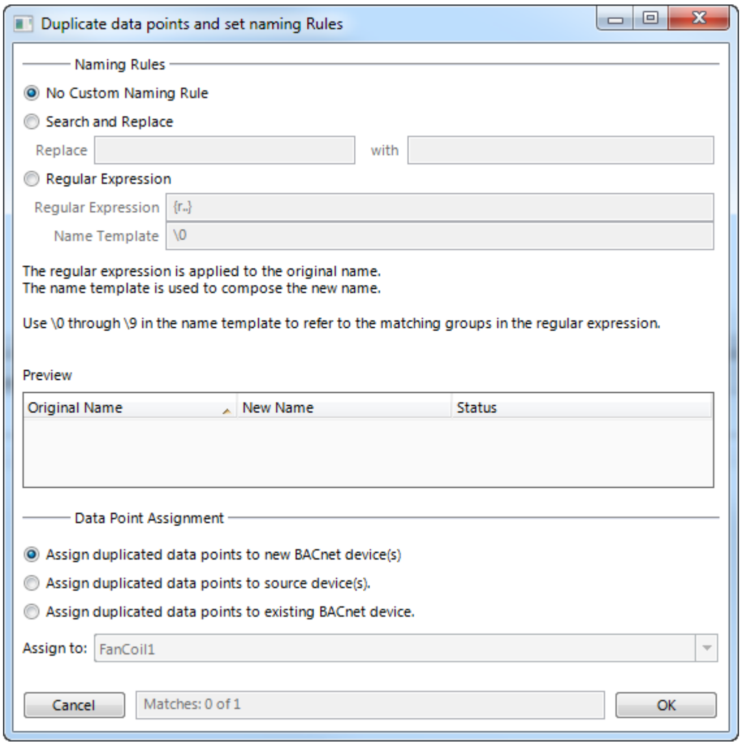


Figure 201: Duplicate BACnet devices.

4. In the **Data Point Assignment** section choose to assign the duplicated data points to a new BACnet device instance and click **OK**.

5. A new BACnet device folder is created with the duplicated client mapping data points. The original BACnet device is also duplicated, leaving the actual device instance number empty and marking the device to be commissioned later on the Web interface as described in Section 5.4.1.

6. The **BACnet Device Manager** shows the created devices as depicted in Figure 202.

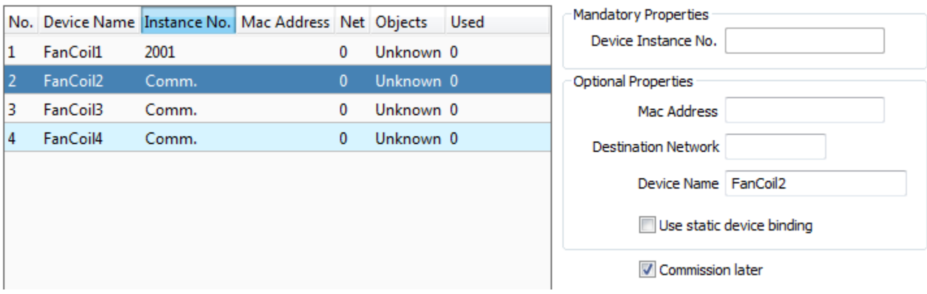


Figure 202: Duplicated BACnet devices for later commissioning.

7.10 Connections

7.10.1 Create a New Connection

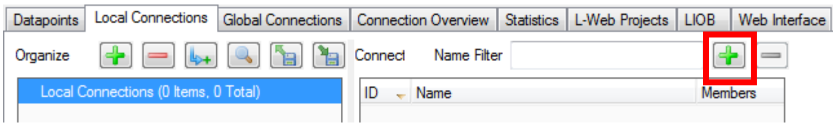
After having configured the device's network ports with data points, internal connections between those data points can be created. Usually, the manual method to create a connection is used to create connections between different named data points.

A connection is an internal mapping in the device between input and output data points. A connection always consists of one or more data points. A value update from an input data point (sender) is distributed to all output data points (receivers). A status change of a receiver data point is propagated back to all sender data points. All data points in the connection must be of a compatible type or use an adaptor.

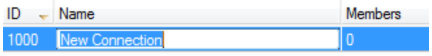
By adding data points as sender and receiver to the same connection, they transfer values in both directions. Doing so with connected data points, bi-directional connections can be built.

To manually create a new connection

1. Click on the **Local Connections** tab



2. in the main window and press the speed button **Create new Connection**. In the menu choose **Standard Connection**. A new connection is added to the connection list. Rename the connection if you want to do so.



3. Over the list **Datapoints in connection** on the right-hand side click on **Attach Data Point** to add data points for this connection. This opens a list of all available data points. Select one and press **OK**. You may use multi-select to select more than one data point at a time.

Note:	*By default only compatible data points are displayed. Sometimes compatible data points are available as member points (e.g., a SNVT structure member). Click on ⊞ to expand the data point and select the desired member point.*

4. Now the connection tab contains the new connection and below the list of data points in that connection as shown in Figure 203.

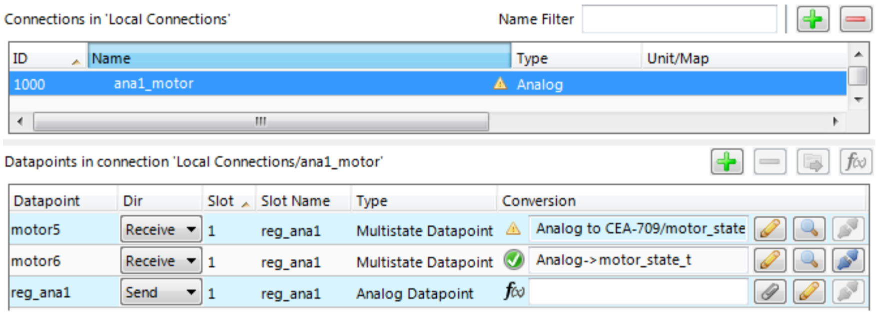


Figure 203: Connection tab with a connection and data points.

5. Change the direction by modifying **Send** or **Receive**. For changing multiple data points use multi-select. Optionally, select **Disable** to temporarily exclude this data point from communication in the connection.

6. For receive items you may optionally define a forward **Delay** in seconds (see Section 6.4.5).

7. If the attached data point needs a conversion, the item displays a yellow exclamation mark and the default conversion (e.g. 'Analog to CEA-709/motor_state_t'). By clicking on the button to view the current conversion.

8. To add a new conversion to this item, click on the button. A dialog opens, which displays the matching adaptors already available in the library as shown in Figure 204.

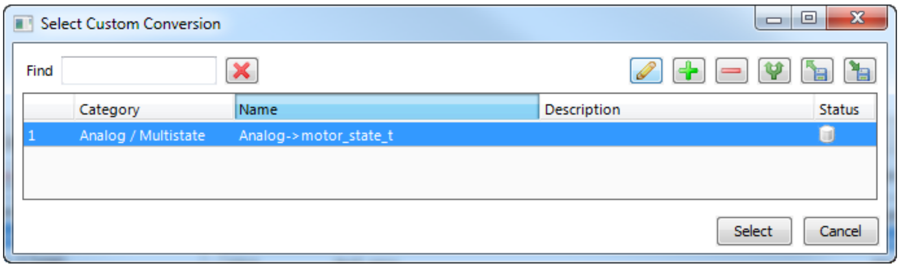


Figure 204: Choose a custom conversion.

9. Select an existing conversion, click the plus button to create a new conversion, or click the edit button to modify an existing conversion.

10. An example for editing an analog to multi-state value conversion is shown in Figure 205. Enter a **Conversion name**, then edit the **Value range from** column and select the desired **Target state** mapping.

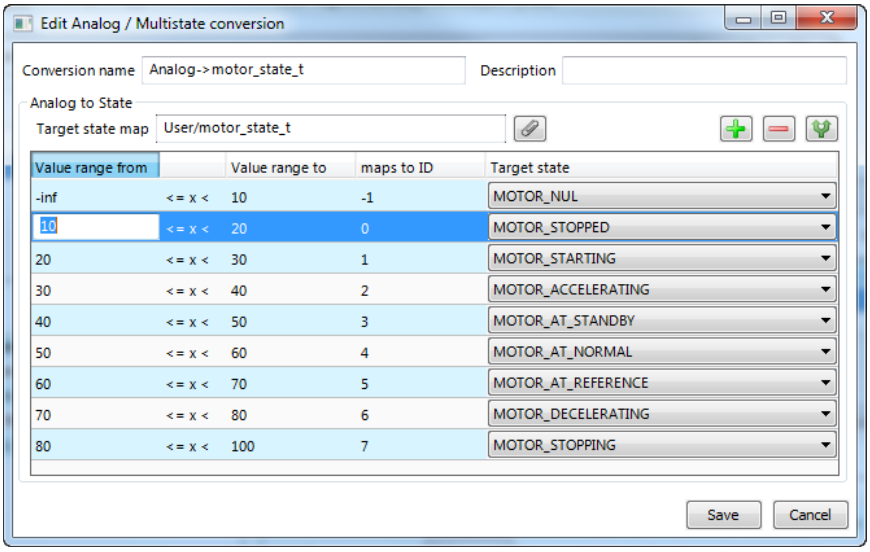


Figure 205: Edit an analog to multi-state conversion.

11. Click **Save** and then choose the newly created conversion by clicking **Select**.

12. The items with an assigned value conversion appears with a green checkmark .

To Create a Connection via Drag-and-Drop

1. Change to the **Datapoints** tab of the main window and navigate to the data point that shall be put into the new connection.

2. In the properties view below the data point list click on the **Local Connections** tab as shown in Figure 206.

3. Then simply drag a data point from the data point list and drop it onto an empty area in the connections list as shown in Figure 206.

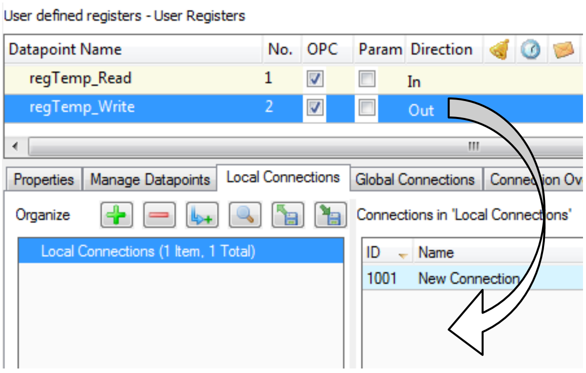


Figure 206: Create a connection with drag-and-drop.

4. To add data point to that connection, drop the new data point into the empty area in the connected data points list below.

To Create a Bi-Directional Connection

1. Start the connection by adding a data point. A value data point is added as **Send** to a new connection.

2. Then add the same data point a second time and do not create a multi-slot connection. This time it is added as **Receive**. This makes the data point send to and receive values from the connection.

3. Continue by adding the data point, which shall be connected. A value data point is added as **Receive** to an existing connection.

4. Then add the same data point a second time to that connection. This time it is added as **Send**.

5. Now both data points send values to and receive values from the connection. This synchronizes value changes back and forth between the involved data points. Update loops are suppressed by the connection. It is not necessary to set a COV on any of the involved data points. An example is shown in Figure 207.

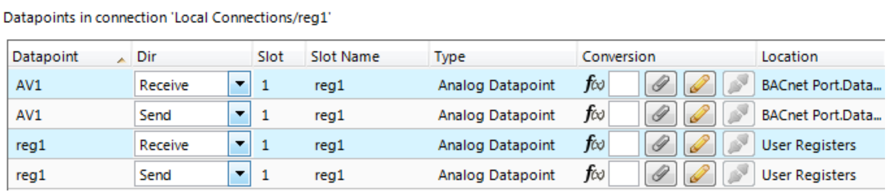


Figure 207: Bi-directional connection.

7.10.2 Create Connections from a CSV File

A quick way to perform batch edit on connections is to export and import connections from the connections CSV file. Each line in the connections CSV file identifies a connection. The first column is the connection name. The second column specifies the hub data point. The full path to the data point must be specified using the dot '.' as the folder separator. The third and following columns specify the target data points.

To Create Connection from a CSV File

1. Select the menu Tools → Export Local Connections …

2. Select an appropriate file name and export.

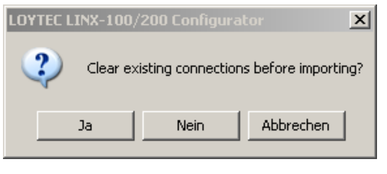
3. Edit the connections CSV file. An example is shown in Figure 208.

```
#connection_csv_ver,1
#ConnectionName,HubDPName,TargetDPName
Ai0,LINX-200.BACnet Port.ai0,LINX-200.User Registers.abs_humid1
Ai1,LINX-200.BACnet Port.ai1,LINX-200.User Registers.abs_humid2
Ai2,LINX-200.BACnet Port.ai2,LINX-200.User Registers.abs_humid3
Ai3,LINX-200.BACnet Port.ai3,LINX-200.User Registers.abs_humid4
```

Figure 208: Example Connection CSV File.

4. Select the menu Tools → Import Local Connections …

5. If connections that are not part of the connection CSV file shall be deleted, click **Yes** when prompted. Click **No** if the other connections shall be left as is.



6. Choose the file to import and click **Ok**.

7. When the import has completed, optionally view the log to check, which connections have been added, modified, and deleted.

7.10.3 Modify Connections

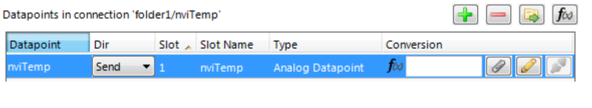
Connections can be edited and deleted. This is also done in the **Connections** tab of the main window. Editing connections does not influence the data point configuration. This means, when deleting a connection or adding/removing data points to/from a connection, the data points are not deleted.

To Edit a Connection

1. Change to the **Local Connections** tab of the main window.

2. Select the connection to edit. Then follow the steps as applied when creating a connection.

3. To detach a data point from the selected connection, select the data point and click on the button **Detach selected data points** over the connection member list.



4. Change the direction of a data point in the connection by choosing one from the **Dir** drop-down. You may select **Disable** to temporarily exclude this data point from the connection altogether.

To Add Data Points via Drag-and-Drop

1. Change to the **Datapoints** tab of the main window and navigate to the data point for being added.

2. In the properties view below the data point list click on the **Connections** tab as shown in Figure 209.

3. Select an existing connection.

4. Drag the selected data point and drop it into the empty area of the **Datapoints in connection** list as shown in Figure 209. This adds the data point to the selected connection.

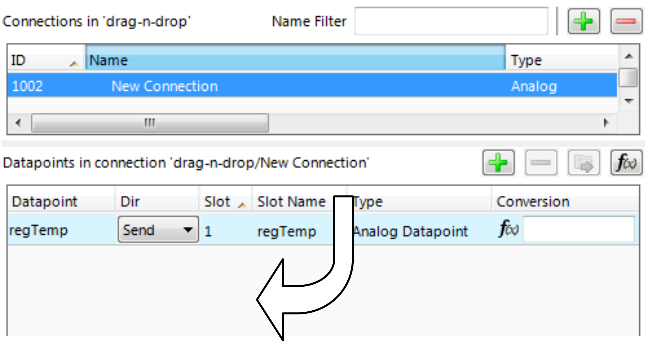


Figure 209: Modify connections in the properties view.

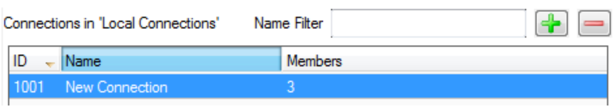
5. To replace a data point in a connection drop the new data point right onto an existing data point in the connection that shall be replaced.

To Delete a Connection

1. Change to the **Local Connections** tab of the main window.

2. Select the connection for removal. Use multi-select to select more than one connection.

3. Click on the button **Delete Connection** over the connections list.



7.10.4 Create a Multi-Slot Connection

A multi-slot connection can be used to connect a number of different data points together under one umbrella. It can be considered as a cable with many wires, each wire represented by a slot with a label. For example, one can create a multi-slot connection for a structured data point, connecting each sub-data point to another technology. All those slots appear under the same connection. But data point only exchange data, if they are added to the same slot.

To Create a Multi-Slot Connection

1. Change to the **Datapoints** tab of the main window and navigate to the data point to be connected.

2. Select a structured data point and drag it into the connections list to create a new connection.

3. A dialog prompts the user, whether to create a multi-slot connection or use the user data point as a single entity in the connection. Choose **Yes** to create a multi-slot connection.

4. A multi-slot connection is created as shown in Figure 210. The multi-slot connection can be collapsed or expanded. In the expanded view it shows all slots. Select the top-level multi-slot connection to view all data points in the connection. Select a single slot beneath to view only those data points in that slot.

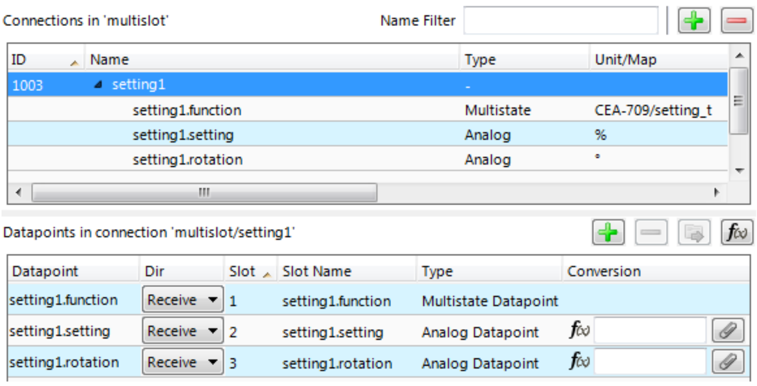


Figure 210: Multi-slot connection.

5. Add more slots by dropping data points onto the top-level multi-slot connection ‘setting1’.

6. Connect other data points to the slots by dropping them onto the slots. For example connect the register ‘regRotation’ by dropping it onto ‘setting1.rotation’ as depicted in Figure 211.

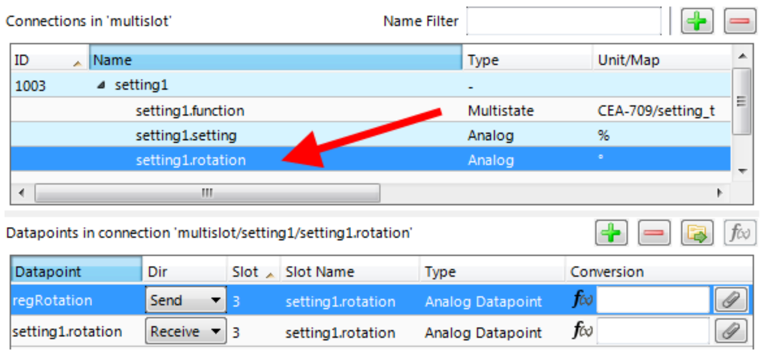


Figure 211: Add data point to connection slot.

7.10.5 Create a Math Block Adaptor

When connecting structured data points the multi-slot connection can be used. If a simple mapping of the sub-data points is not possible and a more advanced mathematical conversion is required, a math block adaptor can be created. This math block is based on a multi-slot connection with *n* inputs and *m* outputs (see Section 6.4.2).

To Create a Math Block Adaptor

1. Create a multi-slot connection from a structured data point, e.g., the input data point.

2. Add output slots to the multi-slot connection, e.g. by adding a structured output data point. An example is shown in Figure 212.

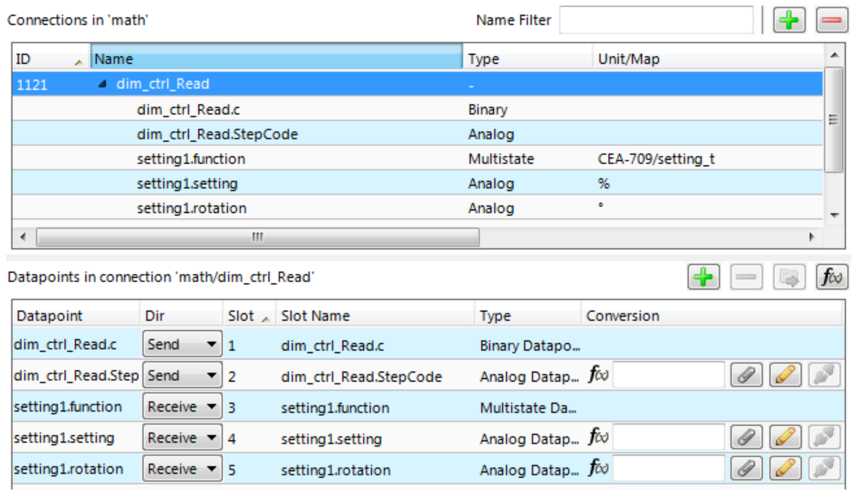


Figure 212: Multi-slot connection for math block adaptor.

3. Click on the **Create math adapter from connection** button. The dialog **Edit Multi-Slot Math Adaptor** opens as shown in Figure 213.

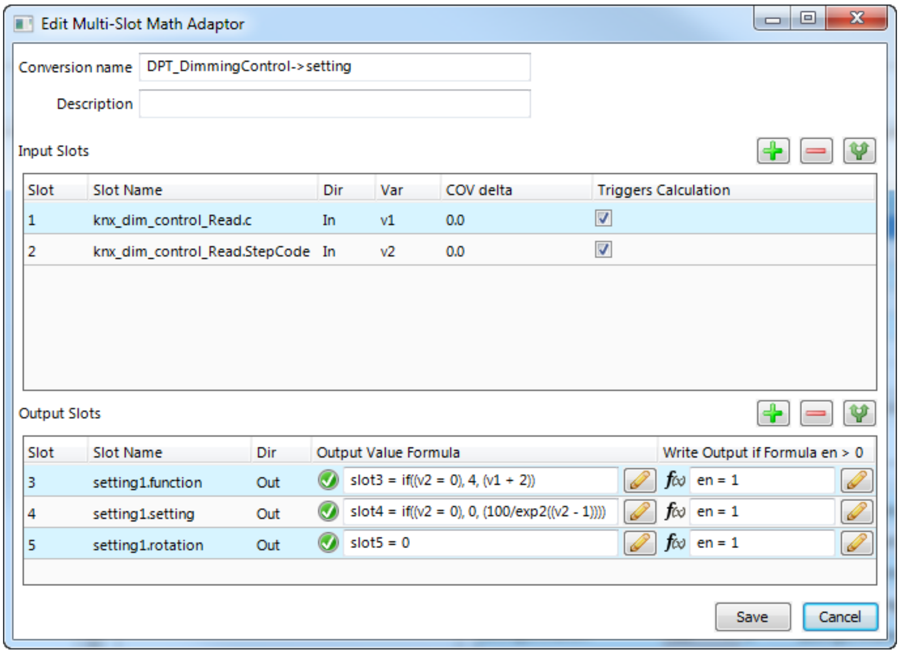


Figure 213: Edit a math block adaptor.

4. Enter a **name** and **description** for the adaptor.

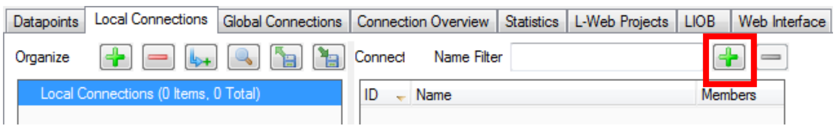
5. For each output enter an **Output Value Formula**. This can be done by directly typing the formula or by clicking the edit button .

6. Optionally, etner an enable formula into **Write Output if Formula en > 0**. As a default enable is ‘1’.

7. When finished with the math block click **Save**.

To Use an Existing Math Block Adaptor

1. Click on the **Local Connections** tab



2. in the main window and press the speed button **Create new Connection**. In the menu choose **Connection with Math Adaptor**.

3. In the dialog **Select Multi-Slot Math Adaptor** select an existing adaptor and click **Select**. A new multi-slot connection is added to the connection list with empty slots as depicted in Figure 214.

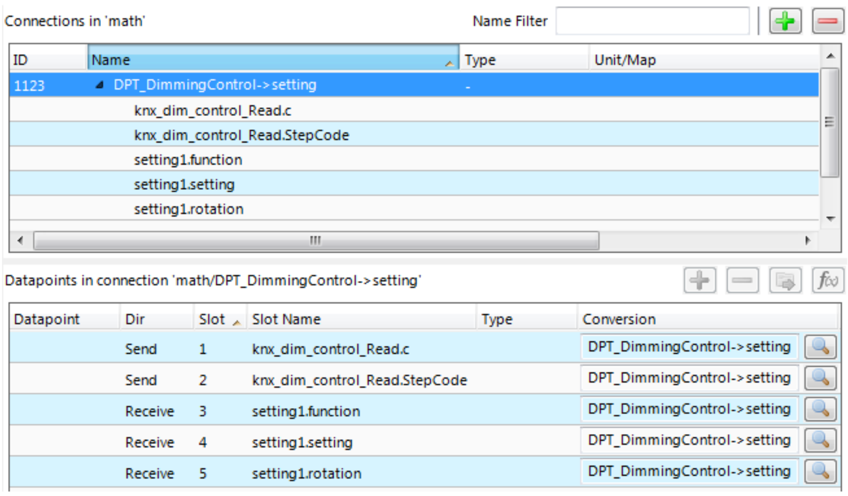


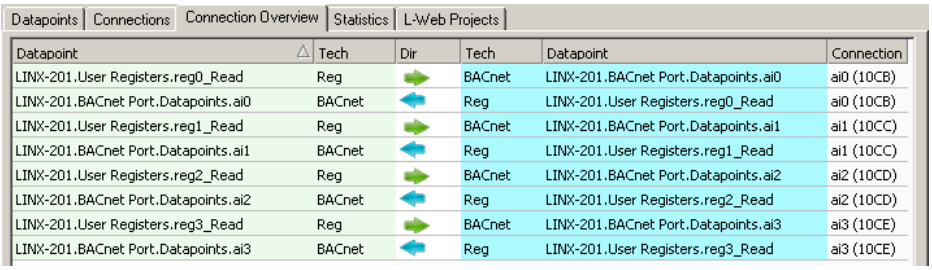
Figure 214: Created new multi-slot connection from math adaptor.

4. Now connect data points by dragging and dropping them onto the empty slots in the data point list below.

5. To view the math conversion click the magnifier button .

7.10.6 Connection Overview

Select the **Connection Summary** tab to get a graphical representation of all connections. It represents the two connected data points, their technology they are based on and the direction of the connection. An example for the overview is shown in Figure 215.



Datapoint	Tech	Dir	Tech	Datapoint	Connection
LINX-201.User Registers.reg0_Read	Reg	→	BACnet	LINX-201.BACnet Port.Datapoints.ai0	ai0 (10CB)
LINX-201.BACnet Port.Datapoints.ai0	BACnet	←	Reg	LINX-201.User Registers.reg0_Read	ai0 (10CB)
LINX-201.User Registers.reg1_Read	Reg	→	BACnet	LINX-201.BACnet Port.Datapoints.ai1	ai1 (10CC)
LINX-201.BACnet Port.Datapoints.ai1	BACnet	←	Reg	LINX-201.User Registers.reg1_Read	ai1 (10CC)
LINX-201.User Registers.reg2_Read	Reg	→	BACnet	LINX-201.BACnet Port.Datapoints.ai2	ai2 (10CD)
LINX-201.BACnet Port.Datapoints.ai2	BACnet	←	Reg	LINX-201.User Registers.reg2_Read	ai2 (10CD)
LINX-201.User Registers.reg3_Read	Reg	→	BACnet	LINX-201.BACnet Port.Datapoints.ai3	ai3 (10CE)
LINX-201.BACnet Port.Datapoints.ai3	BACnet	←	Reg	LINX-201.User Registers.reg3_Read	ai3 (10CE)

Figure 215: Connections Summary.

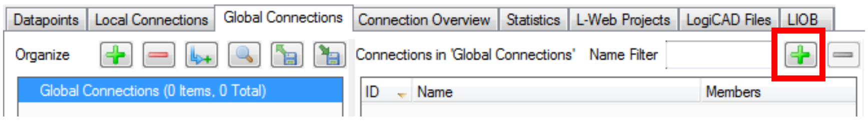
7.10.7 Create a Global Connection

Global connections are an easy way to publish or subscribe to global data, which is distributed among devices. To configure such communication, a device needs to be member of a CEA-852 channel. Once it is member of that channel, global connections need to be configured in the data point configuration. This is similar to creating local connections and most of the configuration steps apply also for global connections. In addition, also network timing parameters can be configured for global connections. For a description see Section 6.4.4.

If other devices already have global connections configured that publish data on the network, their definition can be exported and imported into the new configuration. That will make all the global connection names available. When creating manually, the connection names can be edited.

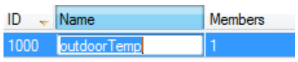
To Create a Global Connection

1. Click on the Global Connections tab



2. in the main window and press the speed button **Create new Connection**. A new connection is added to the connection list.

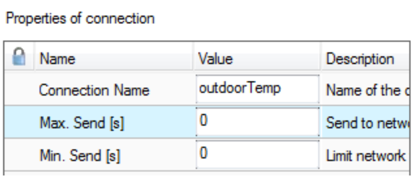
3. Define a name for the global connection. This name is required to be unique on the network. Data will be published or subscribed to under this name.



4. Add data points to the global connection as described in Section 7.10.3. As a default, output data points will be added as sending, input data point as receiving data under the global connection.

5. Change the direction by modifying **send** or **receive**. For changing multiple data points use multi-select.

6. Define timing parameters for a global connection that is sending out data. On the global connections tab of the main window the connection properties are listed below the data point member list. In the tab of the property area click the button .



7. To export the definitions of the created global connections, click the button **Export connections to disk** and choose the XML format.

8. To use those definitions, click the button **Import connections from disk** and choose an exported connections XML file. This creates the global connections structure with connection names but without any data points. Data points can then be added, for example, via drag-and-drop.

7.10.8 Automatic Generation of Connections

The *smart auto-connect* feature of the Configurator provides a quick way to automatically generate target data points out of a source data point selection and generate connections to them. Using this feature a gateway interface is generated with a few mouse clicks.

To Auto-generate Data Points and Connections

1. Go to the **Datapoints** tab.

2. Select those data points of a given port folder that shall be mapped to another technology. The methods innclude sub-folders, data point name filter and multi-select may be used for doing this.

3. Click on the speed button **Generate and connect selected** in the tool bar.

4. Alternatively, you can select the port folder or any sub-folder and click the speed button **Folder-wide Generate points and auto-connect** in the tool bar. This generates target data points and connections for all data points in the folder.

5. The auto-generate preview dialog opens as shown in Figure 216. Choose the target technology. The preview results show a list for each source type found how it will be created as a target type. The **Type Name** column provides a drop-down to modify the result. This choice is remembered and will be applied the next time again. You may click the **Restore defaults** button to revert all custom settings.

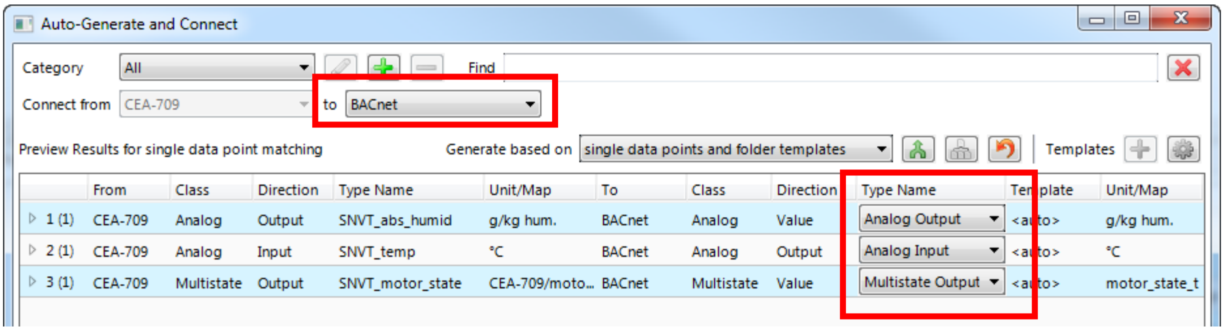
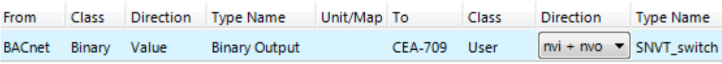


Figure 216: Auto-generate and connect preview.

Note: *The respective port may have to be enabled first in the project settings to have the target technology available in this dialog.*

6. If the selected target technology offers choices on the direction to create, choose one of the offered directions.



7. Structured data points will be flattened in some target technologies. To prevent this from happening, click the **Don't expand structured data points** button. Note, that this may require an auto-generate template, which defines how to map this structured data point.

8. Select an auto-generate template in the drop-down box of the **Template** column as shown in Figure 217.

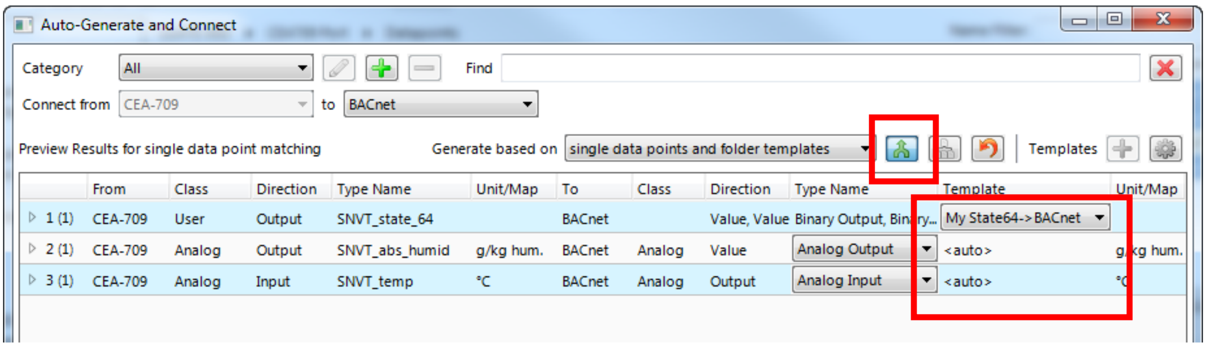


Figure 217: Auto-generate and connect with auto-generate template.

9. After having reviewed all types, click the **Generate** button.

Note, when auto-creating the target data points, the Configurator initializes their properties with default values derived from the properties of the source data points. In particular, the data point name, description, minimum and maximum value, and engineering units are generated. If the default properties do not have the desired values, the user can edit them in the target folder. The user can also craft an auto-generate template where those properties are pre-set.

To auto-generate target data points into multiple technologies from the same source data points, execute the auto-generate and connect function multiple times on the source data points. Select different target technologies in sub-sequent auto-generate actions. The new data points are added to existing connections. This makes it easier to keep track of auto-generated local connections involving multiple technologies.

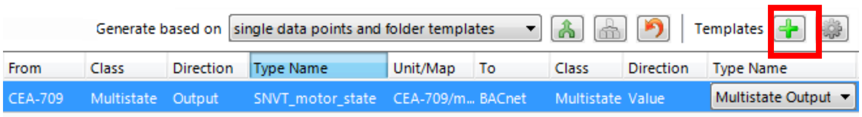
7.10.9 Create an Auto-Generate Template

If the implicit options for generating target types are not sufficient for the envisioned task, a specialized auto-generate template can be crafted. This template contains copies of the source data points as a starting point. The user can then create instances of the desired target data points and use one or more connections (e.g., a multi-slot connection for sub-data points of a structured source). The target data point name and description can contain variable placeholders, which expand to the name and description of the actual sources, which the auto-generate template will be applied on. If required, one can add conversion adaptors including math blocks. The entire configuration serves as a template on how to generate the target data points and apply the appropriate connections. Once saved in the template library, it will be available for selection in the preview dialog.

To Create an Auto-Generate Template

1. Select the source data points and invoke auto-generation as described in Section 7.10.8.

2. In the preview dialog select the source type for which a new template shall be created and click the **Create template for selected source** button.



3. The auto-generate template editor opens as shown in Figure 218. It contains two data point lists. The left-hand (denoted ‘1’) list is pre-filled and contains the selected source data points. They are locked and cannot be modified. The right-hand list (denoted ‘2’) is intended for creating the desired target data points.

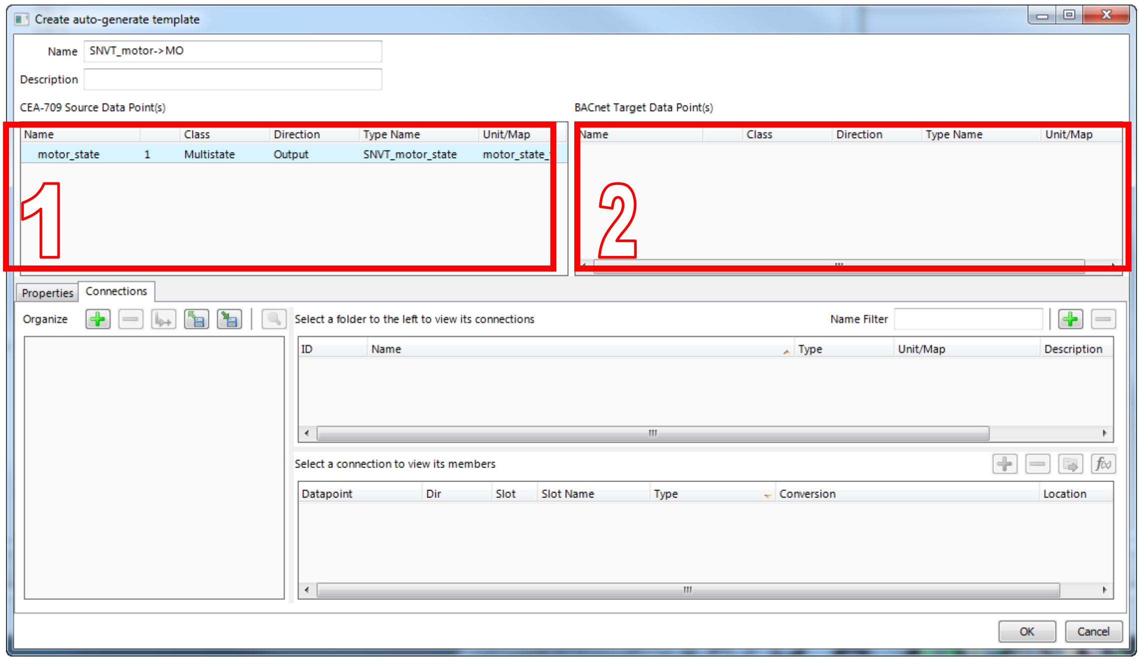


Figure 218: Auto-generate template editor.

4. Enter Name and Description, which is used later to select the auto-generate template.

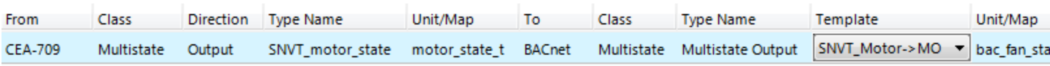
5. Right-click in the target data point list and select **Create Data Point …**. The creation dialog of the respective target technology opens. Create the desired target data points, e.g., a BACnet 'MO' data point with a custom multi-state map.

6. Edit the target data point name to use a variable placeholder for the target data point name such as `%{name}`. Insert the placeholder by selecting the desired entry of the **add var** drop-down list. When applying the auto-generate template, the placeholder is expanded to the name of the actual source data point. Choose `%{path}%{name}` to flatten the folder tree of the source data point and include the path in the target data point name.

7. Drag and drop the source and target data points into the **Connections** tab below to create the needed connections. Add custom conversions to the connection items as needed.

8. Click **OK** to store the auto-generate template.

9. From now on it can be selected in the **Template** column and applied to the source.



7.10.10 Create a Complex Auto-Generate Template

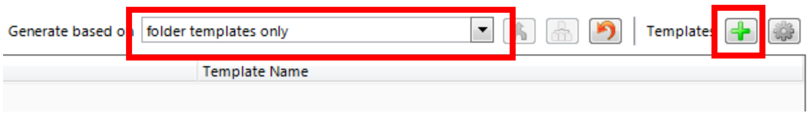
Simple auto-generate templates as described in Section 7.10.9 are based on single data point sources. These can be scalar or structured data points. In any case the decision, which auto-generate template applies, is based on that single data point. If a given set of source data points shall generate another specific set of target data points, so-called *complex* auto-generate templates can be used. These are based on folders that contain the described data points (i.e., name and types must match). With a complex auto-generate template the entire folder is used as a source and an entire target folder will be generated with the target data points defined by the complex auto-generate template.

For example, there are device folders of similar devices on the BACnet port. These shall serve as the connection source. One such device folder contains three analog data points named ‘TempComfort’, ‘TempNight’, ‘RoomTemp’ that have ‘°C’ as engineering units. These shall generate two target data points, one a structured data point with the setpoints ‘TempComfort’ and ‘TempNight’ connected (e.g. a SNVT_temp_setpts) and one data point with the ‘RoomTemp’ connected (e.g., a SNVT_temp). A complex auto-generate template is created based on one of the device folders.

To Create a Complex Auto-Generate Template

1. Select the source folder and invoke auto-generation as described in Section 7.10.8 using the button **Folder-wide Generate points and auto-connect** in the tool bar.

2. In the preview dialog choose generation based on **folder templates only**. The list will then be empty as no such template yet applies. Then click the **Create template for selected source** button.

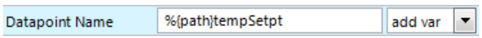


3. The auto-generate template editor opens as shown in Figure 218. It contains two data point lists. The left-hand (denoted ‘1’) list is pre-filled and contains the selected source data points. They are locked and cannot be modified. The right-hand list (denoted ‘2’) is intended for creating the desired target data points.

4. Enter **Name** and **Description**, which is used later to select the auto-generate template.

5. Right-click in the target data point list and select **Create Data Point …**. The creation dialog of the respective target technology opens. Create the desired target data points, e.g., a SNVT_temp_setpts and a SNVT_temp.

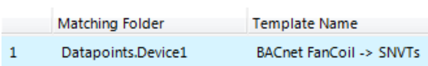
6. Edit the target data point name and local NV name. Since the source data point names will be equal for all source folders, an additional component may be added to the target data point name. Use the variable placeholder `%{path}` in the name to make it unique. Insert the placeholder by selecting the desired entry of the **add var** drop-down list. When applying the auto-generate template, the path placeholder is expanded to the actual folder path relative to the **Datapoints** folder.



7. Drag and drop the source and target data points into the **Connections** tab below to create the needed connections. Add custom conversions to the connection items as needed.

8. Click **OK** to store the auto-generate template.

9. From now on it can be selected in the **Template** column and applied to matching source folders.



7.10.11 Managing Connection Resources

All described resources for connections, including connection adaptors and auto-generate templates are part of the configuration project. When opening the project file on another PC, all project resources will be merged into the local resource repository. After opening a project file, all its resources are therefore available to new projects.

The Configurator provides a resource manager, that can be used to view, edit, import and export connection resources. Select the menu **Tools → Manage Connection Adaptors …** or **Tools → Manage Auto-Generate Templates …** to open the resource manager dialogs.

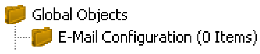
7.11 E-mail Templates

7.11.1 Create an E-mail Template

E-mail templates are used to assemble and transmit e-mails when certain trigger conditions occur. The e-mail template contains the destination e-mail address, the subject, and text. Variable parameters can be added to the text by using data point sources. The transmission of an e-mail is triggered by one or more trigger data points. For setting up e-mails, the e-mail account information has to be configured on the device, e.g., on the Web UI (see Section 5.2.24).

To Create an E-mail Template

1. Under the **Global Objects** folder, select the **E-mail Configuration** sub-folder.



2. Right-click and select **New E-mail Template …** from the context menu.

3. In the **Configure E-mail Template** dialog, which is shown in Figure 219 enter the **To** address and the **Subject**. Optionally, **Cc** and **Bcc** addresses can be specified.

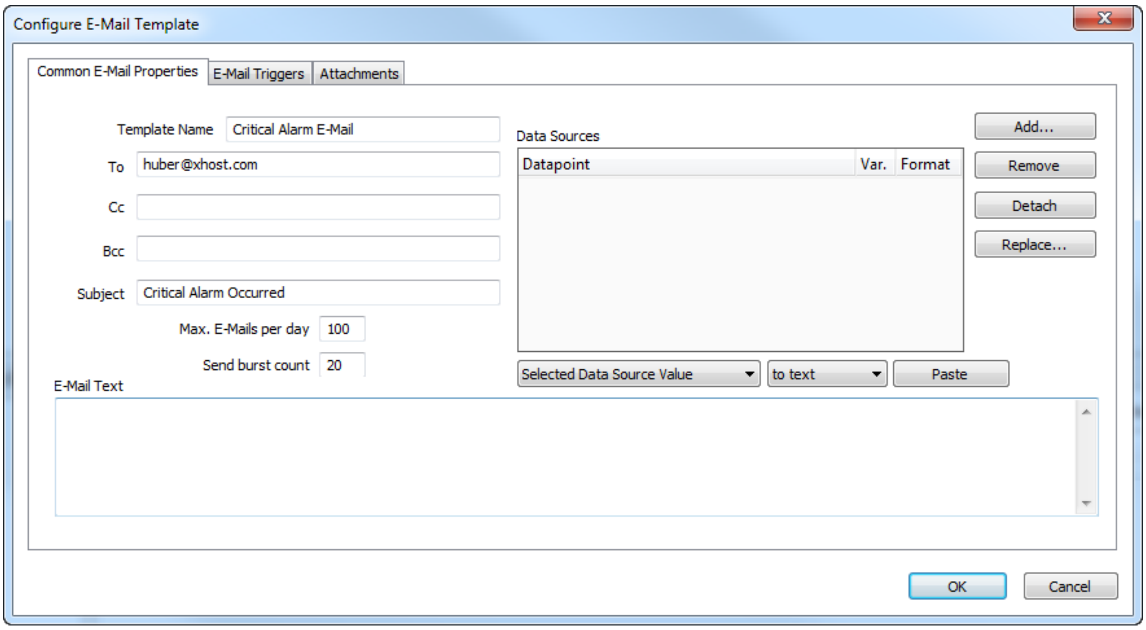
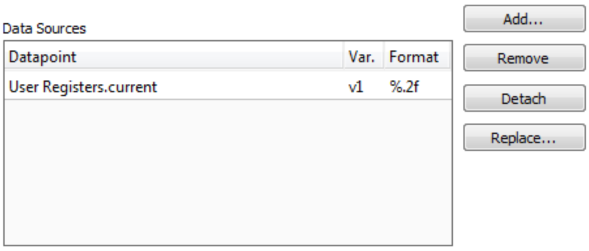


Figure 219: Configure E-Mail Template Dialog.

4. Enter text in the **E-mail Text** multi-line field.

5. If the e-mail text shall contain values of data points, add data points to the **Data Sources** list by clicking the **Add…** button.

6. A data point selector dialog opens. Select one or more data points and click **OK**. The selected data point appears in the **Data Sources** list.



7. If necessary, edit the **Format** string. The default '%.2f' will format the value as a floating point with 2 decimals.

8. Select the data point in the **Data Sources** list. In the drop-down box underneath select **Selected Data Source Value**, in the next drop-down select **to text** click the **Paste** button. Variables can also be pasted to the subject line or any of the address lines.

9. A place holder `%{v1}` for the data point value appears now in the e-mail text.

10. To replace an existing data source select the data point in the **Data Sources** list and click the **Replace…** button. This opens a data point selector dialog for choosing the replacement data point.

7.11.2 Trigger E-mails

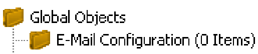
E-mail templates are used to assemble and transmit e-mails when certain trigger conditions occur. For an e-mail template, one or more trigger conditions can be defined. The e-mail will be sent, when one of the trigger conditions is activated. Depending of the trigger data point type, the trigger conditions can be refined.

Note, that the behavior of the trigger data point is influenced by the COV properties of the data point. If the **Only notify on COV** property is checked, the data point triggers only if its value changes to the value of the trigger condition. If that property is not checked, the data point triggers on every write with a value that matches the trigger condition.

The trigger for sending an e-mail can be enabled or disabled altogether by using an *enable* data point. This data point must be of type *binary*. If the value of that enable data point is TRUE, the trigger conditions are evaluated. If the value of the enable is FALSE, no e-mails are be triggered.

To Create an E-mail Trigger

1. Under the **Global Objects** folder, select the **E-mail Configuration** sub-folder.



2. Right-click and select **Configure E-mail Template …** from the context menu.

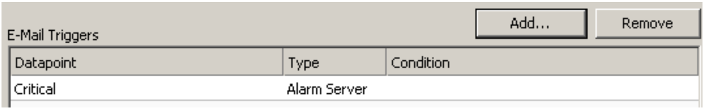
3. Change to the **Mail Triggers** tab.

Note: *Of course, you can also change directly to the **Mail Triggers** tab when creating an e-mail template.*

4. Click the **Add…** button. A data point selection dialog opens.

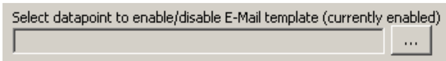
5. Select one or more data points and click **OK**.

6. The triggers appear now in the **E-Mail Triggers** list. The data points that serve as e-mail triggers also appear with the e-mail icon in the data point list.



7. In the **Manage Trigger Conditions** you can setup the trigger condition depending on the trigger data point class.

8. If the trigger condition is depending on the value of an enabling data point, you can add an enable data point by clicking on the **…** button.



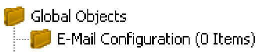
9. To remove such a trigger enable, click the **Remove Enable Trigger** button.

7.11.3 Attachments

E-mail templates can be configured to have file attachments. Basically, any file of the device can be specified as an attachment.

To Configure Attachments

1. Under the **Global Objects** folder, select the **E-mail Configuration** sub-folder.



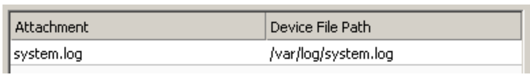
2. Right-click and select **Configure E-mail Template …** from the context menu.

3. Change to the **Attachments** tab.

Note: *Of course, you can also change directly to the **Attachments** tab when creating an e-mail template.*

4. Select an available file from the **Attach File** drop-down box.

5. Click the **Add** button. The file appears in the **Attachments** list.



6. To remove an attachment, select the attachment file in the **Attachments** list and click the button **Remove**.

7.11.4 Limit E-mail Send Rate

The transmission of e-mails is triggered by the configured trigger conditions. It is not predictable, how often the trigger condition will cause the transmission of an e-mail. The e-mail template can be configured to limit the number of transmitted e-mails. This is done in the Configure E-mail Template dialog.

To configure an E-mail Rate Limit, configure the settings:

- **Max. E-mails per day**: This setting defines how many e-mails can be sent on average per day. The actual number of transmitted e-mails on a specific day may be slightly higher than this setting, depending on burst rates. The default is 100 e-mails per day. This results in an average interval of one e-mail per 14 minutes.
- **Send burst count**: This setting defines how many e-mails may be transmitted shortly after each other not limited by the above average interval. After the burst count, the average mails per day limit takes effect. The default is a maximum of 20 e-mails in a row.

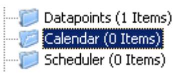
7.12 Local Schedule and Calendar

7.12.1 Create Calendar Patterns

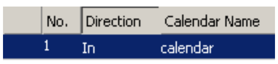
When working with global definitions for calendar-based exception days such as “Holidays” a local calendar is used and needs to be configured with calendar patterns. A calendar pattern represents a class of days such as “Holidays”. The calendar patterns can then be used in a schedule to define daily schedules for exception days. The available calendar patterns should be created when the system configuration is engineered. The actually dates in the calendar patterns can be modified later at run-time.

To Create a Calendar Pattern

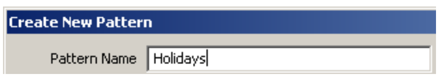
1. Locate the calendar object. When using a generic scheduler the corresponding generic calendar already exists in the **Scheduler** folder under the device folder. For a technology calendar, select the **Calendar** sub-folder of the respective port.



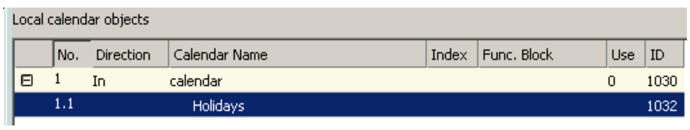
2. Select the calendar data point.



3. Right-click and select **Create Calendar Pattern…**
4. Enter a Pattern Name in the **Create Calendar Pattern** dialog



5. Click **Create Pattern**. The dialog closes and the calendar pattern appears beneath the calendar data point.

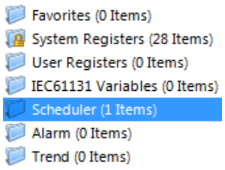


7.12.2 Create a Local Scheduler

For scheduling data points, a scheduler object must be created. Under each port folder, multiple local scheduler objects can be created. These local schedulers can then be configured to schedule data points.

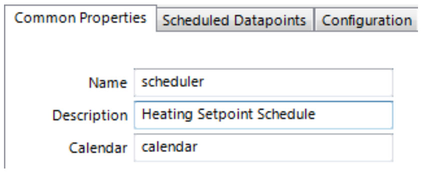
To Create a Local Scheduler

1. Under the device folder, select the **Scheduler** sub-folder to create a generic scheduler. For a technology scheduler, select the **Scheduler** sub-folder of the respective port.



2. Right-click in the data point list view and select **New Local Scheduler …**.

3. Enter a name for the schedule and a description. Note, that the schedule automatically detects a calendar, if it has previously been created.



4. Click **Create Schedule**. The new schedule appears in the data point list of the Scheduler sub-folder.

7.12.3 Configure Scheduled Data Points

When a local scheduler has been created, it needs to be configured, which data points it shall schedule. This is done by attaching data points to the scheduler. Note, that there may be limits, how many and which data points may be attached (see Section 6.7.3).

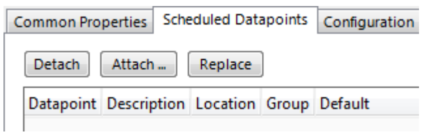
This configuration must be done as an initial setup. The scheduled data points and daily schedules can be changed later in the Web UI or over the network.

To Attach Data Points to a Scheduler

1. Select the scheduler data point in the Scheduler sub-folder.

2. Right-click and select **Configure Schedule** from the context menu. The same dialog which appears when a new scheduler is created is shown and allows configuring the scheduler. Of course, this step can also be done directly when the data point is created.

3. Select the tab Scheduled Datapoints.

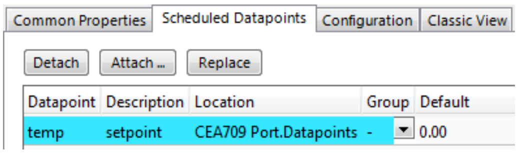


4. Click the button **Attach …** . This opens another data point selector window.

5. Select the data points to attach and click **OK**. For each of the attached data points, one or more lines appear in the list below the attach button. If the attached point is a structure, there will be one line for each element of the structure.

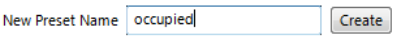
Tip!	*Data points can also be attached to a scheduler by selecting a data point in the data point manager, drag it onto a scheduler data point and drop it on the scheduler data point.*

6. Enter a short text in the **Description** field in the second column of each line. This text will serve as a label, which will be shown on the device's UI to identify the data point.



Datapoint	Description	Location	Group	Default
temp	setpoint	CEA709 Port.Datapoints	-	0.00

7. Add new value presets by entering a name and pressing the **Create** button next to the input field.



Tip! *To generate presets automatically for multi-state data points, click the **Auto-Create** button. This button is available, if no other presets have been defined yet.*

8. For each new preset, a new column will appear in the list. In this column, enter the desired value for each of the attached points, which will be set when this value template is scheduled. The user may later edit the values for each preset on the device but cannot add new value presets unless there is only one line (one value) in the list.

Datapoint	Description	Group	Default	day	night
NV_bac_lonCtrlnvo12_temp	temp	1	0.00	21.00	16.00

9. If there are multiple output values which belong together, they can be grouped in order to save space on the device. For each group, the entered value is stored only once, even if there are more data points in the same group.

Datapoint	Description	Group	Default	day	night
NV_bac_lonCtrlnvo12_temp	temp	1	0.00	21.00	16.00
NV_bac_lonCtrlnvo13_temp	temp	1	0.00	21.00	16.00

10. When done with the point and value setup, switch back to the **Configuration** tab or click **Save Changes** to leave the dialog.

Tip! *A shortcut to creating a scheduler object and attaching a data point is to select a data point in the data point manager, right-click on it and choose **Schedule Datapoint** from the context menu. This generates a scheduler and links that data point to it.*

7.12.4 Configure Scheduled Events

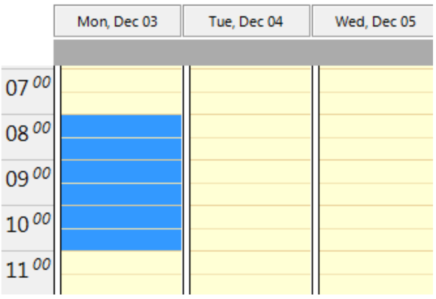
Once a scheduler is configured with attached data points and value presets, the schedule events can be defined. This can be done on the device or over the network at run-time, or also in the configuration software. A schedule consists of a number of scheduled events, each event starts at a defined time, has a scheduled value and ends at a defined within the same 24-hour period (starting at 00:00 and ending at 23:59 hours). The event can be configured to occur at a given date, for each weekday, or for a number of recurring dates.

In addition, scheduled events can be configured to occur on exception days from a calendar, such as "Holidays". An exception day always overrides a normal weekday. If more than one exception day is used, a priority must be assigned. This is necessary so that the system knows which schedule to follow on a day which matches more than one calendar pattern. Considering the priorities the calendar preview shows the effective schedule on a given day.

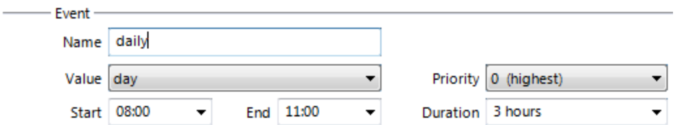
To Configure a Scheduled Event

1. Open the **Configure Schedule** dialog and click on the **Configuration** tab (see Section 7.12.3).

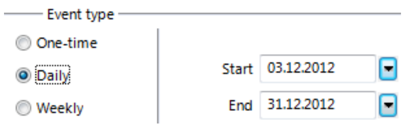
2. In the calendar view select the day for which to configure the scheduled event. Then select the event duration by clicking into the daily planner and dragging the mouse to the desired duration.



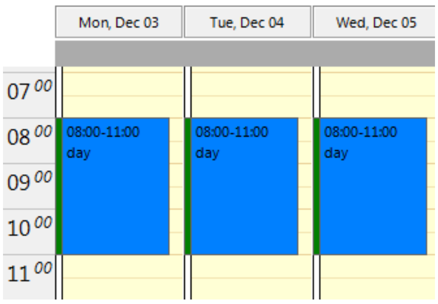
3. Then click on the button **Create new scheduled event** and enter a **Name** for the scheduled event (note, in BACnet there is no name to be specified). Choose the scheduled **value** or enter a scheduled value. Modify the **Start** and **End** time to your needs. Optionally you should set a priority, if scheduled events overlap in the preview.



4. Choose an **Event type**, which defines how the event is recurring.



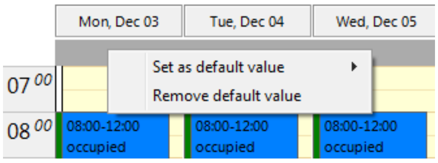
5. The click **OK**. The new event appears in the daily planner.



6. For viewing more details on overlapping events you can change to the **day view**. In all views events can be extended or shrunk using the upper or lower boundary handles or moved by clicking and dragging.

7. Right-click on an event in order to edit it. You can choose to modify it, change its color, or delete it.

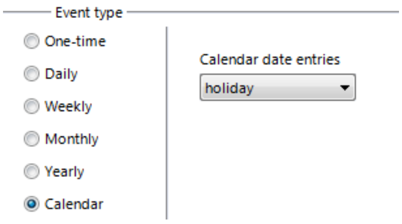
8. To set a default value, right-click onto the grey area right above the day planner. In the context menu choose a default value.



To Use Exception Days from a Calendar

1. For letting a scheduled event occur on exception days from a calendar, create a new scheduled event.

2. Select the event type **Calendar**. and choose one of the defined **calendar date entries**.



3. Note, that if the scheduled event may overlap with other events. In this case edit the priorities of the scheduled events. For example, if a given calendar day falls in both categories, “Holidays” and “Maintenance”, the scheduled event with the higher priority becomes effective. The highest available priority is marked **highest**. Note that the actual priority values depend on the technology (see Section 6.6.5).

Important!	***Choose different priorities for different scheduled event. If two scheduled events overlap and their priorities are equal, it is not determined, which value is in effect.***

7.12.5 Configure Exception Days

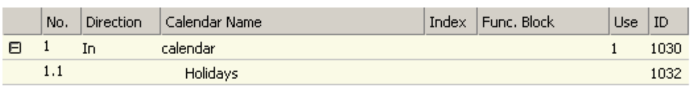
When a local calendar is used, its calendar patterns need to be configured with exception days (pattern entries). The calendar patterns can be configured in the Configurator software, modified at run-time over the Web UI or over the network. When configuring in the software, the current exception days should be uploaded from the device, to work on the current configuration.

To Configure Exception Days in a Calendar Pattern

1. Click on the Upload calendar/scheduler configuration button

in the tool bar of the main connections window. Click **OK** when the upload is finished.

2. Select the **Calendar** sub-folder and select the calendar pattern, which shall be configured



3. Right-click and select **Configure Pattern …** in the context menu.

4. The **Configure Pattern** dialog appears as shown in Figure 220. Add dates to the calendar pattern by entering a Date Configuration. Then click **Add Entry**. The date appears in the Pattern Entries list on the right-hand side.

5. Edit an exception by selecting the pattern entry in the **Pattern Entries** list. Then modify the date configuration in the **Date Configuration** group box.

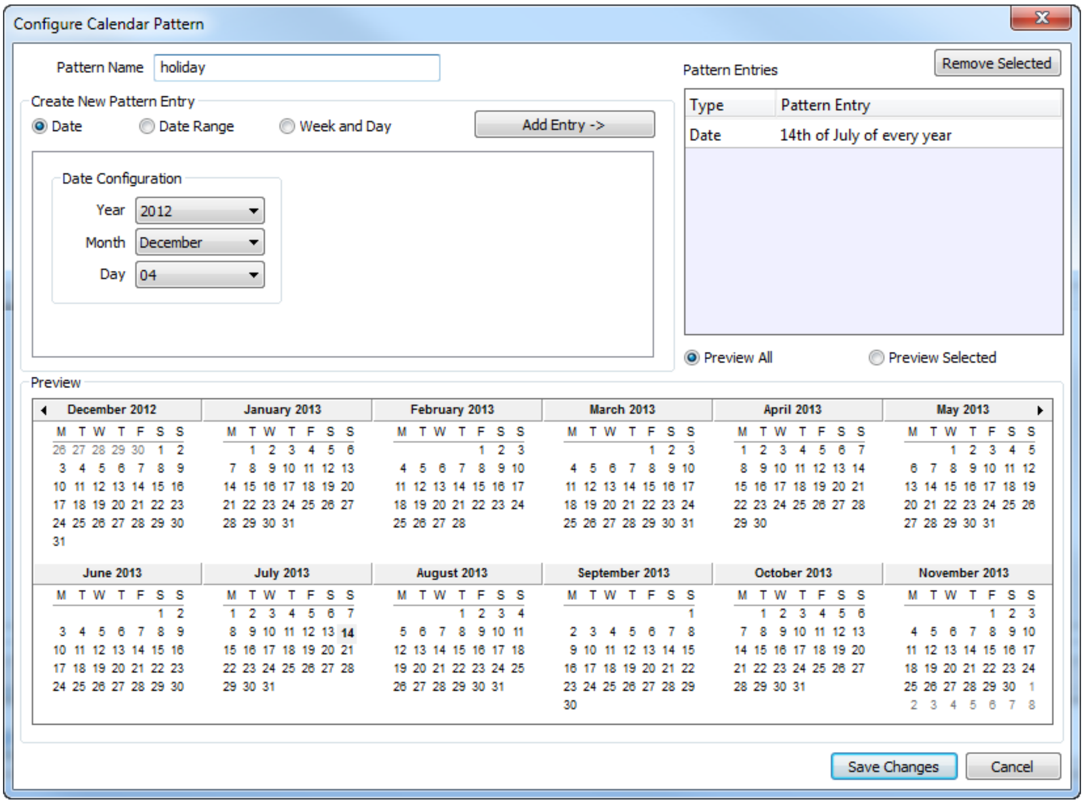


Figure 220: Configure Calendar Pattern Dialog.

6. Click **Save Changes** when all exception days have been entered.

Tip!	*If not sure, how a date configuration affects the calendar days, click on a pattern in the* ***Pattern Entries*** *list and the affected days will be highlighted in the* ***Preview****.*

7.12.6 Configure Control Data Points

A scheduler object can be configured to use special control data points. These data points can control the scheduler and expose additional state information of the scheduler on the network. The following control data points are available:

- **Scheduler Enable/Disable Datapoint**: This data point can be configured, which enables or disables the scheduler depending on its Boolean value.

- **Enable/Disable Feedback Datapoint**: This data point is updated with the current enabled state of the scheduler. This also reflects and an enable from the network.

- **Scheduled Preset Name**: This data point can be attached to be updated with the name of the currently active preset. Only string data points can be attached.

To Configure Control Data Points

1. Open the **Configure Schedule** dialog to configure daily schedules as described in Section 7.12.3.

2. Go to the Scheduled Datapoints tab.

3. In the **Control Datapoints** group box, click the ... button to add the desired control data point. A data point selection dialog opens.

4. Select a matching data point and click **OK**. For the preset name a string data point must be selected.

5. To remove an undesired control data point, click on the **Remove** button.

7.12.7 Using the SNVT_tod_event

On LOYTEC devices with the CEA-709 technology the SNVT_tod_event can be used in a schedule for implementing the next-state feature. The parts of this network variable contain:

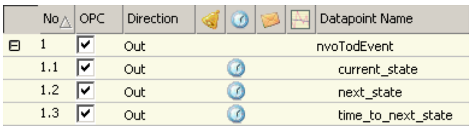
- Current state: This is the currently scheduled occupancy state.
- Next state: This is the next, future occupancy state in the schedule.
- Time to next state: This part reflects the time in minutes until the next state becomes active.

To Use a SNVT_tod_event

1. Create a SNVT_tod_event in the data point configuration.

2. Add the SNVT_tod_event to the scheduled data points of a scheduler as described in Section 7.12.3.

3. All three parts of the SNVT_tod_event are scheduled.



7.12.8 Using the Local Scheduler

Once the setup of the local scheduler is done, it is basically operational. It will immediately start working based on the configuration data downloaded through the configuration software. You can verify the daily schedules and values of scheduled data points on the Web UI (see Section 5.3.3). The local schedule can be altered over the Web UI or over the network using the underlying networking protocol.

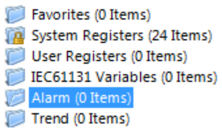
7.13 Local Alarming

7.13.1 Create an Alarm Server

To generate local alarms, an alarm server needs to be created at first. The local alarm sources will report alarms to that alarm server. The alarm server is the interface to access local alarms. This can be done over the network or the Web UI.

To Create an Alarm Server

1. Under the device folder, select the **Alarm** sub-folder to create a generic alarm server. For a technology alarm server, select the **Alarm** sub-folder of the respective port.



2. Right-click in the data point list view and select **New Alarm Server …**.

3. In the **Create New Alarm Server** dialog box (as shown in Figure 221) enter **Name** and **Description** of the alarm server.

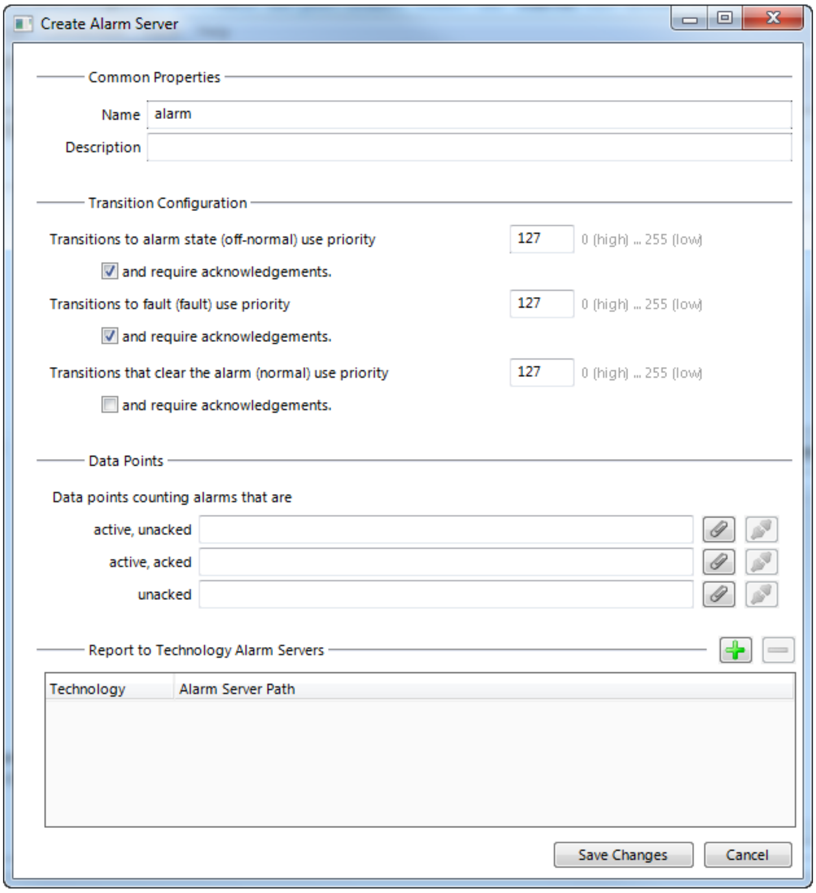


Figure 221: Create New Alarm Server dialog box.

4. For alarm transitions you may define, which require acknowledgement and at which priority they are reported.

5. You may attach data points for storing alarm counters. These will be linked using the respective property relations (see Section 6.2.12).

6. When you create a generic alarm server, you may add technology alarm servers, that shall be reported to. Click on [+] and choose one alarm server of each technology. If they don’t exist, you need to create them under the technology port folders.

7. Click **Create**. The alarm server appears now in the data point list view.

8. For a BACnet technology alarm server, edit the instance number of the Notification Class object to your needs.

7.13.2 Create an Alarm Condition

To generate alarms from data points, intrinsic reporting is used. For each data point an alarm condition must be defined. This condition employs an intrinsic algorithm to generate alarms based on the data point's value or by evaluating a feedback value. Depending on the data point type (analog, binary, multi-state), different conditions are defined. The alarm is reported to the attached alarm server.

To Create an Intrinsic Alarm Condition

1. Select a data point.

2. Right-click and select **Create Alarm Condition…** from the context menu.

3. For the alarm condition edit the following definitions, which apply to all condition types as shown in Figure 222. Select the **Alarm Server** which the alarm shall be reported to. Typically, you will choose a generic alarm server.

4. Enter an **Alarm Message**. This is shown when the alarm becomes active. You may add variable placeholders to this message by selecting one from the drop-down box **add var** on the right-hand side. Enter a **Clear Message**. This is shown when the alarm clears.

5. Check the option **Enable Fault Alarms**, if fault conditions (offline, unreliable) shall generate fault alarms. If enabled, enter a **Fault Message**, which is displayed along with the fault alarm when it occurs.

6. Optionally, enter a **Time Delay**, for which the condition must persist before the alarm becomes active or is cleared again. The delay is entered in seconds.

7. By clicking you may attach a data point, which is evaluated for enabling the alarm. This can also be done by editing the property relation 'enableAlarm' (see Section 7.2.8). Detach the data point by clicking .

8. Choose the option **value is different from** to define a feedback alarm. In this case the setpoint value of the alarmed data point is compared against the feedback value. A feedback data point can be attached for this purpose. This can also be done by editing the property relation 'feedbackRelation' (see Section 7.2.8).

9. Choose the option **value has condition** to define a value alarm. In this case the data point value is compared against the condition. Edit the condition in the box below this option.

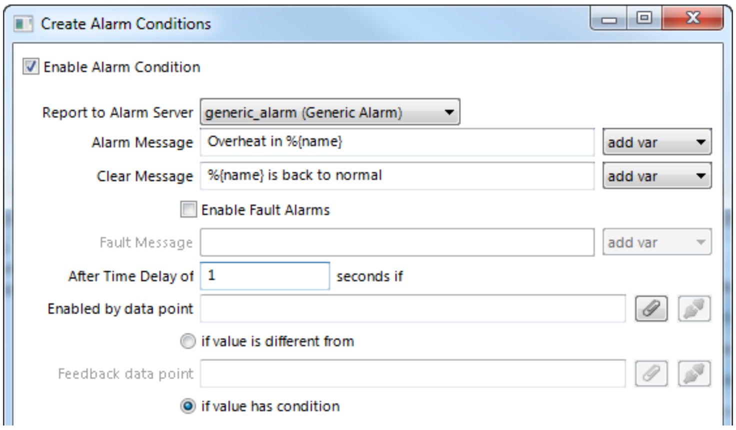


Figure 222: Common settings for an alarm condition.

10. For an analog feedback condition fill in the alarm condition as shown in Figure 223. A feedback alarm is generated, if the setpoint **differs by –** and **differs by +** value from the feedback value. Enter a **Deadband** to account for hysteresis. Attach or detach data points for those limits. This can also be done by editing the property relations 'lowLimit', 'highLimit', and 'deadband', respectively (see Section 7.2.8).

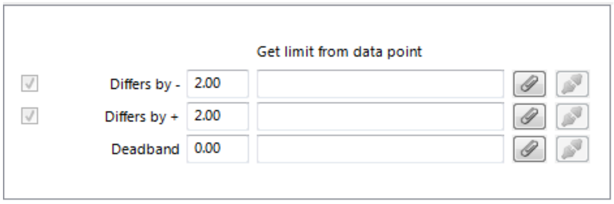


Figure 223: Condition for an Analog Feedback Alarm.

11. For an analog value condition fill in the alarm condition as shown in Figure 224. Select **Low Limit** and **High Limit** and put check marks, if they shall be employed. Enter a **Deadband** to account for hysteresis. Attach or detach data points for those limits. This can also be done by editing the property relations 'lowLimit', 'highLimit', and 'deadband', respectively (see Section 7.2.8).

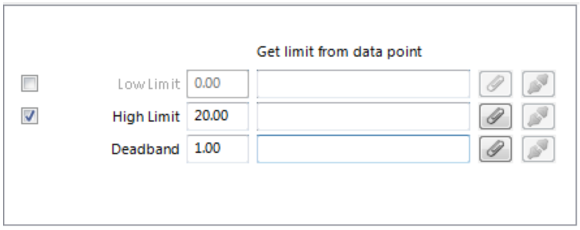


Figure 224: Alarm Condition for an Analog Value Condition.

12. For a binary data point define an alarm value in the alarm condition as shown in Figure 225. Select the **Alarm Value** which triggers the alarm.

Figure 225: Alarm Condition for a Binary Data Point.

13. For a multi-state data point define the alarm values in the alarm condition as shown in Figure 226. Select the states in the list **Not Alarmed** and move them to **Alarm on States** by clicking the arrow buttons.

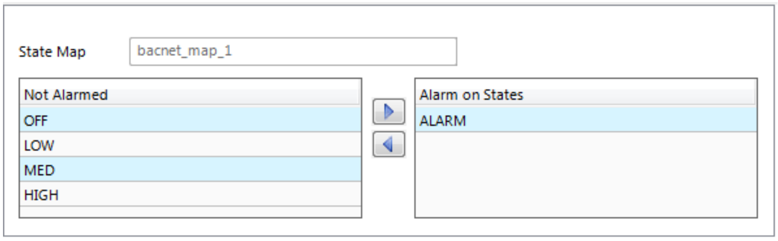


Figure 226: Alarm Condition for a Multi-State Data Point.

14. Click on **Create**. In the alarm column, the alarm sign will be added for those data points that have an alarm condition. If a sub-data point has been alarmed, the top-level data point will indicate this with the sign .

7.13.3 Deliver Alarms via E-mail

Updates in the alarm summary of an alarm object can be used as a trigger to send e-mail. For setting up e-mails, the account information has to be configured on the device, e.g., on the Web UI (see Section 5.2.24). Then an e-mail template can be created and the alarm point attached as a trigger. The e-mail template can be configured to contain certain alarm information in the text or subject field.

To Create an E-mail Template for Alarms

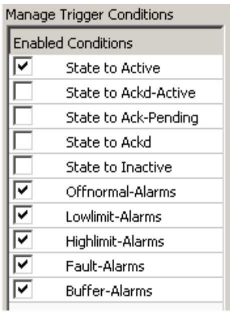
1. Create or configure an e-mail template as described in Section 7.11.1.

2. Change to the **Mail Triggers** tab.

3. Click the **Add…** button and select an alarm data point.

4. In the Mail Triggers list select the added trigger data point.

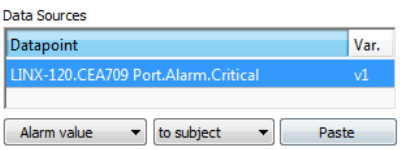
5. In the **Manage Trigger Conditions** list put a check mark on alarm conditions that shall invoke the transmission of the e-mail.



6. Change to the Common E-Mail Properties tab.

7. Add the alarm data point as a data source as described in Section 7.11.1.

8. Choose the desired alarm information from the fields in the drop-down **Selected Data Source Value** and paste a place holder into the e-mail text or subject field.



7.13.4 Create an Alarm Log

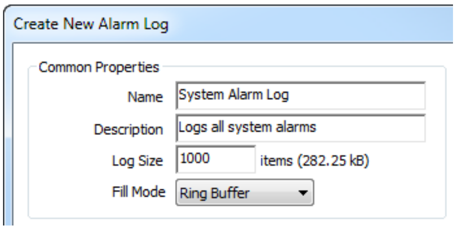
The alarm objects on the device contain an alarm summary (live list) of currently active and acknowledge-pending alarms. As soon as an alarm becomes inactive and has been acknowledged, it disappears from the alarm summary. To store a historical log of alarm transitions an *alarm log* needs to be created.

An alarm log can log transitions of one or more alarm objects. Its size is configurable. The alarm log can be operated as ring or as linear buffer. When operated as ring buffer, the oldest alarm log records are overwritten by new alarm transitions, as soon as its size limit is reached. When operated as linear buffer no more alarms are logged once the log is full.

To Create an Alarm Log

1. Under the Global Objects folder, select the Alarm Log Object Configuration sub-folder.

2. In the data point list right-click and select **New Alarm Log …** from the context menu.



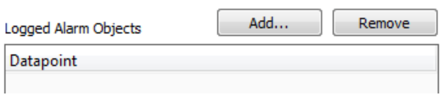
3. In the **New Alarm Log** dialog enter a **Name** for the alarm log. Optionally enter a **Description**.

4. Enter a **Log Size**, which defines how many transitions are resident in the alarm log.

5. Select the desired **Fill Mode**. The default is **Ring Buffer**, which lets the log overwrite old records once it reaches its capacity. Select **Linear Buffer**, if recording shall be stopped in this case. The user has to purge the log before it continues recording.

6. Define a percentage for **Fill Level Notification**, which can be used to trigger the transmission of E-Mails.

7. Click on the button **Add…** on top of the **Logged Alarm Objects** list.



8. A data point selector dialog opens. Select one or more alarm objects that shall be logged and click **OK**. The alarm objects appear in the list.

9. Click **Create** to create the alarm log object.

7.13.5 Multi-Edit Alarm Conditions

For editing a large number of alarm conditions, some multi-edit features are available to assist in this task. On a multi-selection of data points, the user can execute two options from the context menu:

- **Configure Alarm Conditions**: Use this option from the context menu on a multi-selection of data points. The alarm condition of all selected data points can be edited. If all selected data points are of the same class (e.g., all analog data points) the alarm condition can be fully specified. Note that these settings will be applied to all data points. In alarm messages use variable place holders. For assigning different limit and enable data points use the manage relations tab (see Section 7.2.8).

- **Configure Alarm Messages**: Use this option from the context menu for editing alarm messages (alarm, clear, fault) for all data points in the selection. A dialog with a spreadsheet view appears as shown in Figure 227. Edit the messages directly in the spreadsheet. Alternatively, click on the export button to export the spreadsheet as a CSV file for editing in Excel. Import a CSV file with alarm messages in the menu **Tools → Import alarm messages**.

Note: *Use the feature to include data points from sub-folders and filter expressions to expand the ability to perform a multi-select.*

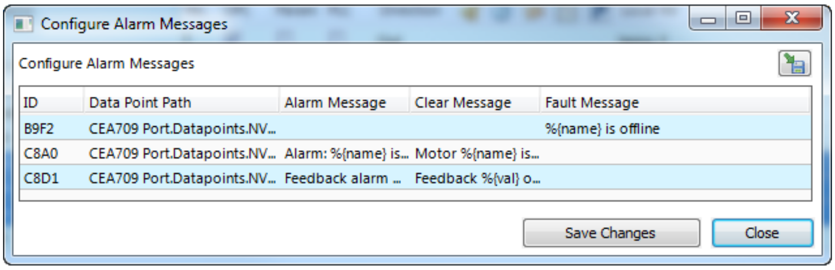


Figure 227: Spreadsheet for multi-edit of alarm messages.

7.14 Local Trending

7.14.1 Create a Local Trend

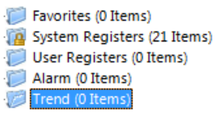
The value of a data point can be logged over time. This is referred to as trend data. To generate trend data a trend object has to be created. The trend data is stored in a data logger file. This file can be downloaded via FTP in binary or CSV format (see Section 18.1.2).

Trend objects can generate trend logs for single and multiple data points and can be operated in one of the following modes:

- **Interval Mode**: In this mode a snapshot of all trended data points is logged into the data logger file. Aligned log intervals can be configured.
- **COV Mode**: In this mode, each of the trended data points is logged separately, if and only if its value changes. For analog data points, a specific COV increment can be configured in the data point configuration properties of the trended data point.
- **Trigger Mode**: In this mode a snapshot of all trended data points is logged each time a trigger condition fires. The trigger condition is applied to a trigger data point.

To Create a Trend Object

1. Select the **Trend** folder of the device.



2. In the data point list right-click and select **New Trend …** from the context menu.

3. In the **Create New Trend Object** dialog (shown in Figure 228) enter a name and optionally a description for the trend log object.

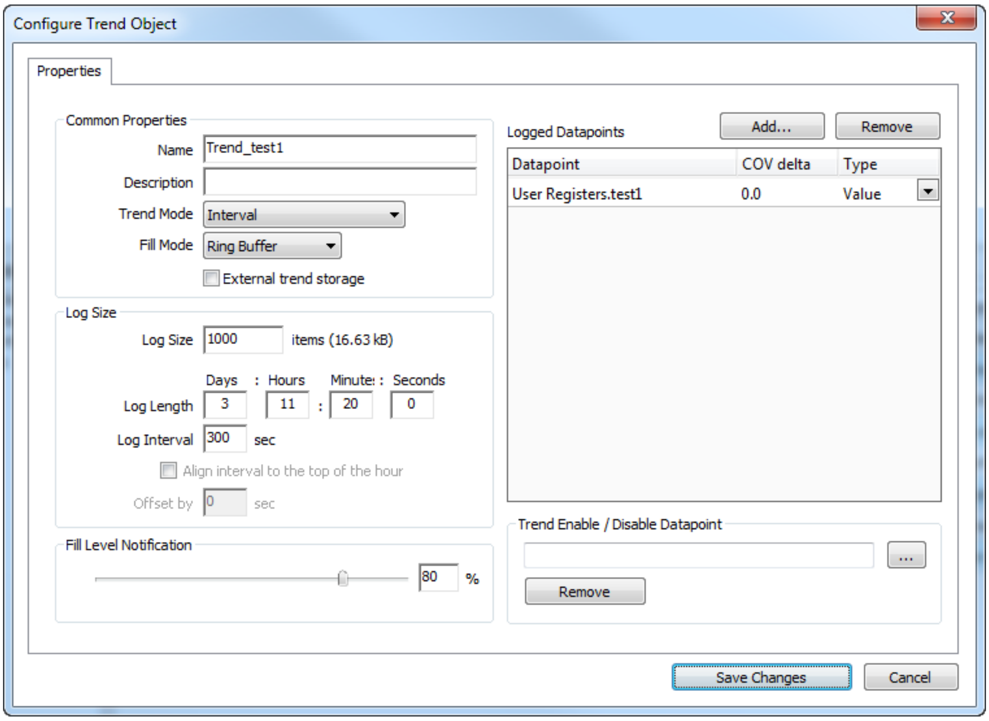


Figure 228: Basic Trend Object Configuration.

4. Select the desired **Trend Mode**.

5. On devices with SD cards, select **External trend storage**, if this trend log shall be backed up to an SD card. If doing so, also set the **Fill Level Notification**, which triggers when a backup is written to the external storage.

6. Select the **Log Size**. The display in the dialog will adapt the estimations for needed data logger file size in KB and duration of the trend log. Alternatively, for interval trends, the estimated log duration and log interval can be edited.

7. In the interval trend mode the **Align interval** option can be activated. Depending on the selected interval, the beginning of the interval is aligned to the wall time (e.g. every 15 minutes aligned to the top of the hour). An additional offset in seconds to that alignment can be specified (e.g. 5 seconds after those 15 minutes).

8. Select a **Fill Level Notification** percentage. This will decide at which fill-level trigger will fire. A fill-level trigger can be used to trigger the transmission of an e-mail (see Section 7.14.5) or a backup of log data to the SD card.

9. Click **Save changes** to store the basic configuration of the trend object. The new trend log object appears in the data point list of the Trend folder.

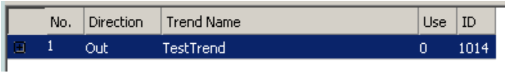
7.14.2 Configure Trended Data Points

When a local trend object has been created, it needs to be configured, which data points it shall log. This is done by attaching data points to the trend object. Only simple data points can be attached for trending, i.e., of class analog, binary, or multi-state. For trend log objects in the BACnet technology, single data points can be attached only.

The trending can be enabled/disabled on behalf of an *enable* data point. This data point should be of type *binary*. If the value of that enable data point is TRUE, the trend object logs data as defined by the trend mode. If the value of the enable is FALSE, trending is disabled. If no enable data point is configured, the trend log is always enabled.

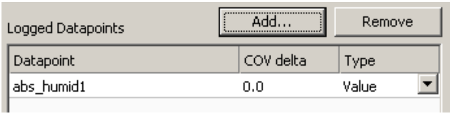
To Attach Data Points for Trending

1. Select the trend object in the **Trend** sub-folder.



2. Right-click and select **Configure Trend** from the context menu. The same dialog which appears when a new trend object is created is shown and allows configuring the trend object. Of course, this step can also be done directly when the object is created.

3. Add data points to be trended. Click on **Add …** which opens a data point selector window.



4. Select the data points and click **OK**. For each of the attached data points, a line appears in the list below the add button. The trended data points will also appear with the trend icon in the data point manager.

Tip!	*Data points can also be attached to a trend by selecting a data point in the data point manager, drag it onto a trend object and drop it on the trend object.*

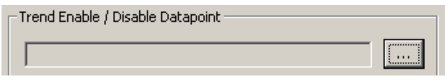
5. Data points can be removed from the trend by clicking **Remove**.

6. If COV mode was selected, the COV increment is displayed in the **COV delta** column. This value can be increased to produce less trend data. Note, that it cannot be lowered under the trended data point's own COV increment. Go to the data point configuration to change the COV increment in this case.

7. If the trended value of the data point shall be aggregated over the log interval, select the desired aggregation in the **Type** column. Available options are **Min**, **Max**, **Avg**.

Tip!	*For creating multiple curves with min, average, and maximum values, add the same data point three times and select the different aggregation types.*

8. In addition, a special **Trend Enable** data point can be selected. If configured, the trend log will only log data, if the value of this data point evaluates **true**, i.e., is not zero. Click the button to select a data point.



9. To remove the enable data point, click the **Remove** button.

10. When done with the data point setup, click **Save Changes** to leave the dialog.

Tip!	*A shortcut to creating a trend log object and attaching a data point is to select a data point in the data point manager, right-click on it and choose **Trend Datapoint** from the context menu. This generates a trend log and links that data point to it.*

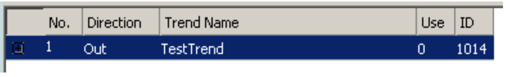
7.14.3 Trend Triggers

Local trend objects in CEA-709 can be operated in *trigger mode*. In this mode, one or more trigger data points cause the generation of a snapshot containing the values of the trended data points at the time instant the trigger is activated. For a trend object, one or more trigger conditions can be defined. Depending on the trigger data point type, the trigger conditions can be refined.

Note, that the behavior of the trigger data point is influenced by the COV properties of the data point. If the **Only notify on COV** property is checked, the data point triggers only if its value changes to the value of the trigger condition. If that property is not checked, the data point triggers on every write with a value that matches the trigger condition.

To Configure Trigger Data Points for Trending

1. Select the trend object in the **Trend** sub-folder.



2. Right-click and select **Configure Trend** from the context menu.

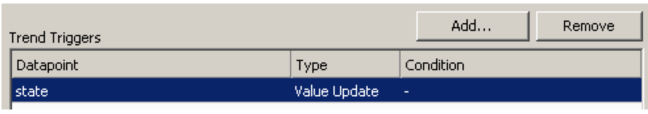
3. Change to the **Triggers** tab.

Note: *Of course, you can also change directly to the **Triggers** tab when creating a trend object.*

4. Click the **Add…** button. A data point selection dialog opens.

5. Select one or more data points and click **OK**.

6. The triggers appear now in the **Trend Triggers** list.



7. In the **Manage Trigger Conditions** you can refine the trigger condition depending on the trigger data point class.

8. When done with the data point setup, click **Save Changes** to leave the dialog.

7.14.4 Download Trend Data in CSV Format

Trend logs can be downloaded from the device via FTP in CSV format (see Section 18.1.2). The CSV contents are generated on-the-fly from the internal binary storage when accessing the file. Each trend log point has one CSV file. The files are located in

`/data/trend/`*TrendLogName_UID*`.csv`

Where *TrendLogName* is the data point name of the trend (Trend Name). The *UID* is the unique ID of the trend log object. The UID can be obtained from the ID column in the data point list of trend log data points as shown in Figure 229. This would result in the trend CSV file '`/data/trend/out_temp_107C.csv`'.

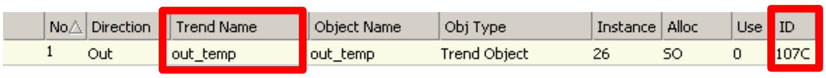


Figure 229: UID of data points.

Because the contents are generated on-the-fly, the file size in the FTP client will appear as 0 Bytes. The decimal point and CSV column separator can be configured in the system configuration of the Web UI (see Section 5.2.1). Note, that for a comma "," as the separator, the decimal point is a point. This is useful for English/U.S. applications. For countries that use the comma as the decimal point, select the semicolon as the CSV separator.

7.14.5 Deliver Trend Data via E-mail

Trend logs can be downloaded from the device via FTP. This requires an active action by the user. Alternatively, trend data can be sent as an e-mail attachment. For doing that, an e-mail template has to be setup for the trend log to be transmitted. The fill-level condition in the trend object can be used as a trigger to send an e-mail with the trend's data logger CSV file as an attachment.

For setting up e-mails, the account information has to be configured on the device, e.g., on the Web UI (see Section 5.2.24). Then an e-mail template can be created and the trend object attached as a trigger.

To Create an E-mail Template for Trends

1. Create or configure an e-mail template as described in Section 7.11.1.

2. Change to the **Mail Triggers** tab.

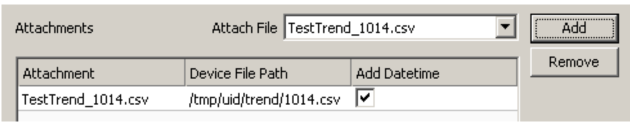
3. Click the **Add…** button and select a trend object.

4. In the **Mail Triggers** list, the added trigger data point appears with the **Fill Level** condition.

5. Change to the **Attachments** tab.

6. Select the trend log CSV file of the trend object in the **Attach File** drop-down box and click **Add**.

Note: *ZIP versions of the CSV files are also available. Select those to save transmission bandwidth and mailbox space.*



7. Click **OK** to complete the e-mail template configuration.

7.14.6 Technology Trends

In the BACnet technology, trend logs can be exposed on the BACnet port via special BACnet Trend Log objects. To create a technology trend select the port folder (e.g., **BACnet Port**) and then the **Trend** folder underneath. Follow the same instructions as described in Section 7.14.1. Please note, that certain restrictions apply to BACnet trends (see Section 6.7.4).

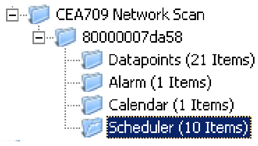
7.15 Remote AST Objects

7.15.1 Remote Scheduler and Calendar

Adding remote access to the configuration of a scheduler and calendar, which is located on another device, is done by creating remote scheduler and calendar objects. These objects can be created from data obtained by a network scan. With a CEA-709 device an LNS scan can also be used. Remote scheduler and calendars are supported by CEA-709 and BACnet technologies.

To Create a Remote Scheduler

1. Execute a network scan, as described earlier in this document. The scan folder is filled with available schedulers.



2. From the data points in the import folder, select the scheduler objects you are interested in and click the  **Use on Device** speed button. This creates suitable remote schedulers and the corresponding calendar objects in the **Remote Devices** folder.

3. Adjust the basic settings for the newly created objects, such as the object name and description. The object name will be used as the name for the scheduler, as seen on the Web UI.

Note: *Due to the static input NV, which is required for a remote CEA-709 scheduler object, adding remote scheduler points will change the static interface of the device.*

The new static input NV representing the remote calendar on the local device (this NV is normally called *nviCalLink*) needs to be bound to the output NV called *nvoCalLink* located in the Calendar functional block of the remote device and the new static *nviSchedLink* NVs which were created for each remote scheduler point need to be bound to the respective *nvoSchedLink* variable located in the Scheduler functional block of the remote device. The binding between the *nvoSchedLink* variable on the remote device to the *nviSchedLink* variable on the local device defines which of the scheduler data points on the local device connect to which scheduler unit on the remote device. All required information is transmitted over the link NVs, so it is possible to later change the binding to any other remote scheduler without rescanning the network.

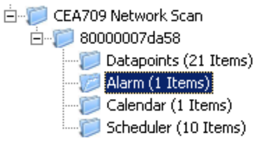
Note: *If connected via LNS, the bindings to the nvoCalLink and nvoSchedLink NVs are made automatically by the configuration software in the download process.*

7.15.2 Alarm Clients

Accessing alarm server objects on remote devices is done by creating remote alarm data points. These points may be created from data obtained by a network scan. The local device is configured as an alarm client and subscribes to alarm updates from the remote alarm server. The alarm client can also be used to acknowledge alarms on the remote alarm server. Any updates are synchronized back to the alarm client. Remote alarm servers are supported by CEA-709 and BACnet technologies.

To Create an Alarm Client

1. Execute a network scan, as described earlier in this document. The scan folder is filled with available remote alarm servers.



2. From the points in the import folder, select the alarm server points you are interested in and click the **Use on Device** speed button. This creates the corresponding alarm client points in your project.

3. In the CEA-709 technology select the new alarm client point and adjust the name of the local NV (default name is *nviAlarm_2*). This NV is located in the Clients functional block.

4. In the CEA 709 technology a static NV is created to receive information from the remote device about changes to the scheduler configuration, so that the local device does not need to poll the remote device. Set a name for this NV (default is nviSchedLink<number>) and assign it to a suitable function block.

Note: *Due to the static input NV which is required for a CEA-709 alarm client point, adding alarm clients will change the static interface of the device.*

The new static input NVs representing the alarm clients on the local device need to be bound to the alarm outputs of the remote device. A CEA-709 device normally delivers alarms through an output NV of type *SNVT_alarm_2* located in the node object of the device, therefore the new input NV on the local device must be bound to the alarm output NV of the remote devices node object. All required information is transmitted over the alarm input NV, so it is possible to later bind the alarm client to any other alarm server without rescanning the network.

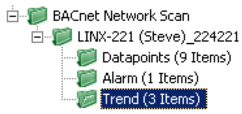
Note: *If connected via LNS, the binding to the nvoAlarm2 NV is made automatically by the configuration software in the download process.*

7.15.3 Remote Trend Logs

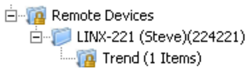
A remote trend log provides access to trend log data, which is actually generated and stored on another device. The remote trend log can load trend data from that device and supply it to L-WEB or a CSV file. Please note, that currently only the BACnet technology supports remote trend logs.

To Create a Remote Trend Log

1. Execute a network scan, as described earlier in this document. The scan folder is filled with available trend logs.



2. From the data points in the import folder, select the trend log objects you are interested in and click the **Use on Device** speed button. This creates suitable remote trend logs in the **Remote Devices** folder.



7.16 Math Objects

7.16.1 Create a Math Object

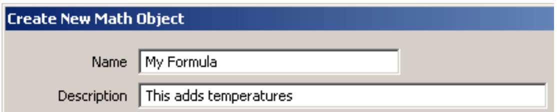
Math objects are advanced application objects that can execute mathematical operations on data points. A math object takes a number of input data points (variables v_1, v_2, …, v_n) and calculates a result value according to a specified formula. When configuring a math object, the input data points, output data points and the formula must be configured by the user. Input data points can be configured with a change-of-value condition, to trigger the math calculation only if the value changes more than a certain delta.

To Create a Math Object

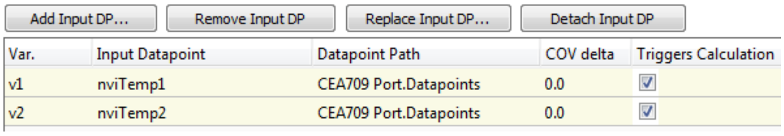
1. Under the **Global Objects** folder, select the **Math Object** sub-folder.

2. Right-click and select **New Math Object …** from the context menu.

3. In the Create New Math Object dialog, enter a name and optionally a description for the math object.



4. Attach input data points by clicking the **Add Input DP** button.

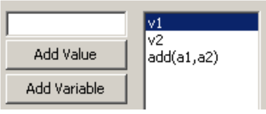


5. In the data point selector dialog, select the input data points and click **OK**. The data points appear as v1, v2, etc.

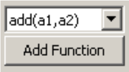
Tip! *A math object can also be created by multi-selecting data points in the data point manager and right-clicking on them. Then choose **Create Math Object …** from the context menu. This opens the dialog and attaches the selected data points as input variables.*

6. If the data point shall trigger the math calculation only after a certain change-of-value, enter a value into the **COV delta** column. If the data point shall never trigger the math calculation, de-select the **Triggers Calculation** check box.

7. Select the input data point and click **Add Variable** to push the variable on the evaluation stack.



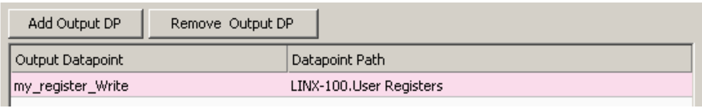
8. Select a function to be applied on the variables and click the **Add Function** button.



9. Te resulting formula is displayed at the bottom of the dialog. Alternatively, the formula can be entered there.

Note: *When the formula entered at the bottom is still incomplete and does not yield a meaningful command sequence, the list showing the RPN equivalent will be empty. This allows the user to immediately see if the current input is valid or not.*

10. Add output data points by clicking the **Add Output DP button**.



11. In the data point selector dialog select the output data points and click **OK**.

12. To create the math object click **Create**.

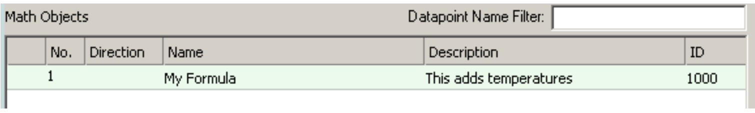
7.16.2 Editing a Math Object

Math objects can be edited once created. The formula can be changed, new variables added, or additional output data points added.

To Edit a Math Object

1. Under the **Global Objects** folder, select the **Math Object** sub-folder.

2. Select the math object in the data point list.



3. Right-click and select **Configure Math Object …** from the context menu.

4. Edit the math object as described in Section 7.16.1.

5. To replace an input data point by another input data point without re-writing the entire formula, click the **Replace Input DP …** button. This opens a data point selector dialog. Select the replacement data point there.

6. To detach an input data point, click the **Detach Input DP** button. This leaves the respective variable slot empty.

7. To finalize the edit click on **Save Changes**.

7.17 Historic Filters

7.17.1 Create Historic Filters

Historic filters are used to work with historic values of a base data point. These historic values are derived by defining historic filter functions. An historic filter template is a collection of such filter functions and can be assigned to any analog, binary, or multi-state base data point. For more general information on historic filters refer to Section 6.5.6.

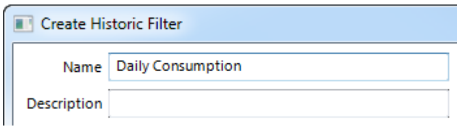
To Create Historic Filters

1. Select one or more data points that shall serve as the base data points.

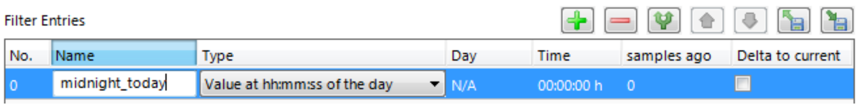
2. Right-click and select the **Configure Historic Filters …** item in the context menu. As an alternative, click on of the **Historic Filter** data point property.

3. The dialog **Select Historic Filter** opens. Click on to create a new one.

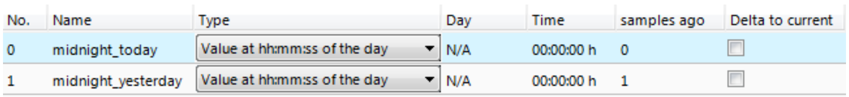
4. In the **Create Historic Filter** dialog enter a name and optionally a description.



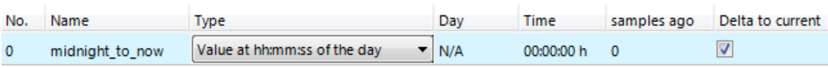
5. To add a new filter function to the list click on the add button . Enter a filter entry **Name** and choose the desired period **Type**. The name will appear with the historic filter relation and helps identifying it. Depending on the type define the arguments **Day** and **Time**. Select how many **samples ago** the filter goes back. The most current sample is ‘0’, the previous one is ‘1’.



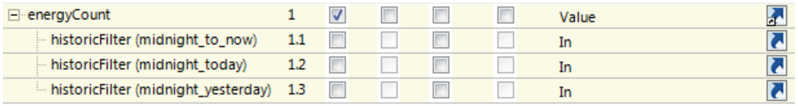
6. To duplicate an entry click on the duplicate button . On the duplicate modify the settings accordingly.



7. For getting the difference of an historic value to the current value check the box **Delta to current**.



8. Click on **Save Changes** and select the created filter template. For each filter entry defined, a historicFilter property relation is created under the base data point(s).



7.17.2 Managing Historic Filter Resources

Historic filters are stored as template types in the project resources. The Configurator provides a resource manager, that can be used to view, edit, import and export historic filter resources. When modifying an historic filter template, all instances that use it are updated.

Select the menu **Tools → Manage Historic Filters …** to open the resource manager dialog. Select a filter type and click the edit button for modifying it. Use the plus button to add new filters and the minus button to delete selected filters. Click the duplicate button to create a duplicate of the selected filter. Click the import button to load historic filters from disk. When importing, filter definitions of the same name are overwritten. Click the export button to store current filters to disk.

7.18 Automated Data Point Creation

7.18.1 Data Point Templates

The Configurator provides an import interface to automatically create a data point configuration from a list of data point templates. This interface can be used by an external tool, which generates such a list from its own project planning data.

Creating data points from data point template works like copy/paste of data points from an existing configuration to a new one. The “copied” data point is just stored in a data point template file (.dptmpl), which can be later “pasted”. For a given set of frequently used data points of a given technology, the user can build a library of such template files.

Data point templates include the following properties of their original data point:

- Generic and technology-specific properties,
- Sub-data points of structures,
- Property relations (without referenced objects),
- Favorites (without referenced objects),
- Alarm conditions (including required alarm servers).

To Create a Data Point Template

1. Select an existing data point in the data point list.
2. Right-click and select **Create Data Point Template …** from the context menu.
3. Enter a file name and store the ‘.dptmpl’ file.

To Apply a Data Point Template

1. Select one or more data points in the data point list.

2. Right-click and select **Apply Data Point Template …** from the context menu.

3. Choose a '.dptmpl' file. This will re-model the data points to the selected template.

7.18.2 Creation from Data Point Template CSV

In order to create a batch of data points from those template files, a list can be specified in a data point template CSV file (.dpcsv). For each line in this CSV file a data point will be created under the specified path and name according to a data point template file. Optionally, the description, PLC and OPC settings of the data point template can be overridden by the CSV file. The format of the CSV file is specified in Section 18.1.4. An example CSV 'dp_template.csv' file can be found in the Configurator install directory under '`examples\Data Point Templates`'.

To Create Data Points from a Template CSV File

1. Create a data point template CSV file. An example is shown in Figure 208. The referenced '.dptmpl' files must be located in the same directory as the CSV file. The path to the data point must be specified using the dot '.' as the folder separator.

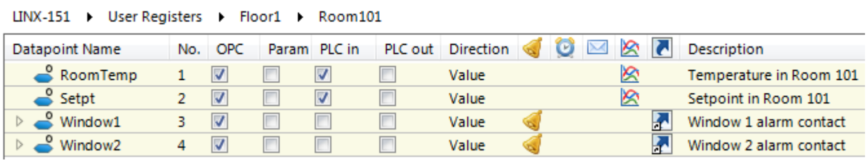
	A	B	C	D	E	F	G	H	I
1	#Path	Name	Description	PlcIn	PlcOut	Opc	Trend	Schedule	TemplateFile
2	Floor1.Room101	RoomTemp	Temperature in Room 101	1	0	1	1	0	Temp_Reg.dptmpl
3	Floor1.Room101	Setpt	Setpoint in Room 101	1	0	1	1	0	Temp_Reg.dptmpl
4	Floor1.Room101	Window1	Window 1 alarm contact	0	0	1	0	0	Alarm_Reg.dptmpl
5	Floor1.Room101	Window2	Window 2 alarm contact	0	0	1	0	0	Alarm_Reg.dptmpl

Figure 230: Example Data Point Template CSV File.

2. Select the menu **Tools → Import Data Point Template CSV …** and choose the created CSV file.

3. The temperature registers in the example are specified as trended. If the device model supports different trend technologies, the import will prompt to choose one.

4. The import creates two temperature analog registers and two binary registers with alarm conditions. The data point templates refer to an alarm server, which is also created, if none exists with the same name.



8 The CEA-709 Router

LOYTEC devices, which are equipped with a standard CEA-709 router (i.e., an embedded L-IP), connect the FT port and the CEA-852 port. Depending on the use case, the CEA-709 router supports different operating modes how packets are routed between the CEA-709 side and the IP-852 side. LOYTEC devices with the router option also contain a configuration server (CS) to manage members on an IP-852 channel.

8.1 CEA-709 Router

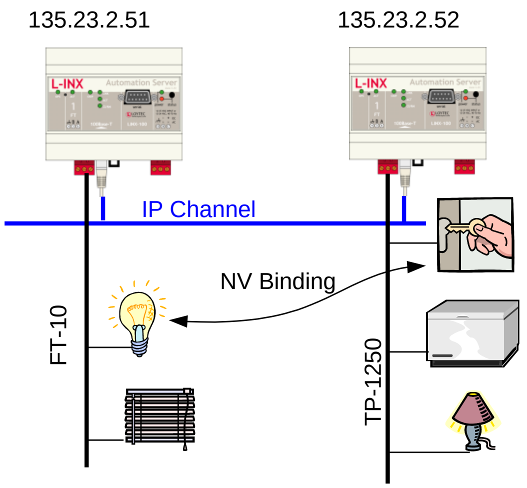


Figure 231: The L-INX supports different methods to route packets between the CEA-709 and IP-852 channel.

Depending on the CEA-709 router configuration (see Section 5.2.12) the CEA-709 router supports 4 different methods to route packets between the CEA-709 and the IP-852 channel. Those operating modes are listed below and described in more detail in the subsequent sections.

- Configured Router: The device acts like a standard CEA-709 configured router (*i*.LON 1000/600 alike)
- Smart Switch: The device acts as a self-learning plug&play router (“smart switch mode”)
- Store-and-Forward Repeater: To freeze a learned configuration and operate the switch based on the existing forwarding tables, disable group learning and Subnet/Node learning.
- Smart switch with no broadcast flooding: Set Subnet/Node Learning to “subnet”. In this mode the router learns the network topology but doesn’t flood subnet broadcasts.

8.1.1 Configured Router Mode

In this operating mode, the router acts like a standard configured router, which can be configured with standard network management tools like LonMaker or NL-220. This operating mode is compatible with the *i*.LON 1000 and the *i*.LON 600.

This operating mode uses the "channel routing" routing strategy on the IP channel. In this mode the device is fully compatible with *i*.LON 1000/600 devices. This operating mode should also be used in networks with more than 10 IP devices on one IP channel and heavy network traffic on the IP channel (more than 500 packets/s) since channel routing sends the IP packet only to the IP-852 device(s) that connect to the CEA-709 node(s) addressed in this IP packet and not to all IP-852 devices on the IP channel. This is the standard operating mode.

8.1.2 Smart Switch Mode

The router can be configured to act as a learning switch in a CEA-709 network. This operating mode is called smart switch mode. In this operating mode, the router decides if the message has to be forwarded or not, based on the destination address of a message. Thus, it isolates local network traffic (e.g., in case of heavily loaded networks).

Important: ***This operating mode doesn't support network loops!***

Important: ***Whenever a network is reconfigured, it is recommended to clear the forwarding tables in the device by pressing the status button for at least 20 seconds (see Section 4.5.1).***

The router supports learning of up to 4 Domains.

Note: *All messages, which are received on an unknown domain, are forwarded to all ports!*

The subnet/node learning algorithm supports segmentation of the network traffic on a subnet/node basis. Thus, the user does NOT need to take care of any subnets spanning multiple physical channels. Even when a node is moved from one channel to another, the router keeps track and modifies its forwarding tables accordingly.

Note: *All messages with a destination subnet/node address not yet learned are forwarded!*

The router supports group learning. Groups can span multiple router ports.

Note: *Group learning only works for messages using acknowledged or request/response service.*

Note: *All messages with a destination group address not yet learned are forwarded!*

The router has no learning strategy for broadcast addresses. As a result, all subnet or domain wide broadcasts are always forwarded. If subnet wide broadcasts shall not be forwarded, please use the smart switch operating mode without subnet broadcast forwarding (see Section 8.1.4).

The router has no learning strategy for unique node ID addresses. Node ID addressed messages are always forwarded.

This operating mode uses the "channel routing" strategy on the IP channel to distribute IP packets. It uses flooding to send all packets on the IP channel to all IP devices on this IP channel. The advantage of this operating mode is that it is fully plug&play and no router configuration is required. The disadvantage is that this operating mode doesn't scale very well with larger networks. We do not recommend this operating mode for IP channels with more than 10 IP-852 devices and packet rates of more than 500 packets/s. Please use the configured router mode from Section 8.1.1 for larger IP channel configurations.

Further, it is recommended to configure a multi-cast group for routers in the smart switch mode to reduce the traffic burden and improve scalability. Refer to Section 8.5 on how to configure the device to use multi-cast.

8.1.3 Store-and-Forward Repeater

The router can be configured to operate in a repeater mode, where all messages are forwarded regardless of the address format.

This operating mode uses the "channel routing" strategy on the IP channel to distribute IP packets. It uses flooding to send all packets on the IP channel to all IP devices on this IP channel. The advantage of this operating mode is that it is fully plug&play and no router configuration is required. The disadvantage is that this operating mode doesn't scale very well with larger networks. We do not recommend this operating mode for IP channels with more than 10 router devices and packet rates of more than 500 packets/s.

Further, it is recommended to configure a multi-cast group for routers in repeater mode to reduce the traffic burden and improve scalability. Refer to Section 8.5 on how to configure the device to use multi-cast.

8.1.4 Smart Switch Mode with No Subnet Broadcast Flooding

This operating mode is the same as the smart switch mode from Section 8.1.2 with the only difference that subnet wide broadcasts are not flooded in this mode. This operating mode can be used in large network installations where the network management tool uses group overloading to replace group addresses with subnet wide broadcasts. In this operating mode, the network installer must ensure that one subnet address may only exist behind one and no more than one network port. This condition is met if nodes are installed using an LNS based tool, on different channels that are separated with a router shape.

This operating mode uses the "channel routing" strategy on the IP channel to distribute IP packets. It uses flooding to send all packets on the IP channel to all IP devices on this IP channel. The advantage of this operating mode is that it is fully plug&play and no router configuration is required. The disadvantage is that this operating mode doesn't scale very well with larger networks. We do not recommend this operating mode for IP channels with more than 10 devices and packet rates of more than 500 packets/s.

Further, it is recommended to configure a multi-cast group for the router in the smart switch mode to reduce the traffic burden and improve scalability. Refer to Section 8.5 on how to configure the device to use multi-cast.

8.2 CEA-852 Device of the Router

Every L-INX acts as a device on the IP channel. It either needs to contact a configuration server or a configuration server needs to contact the device in order to set up the proper routing tables. Before a device can become a member of the IP-852 channel it needs to have proper IP settings (see Section 5.2.4):

- IP address/netmask/gateway (either via DHCP or manual entry), see Section 5.2.4
- Auto-NAT or manual NAT address if used behind a firewall/NAT router, see Section 5.2.10
- MD5 secret if authentication is required, see Section 5.2.10

Please consult Sections 5.2.4 and 5.2.10 on how to setup a CEA-852 device.

The CEA-852 device can be used together with the PC-based *i*.LON Configuration Server utility or with any LOYTEC configuration server. If multiple CEA-852 devices behind one NAT router are added, the Auto-NAT setting in the CEA-852 devices is recommended to be used with the L-INX configuration server or an L-IP configuration server. Please refer to the following sections on how to setup the device and the configuration server.

If the “Auto member” feature is enabled in the configuration server, the CEA-852 device can add itself to the IP-852 channel without explicitly adding the device at the configuration server. Note, that enabling auto member is a potential security hole since any device can add itself to the IP-852 channel.

8.3 Configuration Server for Managing the IP-852 Channel

8.3.1 Overview

Every logical IP-852 channel requires one configuration server that manages all CEA-852 devices (LINX-121, L-IP, LOYTEC NIC852, *i*.LON 1000, *i*.LON 600, LonMaker, etc.) on this channel. A simple example is shown in Figure 232. A configuration server keeps a list of all devices on a logical IP-852 channel and distributes the routing information between those devices. If a device wants to join an IP-852 channel, it needs to register itself at the configuration server. Traditionally, a dedicated Windows PC is used to act as the configuration server. The L-INX contains an embedded configuration server and can therefore replace the PC.

The configuration server can be enabled in the CEA-852 server configuration menu in Section 5.2.13. This configuration server can manage one IP-852 channel and up to 256 devices on this IP-852 channel. In order to setup the configuration server, one must specify the following parameters:

- IP address/netmask/gateway (either via DHCP or manual entry), see Section 5.2.4
- NAT address if used behind a firewall/NAT router, see Section 5.2.10
- MD5 secret if authentication is required, see Section 5.2.10
- Enable the configuration server, see Section 5.2.13 (server LED lights up green)
- A list of IP-852 channel members, see Section 5.2.14.

Note: *If the L-INX is also used as a configuration server it needs a fixed IP address.*

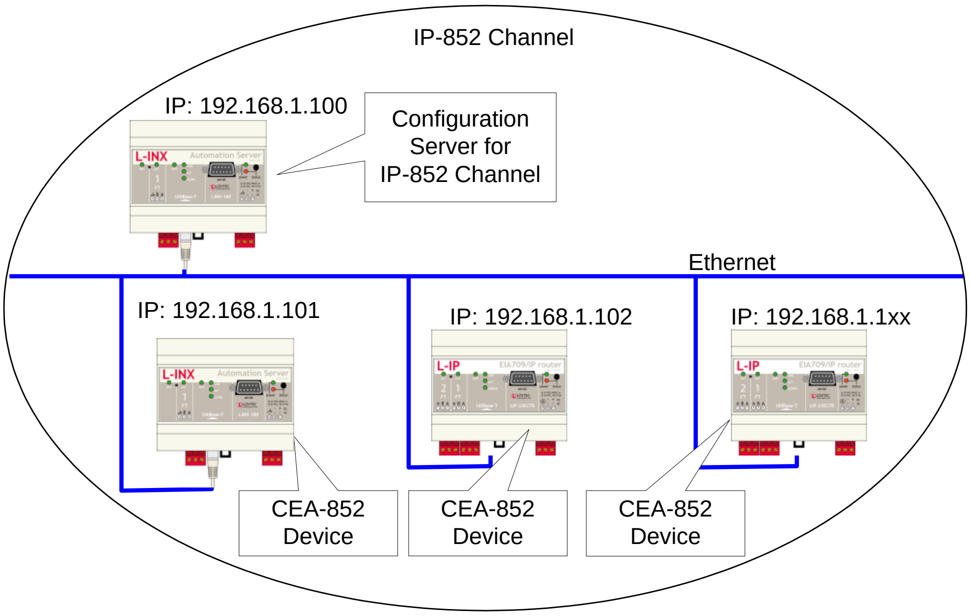


Figure 232: The configuration server manages the devices on an IP-852 channel.

8.3.2 Configuration Server Contacts IP-852 Device

In this scenario, the IP-852 device needs the following parameters set in order for the configuration server to contact the device. The remaining parameters are retrieved from the configuration server.

- IP address/netmask/gateway (either via DHCP or manual entry), see Section 5.2.4
- Auto-NAT or manual NAT address if used behind a firewall/NAT router, see Section 5.2.10
- MD5 secret if authentication is required, see Section 5.2.10

If multiple CEA-852 devices behind one NAT router are added, the Auto-NAT setting in the L-INX is recommended to be used with the L-INX configuration server.

8.3.3 IP-852 Device Contacts Configuration Server

In this scenario, the IP-852 device needs the following parameters set in order to contact the configuration server. The remaining parameters are retrieved from the configuration server.

- IP address/netmask/gateway (either via DHCP or manual entry), see Section 5.2.4
- Auto-NAT or manual NAT address if used behind a firewall/NAT router, see Section 5.2.10
- MD5 secret if authentication is required, see Section 5.2.10
- Configuration server IP address and port number, see Section 5.2.10

If the “Auto member” feature is enabled in the configuration server, the CEA-852 device can add itself to the IP-852 channel without explicitly adding the device at the configuration server. Note, that enabling auto member is a potential security hole since any device can add itself to the IP-852 channel.

8.3.4 Using the Built-In Configuration Server

For security purposes, the configuration server contacts each CEA-852 device on the IP-852 channel. Therefore, one must enter a list of all channel members in the CEA-852

Configuration Server menu (see Section 5.2.14). This ensures that no unwanted device can join the IP-852 channel.

Note that also *i*.LON 1000/600, VNI and LOYTEC NIC852 based network nodes (e.g., LonMaker or NL-220 applications) can join the IP-852 channel managed by the configuration server. Note that the built-in configuration server should be used if LOYTEC CEA-852 devices are communicating across firewalls/NAT routers.

For adding multiple devices behind a NAT router, the configuration server supports the extended NAT mode (see Section 8.4.2). The configuration server automatically switches the channel mode to extended NAT if needed. Note that the *i*.LON 600 must be configured with the *i*.LON CS to extended NAT mode before adding the *i*.LON 600 to the configuration server, because the *i*.LON 600 does not switch to that mode automatically.

8.4 Firewall and NAT Router Configuration

The CEA-709 router can be used behind a firewall and/or NAT (Network Address Translation) router as shown in Figure 233. Note, that in general, only one CEA-852 device can be used behind the NAT router. This mode of operation is referred to as “Standard” channel mode. It is fully compliant with CEA-852.

LOYTEC’s newer devices such as the L-IP and the L-INX family support more than one CEA-852 channel member behind a NAT router. This mode of operation is referred to as “Extended NAT” channel mode. This mode introduces extensions to the standard mode which need to be supported by all members. Other devices supporting the extended NAT mode are the *i*.LON 600. See Section 8.3.4 on compatibility with the *i*.LON 600.

8.4.1 Automatic NAT Configuration

In order to use the L-INX behind a firewall, the public NAT address and the local IP address must be set in the IP configuration menu (see Section 5.2.4). By default, the NAT address is determined automatically when adding the L-INX to the channel in the configuration server. Alternatively, the NAT address can be configured manually. Furthermore, the NAT router must be configured to forward ports 1628 and 1629 for UDP and TCP packets to the private IP address of the L-INX (192.168.1.100 in Figure 233). In summary we can say, the following parameters must be set in order to operate a L-INX behind a NAT router.

- Specify the IP address (private IP address: 192.168.1.100),
- Specify the gateway address (e.g., 192.168.1.1),
- Specify the NAT address (public IP address: 135.23.2.1) or use automatic NAT router discovery,
- Enable port forwarding for ports 1628 and 1629 in the NAT router for TCP and UDP,
- Enable the SNTP port 123 in the firewall if SNTP is used.

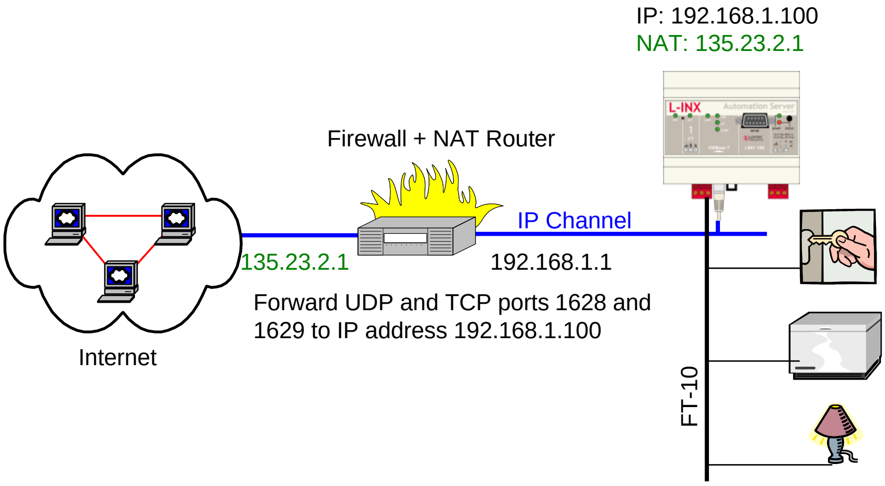


Figure 233: Operating a L-INX behind a NAT router and firewall.

Note that a L-INX must be used as configuration server when the device is installed behind a firewall or NAT router. The L-INX with the configuration server can also be located behind a firewall.

8.4.2 Multiple IP-852 Devices behind a NAT: Extended NAT Mode

When using more than one IP-852 device behind a single NAT router, the recommended method in the L-INX configuration server is to use the extended NAT mode. This mode requires that all devices support this feature. Currently these are L-INX with CEA-709 router, L-IP 3.0, *i*.LON 600, the NIC852 PC software and other CEA-852 capable devices from LOYTEC. If there are other devices in the channel, this method does not work. Incompatible devices are disabled from the channel in this case. Please refer to the classic method in Section 8.4.3 to setup this network.

When using multiple devices behind a NAT router, each device needs a separate port-forwarding rule in the NAT router. This implies that each device must use a unique client port (e.g., 1628, 1630, 1631, etc). The port-forwarding rules must be setup so that each port points to one of the IP-852 devices. In the L-INX, change the client port in the CEA-852 device configuration menu. Figure 234 shows an example configuration for three L-INX devices behind the NAT router 135.23.2.1.

It is recommended that both ports 1628 and 1629 are forwarded to the same private address. It is then also possible to turn on the configuration server behind a NAT router. In this case, activate the CS on the L-INX which has port-forwarding to 1628 and 1629. In the example in Figure 234, the L-INX with private address 192.168.1.100 also acts as a configuration server.

If the CS is activated on a L-INX behind a NAT router, the NAT router must have a fixed public IP address. The L-INX with the CS also cannot use automatic NAT discovery. In this case, enter the NAT address of the NAT router manually in the IP configuration menu (Auto-NAT can no longer be enabled on a L-INX with a CS). To diagnose possible problems in the NAT configuration with port forwarding, use the enhanced communications test (see Section 5.5.5).

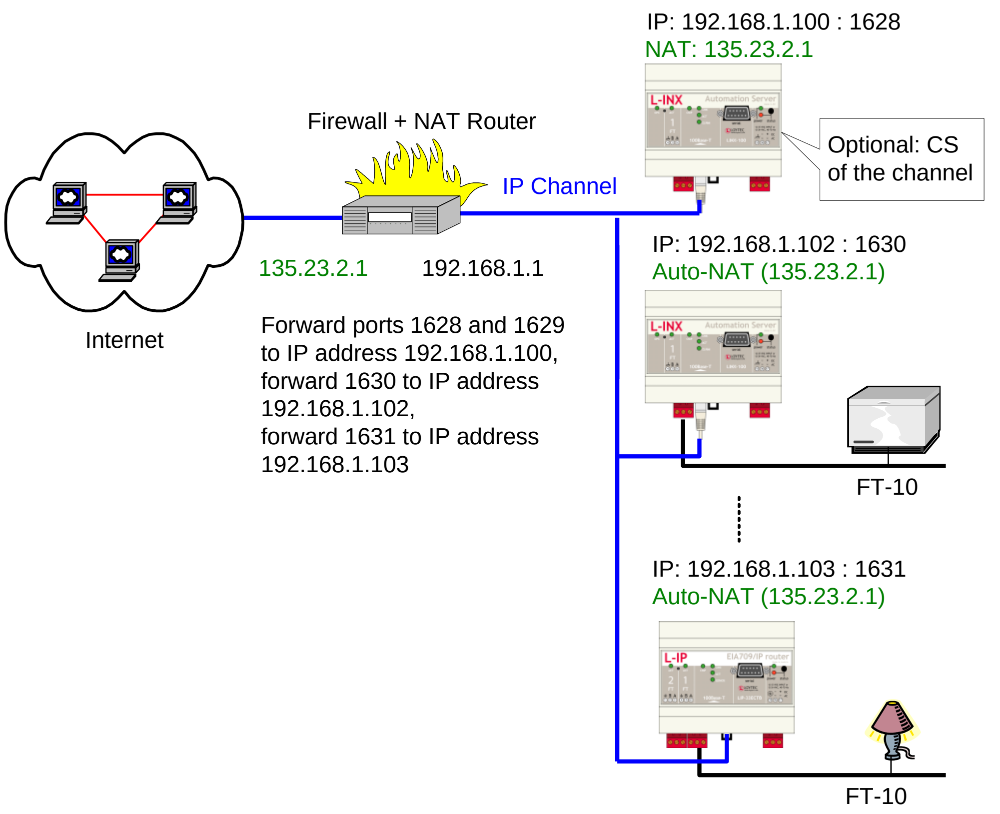


Figure 234: Multiple L-INX devices behind a NAT: Extended NAT Mode.

After the NAT router has been configured with the port-forwarding settings and the CS has been turned on, the channel members can be added. This can be done either through the Web interface of the CS.

In the Web UI, add the members with their private IP addresses and the client ports as defined by the port-forwarding. Then select the added member by checking the check box and select the action **Assign to NAT**. Enter the public NAT address of the NAT router. An example to add the two IP-852 devices in Figure 234 through the Web UI is depicted in Figure 235. To remove a device from a NAT router but not delete it, select it and choose **Remove from NAT** as the action.

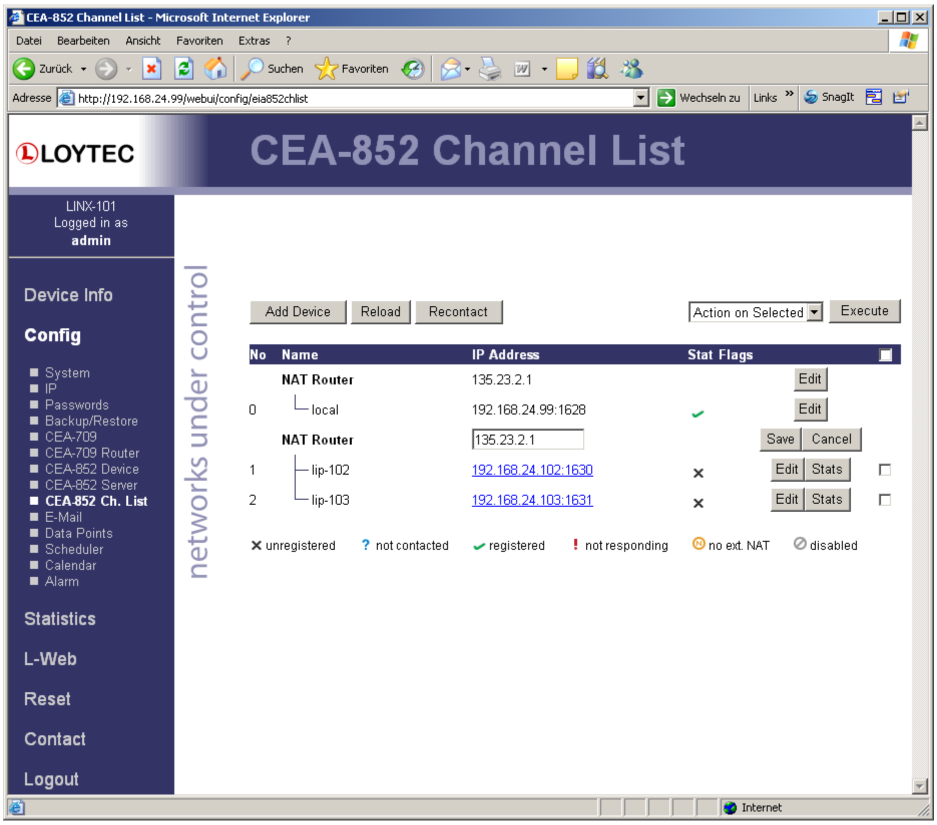


Figure 235: Adding a member with extended NAT Mode on the Web UI.

8.4.3 Multiple IP-852 devices behind a NAT: Classic Method

If more than one CEA-852 device must be used behind the NAT router and there are devices which do not support the extended NAT mode, we propose the setup from Figure 236.

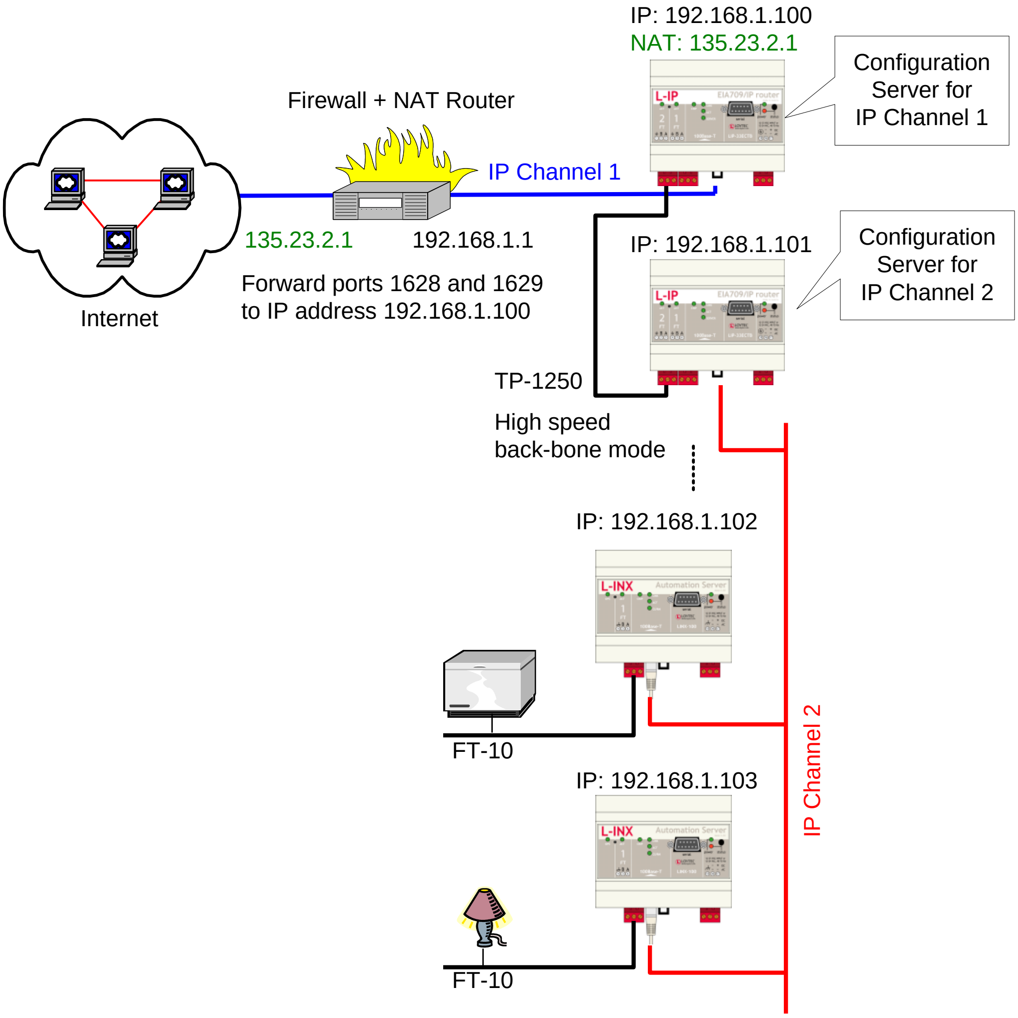


Figure 236: Application that uses multiple L-INX devices behind a NAT router firewall.

The L-INX with IP address 192.168.1.100 is member of IP Channel 1 and can be accessed through the Internet. The L-INX devices with IP addresses 192.168.101 to 192.168.1.110 form another logical IP Channel 2 that communicates with the devices on the IP Channel 1 over the TP-1250 channel, which is used in high-speed backbone mode for optimum networking performance. Note that devices on both IP Channels 1 and 2 can of course connect to the same physical network wiring. Furthermore, both IP Channels 1 and 2 must have a separate configuration server that manages the L-INX devices on the different channels. In the example in Figure 236, the L-INX with address 192.168.1.100 acts as the configuration server for IP Channel 1 and the L-INX with IP address 192.168.1.101 acts as the configuration server for IP Channel 2.

8.5 Multi-Cast Configuration

IP multi-casting is a feature of the IP protocol that allows one packet to be delivered to a group of IP hosts. To receive such multi-cast packets, each IP host must be member of a multi-cast group. This group is identified by a multi-cast address (e.g., 225.0.0.37) and a UDP port number.

The L-INX supports both unicast and multi-cast delivery of CEA-852 data packets. Using multi-cast is recommended when using the router in the Smart Switch Mode. For those devices, configure a multi-cast address in the IP configuration menu. Please contact your system administrator to obtain a valid multi-cast address for your network. Note, that all channel members must be configured with the same multi-cast address and use the same client port (1628 is recommended). Also note, that multi-cast addresses cannot be routed on the Internet. They can only be used in a LAN or VPN environment.

If you configure multi-cast, there may be some devices, which do not support this feature. In this case, the device uses a hybrid scheme and sends unicast to those devices, which are not configured for multi-cast. Note, that the device determines automatically, when to switch to the multi-cast mode depending on what types of devices are in the channel and on the traffic burden for those devices. As a rule of thumb, multi-cast is used when there are only switches/repeaters in the channel and it is not used when there are only configured routers.

To detect if the device utilizes the multi-cast feature, contact the Extended CEA-852 device statistics in the statistics menu (Section 5.5.4). The entry “Channel Routing Mode” reads SL (send list) if packets are routed to the multi-cast group. It reads CR (channel routing) if the normal unicast method is employed. Also the entry “Multi-cast packets sent” in the CEA-852 device statistics menu (Section 5.5.4) counts the number of multicast packets transmitted to the group. If this item remains zero, no multi-cast is used by the device.

8.6 Remote LPA Operation

The L-INX supports remote LPA access. This means that a CEA-709 protocol analyzer connected to the Ethernet network can connect to the L-INX and record all packets on the CEA-709 channel (FT-10). Our LPA-IP supports this sophisticated feature. The functionality is shown in Figure 237.

The LPA-IP runs on a Windows PC that is connected to the Ethernet network. In a device selection window, one can e.g. select the L-INX with IP address 192.168.1.210 and display all packets on the FT-10 channel connected to the L-INX with IP address 192.168.1.210. For this operation, the LPA-IP does not need to be a member of the IP-852 channel. Note that this functionality is only available with LOYTEC CEA-852 devices.

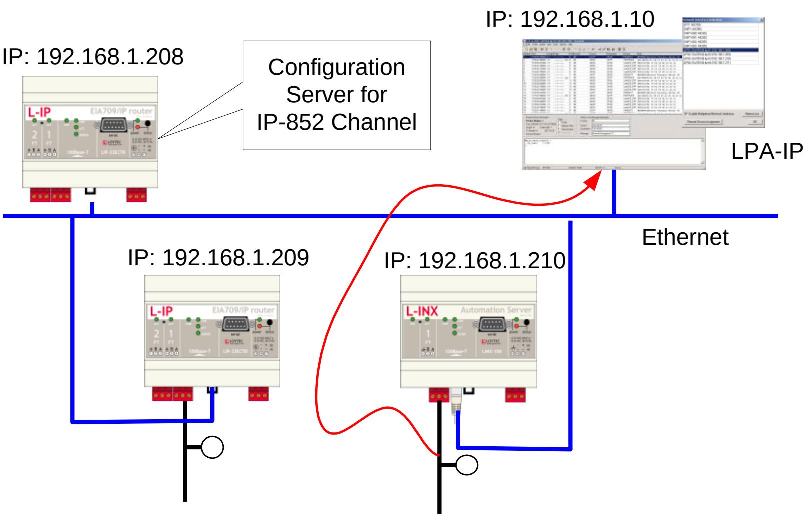


Figure 237: Remote LPA principle.

8.7 Internet Timing Aspects

If the CEA-709 router is used over the Internet or in a large Intranet with unpredictable network delays, the user should become familiar with the following advanced timing aspects. Channel Timeout is set in the configuration server whereas escrowing and aggregation are set in the CEA-852 client device. The Channel Delay is a channel property of LNS and can be set in NL220, LonMaker or other network management tools.

Table 22 summarizes the timing values that must be set when operating the device under WAN conditions.

Timing Parameter	Value
Channel Timeout	Average ping delay + Aggregation Timeout
Escrowing (Packet Reorder Timer)	The smaller value of: 0.25*Channel Timeout or 64ms
Aggregation Timeout (Packet Bunching)	Typically 16 ms
Channel Delay in LonMaker	Average ping delay +10% + 2* Aggregation Timeout

Table 22: Advanced IP-852 timing parameters.

Please use a PC to determine the average ping delay between the different CEA-852 devices in the network. If multiple devices are communicating with each other always use the largest measured average ping delay for the input value for the calculations in Table 22.

Escrowing should be disabled in a LAN (0 ms). The Channel Delay in LonMaker should be set to 2*Aggregation Timeout in a LAN if MD5 is disabled.

In LANs, Channel Timeout is only required if MD5 authentication is enabled. Set Channel Timeout to 200 ms and Channel Delay to 20 ms.

8.7.1 Channel Timeout

The Channel Timeout is a property of the IP-852 channel. If a packet travels across this IP-852 channel for longer than what is specified in Channel Timeout in ms, the packet is discarded. The device always needs to synchronize with an SNTP timeserver when a Channel Timeout is set other than 0 ms.

Channel Timeout is highly recommended if MD5 authentication is enabled in order to prevent replay attacks. Set Channel Timeout to 200 ms and Channel Delay to 20 ms in a LAN environment. Please refer to Section 5.2.13 on how to enable or disable the Channel Timeout.

If an LNS based network management tool like LonMaker or NL220 is used on a network that has channel timeout enabled, please install an NTP client program (e.g., achron4.exe) on this PC that synchronizes the PC clock to the NTP time. Otherwise the PC clock and the clock inside the CEA-852 device will drift apart and communication between the PC and the device will terminate.

8.7.2 Channel Delay

Channel Delay is an LNS channel property that specifies the expected round-trip time of a message and its response. This value is used by LNS to adjust the protocol timers in the CEA-709 nodes. Please consult the documentation for your network management tool about the Channel Delay details.

8.7.3 Escrowing Timer (Packet Reorder Timer)

The Escrowing Timer or Packet Reorder Timer is an IP-852 channel property that specifies the amount of time the device will wait for an out-of-sequence IP packet to arrive. This parameter is important in WANs like the Internet, where packets pass many routers that can change the order in which packets arrive at the destination node. The default value is 64 ms.

Do not use the Escrowing Timer in LANs since the packet order is always guaranteed in a LAN. This will add unnecessary delays, which negatively impacts the performance of your CEA-852 devices if a packet is lost or destroyed.

Whether enabled or disabled, out-of-sequence packets are never sent to the CEA-709 channel. Please refer to Section 5.2.10 on how to enable or disable escrowing.

8.7.4 SNTP Time Server

Small IP networks like LANs have a small propagation delay for packets traveling in these networks. In this case it is not necessary to specify an SNTP server.

In larger IP-852 networks like the Internet with possibly long packet delays, one must specify an SNTP server to synchronize the local clocks of the CEA-852 devices. The local clocks must be synchronized to a common notion of time in order to make CEA-852 protocol features like Escrowing and Channel Timeout work properly.

The SNTP timeserver can be specified on the IP-852 channel level in the configuration server, which distributes the timeserver address to all CEA-852 devices on the IP-852 channel. A primary and a secondary SNTP server can be defined please refer to Section 5.2.10 and Section 5.2.13 on how to enable the SNTP server.

8.8 Advanced Topics

8.8.1 Aggregation

Aggregation (or packet bunching) is a technique that collects multiple CEA-709 packets into a single larger IP packet. Aggregation improves overall system performance since one IP-852 packets, now carries multiple CEA-709 packets und with the same number of IP-852 transactions, more CEA-709 packets can be exchanged between CEA-852 devices thus reducing protocol overhead. The Aggregation Timeout defines the time period in ms in which the transmitting device collects the CEA-709 packets before it transmits the CEA-852 packet over the IP-852 channel. Please refer to Section 5.2.10 on how to enable aggregation. Note, that aggregation adds a delay to the transactions but dramatically improves the throughput of your IP-852 channel. Use aggregation if you have a high channel load but can tolerate some additional propagation delay given by the aggregation time value.

8.8.2 MD5 Authentication

MD5 authentication is a method of verifying the authenticity of the sending device. Only devices that have MD5 enabled and use the same MD5 secret can share information with each other. If the configuration server has MD5 enabled, only devices that have MD5 enabled and use the same MD5 secret as the configuration server can join the logical IP-852 channel. Please refer to Section 5.2.10 and 5.2.13 for details.

8.8.3 Dynamic NAT Addresses

A common practice for Internet providers is to assign addresses on a per-session basis to a client. Each time a connection is established (e.g., an ADSL link is set up), the Internet provider may choose an IP address from a pool. Since this address will be the public address of a NAT router, the NAT address configured in the device would need to be updated. The Auto-NAT feature in the device permanently monitors the current NAT address. When the device detects a change in the NAT address it re-registers with the configuration server using this new address. This feature requires a LOYTEC configuration server (e.g., L-INX, L-IP) and “Roaming Members” enabled on that CS.

A consequence of this monitoring process is that the device contacts the CS every 45 seconds to probe for the NAT address. This causes a small amount of additional traffic on the Internet link. The Auto-NAT feature also causes any shut-down connection to be re-established. The NAT monitoring functions as a keep-alive for the connection. If neither the additional traffic nor the automatic initiation of a new connection is tolerable, the Auto-NAT feature must be disabled and the NAT address configured manually. In this case, the Internet service provider needs to assign a fixed public IP address to the NAT router.

9 Remote Network Interface

9.1 RNI Function

The LGATE-950 and the CEA-709 L-INX devices without the router provide a remote network interface (RNI) function, if the device is configured to use the FT interface (FT mode). In this mode the device provides a remote network interface, which appears like a LOYTEC NIC-IP it is intended to be used together with the LOYTEC NIC software. The RNI can be utilized for remote access and configuration as well as trouble-shooting with the remote LPA.

In particular, the RNI appears as a regular LOYTEC network interface on the PC. The LOYTEC NIC software needs to be installed to utilize this interface also in LNS-based applications such as NL220 or LonMaker. Using this software, the L-INX can act as a direct interface to its local FT channel to be managed by LNS or similar tools. For more information on how to configure the LOYTEC NIC software on the PC, please refer to the LOYTEC NIC User Manual [3].

9.2 Remote LPA Operation

The LGATE-950 and the L-INX supports remote LPA access through its RNI. This means that a CEA-709 protocol analyzer connected to the Ethernet network can connect to the L-INX and record all packets on the CEA-709 channel (FT-10). The LOYTEC LPA-IP supports this advanced feature. The functionality is shown in Figure 238.

The LPA-IP runs on a Windows PC that is connected to the Ethernet network. In the NIC-IP/RNI device selection window, one can for example select the device with IP address 192.168.1.210 and display all packets on the FT-10 channel connected to the device with IP address 192.168.1.210. Please consult our product literature for the LPA-IP to learn more about this IP-based CEA-709 protocol analyzer.

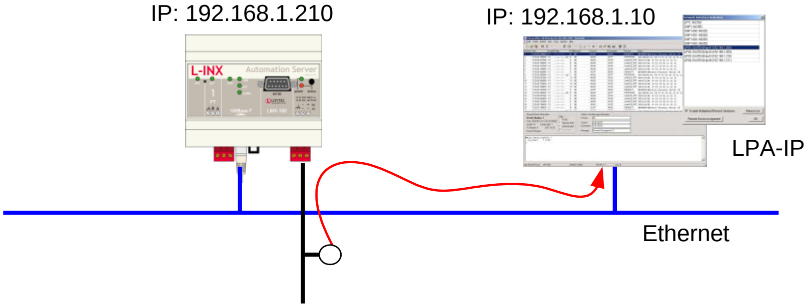


Figure 238: Remote LPA on the L-INX.

10 OPC Server

10.1 XML-DA OPC Server

10.1.1 Access Methods

LOYTEC devices with the built-in OPC server can expose data points over a Web service. The OPC tag namespace is built from the data point hierarchy, which has been configured by the Configurator software. The OPC server on the device implements the data access standard via the Web service interface XML-DA. The OPC XML-DA Web service is accessible via the URI

http://192.168.24.100/DA

where the IP address has to be replaced with the actual IP address of the device. The Web service is accessible over the same TCP port as the Web server. The default TCP port is 80. The Web server port can only be changed via the device configuration tab in the Configurator (see Section 7.3.6) or in the L-Config tool (see NIC User Manual [3]).

Since the Web service is easily routable on the Internet, the embedded OPC server implements the basic authentication method to protect the system from unauthorized access. The basic authentication involves the operator user and the password configured for this user. On how to configure the operator's password, please refer to Section 5.1.

To disable the basic authentication, clear the operator's password. On the LINX-12x, 15x, 22x models, the anonymous OPC access must be enabled. For doing so, change to the port configuration on the Web UI, select the OPC protocol on the Ethernet tab and check the anonymous OPC box.

Note: *It is highly recommended to use basic authentication when exposing crucial data points over the Web service.*

To use the exposed OPC data points, there exist several possibilities:

- Use LOYTEC's L-WEB visualization tool that comes free with the device,
- use LOYTEC's L-VIS device as OPC XML/DA client, or
- use a standard OPC client or SCADA package, or
- create your own Web service client with custom Web Pages.

The easiest way to visualize the network's data points over a Web-based interface using the device is the L-WEB software. This software is fully integrated into the Configurator and allows designing graphical page content. The tool is intuitive to use like the L-VIS graphical page designer. The resulting L-WEB application is stored on the device and can be directly accessed in your Web browser or other Internet appliances, such as tablets or smart phones. For more information on the L-WEB refer to Section 10.3.

Standard OPC clients and SCADA packages, which shall visualize the device's data points, must conform to the OPC XML-DA standard. This means they must support the OPC Web service and not only the COM/DCOM protocol. If your SCADA package does not support OPC XML-DA, a PC-based bridge from XML-DA to the COM-based protocol can be used. The bridge software is running on a PC and translates from COM/DCOM requests into XML-DA Web service requests. The system is depicted in Figure 239.

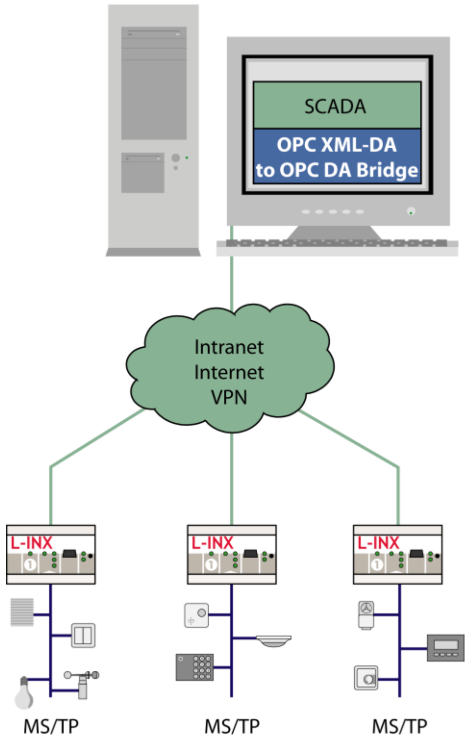


Figure 239: Using a XML-DA/DCOM bridge.

With the bridge is configured to access a number of OPC devices, the COM-based SCADA application can access a COM-based OPC server for each of those devices. The bridge software needs to be purchased from an OPC bridge software vendor.

If L-WEB is not used, customers can create their own XML-DA clients based on the WSDL for OPC XML-DA. Refer to Section 10.4 for more information.

10.1.2 Data Points

The data point hierarchy as configured by the Configurator software is exposed to the OPC tag namespace by the device. This is done internally for all data points, which are marked for OPC exposure (i.e., have the OPC check-mark set).

Folders are translated into OPC nodes. Any of the data point classes, analog, binary, multi-state, string, and user, are exposed as OPC tags. Each OPC tag contains the value of the data point and some of its meta-data. An example of browsing the OPC tags on the device is shown in Figure 240.

The OPC quality property of a given OPC tag is coupled to the data point status. If a data point is offline or unreliable, the OPC quality property changes to *uncertain*.

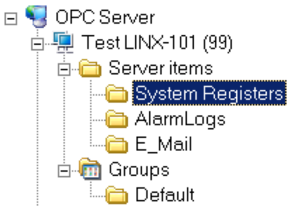


Figure 240: Client browsing the OPC tag namespace on a L-INX.

10.1.2.1 Analog

Analog data points are exposed as a one-to-one mapping to OPC tags. For each analog data point, an OPC tag is created. The OPC tag contains a number of OPC properties, which are derived from the data point’s properties:

- Item Canonical Data Type (SmallInt): This property indicates the data type ‘5’ (Double).
- Item Value (Double): The present data point value.
- Item Quality (SmallInt): The value quality. It is “good” if the data point is in normal state, or “uncertain” if the data point has an off-normal state, e.g., offline or unreliable.
- Item Timestamp (Date): This property contains the timestamp of the last value update.
- Item Access Rights (Integer): This property defines whether the tag is read-only or read/write.
- Item Description (String): This is the description of the data point.
- Item EU Type (Integer): This property is ‘1’.
- High EU (Double): This is the analog maximum value of the data point.
- Low EU (Double): This is the analog minimum value of the data point.
- EU Units (String): This is the human-readable engineering units text of the data point.

10.1.2.2 Binary

Binary data points are exposed as a one-to-one mapping to OPC tags. For each binary data point, an OPC tag is created. The OPC tag contains a number of OPC properties, which are derived from the data point’s properties:

- Item Canonical Data Type (SmallInt): This property indicates the data type ‘11’ (Boolean).
- Item Value (Boolean): The present data point value.
- Item Quality (SmallInt): The value quality. It is “good” if the data point is in normal state, or “uncertain” if the data point has an off-normal state, e.g., offline or unreliable.
- Item Timestamp (Date): This property contains the timestamp of the last value update.
- Item Access Rights (Integer): This property defines whether the tag is read-only or read/write.
- Item Description (String): This is the description of the data point.

- Contact Close Label (String): This property contains the active text of the binary data point.

- Contact Open Label (String): This property contains the inactive text of the binary data point.

10.1.2.3 Multi-state

Multi-state data points are exposed as a one-to-one mapping to OPC tags. For each multi-state data point an OPC tag is created. The OPC tag contains a number of OPC properties, which are derived from the data point's properties:

- Item Canonical Data Type (SmallInt): This property indicates the data type '3' (Integer).

- Item Value (Integer): The present data point value.

- Item Quality (SmallInt): The value quality. It is "good" if the data point is in normal state, or "uncertain" if the data point has an off-normal state, e.g., offline or unreliable.

- Item Timestamp (Date): This property contains the timestamp of the last value update.

- Item Access Rights (Integer): This property defines whether the tag is read-only or read/write.

- Item Description (String): This is the description of the data point.

- Item EU Type (Integer): This property is '2' for multi-state.

- Enumerated EU (Array of String): This property contains the state texts of the data point.

10.1.2.4 User Type

User-type data points contain a byte array of user-defined data. Data points of user-type are also exposed as a one-to-one mapping to OPC tags. For each such data point, an OPC tag is created. The item value of the user-defined data is a hex string without whitespace representing the byte array, e.g., "B034". The OPC tag contains a number of OPC properties, which are derived from the data point's properties:

- Item Canonical Data Type (SmallInt): This property indicates the data type '8' (String).

- Item Value (String): A hex string without whitespace representing the byte array.

- Item Quality (SmallInt): The value quality. It is "good" if the data point is in normal state, or "uncertain" if the data point has an off-normal state, e.g., offline or unreliable.

- Item Timestamp (Date): This property contains the timestamp of the last value update.

- Item Access Rights (Integer): This property defines whether the tag is read-only or read/write.

- Item Description (String): This is the description of the data point.

10.1.2.5 String

String data points contain a string of text characters. Data points of string type are also exposed as a one-to-one mapping to OPC tags. For each such data point, an OPC tag is created. The item value of the tag is the string data, e.g., “Room4”. The OPC tag contains a number of OPC properties, which are derived from the data point’s properties:

- Item Canonical Data Type (SmallInt): This property indicates the data type ‘8’ (String).
- Item Value (String): The string value.
- Item Quality (SmallInt): The value quality. It is “good” if the data point is in normal state, or “uncertain” if the data point has an off-normal state, e.g., offline or unreliable.
- Item Timestamp (Date): This property contains the timestamp of the last value update.
- Item Access Rights (Integer): This property defines whether the tag is read-only or read/write.
- Item Description (String): This is the description of the data point.

10.1.2.6 Structured Data Points

Structured data points are modeled as one user-type data point, which contains the entire structure value as a byte array. The respective structure fields are created as sub-data points of appropriate class. For example, a SNVT_switch in CEA-709 would be modeled as one user-type data point of 2 bytes length, and two sub-data points, one an analog (value member) and one a multi-state (state member).

The relation between user-type data point and sub-data points is also exposed to OPC. In this case, an OPC node is created for the user-type data point. In that node, the sub-data points are exposed as OPC tags. The entire structure is also exposed as a user-type OPC tag under the same OPC node.

Important! ***Deselect any un-used structure members from OPC exposure to reduce the number of total OPC tags.***

It is important to note, that when using structured data points the top-level and all its structure members are exposed as OPC tags by default. Using many structured data points may lead to exceeding the OPC tag limit. Please observe this limit in the Configurator’s statistics tab and deselect the **OPC Tag** check box for unwanted structure members. This helps to keep your configuration lean and improves the performance of the OPC server when browsing and subscribing.

10.1.3 AST Objects

The alarming, scheduling, and trending (AST) objects are more complex than regular data points. The OPC XML-DA standard does not have appropriate tags for those objects. Therefore, the device exposes AST objects as a set of OPC tags describing the object. All tags for one AST object are collected under an OPC node representing the AST object.

10.1.3.1 Scheduler Object

The device exposes the scheduler objects to OPC XML-DA tags. Each scheduler object is represented by a node in the OPC name space. The content of the schedule XML document referred to in this section must be compliant to the scheduleCfg schema. This schema can be found at the LOYTEC Web site. The XML documents can refer to the target namespace ‘http://www.loytec.com/xsd/scheduleCfg/1.0/’.

In that node, the following OPC tags are available:

- ServiceType (string, Read-only, const): This is a constant tag of type string, which contains “schedule”. It identifies this folder as a schedule folder. This can be used as an additional identification to the vendor-specific property of the folder tag.

- Schedule (string, read/write): This tag configures the schedule. The data type is string and the format is in XML. The XML document contains the *scheduleCfg* element as the root element.

- Caps (string, read-only): This tag contains the schedule capabilities. The data type is string and the format is in XML. The XML document contains the *scheduleCapabilities* element as the root element.

- CalItemPath (string, Read-only, const): This is an optional tag. If present, it contains the item path to the calendar object, that the schedule references. To read the calendar referenced by the schedule, use this item path and the “Calendar” item name to read the calendar XML document.

- EmbeddedCal (node): This is an optional OPC node. If present, it contains the OPC tags for the embedded calendar. The embedded calendar structure is as defined for calendar objects in Section 10.1.3.2.

10.1.3.2 Calendar Object

The device exposes the calendar objects to OPC XML-DA tags. Each calendar object is represented by a folder in the OPC name space. In that folder, the following OPC tags shall be available:

- ServiceType (string, Read-only, const): This is a constant tag of type string, which contains “calendar”. It identifies this folder as a calendar folder. This can be used as an additional identification to the vendor-specific property of the folder tag.

- Calendar (string, read/write): This tag configures the calendar. The data type is string and the format is in XML. This document contains the *calendarCfg* element as the root element.

- Caps (string, read-only): This tag contains the calendar capabilities. The data type is string and the format is in XML. The XML document contains the *calendarCapabilities* element as the root element.

10.1.3.3 Alarm Objects

The alarm objects on the device provide the *alarm summary* and can be used to acknowledge alarms. The alarm objects are exposed to XML-DA tags. Each alarm is uniquely identified by an XML alarm ID (XAID). The XAID must identify the alarm object and the alarm ID in that object. The XAID is used in the acknowledge service to identify the alarm. The XAID can also be transmitted in e-mail notifications.

Each alarm object is represented by a folder in the OPC name space. In that folder, the following OPC tags shall be available:

- ServiceType (string, Read-only, const): This is a constant tag of type string, which contains “alarm”. It identifies this folder as an alarm folder. This can be used as an additional identification to the vendor-specific property of the folder tag.

- Summary (string, Read-only): Reading from this tag, the current alarm summary can be obtained. The data type is string and the tag contains an XML document. This tag should not be subscribed to as it contains a large document. Subscribe to NotifyNewCnt instead, to get notified about new alarms. The root element of the XML document is the *alarmSummary* element.

- NotifyCnt (unsigned, Read-only): This tag is updated with an incremented notify count for each alarm update notification. This is the case for new or cleared alarm conditions, and for acknowledged alarms. Clients can subscribe to this tag in order to be notified about changes in the alarm summary. The client has then to read the complete alarm summary when notifications occur.

- NotifyNewCnt (unsigned, Read-only): This tag is updated with an incremented notify count each time a new alarm appears. This tag does not update when alarms are acknowledged or go inactive.

- Ack (string, Write): Writing to this tag acknowledges an alarm. The data type is string. The written data is an XML document, which contains the *alarmAck* element. The write must specify the XAID.

10.1.3.4 Trend Log Objects

Each trend log object on the device is represented by a folder in the OPC name space. This folder contains a number of tags describing and controlling the trend log. To retrieve log records, however, the XML-DA tag interface cannot be used. There are two options: retrieve the complete log as a CSV file, or use the LOYTEC proprietary Data Log Web service (XML-DL). That Web service uses the logHandle provided by a tag. The CSV file location can be obtained from a tag also.

- ServiceType (string, Read-only, const): This is a constant tag of type string, which contains “trendLog”, or “alarmLog”. It identifies this folder as a trend log, data log or alarm log folder. This can be used as an additional identification to the vendor-specific property of the node tag.

- Purge (Boolean, read/write): When writing TRUE to this tag, the log is purged.

- TotalCnt (unsignedInt, read-only): This tag contains the total number of logged records. This number can be larger than the BufferSize.

- BufferSize (unsignedInt, read/write): The size in records of the log buffer. Writing to this tag can resize the log buffer, if it is disabled.

- LogHandle (string, read-only, const): This handle specifies the data log. The logHandle must be used with the proprietary Data Log Web service.

- CsvFile (string, read-only, const): This tag specifies the file path and file name of the CSV data log file.

- CentralDL0, CentralDL1 (string, read/write): These tags are obsolete and kept for backward compatibility.

10.1.3.5 E-mail Templates

E-mail templates can be configured in the Configurator software. When an e-mail template is triggered, the corresponding e-mail is transmitted. The e-mail template can also be triggered over the OPC interface. Therefore, a node is added to the OPC name space for each e-mail template under the “E_Mail” node.

Each e-mail node is named after the e-mail template and contains the following OPC tags:

- ServiceType (string, Read-only, const): This is a constant tag of type string, which contains “email”. It identifies this folder as an e-mail template folder.

- Send (Boolean, read/write): When writing TRUE to this tag, the e-mail transmission is triggered.

10.1.4 OPC Groups

OPC groups are used to subscribe to data coming from OPC tags. The group specifies the subscribed tags and a server refresh rate, which the OPC server can interpret, how often it shall refresh data of the underlying data points.

In network technologies that are event-based, this refresh rate has no further impact, as the data is as fresh as possible. In technologies, that rely on polling, the OPC server activates dynamic poll-groups for the subscribed OPC tags. The data server of the device can then employ dynamic polling, if the technology supports it (see Section 6.2.2).

10.2 OPC UA Server

10.2.1 Introduction

OPC UA (Unified Architecture) is a new International standard (IEC 62541) designed for communication between information systems. It has been specified by the OPC Foundation which also released the “classical” OPC: DA, HDA, A&E and XML-DA. OPC UA has been conceived to be the successor of those old specifications in order to overtake some drawbacks they imply. L-INX and L-GATE models with enhanced security features have a built-in OPC UA server (see Section 1.9).

OPC UA innovates with some new features such as: Security based on certificates exchange, heartbeat for connections in both directions and acknowledgements of transmitted data. LOYTEC devices with a built-in OPC UA server use the binary protocol over TCP/IP giving you the best performance and least overhead while taking minimum resources.

The OPC UA server is accessible via two URIs:

opc.tcp://192.168.24.100:4840

https://192.168.24.100/UA

where the IP address has to be replaced with the actual IP address of the device. The default TCP port is 4840. It can be changed on the OPC UA configuration page of the Web interface. The OPC UA Server is also accessible via secure Web Services with HTTPS.

One of the most interesting features of OPC UA is built-in security. Therefore, the LOYTEC OPC UA server comes with default parameters only allowing secure connections. It is required to setup the OPC UA Server before establishing a connection with any OPC UA client. The Web interface Ethernet port configuration has OPC UA protocol settings and the Web interface **Certificates** page has a dedicated OPC UA tab.

10.2.2 OPC UA and Security

The LOYTEC OPC UA server can be enabled on the **Ethernet** tab of the **Port Config** page. To enable OPC UA select the OPC UA protocol as shown in Figure 241. The OPC UA protocol settings box allows you to configure different options for the OPC UA server: its transport protocols, its port number, the security policies and the user authentication modes available to OPC UA clients. Changing one of them requires rebooting the device.

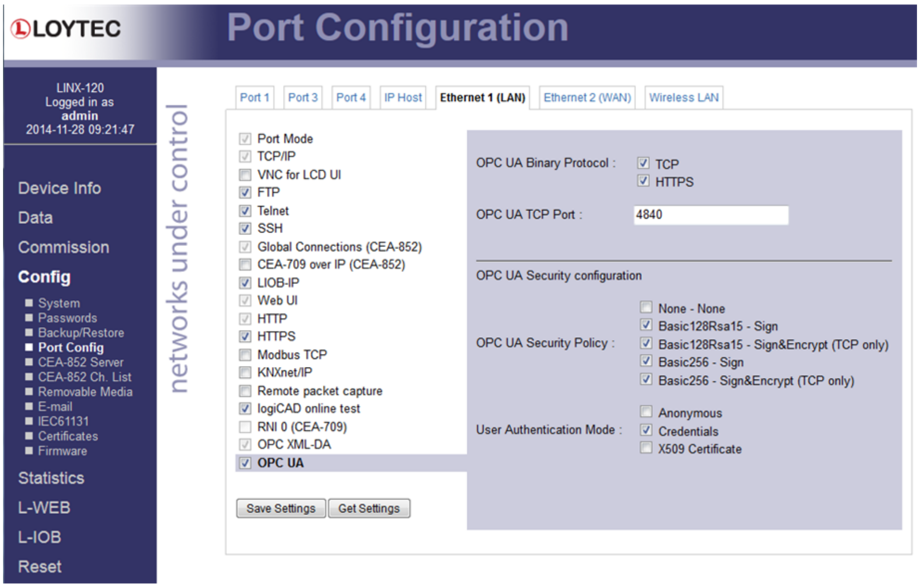


Figure 241: OPC UA server configuration

The OPC UA Server is accessible via two transport protocols: TCP or HTTPS. Both are using binary-encoded data. OPC UA over HTTPS requires activation of the HTTPS service on the device.

The default port number of the OPC UA server is 4840. This is the official registered port at the IANA (Internet Assigned Numbers Authority) for OPC UA TCP protocol for OPC Unified Architecture from OPC Foundation. It is also possible to change it to any other available port number. However, some OPC UA services (Find Servers and GetEndpoints) used to discover OPC UA server endpoints may be only available through the default port number 4840 depending of the client used.

Using the non-default port number might also require to establish a direct connection from your OPC UA client by giving the willing Security Policy and Authentication Mode without having the possibility to know and choose them with FindServers and GetEndpoints OPC UA services. Thus a warning message will appear when changing the port number to the non-default port number value.

By default the LOYTEC OPC UA server is configured to accept secure connections only, requiring certificate exchange. An unsecure mode, however, not requiring certificate exchange can be enabled if so required. The **OPC UA Security Policy** defines the level of security that will be applied to the OPC UA connection:

- **None – None**: Deactivated by default, this is the only security policy not requiring certificate exchange. This connection security policy shall be avoid as much as possible since security is a requirement for any OPC UA Clients, they must have the possibility to establish secure connections and to manage certificates exchange.

- **Basic128Rsa15 – Sign**: The first level of security for an OPC UA connection. Messages between Client and Server will be signed by both Client and Server certificates. However messages will not be encrypted and remain readable with a Network Analyzer such as Wireshark. This security policy uses RSA15 as Key-Wrap-algorithm.

- **Basic128Rsa15 – Sign&Encrypt**: This security policy is using RSA15 as a Key-Wrap-algorithm and 128-Bits key length for encryption algorithm. This Security policy uses encryption implying that message will not be readable from Network Analyzer.

- **Basic256 – Sign**: The third level of security for an OPC UA connection. Messages between Client and Server will be signed by both Client and Server certificates. However the connection will not be encrypted and messages remain readable by a Network Analyzer.

- **Basic256 – Sign&Encrypt**: The safest security policy available on LOYTEC devices using OPC UA. Messages are signed with RSASHA1 and encrypted using a 256Bits key length. This Security policy uses encryption implying that message will not be readable from Network Analyzer. When available on the OPC UA Client, this must be the preferred connection.

The **User Authentication Mode** on the LOYTEC OPC UA Server includes three possible ways for OPC UA clients to authenticate. Each authentication mode could work independently of the Security Policies enabled. By default only the **Credentials** authentication mode is activated. The following authentication modes can be activated:

- **Anonymous**: Not activated by default. This authentication mode does not require actions from users. As security is one major aspect of OPC UA, it should not be used unless for testing.

- **Credentials**: This authentication mode requires a user account with a password to establish an OPC UA connection. Accounts must be managed through the Web interface Account Management. OPC UA does not require any special accounts.

- **X.509 Certificate:** By default this option is not activated. This authentication mode uses X.509v3 certificates to assure identity of OPC UA clients. The certificate used with that authentication mode need also be added to the Trusted Clients List before establishing a connection.

10.2.3 OPC UA Trusted Clients

OPC UA security is based on X.509 certificates exchange. By default a LOYTEC OPC UA server comes with a pre-installed self-signed certificate that will have to be trusted by your OPC UA client. The OPC UA server also requires trusting OPC UA clients before establishing a connection. It is possible to upload a new self-signed or signed-by-CA certificate through the WebUI.

The **Certificates** page on the Web interface has a dedicated **OPC UA** tab allowing you to manage X.509 certificates by getting the OPC UA server certificate from the device, trusting a new OPC UA client, rejecting a trusted OPC UA client or deleting a trusted/rejected OPC UA client certificate.

The OPC UA specification requires that each managed certificate must be compliant with the X.509v3 certificate standard. The LOYTEC OPC UA server also requires that every OPC UA client certificate must be encoded into the binary .DER format.

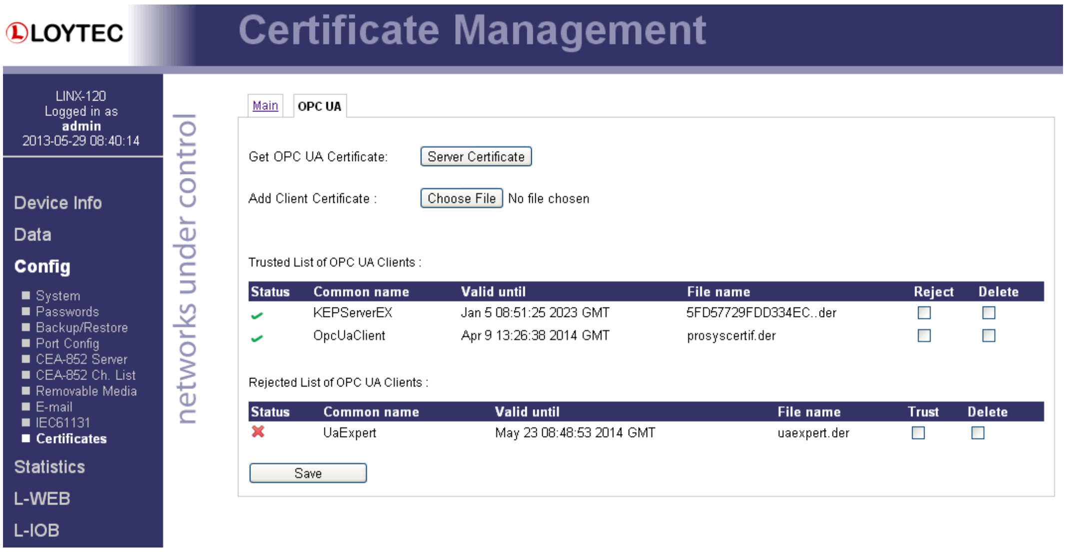


Figure 242: OPC UA client certificate management.

To allow a secure connection initiated by an OPC UA client, the OPC UA server needs to find the client's X.509 certificate in its **Trusted List of OPC UA Clients**. To trust a new OPC UA client, add a new client certificate by clicking on the **Choose File** button. A popup window will be opened asking you to provide a valid X.509v3 Certificate. Once a correct OPC UA client certificate has been added to the trusted list of OPC UA clients, a client will be granted access to the OPC UA server. The Web interface displays some internal certificate information such as the common name or the time limit of validity. No reboot is necessary after adding a trusted client certificate.

The list of trusted OPC UA clients can be edited. The following actions are available:

- **Reject a trusted OPC UA Client**: Check the reject checkbox of one or more certificates in the trusted list and click **Save**. The certificates are now be considered as rejected and a secure connection with the selected clients will not be possible until they are added back to the trusted list. They appear in the rejected list.

- **Trust a rejected OPC UA Client:** Check the trust checkbox of one or more certificates in the rejected list and click **Save**. The selected certificates will be moved to the trusted list again and a secure connection with the selected clients can be established again.

- **Delete a trusted/rejected OPC UA Client:** Check the delete checkbox of one or more certificates in either the trusted or rejected list and click **Save**. The selected certificates will be entirely removed from the LOYTEC device.

10.2.4 OPC UA Client Setup

The last step before establishing a secure connection is on the OPC UA clients side. The OPC UA client also requires trusting a LOYTEC OPC UA server before connecting to it. This is usually done by adding the LOYTEC device's certificate to the trusted server list of the OPC UA client. The LOYTEC OPC UA server X.509v3 Certificate is downloadable by clicking on the **Server Certificate** button. Then add it to the trusted list of OPC UA server of your client.

Some OPC UA clients may not require to be configured with a trusted server certificate and will ask you, if you want to trust it when connecting. It may also warn that the pre-installed LOYTEC OPC UA server certificate is a self-signed certificate. However, installing the

LOYTEC server certificate in your client list of trusted OPC UA servers is the safest approach since it reliably prevents man-in-the-middle attacks.

Note : *Installing a new Server certificate in the WebUI with a key size > 4096 bits might lead to connection problems with some OPC UA Clients !*

10.2.5 OPC UA Address Space

The data point hierarchy as configured by the Configurator software is exposed to the OPC UA address space by the device. This is done internally for all data points, which are marked for OPC exposure (i.e., have the OPC check-mark set). That hierarchy is accessible under the folder Loytec ROOT.

Folders are translated into OPC UA folder object. Any of the data points classes, analog, binary, multi-state, are exposed as OPC UA variables with OPC UA properties. Any of the data point classes string and user are exposed as OPC UA variables. An example of browsing the OPC UA address space on the device is shown in Figure 243.

All OPC UA variables also possess a Node ID by which they can be identified by an OPC UA client alternatively to the full browse path. The node ID remains constant between device restarts and is derived from the data point ID:

UA Node ID = (Data Point ID * 16) + 20000.

The OPC UA specification requires some special nodes, which are not part of the data point hierarchy. They can be found under the folder 'Types' and under the OPC UA object server.

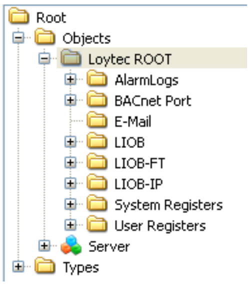


Figure 243: Client browsing the OPC UA address space on a LOYTEC device.

10.2.5.1 OPC UA Data Types

OPC UA specification defines some special data types used as internal structures by OPC UA node instances.

- **NodeId**: A structure containing a unique identifier in an OPC UA server used to address the node by OPC UA services. The LOYTEC OPC UA server only uses numeric NodeIds.

- **QualifiedName**: The structure used when browsing the OPC UA server. It also contains a unique identifier used for multiple address spaces. In the LOYTEC OPC UA server, this identifier is '0'.

- **LocalizedText**: A structure composed of two strings. The LOYTEC OPC UA server always contains 'en' as the **Locale** and the data point's value as **Value**. For example, an attribute using this data type is the optional description of a data point.

- **DataValue**: The structure used to expose the value. It contains the variant value, a status (Integer) and two timestamps, one for the OPC UA server read time, the other one for the last data point update.

10.2.5.2 Common Attributes

Scalar data points exposed to OPC UA contain the following list of attributes common to analog, binary, multi-state data points:

- **NodeId** (NodeId): The unique numeric identifier of a data point in the OPC UA server.
- **NodeClass** (Integer): This attribute indicates that the OPC UA node is a variable and is equal to ‘2’.
- **BrowseName** (QualifiedName): The name of the data point used by the browse service.
- **DisplayName** (LocalizedText): The name of the data point used by the read service.
- **Description** (LocalizedText): The optional attribute containing the description of the data point.
- **WriteMask/UserWriteMask** (Byte): Optional attribute defining which attributes of the OPC UA variable are writable. In the LOYTEC OPC UA server, only the value is writeable so WriteMask and UserWriteMask contain ‘0’.
- **Value** (DataValue): The actual variant value of the data point, the data point’s times tamp of the last update, the OPC UA server time stamp of the last update and the actual OPC UA status code of the data point.
- **DataType** (NodeId): The actual data type of the OPC UA variable.
- **ValueRank** (Integer): This attribute classifies the value as a scalar or an array.
- **ArrayDimensions** (Integer): If the value is an array, this attribute specifies the size of that array.
- **AccessLevel/UserAccessLevel** (Byte): Contains information whether the OPC UA node is readable and/or writable.
- **MinimumSamplingInterval** (Double): Optional attribute providing a sample interval used for subscriptions. This is set to ‘0’ (default) in each OPC UA variable.
- **Historizing** (Boolean): Indicates whether the server is currently using that OPC UA variable as an historical data point.

10.2.5.3 Analog

Analog data points are exposed as one OPC UA variable with three Properties to OPC UA nodes. The OPC UA variable holds the data point’s name. The properties are: EU Range, EU Type and EU Units, where EU stands for engineering units.

The EU Range property contains an Array value of Double with the analog maximal value and analog minimal value of the data point. The EU Type contains a scalar value ‘1’ for analog data points and the EU Units property holds a string value for the human-readable engineering unit text of the data point.

10.2.5.4 Binary

Binary data points are exposed as one OPC UA variable with two Properties to OPC UA nodes. The OPC UA variable holds the data point’s name. The properties are: EU Info, EU Type.

The EU Info property contains an array value of strings with active and inactive text of the data point. The EU Type contains a scalar value ‘2’ for binary data points.

10.2.5.5 Multi-state

Multi-state data points are exposed as one OPC UA variable with two Properties to OPC UA nodes. The OPC UA variable holds the data point’s name. The properties are: EU Info, EU Type.

The EU Info property contains an array value of strings with the different state texts of the data point. The EU Type contains a scalar value ‘2’ for a multi-state data point.

10.2.5.6 User Type

Data points of class user are exposed as a one-to-one mapping to OPC UA nodes. The OPC UA node value is a hexadecimal string without whitespace representing the byte array, e.g., ‘B034’.

10.2.5.7 String

String data points are exposed as a one-to-one mapping to OPC UA nodes. The OPC UA value of a string OPC UA variable contains the string data, e.g., ‘Room4’.

10.2.5.8 Structured Data Points

In OPC UA, structured data points are represented as in OPC XML-DA: one user-type data point containing the entire structure value as a byte array. The respective structure fields are created as sub-data point of the appropriate class. For detailed information on how structure sub-data points are mapped to OPC variables please refer to Section 10.1.2.6.

10.2.6 AST Objects

The alarming, scheduling, and trending (AST) objects are more complex than regular data points. Therefore, the device exposes AST objects as a set of OPC UA nodes describing the object as in the OPC XML-DA Server. All tags for one AST object are collected under an OPC UA node representing the AST object. For a detailed description of the involved OPC variables please refer to Section 10.1.3.

10.2.7 Subscriptions and Monitored Items

The LOYTEC OPC UA server allows users to create and manage subscriptions and monitored items. Subscriptions enable monitoring a group of monitored items. The total number of subscriptions that may be active at a time on a LOYTEC device is 64. The user can setup the following parameters for its favorite OPC UA client:

- **PublishingInterval**: The interval in ms between each publish response. The minimum publish interval value is 1second and the maximum is 86,400,000ms (1 day).

- **LifetimeCount**: The number of times the publish interval can expire without the server sending data updates or keep-alive messages. It must be at least 3 times the MaxKeepAliveCount. The valid range is 3 to 4,294,967,295 (MaxUInt32).

- **MaxKeepAliveCount**: This number specifies the number of publish intervals that must expire before a keep-alive message is sent. The valid range is 1 to 1,431,655,765 (MaxUInt32/3).

- **Priority:** This byte number indicates the relative priority of one subscription. Subscription notifications are sent regarding subscription priorities when there is more than one subscription active on the OPC UA Server. This parameter can not be managed by all OPC UA Clients.

A monitored item is created for each OPC UA node requested to be monitored. When data points associated with OPC UA nodes, which are linked to the created monitored items, are updated, a new publish response is sent containing the updated value.

AST object data points cannot be monitored excepted CentralDL0, CentralDL1, TotalCnt of Trend Log Objects and NotifyCnt, NotifyNewCnt of Alarm Objects.

Parameters for monitored items on LOYTEC device are:

- **SampleInterval**: The rate, at which the monitored items are sampled. It is always the interval of the subscription's publishing interval.
- **QueueSize**: The OPC UA Server supports queue size <= 2 as required by the Embedded OPC UA Server Profile.When QueueSize =2 and a monitored Items changes more than 1 time between a timelapse < PublishInterval, 2 values will be sent to Client.
- **DiscardOldest**: This parameter only matters if QueueSize > 1. If true, the oldest value gets deleted from the subscription DataNotification queue, if False the last value added to the subscription DataNotification queue gets replaced with the new value.
- **DataChangeFilter**: The OPC UA Server only supports DataChangeTrigger: Status/Value and Status/Value/Timestamps will only send a DataNotification when value has been updated! The default value is Status/Value. Deadband is only supported by datapoints with Numeric values.

10.2.8 OPC UA Statistics

For analyzing problems with OPC UA communication the device offers statistics information. The Web interface displays that information on the **OPC UA Server** page of the **Statistics** menu. Figure 244 shows a typical output of such statistics information.

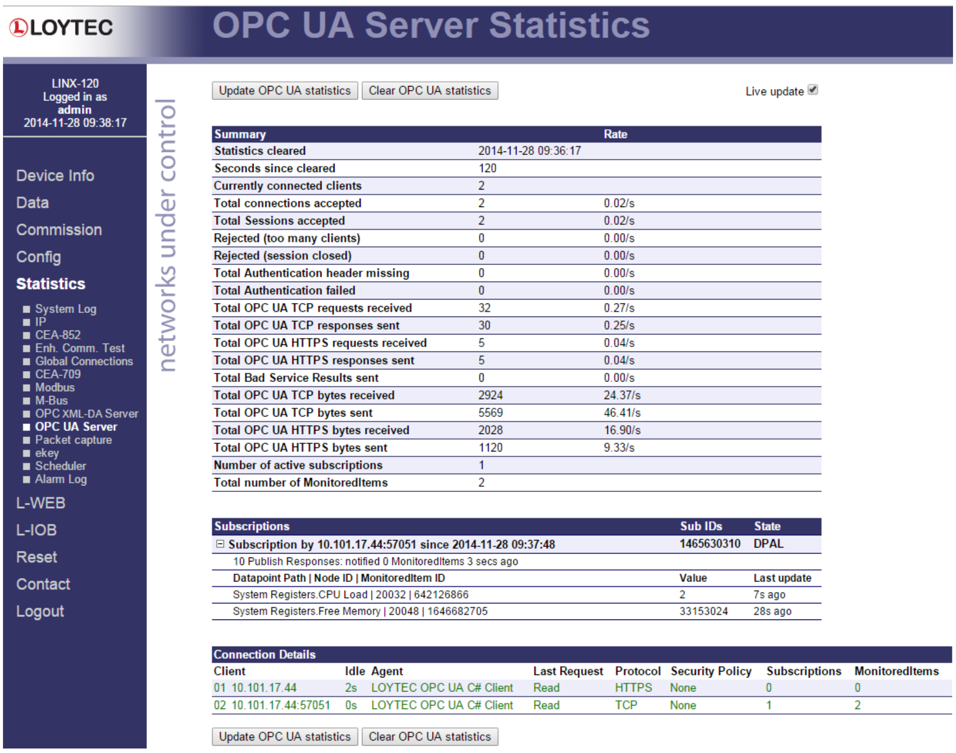


Figure 244: OPC UA Server Statistics.

The following information is available:

- Currently connected Clients: Number of OPC UA clients currently connected to the OPC UA server.
- Total connections accepted: Number of connection accepted since last cleared.
- Total Session accepted: Number of sessions accepted since last cleared.
- Rejected (too many clients): Number of sessions rejected due to too many clients already connected since last cleared.
- Rejected (Session closed): Number of sessions rejected due to the fact that the session has been closed by the server or the client.
- Total Authentication header missing: Number of sessions opened without a matching authentication mode in the server.
- Total Authentication failed: Number of sessions opened without valid authentication.
- Total OPC UA TCP requests received: Number of OPC UA TCP requests received since the last statistic reset.
- Total OPC UA TCP responses sent: Number of OPC UA TCP responses sent since the last statistic reset.
- Total OPC UA HTTPS requests received: Number of OPC UA HTTPS requests received since the last statistic reset
- Total OPC UA HTTPS responses sent: Number of OPC UA HTTPS responses sent since the last statistic reset.
- Total Bad Service Results sent: Number of Bad Response sent in OPC UA TCP Responses.
- Total OPC UA TCP bytes received: Number of TCP bytes received by the OPC UA server.
- Total OPC UA TCP bytes sent: Number of TCP bytes sent by the OPC UA server.
- Total OPC UA HTTPS bytes received: Number of HTTPS bytes received by the OPC UA server.
- Total OPC UA HTTPS bytes sent: Number of HTTPS bytes sent by the OPC UA server.
- Number of active subscriptions: Total number of subscriptions for all the currently connected clients.
- Total number of monitored items: Total number of monitored items for all the currently connected clients.

10.2.9 Error Codes and Solutions

The OPC UA specification defines numerous status codes. Table 23 lists some of them which may occur and provides possible solutions to resolve those problems.

OPC UA Error Code	Meaning	Solutions
BadNotReadable	OPC UA client read a non-readable node.	Set the data point to readable in the Configurator Software.
BadNotWritable	OPC UA client tried to write to a non-writeable node.	Set the data point to writeable in the Configurator Software.
BadNodeIdInvalid	This node does not support the requested operation	Trying to add non-data point to subscription lead to that error codes.
BadIdentityTokenInvalid	OPC UA client tried to connect with a non-valid authentication mode	Change the authentication mode of the client or activate the desired authentication mode on the server.
BadServiceUnsupported	OPC UA client tried to use a currently unsupported service.	The OPC UA server does not implement all available OPC UA services. Operations like adding a node to the address space or using a method are currently not supported.
BadUserAccessDenied	OPC UA client did not send the right credentials.	Connect with one of the account credentials available on the device.
BadTooManyMonitoredItems	OPC UA server cannot create any more monitored items for that subscription.	Create a new subscription and add the monitored items wanted. The maximum number of monitored items per subscription is 5000.
BadTooManySubscriptions	OPC UA server can not create more subscriptions.	Delete one active subscription to create a new one. The maximum number of subscriptions is 64.
BadTooManySessions	OPC UA server cannot create more Sessions.	Disconnect one active session to create a new one. The maximum number of active sessions is 32.
UncertainInitialValue	OPC UA server does not know the status of the data point	Configure a default value for this data point in the Configurator software.
BadCertificateTimeInvalid	OPC UA Client certificate time entry is not valid.	Verify that your certificate time (notBefore and notAfter fields) is matching with the LINX Local Date.
BadCertificateRevoked	OPC UA Client certificate is revoked.	Verify that the client certificate is not in the LINX Rejected List.
BadCertificateUntrusted	OPC UA Client certificate is not trusted.	Verify that the client certificate is in the LINX Trusted List.
BadCertificateHostNameInvalid	OPC UA Client certificate host name is not valid.	Verify that the certificate HostName is valid.
BadCertificateInvalid	Client certificate is not valid.	The OPC UA Client certificate sent is not valid. Verify that information and keys are correct.
BadSecurityChecksFailed	OPC UA Client certificate is not valid.	OPC UA does not allow sending internal certificate error code to Client. Log into the WebUI and check the System Log to know the right Error Code.

Table 23: OPC UA Error Codes and Solutions

10.3 Using L-WEB

The L-WEB is a Web-based visualization software that comes free with the device. It uses the standard Web technologies to visualize and control data provided by one or more L-INX Automation Servers on a Windows PC.

The L-WEB software uses the standardized OPC XML-DA Web service to communicate between L-WEB and remote L-INX Automation Servers, which makes it extremely firewall-friendly and easy to setup.

The graphical design of the L-WEB user interface consists of pages, which can simply be created by using the L-VIS/L-WEB Configurator software without any know-how in HTML, Java, etc. Dynamic information is shown in the form of numeric values, text, changing icons, bar graphs, trend logs, alarm and event lists, or schedule controls.

The complete set of automation functions of the L-INX Automation Server is fully supported by L-WEB. The automation services are residing in the embedded devices and are distributed over the network to build up a dependable system with L-WEB only accessing these services. Furthermore, any kind of calculations, data point connections, etc., are implemented on the embedded Automation Server, which makes the application on the Automation Server completely independent from the connection to the L-WEB application.

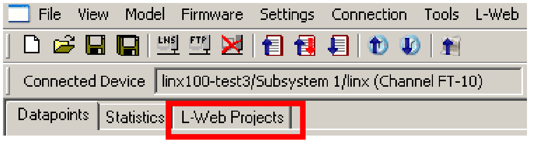
Starting from the data point configuration, the user can create an L-WEB project. The L-WEB project contains the data point configuration of the Web service interface and a graphical design for the L-WEB user interface. For more information on creating graphical designs using the L-VIS/L-WEB Configurator software refer to the L-VIS User Manual [6].

10.3.1 Create a new L-WEB Project

The Configurator provides the data point configuration, which is downloaded to the device. On top of that configuration, an L-WEB design can be created for visualization.

To Create an L-WEB Project

1. Start the Configurator software and change to the **L-WEB Projects** tab.



2. The L-WEB project tab appears as in Figure 245.

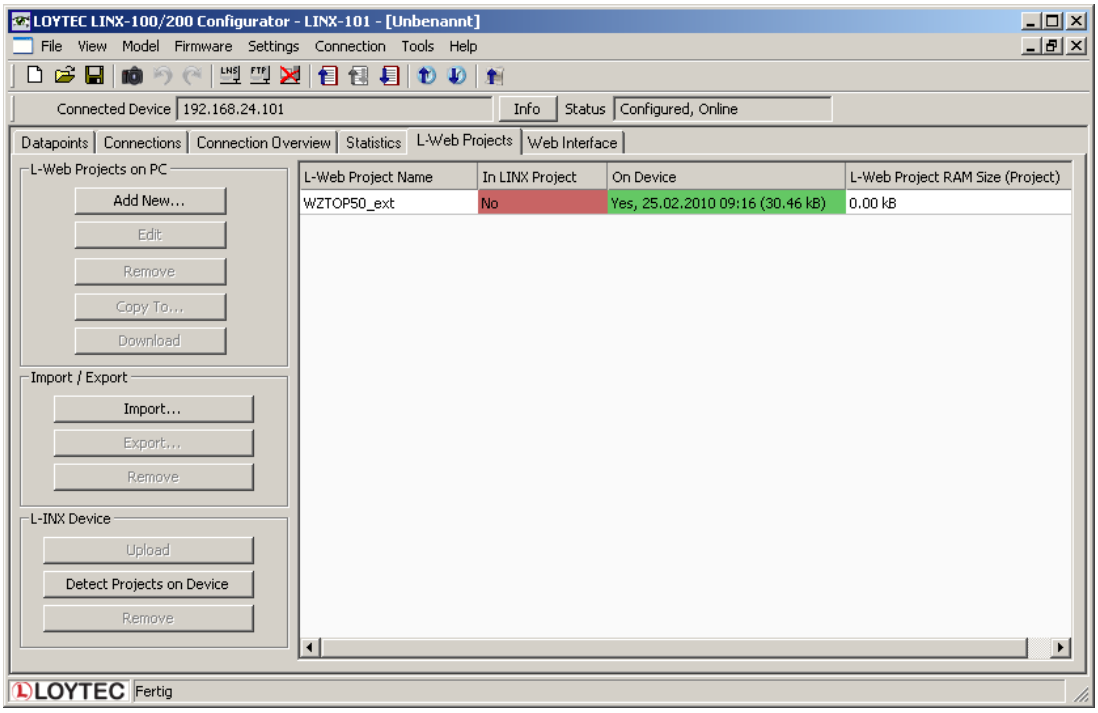
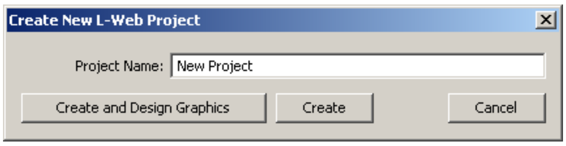


Figure 245: L-WEB Projects Tab.

3. Click on **Add New …**

4. Enter a new **Project Name**.



5. Click on **Create**. The new project appears in the projects list.

10.3.2 Start a Graphical L-WEB Design

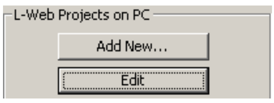
The L-WEB graphical design tool is started from within the L-WEB projects tab. The graphical design for the L-WEB project is created in the L-VIS design tool (L-VIS/L-WEB Configurator). The data point configuration created in the Configurator project is available for the L-WEB project and its graphical design.

To Start a Graphical Design

1. Select the **L-WEB Projects** tab.

2. Select an L-WEB project.

3. Click **Edit**.



4. This opens the L-VIS graphical design tool. Complete the graphical design in the tool and click the **Write Project to Device** speed button

5. The graphical design is now part of the project.

L-Web Project Name	In LINX Project	On Device	L-Web Project RAM Size (Project)
New Project	Yes, 28.02.2010 17:01 (5.32 kB)	No	57.17 kB

Note: *If the Configurator had been connected to the device, the graphical design would have been added to the device in the same step.*

10.3.3 Organize L-WEB Projects

L-WEB projects can be organized within the L-INX configuration project. L-WEB projects can be part of the configuration project and/or stored on the device. For instance, the configuration project may contain a number of L-WEB projects, but for saving space on the device, only one of them is downloaded on the device. The **L-WEB projects** tab provides a number of tools to organize a set of L-WEB projects.

To Organize L-WEB Projects

1. Connect to the device as described in Section 7.6.1.

2. Select the **L-WEB Projects** tab.

3. Click **Detect Projects on Device**. This scans for all projects found on the device.

L-Web Project Name	In LINX Project	On Device	L-Web Project RAM Size (Project)
New Project	Yes, 28.02.2010 17:01 (5.32 kB)	No	57.17 kB
Second Project	Yes, 28.02.2010 17:03 (5.31 kB)	Yes, 28.02.2010 16:03 (5.31 kB)	57.18 kB

Projects marked as a green **Yes** in the **In LINX Project** column are L-WEB projects, which are part of the current L-INX configuration project. Projects marked as a green **Yes** in the **On Device** column are L-WEB projects, which are also stored on the device. A red **No** identifies the L-WEB project to be missing in the project or on the device, respectively.

4. If you want to download an L-WEB project to the device, which is missing there, select the project and click **Download**. After the download the project appears with a green **Yes** in **On Device**.

L-Web Project Name	In LINX Project	On Device	L-Web Project RAM Size (Project)
New Project	Yes, 28.02.2010 17:06 (5.32 kB)	Yes, 28.02.2010 16:06 (5.32 kB)	57.17 kB
Second Project	Yes, 28.02.2010 17:07 (5.31 kB)	Yes, 28.02.2010 16:07 (5.31 kB)	57.18 kB

5. If you want to remove a project from the device, click **Remove** in the **LINX Device box**.

L-Web Project Name	In LINX Project	On Device	L-Web Project RAM Size (Project)
New Project	Yes, 28.02.2010 17:06 (5.32 kB)	Yes, 28.02.2010 16:06 (5.32 kB)	57.17 kB
Second Project	Yes, 28.02.2010 17:03 (5.31 kB)	No	57.18 kB

6. If you want to remove the project from the current L-INX project file, click **Remove** in the **L-WEB Projects on PC** box.

L-Web Project Name	In LINX Project	On Device	L-Web Project RAM Size (Project)
New Project	Yes, 28.02.2010 17:06 (5.32 kB)	Yes, 28.02.2010 16:06 (5.32 kB)	57.17 kB
Second Project	No	No	0.00 kB

7. If you want to export the L-WEB project into a separate L-WEB project file, click **Export…** and select a file name in the file requestor dialog.

8. If you want to import an L-WEB project from a separate L-WEB project file, click **Import…** and select the file in the file requestor dialog. The L-WEB project appears in the project but not on the device.

10.4 Using Custom Web Pages

Custom Web pages can also be developed for the L-INX. For doing so, the applications engineer must implement an OPC XML-DA Web service client, which adheres to the WSDL interface. This can be done in C++ or script languages such as Perl. The WSDL must be obtained from the OPC Foundation's Web site following the OPC XML-DA namespace http://opcfoundation.org/webservices/xmlda/1.0/.

Any Web content, including scripts, applications or static Web pages can be stored directly on the L-INX's file system. Use the admin account to upload the content via FTP into the directory

```
/var/www
```

For example, a page named 'my_page.html' put directly into '/var/www' can be accessed via 'http://192.168.24.100/my_page.html', given that the IP address is correct.

11 M-Bus

11.1 Introduction

The M-Bus (Meter-Bus) is a European standard (EN 13757-2, EN 13757-3) designed for remote reading of meters. With its standardization as a galvanic interface for remote readout of heat, water, and energy meters, this bus has become an important interface for automatic meter reading applications with different vendor's meters on the same cable.

The M-Bus is a serial bus, which is controlled by a single bus master. This master can request data from several slave devices connected to the network. The data transmission from master to slave is done by a modulation of the output voltage (36 V means a logical '1', 24 V means logical '0'). During data transmission from slave to master the current is modulated (1.5 mA represent the logical '1', 11-20 mA represent a logical '0'). M-bus devices can be powered over the bus. The number of devices which can be powered depends on the M-Bus transceiver used.

11.2 Hardware Installation

For using the M-Bus with the device an external M-bus interface with an RS-232 connector is required. The external interface must be connected to the LOYTEC device either via the serial connector or the extension port EXT. The M-Bus functionality must be enabled on the device itself.

11.2.1 Console Connector

When using the serial console connector, the M-Bus interface is enabled on the device by setting DIP-switch 7 to ON. This disables the console of the device and activates M-Bus over the console connector. After the device is rebooted, the M-Bus is active on the device. When rebooting the device with M-Bus activated and the console connected, a couple of boot messages are displayed in the console, before the console is turned off.

If the console is needed again, set the DIP switch 7 to OFF and reboot the device. This is required, if the firmware shall be updated over the serial port using the LSU tool.

When using the L-MBUS coupler, connect the L-MBUS to the device's console connector via a null-modem cable (female to female sub-D connector) as shown in Figure 246. On the L-MBUS remove the jumper as indicated on the front label The null-modem cable must be an 8-wire, single shielded, type AWG28 with the pin assignment according to Table 24.

Terminal 1	Terminal 2
1	4
2	3
3	2
4	1
5	5
7	8
8	7
9	9
S	S

Table 24: Pin assignment for the L-MBUS null-modem cable

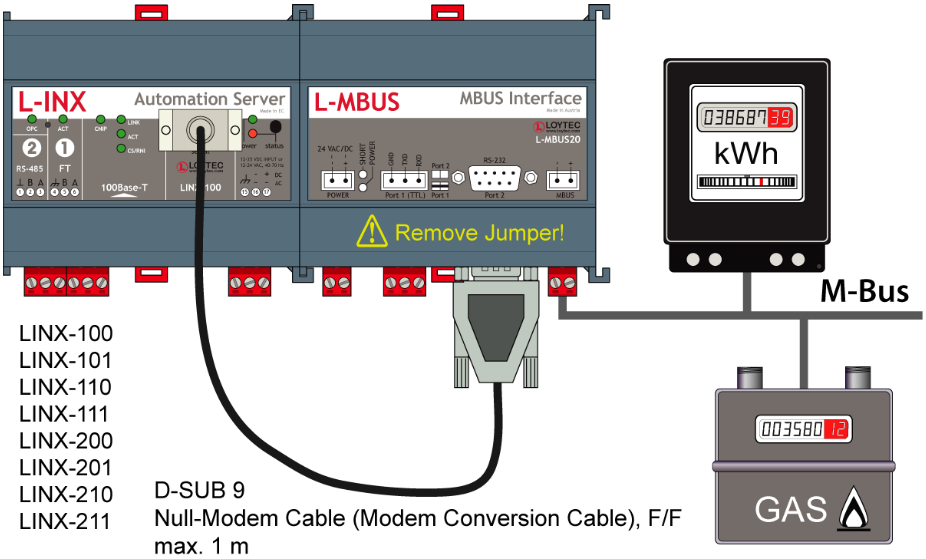


Figure 246: Connecting L-MBUS over console connector.

11.2.2 Extension Port

Devices that do not have a serial console connector provide an extension port marked **EXT** on the device as shown in Figure 247. Those devices need to use the L-MBUS converter. On the L-MBUS set the jumper as indicated on the front label. Follow the cabling instructions of Figure 247 between the **EXT** port of the L-INX and the **PORT1** on the L-MBUS.

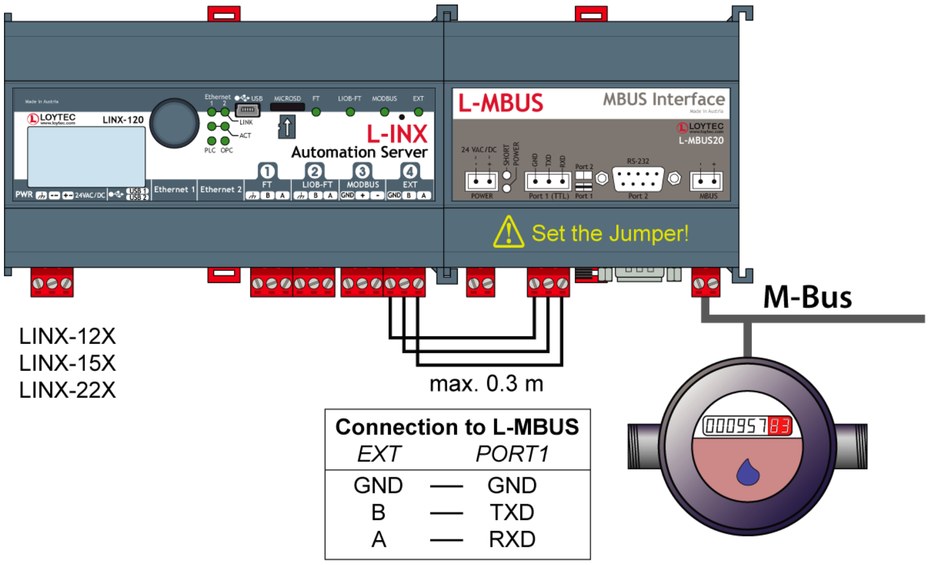


Figure 247: Connecting L-MBUS over the extension port.

11.3 M-Bus Network

The M-Bus network utilizes a two-wire connection. Several M-Bus slave devices are connected in parallel to the transmission medium. Each M-bus device has a primary address in the range from 0 to 250. This primary address must be unique for each device. Also a secondary address can be used for the slave devices. The secondary addressing mode is currently not supported by the LOYTEC device. The M-Bus network also supports two kinds of broadcast messages. A broadcast to address 255 does not force the slaves to give a response. A broadcast to address 254 forces an answer from the slave. This broadcast message is mainly used for peer-to-peer connections.

The M-Bus allows the use of devices with different Baud rates of up to 9600 Baud. M-Bus devices are not always able to fulfill the complete functional specification of the M-Bus standard. For readout two different modes are known (some slave devices implement both):

- A default read usually reads all the data of a device.
- A selective read selects the data points, which are to be read.

Some devices also support writing special data points.

Some M-Bus devices support a synchronized action. This means that when a device receives such a synchronize command, it stores specific data points for later readout. This way, specific information of even a larger number of devices can be read out in a synchronized manner.

M-Bus supports network management functions such as changing the primary address of a device, changing the Baud rate of a device, reading all data from a device as well as pinging a device. It is therefore possible to scan M-Bus devices in an M-Bus network.

M-Bus data points are specified by a DIF/DIFE and VIF/VIFE combination. The DIF/DIFE (data information field and data information field extension) specify storage number, tariff, subunit as well as the data coding (BCD, int, etc.) and the function (min, max, etc.) of the data point. The DIF/DIFE can be up to 11 bytes long (1 byte for the DIF and up to 10 extensions). The VIF/VIFE combination (value information field and value information field extension) specifies the type of the data point (e.g. energy count value) and how it is presented (e.g., the value is given in “Wh”). Like the DIF/DIFE, the VIF/VIFE can consist of one VIF and up to 10 VIF extensions. It defines the fixed network unit of an M-Bus data point. The Configurator allows the configuration by either entering the DIF/DIFE combination or by specifying the appropriate numbers.

11.4 Web Interface

This section describes the Web interface for the M-Bus port.

11.4.1 Configuration

A configuration of the M-Bus port is not necessary, as no parameters are required for the M-Bus.

11.4.2 Data Points

M-Bus data points can be accessed through the Web UI as described in Section 5.3.1.

11.4.3 Commission

The commissioning Web UI allows assignment of physical devices to existing devices in the data point configuration, that have been created with the commission later option. Under the **Commission** menu choose the M-Bus technology to open the M-Bus commissioning interface.

The Web page shows a list of all **Devices in configuration**. An example is shown in Figure 252. Each line displays the device name, the primary **Address** and the secondary address (**ID**). The **Status** column shows their current status. It can be one of the following:

- OK: The device is configured for communication.
- Offline: The device is configured for communication but appears offline.
- Uncommissioned: The device is not yet commissioned.
- Disabled: The device is disabled.

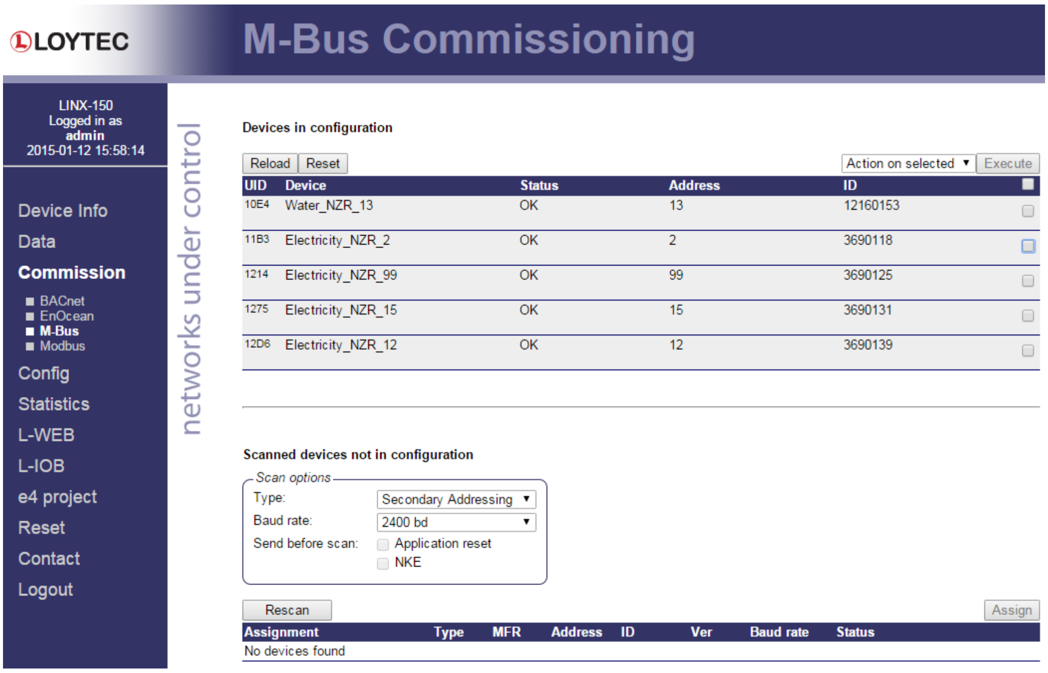


Figure 248: M-Bus commissioning Web interface.

In order to execute an action on devices, select the checkbox at the end of the line. Then choose an action in the drop-down **Action on selected** and click on the **Execute** button. Actions that can be executed on all devices are enable and disable. A disabled device will stop communication on the network until it is enabled again.

Those devices created as commission later can be assigned to physical devices on the network. The device description displayed beneath the device name can be edited, where the edit symbol appears. The assignment can be done manually by editing the fields in the **Address** and **ID** column or by executing a network scan. Edit the scan options as appropriate for your M-Bus network and click on **Rescan**. The scan progress will be displayed and fill the list for **Scanned devices not in configuration**. An example is shown in Figure 249.

Rescan Assign

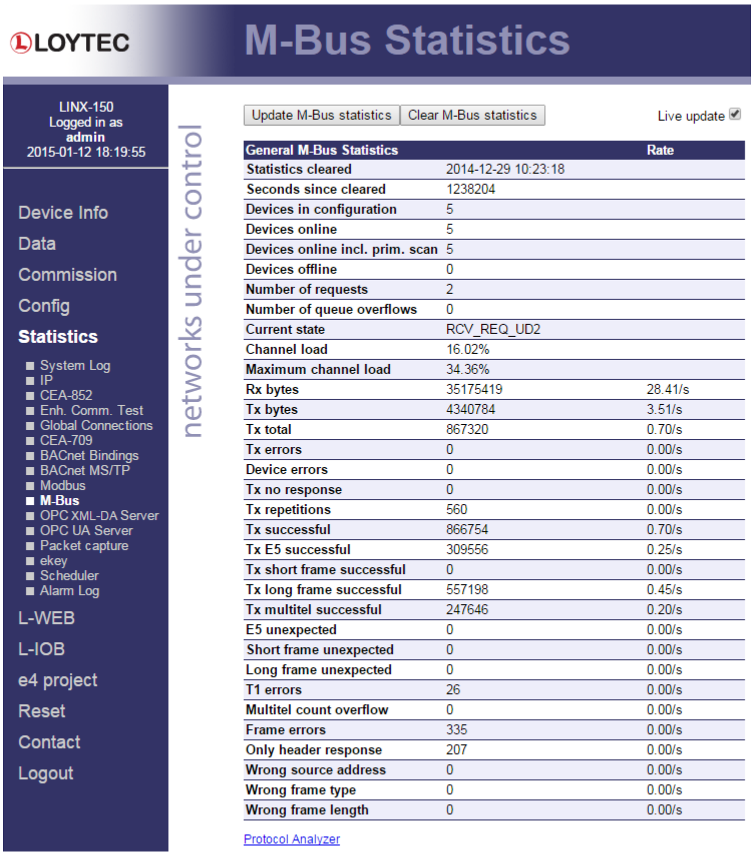
Assignment	Type	MFR	Address	ID	Ver	Baud rate	Status
Electricity_NZR_2	Electricity	NZR	2	3690118	0	2400	OK
Electricity_NZR_12	Electricity	NZR	12	3690139	0	2400	OK
Water_NZR_13	Water	NZR	13	12160153	2	2400	OK
Electricity_NZR_15	Electricity	NZR	15	3690131	0	2400	OK
Electricity_NZR_99	Electricity	NZR	99	3690125	0	2400	OK

Figure 249: Result of the M-Bus scan on the Web interface

To assign a scanned device to an uncommissioned device in the configuration, select the corresponding device name from the drop-down box in the **Assignment** column. Repeat that for all other devices and then click the button **Assign**.

11.4.4 Statistics

Figure 250 shows a typical output of the statistics information which can be displayed for the M-Bus port. The statistics can be cleared for each M-Bus port separately by pressing the **Clear M-Bus statistics** button. A refresh of the statistics is done automatically. To stop automatic update, deselect the **Live update** checkbox. For manual update press **Update M-Bus statistics**.



General M-Bus Statistics		Rate
Statistics cleared	2014-12-29 10:23:18	
Seconds since cleared	1238204	
Devices in configuration	5	
Devices online	5	
Devices online incl. prim. scan	5	
Devices offline	0	
Number of requests	2	
Number of queue overflows	0	
Current state	RCV_REQ_UD2	
Channel load	16.02%	
Maximum channel load	34.36%	
Rx bytes	35175419	28.41/s
Tx bytes	4340784	3.51/s
Tx total	867320	0.70/s
Tx errors	0	0.00/s
Device errors	0	0.00/s
Tx no response	0	0.00/s
Tx repetitions	560	0.00/s
Tx successful	866754	0.70/s
Tx E5 successful	309556	0.25/s
Tx short frame successful	0	0.00/s
Tx long frame successful	557198	0.45/s
Tx multitel successful	247646	0.20/s
E5 unexpected	0	0.00/s
Short frame unexpected	0	0.00/s
Long frame unexpected	0	0.00/s
T1 errors	26	0.00/s
Multitel count overflow	0	0.00/s
Frame errors	335	0.00/s
Only header response	207	0.00/s
Wrong source address	0	0.00/s
Wrong frame type	0	0.00/s
Wrong frame length	0	0.00/s

Figure 250: Statistics of the M-Bus port.

The following information is available:

- Statistics cleared: last time of statistics reset
- Devices in configuration: number of devices in data point configuration
- Devices online: number of devices which are currently online
- Devices online incl. prim. scan: number of devices in configuration which are online
- Devices offline: number of devices which currently appear offline
- Current state: state of the M-Bus stack (IDLE, SEND_REW, SEND_NOK, SEND_E5, RCV_REQ_UD2, WAIT_DONE)
- Channel load: channel load of the M-Bus channel averaged over 5 minutes
- Maximum channel load: maximum channel load since the last statistics clear
- Rx bytes: number of bytes received
- Tx bytes: number of bytes sent
- Tx total: total number of transmissions
- Tx errors: number of errors during a transmission
- Device errors: number of timed-out transmissions
- Tx no response: number of transmissions without a response
- Tx repetitions: number repeated messages
- Tx successful: number of successful transmissions
- Tx E5 successful: number of successfully received E5 responses

- Tx short frame successful: number of successfully received short frames
- Tx long frame successful: number of successfully received long frames
- Tx multitel successful: number of successfully received multi telegram frames
- E5 unexpected: number of unexpected E5 responses (misbehaving slave)
- Short frame unexpected: number of unexpected short frame responses (misbehaving slave)
- Long frame unexpected: number of unexpected long frame responses (misbehaving slave)
- T1 errors: number of slave responses not received
- Multitel count overflow: number of messages exceeding the maximum multi telegram number
- Frame errors: number of received frames with errors
- Only header response: number of responses containing only the message header
- Wrong source address: number of responses with a wrong slave address
- Wrong frame type: number of responses with wrong frame type
- Wrong frame length: number of responses with wrong frame length

11.4.5 M-Bus Protocol Analyzer

By activating the link Protocol Analyzer (available in the M-Bus statistics tab), the protocol analyzer page is shown as displayed in Figure 251.

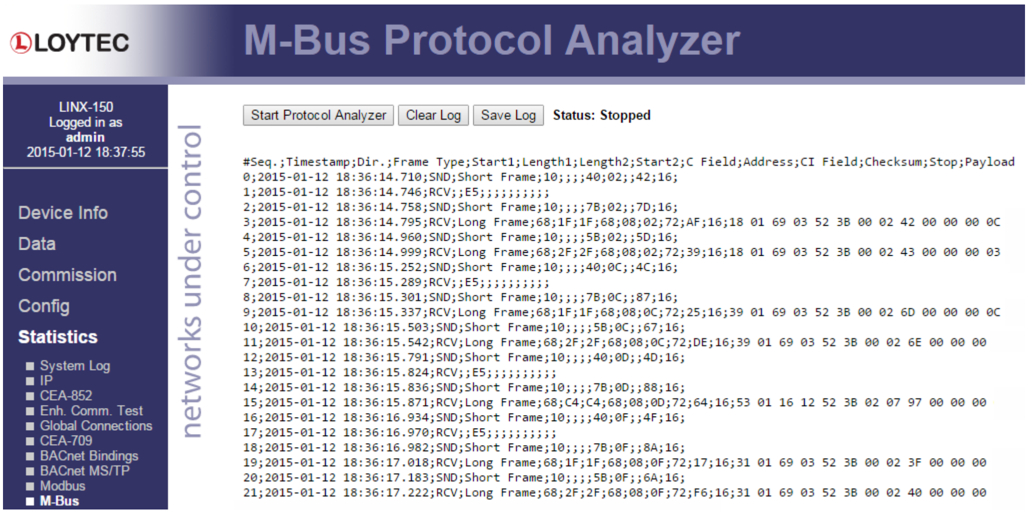


Figure 251: M-Bus protocol analyzer.

Next to the button the status of the protocol analyzer is shown. If the analyzer is started, an automatic refresh is performed every 60 seconds. By pressing the button **Start Protocol Analyzer** / **Stop Protocol Analyzer** the protocol analyzer can be started / stopped.

For every frame sent or received a line is presented using comma separated values. When stopped click on **Save Log** to store the protocol log as a CSV file. **Clear Log** clears the log data.

11.5 Configurator

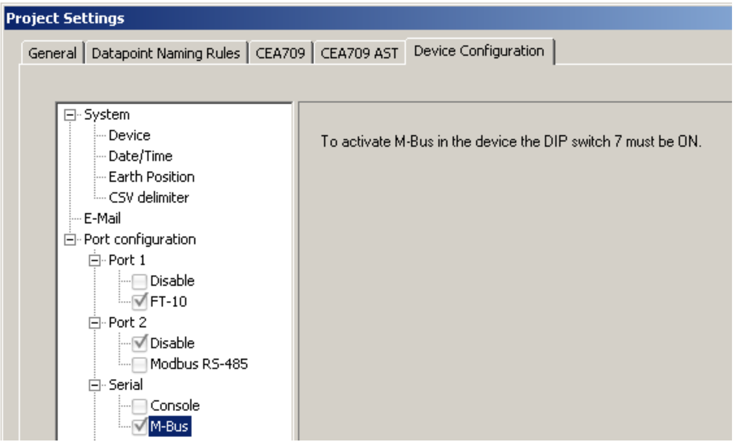
This section describes how to use the Configurator software for the management of M-Bus data points. For further information on the Configurator software refer to Chapter 7.

11.5.1 Activating M-Bus Configuration

Before a new M-Bus configuration can be managed, the M-Bus option must be enabled. The project settings are described in detail in Section 7.3.

To Activate the M-Bus Configuration

1. Open the project settings dialog.
2. In the **Device Configuration** tab enable the M-Bus check box.
3. Press the **OK** button.



Important:	***If the M-Bus Port is deactivated via the checkbox or a firmware or model version is chosen, which does not support M-Bus, the complete M-Bus configuration is deleted. In this case a dialog is displayed, which has to be confirmed.***

11.5.2 Data Point Manager for M-Bus

The Configurator uses a central concept to manage data points. The data point manager is used to select, create, edit and delete data points. The dialog is divided into three sections:

- The folder list (Figure 252)
- The data point list (Figure 253),
- And a property view.

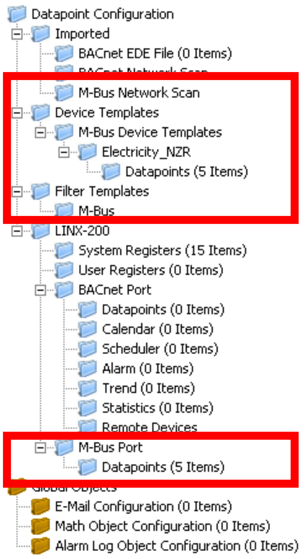
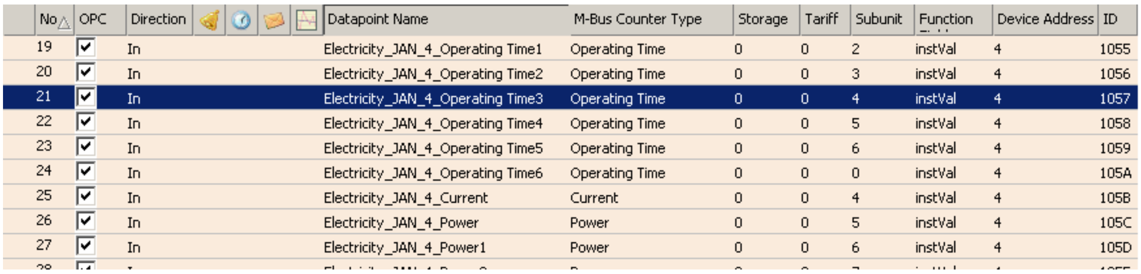


Figure 252: Data Point Manager Dialog with M-Bus folder list



No	OPC	Direction	Datapoint Name	M-Bus Counter Type	Storage	Tariff	Subunit	Function	Device Address	ID
19	☑	In	Electricity_JAN_4_Operating Time1	Operating Time	0	0	2	instVal	4	1055
20	☑	In	Electricity_JAN_4_Operating Time2	Operating Time	0	0	3	instVal	4	1056
21	☑	In	Electricity_JAN_4_Operating Time3	Operating Time	0	0	4	instVal	4	1057
22	☑	In	Electricity_JAN_4_Operating Time4	Operating Time	0	0	5	instVal	4	1058
23	☑	In	Electricity_JAN_4_Operating Time5	Operating Time	0	0	6	instVal	4	1059
24	☑	In	Electricity_JAN_4_Operating Time6	Operating Time	0	0	0	instVal	4	105A
25	☑	In	Electricity_JAN_4_Current	Current	0	0	4	instVal	4	105B
26	☑	In	Electricity_JAN_4_Power	Power	0	0	5	instVal	4	105C
27	☑	In	Electricity_JAN_4_Power1	Power	0	0	6	instVal	4	105D

Figure 253: Datapoint Manager Dialog with M-Bus Data Point List.

11.5.3 Folder List

At the left is a list of folders which is used to sort the available data objects by their category. There are a number of predefined M-Bus folders available. All other folders are described in section 7.2.1:

- **Imported**: This folder has a number of sub-folders for different import methods:
  - **M-Bus Network Scan**: This folder holds data points scanned online from an attached M-Bus network. When scanning an M-Bus device, a subfolder is created under M-Bus Network Scan. The name of this subfolder is generated automatically from the information of the scanned device. Additionally under the device sub-folder a data point folder is created.
- **Device Templates**: This folder contains created data point templates for the different technologies.
  - **M-Bus Device Templates:** This folder contains a sub-folder for each device, which is imported from an M-Bus device template. This device folder also contains a sub-folder with the data points specified in the template. Data points can be added to the folder. Additionally suitable data objects can be created for the use on the device by selecting the **Use on Device** option.
- **Filter templates**: This folder contains filter templates for scanned M-Bus devices
  - **M-Bus**: This folder contains a folder with data points for each created filter template.
- **LINX-XXX**: This is the device folder (see Section 7.2.1). For M-Bus an additional port folder exists:

- **M-Bus Port**: This folder contains the remote M-Bus data points, which are used on the device.

11.5.4 Network Port Folders

The M-Bus network port folder on the device has the same structure of sub-folders as the other network port folders in Section 7.2.2. Currently only the **Datapoints** folder exists for the M-Bus network port.

11.5.5 M-Bus Properties

Apart from the common data point properties discussed in Section 7.2.4 the data points of the M-Bus technology have additional properties:

- **Storage Number**: This property defines the M-Bus storage number of the data point. This number can also be specified by the manufacturer using a DIF/DIFE combination.
- **Tariff**: This property defines the tariff of the M-Bus data point. This number can also be specified by the manufacturer using a DIF/DIFE combination.
- **Subunit**: This property defines the subunit of the M-Bus data point. This number can also be specified by the manufacturer using a DIF/DIFE combination.
- **Function Field**: This property defines the function field of the M-Bus data point. Possible values for this property are instantaneous value, maximum value, minimum value and value during error state. This number can also be specified by the manufacturer using a DIF/DIFE combination.
- **Data Coding**: This property defines how the value for this data point is coded. The information is not relevant for input data points but it is mandatory for output data points. Possible values for this property are instantaneous value, maximum value, minimum value and value during error state. This number can also be specified by the manufacturer using a DIF/DIFE combination.
- **VIF/VIFE**: The VIF and VIFE (Value Information Field and VIF Extension) specify the counter type and its scaling. When the most significant bit of a VIF or VIFE is set, another VIFE follows. Up to ten VIFEs can be specified. When this combination is entered, also the M-Bus counter type and the unit are updated according to the VIF/VIFE combination.
- **M-Bus Counter type:** This information is derived from VIF/VIFE. It informs about the type of the data point value (e.g. Energy counter value or operating time).
- **M-Bus Device Name**: This property shows the name of the M-Bus device the remote data point is connected to.
- **M-Bus Device Address**: This property shows the address name of the M-Bus device the remote data point is connected to.
- **M-Bus Pollgroup**: Each M-Bus input data point is attached to a poll group. If more than one poll group is available, the poll group can be selected. This property is not shown for output data points.
- **M-Bus Poll Mode**: Usually M-Bus data points are read via a default read or a selective read (REQ_UD2 telegram specifying the appropriate data point). When this value is set to **Defined by device setting**, the method which is specified in the device settings is used. Using **Default Read** or **Selective Read** overrides the device setting for this data point. It is advised to leave this option at the default setting and adapt it only, if a device requires a different configuration for some specific data points.

11.5.6 M-Bus Device Capabilities

Communication to an M-Bus device can be configured with a number of options to match the target device’s capabilities. Some of them are important to treat special behavior of certain M-Bus devices on the market. In principle, the device capabilities specify, what kind of M-Bus read requests the device is able to process and which other preconditions apply.

- **Default Read**: If nothing else is specified, the default read method is used. Default read usually means that the M-Bus device transfers all available data during a read operation.
- **Selective Read**: Selective read means that the particular data point to be read can be selected during the read request. If the device is able to perform such a request, the selective read method should be chosen as it saves bandwidth.
- **Send NKE before default read**: If this option is enabled, an NKE message is sent to the device before sending the default read request. This reinitializes the M-Bus device. Devices which send multi-telegram messages start with their first telegram in the next read request; it is advised to set this option for these devices. Please note that some devices perform a complete device reset when they receive an NKE message. In this case the read request may fail. Please refer to the manufacturer's manual of the M-Bus device for more information.
- **Send application reset before default read**: If this option is enabled, an application reset is sent to the device before sending the default read. Please refer to the manufacturer's manual of the M-Bus device for detailed information.
- **Ignore multi-telegram message**: This flag only has an impact on reading M-Bus devices, which send multi-telegram messages. In this case, the transfer is aborted after the reception of the first telegram. It is advised to also set the **Send NKE before default read** when setting this flag.
- **Max. per Request**: This setting specifies the maximum number of data points to be specified within one selective read request (REQ_UD2 telegram).

11.6 M-Bus Workflow

This section discusses the workflows for setting up an M-Bus environment. The network can either be set up online using the network scan function or also offline by either setting up the devices and data points manually or by using M-Bus device templates. The change of primary addresses and Baud rates can only be done online.

11.6.1 Offline Engineering

This section describes how an M-Bus network can be set up without using the M-Bus network. Figure 254 shows the workflow. First, the M-Bus devices, address and Baud rates of the M-Bus devices must be configured for example by following the device manufacturer's guidelines (see Section 11.7.3). Afterwards the devices and data points are configured in the Configurator either configured manually (see Section 11.7.4) or by using M-Bus device templates (see Section 11.7.5). Also mixing the two methods is possible. When using the device templates also data points can be added manually. The configuration is then downloaded to the device and the device is rebooted (see Section 7.6.4).

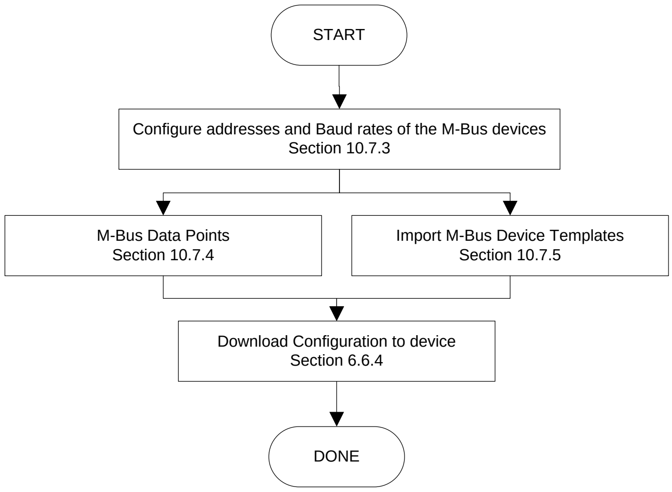


Figure 254: Workflow for offline engineering.

11.6.2 Online Engineering

This section describes how the M-Bus network is set up when the network can be accessed. Figure 255 shows the workflow. If necessary the addresses and Baud rates the M-Bus devices are using can be configured using the Configurator (see Section 11.7.3). The devices and data points can then be configured by scanning the connected devices (see Section 11.7.2), manual configuration (see Section 11.7.4) or also by using the M-Bus device templates (see Section 11.7.5). The configuration is then downloaded to the device and the device is rebooted (see Section 7.6.4).

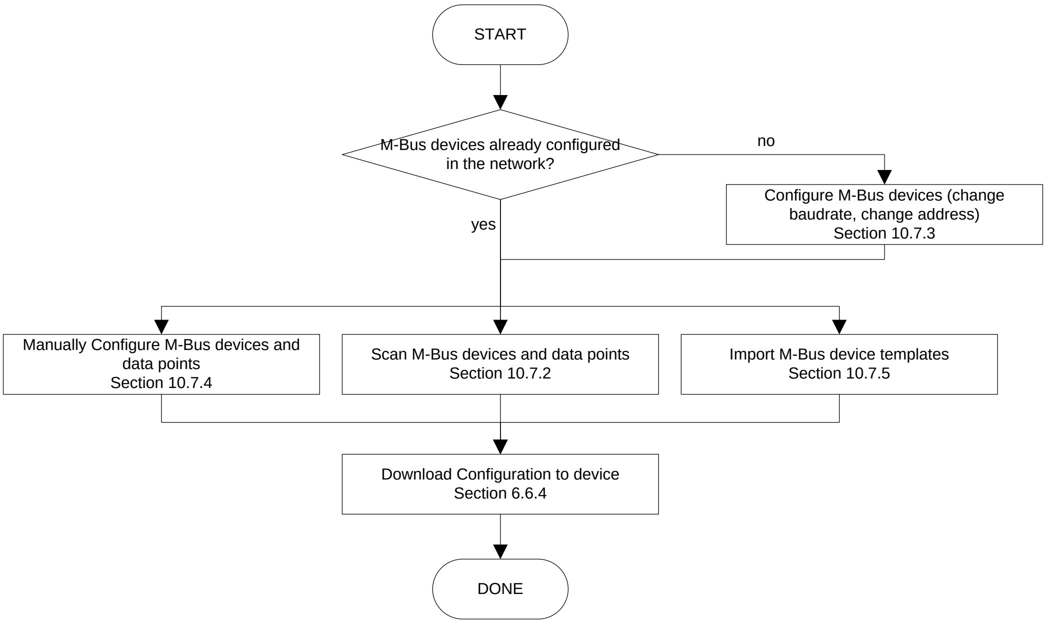


Figure 255: Workflow for online engineering.

11.7 Using the Configurator for M-Bus

11.7.1 Automatic Naming

Operations which automatically generate M-Bus data points or allocate M-Bus devices use auto naming. Automatic naming is also used, when a device is applied and no device name is specified.

The automatic device name is concatenated from the device medium (e.g. Electricity), the 3 character M-Bus manufacturer code and the address. For example the device could be automatically named “Electricity_LOY_7”. If a name is specified, the device name of the applied device is concatenated the address, e.g., “Device_7”.

The automatic data point name is concatenated from the device name the data point is related with and the type of the data point. For example if a data point is an energy counter value in the device “Electricity_LOY_7”, the name “Electricity_LOY_7_Energy” is created.

11.7.2 Scanning the M-Bus Network

The Configurator software can connect to the device and perform an M-Bus network scan. The network scan searches for connected M-Bus devices and data points on those devices. The device scan goes through each address on the M-Bus network using the specified Baud rates. When scanning for a device the M-Bus scanner starts with the highest specified Baud rate. If the device is found, it is added to the device list, if not, the scan tries to find the device address with the next lower Baud rate and so on.

The M-Bus scan can only scan for input data points. Output data points can be created manually or imported via a device template.

To Scan for Devices

1. Connect to the device via FTP as described in section 7.6.1.

2. Right click on the Folder **M-Bus Network Scan** and select **Scan M-Bus Network**. This opens the **M-Bus Network Scan** Dialog shown in Figure 256.

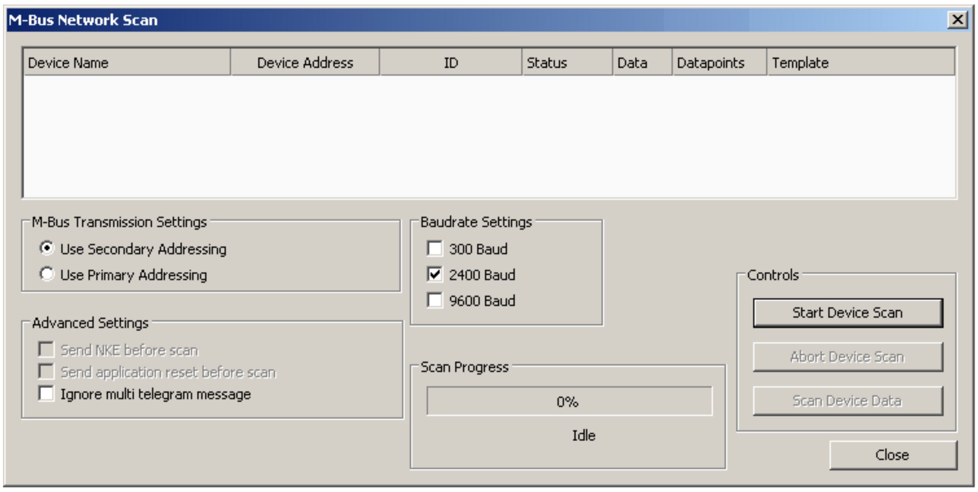
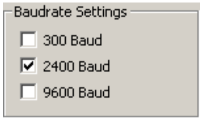


Figure 256: M-Bus Network Scan dialog.

3. In **M-Bus Transmission Settings** choose the address scanning mode. The default is secondary address scanning. This prevents problems with duplicate primary addresses of previously uninstalled M-Bus devices.

4. Select all applicable Baud rates for the device scan.



Important:	***Selecting 300 Baud results in a very slow scan. Aborting the scan is possible using the Abort Device Scan button***

5. Start the scan by pressing the **Start Device Scan** button. The progress bar shows the progress of the scan. Under the progress bar, a text displays, which device is currently scanned. When a device is found, it is displayed in the device list. The name of the device is automatically created as described in section 11.7.1.

6. The scan can be aborted by selecting the **Abort Device Scan** button.

7. When the device scan is finished (either aborted or ended), the devices can be selected for a data scan. Also multi-select is possible.

8. Select the devices which have to be scanned for data points and press the **Scan Device Data** button. This scans all data points of the selected devices. For every device a folder with the name of the device is created under the **M-Bus Network Scan** folder. The data points found are placed in the **Datapoints** folder of the appropriate devices.

To Use Datapoints from a Scan

1. Go to the **Datapoints** folder of the device of the M-Bus scan.

2. Select the desired data points, also multi-select is possible.

3. Either press on the **Use on Device** button  or right-click and select **Use on Device**. The selected data points are now available in the **Datapoints** subfolder of the **M-Bus Port** folder.

To Create Filter Templates from a Network Scan

1. Go to the **Datapoints** folder of the device of the M-Bus scan.

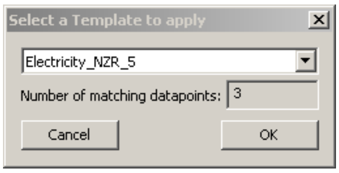
2. Select the desired data points, you wish to create a template from, also multi-select is possible.

3. Right click on one of the selected data points and select **Use as Template**. This creates a folder, containing the selected data points.

To Use data points from a Scan using Filter Templates

1. Right click on the folder M-Bus Network Scan and select either **Use on device and apply single M-Bus filter templates** or select **Use on device an apply all M-Bus filter templates**. When all filter templates are applied, all matching data points from the scan are used on the M-Bus port of the device.

2. When **Use on device and apply single M-Bus filter templates** is selected, the following dialog is opened:



3. The drop down box shows all available M-Bus filter templates. As additional information, when a filter template is chosen, the number of data points is displayed, which match the template.

4. Select **OK** to use the data points on the device.

11.7.3 Network Management Functions

This section describes how the M-Bus network management functions can be used. It describes adding and removing M-Bus devices as well as changing the Baud rate or the primary address. The tasks can be performed offline as well as online.

The **Network Management** dialog shown in Figure 257 shows a list of devices. Devices which have been scanned using a network scan have the status online, devices which have been created manually or using device templates have the status offline. If a device which is online also has been scanned for device data, a green checkmark is displayed in the **Data** column.

To Start the M-Bus Network Management Dialog

1. Connect to the device via FTP as described in section 7.6.1.

2. Select the M-Bus dialog by clicking on the **M-Bus** button

in the tool bar of the **Datapoints** tab. The M-Bus Management dialog opens, showing the **Network Management** tab displayed in Figure 257.

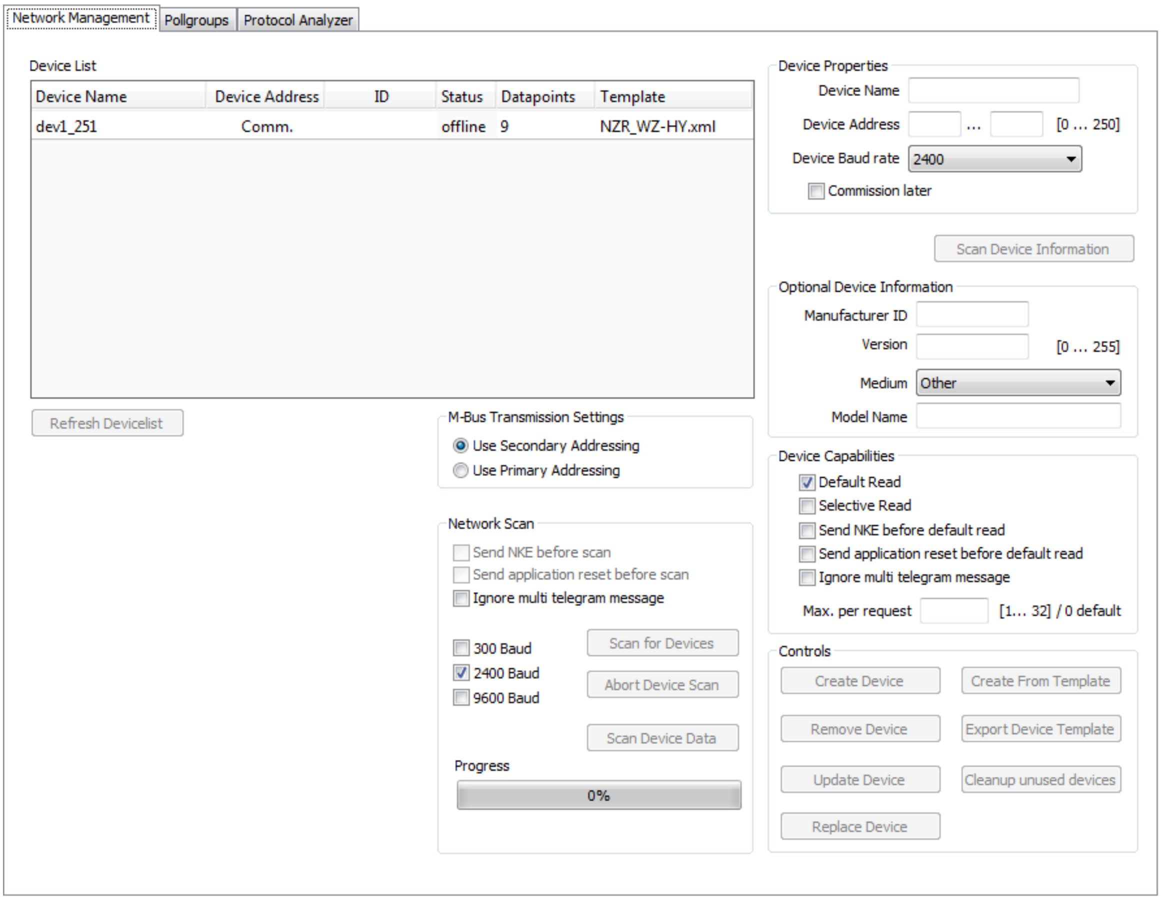


Figure 257: M-Bus network management dialog.

To Add an M-Bus Device Manually Without Scanning

1. Fill in the **Device Address** and select a **Device Baudrate** from the drop down box.

2. A **Device Name** can be specified. If more devices with the same properties have to be created using subsequent addresses, the end address can be specified in the input field on the right hand side of the **Device Address** field.

3. If the device shall be created without knowing the address of any physical device, leave the above address fields empty and select **Commission later**. This device can then be commissioned on the Web UI (see Section 11.4.3).

4. The device capabilities specify, what kind of M-Bus read requests the device is able to process. If nothing is specified here, **Default Read** is used. For other options please refer to Section 11.5.6.

5. The optional device information just represents manufacturer details and model details. This information is used in two cases:

   a. The device name is not specified – in this case a device name is automatically created from the optional device information. The name is concatenated to MAN_Medium_address, where MAN is the 3 character manufacturer code, Medium is the M-Bus medium (e.g. “Heat”) and address is the specified address.

b. Device templates can be created from the M-Bus devices. The device templates store the device information in order to identify a device.

6. Click on the **Create Device** button. This creates the M-Bus device and adds it to the device list on the left hand side of the dialog.

7. When a device is selected the device information is displayed in the appropriate fields on the right hand side of the dialog. The information can be changed in the fields. Press the **Update Device** button to store the changes.

8. If a device has to be deleted select the device and press the **Remove Device** button.

To Add an M-Bus Device Manually Using the Scan Device Information

1. Enter the address of the device which has to be scanned.

2. Press the **Scan Device Information** button. If the device is found in the network, the device properties are filled.

3. Enter a name for the M-Bus device in the **Device Name** field. If no name is specified the name is created automatically as described in section 11.7.1 from the M-Bus data the device sends back.

4. Enter a device address in the **Device Address** field. If more devices with the same properties have to be created using subsequent addresses, the end address can be specified in the input field on the right hand side of the Device Address field.

5. If the device shall be created without knowing the address of any physical device, leave the above address fields empty and select **Commission later**. This device can then be commissioned on the Web UI (see Section 11.4.3).

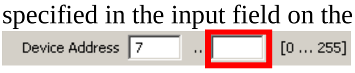
6. Select a device Baud rate from the combo box.

7. Press the **Create Device** Button to add the number of devices.

To Add an M-Bus Device Manually Using a Template

1. Enter a name for the M-Bus device in the **Device Name** field. If no name is specified the name is created automatically as described in section 11.7.1 from the M-Bus data the device sends back.

2. Enter a device address in the **Device Address** field. If more devices with the same properties have to be created using subsequent addresses the end address can be specified in the input field on the right hand side of the **Device Address** field.



3. Click on the **Create From Template** button. This opens the **Import M-Bus Device Template** dialog shown in Figure 258.

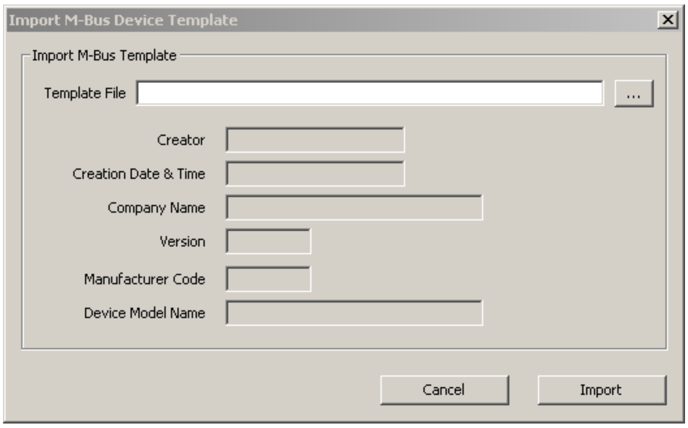


Figure 258: Import M-Bus Device dialog.

4. Press the ... button and select a template file from the **Open** dialog.

5. After selecting the file, the device information is displayed.

6. Press **Import** for importing the template or **Cancel** for closing the dialog without any changes.

7. When a template is imported, a folder with the name of the device is created. Under this folder a **Datapoints** folder containing the data points from the template file is created.

Tip: ***Data points can be added to the data points of the template by right-clicking in the data point list and selecting New Datapoint.***

When you want to create one or more device instance together with its data points, use the button Create with DP instead of the button Create From Template. This creates the device instance, all the data points from the template can be found in the M-Bus Port's data points folder.

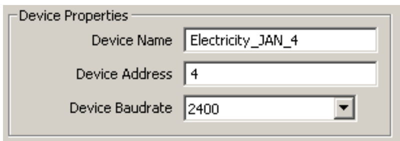
To Remove an M-Bus Device

1. Select the device which has to be removed, also multi-select is possible.

2. Press the **Remove Device** button. If the device already has data points, these data points have to be deleted before the remove can be performed.

To Change the Properties of an M-Bus Device

1. Select the device. This shows the device properties.



2. Update the field **Device Address** or select another Baud rate in the combo box.

3. Press the button **Update Device**.

4. If a device address is specified, which already exists, a failure message is displayed.

To Scan Devices Using the M-Bus Network Management Dialog

1. Select the Baud rate for the device scan. For more information on scanning the M-Bus network refer to Section 11.7.2.

2. Start the scan by pressing the **Scan for Devices button**.

3. When the device scan is finished, the devices can be selected for a data scan. Also multi-select is possible.

4. Select the devices which have to be scanned for data points and press the **Scan Device Data** button. This scans all data points of the selected devices. For every device a folder with the name of the device is created under the **M-Bus Network Scan** folder. The data points found are placed in the **Datapoints** folder of the appropriate devices.

Important: ***If a device, which is scanned, is already in the device list, the existing device can either be overwritten; deleting all previously scanned data points of the existing device or the scanned device can be discarded. A dialog is displayed for this decision.***

11.7.4 Manual Configuration of Data Points

It is possible to manually configure M-Bus data points. Manual configuration is done by specifying all information, the M-Bus device manufacturer provides.

To Manually Create an M-Bus Data Point

1. Click on the M-Bus port **Datapoints** folder.

2. Right click in the data point list view and select **New Datapoint** in the context menu.

3. This opens the **Create New M-Bus Datapoint** dialog shown in Figure 259.

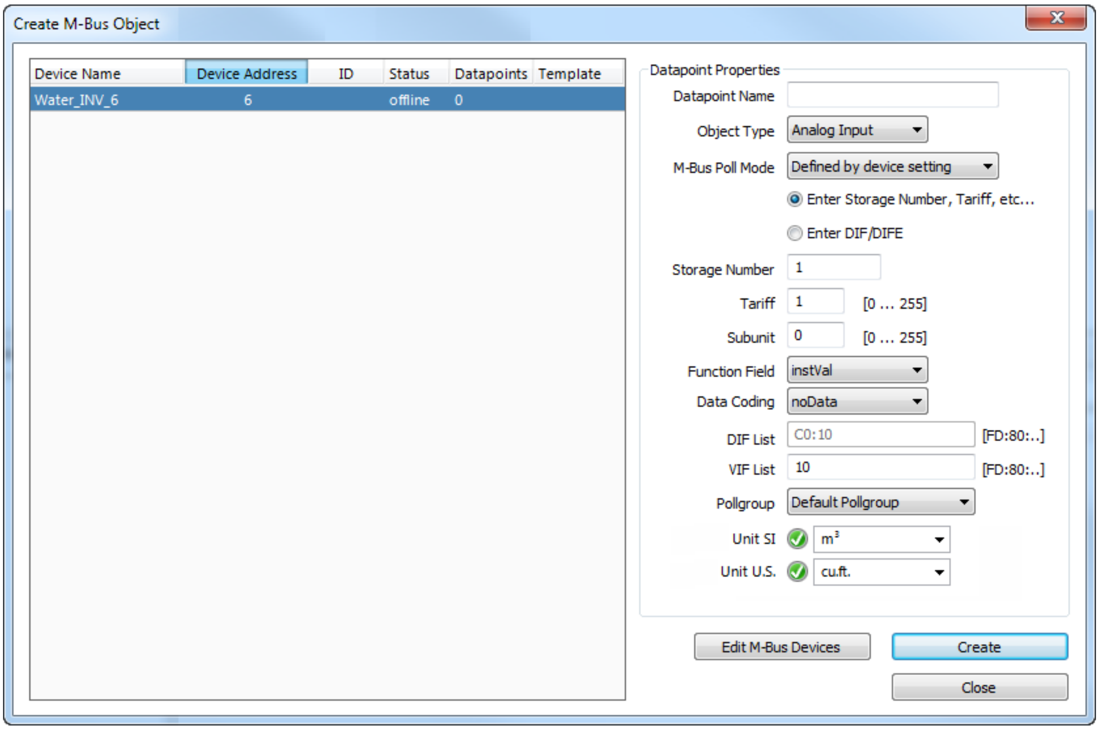


Figure 259: Create M-Bus Object dialog.

4. If the M-Bus device which provides the data point is not in the list, it has to be created. In this case open the Network Management dialog by clicking the Edit M-Bus Devices button.

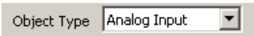
5. Create the device in the **Network Management** dialog and close the dialog.

6. Select the device which provides the M-Bus data point.

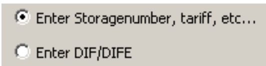
7. The data point properties are entered in the group **Datapoint Properties** as shown in Figure 259.

8. Enter a data point name. If no name is entered, the data point is named as described in section 11.7.1.

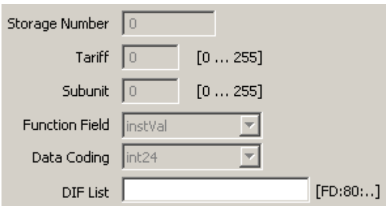
9. At the moment only analog M-Bus data points are supported. Select if the data point is an analog input or output. For analog inputs no M-Bus data coding can be specified.



10. Select if the data point shall be specified by providing the numbers for Storage number, tariff, subunit, function field and data coding or if the information configured using the DIF/DIFE list.



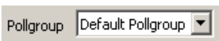
For reference, if one piece of information is entered, the other one is derived from the specified data. Enter the data point information.



If the DIF list is entered, the dialog expects hexadecimal numbers. As soon as the information is entered the other fields are updated.

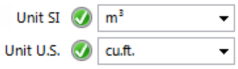
11. Enter the VIF/VIFE list. This list specifies the M-Bus counter type, and unit of the data point. Also this field has to be specified using hexadecimal numbers.

12. Usually data points are added to a default poll group. If the data point has to be member of another poll group than default, the poll group can be selected using the drop down box.



In the drop down box the previously specified poll groups are shown. If no poll group is configured only the default poll group is displayed. Refer to section 11.7.7 for more information on poll groups.

13. The FIV list defines the network unit of the M-Bus data point. In addition you may choose its representation in the selected unit system.



14. Press the **Create** button to create the M-Bus data point

Tip: ***After creating a data point, the poll group can be changed in the data points property view. Also multi-select can be used.***

11.7.5 Importing via Device Templates

For some M-Bus devices special templates are available which specify all available data points of an M-Bus device as well as the device properties. Such templates can be imported into the configuration.

To import an M-Bus Device Template

1. Right click on the Folder **M-Bus Device Templates** and select **Import device template** from the context menu.

2. This opens the **Import M-Bus Device Template** dialog shown in Figure 260.

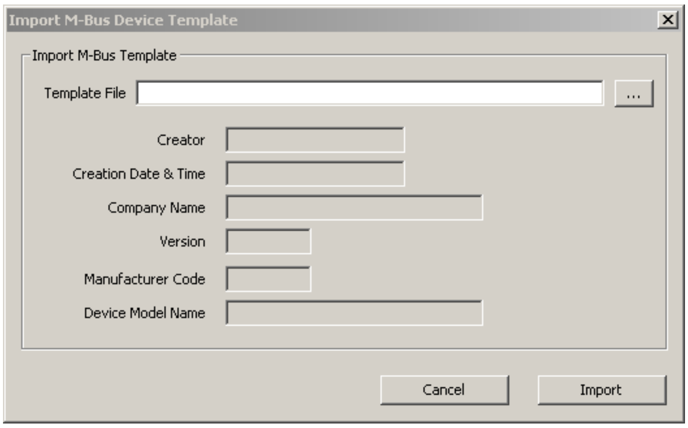


Figure 260: Import M-Bus Device Template dialog.

3. Press the  button and select a template file from the **Open** dialog.

4. After selecting the file, the device information is displayed.

5. Press **Import** for importing the template or **Cancel** for closing the dialog without any changes.

6. When a template is imported, a folder with the name of the device is created. Under this folder a **Datapoints** folder containing the data points from the template file is created.

Tip: *Data points can be added to the data points of the template by right clicking in the data point list and selecting New Datapoint.*

Importing a device template from the folder list does not create a device instance. Device instances can only be created using the import in the Network Management Dialog.

To Use Imported Data Points

Using imported data points is a little different to the use of scanned data points. For a scanned device a device instance already exists – the important information address and Baud rate – devices imported from templates do not have an address or Baud rate.

1. Go to the **Datapoints** folder of the device of the M-Bus Templates.

2. Select the desired data points, also multi-select is possible.

3. Either press on the **Use on Device** button or right-click and select **Use on Device**. This opens the **M-Bus Configure Device For Use** dialog shown in Figure 261.

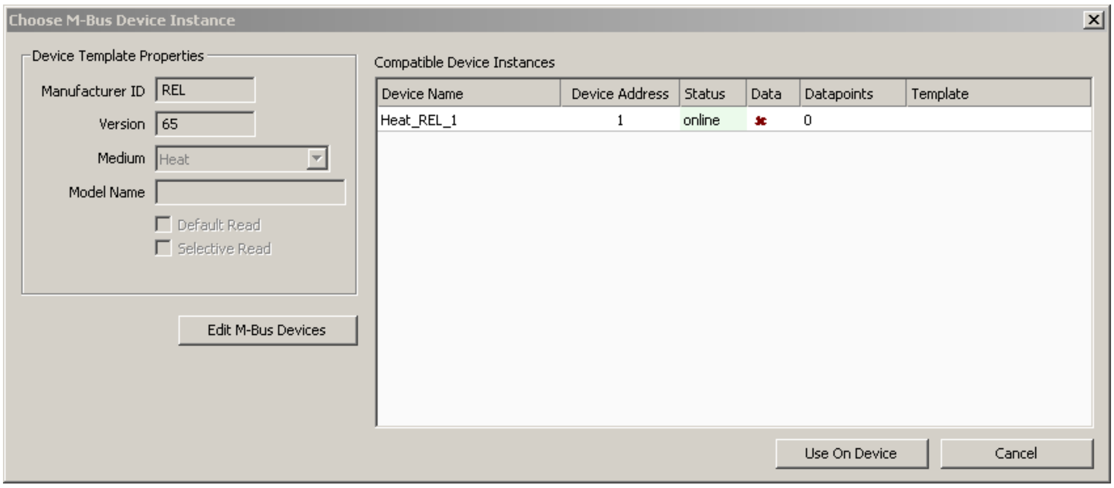


Figure 261: Use on Device dialog.

4. The device list displays all devices which have the same Manufacturer ID, version number and Medium. If no device instance matches the device template, create one by entering the network management dialog. The dialog can be entered by pressing the **Edit M-Bus Devices** button. In the management dialog, the device instance can be either created manually (take care of entering the correct Manufacturer ID, Version and Medium) or by simply importing the template again.

5. Select one or more device instances from the list and press the **Use On Device** button. This creates for each selected data point and each selected device one data point in the M-Bus Port's data point list.

11.7.6 Creating Device Templates

M-Bus device templates can be created from a data point configuration. In fact, it is only possible to create a device template using an existing device or an existing device template with data points. This device and its data points can either be configured manually, by a scan or also imported from a device template itself.

An M-Bus device template contains the following configuration items describing an M-Bus device:

- M-Bus device settings,
- M-Bus data points,
- Folders used to organize M-Bus data points on a device,

- Structure types, multi-state maps, historic filters used by M-Bus data points,
- Pollgroup definitions used by M-Bus data points.

To Create an M-Bus Device Template Using Devices

1. Select the M-Bus dialog by clicking on the **M-Bus** button

in the tool bar of the **Datapoints** tab. This opens the **Network Management** dialog as displayed in Figure 262.

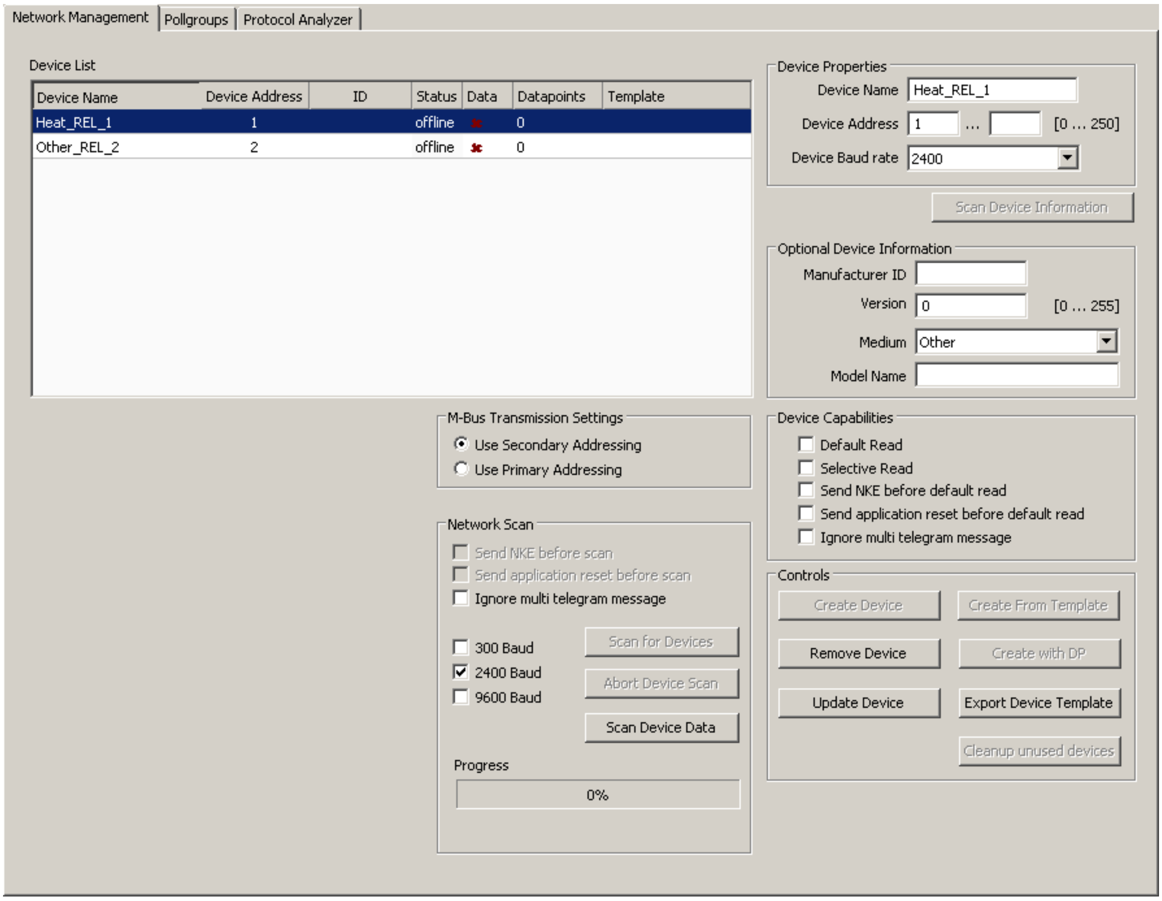
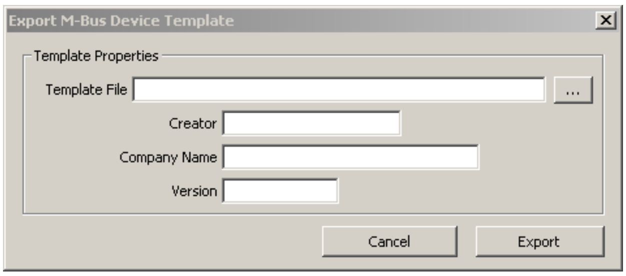


Figure 262: M-Bus Network Management dialog.

2. The device list shows all devices of the current configuration. Select a device.

3. Press the **Export Device Template** button. This opens the **M-Bus Export Template** dialog.



4. Press the ... button and select a template file from the **Save** dialog.

5. Enter the **Creator**, **Company Name** and **Version** for the template. This information is stored in the template file, when importing the template file the information is displayed after selecting the file.

6. Press the **Export** button.

To Create an M-Bus Device Template File Using a Device Template

1. Right-click on the folder of the device template that has to be exported or its data point folder and select **Export Device Template…** from the context menu.

2. Proceed as described above to export the template in the **Export M-Bus Device Template** dialog.

11.7.7 Poll Groups

In an M-Bus network, the master has to poll the slave devices. Input data points are therefore attached to a poll group. If nothing else is specified, the default poll group is used for input data points. The default poll group has a poll cycle of 60 seconds.

Three different types of poll groups can be specified:

- Time-based: The poll group is triggered on a time base. This means that after a specific time – the poll cycle – the poll group is processed.

- Trigger-based: The poll group is triggered on a special trigger data point. As soon as the trigger condition is met, the poll group is processed.

- Trigger-based with synchronization: This type is similar to the trigger-based. The difference is that when the trigger condition is met, a broadcast synchronization message is sent over the M-Bus. This causes the devices which are able to perform a sync operation to store special data points for later reading. After the broadcast is sent, the poll group is processed.

Tip: *If an M-Bus device is only able to process default read requests, it is advisable to attach all data points of this device to the same poll group (this increases the performance).*

The poll group a data point is attached to can be changed in the properties view of the data points. The poll group can also be changed for multiple data points using multi-select.

To Create a Time-Based M-Bus Poll Group

1. Select the M-Bus dialog by clicking on the **M-Bus** button

in the tool bar of the **Datapoints** tab. The **M-Bus Management** dialog opens.

2. Open the **Pollgroups** tab. This shows the dialog displayed in Figure 263.

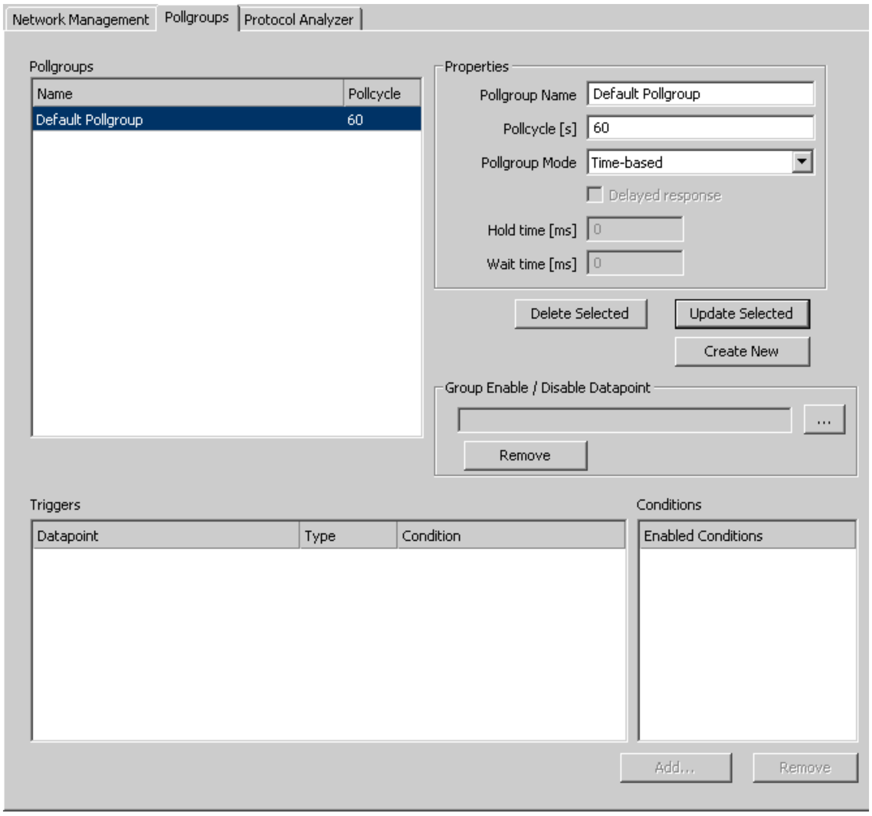


Figure 263: Pollgroup Management dialog.

3. The default poll group is selected and its properties are displayed. Enter the name of the new poll group and enter the poll cycle in seconds. Make sure that under **Pollgroup Mode** Time-based is selected.

4. Press the **Save** button to store the poll group and continue editing.

5. If a poll group needs to be updated or deleted, select the poll group edit the data and press the **Update Selected** or **Delete Selected** button.

6. Press the **Close** button to finish editing. When the poll groups have not been saved, a dialog asks whether the changes have to be saved or not.

To Create a Trigger-Based M-Bus Poll Group

1. Select the M-Bus dialog by clicking on the **M-Bus** button

in the tool bar of the **Datapoints** tab.

2. In the **M-Bus Management** dialog, open the **Pollgroups** tab. This displays the poll groups management tab displayed in Figure 263.

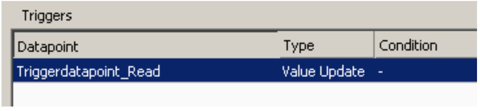
3. Enter a new poll group name and select Trigger-based or Trigger-based with synchronization.

4. Create the poll group by pressing the **Create New** button.

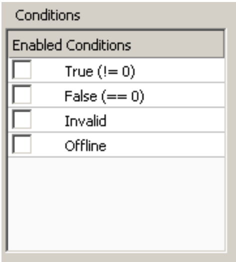
5. Select the new poll group. This enables the **Add…** and **Remove** buttons from the triggers.

6. Press the **Add…** button and select a trigger data point. This can for example be a binary user register.

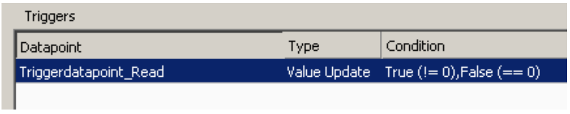
7. The selected trigger data point appears in the trigger list as shown:



8. Select the trigger from the list and check the desired trigger conditions.



9. The trigger conditions are then displayed in the trigger list.



10. Press the **Close** button to leave the dialog.

11.7.8 Trending Synchronized Meter Data

The use of trigger-based poll groups with synchronization allows trending synchronized meter data. A trigger data point triggers a synchronization message over the M-Bus network. The M-Bus meters which are able to perform the synchronization action store the meter values. These values are read out afterwards.

To Trend Synchronized Meter Data

1. Create a binary trigger data point.

2. Create a trigger-based poll group with synchronization.

3. Create a trend log as described in Section 7.14. Set the **Trend Mode** to **Change of Value (COV)** and add all required data points.

11.7.9 M-Bus Protocol Analyzer

When connected to a device a protocol analyzer is available for the M-Bus port. The protocol analyzer can be found in the M-Bus Management dialog. Figure 264 shows the dialog for the M-bus protocol analyzer.

The status on the right hand side of the dialog shows, if the device is connected or if the protocol analyzer is stopped or started. When connected to a device, the protocol analyzer can be started by pressing the **Start Protocol** button. This starts the protocol analyzer in the device. Every time a transmission is made on the M-Bus port, the transmission is displayed in the list. Additionally the protocol data is stored in the device in a rotating log file. The protocol log can hold up to 40 kB of protocol data. So also when the Configurator was not running for an interesting time, the protocol data can be loaded from the device using the

Load From Device button. The protocol data can be stored as CSV file using the **Save** button, with the **Clear** button, the shown protocol is deleted.

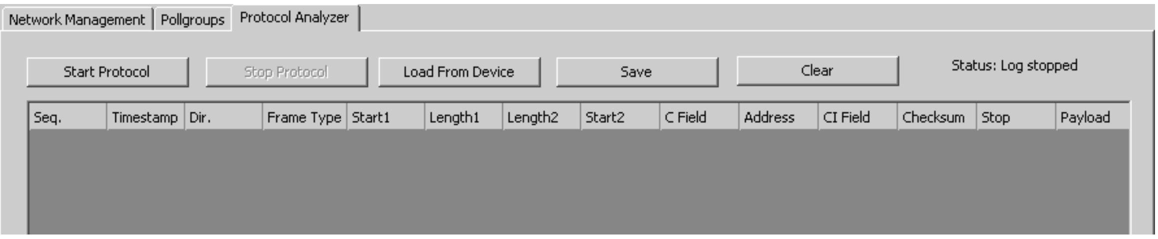
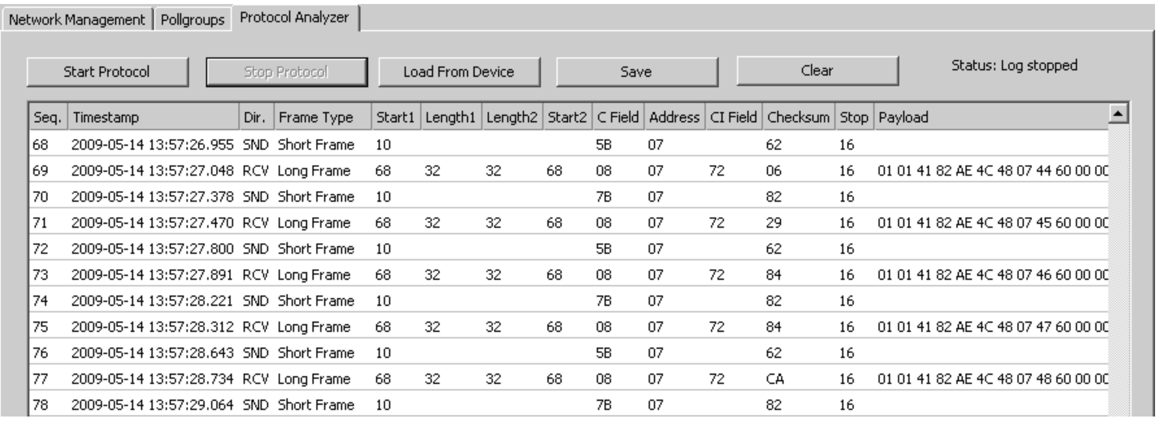


Figure 264: M-Bus protocol analyzer.

Figure 265 shows a typical protocol analyzer output for the M-Bus port. It shows the following information:

- **Seq.**: sequence number, which is automatically created in the device. This number is unique for one port.
- **Timestamp:** transmission time.
- **Frame Type:** M-Bus frame type (Short Frame, Control Frame Long Frame, E5 – in this case, no other data follows).
- **Dir.**: direction. Either SND (send) or RCV (receive).
- **Start1, Start2:** Start byte (must be equal).
- **Length1, Length2:** Frame length according to the M-Bus standard (must be equal).
- **C Field:** Control field.
- **Address:** M-Bus address.
- **CI Field:** Control information field.
- **Checksum:** Checksum of the frame.
- **Stop:** Stop byte.
- **Payload:** Payload in hexadecimal numbers (this column cannot be used for sorting).

Some frame types do not contain the full set of fields. Please refer to the M-Bus standard for additional information.



Seq.	Timestamp	Dir.	Frame Type	Start1	Length1	Length2	Start2	C Field	Address	CI Field	Checksum	Stop	Payload
68	2009-05-14 13:57:26.955	SND	Short Frame	10				5B	07		62	16	
69	2009-05-14 13:57:27.048	RCV	Long Frame	68	32	32	68	08	07	72	06	16	01 01 41 82 AE 4C 48 07 44 60 00 0C
70	2009-05-14 13:57:27.378	SND	Short Frame	10				7B	07		82	16	
71	2009-05-14 13:57:27.470	RCV	Long Frame	68	32	32	68	08	07	72	29	16	01 01 41 82 AE 4C 48 07 45 60 00 0C
72	2009-05-14 13:57:27.800	SND	Short Frame	10				5B	07		62	16	
73	2009-05-14 13:57:27.891	RCV	Long Frame	68	32	32	68	08	07	72	84	16	01 01 41 82 AE 4C 48 07 46 60 00 0C
74	2009-05-14 13:57:28.221	SND	Short Frame	10				7B	07		82	16	
75	2009-05-14 13:57:28.312	RCV	Long Frame	68	32	32	68	08	07	72	84	16	01 01 41 82 AE 4C 48 07 47 60 00 0C
76	2009-05-14 13:57:28.643	SND	Short Frame	10				5B	07		62	16	
77	2009-05-14 13:57:28.734	RCV	Long Frame	68	32	32	68	08	07	72	CA	16	01 01 41 82 AE 4C 48 07 48 60 00 0C
78	2009-05-14 13:57:29.064	SND	Short Frame	10				7B	07		82	16	

Figure 265: Typical protocol analyzer output for M-Bus port.

11.7.10 Device Replacement

This section describes how an M-Bus device can be replaced using the M-Bus network management functions.

To Replace an M-Bus Device on the Network

1. Connect to the device via FTP as described in Section 7.6.1.

2. Disconnect the device which has to be replaced from the M-Bus and connect the new device to the network.

3. Start the M-Bus device scan (see Section 11.7.2). The new device may produce a primary address conflict. This conflict may either be resolved automatically after the scan or manually during device replacement.

4. Select the M-Bus dialog by clicking on the **M-Bus** button

in the tool bar of the **Datapoints** tab. The M-Bus Management dialog opens, showing the **Network Management** tab displayed in Figure 257.

5. Select the device which has to be replaced and press the **Replace Device** button. This opens the dialog displayed in Figure 266.

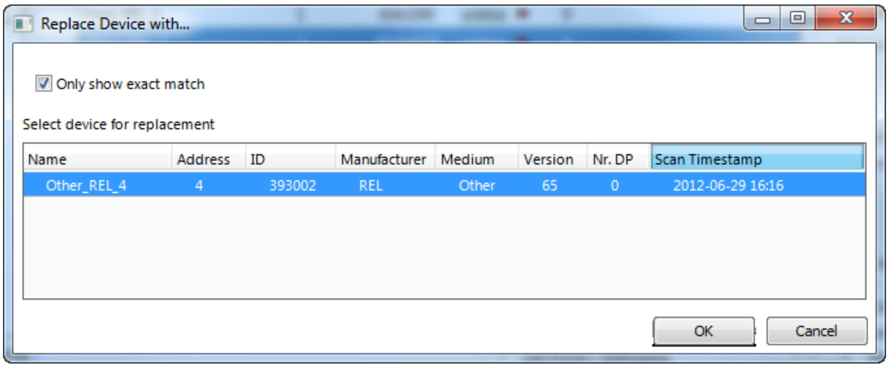


Figure 266: M-Bus Device Replacement Dialog.

6. In the replacement dialog all compatible devices for replacement are shown. If the required device is not shown in the list, remove the check **Only show exact match**.

This will show all possible devices. Note, that only M-Bus devices can be selected as a replacement device, which do are not yet using data points on the device.

7. Select the replacement device and press **OK**.

8. If the selected replacement device is still in conflict, the Configurator will suggest to automatically update the primary address to that of the replaced device.

12 Modbus

12.1 Introduction

The Modbus is a de facto industrial standard which was initially intended for the communication between PLCs. In the meantime the Modbus has become an important interface for automatic meter reading applications and industrial applications. As the Modbus is an open protocol a large number of automation devices providing Modbus communication is available.

Two communication methods are available for Modbus. Modbus TCP utilizes Ethernet for the Modbus communication. Modbus RTU uses a RS-485 bus for Modbus data transfer. Modbus RTU uses a serial bus, which is controlled by a single master. This master can request data from several slave devices connected to the network.

12.2 Modbus Network

Modbus TCP utilizes the Ethernet for the communication. Several Modbus slave devices can be connected. In Modbus TCP a unit identifier is used instead of the slave address. Modbus slaves may have the same unit identifier. This unifier is usually used to communicate with bridges, routers and gateways. According to the Modbus specification, Modbus TCP masters should always use identifier 255 for the communication with a Modbus TCP slave.

The Modbus RTU network utilizes an RS485 connection. Several Modbus slave devices are connected in parallel to the transmission medium. Each Modbus device has a unique slave address in the range from 1 to 255. The Baud rate of the RS485 can be configured to use 1200, 2400, 4800, 9600, 19200, and 38400 Baud. The parity on the LINX-10x/11x/20x/21x models is always none and stop bits are not configurable. On the LINX-12x/15x/22x models and on the LGATE-950 parity and stop bits can be configured.

Modbus devices use function codes for the specification of the desired data. The following function codes for read/write actions are supported by the LOYTEC device:

- 02: Read discrete inputs.
- 01: Read coils.
- 05: Write coil.
- 04: Read input register
- 03: Read holding registers

- 06: Write holding registers

For optimizing the communication adjacent holding registers are usually read using a single read request. This behavior can be turned off for devices which do not support reading a random number of holding registers.

Modbus data points are specified by the function code, the address and the length. Furthermore, a data type has to be specified together with the information how the order of the bytes look like. Analog Modbus Master data points have a fixed network unit, while Modbus Slave data points adapt their representation on the network according to the selected unit system.

12.3 Web Interface

This section describes the Web interface for the Modbus port.

12.3.1 Port Configuration

The Modbus ports can be configured under the port configuration tabs of the Web UI (see Section 5.2.3). If available on a given port, the Modbus protocol can be enabled. If enabled, the Modbus communication settings on that port are displayed on the right-hand side. The RS485 port configuration tab is.

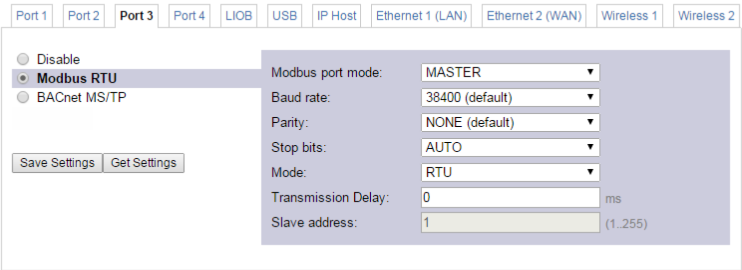


Figure 267: Modbus RS485 port configuration.

The settings for Modbus RS485 are displayed in Figure 267. For **Modbus port mode**, the following selections are available: **MASTER** can be selected, if the device shall operate as Modbus master. **SLAVE** can be selected, if the device shall operate as a Modbus slave on this port.

Under Baud rate the Baud rate for the Modbus communication can be configured. The available Baud rates are **1200, 2400, 4800, 9600, 19200, 38400 (default)**. Depending on the device model different settings for paritiy are available while stop bits are not configurable. On LINX-10x/11x/20x/21x models only **NONE** can be selected as parity, using 2 stop bits (8N2). On LINX-12x/15x/22x models parity can be configured to **NONE**, **ODD** or **EVEN**, leading to the options 8N2, 8O1, 8E1, respectively.The **Mode** specifies if the communication shall use the Modbus **RTU** mode or the Modbus **ASCII** mode.

If operated as Modbus master, an additional **Transmission delay** can be defined in milliseconds. This can improve communications on Modbus RS485 with slow devices that operate outside the timing specifications. Normally, leave this setting at ‘0’. If set, the transmission delay ensures that after sending a frame on the Modbus, the Modbus master holds back transmission of further frames for the specified time. As a Modbus slave, the user needs to define a unique **slave address** that this port will have on the Modbus channel.

Press the button **Save Settings** for storing the parameter configuration into the device or press **Get Settings** for overwriting the changes with the original configuration.

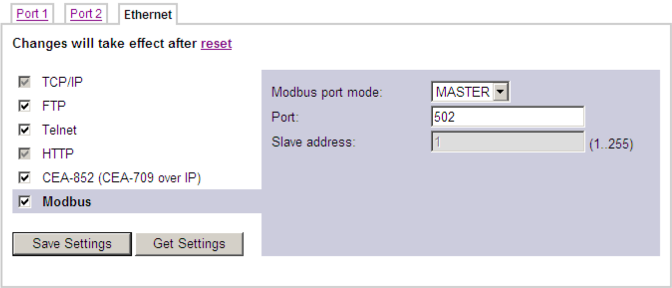


Figure 268: Modbus TCP port configuration.

On an Ethernet port, Modbus TCP can be enabled by selecting the check box. On devices with multiple IP interfaces, the Modbus TCP protocol can be activated only on one of them.

The Modbus TCP communication settings are displayed on the right-hand side as shown in Figure 268. For **Modbus port mode**, the following selections are available: **MASTER** can be selected, if the device shall operate as Modbus master. **SLAVE** can be selected, if the device shall operate as a Modbus slave on this port. Configure the desired TCP port, which is used by Modbus TCP devices on that channel. The default Modbus port number is 502.

12.3.2 Data Points

Modbus data points can be accessed through the Web UI as described in Section 5.3.1.

12.3.3 Commission

The commissioning Web UI allows assignment of physical devices to existing devices in the data point configuration, that have been created with the commission later option. Under the **Commission** menu choose the Modbus technology to open the Modbus commissioning interface. Select the appropriate Modbus interface tab.

The Web page shows a list of all **Devices in configuration**. An example is shown in Figure 269. Each line displays the device name and the Modbus device **Address**. The **Status** column shows their current status. It can be one of the following:

- OK: The device is configured for communication.
- Offline: The device is configured for communication but appears offline.
- Uncommissioned: The device is not yet commissioned.
- Disabled: The device is disabled.

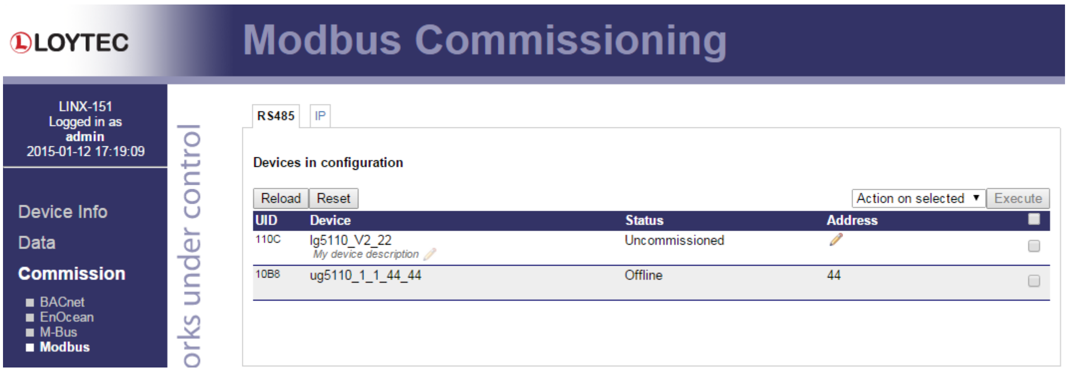


Figure 269: Modbus commissioning Web interface.

In order to execute an action on devices, select the checkbox at the end of the line. Then choose an action in the drop-down **Action on selected** and click on the **Execute** button. Actions that can be executed on all devices are enable and disable. A disabled device will stop communication on the network until it is enabled again.

Those devices created as commission later can be assigned to physical devices on the network. The device description displayed beneath the device name can be edited, where the edit symbol appears. The assignment can be done manually by editing the field in the **Address** column.

12.3.4 Statistics

Figure 270 shows a typical output of the statistics information which can be displayed for the Modbus ports. For each port available one statistics tab is displayed. The statistics can be cleared for each Modbus port separately by pressing the **Clear Modbus statistics** button. A refresh of the statistics is done automatically. To stop automatic update, deselect the **Live update** checkbox. For manual update press **Update Modbus statistics**.

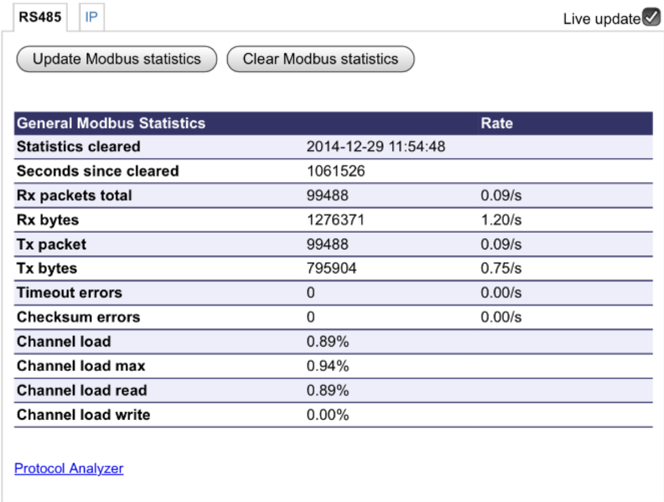


Figure 270: Statistics of the Modbus port.

The following information is available:

- Statistics cleared: last time of statistics reset
- Rx packets: number of Modbus packets received
- Rx bytes: number of bytes received
- Tx packets: number of Modbus packets sent
- Tx bytes: number of bytes sent
- Timeout errors: number of communication errors (timeout)
- Checksum errors: number of communication errors (wrong checksum)
- Channel load: Current channel load averaged over 5 minutes.
- Channel load max: Maximum channel load since last statistics clear.
- Channel load read: Current channel load due to read requests.
- Channel load write: Current channel load due to write requests.

12.3.5 Modbus Protocol Analyzer

By activating the link Protocol Analyzer (available in all Modbus statistics tabs), the protocol analyzer page is shown as displayed in Figure 271.

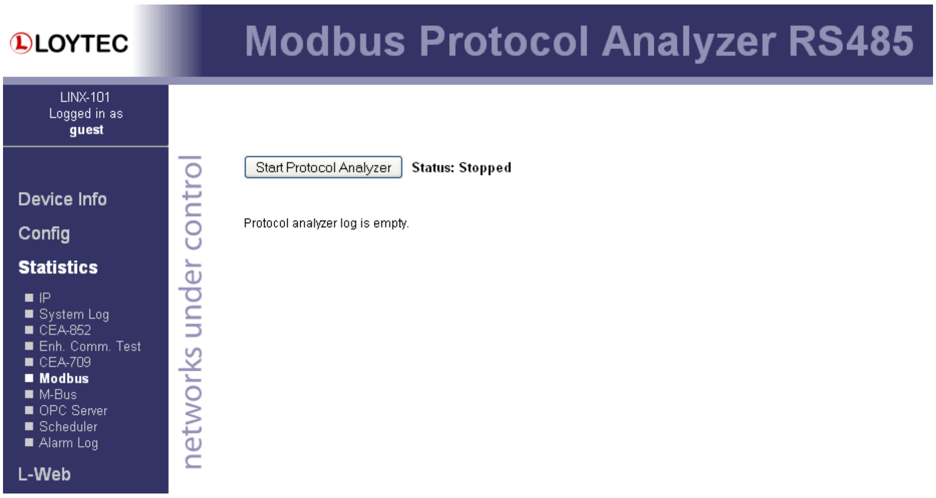


Figure 271: Modbus protocol analyzer.

Next to the button the status of the protocol analyzer is shown. If the analyzer is started, an automatic refresh is performed every 60 seconds. By pressing the button **Start Protocol Analyzer** / **Stop Protocol Analyzer** the protocol analyzer can be started / stopped.

For every frame sent or received a line is presented using comma separated values. When stopped click on **Save Log** to store the protocol log as a CSV file. **Clear Log** clears the log data.

12.4 Configurator

This section describes how to use the Configurator software for the management of Modbus data points. For further information on the Configurator software refer to Chapter 7.

12.4.1 Activating Modbus Configuration

Before a new Modbus configuration can be managed, the Modbus option must be enabled for the appropriate port. The project settings are described in detail in Section 7.3.

To Activate the Modbus Configuration

1. Open the project settings dialog.

2. In the **Device Config** tab enable the Modbus check boxes on the desired ports as shown in Figure 272. Setting the check box enables Modbus on that port. Edit the Modbus communication settings.

3. If slave mode is enabled, you may change the default Modbus register layout in the **Modbus Slave Register Configuration** box.

4. Click the **Download** button to activate the changes in the configuration.

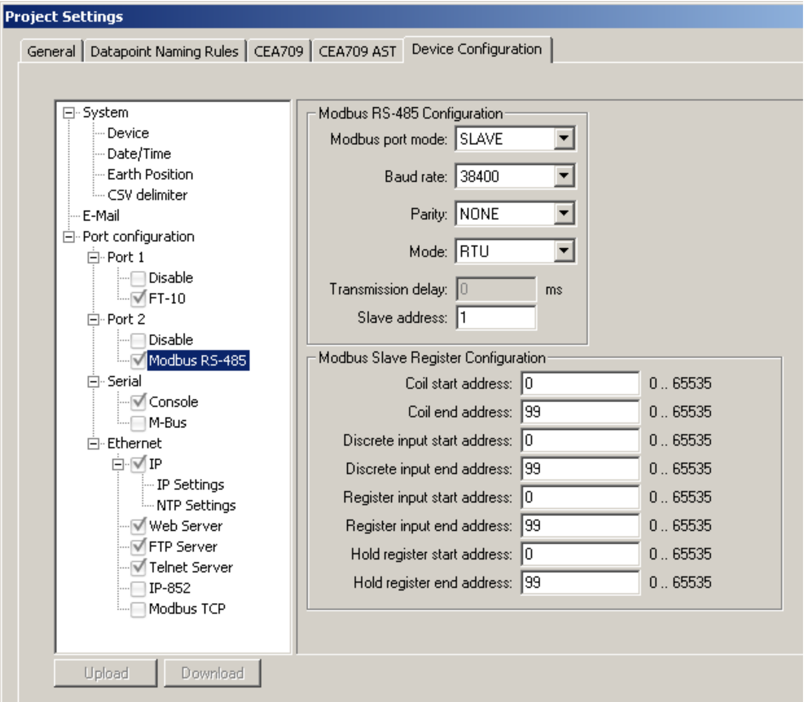


Figure 272: Project settings for Modbus.

Important: ***If the Modbus port is deactivated via the checkbox or a firmware or model version is chosen, which does not support Modbus, the entire Modbus configuration is deleted. In this case a dialog is displayed, which has to be confirmed.***

12.4.2 Data Point Manager for Modbus

The Configurator uses a central concept to manage data points. The data point manager is used to select, create, edit and delete data points. The dialog is divided into three sections:

- The folder list (Figure 273),
- The data point list (Figure 274),
- And a property view.

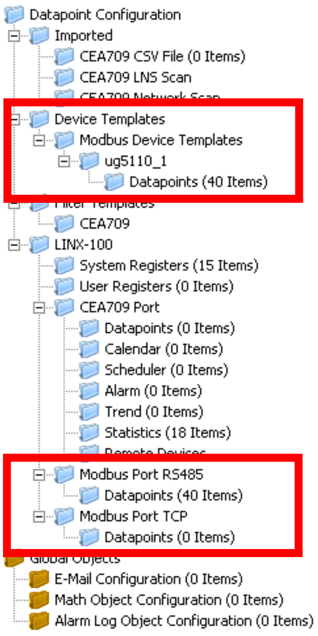
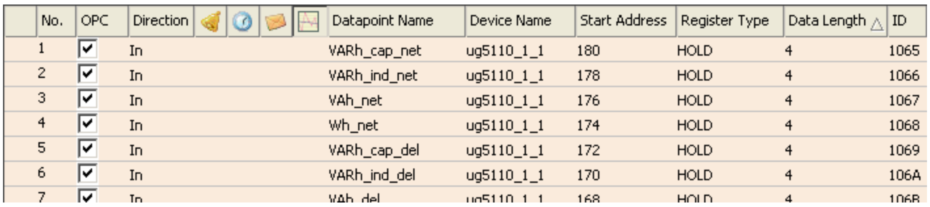


Figure 273: Data Point Manager Dialog with Modbus folder list.



No.	OPC	Direction	Datapoint Name	Device Name	Start Address	Register Type	Data Length	ID
1	☑	In	VARh_cap_net	ug5110_1_1	180	HOLD	4	1065
2	☑	In	VARh_ind_net	ug5110_1_1	178	HOLD	4	1066
3	☑	In	VAh_net	ug5110_1_1	176	HOLD	4	1067
4	☑	In	Wh_net	ug5110_1_1	174	HOLD	4	1068
5	☑	In	VARh_cap_del	ug5110_1_1	172	HOLD	4	1069
6	☑	In	VARh_ind_del	ug5110_1_1	170	HOLD	4	106A
7	☑	In	VAh_del	ug5110_1_1	168	HOLD	4	106B

Figure 274: Datapoint Manager Dialog with Modbus Data Point List.

12.4.3 Folder List

At the left is a list of folders which is used to sort the available data objects by their category. There are a number of predefined Modbus folders available. All other folders are described in section 7.2.1:

- **Device Templates**: This folder contains created data point templates for the different technologies.
  - **Modbus Device Templates:** This folder contains a sub-folder for each device, which is imported from an Modbus device template. This device folder also contains a sub-folder with the data points specified in the template. Data points can be added to the folder. Additionally suitable data objects can be created for the use on the device by selecting the **Use on Device** option.
- **LINX-XXX**: This is the device folder (see Section 7.2.1). For Modbus additional port sub-folders exist:
  - **Modbus Port RS-485**: This folder contains the remote Modbus data points of the Modbus RS-485 port, which are used on the device.
  - **Modbus Port TCP**: This folder contains the remote Modbus data points of the Modbus TCP port, which are used on the device.

12.4.4 Network Port Folders

The Modbus network port folder on the device has the same structure of sub-folders as the other network port folders in Section 7.2.2. Currently only the **Datapoints** folder exists for the Modbus network ports.

12.4.5 Modbus Properties

Apart from the common data point properties discussed in Section 7.2.4 the data points of the Modbus technology have additional properties:

- **Modbus Device Name**: This property defines the name of the Modbus slave device which contains the data point.
- **Modbus Device Address**: This property defines the address of the Modbus slave device which contains the Modbus data point.
- **Modbus IP Address**: This property is available for Modbus/TCP master only. It specifies the IP address of the Modbus slave device, which contains the Modus data point.
- **Modbus Register Start Address**: This property defines the address of the Modbus register.
- **Modbus Register Type**: This property defines the register type of the data point. Also the function code, which specifies the register, is displayed. When the Modbus register type is changed from a read to a write register, also the direction of the data point is changed.
- **Modbus Data Type**: This property defines the representation of data in the slave. This is the data type the Modbus slave uses for the data point internally. This can for example be float, double, int16 or uint32.
- **Modbus Scaling (multiplier, exponent and offset)**: These properties define the scaling parameters for the data point. The value of the data point in the device is calculated as follows:

  $$\text{Value} = (\text{ModbusValue} + \text{Offset}) \cdot \text{Multiplier} \cdot 10^{\text{Exponent}}.$$

- **Modbus Swap 16 bit, Swap 32 bit and Swap 64 bit:** This information specifies, if the order of received Modbus data has to be changed. When Swap 16 bit is set, the two bytes of a 16 bit word are swapped, if Swap 32 is set, the two words of 32 bit are swapped, and if Swap 64 bit is set, the two 32 bit words of 64 bit long data are swapped. Also combinations are possible. This configuration is necessary because the Modbus slaves can store information in any byte order (the Modbus protocol only specifies, how 16 bit data is transferred).
- **Pollgroup**: This property is only available for input data points. It shows the poll group, the data point is connected to.

12.4.6 Modbus Workflow

This section discusses the workflows for setting up a Modbus environment. Modbus does not provide a scan function and therefore the network has to be setup mostly offline. This can be eased by the use of templates. If no templates are available, the data points have to be set up manually. If devices are online, an online test feature can help in identifying Modbus registers and their settings.

Figure 275 describes the workflow for setting up a Modbus network. For using Modbus the Modbus ports of the LOYTEC device and the Modbus devices have to be configured. The RS-485 Modbus port must get a Baud rate, the parity is fixed at none and the stop bits are configured to 2. The Modbus TCP port must get the TCP port number of the slave devices (see Section 12.3.1). The Modbus devices have to be configured according to the LOYTEC device's port configuration (see Section 12.5.1). When no device template is available for a Modbus device, this template can be created by manually configuring Modbus data points for the device (see Section 12.5.2). If the Modbus device is already online, data points can be created using the online test feature by scanning register values out of the device (see Section 12.5.3). In both cases the created data points can be exported as a device template. The exported device template can then be used to easily add additional Modbus devices with the same data point configuration (see Section 12.5.4). Also mixing the two methods is possible. When using the device templates also data points can be added manually. The configuration is then downloaded to the device and the device is rebooted (see Section 7.6.4).

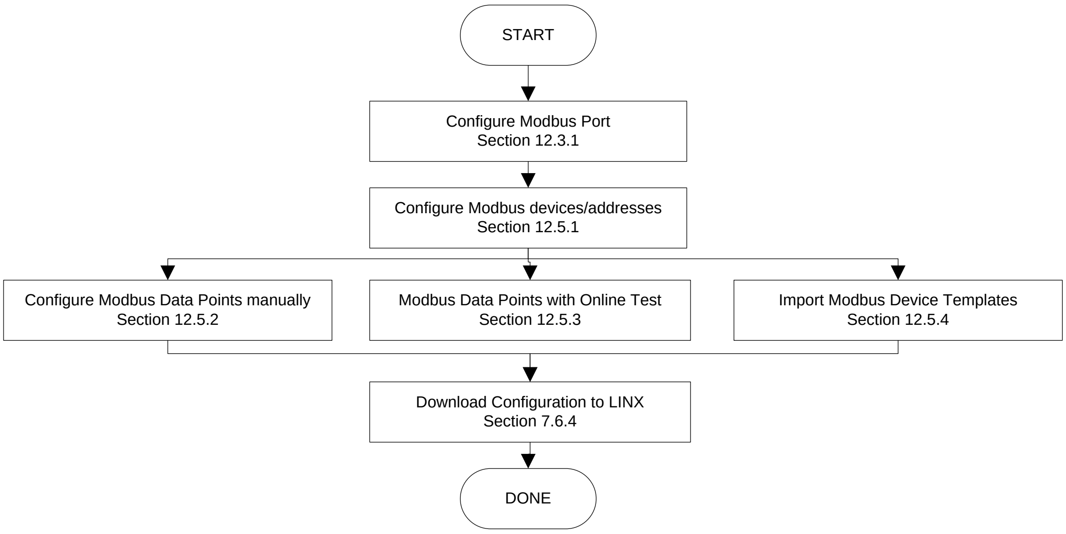


Figure 275: Workflow for offline engineering.

12.5 Using the Configurator for Modbus

12.5.1 Modbus Management Functions

This section describes how the Modbus network management functions can be used. It describes adding and removing Modbus.

The **Modbus Management** dialog shown in Figure 276 shows the list of devices. Devices which have been imported from a template show the template name in the last column. For each device the address, the IP address (if available), the port and the number of data points available on that device is shown.

The Modbus Management dialog can also be used for the configuration of poll groups and for accessing the LOYTEC devices protocol analyzer.

To Start the Modbus Network Management Dialog

1. Connect to the device via FTP as described in section 7.6.1.

2. Select the Modbus dialog by clicking on the **Modbus** button

in the tool bar of the **Datapoints** tab. The Modbus Management dialog opens, showing the **Modbus Device Management** tab displayed in Figure 276.

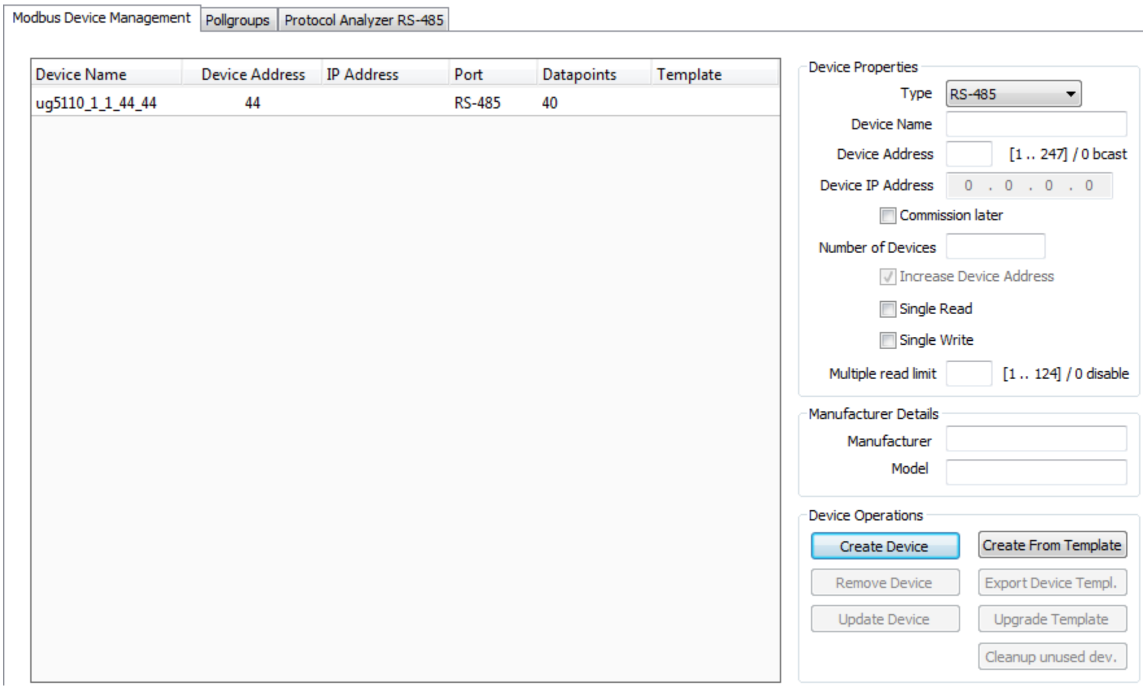


Figure 276: Modbus management dialog.

To Add a Modbus Device Manually

1. Fill in the **Device Address** and select the **Type** (either RS-485 or TCP) from the drop down box.

2. A **Device Name** can be specified. If no device name is specified, the device name is created automatically. If more devices with the same properties have to be created using subsequent addresses, the number of devices can be specified in the input field under the **Device IP Address** field. If a number of TCP devices has to be created, subsequent IP addresses are configured. If the checkbox **Increase Device Address** is checked, also the Device Addresses (unit IDs) of the TCP devices are increased.

3. The **Device Address** specifies the address of the Modbus device ranging from 1 to 255. In case of a TCP device the device address specifies the unit ID. For a RS-485 device the device address has to be unique, TCP devices can have equal device addresses.

4. For TCP devices the **Device IP Address** has to be specified.

5. If the device shall be created without knowing the address of any physical device, leave the above address fields empty and select **Commission later**. This device can then be commissioned on the Web UI (see Section 12.3.3).

6. If the device is not able to read adjacent registers with one read command, activate the checkbox **Single Read**. If the device is not able to process writes of multiple registers in one command, activate the checkbox **Single Write**.

7. The setting **Multiple read limit** allows limiting the number of registers in a multi-read request. The default is disabled. Set this value, if a Modbus device has problems with the default multi-read.

8. The optional manufacturer details just represent **Manufacturer** name and **Model** name. This information is used to identify device templates.

9. Click on the **Create Device** button. This creates the Modbus device and adds it to the device list on the left hand side of the dialog.

10. When a device is selected the device information is displayed in the appropriate fields on the right hand side of the dialog. The information can be changed in the fields. Press the **Update Device** button to store the changes.

11. If a device has to be deleted select the device and press the **Remove Device** button.

To Add a Modbus Device Manually Using a Template without Creating Data Points

1. Enter a name for the Modbus device in the **Device Name** field. If no name is specified the name is created automatically from the name in the template file.

2. Fill in the **Device Address** and select the **Type** (either RS485 or TCP) from the drop down box. If TCP is selected also enter the **Device IP Address**. If more devices with the same properties have to be created using subsequent addresses, the number of devices can be specified in the input field under the **Device IP Address** field.

3. Click on the **Create From Template** button. This opens the **Import Modbus Device Template** dialog shown in Figure 277.

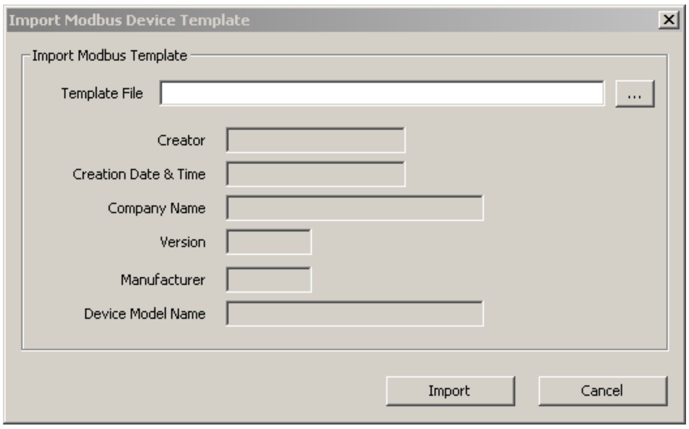


Figure 277 Import Modbus Device Template dialog.

4. Press the  button and select a template file from the **Open** dialog.

5. After selecting the file, the device information is displayed.

6. Press **Import** for importing the template or **Cancel** for closing the dialog without any changes.

7. When a template is imported, a folder with the name of the device is created. Under this folder a **Datapoints** folder containing the data points from the template file is created.

Tip: ***Data points can be added to the data points of the template by right-clicking in the data point list and selecting New Datapoint.***

To Add a Modbus Device Manually Using a Template Creating Data Points

1. Enter a name for the Modbus device in the **Device Name** field. If no name is specified the name is created automatically from the name in the template file.

2. Fill in the **Device Address** and select the **Type** (either RS-485 or TCP) from the drop down box. If TCP is selected also enter the **Device IP Address**. If more devices with the same properties have to be created using subsequent addresses, the number of devices can be specified in the input field under the **Device IP Address** field.

3. Click on the **Create with DP** button. This opens the **Import Modbus Device Template** dialog shown in Figure 277.

4. Press the  button and select a template file from the **Open** dialog.

5. After selecting the file, the device information is displayed.

6. Press **Import** for importing the template or **Cancel** for closing the dialog without any changes.

7. When a template is imported, a folder with the name of the device is created. Under this folder a **Datapoints** folder containing the data points from the template file is created.

8. The created device is shown in the list together with the number of data points.

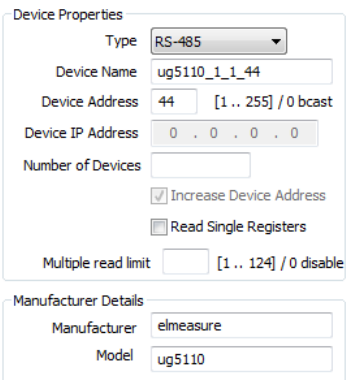
To Remove a Modbus Device

1. Select the device which has to be removed, also multi-select is possible.

2. Press the **Remove Device** button. If the device already has data points, these data points have to be deleted before the remove can be performed.

To Change the Properties of a Modbus Device

1. Select the device. This shows the device properties.



2. Update the properties which have to be changed.

3. Press the button **Update Device**.

4. When the device type is changed, it is verified that no device with the address exists on the appropriate port – on RS485 the device address has to be unique, on TCP the device IP address has to be unique.

12.5.2 Manual Configuration of Data Points

It is possible to manually configure Modbus data points. Manual configuration is done by specifying all information, the Modbus device manufacturer provides.

To Manually Create an Modbus Data Point

1. Click on the Modbus port **Datapoints** folder.
2. Right-click in the data point list view and select **New Datapoint…** in the context menu.
3. This opens the **Create New Modbus Datapoint** dialog showing only the devices which are available on the appropriate port. This dialog is shown in Figure 278.

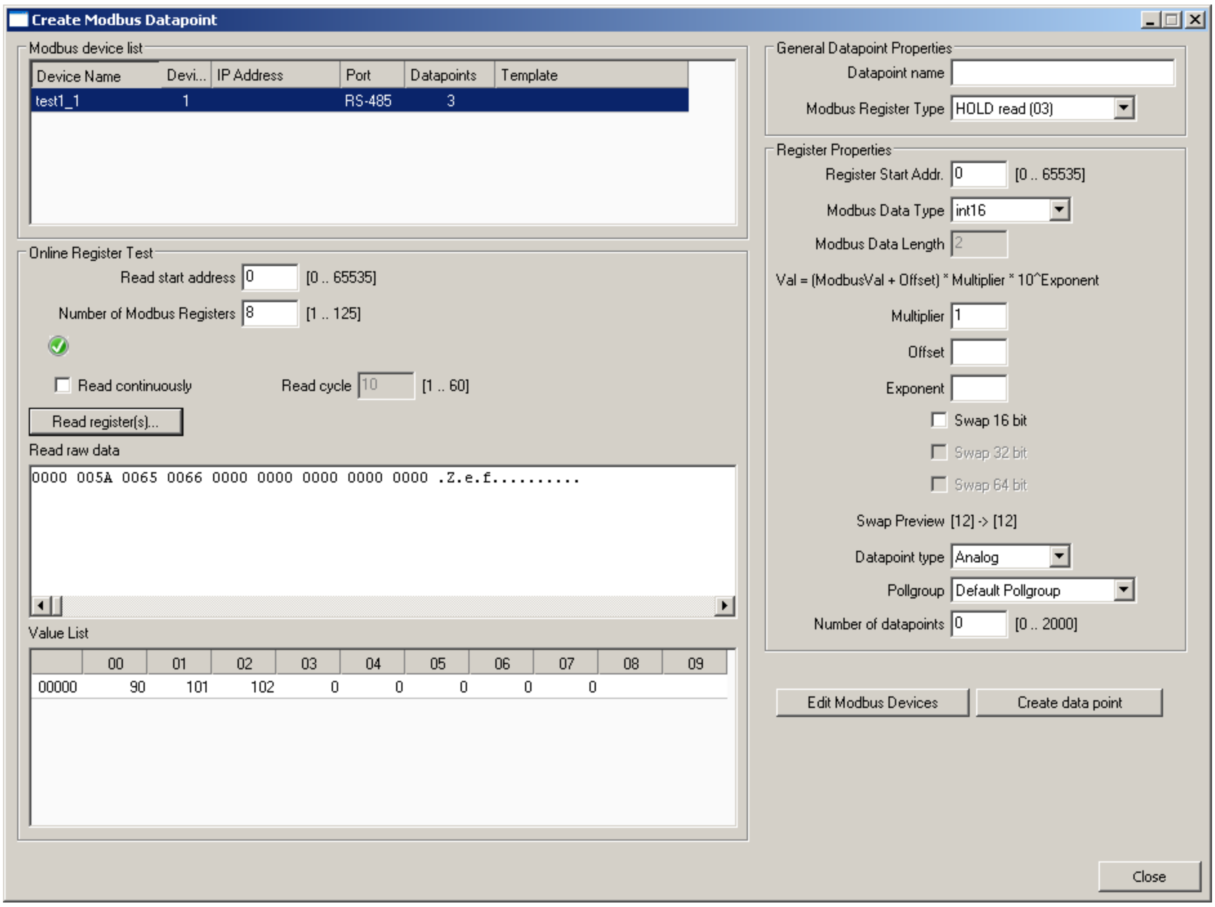


Figure 278: Create Modbus Datapoint dialog.

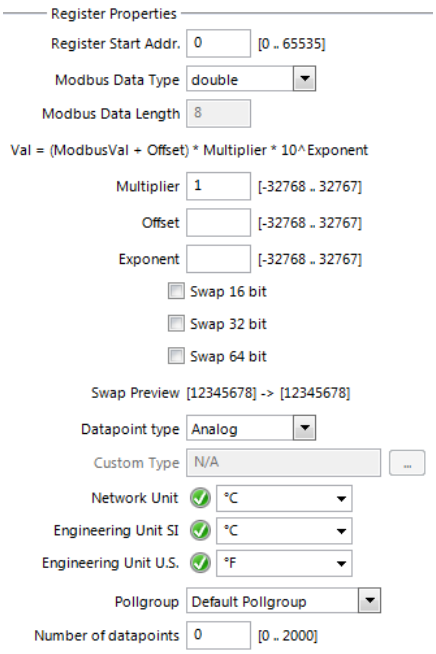
4. If the Modbus device which provides the data point is not in the list, it has to be created. In this case open the Modbus management dialog by clicking the **Edit Modbus Devices** button.
5. Create the device in the **Modbus Management** dialog and close the dialog.
6. Select the device which provides the Modbus data point.

7. Enter the General Data Point Properties. These are the **data point name**, which is automatically created when not specified, and the **Register Type**. The register type of the data point is provided in the Modbus device documentation. The drop down menu shows the Modbus register type, the direction (read and write) and the function code. The data point properties are entered in the presented section of the dialog.

8. Enter the properties of the data point. The register address is specified by the manufacturer. Select the **Modbus Data Type**. This type specifies how the manufacturer stores data in the Modbus device. The **Modbus Data Length** is automatically updated according to the data type. Offset, Multiplier and Exponent can be used for mapping purposes. The Value of the data point is calculated as follows:

$$\text{Value} = (\text{ModbusValue} + \text{Offset}) \cdot \text{Multiplier} \cdot 10^{\text{Exponent}}.$$

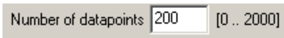
Modbus does not specify any byte orders of the data stored in devices. For some devices it may be necessary to change the byte order. This is done by the check boxes **Swap 16 bit**, **Swap 32 bit** and **Swap 64 bit**. When Swap 16 bit is activated, the 2 byte of a word are swapped, when Swap 32 bit is activated, the 2 words of a 32 bit value are swapped and if Swap 64 bit is activated, the two 32 bit words of a 64 bit value are swapped. A preview of the byte order is shown under the check boxes. Select the **Data Point Type** of the data point (analog value, multi-state or binary) – only the types which are available for the register type-data type combination are shown.



9. For an analog Modbus Master data point define a **Network Unit**. This is the unit of the register on the Modbus device. Optionally, define a unit representation of that remote register value for the metric (SI) and U.S. unit system.

10. Select a poll group for read data points from the **Pollgroup** drop down box. The drop down box is grayed out for write registers. Additional poll groups can be configured in the Modbus Management dialog.

11. In order to create a series of data points in one swoop, enter the **Number of data points** to be created.



12. Press the **Create** button to create the Modbus data points. The register indices will be adapted in ascending order. If data points should already exist at any of those indices, they are skipped.

13. After the point is created the dialog is not closed, so additional data points can be created.

Tip: ***After creating a data point, the poll group can be changed in the data points property view. Also multi-select can be used in the data point property view.***

12.5.3 Data Point Creation with Online Test

The Modbus technology does not provide for an online scan for devices and data points like others do. Sometimes the data sheets from the vendors are inaccurate regarding register indices and swap information. If the Modbus device is already online, the online test function can be used to scan value information out of the device and create data points with that information. The online test function can also be used to test existing data points.

To Create Data Points from Online Test

1. Connect to the device.

2. Open the **Create Modbus Datapoint** dialog as described in Section 12.5.2.

3. In the **Modbus Device List** select the Modbus device, which is online and shall be read.

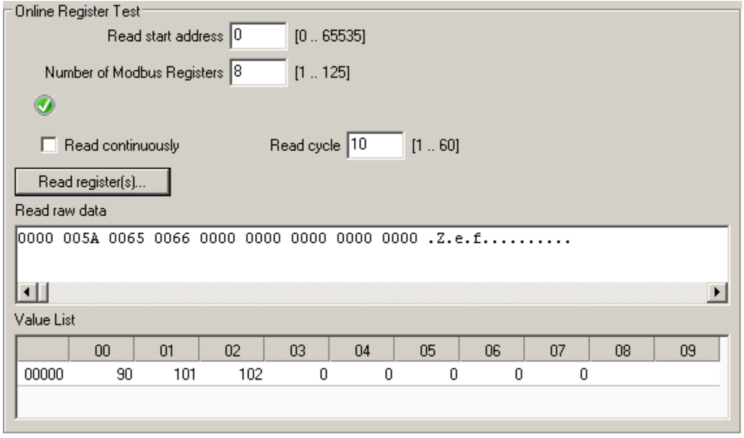
4. Enter a data point name and select the **Modbus Register Type** in the **General Datapoint Properties** group box.

5. In the **Register Properties** group box select the **Modbus Data Type**, scaling and swap setting as assumed.

6. In the **Online Register Test** area enter the **Read Start Address** and **Number of Modbus Registers** to read. Note, that a Modbus register is always 2 Bytes long.

7. Click the **Read register(s)** button.

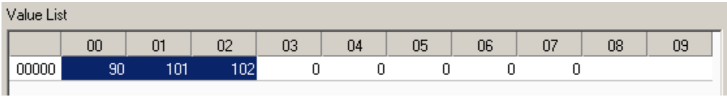
8. The data read from the Modbus device is displayed in the **Read raw data** box as hex bytes and in the **Value list** box as interpreted data.



9. Adapt the value interpretation in the **Register Properties** area until the expected result appears. The results are immediately visible.

10. You may enable the **Read continuously** option, which periodically reads the registers and displays them in the value list. You may use this option to monitor register values and particular registers of interest.

11. Select the register values in the Value list. You may also use multi-select.



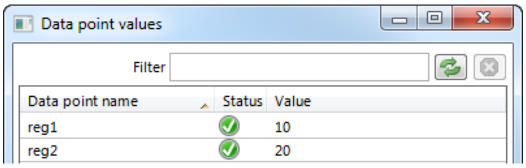
12. Click **Create Data Point**. This creates data points that will provide the selected register values.

To Check Data Points Using Online Test

1. Select one ore more existing Modbus Master data points.

2. Right-click and choose **Test Selected Data Points** from the context menu.

3. The **Data Point Values** window opens and shows the read progress for the selected data points. For each completed read a status result and a value is shown.



12.5.4 Importing via Device Templates

For some Modbus devices special templates are available which specify all available data points of a Modbus device as well as the device properties. Such templates can be imported into the configuration.

To import a Modbus Device Template

1. Right click on the Folder **Modbus Device Templates** and select **Import device template** from the context menu.

2. This opens the **Import Modbus Device Template** dialog shown in Figure 279.

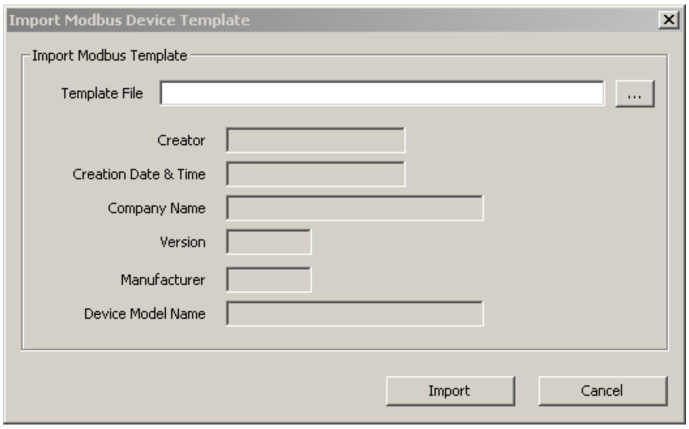


Figure 279 Import Device Template dialog.

3. Press the  button and select a template file from the **Open** dialog.

4. After selecting the file, the device information is displayed.

5. Press **Import** for importing the template or **Cancel** for closing the dialog without any changes.

6. When a template is imported, a folder with the name of the device is created. Under this folder a **Datapoints** folder containing the data points from the template file is created.

Tip: *Data points can be added to the data points of the template by right clicking in the data point list and selecting New Datapoint.*

Importing a device template from the folder list does not create a device instance. Device instances can only be created using the import in the Network Management Dialog. In this dialog also the device instances can also be created with their data points.

To Use Imported Data Points On the Device

1. Go to the **Datapoints** folder of the device of the Modbus Templates.

2. Select the desired data points, also multi-select is possible.

3. Either press on the **Use on Device** button  or right-click and select **Use on Device**. This opens the **Choose Modbus Device Instance** dialog.

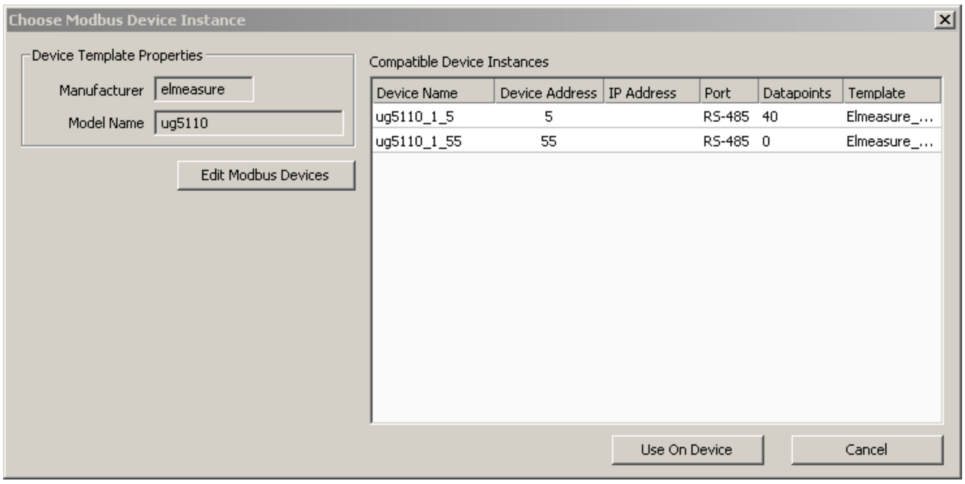


Figure 280: Choose Modbus Device Instance dialog.

4. The device list displays all devices which have the same Manufacturer, and the same Model Name as the template. If no device instance matches the device template, create one by entering the Modbus management dialog. The dialog can be entered by pressing the **Edit Modbus Devices** button. In the management dialog, the device instance can be either created manually (take care of entering the correct manufacturer and model) or by simply importing the template again.

5. Select one or more device instances from the list and press the **Use On Device** button. This creates for each selected data point and each selected device one data point in the Modbus Port's data point list.

12.5.5 Creating Device Templates

Modbus device templates can be created from a data point configuration. In fact, it is only possible to create a device template using an existing device or an existing device template with data points. This device and its data points can either be configured manually or also imported from a device template itself.

A Modbus device template contains the following configuration items describing a Modbus device:

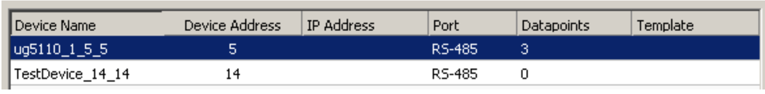
- Modbus device settings,
- Modbus data points,
- Folders used to organize Modbus data points on a device,
- Structure types, multi-state maps, historic filters used by Modbus data points,
- Pollgroup definitions used by Modbus data points.

To Create a Modbus Device Template Using Devices

1. Select the Modbus dialog by clicking on the **Modbus** button

in the tool bar of the **Datapoints** tab. This opens the **Network Management** dialog opens as described in Section 12.5.1.

2. The device list shows all devices of the current configuration. Select the device you want to export.



3. Press the **Export Device Templ.** button. This opens the **Export Modbus Device Template** dialog shown in Figure 281. The list on the left side of the dialog shows the names of the data points which are exported to the template.

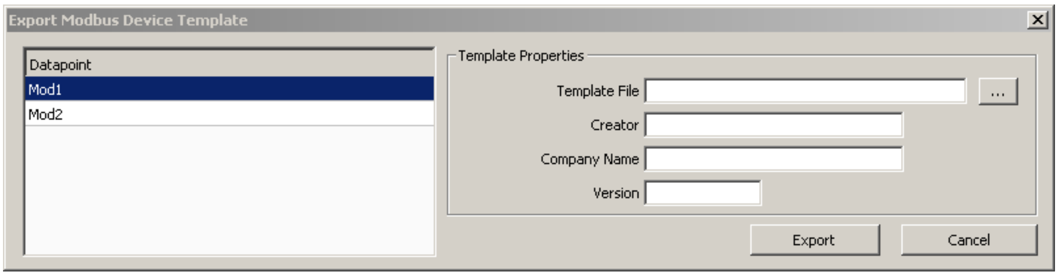


Figure 281 Export Modbus Device Template dialog.

4. Press the ... button and select a template file from the **Save** dialog.

5. Enter the **Creator**, **Company Name** and **Version** for the template. This information is stored in the template file, when importing the template file the information is displayed after selecting the file.

6. Press the **Export** button.

To Create a Modbus Device Template File Using a Device Template

1. Right-click on the folder of the device template that has to be exported or its data point folder and select **Export Device Template…** from the context menu.

2. This opens the **Export Modbus Device Template** dialog as shown in Figure 281. Proceed as described above.

12.5.6 Poll Groups

In a Modbus network, the master has to poll the slave devices. Input data points are therefore attached to a poll group. If nothing else is specified, the default poll group is used for input data points. The default poll group has a poll cycle of 60 seconds.

Two different types of poll groups can be specified:

- Time-based: The poll group is triggered on a time base. This means that after a specific time – the poll cycle – the poll group is processed.

- Trigger-based: The poll group is triggered on a special trigger data point. As soon as the trigger condition is met, the poll group is processed.

Tip: ***The poll group a data point is attached to can be changed in the properties view of the data points. The poll group can also be changed for multiple data points using multi-select.***

To Create a Time-Based Poll Group

1. Select the Modbus dialog by clicking on the **Modbus** button

in the tool bar of the **Datapoints** tab. This opens the **Modbus Management** dialog.

2. Open the **Pollgroups** tab. This shows the dialog displayed in Figure 282.

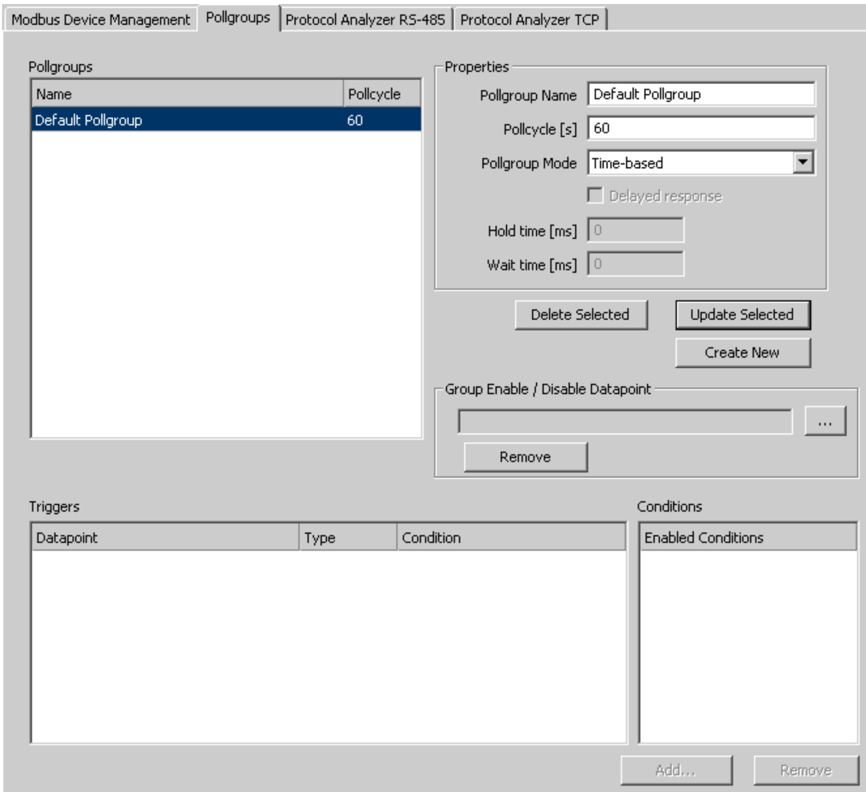


Figure 282: Pollgroup Management dialog.

3. The default poll group is selected and its properties are displayed. Enter the name of the new poll group and enter the poll cycle in seconds. Values in multiples of '0.1' are allowed for fast polling. Make sure that under **Pollgroup Mode** Time-based is selected.

4. Press the **Save** button to store the poll group and continue editing.

5. If a poll group needs to be updated or deleted, select the poll group edit the data and press the **Update Selected** or **Delete Selected** button.

6. Press the **Close** button to finish editing. When the poll groups have not been saved, a dialog asks whether the changes have to be saved or not.

To Create a Trigger-Based Modbus Poll Group

1. Select the Modbus dialog by clicking on the **Modbus** button

in the tool bar of the **Datapoints** tab. This opens the **Modbus Management** dialog.

2. In the **Modbus Management** dialog, open the **Pollgroups** tab. This displays the poll groups management tab displayed in Figure 282.

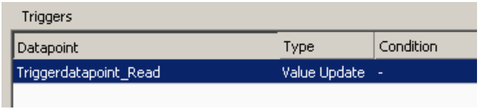
3. Enter a new poll group name and select **Trigger-based**.

4. Create the poll group by pressing the **Create New** button.

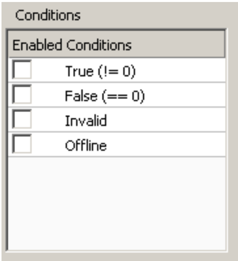
5. Select the new poll group. This enables the **Add…** and **Remove** buttons from the triggers.

6. Press the **Add…** button and select a trigger data point. This can for example be a binary user register.

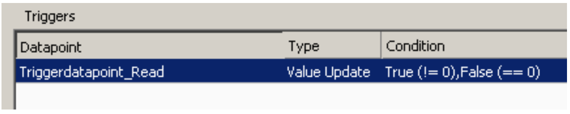
7. The selected trigger data point appears in the trigger list as shown:



8. Select the trigger from the list and check the desired trigger conditions.



9. The trigger conditions are then displayed in the trigger list.



10. Press the **Save** button to store the changes in the poll groups.

12.5.7 Create Modbus Slave Data Points

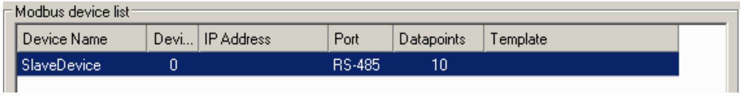
The Modbus interface of the device can also be configured to be in Modbus slave mode. In this mode, Modbus slave data points can be created. These data points can represent any of the supported Modbus register types. They are locally available as registers to other Modbus masters for reading and writing. The Modbus slave registers appear like user registers; they are value data points. The register types and indices are defined when creating the data points. They must, however, lie within the register index range, which has been set in the system settings for the Modbus port (see Section 12.4.1).

To Create Modbus Slave Registers

1. Select the **Datapoints** folder of the Modbus port.

2. Right-click in the data point list view and select **New Datapoint…** in the context menu.

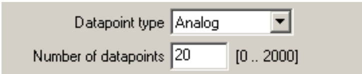
3. This opens the **Create New Modbus Datapoint** dialog. In the Modbus device list there is only the **SlaveDevice** entry. It cannot be de-selected.



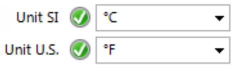
4. Enter the data point name and select the desired **Modbus Register Type**.

5. In the **Register Properties** group box enter the register start address, the Modbus data type and optionally scaling and swap settings.

6. Select the data point class, which shall be created and the number of data points below.

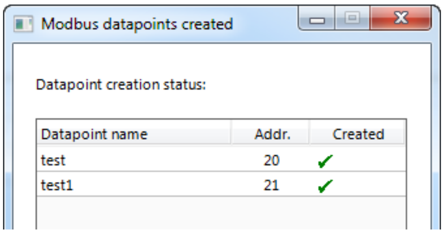


7. For analog Modbus Slave registers you may define a metric (SI) and U.S. unit. Depending on the chosen unit system, the Modbus Slave register will change its representation (i.e., its value) on the network.



8. Click on the button **Create data point**.

9. The number of Modbus slave register data points is created. The register indices are incremented for each created register. If a register index is already in use, the data point for this index is skipped. Created data points are reported with a green check mark.

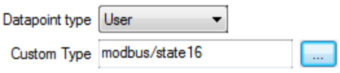


12.5.8 Structured Modbus Data Points

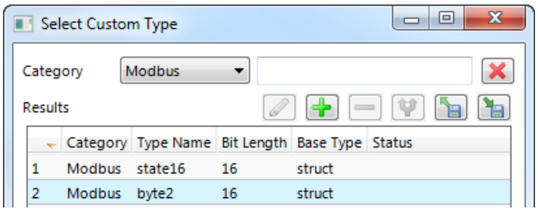
Modbus registers are always multiples of two Bytes. Modbus devices may encode two Byte fields or bit fields into the registers. For accessing byte or bit-fields, data point structures can be used that provide single data points for the respective sub-fields (see Section 6.2.11). The Configurator provides pre-defined types for byte and bit-fields in the Modbus scope: modbus_state16 and modbus_byte2.

To Create a Structured Modbus Register

1. Create the register as described in Sections 12.5.2 and 12.5.7 and select the data point type **User**.



2. Click the button and select a structure data type from the chooser dialog.



3. Continue with the data point creation as described in the previous Sections. The Configurator will create structured Modbus data points.

12.5.9 Modbus Protocol Analyzer

When connected to a device a protocol analyzer is available for each Modbus port. The protocol analyzer can be found in the Modbus Management dialog. On every Modbus port a protocol analyzer tab is available. Figure 283 shows the Modbus protocol analyzer

The status on the right hand side of the dialog shows, if the device is connected or if the protocol analyzer is stopped or started. When connected to a device, the protocol analyzer can be started by pressing the **Start Protocol** button. This starts the protocol analyzer in the device. Every time a transmission is made on the Modbus port, the transmission is displayed in the list. Additionally the protocol data is stored in the device in a rotating log file. The protocol log can hold up to 40 kB of protocol data. So also when the Configurator was not running for an interesting time, the protocol data can be loaded from the device using the **Load From Device** button. The protocol data can be stored as CSV file using the **Save** button, with the **Clear** button, the shown protocol is deleted.

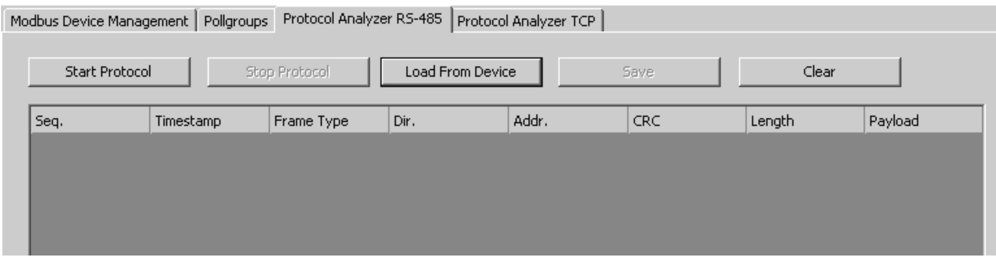
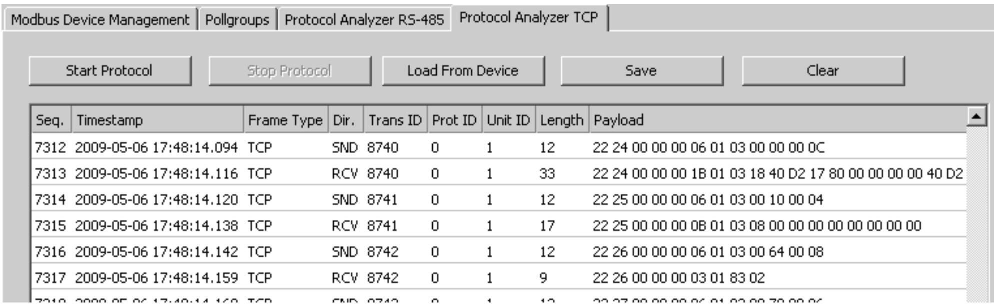


Figure 283: Modbus protocol analyzer.

Figure 284 shows a typical protocol analyzer output for the Modbus TCP port. It shows the following information for TCP:

- **Seq.**: sequence number, which is automatically created in the device. This number is unique for one port.
- **Timestamp:** transmission time.
- **Frame Type:** 'TCP' or 'Damaged' when something happened with the frame.
- **Dir.**: direction. Either SND (send) or RCV (receive)
- **Trans ID:** Transaction ID
- **Prot ID:** Protocol ID
- **Unit ID:** Unit ID
- **Length:** Payload length of the data frame

- **Payload:** Payload in hexadecimal numbers (this column cannot be used for sorting).

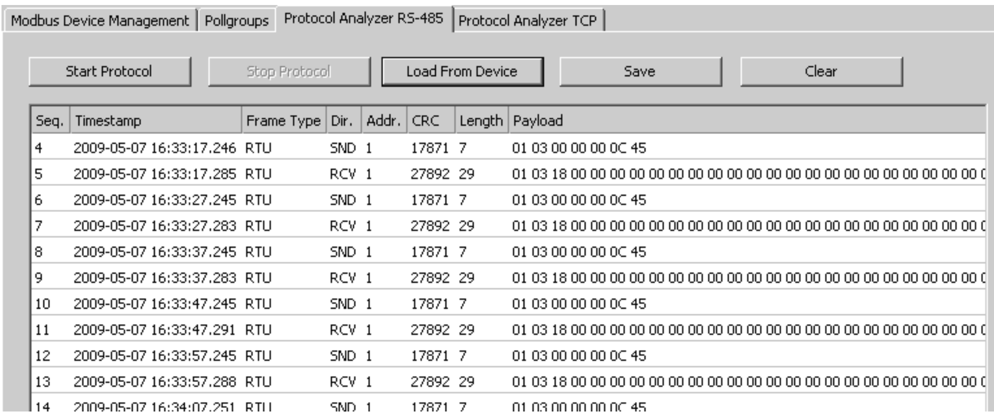


Seq.	Timestamp	Frame Type	Dir.	Trans ID	Prot ID	Unit ID	Length	Payload
7312	2009-05-06 17:48:14.094	TCP	SND	8740	0	1	12	22 24 00 00 00 06 01 03 00 00 00 0C
7313	2009-05-06 17:48:14.116	TCP	RCV	8740	0	1	33	22 24 00 00 00 1B 01 03 18 40 D2 17 80 00 00 00 00 40 D2
7314	2009-05-06 17:48:14.120	TCP	SND	8741	0	1	12	22 25 00 00 00 06 01 03 00 10 00 04
7315	2009-05-06 17:48:14.138	TCP	RCV	8741	0	1	17	22 25 00 00 00 0B 01 03 08 00 00 00 00 00 00 00 00
7316	2009-05-06 17:48:14.142	TCP	SND	8742	0	1	12	22 26 00 00 00 06 01 03 00 64 00 08
7317	2009-05-06 17:48:14.159	TCP	RCV	8742	0	1	9	22 26 00 00 00 03 01 83 02

Figure 284: Typical protocol analyzer output for Modbus TCP port.

Figure 285 shows a typical protocol analyzer output for the Modbus RS-485 port. It shows the following information for RS-485:

- **Seq.**: sequence number, which is automatically created in the device. This number is unique for one port.
- **Timestamp:** transmission time.
- **Frame Type:** RTU, ASCII or Damaged when something happened with the frame.
- **Dir.**: direction. Either SND (send) or RCV (receive)
- **Addr.:** Slave address
- **CRC:** CRC of the transmission
- **Length:** Payload length of the data frame
- **Payload:** Payload in hexadecimal numbers (this column cannot be used for sorting)



Seq.	Timestamp	Frame Type	Dir.	Addr.	CRC	Length	Payload
4	2009-05-07 16:33:17.246	RTU	SND	1	17871	7	01 03 00 00 00 0C 45
5	2009-05-07 16:33:17.285	RTU	RCV	1	27892	29	01 03 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6	2009-05-07 16:33:27.245	RTU	SND	1	17871	7	01 03 00 00 00 0C 45
7	2009-05-07 16:33:27.283	RTU	RCV	1	27892	29	01 03 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8	2009-05-07 16:33:37.245	RTU	SND	1	17871	7	01 03 00 00 00 0C 45
9	2009-05-07 16:33:37.283	RTU	RCV	1	27892	29	01 03 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10	2009-05-07 16:33:47.245	RTU	SND	1	17871	7	01 03 00 00 00 0C 45
11	2009-05-07 16:33:47.291	RTU	RCV	1	27892	29	01 03 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
12	2009-05-07 16:33:57.245	RTU	SND	1	17871	7	01 03 00 00 00 0C 45
13	2009-05-07 16:33:57.288	RTU	RCV	1	27892	29	01 03 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
14	2009-05-07 16:34:07.251	RTU	SND	1	17871	7	01 03 00 00 00 0C 45

Figure 285: Typical protocol analyzer output for Modbus RS-485 port.

The frame type **Damaged** indicates that either a bad packet (malformed frame, CRC error) was received or that no response packet was received where it is expected.

13 KNX

13.1 Introduction

KNX is a European and international standard (EN 50090, ISO/IEC 14543-3) designed for building automation. It is the successor to the European Installation Bus (EIB) and also merged Batibus and the European Home System Protocols (EHS) into one standard. KNX covers lighting, HVAC, sunblinds, A/V control, metering among many other application areas.

KNX models actuators and sensors as a set of communication objects. A communication objects represents a typed value, such as a room temperature, a switch state or an illumination level. These objects are then mapped to group addresses. Typically, a sensor emits a group message containing its present value which is received and processed by the actuators. Sensors and actuators can be parameterized to implement various switching and controlling scenarios.

KNX supports different media, among them twisted pair (TP1), powerline, RF and IP-based networks (KNXnet/IP). Each KNX node is a master and can transmit a frame at any time.

LOYTEC devices support two media types, KNX/TP1 and KNXnet/IP. KNX/TP1 is a twisted pair medium, which uses a CSMA protocol with collision avoidance at 9600 baud. Simple devices can be bus-powered while routers and gateways are typically self-powered. KNXnet/IP transmits KNX frames over a UDP multicast address and is typically used in the backbone area.

A KNX network can be structured in a loopless topology. KNX routers are called couplers and connect two network segments, which are called lines. The smallest topology contains a single line with up to 64 devices which can be extended to 256 devices. If the system exceeds this limit, up to 15 lines can be connected to an area using line couplers. Finally, up to 15 areas can be connected which allows approximately 57600 devices to be connected to a KNX network.

13.2 Hardware Installation

KNXnet/IP is supported on the built-in Ethernet interface, so no additional hardware is required for KNXnet/IP. KNXnet/IP can be enabled in the port configuration of the Ethernet interface.

For using KNX/TP1, the LKNX-300 interface has to be attached to the extension port of the LOYTEC device. This port is labeled EXT on the LINX-12x/15x/22x and LGATE-950 devices. The KNX/TP1 functionality has to be enabled in the port configuration settings for the extension port.

13.2.1 LKNX-300 Installation

The LKNX-300 needs to be attached to the extension bus of the L-INX or L-GATE. It is possible to connect the LKNX-300 before, mid of or behind LIOB devices. In any case, you need to connect the 3-wire **EXT** port. Follow the cabling instructions of Figure 286 between the **EXT** port of the L-INX and the **EXT** port on the LKNX-300.

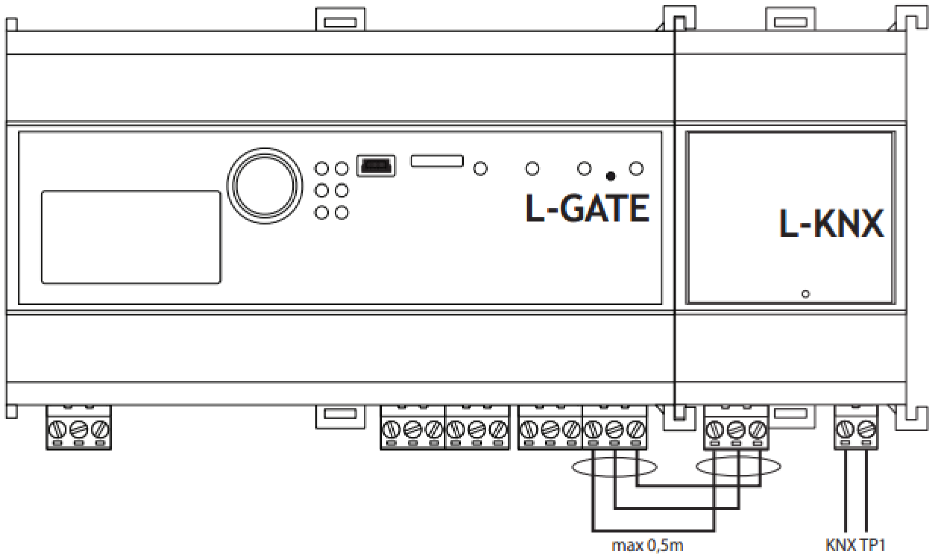


Figure 286: Connection L-KNX over the extension port.

13.3 KNX Network

KNX devices are modeled as a set of communication objects which communicate over group message to each other. Each possible communication target receives the group message and checks if there are communication objects connected to this group address. The communication objects use the standardized KNX data point types (DPTs).

The KNX interfaces can be used independently. It is possible to use a KNX TP1 and a KNXnet/IP network at the same time. The interfaces can be connected to the same or to two different KNX networks.

The LOYTEC KNX interface allows sending and receiving these messages and to connect them to data points which can be used for Alarming, Scheduling and Trending. They also can be used in connections to other technologies and as variables for math objects and IEC 61131 programs.

13.4 Web Interface

This section describes the Web interface for the KNX interfaces.

13.4.1 Configuration

To enable the KNX ports, use the port configuration web interface. The settings for enabling the KNX/TP1 interface are shown in Figure 287 and the settings for enabling the KNXnet/IP interface are shown in Figure 288. On devices with multiple IP interfaces, the KNXnet/IP protocol can be activated only on one of them.

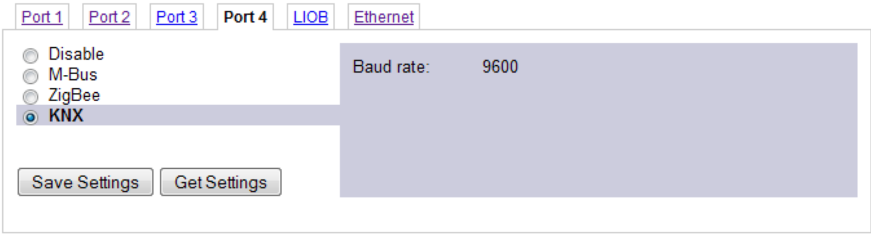


Figure 287: Enabling the KNX/TP1 interface

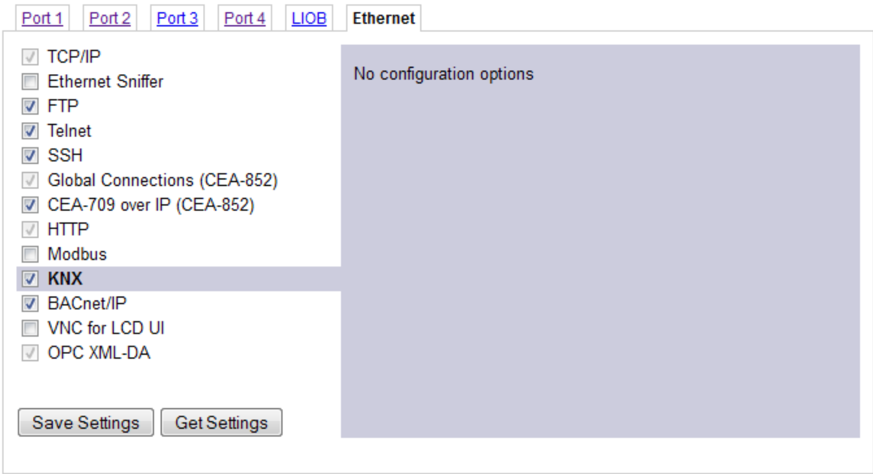


Figure 288: Enabling the KNXnet/IP interface

13.4.2 Data Points

KNX data points can be accessed through the Web UI as described in Section 5.3.1.

13.4.3 KNX Protocol Analyzer

By clicking the link **KNX PA** in the **Statistics** menu, the KNX protocol analyzer page is shown as displayed in Figure 289.

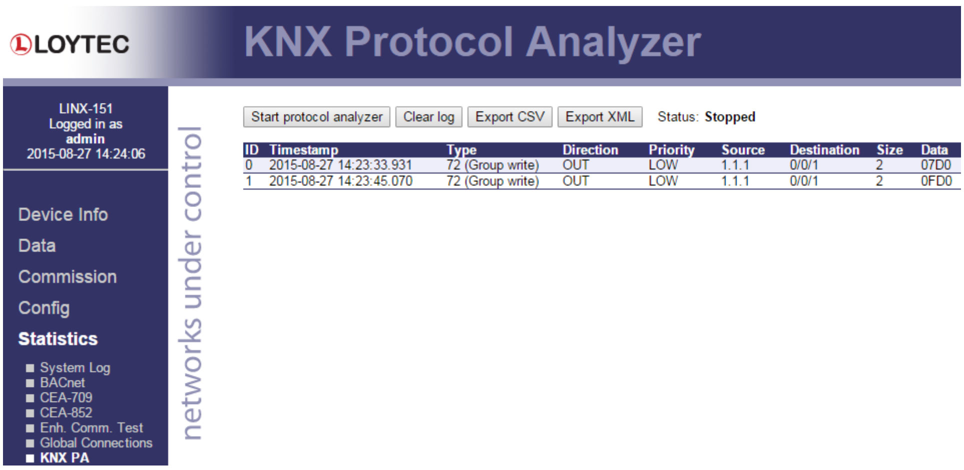


Figure 289: KNX protocol analyzer.

Next to the control buttons the status of the protocol analyzer is shown. If the analyzer is started, an automatic refresh is performed every 60 seconds. By pressing the button **Start**

Protocol Analyzer / **Stop Protocol Analyzer** the protocol analyzer can be started / stopped.

For every frame sent or received a line is presented in the table. When the protocol analyzer has been stopped, click on **Export CSV** to store the protocol log as a CSV file or **Export XML** to store as an XML file. The XML file can be opened in the ETS protocol viewer. **Clear Log** clears the log data.

13.5 Configurator

This section describes how to use the Configurator software for the management of KNX data points. For further information on the Configurator software refer to Chapter 7.

13.5.1 Activating KNX Configuration

Before a new KNX configuration can be managed, at least one KNX interface must be enabled. The project settings are described in detail in Section 7.3.

To Activate the KNX Configuration

1. Open the project settings dialog.
2. In the **System Settings** tab enable the KNX for Port 4 and/or for the Ethernet port, as shown in Figure 290.
3. Press the **OK** button.

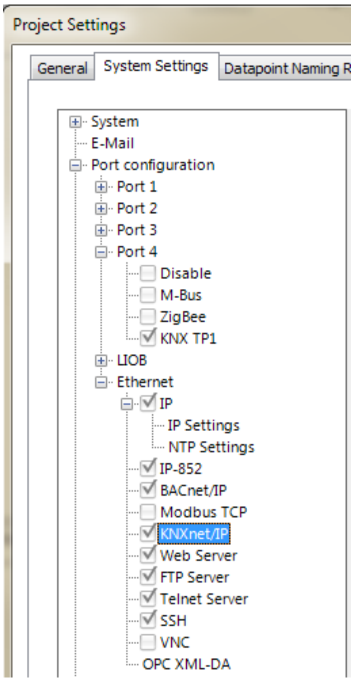


Figure 290: Enabling KNX/TP1 and KNXnet/IP in the Project Settings Dialog.

Important: ***If the KNX Port is deactivated via the checkbox or a firmware or model version is chosen, which does not support KNX, the complete KNX configuration is deleted. In this case a dialog is displayed, which has to be confirmed.***

If the configurator is connected to a device, the **Download** button can be used to download the device configuration.

13.5.2 KNX Project Settings

The project settings dialog provides a **KNX** tab for interface specific KNX settings. The interface dialog, as shown in Figure 291 allows selecting a KNX interface for editing in the left-hand tree view. The right-hand pane allows modifying the following properties of the selected KNX interface:

- **Group addressing mode**: The group addressing mode defines how group addresses are displayed numerically. This setting is a display property and has no influence on the KNX communication itself.
    - **Free**: The addresses are displayed as integers, e.g. 12345.
    - **Two-Level**: The addresses are displayed as main groups (5 bit) and group addresses (11 bit), e.g. 6/57.
    - **Three-Level**: The addresses are displayed as main groups (5 bit), middle groups (3 bit) and group addresses (8 bit), e.g. 6/0/57.
- **Unique project ID**: This field contains the ETS project ID from the last import, for example P-053C. This field is used to avoid importing two different projects into one configuration.
- **Physical address**: This field configures the physical (individual) address of the device, e.g. 1.2.3. It needs to be unique in the KNX network and needs to match the line to which the device is connected to.

The following settings are available for the KNXnet/IP interface only:

- **Multicast address**: This IPv4 multicast address is used for sending and receiving KNX. The default value is **224.0.23.12**.
- **Multicast port**: This UDP port number is used for sending and receiving KNX frames. The default value is **3671**. It is recommended to keep the default value, as some KNXnet/IP devices cannot operate on alternate ports.
- **Multicast TTL**: This is the Time to live value for UDP multicast packets. The default value is **16**.

The following fields are informational only and represent the state of the latest KNX database import.

- **Project name**: This field contains the ETS project name.
- **Project number**: This field contains the ETS project number.
- **Contract number**: This field contains the ETS contract number.
- **Last modification**: This field contains the modification time of the ETS project.
- **Start date**: This field contains the start date of the ETS project.
- **Project Id**: This field contains the imported project Id.
- **Comment**: This field contains the text comment of the ETS project.
- **Completion status**: This field contains the completion status of the ETS project.

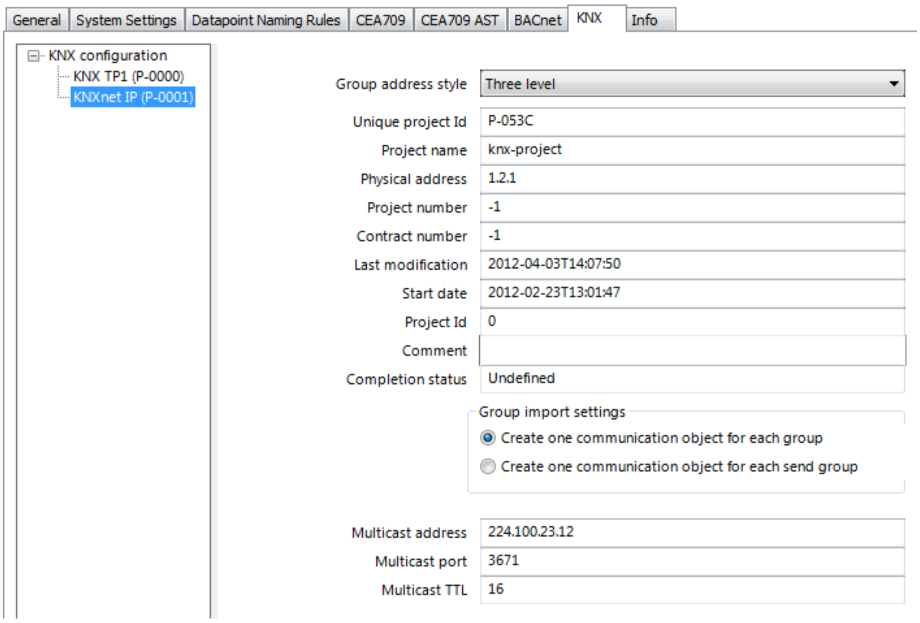


Figure 291: KNX Project Settings

- **Group import settings**: This setting allows modifying the mapping of linked groups:
  - **Create one communication object for each group**: This setting will create one communication object for each imported group.
  - **Create one communication object for each send group**: This setting will create one communication object for each send group and attach linked group addresses to this communication object.

The data point naming rules tab (see Figure 292) allows specifying how data point names are automatically derived from imported KNX communication objects and groups. The preview shows how names would look like, when the check marks are modified. Note, that changing the name components does not change already created KNX data points; this setting affects only new data points created by an import.

- **Separator**: This setting defines a separator character, which is inserted between the data point name components. As a default no separator is configured.

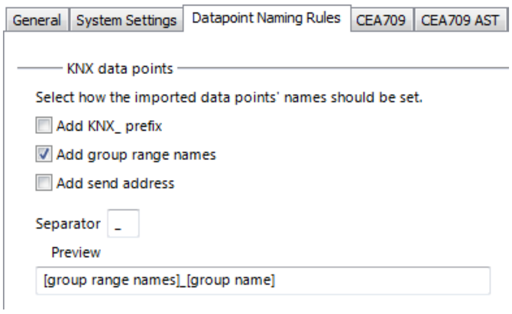


Figure 292: KNX Data Point Naming Rules

13.5.3 Data Point Manager for KNX

The Configurator uses a central concept to manage data points. The data point manager is used to select, create, edit and delete data points. The dialog is divided into three sections:

- The folder list (Figure 293),
- The data point list (Figure 294),
- And a property view.

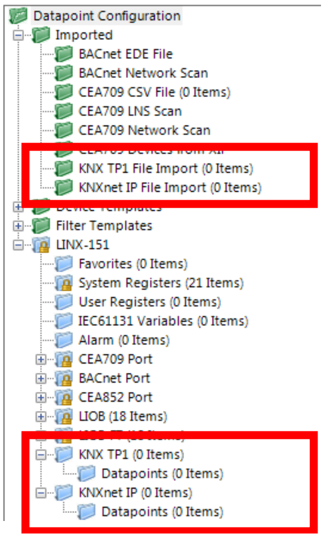
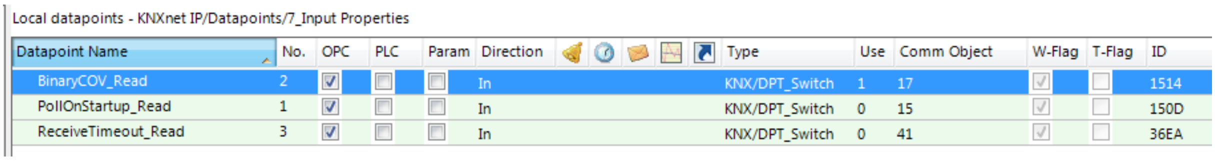


Figure 293: Data Point Manager Dialog with KNX import and device folders.



Local datapoints - KNXnet IP/Datapoints/7_Input Properties

Datapoint Name	No.	OPC	PLC	Param	Direction	Type	Use	Comm Object	W-Flag	T-Flag	ID
BinaryCOV_Read	2	☑	☐	☐	In	KNX/DPT_Switch	1	17	☑	☐	1514
PollOnStartup_Read	1	☑	☐	☐	In	KNX/DPT_Switch	0	15	☑	☐	150D
ReceiveTimeout_Read	3	☑	☐	☐	In	KNX/DPT_Switch	0	41	☑	☐	36EA

Figure 294: Data Point Manager Dialog with KNX Data Point List.

13.5.4 Folder List

At the left is a list of folders which is used to sort the available data objects by their category. There are a number of predefined KNX folders available. All other folders are described in section 7.2.1:

- **KNX TP1 File Import**: This folder contains the data points from the ETS database import for the KNX/TP1 interface.
- **KNXnet IP File Import**: This folder contains the data points from the ETS database import for the KNXnet/IP interface.
- **LINX-XXX**: This is the device folder (see Section 7.2.1). For KNX, additional port folders exist:
    - **KNX TP1:** This folder contains the data points for the KNX TP1 interface.
    - **KNXnet IP:** This folder contains the data points for the KNXnet/IP interface.

13.5.5 Network Port Folders

The KNX network port folder on the device has the same structure of sub-folders as the other network port folders in Section 7.2.2. Currently only the **Datapoints** folder exists for the KNX network port.

13.5.6 KNX Properties

Apart from the common data point properties discussed in Section 7.2.4 the data points of the KNX technology have additional properties. These properties are shared among all data points which are connected to the same KNX communication object.

- **Data Point Type**: This property allows setting a data type for the data point. The configurator restricts changes to data point types of the same bit length. The configurator data point repository contains all common KNX data points types (DPTs). For analog objects the DPT defines the fixed network unit.
- **Communication flag**: This flag acts as a communication master switch for the communication object. If it is cleared, the communication object does neither receive nor transmit KNX frames. This flag should be typically kept **on**.
- **Read flag**: The read flag enables communication objects to respond to network read requests. If the flag is cleared, read requests are ignored. Only one object in a group should have the read flag set. This flag is usually set to **off**.
- **Transmit flag**: The transmit flag enables sending group writes to the KNX network when the data point value is written by the application. This flag should be **on** for communication objects which have at least one write data point. The configurator automatically sets this flag when a write data point is created. If the transmit flag is cleared and write data points exist for this communication object, the configurator asks for confirmation before removing write data points.
- **Update flag**: The update flag determines how the communication object reacts on receiving read responses. If the flag is **off**, the communication object ignores read responses initiated by other communication objects. If the flag is **on**, the communication object treats all read responses as group writes and updates the data point properly. This flag is usually set to **off**.
- **Write flag**: The write flag enables receiving group writes from the KNX network. This flag should be **on** for communication objects which have at least one read data point. The configurator automatically sets this flag when a read data point is created. If the write flag is cleared and read data points exist for this communication object, the configurator asks for confirmation before removing read data points.
- **KNX transmission priority**: This property allows selecting one of the KNX transmission priorities for the data point, **low**, **high** and **alert**. Typically, this property is set to **low**.
- **KNX send group address**: The send group address is the group address used for sending group writes. It is also automatically used as a receiving address. As the group addresses are managed by the database import, they should not be modified manually. The KNX project settings determine the format of this string (free, two-level or three-level).
- **KNX receive group addresses**: This is a semicolon separated list of additional receive addresses. As the group addresses are managed by the database import, they should not be modified manually. The KNX project settings determine the format of this string (free, two-level or three-level).
- **KNX communication object ID**: The communication object ID is used for identifying a communication object. As this ID is managed internally, it should not be modified manually.

13.6 KNX Workflow

This section discusses the workflows for adding the device to an existing KNX network. Figure 295 shows the workflow from project start to a working device configuration. The following sections will describe this process in detail.

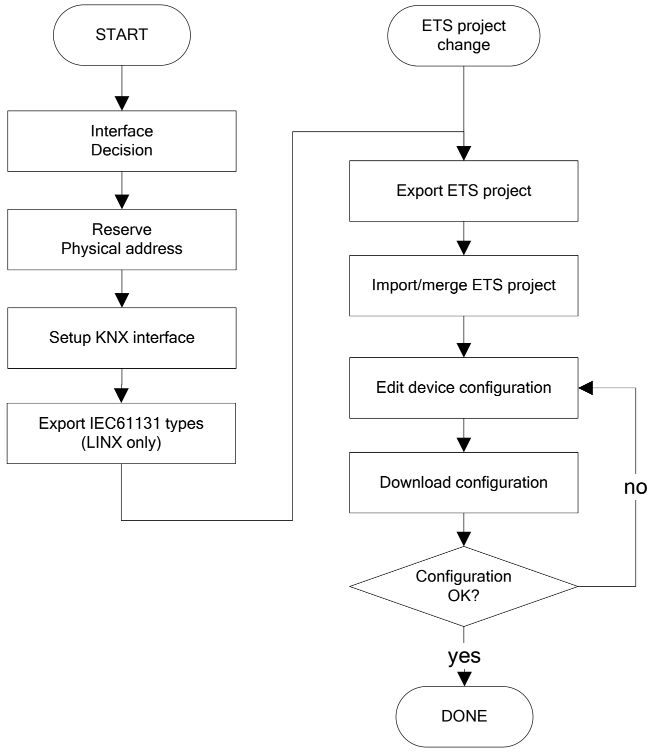


Figure 295: Workflow for KNX projects.

13.6.1 Selecting a KNX Interface

This section describes how the device can be connected to a KNX network and how to decide which media type is to be used:

1. If the KNX network consists of a single TP1 line, the device is simply connected to the line via the LKNX-300.

2. If the KNX network consists of several TP1 lines and couplers, there are two options:

    a. If the device shall use group addresses that are mainly used within a line, add the device to this line. This will reduce the traffic over line couplers.

    b. If the device shall communicate with group addresses spread over several lines, add the device to the main line or to the area line.

3. If the KNX network contains a TP1/IP router and/or has a KNXnet/IP backbone, use the KNXnet/IP interface.

13.6.2 Reserve a Physical Address

After selecting the line, a free physical address on this line must be reserved.

Note: *To make sure that the address is reserved in ETS4, you can add an arbitrary device to the line, rename it to "Reserved for LINX" and use the assigned address for the KNX interface.*

Use this address in the interface configuration dialog (see Section 13.5.2).

13.6.3 Coupler Configuration

If the KNX network contains couplers, it must be made sure that the device receives all group telegrams contained in the device configuration. If the network consists of a single line, this section can be ignored.

To configure the couplers to forward all groups to the device, the following options are available:

1. Enable the switch **Pass through line couplers** option in the ETS4 group address or group range options, as shown in Figure 296. If this switch is set, the couplers will forward the group addresses towards the main line.

2. Set the “group telegrams” coupler parameters to **route** instead of **filter** of the coupler. As this will forward all group traffic, this is not the recommended method for larger networks.

3. Drag & drop a group address or an address range on the line which contains the device. This will add manual filter rules to the affected couplers.

Note: *Only configure the group addresses or address ranges used by the device. Forwarding all group telegrams to the main line will only work well for small installations due to the backbone traffic. The coupler filter table can be previewed in the ETS4 with the coupler context menu “Preview filter table …”*

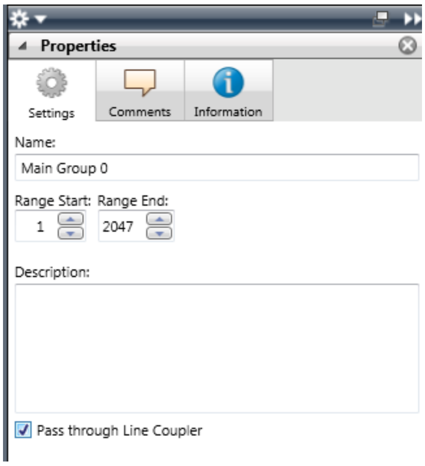


Figure 296: Setting the Pass through Line Coupler option.

13.6.4 Use KNX Data Types in IEC61131

KNX data points can be used in an IEC61131 program, just like other data points. With a L-LOGICAD version as of 4.2.4, all necessary KNX data types are pre-installed. They can be found under ‘Standard_Libs/KNX_Types’.

To use KNX data points in the IEC61131 program with an older L-LOGICAD installation, the KNX data types have to be exported as structured text and have to be imported into a logi.CAD library.

To export KNX types, select **Manage Structured Types** from the Configurator **Tools** menu. Select the export button, as shown in Figure 297.

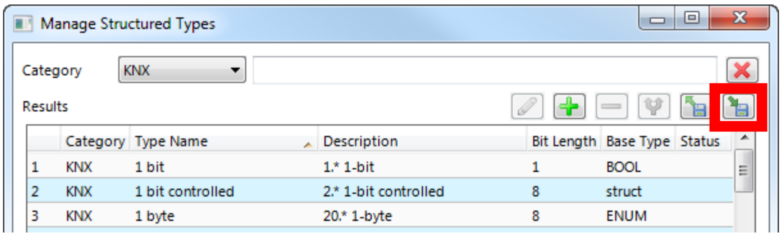


Figure 297: Export KNX types

In the **Export Types to Disk** dialog, as shown in Figure 298,

1. clear the filter and select the KNX category,
2. copy all KNX data types to the export list
3. select an export file (*.ST) and
4. click the save button.

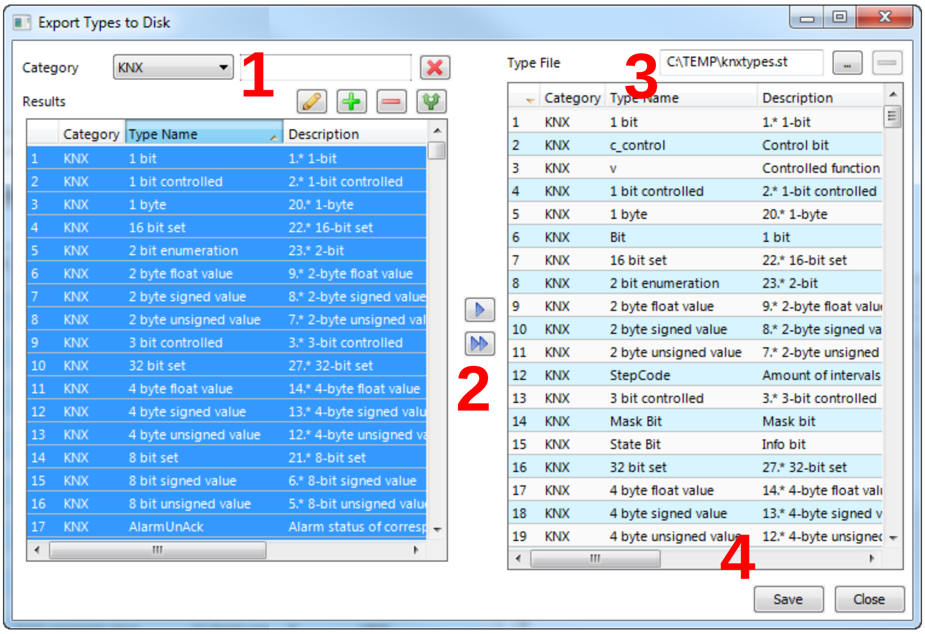


Figure 298: Exporting KNX types as Structured Text.

5. In logi.CAD, create a new library KnxTypes with the context menu **New** → **Library**.
6. Select **Export/Import** → **Start ST-Import** from the context menu of the newly created library, as shown in Figure 299.

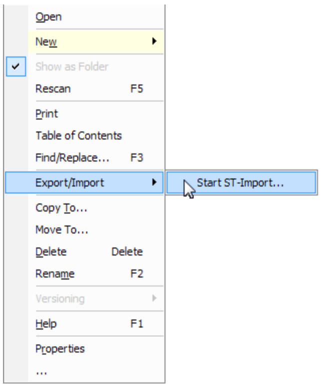


Figure 299: Importing KNX types in logi.CAD.

7. Check that there have not been any warnings or errors during the import.

13.6.5 Setup a Configurator Project

After creating a new project, at least one KNX interface needs to be enabled. Enable the KNX interfaces in the project configuration, as shown in Section 13.5.1. To make sure that the system settings are written to the device, connect to the device and download the system settings from the system settings editor.

Use the Web interface to verify that the device has the desired KNX interfaces enabled, as shown in Section 13.4.1.

13.6.6 ETS Project Export

When the KNX network has been installed, the database can be exported to a .knxproj file. The export function can be found in the **Projects** tab of the ETS4 main screen, as shown in Figure 300.

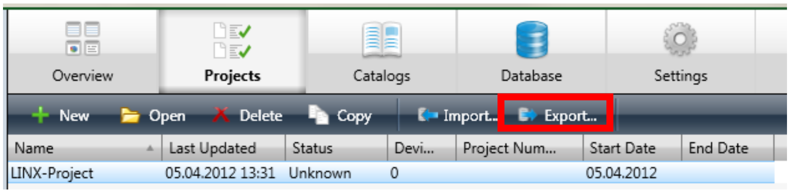


Figure 300: Exporting the ETS project

The KNX project then can be saved to a .knxproj file.

The project export has to be repeated if

- Group numbers have changed or
- New groups should be available to the device.

13.6.7 Configurator Project Import

For each enabled KNX interface, there exists a KNX import folder in the data point tree view. The folders are named “KNX TP1 File Import” and “KNXnet IP File Import” respectively.

To import a KNX project file, the context menu entry **Import File** of the import folder needs to be selected. This opens a file requester for selecting the .knxproj-File.

After selecting the file, the configuration merge dialog is displayed, as shown in Figure 301. This dialog allows the initial group address import and is also used to merge configuration changes when the ETS project has evolved.

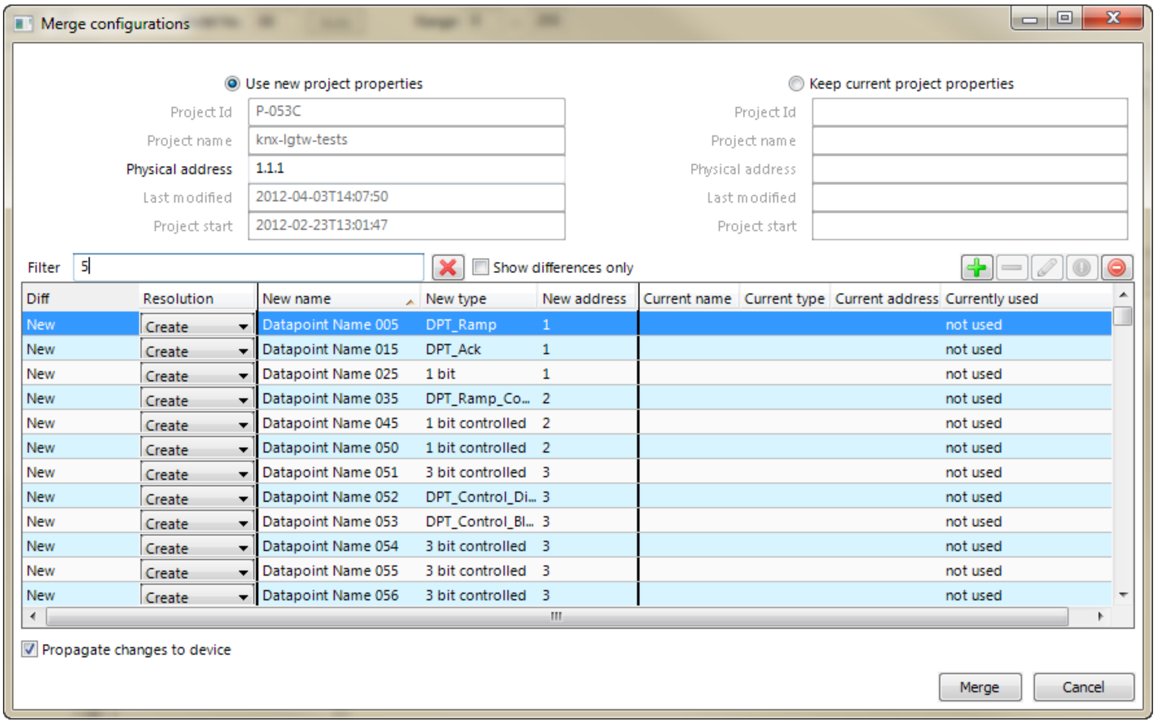


Figure 301: KNX import dialog

The top part of the dialog shows the project description. For the first import, you need to enter the reserved physical address for this device.

The main part of the dialog is the merge editor which compares the current device configuration with the KNX project configuration. The table displays the following columns:

- **Diff**: This column displays whether the group has been added (**New**), is unchanged (**Unchanged**), has been changed (**Changed**), or has been removed (**Removed**) in the KNX project.
- **Resolution**: This columns allows to select different actions for each change:
  - **Create**: This action is the default for groups that exist in the KNX project file, but not yet in the device configuration.
  - **Delete**: This action is the default action for groups that exist in the configuration, but not in the KNX project file. Deleting a data point being in use is not possible, as this would break the PLC program or the gateway configuration.
  - **Update**: This action changes the properties of the group, e.g. if the name has changed.
  - **Force**: This action forces the configurator to the KNX database file as the authoritative data source, even if this will break the PLC program or gateway configuration. It can be used to forcibly delete data points.
  - **Ignore**: This action will ignore the change.

- **New name**: This is the group name in the imported file.
- **New type**: This is the data point type in the imported file.
- **New address**: This is the group address in the imported file.
- **Current name**: This is the group name in the current configuration.
- **Current type**: This is the data point type in the current configuration.
- **Current address**: This is the group address in the current configuration.

Additional controls:

- The filter field above the merge editor allows a substring search to select a subset of the imported groups. The filter can be disabled with the clear filter button.
- The checkbox **Show differences only** allows displaying only those groups which have changes.
- The action buttons on the right top allow switching the Resolution state of the selected data points.
- If the **Propagate changes to device** checkbox is enabled (default), changes will not only be updated in the import folder, but also propagated to existing data points on the device.

The KNX project import presents all groups which are used in the KNX project by at least one device. The default data point names are assembled by the group names, separated by underscores.

To import or merge the KNX project configuration, press the **Merge** button. To leave the device configuration unchanged, press **Cancel**.

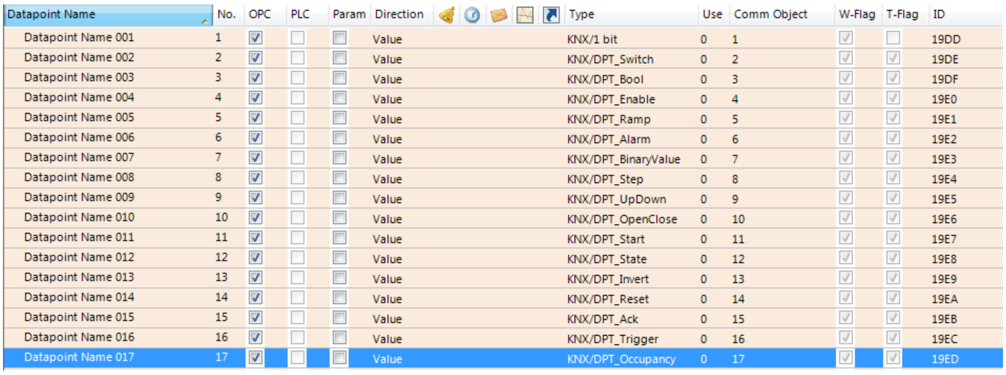
13.6.8 Creating Data Points

After importing the KNX project file, the import folder contains a list of possible data points. The data point properties can be edited in the import folder. Especially, the data type and communication flags can be selected for each group address. Only data types matching the configured byte length of the group address can be selected. The import folder will typically look like Figure 302.

To use the data points on the device, multi-select the desired data points and select **Use on Device** from the context menu.

When **Use on Device** is selected on an imported data point, the value of the communication flags decide which data points are created:

- If the **Write flag** is enabled, an input data point will be created.
- If the **Transmit flag** is enabled, an output data point will be created.
- If both, the **Write flag** and the **Transmit flag** are enabled, a value data point will be created.



Datapoint Name	No.	OPC	PLC	Param	Direction	Type	Use	Comm Object	W-Flag	T-Flag	ID
Datapoint Name 001	1	☑	☐	☐	Value	KNX/1 bit	0	1	☑	☐	19DD
Datapoint Name 002	2	☑	☐	☐	Value	KNX/DPT_Switch	0	2	☑	☑	19DE
Datapoint Name 003	3	☑	☐	☐	Value	KNX/DPT_Bool	0	3	☑	☑	19DF
Datapoint Name 004	4	☑	☐	☐	Value	KNX/DPT_Enable	0	4	☑	☑	19E0
Datapoint Name 005	5	☑	☐	☐	Value	KNX/DPT_Ramp	0	5	☑	☑	19E1
Datapoint Name 006	6	☑	☐	☐	Value	KNX/DPT_Alarm	0	6	☑	☑	19E2
Datapoint Name 007	7	☑	☐	☐	Value	KNX/DPT_BinaryValue	0	7	☑	☑	19E3
Datapoint Name 008	8	☑	☐	☐	Value	KNX/DPT_Step	0	8	☑	☑	19E4
Datapoint Name 009	9	☑	☐	☐	Value	KNX/DPT_UpDown	0	9	☑	☑	19E5
Datapoint Name 010	10	☑	☐	☐	Value	KNX/DPT_OpenClose	0	10	☑	☑	19E6
Datapoint Name 011	11	☑	☐	☐	Value	KNX/DPT_Start	0	11	☑	☑	19E7
Datapoint Name 012	12	☑	☐	☐	Value	KNX/DPT_State	0	12	☑	☑	19E8
Datapoint Name 013	13	☑	☐	☐	Value	KNX/DPT_Invert	0	13	☑	☑	19E9
Datapoint Name 014	14	☑	☐	☐	Value	KNX/DPT_Reset	0	14	☑	☑	19EA
Datapoint Name 015	15	☑	☐	☐	Value	KNX/DPT_Ack	0	15	☑	☑	19EB
Datapoint Name 016	16	☑	☐	☐	Value	KNX/DPT_Trigger	0	16	☑	☑	19EC
Datapoint Name 017	17	☑	☐	☐	Value	KNX/DPT_Occupancy	0	17	☑	☑	19ED

Figure 302: Imported KNX data points

13.6.9 Editing KNX Data Points

Once KNX data points have been created, they can be edited using the property editor, as described in Section 13.5.6.

Now it is possible to use Alarming, Scheduling and Trending on the KNX data points. They can be exposed to OPC, and can be used as parameters and IEC 61131 variables. Further they can be used in connections, global connections and math objects.

The KNX data point properties are synchronized between the read and write properties, so for example changing the data type of the read data point will also change the data type of the write data point.

Note: *Because the PLC program and the OPC tags are derived from the data point name, it is advisable to keep the data point name constant. If the KNX group names are expected to be changed, it is a good idea to create data point links in the Favorites folder and use the links in the PLC program or for the OPC server. Then the KNX project can be merged without having to take care of the group names.*

13.6.10 Alarming, Scheduling and Trending

KNX data points can be alarmed by the generic alarm server. The configuration of alarming is described in Section 7.13.

KNX data points can be scheduled by the generic scheduler. For devices with CEA709 interfaces, the CEA709 scheduler is used to schedule KNX networks. This works also when the CEA709 node is no commissioned. For BACnet devices, the KNX data points can be also scheduled by a BACnet scheduler. The configuration of scheduling is described in Section 7.12.

KNX data points can be trended with the generic trends which are described in Section 7.14. They can also be configured with historic filters as described in Section 7.17.

14 EnOcean

14.1 Introduction

EnOcean is an international standard (ISO/IEC 14543-3-10) designed for wireless devices optimized for solutions with ultra-low power consumption and energy harvesting. This means EnOcean sensors can be self-powered and draw energy from a button press or a solar cell without the need for a battery. EnOcean wireless communication uses different frequency bands in Europe, U.S. and Japan.

The EnOcean Alliance standardizes so-called EnOcean equipment profiles (EEPs) to make sensors and actuators from different vendors interoperable. The EEP is an identifier which is used for interpreting a wireless datagram and extracting its contents. For secure communication EnOcean provides a security option, that device vendors may choose to implement.

The LOYTEC product supports the EnOcean standard and provides the following features:

- Support all common EnOcean equipment profiles (EEPs) for sensors and actuators.
- Configurable through device templates using the Configurator software.
- Web UI for teach-in, signal strength, and value test.
- Easy device replacement on the Web UI.
- Connected via L-ENO EnOcean interface over the USB port.
- Support of multi-channel EnOcean devices.
- Encrypted wireless connection for EnOcean devices that support it.
- Supports Mailbox function for sleepy actuators (e.g., battery-powered radiator valve).

14.2 Hardware Installation

For using EnOcean, the LENO-80x EnOcean interface has to be attached to the USB port of the LOYTEC device. This port is labeled USB1 or USB2 on the LINX-12x/15x/22x and LGATE-950 devices. The EnOcean functionality has to be enabled in the port configuration settings for the USB port.

The L-ENO EnOcean interfaces are available in three different versions for worldwide use:

- LENO-800: Europe 868 MHz band
- LENO-801: USA/Canada 902 MHz band
- LENO-802: Japan 928 MHz band

14.3 Web Interface

14.3.1 Configuration

To enable the EnOcean protocol on the USB port, use the port configuration Web interface. The settings for enabling the EnOcean protocol are shown in Figure 303. The protocol information area shows whether a LENO-80x EnOcean interface has been connected and provides some details on that interface such as EnOcean ID and serial number.

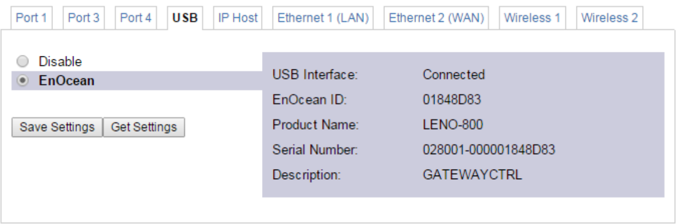


Figure 303: Enabling the EnOcean protocol

14.3.2 Data Points

EnOcean data points can be accessed through the Web UI as described in Section 5.3.1.

14.3.3 Commissioning

The **Commission** Web interface provides a page for managing and commissioning EnOcean devices. This page lists all EnOcean devices, which have been created in the data point configuration as shown in Figure 304. The list shows device names as created in the configuration. The device **Status** can be one of the following:

- OK for a working EnOcean device,
- Uncommissioned for an unlinked EnOcean device that needs teach-in,
- Old Data if no new data has been received within a reasonable timeframe (one day),
- Waiting for Device ID if a teach-in is in progress.

The **Profile** column shows information on the profile ID and received data when expanding it. The **RSSI** column shows the signal strength and intermediate hops this device is reached over.

To teach-in an uncommissioned device, click the **Teach-In** button. The Web interface then waits for a teach-in message sent by an EnOcean device with matching profile ID. Press the button on the EnOcean device and the corresponding EnOcean device ID is associated with the device. A manual assignment can be done by editing the device ID of the respective device.

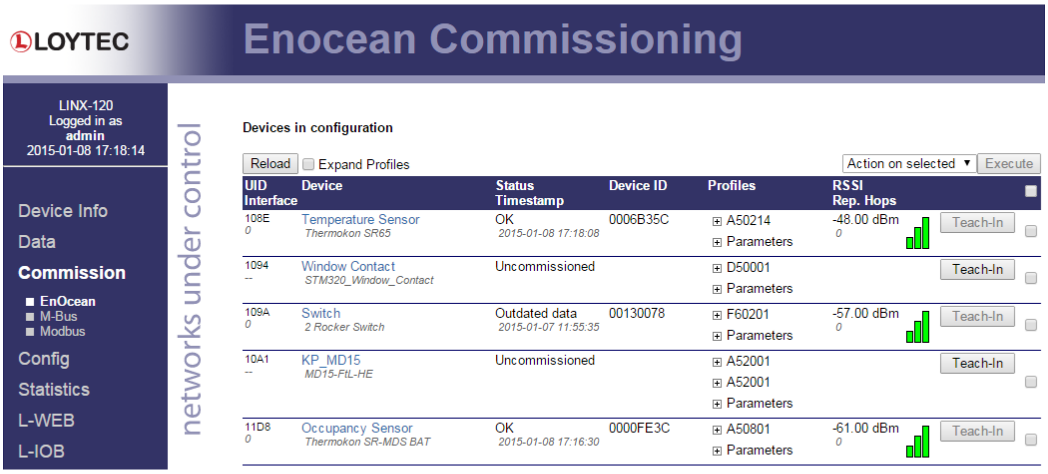


Figure 304: Commission EnOcean devices

Expand the **Parameters** item to display and edit a **Description** and **Location** string describing the EnOcean device. These strings can also be found as data points under the respective EnOcean device folder and appear as parameters in LWEB-900. A click on the device name link will lead you to that folder location in the data point Web interface.

To clear device assignments check one or more check boxes at the end of each line, choose the **Decommission** option in the **Action on selected** drop-down and click **Execute**. The selected devices will then be unassigned and appear as uncommissioned again. Devices can also be temporarily disabled. When disabled, no further data is processed from the respective devices until they are enabled again later.

14.3.4 Statistics

Figure 305 shows a typical output of the statistics information which can be displayed for the EnOcean ports. For each port available one statistics tab is displayed. The statistics can be cleared for each EnOcean port separately by pressing the **Clear EnOcean statistics** button. A refresh of the statistics is done automatically. To stop automatic update, deselect the **Live update** checkbox. For manual update press **Update EnOcean statistics**.

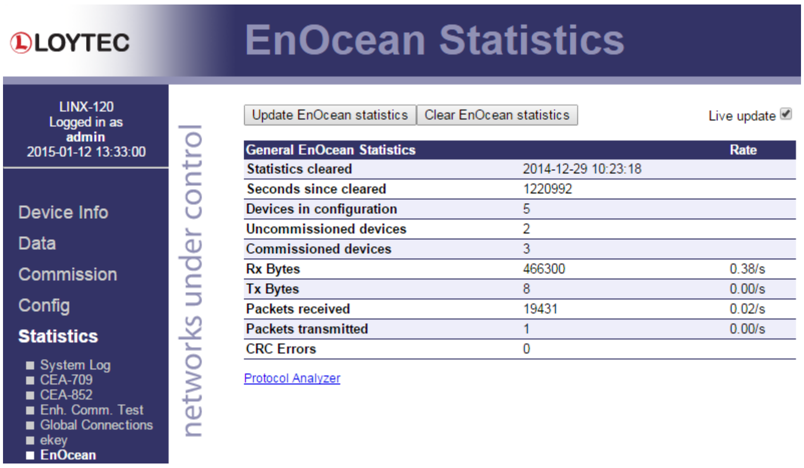


Figure 305: Statistics of the EnOcean port.

The following information is available:

- Statistics cleared: last time of statistics reset,
- Devices in configuration: Total EnOcean devices in data point configuration,

- Uncommissioned devices: Devices that need tech-in,
- Commissioned devices: Devices with teach-in completed,
- Rx Bytes: number of bytes received,
- Tx Bytes: number of bytes sent,
- Packets received: number of EnOcean packets received,
- Packets transmitted: number of EnOcean packets transmitted,
- CRC errors: number of communication errors with wrong CRC.

14.3.5 Protocol analyzer

By activating the link **Protocol Analyzer** (available in all EnOcean statistics tabs), the protocol analyzer page is shown as displayed in Figure 306.

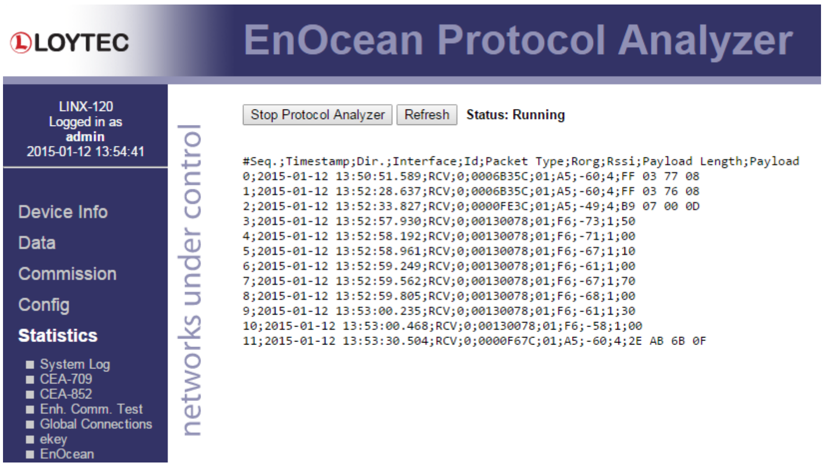


Figure 306: EnOcean protocol analyzer.

Next to the button the status of the protocol analyzer is shown. If the analyzer is started, an automatic refresh is performed every 60 seconds. By pressing the button **Start Protocol Analyzer** / **Stop Protocol Analyzer** the protocol analyzer can be started / stopped.

For every frame sent or received a line is presented using comma separated values. The page refreshes automatically with a given interval. Press the **Refresh** button to get the latest updates immediately. When stopped click on **Save Log** to store the protocol log as a CSV file. **Clear Log** clears the log data.

14.4 Configurator

14.4.1 Activating EnOcean

Before EnOcean devices can be added to a data point configuration, the EnOcean interface in the project settings must be enabled. The project settings are described in detail in Section 7.3.

To Activate EnOcean in the Configuration

1. Open the project settings dialog.

2. In the **System Settings** tab enable the EnOcean protocol on the USB port, as shown in Figure 307.

3. Press the **OK** button.

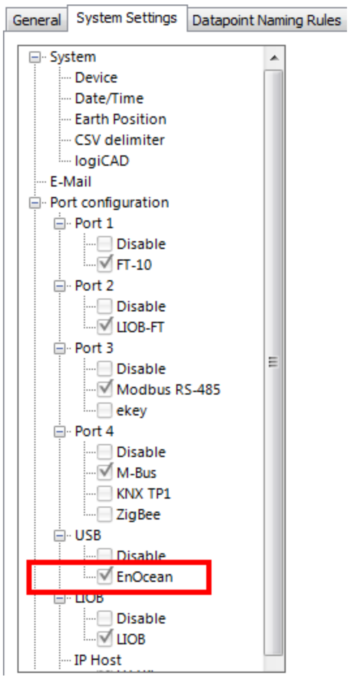


Figure 307: Enabling EnOcean in the Project Settings Dialog.

Important:	***If the EnOcean protocol is deactivated via the checkbox or a firmware or model version is chosen, which does not support EnOcean, the entire EnOcean configuration is deleted. In this case a dialog is displayed, which has to be confirmed.***

If the Configurator is connected to a device, the **Download** button can be used to download the device configuration.

14.4.2 Data Point Manager for EnOcean

The Configurator uses a central concept to manage data points. The data point manager is used to select, create, edit and delete data points. The dialog is divided into three sections:

- The folder list (Figure 308),
- The data point list (Figure 309),
- And a property view.

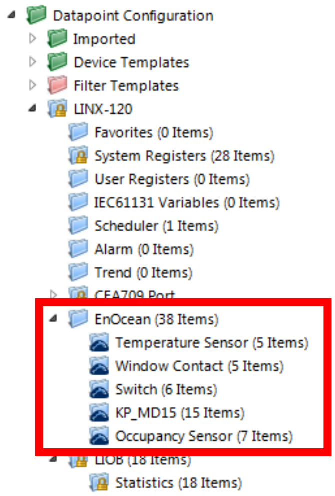


Figure 308: Data Point Manager with EnOcean device folders.

LINX-120 ▸ EnOcean ▸ Temperature Sensor

Datapoint Name	No.	OPC	Param	PLC in	PLC out	Direction	Description	Type	Use	ID
Temperature	1	☑		☐		In	Temperature (linear)	EnOcean/Temperature -20...	0	108F
RSSI	2	☑		☐		In	Received signal strength	EnOcean/RSSI	0	1090
Location	3	☑	☑	☐	☐	Value	Location of the device	EnOcean/string	0	1091
Description	4	☑	☑	☐	☐	Value	Description of the device	EnOcean/string	0	1092
ID	5	☑	☑	☐	☐	Value	Unique EnOcean ID	EnOcean/string	0	1093

Figure 309: Data Point Manager Dialog with EnOcean Data Point List.

14.4.3 Port Folder

The EnOcean port folder represents the EnOcean interface. In the EnOcean port folder, one folder per EnOcean device is created, which contains the data points for that EnOcean device. EnOcean device folders are identified by a small EnOcean logo on the folder icon. They can be deleted, duplicated, renamed and organized in sub-folders.

The data points in an EnOcean device folder cannot be deleted or renamed. Some of their properties such as OPC exposure, PLC in/out, parameter can be modified. The data points named ID, Description, Location are parameters and are also available in LWEB-900.

14.5 EnOcean Workflow

14.5.1 Creating EnOcean Devices from Device Templates

EnOcean devices are created from EnOcean device templates. The Configurator is distributed with a library of common device templates. Device templates can also be imported from an external source and are stored in the data point configuration.

To Create an EnOcean Device

1. Click on the **EnOcean** port folder.

2. Right-click in the data point list view and select **New EnOcean Device…** in the context menu.

3. In the **Create Device** dialog enter a Device Name and a Count of devices to be created as shown in Figure 310.

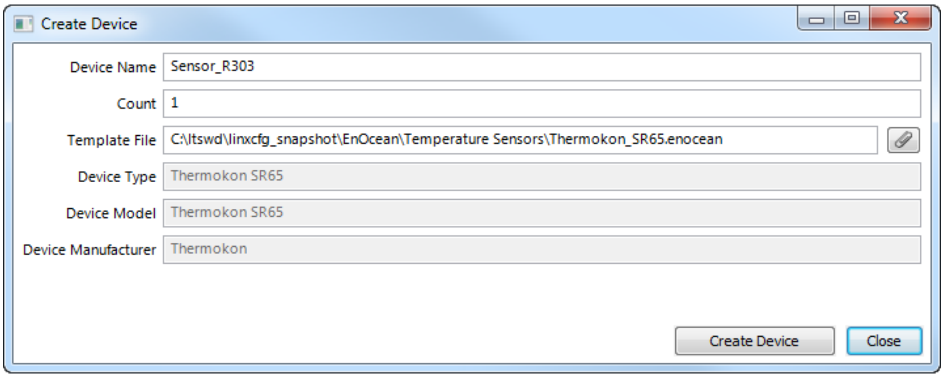


Figure 310: EnOcean device creation dialog

4. Then click on and choose a template file. As a default the directory of the distributed EnOcean device templates is opened. Choose the desired template file.

5. The device type, device model and device manufacturer information is displayed. Then click **Create Device**.

14.5.2 Edit EnOcean Data Points

The EnOcean data points are located under their respective device folders. The data points cannot be deleted or renamed. Some of the data point properties can be edited using the property editor.

Now it is possible to use Alarming, Scheduling and Trending on the EnOcean data points. They can be exposed to OPC, and can be used as parameters and IEC 61131 variables. Further they can be used in connections, global connections and math objects.

14.5.3 Alarming, Scheduling and Trending

EnOcean data points can be alarmed by the generic alarm server. The configuration of alarming is described in Section 7.13.

EnOcean data points can be scheduled by the generic scheduler. For devices with CEA709 interfaces, the CEA709 scheduler is used to schedule EnOcean networks. This works also when the CEA709 node is no commissioned. For BACnet devices, the EnOcean data points can be also scheduled by a BACnet scheduler. The configuration of scheduling is described in Section 7.12.

EnOcean data points can be trended with the generic trends which are described in Section 7.14. They can also be configured with historic filters as described in Section 7.17.

14.5.4 Teach-In EnOcean Devices

The EnOcean device templates do not contain any specific addressing information. The assignment between a device instance in the data point configuration and a physical device has to be done later on the Commission Web interface. This is called the *teach-in* of EnOcean devices.

To Teach-In EnOcean Device

1. On the Web interface go to the menu **Commission** and select the **EnOcean** technology.

2. The Web UI lists all EnOcean devices found in the configuration. To teach-in an uncommissioned device, click the **Teach-In** button.

3. The device status changed to “Waiting for Device ID”. Then press the button on the EnOcean device that shall be associated with this device in the configuration. Or click **Cancel** to abort.

4. If the teach-in worked fine, the device status will change to “OK”. The RSSI value is updated. Expand the **Profile** item and check the received data value.

UID Interface	Device	Status Timestamp	Device ID	Profiles	RSSI Rep. Hops
108E *0*	Temperature Sensor *Thermokon SR65*	OK *2015-01-12 11:54:27*	0006B35C	⊟ A50214 Temperature 22.61 °C ⊞ Parameters	-57.00 dBm *0*

5. Expand the **Parameters** item and enter a device description and location string. This information will be available in LWEB-900 and on the Web UI.

14.5.5 Organize EnOcean Devices

Once the EnOcean devices have been created, they can be modified using the context menu on the device folder. EnOcean devices can be renamed, moved (but not into other devices) and organized in sub-folders. EnOcean devices can also be duplicated and deleted.

To Organize EnOcean Devices

1. Click on the EnOcean port folder and create a new folder by choosing **New Folder** in the context menu.

2. Select one or more EnOcean device folders and drag them with the mouse onto a sub-folder.

3. To duplicate an existing EnOcean device, select the EnOcean device folder and choose **Duplicate Device** from the context menu.

4. To rename an EnOcean device, select the EnOcean device folder and choose **Rename Device** from the context menu.

Note: *When moving EnOcean device folders to other sub-folders or renaming EnOcean devices, their teach-in data is maintained. When duplicating an existing EnOcean device the duplicated devices require a teach-in.*

5. To To delete EnOcean devices, select one or more EnOcean device folders and choose **Delete Device** from the context menu.

15 OPC Client

15.1 Introduction

LOYTEC devices that support the OPC XML-DA standard as a client can integrate compatible OPC server implementations, for example L-INX or L-DALI devices or an LWEB-900 server running on a PC. The OPC tags of these OPC servers are added as data points to the device. An OPC server is represented by an OPC device in the OPC client. The OPC device address information consists of a URL to the Web service, including the IP address or hostname, port, username/password and secure service option.

The OPC client function in the Configurator can directly integrate OPC tags from other device configurations or import OPC tags lists. These tags lists can also be used to integrate third-party OPC servers. It is also possible, to assign OPC server URLs and username/password tokens later in the commission Web UI.

To support environments, where a single OPC client configuration shall be used on the local network (LAN) and from an external (public) network, a secondary address and port can be specified for an OPC device. This secondary address will be tried by the client in case the server is not reachable via the primary address. This can be used in NATed environments, where different addresses need to be used depending on the location of the client.

15.2 Web UI

15.2.1 Data Points

OPC client data points can be accessed through the Web UI as described in Section 5.3.1. The native info on the data point details page shows the effective tag path on the OPC server and the OPC server timestamp of the tag value.

15.2.2 Commissioning

The **Commission** Web interface provides a page for managing and commissioning OPC devices used by the OPC client technology. This page lists all OPC devices, which have been created in the data point configuration as shown in Figure 311. The list shows device names as created in the configuration. The device **Status** can be one of the following:

- Running for a working OPC device,
- Stopped if the OPC device communication is halted,
- Disabled if the OPC device has been disabled,

- Unreachable if the OPC device is commissioned but cannot be contacted at the moment,
- Auth Error if the user authentication failed on this OPC device,
- Uncommissioned for an OPC device that needs to be commissioned.

To commission an OPC device, enter the IP address of the OPC server (or the entire Web service URL) and the operator password. The default password is ‘operator’. Then click on the save icon. The OPC device stats should change to Stopped and Running, if successful. Otherwise, the status goes to an error state.

Optionally, a **Replacement Path** can be entered. This relocates the tag path of an OPC device to another path. This option is only available, if the OPC tag list has been generated with a path offset. See Section 15.4.2 for more detailed instructions.

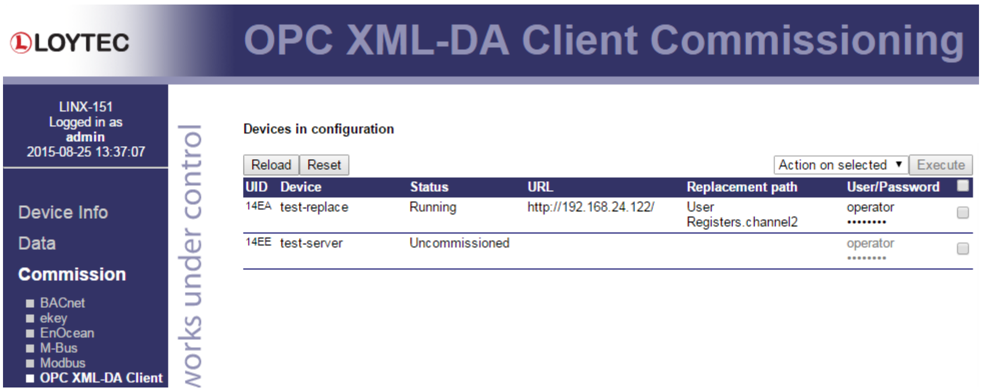


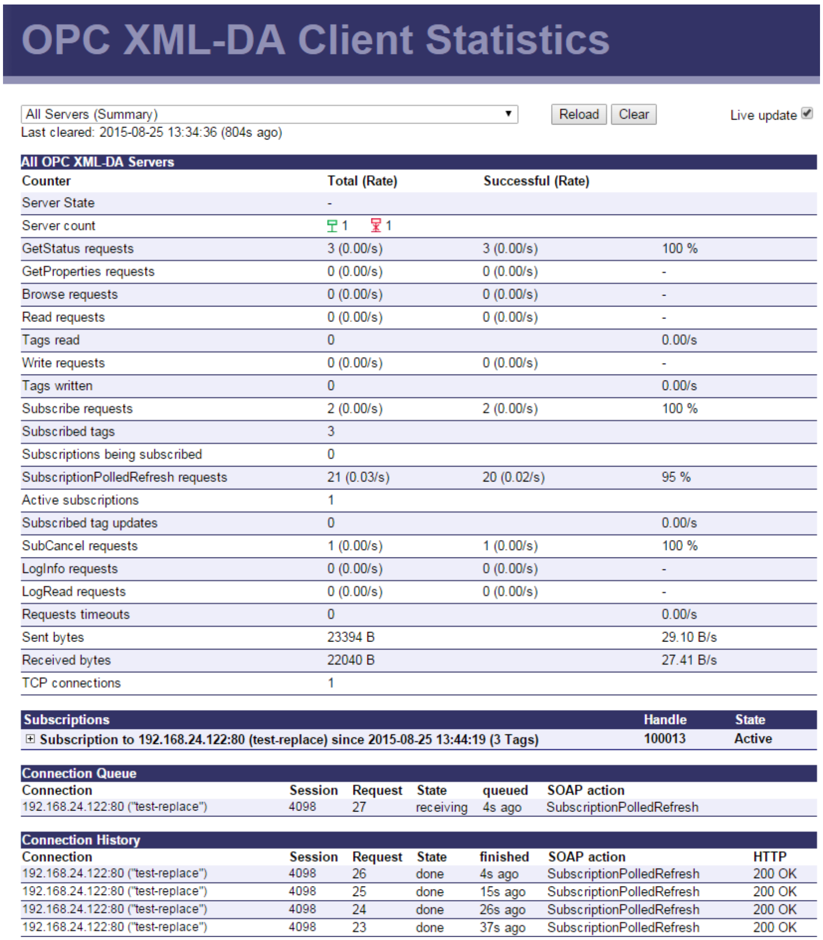
Figure 311: Commission OPC devices

To clear device assignments check one or more check boxes at the end of each line, choose the **Decommission** option in the **Action on selected** drop-down and click **Execute**. The selected devices will then be unassigned and appear as uncommissioned again. Devices can also be temporarily disabled. When disabled, no further data is processed from the respective devices until they are enabled again later.

15.2.3 Statistics

The OPC XML-DA Client Statistics page provides OPC client side communication statistics. For each OPC server defined in the project, there is a separate box of information regarding this server. At the top of this list is an additional box, showing a summary of all servers. An example of this page is shown in Figure 312.

Each box contains a list of statistic items, such as requests sent, OPC tags read or written, subscriptions made, the number of subscribed OPC tags, and so on. For each entry in this list, there are four columns. The first one describes the counted item, the second is the total count since the statistics were cleared the last time, and the last two columns show the number of successful operations, if applicable.



OPC XML-DA Client Statistics

All Servers (Summary) — Reload — Clear — Live update

Last cleared: 2015-08-25 13:34:36 (804s ago)

All OPC XML-DA Servers

Counter	Total (Rate)	Successful (Rate)	
Server State	-		
Server count	1 1		
GetStatus requests	3 (0.00/s)	3 (0.00/s)	100 %
GetProperties requests	0 (0.00/s)	0 (0.00/s)	-
Browse requests	0 (0.00/s)	0 (0.00/s)	-
Read requests	0 (0.00/s)	0 (0.00/s)	-
Tags read	0		0.00/s
Write requests	0 (0.00/s)	0 (0.00/s)	-
Tags written	0		0.00/s
Subscribe requests	2 (0.00/s)	2 (0.00/s)	100 %
Subscribed tags	3		
Subscriptions being subscribed	0		
SubscriptionPolledRefresh requests	21 (0.03/s)	20 (0.02/s)	95 %
Active subscriptions	1		
Subscribed tag updates	0		0.00/s
SubCancel requests	1 (0.00/s)	1 (0.00/s)	100 %
LogInfo requests	0 (0.00/s)	0 (0.00/s)	-
LogRead requests	0 (0.00/s)	0 (0.00/s)	-
Requests timeouts	0		0.00/s
Sent bytes	23394 B		29.10 B/s
Received bytes	22040 B		27.41 B/s
TCP connections	1		

Subscriptions	Handle	State
Subscription to 192.168.24.122:80 (test-replace) since 2015-08-25 13:44:19 (3 Tags)	100013	Active

Connection Queue

Connection	Session	Request	State	queued	SOAP action
192.168.24.122:80 ("test-replace")	4098	27	receiving	4s ago	SubscriptionPolledRefresh

Connection History

Connection	Session	Request	State	finished	SOAP action	HTTP
192.168.24.122:80 ("test-replace")	4098	26	done	4s ago	SubscriptionPolledRefresh	200 OK
192.168.24.122:80 ("test-replace")	4098	25	done	15s ago	SubscriptionPolledRefresh	200 OK
192.168.24.122:80 ("test-replace")	4098	24	done	26s ago	SubscriptionPolledRefresh	200 OK
192.168.24.122:80 ("test-replace")	4098	23	done	37s ago	SubscriptionPolledRefresh	200 OK

Figure 312: OPC Client Statistics Page.

Below a **Subscriptions** list of all active subscriptions for the selected OPC server is shown. Each subscription can be expanded to list the subscribed OPC tags and their last updated values. The **Connection Queue** and **Connection History** shows the next queued request on active connections and the history of completed requests, respectively.

15.3 Configurator

15.3.1 Port Folder

The OPC port folder represents the OPC client interface in the Configurator. In the OPC port folder, one folder per OPC server is created, which contains the data points for that OPC server. OPC device folders are identified by the OPC server folder icon known from LWEB-900. OPC device folders can be deleted, renamed and organized in sub-folders.

Data points and sub-folders in an OPC device folder can be deleted, renamed and re-organized. Data points can also be OPC exposed, which makes them available as OPC tags in the local OPC server. All other properties can be edited as described in Section 7.2.4.

15.3.2 Data Point Properties

Apart from the common data point properties discussed in Section 7.2.4 the data points of the OPC client technology have additional properties.

- **OPC Data Type**: This property shows the OPC data type of the OPC tag on the server.

- **OPC Tag Name**: This property shows the OPC tag name on the OPC server.
- **OPC Tag Path**: This property shows the OPC path to the tag on the OPC server.
- **OPC Device Commission Later**: This property defines whether the OPC device in the OPC client shall be commissioned later on the Web interface.
- **OPC Tag Base Path**: This property shows the OPC tag base path of an imported OPC tag list. This path can be replaced later by another path on the Web UI of the device.
- **OPC Device Name**: This property defines the logical device name used to denote the OPC server instance in the data point configuration. Changing it also updates the OPC device folder name.
- **OPC Device Local uses HTTPS**: This property specifies, whether communication to the OPC server on the local network shall use HTTPS.
- **OPC Device External uses HTTPS**: This property specifies, whether communication to the OPC server from an external (public) local network shall use HTTPS.

15.3.3 OPC Device Manager

To add new OPC servers to the project or update the data points associated with an existing OPC server, the OPC device manager window is used. It can be opened from the context menu of the OPC client port folder. Choose **Manage OPC Devices …** .

On the top right, a tool bar is available to execute various operations.

The buttons from left to right are:

- **Add:** Use this button to add a new OPC server from a configuration file of the device. Supported configuration types include all L-INX, L-GATE, and L-DALI configuration files. Once the device configuration was loaded, suitable OPC client data points will be generated and added to the current project.

- **Refresh Status:** This button refreshes the status of the selected OPC device. It will check if the current data point configuration of the device is still in sync with the data points present in the current project, or if an update is needed.

- **Reload:** This button reloads the device configuration and updates the OPC data points in the current project to match the current configuration of the OPC server. Use this button to update the OPC data points whenever changes were made to the data point configuration of the OPC server.

- **Select Source:** This button allows you to change the device configuration file associated with the selected OPC device. In case the name or location of the configuration file was changed, the OPC device definition must be updated with the new location of the file.

- **Open in Editor:** This button will launch the configuration software suitable for the selected OPC device and load the associated data point configuration.

- **Delete:** This button deletes the selected OPC devices and all associated data points. Note that once the data points have been deleted, there is no easy way to restore them in a way that references from configuration objects to these data points will work again. Even after adding the same device again, the new data points will have different UIDs and all references to the old data points will be broken.

Below the tool bar is a list of all OPC devices. Each of the device entries can be expanded to show a number of properties. These properties can be edited and are explained below:

- **Name:** Internal name for this OPC device. Note that changing the name will also change the data point folder name in which the associated OPC data points are located. In case there are templates or template instances referencing the old data point folder name, they will need adjustment to fit the new name.

- **Local Address / Port:** These properties specify the primary IP address and port number which should be used to contact the OPC server.

- **Local use HTTPS:** When this option is checked, the OPC client will use HTTPS instead of HTTP when contacting the server.

- **Public Address / Port:** These properties define a secondary IP address and port number, which can be tried by the client in case the server is not reachable via the primary address. This can be used in NATed environments, where different addresses need to be used depending on the location of the client.

- **Public use HTTPS:** Same as the option for the local address, but related to the public address.

- **Operator Password:** The operator password to use when contacting the server.

- **Write Aggregation:** Time in milliseconds to wait for more values to write, before a write request is sent to the server. This increases efficiency by reducing the overhead involved in building and transmitting a SOAP request, compared to the number of values written with that request.

- **Lower Limit to Min. Send:** Minimum time in milliseconds for the Min. Send property of the OPC data point. Any value lower than this will be replaced by the specified minimum time.

- **Config Status:** Shows the current status of the server configuration. The configuration is either shown as *up to date*, meaning that the current project is still in-sync with the server configuration, or the status indicates that the server configuration was changed or removed since it was last imported.

- **LWEB Time:** The time when the server configuration was last imported.

- **Source Time:** The time when the server configuration was last modified.

- **Source Path:** The location of the imported server configuration on disk.

15.4 OPC Client Workflow

15.4.1 Integrate Devices via OPC

The Configurator allows adding entire LOYTEC device configurations, which expose data points as OPC tags. The resulting OPC device can be fully configured with its device URL and username/password or created without addressing information to be commissioned later. The imported OPC tags can be edited and trimmed to the required number of tags.

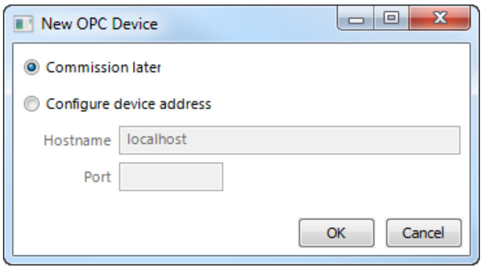
To Add an OPC Server

1. Select the **OPC Client** port folder.

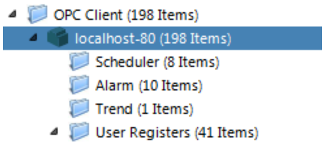
2. Right-click and choose **New OPC Device …** from the context menu.

3. Select a LOYTEC device configuration file such as '.linx' or '.dali'.

4. In the **New OPC Device** dialog select **Commission later** or enter an IP address or host name and HTTP port for the Web service.



5. A new OPC device folder will be created.



6. Right-click on the OPC device folder and choose **Rename device …** from the context menu. Enter the desired OPC device name. This name will be shown on the OPC device commission Web page (see Section 15.2.2).

15.4.2 Integrate Sub-Trees and Relocate

Apart from integrating entire OPC servers it is also possible to import only a sub-tree. This sub-tree has a base path that can be later replaced by another path. This makes it possible to create tag trees on the OPC server that are identical and cover different areas, such as 'channel1' and 'channel2'. When importing such a sub-tree, the assignment to channel 1 or 2 can be made later on the OPC commission Web page.

For example, the tag tree 'User Registers.channel1' contains the OPC tags 'reg1' to 'reg3'. The tag tree 'User Registers.channel2' contains exactly the same tags, that represent channel 2. The tag tree 'User Registers.channel1' can be exported and imported into the OPC client. Later on the Web UI, the base path 'User Registers.channel1' can be replaced by 'User Registers.channel2'.

To Export a Tag Tree

1. Select a data point folder in a device configuration which has the desired OPC tags.

2. Right-click and choose **Export OPC Tags …** . Store them in a file ending '.opc'.

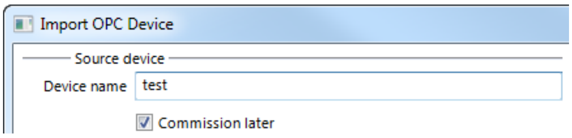
To Import a Tag Tree

1. Select the **OPC Client** port folder.

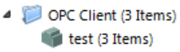
2. Right-click and choose **New OPC Device …** from the context menu.

3. Select a tag list file ending '.opc'.

4. In the **Import OPC Device** dialog, edit the device name and check **Commission later**.



5. Then click the **Create New** button. This creates a new OPC device folder for exactly that OPC tag tree.



To Relocate a Tag Tree

1. On the Web UI go to **Commission → OPC XML DA Client**.

2. Enter the IP address in the **URL** column and click the save icon.

3. The column **Replacement Path** shows the original OPC base path of the tag import.

4. To relocate it, enter a replacement path, e.g. 'User Registers.channel2'. Click the save icon.

16 ekey

16.1 Introduction

The ekey function allows adding fingerprint readers to LOYTEC devices and building applications where access restrictions using biometric measures are required. ekey fingerprint readers must be purchased separately and can be attached to the RS-485 port. Up to 16 fingerprint readers are supported in a bus topology. Communication to reader devices on the RS-485 bus is encrypted.

Figure 313: Example ekey fingerprint reader device

The reader devices are operated in active mode. In this mode each device is required to be polled in a timely fashion in order to be online. Reader devices indicate whether they are online as shown in Figure 313. The following color codes are in use:

- Blue permanent: The reader is online.
- Orange flashing: The reader is offline and cannot be contacted by the LOYTEC device.
- Orange permanent: The reader is waiting for a finger to be enrolled.
- Green flash on: A user has been authenticated.
- Red flash on: A user has been rejected.

The primary function of a fingerprint reader device it to authenticate a user, who has been enrolled for that device. One or more fingerprints can be used for the enrollment. The user is identified by a user string.

LOYTEC products that have the ekey function can be used for:

- Building access control using biometric measures (fingerprints),
- Creating users and enrolling fingerprints,
- Distributing users over several fingerprint sensors,
- Controlling data points each time a user is authenticated.

16.2 Web UI

16.2.1 Data Points

The ekey data points can be accessed through the Web UI as described in Section 5.3.1.

16.2.2 Commissioning

The **Commission** Web interface provides a page for managing and commissioning ekey reader devices. It allows scanning for devices and assigning them to reader device instances in the data point configuration. An example is shown in Figure 314.

The **Devices in configuration** section lists the ekey reader instances found in the data point configuration. Each line allows editing the device settings and user configuration on that device by clicking the icons at the end of the line (see also Section 16.4.3). Edit the device description below the device name in the **Device** column. The **Status** column shows the reader state. It can be one of the following:

- OK for a working reader device,
- Uncommissioned for an unassigned reader device,
- Disabled for an ekey device that has been disabled,
- Offline for a reader that does not respond within 6 seconds,
- Busy for a reader with an ongoing data transfer,
- Encryption error for a reader that uses different encryption keys than the host device,
- Error if any other error occurred, e.g., the commissioning data has been corrupted.

The **Data** column shows enrolled users and fingers on the reader devices. **Encryption** is indicated if active.

Figure 314: ekey Commissioning Web page.

To clear reader device assignments check one or more check boxes at the end of each line, choose the **Decommission** option in the **Action on selected** drop-down and click **Execute**. The selected devices will then be unassigned and appear as uncommissioned again. Devices can also be temporarily disabled. When disabled, no further data is processed from the respective devices until they are enabled again later.

By clicking the **Scan ekey network** button a device scan can be started. The scan searches for connected reader devices and puts them in the scanned devices list. For devices found that have not yet been assigned to devices in the data point configuration, the user can select a reader device in a drop-down box and click on the **Assign** button.

16.3 Configurator

16.3.1 Activating ekey

Before ekey fingerprint reader devices can be added to a data point configuration, the ekey interface in the project settings must be enabled. The project settings are described in detail in Section 7.3.

To Activate ekey in the Configuration

1. Open the project settings dialog.

2. In the **System Settings** tab enable the ekey protocol on an RS-485 port, as shown in Figure 315.

3. Press the **OK** button.

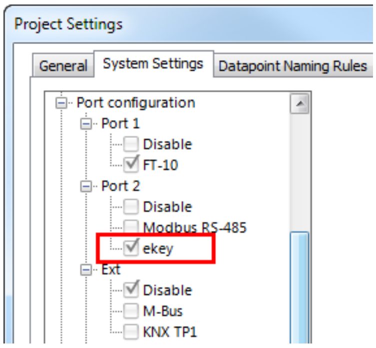


Figure 315: Enabling ekey in the Project Settings Dialog.

Important: ***If the ekey protocol is deactivated via the checkbox or a firmware or model version is chosen, which does not support ekey, the entire ekey configuration is deleted. In this case a dialog is displayed, which has to be confirmed.***

If the Configurator is connected to a device, the **Download** button can be used to download the device configuration.

16.3.2 Data Point Manager for ekey

The Configurator uses a central concept to manage data points. The data point manager is used to select, create, edit and delete data points. The dialog is divided into three sections:

- The folder list (Figure 316),
- The data point list (Figure 317),
- And a property view.

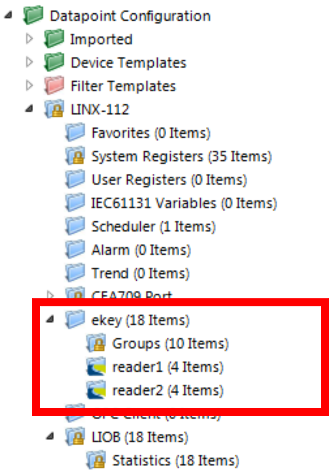


Figure 316: Data Point Manager with ekey device folders.

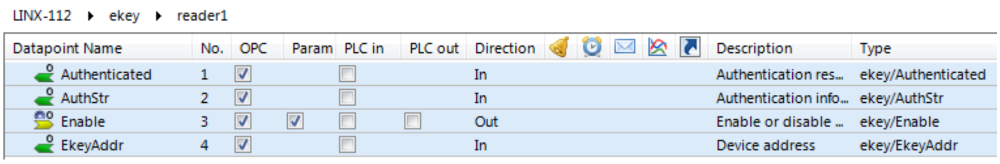


Figure 317: Data Point Manager Dialog with ekey Data Point List.

16.3.3 Port Folder

The ekey port folder represents the ekey interface. In the ekey port folder, one folder per ekey fingerprint reader device is created, which contains the data points for that reader. The ekey folders are identified by a small ekey logo on the folder icon . They can be deleted, duplicated, renamed and organized in sub-folders.

The data points in an ekey device folder cannot be deleted or renamed. Some of their properties such as OPC exposure, PLC in/out, parameter can be modified. The data points per fingerprint reader are:

- Authenticated (binary): This register becomes TRUE for a short period of time when a finger was authenticated and then falls back to FALSE.
- AuthStr (string): When a finger is authenticated, it contains the full authentication information of user and fingerprint. After that it falls back to the empty string.
- Authenticated user ID (analog): When a finger is authenticated, it contains the user ID. After that it falls back to invalid value.
- Authenticated user name (string): When a finger is authenticated, it contains the user name. After that it falls back to the empty string.
- Enable (binary): When this register is set TRUE, the finger reader is enabled and performs authentication of fingers. When setting FALSE this particular fingerprint reader device is disabled. This is a parameter by default and thus available in LWEB-900.
- EkeyAddr (string): This data point contains the ekey device address.
- Users (string): This parameter data point contains the user and finger enrollment data stored on that reader device. The parameter data point is available in LWEB-900 and can be used to distribute user enrollment over reader devices.

Apart from the ekey fingerprint reader device folders, a **Groups** folder exists. This is a fixed folder and contains 10 group enable data points:

- groupEnable_X (X=0, …, 9, binary): Each of the 10 group enable registers corresponds to a user group. Setting it FALSE temporarily denies access to all members of the corresponding user group. Setting it TRUE allows access to all members of the corresponding user group. This is a parameter by default and thus available in LWEB-900.

16.4 ekey Workflow

16.4.1 Create From Device Templates

The ekey fingerprint reader devices need to be created using device templates. The Configurator is distributed with a library of ekey fingerprint reader device templates. An ekey device created in the data point configuration needs to be commissioned later online on the Web UI.

To Create an ekey Reader Device

1. Click on the **ekey** port folder.
2. Right-click in the data point list view and select **New ekey Device…** in the context menu.

3. In the **Create Device** dialog enter a **Device Name** and a **Count** of devices to be created as shown in Figure 318.

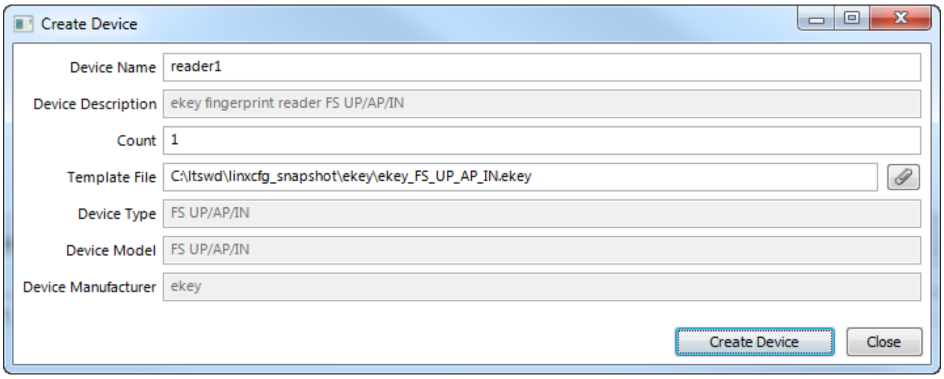


Figure 318: ekey fingerprint reader device creation dialog

4. Then click on and choose an '.ekey' device template file. As a default the directory of the distributed ekey device templates is opened. Choose the desired template file.

5. The device type, device model and device manufacturer information is displayed. Then click **Create Device**.

16.4.2 Enroll Fingerprint Readers

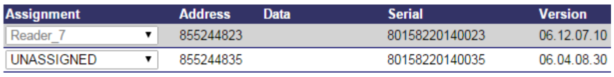
The fingerprint reader devices do not contain any specific addressing information. The assignment between a reader instance in the data point configuration and a physical reader device has to be done later on the Commission Web interface.

To Enroll a Reader Device

1. On the Web interface go to the menu **Commission** and select the **ekey** technology.

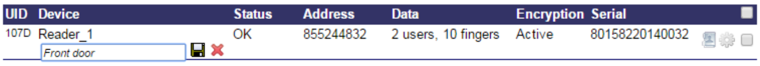
2. The Web UI lists all ekey reader devices found in the configuration. To enroll an uncommissioned device, which is connected and online, click the **Scan ekey network** button.

3. When the scan completes it lists all found ekey devices. Those which are unassigned have a drop-down box.



4. Choose an uncommissioned reader device from the configuration and click on the **Assign** button.

5. The assigned reader should now go online and show address and serial number in the device list. Edit a description and click the save icon.



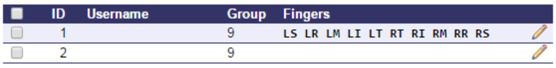
16.4.3 Enroll Users and Fingers

To enroll fingerprints for a user locate a reader device where the user has physical access. The enrollment process is performed on that reader device. The enrollment data can then be transferred to other reader devices that shall grant access to that user as well.

To Enroll User Fingers

1. On the commission Web UI locate the desired reader in the device list and click on the **Show users** icon.

2. The user list of the selected reader device is displayed. Click on the plus icon to add a new user.



3. To edit the user click the pencil icon which opens the **User configuration** page as shown in Figure 319. Enter a user name and select a finger to enroll, e.g. left index finger.

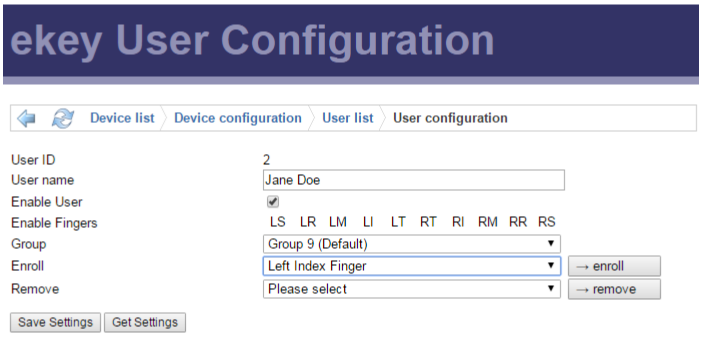


Figure 319: ekey User Configuration page.

4. Then click on the **enroll** button and move the correct finger as described in the instructions.

5. Repeat this enrollment with other fingers as well. At least two fingers should be enrolled. To finalize click on **Save Settings**.

To Transfer a User Configuration

1. In the **ekey User Configuration** page of a given user, click on the **Export link** to generate a user configuration XML file.

2. Then navigate to a different reader device and pull up the **User List** of that device.

3. Add a new user as described above. On the user configuration page of that new user, click on the **Choose File** button. Select the previously stored XML file.

4. Then click the **Import** button.

17 IEC 61131

To design the IEC61131 program which should run on the L-INX device, the graphical programming tool logiCAD is required. The tool allows creating IEC61131 programs using various IEC61131 programming languages. It offers additional features downloading and debugging of the created program.

In addition to logiCAD, the L-INX Configurator necessary to create an appropriate data point configuration in the automation server. The usage of logiCAD itself is beyond the scope of this manual. Please refer to the logiCAD online help in case of additional questions.

17.1 Overview

The PLC in the L-INX device is intended to perform IEC61131 programs operating on IEC61131 data points. The operating principle is to connect IEC61131 data points to data points derived from CEA-709, BACnet or LIOB network. Figure 320 depicts the usage of data points for IEC61131 programs at the example of using CEA-709 network variables.

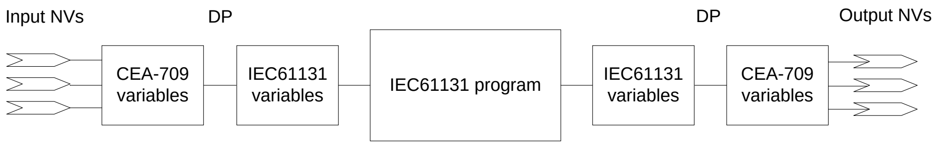


Figure 320 Usage of data points.

Alternatively, data points can also be directly exposed as IEC61131 variables. The data points provide a PLC check box for this purpose. In this use case, not special IEC61131 data points are created and no connections are necessary.

17.2 Installing logiCAD

For developing IEC61131 programs with logiCAD the following components must be installed:

1. L-logiCAD setup package. This package installs the logiCAD software, which is needed to design PLC programs for the L-INX device.

2. L-INX Configurator. This software is required to configure the L-INX device to provide the necessary data points to the PLC and integrate the device into the network.

3. logiCAD license for using logiCAD on the PC. The license is available as a softlock version or as a hardlock version with a USB dongle. On virtual machines it is mandatory to use the hardlock license. How to obtain and install the license is described in this Section.

The L-logiCAD installer installs the IEC61131 programming environment logiCAD and all L-INX related software packages. These packages include the template project for the LINX, the required software to build IEC61131 programs for the L-INX device, and required extensions to interface to CEA-709 networks. Follow the instructions of the installer to install logiCAD for the L-INX.

The language for the logiCAD user interfaces can be set to German or English using the administration folder of the logiCAD control center. The logiCAD control center can be started from the Windows start menu.

17.2.1 Softlock License

The license to run a copy of logiCAD on any PC is based on a softlock license. In case the original softlock license file was generated for an older L-logiCAD version, the software will request an update signature key for the new L-logiCAD installation. This signature key can be found in the file 'logicad_Readme.txt' that is located in the installation directory of the L-logiCAD software.

To obtain and install a new license start logiCAD. The product activation dialog will appear as shown in Figure 321. In this dialog press the button **Download Softlock License.**

Important! ***When using Windows7 start logiCAD as administrator in order to allow the script read the computer number code. If there are x's in the code, the code was not read by the script.***

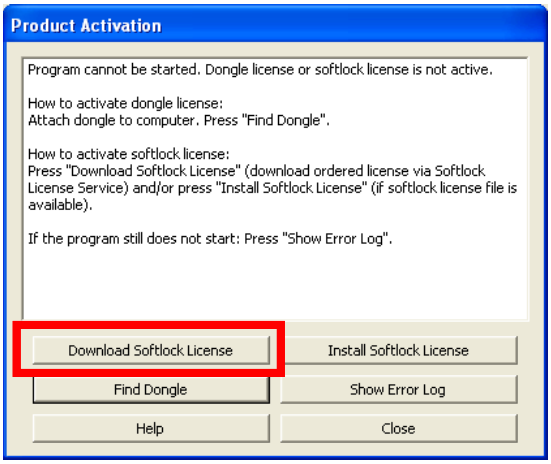


Figure 321: logiCAD Product Activation

Enter the license data provided with the product information on the Web site form as shown in Figure 322. Fill in the **SL-number** and **Licensing Code** from your L-LOGICAD registration form and click on **Download License File**. Save the license file on your computer or have it e-mailed to you.

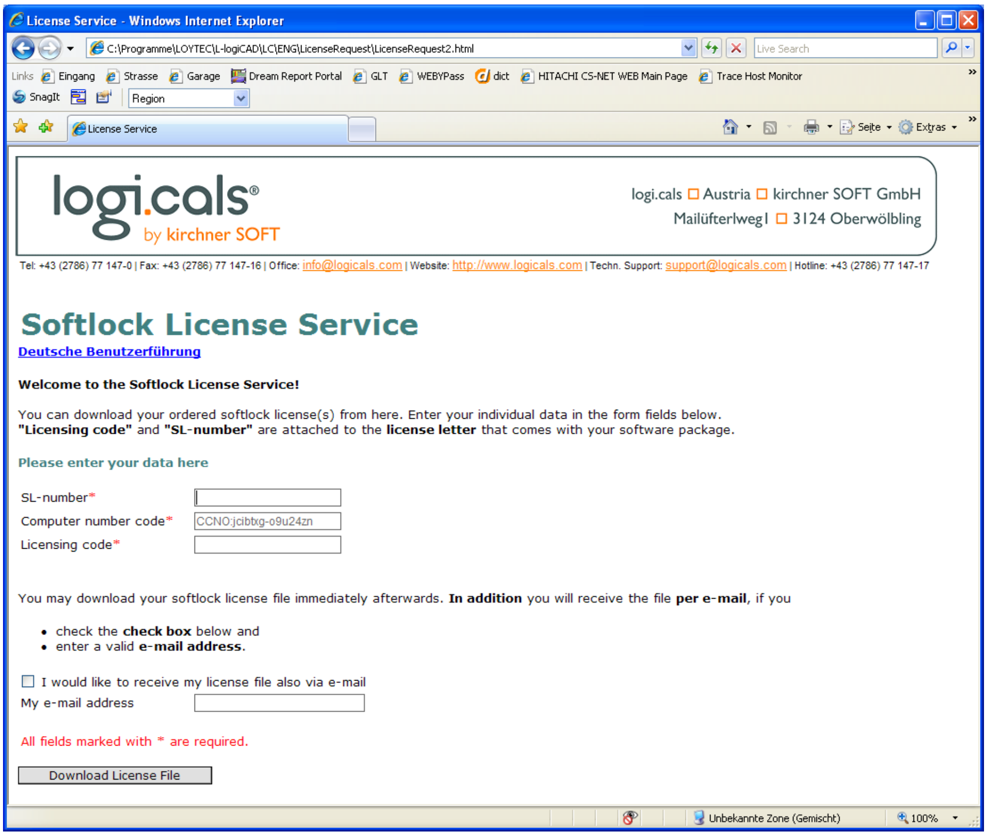


Figure 322: logiCAD softlock license Web form.

Then install the license using the **Install Softlock License** button in the product activation dialog as indicated in Figure 323.

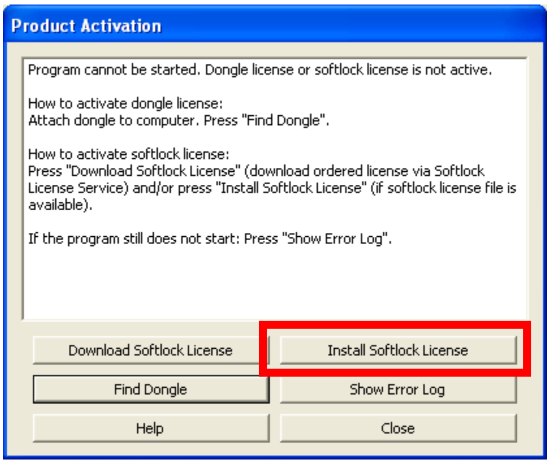


Figure 323: logiCAD softlock installation.

A file requestor dialog opens. Locate the downloaded license file (logicad.lf) and click **Open**. Finally click **Close** in the product activation dialog and start logiCAD again. The softlock license is now activated.

17.2.2 Hardlock License

The hardlock license is needed to run a copy of logiCAD in a virtual machine on the PC. It needs to be puchased separately as L-LODICAD-USB and is distributed as a USB hardlock key of type ‘CodeMeter’. If no driver for this type of hardlock is already installed on the

PC, install the hardlock driver provided on the LOYTEC website. It can be found in Support → Download and the product selection for L-LOGICAD. After the driver installation has completed, plug in the USB key. It must be recognized as a mass storage device (the details on it show the location ‘CodeMeter-Stick’). When starting logiCAD it recognizes the hardlock automatically.

If the driver installation failed, or the USB key was not detected or connected, the window shown in Figure 324 will be shown when logiCAD is started. In this case plug in the hardlock stick, wait for Windows to recognize it and then click **Find Dongle**.

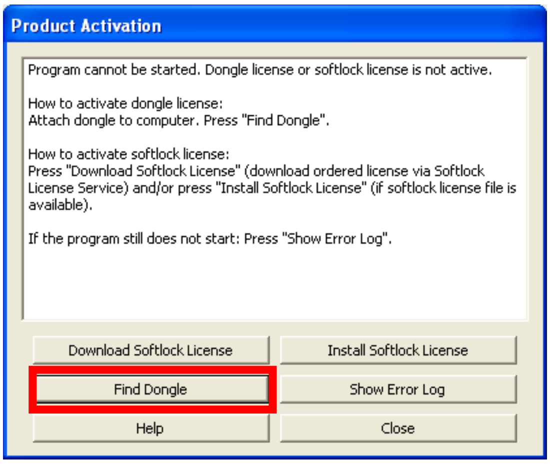


Figure 324: logiCAD Product Activation with Hardlock

Note: *logiCAD checks frequently if the USB key is present. When unplugging the USB key, even if logiCAD was successfully started, all major features are automatically disabled. But there is no additional message for the user!*

17.3 IEC61131 Project Files

In the L-INX Configurator select the **LogiCAD Files** tab to attach an IEC61131 program and a logiCAD project to the L-INX project. The tab is shown in Figure 325.

If there is an IEC61131 program attached to the project, every time a new configuration is downloaded to the device, the L-INX Configurator asks, if the attached program should be downloaded as well. This way, no logiCAD is required to download a suitable IEC61131 program in a separate step. The project designed in the L-INX Configurator can hold all necessary information to set up a running device:

- IEC61131 data point configuration,
- Data point configuration in the LINX,
- Required connections,
- IEC61131 program.

The logiCAD project directory can also be attached to the L-INX configuration file, in order to include the logiCAD project sources from which the program was compiled.

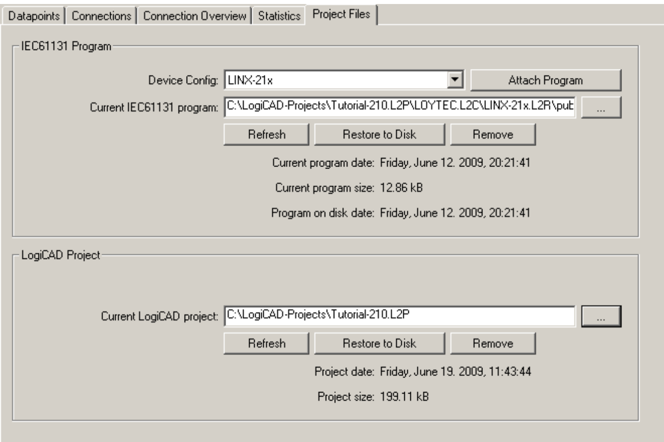


Figure 325: IEC61131 Project Files.

To attach an IEC61131 program or a logiCAD project select the file/folder to attach. The selected data will be attached automatically the project. It can be restored to disk by pressing the **Restore to Disk** button. During the development process the attached data may change several times. To update the attached data, located on the before provided path, press the **Refresh** button.

Every time a logiCAD project is successfully compiled, the file MBRTCode.so, the compiled IEC61131 program, is copied to the *public* directory of the device resource for which the program was compiled. Select this file to attach it to the project of the Configurator. Note that the time and date of the file indicates the time of the last code generation. If logiCAD is not able to build a new program, the old file will not be deleted.

The project will be scanned for device resources and the available devices will be listed in the dropdown box called **Device Config**. Select the desired device and press the button **Attach Program** to automatically attach the correct MBRTCode.so file.

17.4 Working with logiCAD

This section provides all necessary background information how to use the L-INX with logiCAD. For using the L-INX with logiCAD, a predefined project template for the L-INX must be used. Hence, when creating a new project, select the project template “Project for LINX-110”, see Figure 326. For additional information how to create, delete and manage projects please refer to the logiCAD online help.

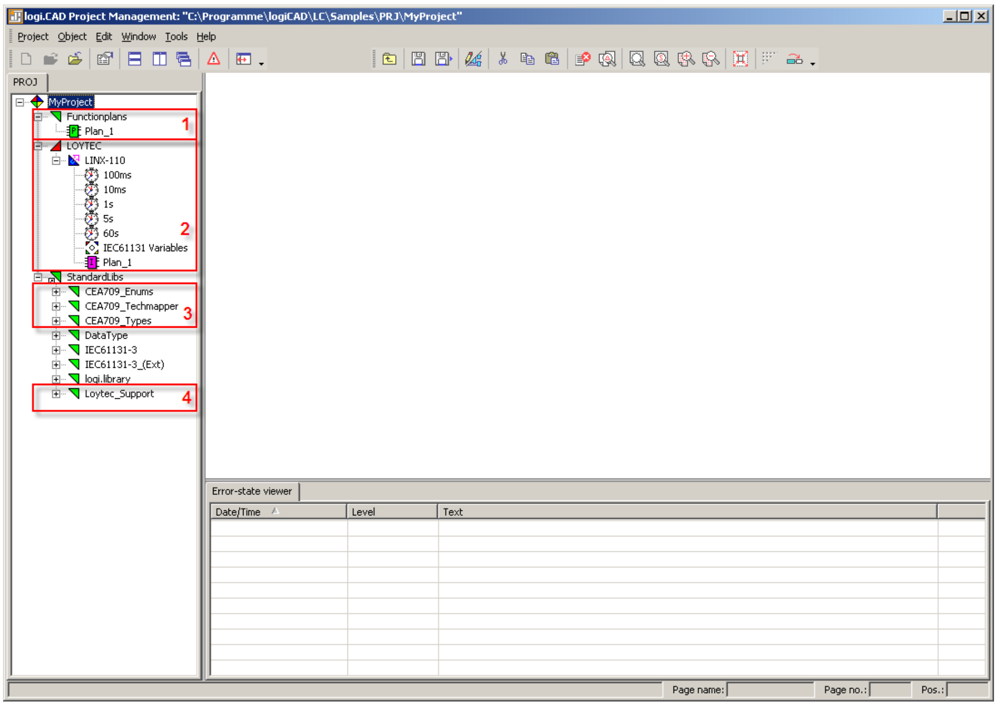


Figure 326: L-INX specific extensions

Figure 326 shows the standard project for the L-INX including all LINX-specific extensions for logiCAD. It shows the structure window showing the project structure on the left, an empty working area top right and the **Error-state viewer** on the bottom right side.

The structure window offers interfaces to the following features:

1. The folder **Functionplans** holds all the program types created within logiCAD. ‘Plan_1’ is the default plan to start with.

2. The folder **LOYTEC/LINX-110** represents the LINX-110 device. The folder **LOYTEC** represents a configuration containing one LINX-110 resource; please refer to the logiCAD online help for details. To run a program, located in the folder **Functionplans**, a program instance of the corresponding function plan must be created. In the standard LINX-110 template a program instance of the ‘Plan_1’ is already defined. To be able to transfer IEC61131 variables from the L-INX device to the IEC61131 program a global variables object within the **LINX-110** folder is required, see Section 17.4.1 for details.

3. LogiCAD operates on variable types standardized in the IEC61131 standard. Look for “Elementary and Generic Data Types” within the logiCAD online help to get information about the available data types. For those L-INX devices, which are intended to operate on structured NVs, appropriate type definitions for the NVs are required. These definitions are inside the folder **CEA-709_Types**. Additional NVs must be converted to data types that can be processed by logiCAD. Therefore *Technology Converters* are supplied with the L-INX project, which perform this conversion. See Section 17.4.3 for details.

4. For designing programs that support the force update functionality (see Section 17.6.1), or designing user-defined *Technology Mapper* (see Section 17.6.2) additional function blocks are required. These blocks are located within the **Loytec_Support** directory.

All LINX-specific add-ons are provided using function blocks. Hence, in the following, all samples are based on designs using function blocks.

17.4.1 Managing Variables

On a function plan, three basic types of variables may be created using the tabs shown below the function plan sheet:

- **VAR:** Variables created on this tab will be visible only to the logic designed on this group of sheets. It will not be accessible to any other programs or to any function blocks which are used in this program. It is similar to a 'static' variable declaration in a C code function.

- **VAR GLOBAL:** Variables declared here will be accessible to the entire program, including any function blocks which are called by the program. Function blocks which need to reference this variable need to have a suitable declaration of an external variable (see next point). This declaration is similar to a 'static' declaration of a C variable outside a function, which will be visible to all functions inside the C code module, but not visible to other modules.

- **VAR EXTERNAL:** Variables declared in this list will be treated as open references to a global variable which exists somewhere in the scope of the device on which the program will be executed. This means that a global variable needs to be declared on the device resource, which will be available to all programs running on the device. If the physical address parameter of the variable is set to %I, %O, or %M, the variable will be handed down to the I/O driver of the L-INX for processing. If a suitable IEC61131 variable exists in the data point configuration of the device, its value will be forwarded by the I/O driver between the PLC variable and the data point. If no physical address is set, the variable will only be visible to the PLC programs but not to the I/O driver, which may be used to exchange data between PLC tasks.

The basic data flow between the CEA709 network (or BACnet and other technologies) and the PLC program is depicted in Figure 327.

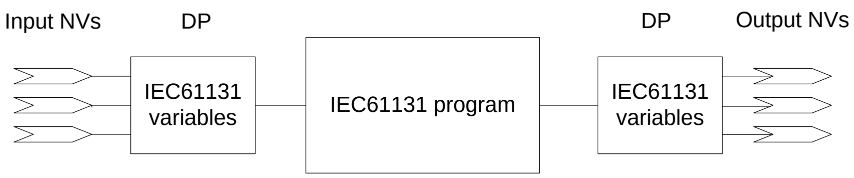


Figure 327: Connecting IEC61131 variables

The place where global variables are created on the device is shown in Figure 328.

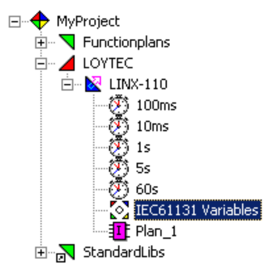


Figure 328: Global Variables Object

When starting a new project, no global variables object is available; it must be created before compiling the IEC61131 application. The global variables object can be created automatically or manually, depending on the different workflows, please refer to Section 7.5.2.

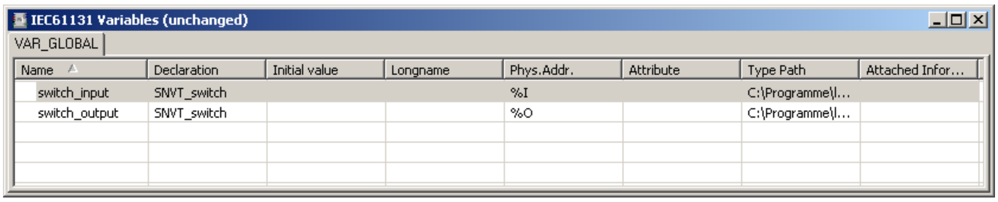


Figure 329: Variables defined in global variables object

Figure 329 shows the contents of a sample global variables object. As shown a global variable is defined by the fields **name**, **declaration** and **phys.addr.**:

- **Name:** The name of a global variable must be unique. The name is used from the IO driver to identify the global variable and from the L-INX Configurator to generate the corresponding IEC61131 data points.

- **Declaration:** Here the type of the global variable is defined.

- **Phys.Addr.:** The IO driver needs to know the data flow direction to be able to update variables. The direction is defined by adding %I for an input variable, %O for an output variable and %M for a marker (input and output). If the address is empty, the I/O driver will not handle the variable, but it may still be used by the tasks running on the device.

Important: ***Only ASCII characters can be used for naming the global variables.***

17.4.2 Build and Download the IEC61131 Program

IEC61131 programs, designed using logiCAD, must be cross compiled in order to run on the L-INX device. The prerequisite to compile an IEC61131 program are as follows:

- A program instance with associated program type

- A corresponding global variables object

Right click on the LINX-110 resource, see Figure 328, and select Code Generation. Please refer to the logiCAD online help for the meaning of the options. Take care about the option breakpoint support, see section 17.4.5 for details.

After successfully finished code generation, the IEC61131 application can be downloaded to the LINX-110. Right click on the LINX-110 resource and select download.

The IEC61131 program can be downloaded to the LINX-110 device via TCP/IP, CEA-709 and RS232:

- **TCP/IP:** Enter the IP address of the LINX-110 device. Do not change the default communication port (2048). This is the easiest and fastest way to connect to the device.

- **CEA-709**: Select the network interface to use and fill out all other fields, see Figure 330. Alternatively, select the network interface and press Auto-detect via Service-Pin. Then press the service pin on the LINX-110 device. Note that this connection method requires an installed LOYTEC network interface.

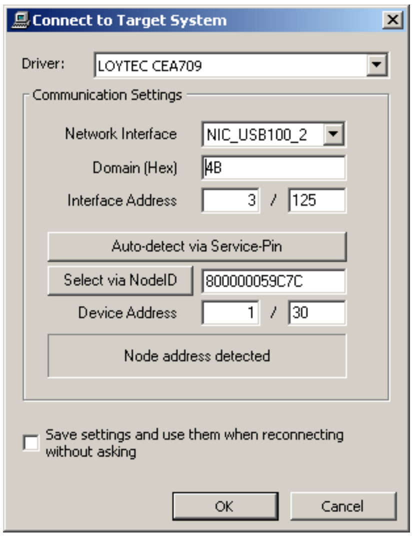


Figure 330: Connect via CEA-709

Important:	***To be able to communicate with the LINX-110 device via CEA709 the LINX-110 must be commissioned.***

- **RS232:** To connect via RS232 select LOYTEC RS232 as Driver and LOYTEC as transmission protocol. All other values are set correctly per default.

17.4.3 Usage of NVs, Technology Converters

To used CEA-709 variables the content of the NVs must be converted to IEC61131 compliant data types. Look for “Elementary and Generic Data Types” within the logiCAD online help to get information about the available data types. Technology Converters are used to perform the transformation from CEA-709 data types to IEC61131 data types. All Technology Converters are summarized in the subfolder CEA709_Conv located in the StandardLibs folder.

Depending on the type of the NV there are three different ways to use the NV within IEC61131 programs:

- Simple NVs that hold only one scalar value, e.g. SNVT_amp:

  Those kinds on NVs are represented as IEC61131 REAL values within logiCAD. There is no additional conversion necessary. Figure 342 shows an example program for scalar data types.

- Simple NVs based on an enumeration, e.g. SNVT_date_day:

  The active identifier of the enumeration is represented as Boolean value. When using NVs based on enumerations, Enumeration Converters are used to identify the current state. There are two kinds of Enumeration Converters. First, the Enumeration Converters that convert the enumeration types to Boolean types, grouped in the folder **Convert from CEA709_Enums**. Second, the Enumeration Converters that converts several Boolean inputs to an enumeration type, grouped in the folder **Convert to CEA709_Enums**.

- Structured NVs that consists of a number of fields, e.g. SNVT_switch:

  On structured NVs two tasks must be performed by the Technology Converters. First the structure of the NV is mapped to IEC61131 conform data types. Second, if necessary scaling factors are applied. Similar to the Enumeration Converters, the

Technology Converters are split up into two subfolders. The first one, which converts the NV into IEC61131 compliant data types, is located in the folder **From_CEA709_Types**. Technology Converters to set up NVs based on IEC61131 data types are grouped in the folder **To_CEA709_Types**.

Figure 331 shows the three possibilities how to use NVs within an IEC61131 program.

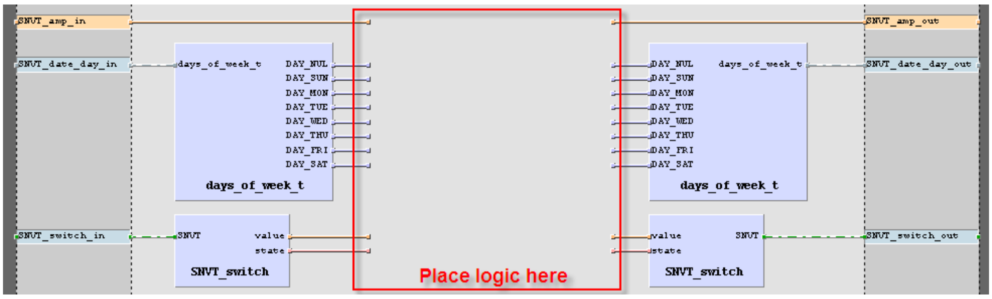


Figure 331: Usage of NVs

If a structured NV comprises enumerations types, these enumerations are not split up by the Technology Converter. To get the value of the enumeration, connect an Enumeration Converter to the corresponding output of the Technology Converter.

For every Technology Converter and Enumeration Converter an online help window, displaying the interface description is available. Select the Technology Converter and press F1 to get the interface description.

17.4.4 IEC61131 Program Cycle Time

IEC61131 programs are performed in a periodical manner. IEC61131 tasks are used do control the execution of an IEC61131 program. As shown in Figure 328 several default tasks are defined within the template project. Right click on the clock symbol and select properties to change the cycle time of the task.

As described in section 17.4 a program instance is required to execute the IEC61131 program. The cycle time of the IEC61131 program is controlled by the task assigned to the program instance. In order to change the cycle time right click on the program instance and select properties. In the upcoming window the task assignment for the selected program type can be changed, see Figure 332.

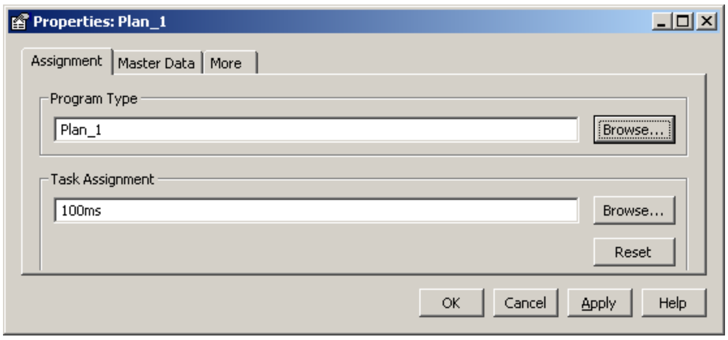


Figure 332: Task assignment

Please take care when defining names for the tasks. The names displayed in the project view are symbolic names, they do not correspond to the settings for the cycle time, even if the L-INX template project use the configured cycle time as task name.

17.4.5 CPU Overload

Several conditions affect the CPU utilization of the IEC61131 program. As a result it is not possible to predict the system load caused by the IEC61131 program. E.g. the following parameters are of particular importance when designing IEC61131 programs:

- Number of inputs and outputs handled by the I/O driver.
- Complexity of the logic in the running IEC61131 program.
- Number of simultaneously running program instances on one L-INX.
- Cycle time of IEC61131 programs.
- logiCAD breakpoint support and force-able code enabled or disabled

The developer of the IEC61131 program is able to check the current system load within the Web UI, see Section 5.1. To get a rough estimate of the CPU load investigate the PLC-LED. Every time, the system load increases 80% for a certain period of time, the PLC-LED switches to red until the system load goes below 80%.

In case of CPU overload, the IEC61131 program may not be able to finish its work within the defined cycle time. Adapt the program in order to reduce the total system load below 80%. Here are a few tips to keep the CPU load down:

- Increase the cycle time, so that the task may finish in time before the next cycle start is scheduled. The PLC kernel will always schedule the next run at an absolute time, no matter how long the previous run took, in order to compensate for irregular execution times and keep a steady cycle time if possible.
- Reduce the number of I/O variables, to reduce the load caused by exchanging data between the PLC program and the data points of the automation server.
- Reduce the number of independent tasks and try to place as much functionality as possible into one task. Every running task will call the I/O driver for new inputs and outputs independently, therefore two tasks running at a 1s cycle time each will cause twice the I/O load of one task running at 1s cycle time.
- Take special care about the complexity of function blocks which are used a lot. Bad performance of one such block may dramatically increase CPU load if it needs to be calculated several hundred times in one cycle.
- Try to disable breakpoint support and force-able code when generating code for the target, to get the most efficient PLC code out of your logic.
- For complex designs, it may be possible to add a state machine using SFC elements and enable/disable large parts of the logic based on the current state of operation.
- Whenever a function does not need to calculate new output values under certain conditions, use the built-in EN input of the block to disable execution and thus reduce the required CPU time, instead of adding your own ‘Enable’ input which causes the logic to ‘behave’ as if it would be disabled, while it is actually calculated every cycle. This is similar to power saving methods used in modern electronic devices. Parts which are not required are put into a low power mode instead of keeping them running in an unproductive state.

17.4.6 I/O Driver Settings

Before starting the IEC61131 program, the device performs a check on all defined global variables, which are marked %I, %O, or %M in the physical address parameter. For all variables, which cannot be loaded, because the corresponding data points cannot be located on the device, the I/O driver reports a warning in the system log and shows a listing on the Web interface (see Section 5.2.25).

Since IEC61131 program and the data point configuration are downloaded separately, it may be possible that the IEC61131 program does not match the currently active data point interface. In this case it could be dangerous to write values onto those potentially wrong data points. The fact that there are any variables that could not be loaded is taken as an indication for a configuration mismatch. The **I/O Check** feature disables the I/O driver of the IEC61131 kernel automatically in this situation. This setting is enabled by default in a new configuration. In the rare case that it is intended to have missing data points, this feature can be disabled in the Configurator. In the system settings de-select the I/O check (see Figure 333). If the I/O driver is disabled, it can be temporarily enabled again on the Web interface until the next reboot.

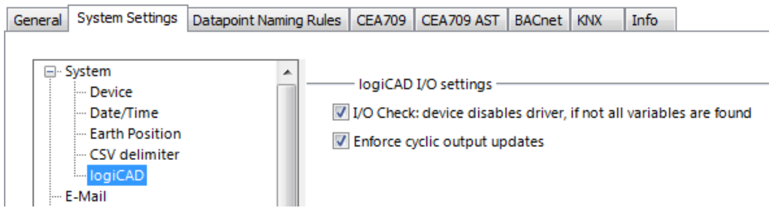


Figure 333: System settings for logiCAD I/O settings.

The project setting **Enforce cyclic output update** enables a cyclic update of output data points, which will ensure that those output data points will contain the calculated value after each cycle. This is the default setting. It can be disabled, if the output data point shall be written only if the calculated value changes. In this mode the output data point can be modified over the Web UI for debugging reasons until the IEC61131 program calculates a new value. This mode can also be used to implement an event-style output operation.

17.4.7 PLC Conflicts

PLC output variables are cyclically updating the respective data points, which are configured as PLC out. If those data points are also written to by other objects (e.g. output of a math object, receiver in a connection) this will not have the desired effect. Also favorites that are PLC out and linked to a PLC out data point will result in two different PLC output variables writing to one and the same data point.

The **PLC Conflicts** tab provides information for detecting such write conflicts on PLC out data points. The tab shows a list of PLC write conflicts and the writing objects, which are in conflict with a PLC out data point. Each reported conflict line can be selected. By clicking the **Go to data point** button the Configurator navigates to the conflicting object.

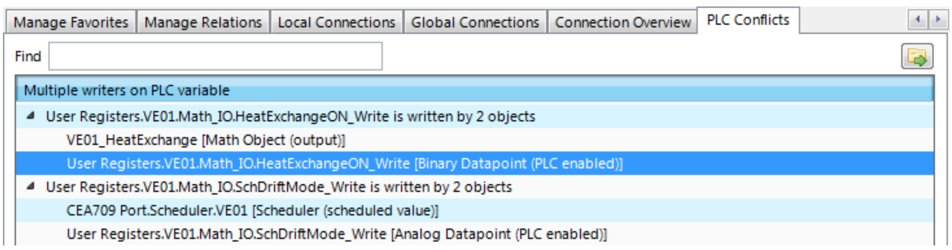


Figure 334: PLC conflicts tab.

An example is shown in Figure 334. The user register 'HeatExchangeON_Write' is written to by the PLC, because it is PLC out. But the register is also an output in the math object 'VE01_HeatExchange'. To resolve the conflict, either remove the register from the math object or remove the PLC out check box from the user register. For doing the latter, select the conflict line reporting the register and select **Go to data point**. Then uncheck PLC out on the data point.

17.5 Workflows for the L-INX

17.5.1 Starting with Data Points

This workflow is based on defining data points used for the IEC61131 program in the L-INX Configurator and then export them to logiCAD. Figure 335 presents the basic steps of this workflow. To follow the steps refer to the quick-start description of Section 3.6. Refer to section 7.7.2 about additional information how to gather information from the network.

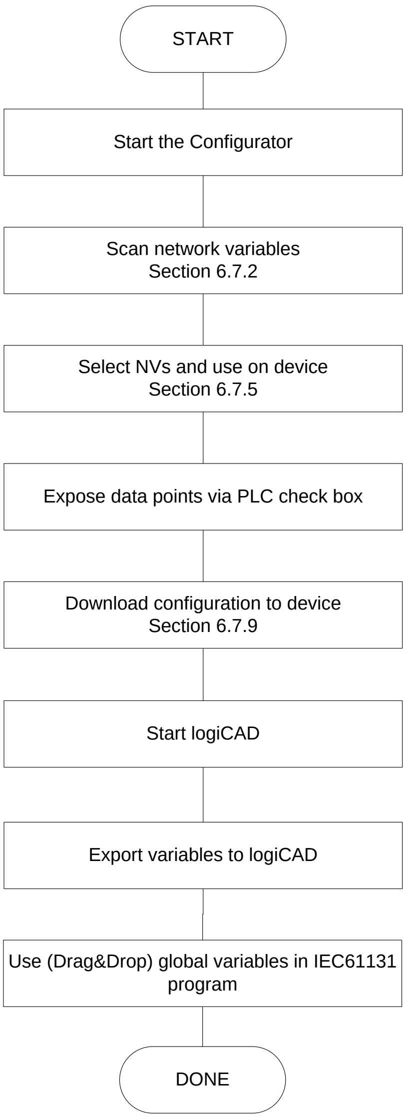


Figure 335: Start with network based information

It is assumed that there are already network data points (CEA-709, BACnet, L-IOB, etc.) available. Clicking the Export variables to logiCAD button in the Configurator creates the IEC61131 variables in logiCAD. The following rules are applied in this process:

- The name of the global variables object is derived from the technology base folder (e.g. CEA709, User Registers, etc.). If there is no matching global variables object one is created. If there is already a suitable global variables object, the existing variables are saved.
- There is already a suitable global variables object: If there is an old and new variable with identical name, the type of the variable is checked. In case of a type mismatch of the old and new variable, the old one is discarded and the new one is imported. Additional a warning is printed on the **Error-state viewer**.
- The name of the global variables object represents the folder name in the L-INX Configurator.

After exporting the data points as global variables to logiCAD, they can be used in the Functionplan 'Plan_1'. To open the respective global variables folder double-click on it. To use the imported variables simply drag and drop the required global variable on to the Functionplan, see Figure 336. The external variables (see Section 17.5.2) are automatically created when adding the global variable to the Functionplan.

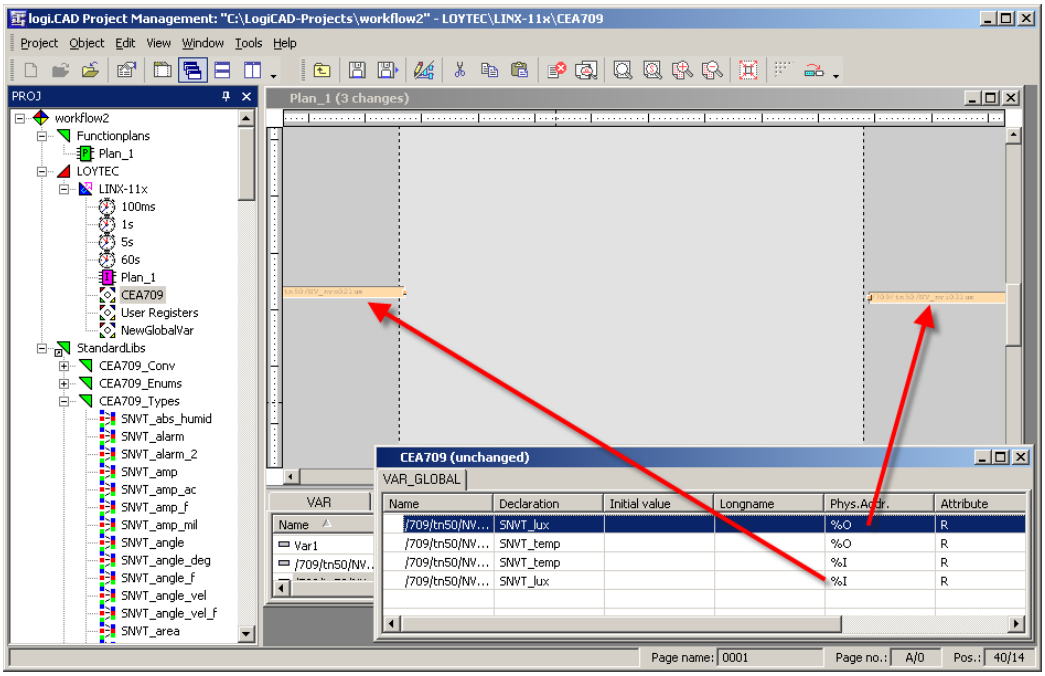


Figure 336: Adding global variables to Functionplans

After adding function blocks that perform the expected work, the IEC61131 program is ready to compile and download (see Section 17.4.2).

17.5.2 Starting with logiCAD

This section introduces a workflow how to develop a new IEC61131 program from scratch using logiCAD. Figure 337 shows the necessary steps to perform. For starting a new logiCAD project refer to the quick-start of Section 3.6.

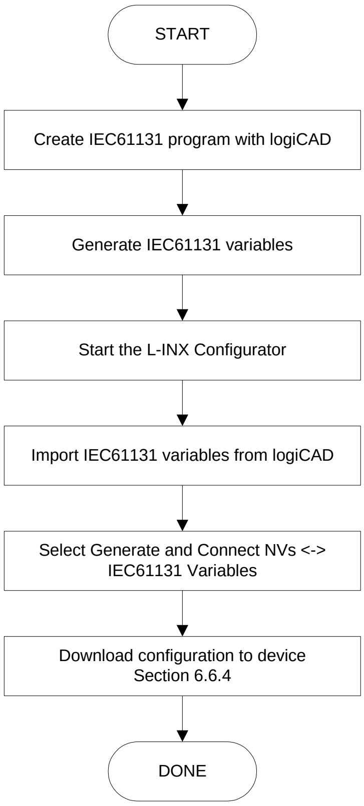


Figure 337: Starting with logiCAD

After creating a new project for a L-INX resource and opening the Functionplan Plan_1 an empty input area is shown, similar to Figure 338.

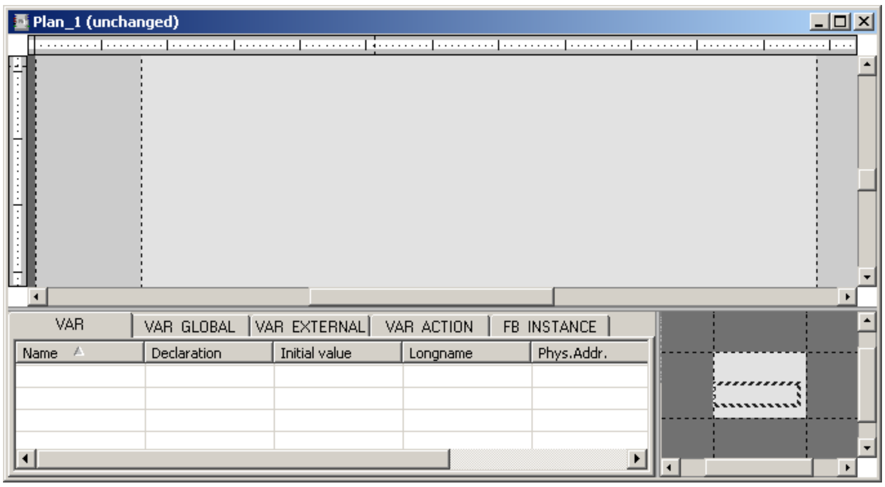


Figure 338: Start new function plan

The dark grey areas on the left and right side are intended to place the input and output variables, the light grey area is used to place the functional blocks.

As described above, global variables are used to interface IEC61131 data points on the LINX-110 device. As assumed for the current workflow, the IEC61131 data points are created on information based on the global variables exported from logiCAD. Hence, for designing the program, external variables are used during the design phase. To create a new variable select the **VAR_EXTERNAL** tab, right click in the declaration area and select **New**, as shown in Figure 339.

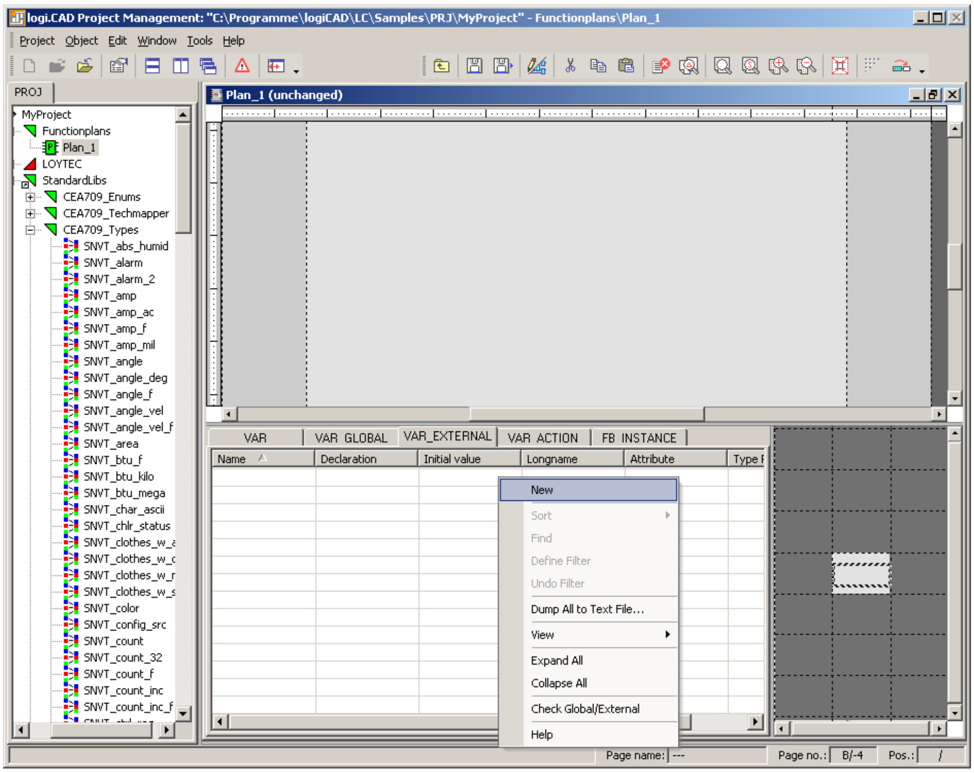


Figure 339: New external variable

In the upcoming dialog the name and the type declaration of the variable must be specified. The type declaration can be done directly by prompting the type into the declaration field, by selecting the type from the pull-down menu or by drag-and-drop of a specific type from the project tree, see Figure 340. Finally, the new variable is added by pressing the **Add** button.

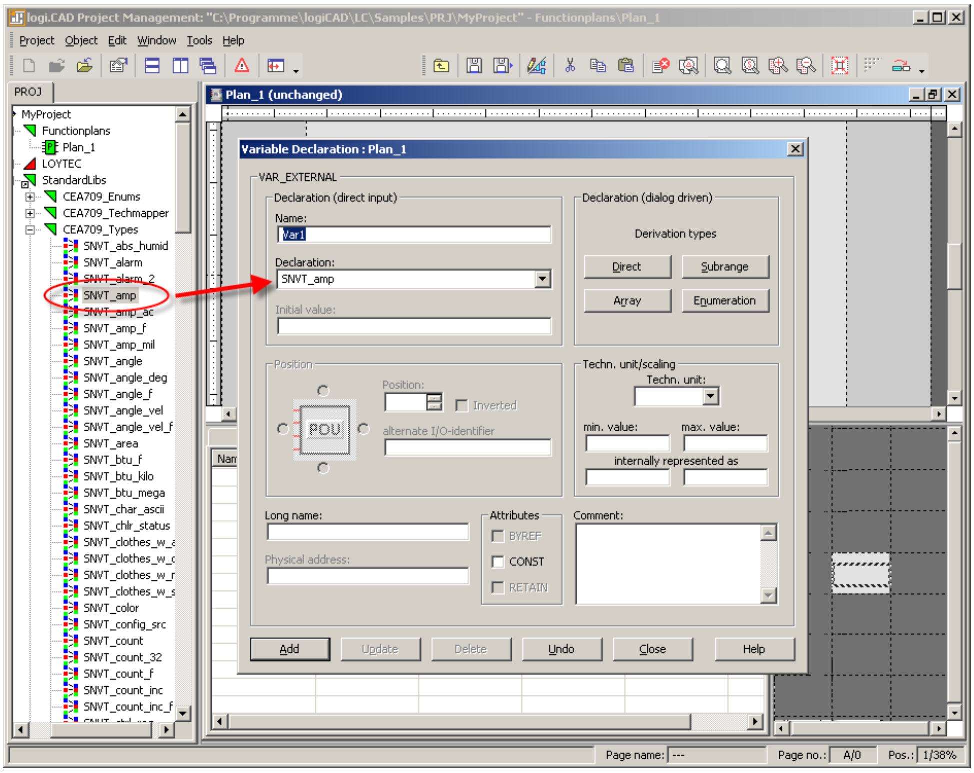


Figure 340: External variable type declaration

The created variable is added to the declaration area and placed to the drawing area by drag-and-drop. At this time the direction of the external variable is not defined, it can be used as input as well as output variable.

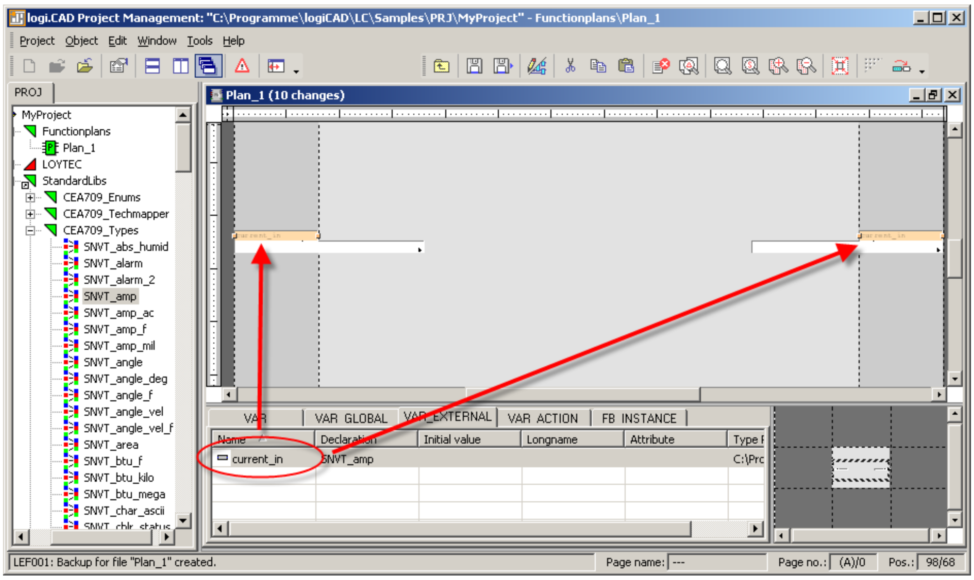


Figure 341: Drag-and-drop external variable to drawing area

Please take care to use an external variable only as input OR output. After adding the external variables to the drawing area, add function blocks to perform the desired actions, see Figure 342 for a sample configuration.

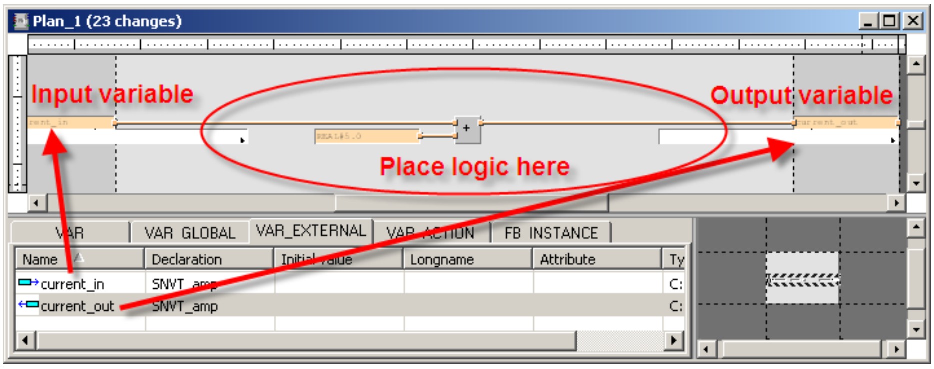


Figure 342: Use external variables

The Functionplan 'Plan_1' represents now a simple program. It adds a defined value to the value of the input variable and sends the result to the output variable.

Add all expected functionality to the 'Plan_1' or use different Functionplans to split up the expected functionality into smaller pieces. But take care about the name and type declaration of external variables when using more than one Functionplan. All external variables with the same name refer to the same global variable.

After adding all functionality, global variables matching the requirements of the defined external variables, must be generated. A tool automatically performs the process of generating the global variables object and the required global variables.

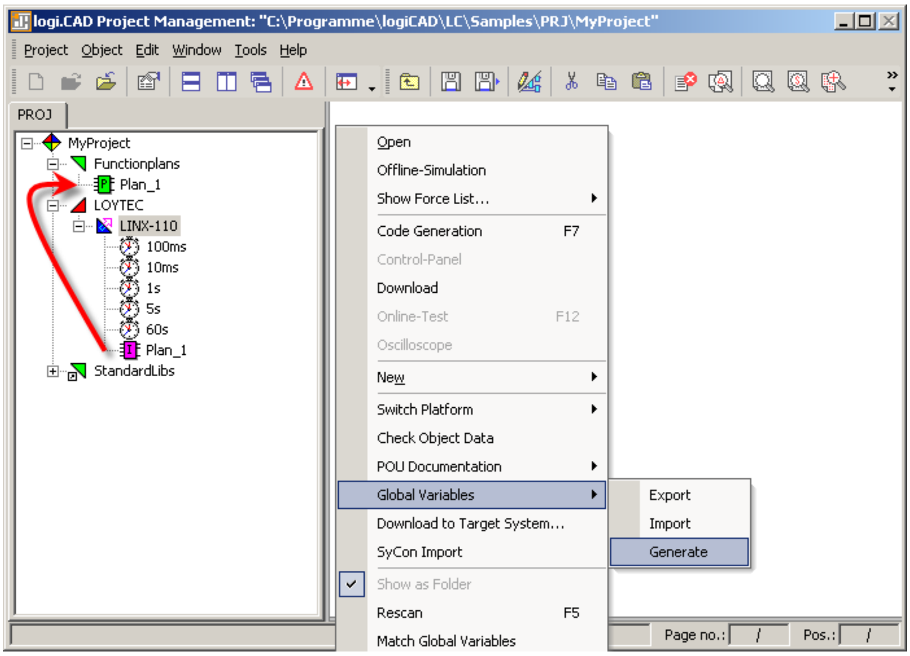


Figure 343: Auto-create global variables

To start the creation of global variables, based on external variables, **Save** all changes, then right click the L-INX device and select **Global Variables → Generate**. Then the selected L-INX resource is parsed and every program instance found is checked for external variables. In Figure 343 the type instance Plan_1 refers to the Functionplan 'Plan_1', as defined in Figure 342. If there are more Functionplans than the predefined 'Plan_1' appropriate program instances for these plans must be added to the resource before creating the global variables. Anyway, only Functionplans referred from program instances are executed.

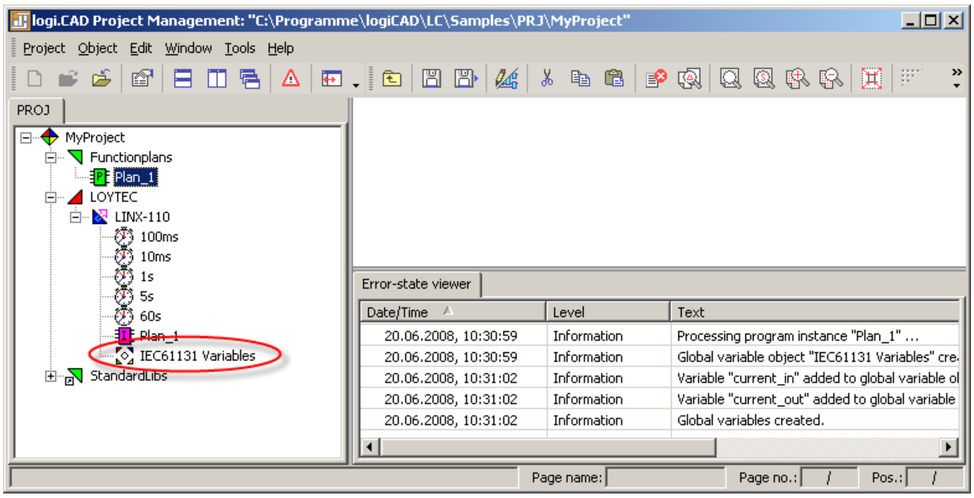


Figure 344: Created global variables object

Now there is a global variables object called IEC61131 Variables available, containing all global representations of the external variables defined before. The **Error-state viewer** reports all processed program instances and added variables, see Figure 344 for details.

The global variables are crated based on the following rules:

- The direction of the variable is determined based on the graphical representation, shown in Figure 345.

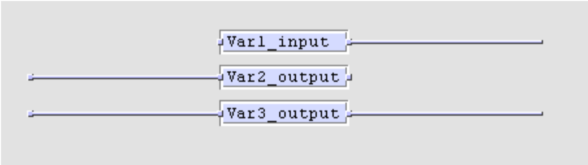


Figure 345: Direction of global variables

Every external variable connected on the right terminal results in a global input variable. External variables connected on the left terminal or on both sides results in a global output variable. Variables connected on both sides can either be used for the force update feature (see section 17.6.1) or as marker (see section 17.6.4). As the tool can not distinguish between these two possibilities, per default a global output variable is created.

- If there is already a global variables object, only new variables are added. In case of external variables using the same name as an already existing global variable, the new definition is used and a warning is printed in the **Error-state viewer**.

- In case of two Functionplans, each referring to a global variable with the same name but a different type, the creation process is stopped and an error is printed in the **Error-state viewer**.

Now the IEC61131 program is ready to compile and download, refer to section 17.4.2 for details.

Based on the above created global variables corresponding IEC61131 data points are created on the L-INX. For doing so, start the L-INX Configurator and click the speed button **Import variables from logiCAD**. Data points are created in the **IEC61131 Variables** folder.

The Configurator reports the results of the import. For the import process the following rules are applied:

- New variables are added
- Variables with same name and type are ignored
- If there are variables with the same name but different type or direction, the variable to import is ignored and a warning is added to the import log.

The name of the folder to import the new variables corresponds to the name of the global variables object. Figure 346 shows the result of the import process.

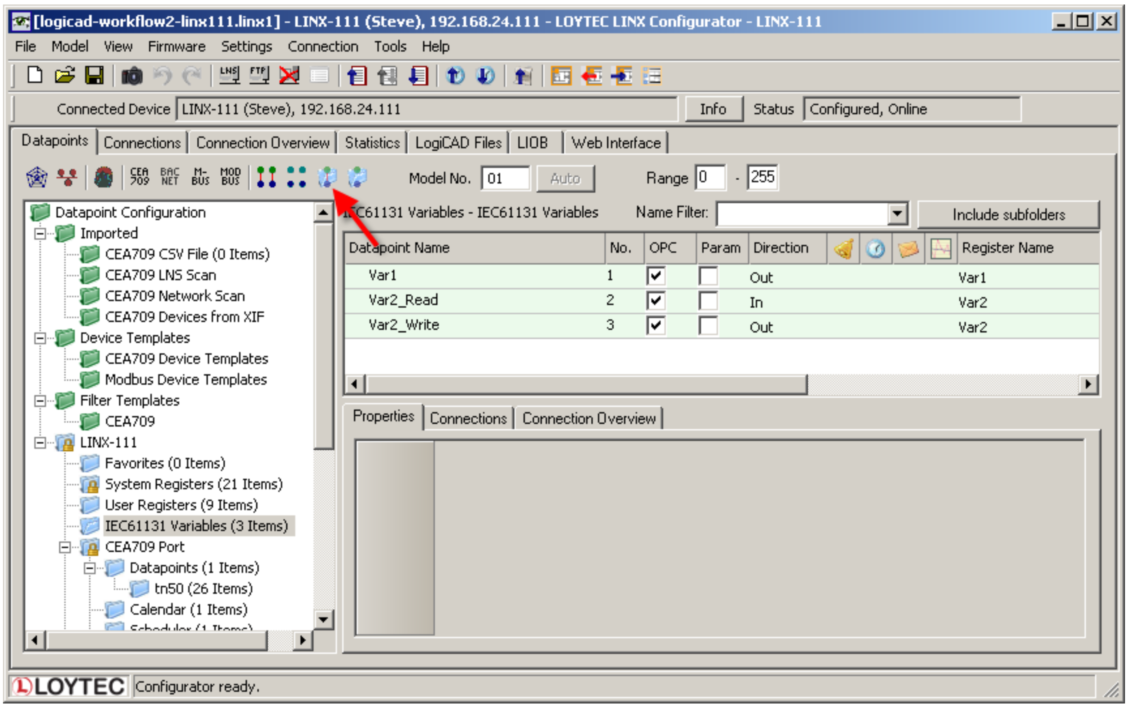


Figure 346: Connect IEC61131 variables

To create the appropriate network data points for all imported IEC61131 variables select the **IEC61131 Variables** folder and press the button **Generate and Connect Network <-> IEC61131 variable from folder**. Check the log output for errors and finally download the configuration to the L-INX device. See Chapter 7 for a general description of the L-INX Configurator.

After rebooting the L-INX device the IEC61131 program is up and running. Check the PLC LED for potential overload.

17.5.3 Pre-compiled IEC61131 Program

In opposite to the last two chapters it is assumed that there are already some components finished, hence starting up from scratch is not suitable. Second, there is the possibility to have an already defined IEC61131 program or an already fixed network interface.

Starting with an already precompiled IEC61131 program results in a similar workflow as presented in Section 17.5.1. The difference is that all logiCAD related tasks are missing. As the IEC61131 program is compiled, the name of the IEC61131 data points is already fixed. The definitions for the data points must be available either in form of a CSV file to import to the L-INX Configurator or as part of a L-INX Configurator project. If the network interface for the L-INX was not already defined, the L-INX Configurator can be used to generate and connect the needed data points.

Additionally, there is the possibility that also the network data points are already fixed or that a given, user-defined interface is necessary. Then the developer has to connect the IEC61131 data points to the corresponding network data points by himself. See section 7.9.10 for details on creating connections manually.

Finally, after downloading the configuration and rebooting the L-INX, the IEC61131 program can be downloaded to the L-INX via the Web UI or the L-INX Configurator. After a final reboot the L-INX loads and executes the IEC61131 program.

17.6 Additional Features

17.6.1 Force Update Functionality

Per default the IEC61131 program only sends updates on changed output values. Every program cycle the input values are fetched, the IEC61131 program is executed and the calculated values are sent to the output driver. If the old values are identically with the new one no updates are sent to the IEC61131 data points. As a result no update is sent to the network.

For some applications, e.g. for a scene controller, it is necessary to send an update on request. E.g. every time the input value is updated, the output value is forwarded to the network, regardless if the value of the output value was changed or not.

For implementing this feature, special vendor blocks are available. First it is necessary to check if there was an update on a selected input within the last execution cycle of the running IEC61131 program. That functionality is offered by the function block **Update Notify** located in the StandardLibs→Loytec_Support folder. Second, an output must be forced to send an update even if the value was not changed. The function block Force Update is used for that functionality, it is located in the StandardLibs→Loytec_Support.

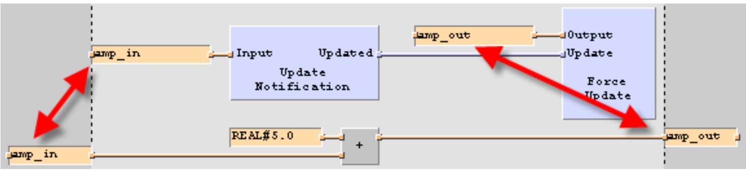


Figure 347: Force update

Figure 347 depicts how to use the force update functionality. Besides of the part of the IEC61131 program that defines the calculations to perform (lower half of Figure 347), additional logic for the force update functionality is required. The global input variable that is monitored for changes is connected to the Update Notification function block. As a result, the Boolean output 'Updated' is set to TRUE for one program cycle, if the value of the connected variable has been updated since the last cycle start. To force the I/O driver send out an update, the global output variable, which shall be updated, is connected to 'Output' of the Force Update block. Hence, every time the 'Update' input of the Force Update block is TRUE, the connected global output variable sends out an update at the end of the program cycle.

Important:	***Every global variable connected to the update notification or force update block must be connected via the right-hand side terminal!***

17.6.2 Using UNVT variables

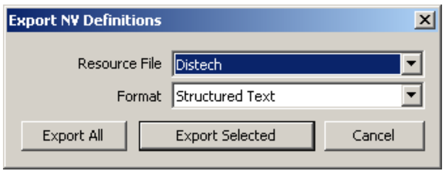
Similar to the predefined CEA709 data types and the technology converter functions, user-defined network variable data types can be used. The L-INX Configurator supports the developer to generate the type definitions needed for UNVTs and enumerations based on resources files. LONMARK

To generate type definitionf for UNVTs in logiCAD

1. Start the L-INX Configurator.

2. Select the menu Tools → Export NV Resource File….

3. Select the resource file to be exported. Select structured text as format and export the file to a location of your choice by clicking **Export Selected**.



4. To import the created type definition file into logiCAD, add a new library to the project. Right-click on the created library and select **Export/Import → Start ST-Import**.

5. Select the file to be imported and check the **Error-state viewer** for the results of the import process.

Using the newly created data types, suitable technology converter function blocks can be created. For each UNVT type, create a normal function block to convert an input of the UNVT type to a number of standard IEC61131 data types and vice versa. You may look at the technology converter blocks for SNVT types which are provided by LOYTEC, to get ideas how to implement your own converter functions.

17.6.3 Create Your Own Data Type

For special applications, custom IEC61131 compliant data types may be created by the user, which do not correspond to any CEA709 network type but should still be available as a data point on the L-INX automation server, in order to access the data point value from the Web interface or over XML as a parameter. Further, such data points can also be made persistent, which makes variables of such custom data types persistent.

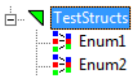
Most of the IEC61131 data types may be used as global variables on the device and the L-INX Configurator will be able to create a suitable register data point for the IEC61131 data type. Supported data types include custom enumeration types (a multi-state register with the required states will be created automatically), strings (a string data point with a maximum length of 128 characters will be built), structures and simple arrays.

If such structured variables shall be mapped to a data point, which shall only be persistent to store the structure but no access to its elements are required in the data point interface create a suitable register data point during the IEC61131 variable import. In this case the data size of custom structures cannot be determined automatically, so the L-INX Configurator does not know how to create a suitable user data point. As a workaround, the type name must contain the desired size of the data point in bytes, for example: MyStructuredType(UT16) will tell the L-INX Configurator to create an IEC61131 register of type ‘user’ with a length of 16 bytes, to hold the data for the IEC61131 structure defined in the logiCAD program.

As an alternative, structures and arrays the logiCAD type definitions can be exported to the Configurator using structured text. The Configurator builds custom structured data point types in the logiCAD category out of the structured text definitions. In this case, the exact data size of a user register is determined by the structured data type. Structure members of the logiCAD types are represented as a tree of sub-data points. logiCAD arrays (like ARRAY [1..16] OF INT) are represented as a series of single data points where each data point represents one array element. Note, that the array in logiCAD can still be access by index.

To Use logiCAD Types in User Registers

1. In logiCAD right-click on the library with the types that shall be exported and select **Export code…**.



2. In the export dialog choose '.ST' as the export configuration, select the option single file and choose a target file (see Figure 348). Then click **Export**, which creates the structured text file.

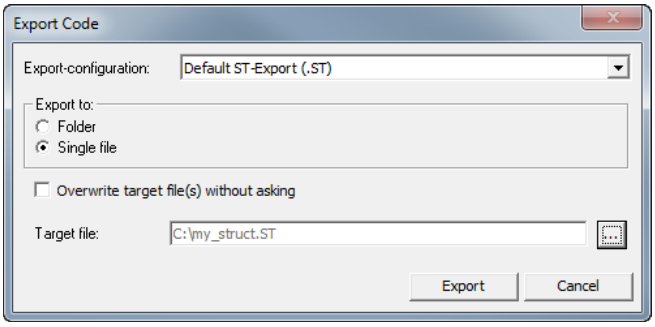


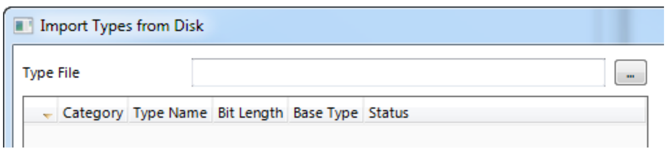
Figure 348: Export logiCAD types as structured text.

3. In the Configurator go to the menu **Tools → Manage Structured Types**.

4. In the dialog choose the type **Category** 'logiCAD'.

5. Click the speed button **Load types from disk** .

6. In the dialog **Import Types from Disk** click on .

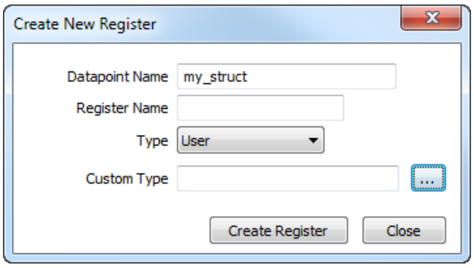


7. In the file selector dialog, choose the file type Structured Text '.ST' and open the previously exported file.

8. The available types in the structured text file appear on the left-hand side of the dialog. Transfer all listed types to the Configurator by clicking . Alternatively select specific types and click . Then click **Close**.

9. In the **Manage Structured Types** dialog, the imported logiCAD types are now listed. Click **Close**.

10. Now you are able to create a structured user register. For **Custom Type** choose one of the imported logiCAD structure types.



17.6.4 Using Persistent Data Points and Markers

Persistent data points are data points on the device that hold their value even after power loss (see Section 6.2.4). In the IEC61131 program there is no difference in handling global variables connected to persistent data points or to non-persistent data points. When using global variables on the device resource that shall retain their values after a reboot, the respective data points need to be configured with the persistent flag in the L-INX Configurator.

Global input variables marked as persistent, supply the IEC61131 program every time with the last received, valid data, even after a power failure. To enable this behavior, set the respective input data points persistent in the L-INX Configurator and download the data point configuration to the device.

Global variables declared as marker can be used as input and output variable in IEC61131 programs. LogiCAD is not able to distinguish markers from global output variables used in combination with the force update feature, see Figure 347. As a result it is not possible to create a global variables object holding markers automatically. The procedure as described in Section 17.5.1 is not able to decide when to create an output variable or a marker. Markers must be created manually by adding them to the global variables object and setting the physical address to %M.

17.6.5 Using Retain Variables

Retain variables can be used to maintain certain states or parameters of a program over a reboot in those variables, which cannot be kept in persistent data points. Typically, input variables (%I) will be marked as retain variables. For doing so check the **RETAIN** attribute in the variable declaration when creating or editing a variable as shown in Figure 349.

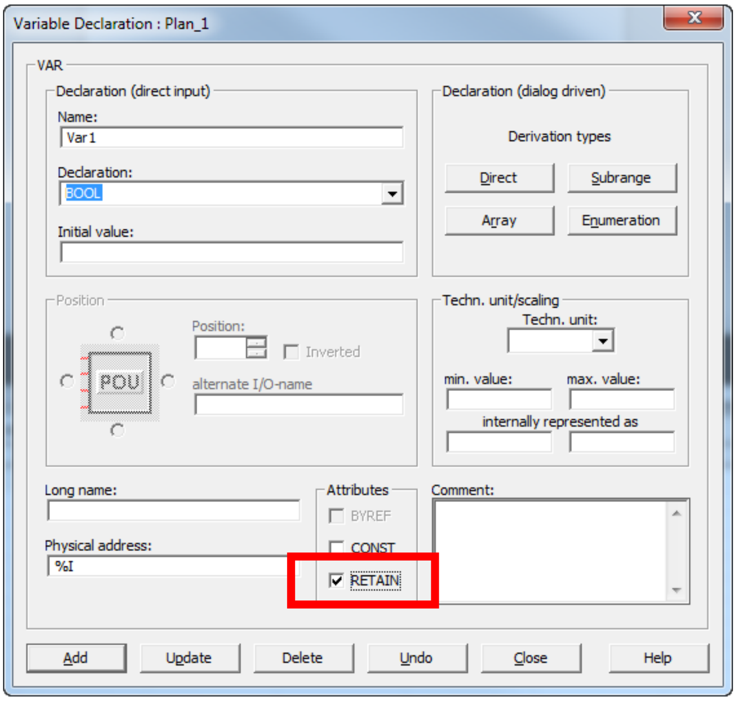


Figure 349: Declaring a retain variable.

The retain attribute should be used in global variables and local variables on function plan only. Global variables on the device resource are managed by the I/O driver and must not use the retain attribute. Instead data point persistency (see Section 17.6.4) must be used for those variables. Furthermore, it is not recommended to mark local variables in function blocks as retain variables.

The contents of retain variables are written to Flash storage every 5 minutes and when shutting down the device. This provides good compromise between write granularity for normal operation and ensuring Flash life-time. If the contents shall be written explicitly at a defined time, use the **RetainCtl** function block located in the **logi.library/Data** folder. Connect the store input (S) and set this input TRUE for one cycle to trigger writing retain data. Make sure to set the input back to FALSE after this cycle to avoid multiple consecutive writes.

17.6.6 L-INX System Registers, System Time

L-INX system registers, such as the system time or the CPU load, can be used within IEC61131 programs. Therefore, for each system input variable, a global input variable of type UDINT may be created within the IEC61131 program. Then, connections to the appropriate system registers are created manually with the L-INX Configurator (see section 7.9.10).

To use the system time within the IEC61131 program, connect the AtoDT converter (located in the StandardLibs->IEC61131-3_ (EXT) folder) to the global input variable that receives the system time.

17.6.7 Code Protection

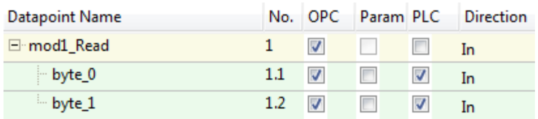
There are 4 data points used for code protection. These data points, in combination with an adapted IEC61131 program, can be used to protect your 61131-Program Intellectual Property. Please contact LOYTEC sales for further information.

17.6.8 Using Structured Data Point Members

Some network technologies provide structured data points. Their types are defined by the network technology and are available in the Configurator. Those structure types can either be used in logiCAD by implementing a technology converter (see Section 17.4.3) or by exposing the sub-data points of the respective structures.

To Use Sub-Data Points in logiCAD

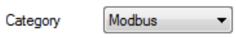
1. In the Configurator, expand the structured data point and expose the desired sub-data points of its structure members to the PLC.



2. Export the variables to logiCAD. The exposed structure members appear for example as '/UR/mod1_Read.byte_0' in the 'User Registers' global variables block.

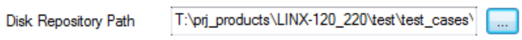
3. Export the types for the respective type category in the menu **Tools → Manage Structured Types…** . This opens the structured types management dialog.

4. Choose the type **Category**, e.g. 'Modbus'.



5. Click the speed button Save to disk .

6. In the dialog **Export Type to Disk** choose a **Disk Repository Path** and select the extension '.ST' for structured text.

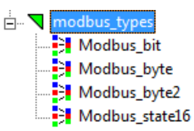


7. Copy all listed types to the disk repository by clicking . Alternatively select specific types and click . Then click **Save** and exit the dialog.

8. In logiCAD create a new library folder, e.g. 'modbus_types'.

9. Right-click on that new library and select **Export/Import → Start ST Import**. In the file dialog choose the previously exported structured text file.

10. The type definitions are now available in the new library.



17.6.9 BACnet Server Objects

BACnet server objects provide additional functionality compared to other network technologies. The most important tasks in an IEC61131 program are:

- Reading sensor data from BACnet input objects,
- writing to commandable BACnet objects with a priority,
- revoking values from commandable BACnet objects,

- handling objects when out of service.

The default data flow direction for a commandable BACnet object (e.g. AO) is input to the PLC (%I). This means its value is commanded over the BACnet network and the logic processes the resulting value. The default for non-commandable BACnet objects (e.g. AI) is output (%O). This means the logic writes the value, which is read out of the object over the BACnet network. With this type of variables the BACnet objects are treated as variables of the IEC61131 program.

When exposing local I/O to BACnet server objects, the BACnet objects represent the I/O values and the IEC61131 program behaves like a user from the network on those objects. Typically sensor values are connected to BACnet input objects (e.g. AI). To read the sensor value the PLC needs an input variable. Since the default data point direction is output, a separate access data point for the PLC must be created. For doing so, use the **Add/Remove BACnet properties** option from the context menu of the data point list (see Section 7.9.8) and create a read data point for the Present_Value. Figure 350 shows an example, where the PLC uses the input '/BAC/AI2.Present_Value_Read' to get the sensor values from AI2.

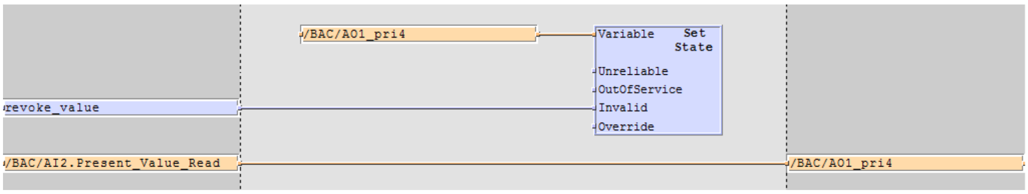


Figure 350: PLC with BACnet objects.

BACnet output objects (e.g. AO) are connected to actuator I/Os. On those BACnet objects the PLC behaves like a user from the network and needs to write and revoke values at certain priority slots. For writing with a certain priority to a commandable object, a priority write data point needs to be created (see Section 7.9.11). This data point is configured with a BACnet write priority in the Configurator and used as an output in the PLC. The example in Figure 350 writes with priority '4' to AO1 over '/BAC/AO1_pri4'.

To revoke values in commandable server objects, the **SetValueState** function block must be used. It is located in the **Loytec_Support/Service** folder. The I/O variable of the controlled BACnet object is connected to the **Variable** input of the function block. A Boolean signal needs to be connected to the **Invalid** input. When this input changes to TRUE, the value is revoked at the respective priority slot. When it remains TRUE in the next cycle the service function block is idle. Figure 350 shows an example, how to revoke a value in 'AO1' at priority slot '4'.

With the out-of-service feature BACnet server objects can be decoupled from the senor and actuator I/O equipment. When taken out of service, the I/O variables of the PLC are then decoupled from the network. The input variable Present_Value_Read of an AI no longer tracks the sensor value but reflects the value set in the AI from a BACnet OWS. This holds true for all non-commandable BACnet objects. Writing to the priority output variable of an AO only updates the priority array but no longer drives the actuator. This holds true for all commandable BACnet objects.

The out-of-service state is reflected in the data point status and can be accessed with the **ValueState** function block. This function block reads the value state and is used in a similar fashion as the **SetValueState** function block.

17.6.10 Custom Serial Protocols

A logiCAD program can implement simple custom serial protocols using the **SerialComm** function block. The logic program assigns the Baud rate settings for the protocol as well as a logical *bus number*. This bus number can be selected on the port configuration Web UI in order to activate the custom serial protocol on that port.

Custom serial protocols with the following communication properties can be supported:

- ASCII protocol: The serial communication block works with strings that are transmitted and received.
- Line-Oriented: Each command and reply is modeled as a text line. The line can be terminated by any combination of CR and LF.
- Start/End Character: If not line-oriented, a start and end character can be specified to detect start and end of a frame. The end character must be escaped in the frame content.
- Master/Slave: The serial communication block is a master, which sends requests over the serial port and expects slave(s) to respond. The serial communication block can also implement a slave, which receives requests and creates responses.
- Addressing Information: If the serial protocol has more than one slave, each request must have address information, which selects a specific slave. Especially for RS-485 bus communication address information is required. Since frames sent out are also received, they need to be filtered out by address.

To start the implementation of a custom serial protocol use the function block **SerialComm** located in the StandardLibs→Loytec_Support→Service folder. Add a bus number and communication settings as a string (e.g., „38400-8N1-X“) as shown in Figure 351. Best practice for an implementation is to use an SFC-based state machine. All necessary input and output signals are available on this function block. The minimum number of states is Idle, Rx, Tx. For a full specification of this block use the F1 help.

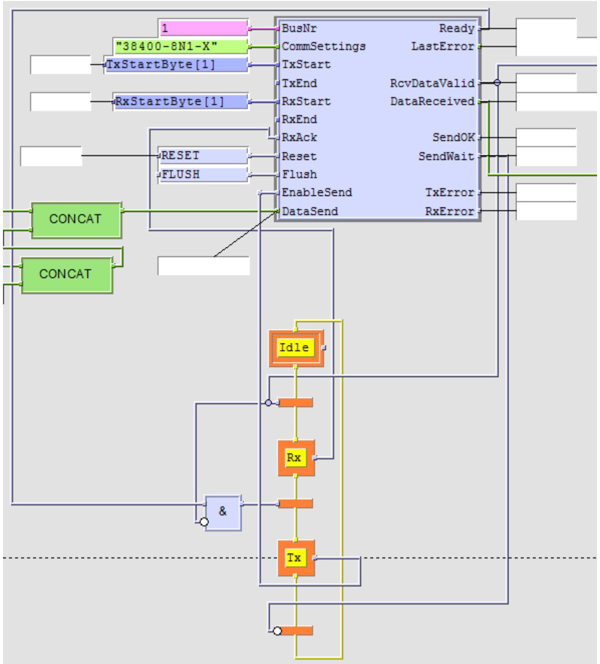


Figure 351: Example for the SerialComm block.

String processing may build commands from data point values and extract data from responses, which are written to data points. The data points should be user registers that are organized into a separate sub-folder for the custom serial protocol.

18 Operating Interfaces

18.1 Common Interface

18.1.1 Schedule and Calendar XML Files

The daily schedule and calendar pattern configuration can be changed at run-time over the Web UI or the network. An alternative way to change that configuration is to download a schedule and calendar XML file via FTP onto the device. After the file has been downloaded, the new configuration becomes effective immediately. The device does not need to be rebooted. The files are located in

```
/tmp/uid/sched/UID.xml
/tmp/uid/cal/UID.xml
```

The *UID* is the unique ID of the data point. The UID can be obtained from the ID column in the data point list as shown in Figure 229. A schedule data point with UID 107C would result in the schedule XML file '`/tmp/uid/sched/107C.xml`'. The UID remains constant for the life time of the data point even when the name or description is changed.

The content of the XML file must be compliant to the scheduleCfg schema. This schema can be found at the LOYTEC Web site. The XML documents can refer to the target namespace http://www.loytec.com/xsd/scheduleCfg/1.0/

18.1.2 Trend Log CSV File

The CSV file format for a trend log and the location of those files are defined in this section. The trend log CSV files are accessible either via their UID only, or in combination with contents of the trend log object name. The files are located in

```
/tmp/uid/trend/UID.csv
/data/trend/Datapointname_UID.csv
```

The *UID* is the unique ID of the data point. The UID can be obtained from the ID column in the data point list as shown in Figure 229. For a more user-friendly listing of the files, the *Datapointname* contains the trend log's object name. It is truncated after 23 ASCII characters to fit the requirements of the file system. A trend CSV file for the trend object 'trend0' and the UID '107C' would result in the CSV file '`/data/trend/trend0_107C.csv`'. The UID remains constant for the life time of the object even when the name is changed.

The CSV file format for a trend log is defined in this section. The CSV file starts with a header, containing at least the first line, which specifies the CSV format (log_csv_ver). The current version is 2. The next line contains the field log_device. It has trailing fields that specify the vendor, product code, firmware version and device ID string. The Device ID String can be one of the following: (IP) 192.168.24.100, (BACnet Device) 224100, (CEA-709 NID) NID.

The log_info line specifies the fields UID and name of the trend log object. The line log_create has two fields specifying the date and time when this CSV log was generated. The line log_capacity has two fields: the current number of log entries in the file and the log capacity.

Following are one or more lines of log_item. Each line specifies a trended data point. The first field is the index, the second the ID of the logged data point, the third the data point name. The data point name can be augmented by engineering units in square brackets. Log entries in the CSV refer to the item index to identify the data point, for which the entry was logged.

```
#log_csv_ver,2
#log_device;LOYTEC;Product Code;Firmware Version;Device ID String;Serial No
#log_info;Log-ID;Log Name
#log_create;YYY-MM-DD;HH:MM:SS
#log_capacity;filled;capacity
#log_item;index;UID;data point name [units]
```

After those lines any number of comment lines starting with a hash character ‘#’ are allowed. One line contains the column headings. Lines that are not comments specify one log record per line, using the column information as described below. The columns are separated by commas ‘,’ or semi-colons ‘;’. If commas are used as a separator, the decimal point must be a point ‘.’. If semi-colons are used, the decimal point must be a comma ‘,’.

Column	Field	Example	Description
A	Sequence Number	50	The log record sequence number. This is the monotonously increasing sequence number, which is unique for each log record.
B	Source	0	Data point source identifier. Indexes into logger_entry header. For value lines in a multi-column CSV, this field indexes the first column, which has a value. For the ERROR record type, the field indexes the data source that caused the error. For LOGSTATE, TIMECHANGE records this field is not applicable and set to 255.
C	Record Type	2	The record type: LOGSTATE (0), BOOL (1), REAL (2), ENUM (3), UNSIGNED (4), SIGNED (5), NULL (7), ERROR (8), TIMECHANGE (9)
D	Error/Time Change/Log Status	1	This field is valid for records of type ERROR, TIMECHANGE, and LOGSTATUS.
E	Date/Time	2007-11-02 15:34:22	The date/time of the log record. This is in the format YYYY-MM-DD HH:MM:SS.
F	Value 0	24,5	Logged value from source 0 or empty
G	Value 1	200	Logged value from source 1 or empty
…	…		
…	Value *n* – 1	5000	Logged value from source *n* – 1 or empty

Table 25: Columns of the Trend Log CSV File.

There are as many value columns as value sources specified in the header. If at a given date/time more values are logged, all of them appear in the same line. If at that given time some sources did not log values, those columns are left empty. The “Source” column in a multi-value CSV refers to the first data source that supplied a value in a given line.

18.1.3 Alarm Log CSV File

The historical alarm logs are also accessible as CSV-formatted files. The alarm log CSV files are accessible either via their UID only, or in combination with contents of the alarm log object name. The files are located in

```
/tmp/uid/allog/UID.csv
/data/allog/Alarmlogname_UID.csv
```

The *UID* is the unique ID of the alarm log object. The UID can be obtained from the ID column in the data point list of the alarm log folder, similar to obtaining the UID of trend log objects. For a more user-friendly listing of the files, the *Alarmlogname* contains the alarm log's object name. It is truncated after 23 ASCII characters to fit the requirements of the file system. A trend CSV file for the alarm log object 'alarmlog0' and the UID '100C' would result in the CSV file '`/data/allog/alarmlog0_100C.csv`'. The UID remains constant for the life time of the object even when the name is changed.

The CSV format of the alarm log CSV file is identical to the trend log CSV format as described in Section 18.1.2.

18.1.4 Data Point Template CSV File

The data point template CSV file (.dpcsv) allows the creation of data points from a list of data point templates. Each line in this CSV file will create a data point under the specified path and name following a data point template file. A data point template file (.dptmpl) contains all the information to create a data point of a given technology (e.g., a BACnet server object or a user register with an alarm condition).

The first line of the file must contain a comment, starting with a hash character '#' specifying the columns used by the template CSV file:

```
#Path;Name;Description;PlcIn;PlcOut;Opc;Trend;Schedule;TemplateFile
```

After that line any number of comment lines starting with the hash character '#' are allowed. Lines that are not comments specify a data point to be created, using the column information as described in Table 26. The columns are separated by commas ',' or semi-colons ';'. The paths of the referenced data point template XML files are relative to the location of the data point template CSV file.

Column	Field	Example	Description
A	Path	Floor1.Room101	The path under which the data point shall be created. The path is relative to the technology's root path, e.g. relative to 'User Registers'.
B	Name	RoomTemp	The name of the data point to be created. For structured data points, this is the name of the top-level data point. This name overrides the name of the template data point.
C	Description	Room Temperature	Optional description used in the data point. This overrides the description in the template data point.
D	PlcIn	1	Set the PLC in flag ('1') or clear it ('0') on the created data point. This overrides the PLC setting of the template data point. If left empty, the PLC setting of the template data point is used. For structured data points this setting is ignored.
E	PlcOut	0	Set the PLC out flag ('1') or clear it ('0') on the created data point. This overrides the PLC setting of the template data point. If left empty, the PLC setting of the template data point is used. For structured data points this setting is ignored.
F	Opc	1	Set the OPC flag ('1') or clear it ('0') on the created data point. This overrides the OPC setting of the template data point. If left empty, the OPC setting of the template data point is used. For structured data points this setting is ignored.
G	Trend	1	If this field is '1' the created data point is trended in a generic trend. The user can override this decision to use another trend technology during the import.
H	Schedule	1	If this field is '1' the created data point is scheduled by a generic scheduler. The user can override this decision to use another scheduler technology during the import.
I	TemplateFile	TempBACnet.dptmpl	The data point is created from this data point template. The location is relative to this CSV file.

Table 26: CSV columns of the data point template import file.

18.2 CEA-709 Interface

18.2.1 NV Import File

Network variables can be imported to the Configurator software in a CSV file. The format of this file is described in this section.

The first line of the file must contain a comment, starting with a hash character '#' specifying the format version and import technology:

```
#dpal_csv_config;Version=1;Technology=CEA709
```

After that line any number of comment lines starting with the hash character '#' are allowed. Lines that are not comments specify one NV per line, using the column information as described in Table 27. The columns are separated by commas ',' or semi-colons ';'. Which separator is used can be configured in the Web UI (see Section 5.2.1).

Column	Field	Example	Description
A	SNVT	39	A numeric value of the SNVT (as defined in the SNVT master list). The example value 39 represents a SNVT_temp.
B	NV index	0	The NV index in decimal notation of the NV on the network node. Index starts at 0.
C	NV selector	1	The NV selector in decimal notation of the NV on the network node.
D	NV name	nvoTemp	The NV programmatic name of the NV on the network node.
E	is output	1	Defines if this NV is an output on the network node. '1' means the NV is an output on the network node.
F	flag auth cfg	1	'1' defines that authentication can be configured for this NV on the network node.
G	flag auth	0	'1' defines that the NV is authenticated.
H	flag priority cfg	1	'1' defines that the priority can be configured for this NV on the network node.
I	flag priority	0	'1' defines that the NV is using priority.
J	flag service type cfg	1	'1' defines that the service type can be configured for this NV on the network node.
K	flag service ack	1	'1' defines that the NV is using acknowledged service.
L	flag polled	0	'1' defines that the NV is using the polled attribute
M	flag sync	0	'1' defines that the NV is a synchronous NV.
N	Deviceref	1	This field is a numeric reference to a device description. If it is the first occurrence of this reference in the file, the columns defined below must be filled in. Otherwise, they can be left out.
O	programID	9000A44850060402	The program ID string of the network device.
P	neuronID	80000000C8C8	The NID of the network device.
Q	Subnet	2	The subnet address of the network device. Use '0' if the device has no subnet address information.
R	Node	3	The node address of the network device. Use '0' if the device has no node address information.
S	location str	0	The location string of the network device. Use '0' if no information is available.
T	Device name	DDC	The device name of the network device. Leave this field blank if this information is not available.
U	node self-doc	&3.2@0,2	Self-documentation string of the device (special characters are escaped)
V	NV length	2	NV length in bytes
W	NV self-doc	@0|4	NV self-documentation string (special characters are escaped)

Table 27: CSV Columns of the NV Import File.

18.2.2 Node Object

The L-INX and the L-GATE provide a node object conforming to the LONMARK guidelines.

- The Node Object accepts the following commands via *nviRequest*: RQ_NORMAL, RQ_UPDATE_STATUS, RQ_REPORT_MASK, RQ_ENABLE, RQ_DISABLE, RQ_UPDATE_ALARM, RQ_CLEAR_ALARM, RQ_RESET, RQ_CLEAR_RESET

- LONMARK alarming is supported via *nvoAlarm* (SNVT_alarm) and *nvoAlarm_2* (SNVT_alarm_2). This allows devices supporting the LONMARK alarm notifier profile to receive alarms generated by the device and react with a defined action (e.g., send an email). By supporting both alarm SNVTs, SNVT_alarm and SNVT_alarm_2, legacy and state-of-the-art alarm handling is supported.
- nviDateEvent (*SNVT_date_event*), nvoDateResync (*SNVT_switch*): These NVs are part of the standard LONMARK node object, if schedulers are used. If not bound, the local calendar is used. If a global calendar shall be used, both of these NVs must be bound to the respective NVs of the global calendar object.
- nviTimeSet (*SNVT_time_stamp*): When writing to this NV, the system is set, if the configure time-source is “LonMark” or “Auto” (see Section 5.2.1). The time value is interpreted as local time
- nvoSystemTemp (*SNVT_temp*): This NV can be used to poll the system temperature of the device. It does not send updates and must be polled.
- nvoSupplyVolt (*SNVT_volt*): This NV can be used to poll the supply voltage of the device. It does not send updates and must be polled.
- nvoIpAddress (*SNVT_str_asc*): This NV can be used to poll the IP address of the device. It does not send updates.
- nciEarthPos (*SNVT_earth_pos*): This configuration property can be used to set the earth position of the device. It has been implemented as an NV to make other devices send that configuration to the device over the network (e.g., from a GPS receiver).
- nviClearStat (*SNVT_switch*): When writing {100.0 1} to this NV, the channel monitor objects’ statistics data are cleared.
- nvoUpTime (*SNVT_elapsed_tm*): This NV contains the elapsed time since the last reboot.

18.2.3 Real-Time Keeper Object

When the scheduler objects are enabled in the project settings, the device includes the standard LONMARK real-time keeper object. The Real-Time Keeper Object is used to synchronize the system time of multiple LONMARK compliant devices.

The object has the following network variables:

- nvoTimeDate (*SNVT_time_stamp*): Propagates the devices current system time and date (local time). It is typically bound to the nviTimeSet input network variable of the node objects of the LONMARK compliant devices, which are synchronized with the system time of the device. The update rate of the nvoTimeDate can be configured using the configuration property SCPTupdateRate (default every 60 seconds).

18.2.4 Channel Monitor Object

The Channel Monitor Object functional block is responsible for network monitoring. There is one object for each channel, the device is attached to: The channel monitor object with index 0 corresponds to the FT port of the device, while the object with index 1 corresponds to the IP-852 port of the device. If a port is not available in the current system configuration, the nvoElapsedTime is set to the invalid value and nvoPort is set to 255. The LINX-11x models do not possess a channel monitor object.

Each object has the following network variables:

- nvoPort (*SNVT_count*): Index of port associated with this Channel Monitor Object instance. Port 1 corresponds to the FT port of the device, while port 2 corresponds to the IP-852 port of the device. If the monitored port is not available in a system configuration, the value is 255. This NV is polled only.

- nvoElapsedTime (*SNVT_elapsed_tm*): Time since device powered up or since the statistics for this port where reset. The statistics can be reset with the network variable *nviClearStat* in the node object (see Section 18.2.2) or if the node is reset with a network management command (e.g., while the device is commissioned). If the monitored port is not available in a system configuration, the value is set to the invalid value. The NV is polled only.
- nvoAvgPkts (SNVT_count_32): The average number of packets per second received or transmitted via the associated channel since power-up or since the statistics for this port where reset.
- nvoIvalBandUtl (*SNVT_lev_cont*): Bandwidth utilization of associated channel during the last interval. For a smooth operation of the CEA-709 segment, the average bandwidth utilization must remain below 50 %.
- nvoIvalCrcErr (*SNVT_lev_cont*): Percentage of packets with CRC error received on the associated channel during the last interval.
- nvoIvalMissed (*SNVT_lev_cont*): Percentage of packets from the associated channel which could not be processed during the last interval.
- nvoIvalPkts (*SNVT_count_32*): Number of packets received or transmitted via the associated channel during the last interval.
- nvoTotalCrcErr (*SNVT_count_32*): Total number of packets with CRC error received via the associated channel since power-up or since the statistics for this port where reset.
- nvoTotalMissed (*SNVT_count_32*): Total number of packets from the associated channel which could not be processed since power-up or since the statistics for this port where reset.
- nvoTotalPkts (*SNVT_count_32*): Total number of packets received or transmitted via the associated channel since power-up or since the statistics for this port where reset.
- nvoMaxBandUtl (*SNVT_lev_cont*): Maximum value of *nvoIvalBandUtl* since power-up or since the statistics for this port where reset. For a smooth operation of the CEA-709 segment the average bandwidth utilization must remain below 50 %.
- nvoMaxCrcErr (*SNVT_lev_cont*): Maximum value of *nvoIvalCrcErr* since power-up or since the statistics for this port where reset.
- nvoMaxMissed (*SNVT_lev_cont*): Maximum value of *nvoIvalMissed* since power-up or since the statistics for this port where reset.
- nvoMaxPkts (*SNVT_count_32*): Maximum value of *nvoIvalPkts* since power-up or since the statistics for this port where reset.
- nvoIvalMisPre (*SNVT_count_32*): Number of missed preambles per second on the associated channel measured during the last interval. A missed preamble is detected whenever the link layer receives a preamble which is shorter then the defined preamble length. A large number in this counter is usually due to noise on the channel.
- nvoTotalMisPre (*SNVT_count_32*): Total number of missed preambles per second on the associated channel measured since power-up or since the statistics for this port where reset.
- nvoMaxMisPre (*SNVT_count_32*): Maximum value of *nvoIvalMisPre* since power-up or since the statistics for this port where reset.
- nvoChnlAlarm (*SNVT_switch*): Signals an overload alarm condition of the channel during the last statistic interval. A channel can be overloaded due to one of the following conditions:
  - The bandwidth utilization during the last statistic interval (*nvoIvalBandUtl*) exceeded the limit defined by the *SCPThighLimit1* (default 70 %) OR

  - The CRC Error Rate during the last statistic interval (*nvoIvalCrcErr*) exceeded the limit defined by the *SCPThighLimit1* (default 5 %) OR
  - The Missed Packets Rate during the last statistic interval (*nvoIvalMissed*) was not zero OR
  - The Missed Preamble Rate during the last statistic interval (*nvoIvalMisPre*) exceeded the limit defined by the *SCPThighLimit1* (default switched off).

  If an overload is detected, the network variable is set to {100, ON}. If no error occurred, it is set to {0, OFF}.

- nvoChnlAlarmRat (*SNVT_lev_cont*): Ratio between statistic intervals during which the channel was in overload alarm condition and intervals during which the channel was not in overload alarm condition since power-up or since the statistics for this port where reset.

In addition, each channel monitor object has the following SCPTs:

- SCPTifaceDesc: This configuration property contains a human-readable name of the monitored port. Possible values on the device are “CEA-709”, “IP”, or “inactive”.
- SCPTmaxSndT: Defines how often output NVs are transmitted. Exceptions are nvoPort, and nvoElapsedTime, which are polled-only.

18.2.5 Calendar Object

When the scheduler objects are enabled in the project settings, the device includes the standard LONMARK calendar object.

18.2.6 Scheduler Object

When the scheduler objects are enabled in the project settings, the device includes the configured number of standard LONMARK scheduler objects.

18.2.7 Clients Object

When the remote AST object feature is enabled in the project settings, the device includes a proprietary object, which is a container for network variables required to implement the remote object features.

For remote schedulers and calendars, *nviSchedLink* and *nviCalLink* NVs are created. For alarm clients, nviAlarm_2 NVs are created.

18.2.8 Gateway/PLC Objects

The device contains eight proprietary Gateway/PLC objects. If the device contains the IEC61131 function, the blocks are called ‘PLC’, otherwise they are called ‘Gateway’. These are containers for all NVs which are configured on the device’s CEA-709 port. They are intended for grouping NVs. When static NVs are created, they can be assigned to any of the eight gateway/PLC blocks. When creating dynamic NVs in the LNS-based tool, the NVs should be added to the gateway/PLC blocks.

18.3 BACnet Interface

18.3.1 Device Object

The BACnet interface provides one device object as shown in Table 28. The following Sections describe the device object’s properties in detail, subsuming related properties in a single Section in order to provide a coherent overview.

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Vendor_Name	CharacterString	R
Vendor_Identifier	Unsigned16	R
Model_Name	CharacterString	R
Firmware_Revision	CharacterString	R
Application_Software_Version	CharacterString	R
Location	CharacterString	W
Description	CharacterString	W
Protocol_Version	Unsigned	R
Protocol_Revision	Unsigned	R
Protocol_Services_Supported	BACnetServicesSupported	R
Protocol_Object_Types_Supported	BACnetObjectTypesSupported	R
Object_List	BACnetARRAY[N] of BACnetObjectIdentifier	R
Database_Revision	Unsigned	R
Max_APDU_Length_Accepted	Unsigned	R
Segmentation_Supported	BACnetSegmentation	R
Max_Segments_Accepted	Unsigned	R
APDU_Segment_Timeout	Unsigned	W
APDU_Timeout	Unsigned	W
Number_Of_APDU_Retries	Unsigned	W
Max_Master	Unsigned(1..127)	R
Max_Info_Frames	Unsigned	R
System_Status	BACnetDeviceStatus	R
Device_Address_Binding	List of BACnetAddressBinding	R
Active_COV_Subscriptions	List of BACnetCOVSubscription	R
UTC_Offset	Integer	W
Daylight_Savings_Status	Boolean	R
Local_Date	Date	R
Local_Time	Time	R
Time_Synchronization_Recipients	List of BACnetRecipient	W
UTC_Time_Synchronization_Recipiens	List of BACnetRecipient	W
Time_Synchronization_Interval	Unsigned	W
Align_Interval	Boolean	W
Interval_Offset	Unsigned	W
Configuration_Files	BACnetARRAY[N] of BACnetObjectIdentifier	R
Last_Restore_Time	BACnetTimeStamp	R
Slave_Proxy_Enable[1]	BACnetARRAY[N] of Boolean	W
Auto_Slave_Discovery[1]	BACnetARRAY[N] of Boolean	W
Manual_Slave_Address_Binding[1]	List of BACnetAddressBinding	W
Slave_Address_Binding[1]	List of BACnetAddressBinding	R

Table 28: Properties of the Device Object.
[1] Only available if the device is a BACnet/IP-BACnet MS/TP router.

18.3.2 Device Name and ID

The following properties of the Device object, which are part of every BACnet object, identify the device uniquely.

Object_Identifier (Read-Only). This property, of type *BACnetObjectIdentifier*, is a numeric code that is used to identify the object. For the Device object, the object identifier must be unique internetwork-wide.

The *Object_Type* part of the *Object_Identifier* of the Device object is 8 (= DEVICE). The instance part of the *Object_Identifier* of the Device object is configurable via the configuration UI (see Section 5.2.15). The default value is 17800.

Object_Name (Read-Only). The name of the object. The value of *Object_Name* of the Device object is configurable via the configuration UI (see Section 5.2.15). For the Device object, this name shall be unique within the BACnet internetwork.

Object_Type (Read-Only). The object's type. For the Device object, the value of this property is 8 (= DEVICE).

18.3.3 Device Information

A whole set of properties provides general purpose information about the device.

Vendor_Name (Read-Only). The value of this property is "LOYTEC electronics GmbH".

Vendor_Identifier (Read-Only). A numerical value identifying the BACnet vendor. The value of this property is 178.

Model_Name (Read-Only). The value of this property is equal to the product code of the device. Examples are "LINX-200" or "LINX-221".

Firmware_Revision (Read-Only). The value of this property gives the current BACnet module version used on the device.

Application_Software_Version (Read-Only). The value of this property gives the build date and the version of the current application on the device.

Location (Read-Writable). A string intended to be used to describe the physical location of the device, e.g., "1st floor". This property can be set via the configuration UI (see Section 5.2.15). The default value is "unknown".

Description (Read-Only). A string intended to be used to describe the device's purpose. This property can be changed via the configuration UI (see Section and 5.2.15).

Protocol_Version (Read-Only). The BACnet protocol version supported by the device. The value of this property is 1.

Protocol_Revision (Read-Only). The BACnet protocol revision of the BACnet version supported by the device. The value of this property is 6.

Protocol_Services_Supported (Read-Only). A string of bits marking which BACnet services can be executed by the device. For a detailed list of the BACnet services supported, please refer to the product's PICS document.

Protocol_Object_Types_Supported (Read-Only). A string of bits identifying which BACnet object types are supported by the device. For a detailed list of supported object types, please refer to the product's PICS document.

18.3.4 Object Database

The following properties provide information about the BACnet objects contained in the device.

Object_List (Read-Only). This property holds a *BACnetARRAY* of object IDs (object type, object instance pairs), one object ID for each object within the device that is accessible through BACnet services.

Database_Revision (Read-Only). This property, of type *Unsigned*, is a logical revision number for the device's object database. It is incremented when an object is created, an object is deleted, an object's name is changed, an object's Object_Identifier property is changed, or a restore is performed.

18.3.5 Protocol Parameters

BACnet protocol parameters are accessible via the properties listed below.

Max_APDU_Length_Accepted (Read-Only). The maximal size of an APDU (Application Protocol Data Unit) accepted by the device. The value of this property is 487 if BACnet MS/TP is used and 1476 if BACnet/IP is used. When the device can act as a router between BACnet/IP and BACnet MS/TP, the value of this property is 1476.

Segmentation_Supported (Read-Only). The value of this property indicates whether and which kind of segmentation is supported by a device. The value of this property is SEGMENTED_BOTH.

Max_Segments_Accepted (Read-Only). The maximum numbers of segments accepted by a device. The value of this property is 16.

APDU_Segment_Timeout (Read-Writable). Timeout in milliseconds allowed between segments. The value of this property is 2000 milliseconds by default. On MS/TP networks, this value should be increased to 40000 (40 sec).

APDU_Timeout (Read-Writable). Time in milliseconds the device waits for an answer before retrying or giving up on a request; also see *Number_Of_APDU_Retries*. The value of this property is 3000 milliseconds. On MS/TP networks, this value should be increased to 60000 (1 min).

Number_Of_APDU_Retries (Read-Writable). The number of times the device will try to re-send a packet before giving up on a request; also see *APDU_Timeout*. The value of this property is 3 by default.

Max_Master (Read-Writable). This property is only present if BACnet MS/TP is enabled. It defines maximal MS/TP MAC number at which the device expects an MS/TP master. The value of this property is configurable via the configuration UI (see Section 5.2.15) and must be in the range 1-127.. The default value of this property is 127.

Max_Info_Frames (Read-Writable). This property is only present if BACnet MS/TP is enabled. It defines the maximal number of MS/TP packets the device can send when it holds the MS/TP token. Increasing this value will increase latency on the MS/TP network. The value of this property is configurable via the configuration UI (see Section 5.2.15). The default value of this property is 1.

18.3.6 Diagnostics

Several properties provide run-time information about the device.

System_Status (Read-Only). The value of this property is always OPERATIONAL.

Device_Address_Binding (Read-Only). This property contains a list of bindings between BACnet device instance numbers (the instance number part of the Device object ID) and BACnet addresses. This property tells a user which BACnet address the device will actually use when trying to communicate with another device known only by its device instance number. This information can be helpful when diagnosing network configuration problems.

Important! ***A BACnet address consists of the BACnet network number, which is 0 for the local network, and the BACnet MAC address of the device.***

In particular problems exist, if two or more devices in the network have been wrongly assigned the same device instance number. In this case two BACnetAddressBinding entries with the same instance number but different BACnet addresses will be listed—provided the ambiguous instance number is in some way required by the device (e.g., by a client mapping).

Important! ***Bindings between device instance numbers and BACnet addresses are only listed in Device_Address_Binding if they are actually required by a given configuration, and are currently known or ambiguous.***

Slave_Address_Binding (Read-Only). This property is only present if the device is a BACnet/IP-BACnet MS/TP router. It lists bindings between BACnet MS/TP slave instance numbers (the instance number part of the slave's Device object ID) and BACnet addresses of slaves on the MS/TP network for which the device serves as a slave proxy, see Section 18.3.10 for details.

Active_COV_Subscriptions (Read-Only). This property lists currently active COV subscriptions.. Each entry of type *BACnetCOVSubscription* provides information about the recipient address, the monitored property ID, whether notification are confirmed or unconfirmed, the remaining time of the subscription, and optionally the COV increment.

Whenever the device receives a COV subscription via one of the services SubscribeCOV or SubscribeCOVProperty, a new entry is added to the list or an existing entry is updated (re-subscription). An entry is removed from the list when a subscription terminates, either because it times out or because it was actively unsubscribed by the subscriber.

18.3.7 Date & Time

The device's time and date are exposed to the network via the following set of properties.

UTC_Offset (Read-Writable). This *Integer* value specifies the time difference between local time and UTC in minutes. The value of this property is configurable via the configuration UI (see Section 5.2.1).

Important! ***Note that UTC_Offset is relative to local time and not relative to UTC, i.e., a time zone offset of GMT+1 (Berlin, Paris, Vienna) corresponds to UTC_Offset being set to -60 (minutes).***

Daylight_Savings_Status (Read-Only). This *Boolean* value indicates whether (TRUE) or not (FALSE) daylight saving correction of the local time is currently active. The daylight saving scheme is configurable via the configuration UI (see Section 5.2.1).

Local_Date (Read-Only). The current date according to the device's clock. The value of this property can be changed via the configuration UI (see Section 5.2.1).

Local_Time (Read-Only). The current time according to the device's clock. The value of this property can be changed via the configuration UI (see Section 5.2.1).

18.3.8 Time Master

The device can serve as a BACnet time master, i.e., it can issue TimeSynchronization and UTCTimeSynchronization request on time synchronization events. A time synchronization event occurs after rebooting, when the device's clock changes, or, if so configured, the event is generated periodically. The following properties are used to configure the time master. Use a BACnet operator workstation to write these properties over the BACnet network.

Time_Synchronization_Recipients (Read-Writable). This list of recipients will receive TimeSynchronization requests on time synchronization events. A recipient is either specified by its device ID (the object ID of its Device object), or its BACnet address. By default, this list is empty.

UTC_Time_Synchronization_Recipients (Read-Writable). This list of recipients will receive UTCTimeSynchronization requests on time synchronization events. A recipient is either specified by its device ID (the object ID of its Device object), or its BACnet address. By default, this list is empty.

Time_Synchronization_Interval (Read-Writable). The *Unsigned* value of this property specifies the time interval in minutes in which periodic time synchronization events are created. If set to zero, no periodic time synchronization events are generated.

The actual clock time at which periodic time synchronization events are generated is determined by the properties *Time_Synchronization_Interval*, *Align_Interval*, and *Interval_Offset*; Table 29 outlines how these properties interact.

Time_Synchronization_Interval	Align_Intervals	Periodc Time Synchronization Event At…
Multiple of 1440 (minutes), i.e., multiple of one day	TRUE	*Interval_Offset* minutes after midnight, every (*Time_Synchronization_Interval*/1440) days
Multiple of 60 (minutes) but *not* multiple of 1440 (minutes), i.e., multiple of one hour	TRUE	*Interval_Offset* minutes from the current* hour, every (*Time_Synchronization_Interval*/60) hours
Multiple of 1440 (minutes), i.e., multiple of one day	FALSE	*Interval_Offset* minutes from the current* minute, every (*Time_Synchronization_Interval*/1440) days
Multiple of 60 (minutes), but *not* multiple of 1440 (minutes), i.e., multiple of one hour	FALSE	*Interval_Offset* minutes from the current* minute, every (*Time_Synchronization_Interval*/60) hours
Neither multiple of 60 or 1440, but greater than zero	TRUE or FALSE	*Interval_Offset* minutes from the current* minute, every *Time_Synchronization_Interval* minutes
Zero	TRUE or FALSE	Never

Table 29: Periodic time synchronization events are parameterized by the properties *Time_Synchronization_Interval*, *Align_Interval*, and *Interval_Offset*.
* Current hour or minute refers to the hour or minute at which one of the properties *Time_Synchronization_Interval*, *Align_Interval*, and *Interval_Offset* is written, e.g., the hour or minute the device completes the boot process or one of these properties is modified via BACnet services.

By default, the value of *Time_Synchronization_Interval* is 1440 (minutes), i.e., one day.

Align_Intervals (Read-Writable). The *Boolean* value of this property determines whether or not periodic time synchronization events shall be anchored at the start of a day or hour (TRUE) or not (FALSE), provided *Time_Synchronization_Interval* is a multiple of a day (1440 minutes) or hour (60 minutes). Table 29 details on how this property influences generating periodic time synchronization events. The default value of this property is TRUE.

Interval_Offset (Read-Writable). While *Time_Synchronization_Interval* specifies the period in which time synchronization events are generated, the *Unsigned* value of this property determines the point of time in minutes within this interval at which the time synchronization event is actually triggered. If the value of *Interval_Offset* is larger than the value of *Time_Synchronization_Interval*, the remainder of *Interval_Offset* divided by *Time_Synchronization_Interval* is used. The default value of this property is 0.

18.3.9 Backup & Restore

The following properties are related to backup & restore procedures.

Configuration_Files (Read-Only). The contents of this property is an array of object IDs of File objects that can backed-up or restored during a BACnet backup or restore procedure. Outside a BACnet backup or restore procedure, this property is empty. After a BACnet backup or restore procedure has been initiated, it contains the object ID *(File, 0)*, i.e., the File object whose instance number is 0.

Last_Restore_Time (Read-Only). The *BACnetTimeStamp* of the last restore procedure.

18.3.10 Slave Proxy

A device configured as BACnet/IP-BACnet MS/TP router, can serve as a slave proxy i.e., the device can answer Who-Is broadcast requests with I-Am responses for BACnet MS/TP slaves which, by definition, cannot initiate any communication and, thus, cannot answer broadcasts. The following properties allow configuring and monitoring the slave proxy.

Slave_Proxy_Enable (Read-Writable). For each BACnet MS/TP port, this property contains a *Boolean* that allows a user to enable (TRUE) or disable (FALSE) the slave proxy for the given port. By default, the slave proxy is enabled on all MS/TP ports.

Auto_Slave_Discovery (Read-Writable). For each BACnet MS/TP port, the slave proxy is capable of auto-detecting slaves on the MS/TP network attached to the port. This auto-detection mechanism can be disabled (FALSE) or enabled (TRUE) by changing the *Boolean* values stored in this property. Aside from auto-detecting slaves, the presence of slaves can also be manually configured in the property *Manual_Slave_Address_Binding*. By default, slave auto-detection is enabled on all MS/TP ports.

Note: *Due to bandwidth and latency limitations on MS/TP networks, the auto-discovery process may initially take up to 10min. However, once, slaves have been discovered, slaves will be quickly re-discovered after reboots or power-outs since the slave proxy caches information about slaves found on the MS/TP networks. To speed up auto-detection of slaves newly added to an existing MS/TP network for which auto-detection is enabled, simply disable and then re-enable auto-detection on given MS/TP port, i.e., set Auto_Slave_Discovery for the port to FALSE and then back to TRUE.*

Manual_Slave_Address_Binding (Read-Writable). Aside from auto-detecting slaves, see *Auto_Slave_Discovery,* slave bindings can also be manually configured via this property. Each entry of this list is a BACnetAddressBinding, i.e., a pair consisting of a slave device’s instance number and its BACnet address. Note, that bindings in this list may not necessarily appear in the property *Slave_Address_Binding*, e.g., if for a given binding no physical slave is present at the given MS/TP MAC address. By default, this list is empty.

Important! ***Only use Manual_Slave_Address_Binding if the slave is not auto-detected. Note, that bindings in Manual_Slave_Address_Binding must contain the correct network number of the MS/TP network to which the slave is attached.***

Slave_Address_Binding (Read-Only). This property lists bindings of instance numbers and BACnet addresses of all slaves for which the slave proxy answers Who-Is requests. Thus, this property can be used to check if slaves have been auto-discovered or manually bound successfully. The property is also helpful in discovering network configuration issues involving slaves: If two or more slaves on the attached MS/TP networks have been erroneously assigned the same device instance number (the instance number of the slave's Device object), the given instance number will be listed accordingly often in this property.

18.3.11 Client Mapping CSV File

Client functionality for the BACnet server objects can be defined by so-called *client mappings*. These mappings basically specify whether present value properties shall be written to or polled from the BACnet network, and what the destination address and objects are. These definitions can be downloaded as a CSV file onto the device using FTP.

The CSV file must be named 'bacclnt.csv' and stored in the directory '/var/lib/bacnet' on the device. The file is read when the device boots. If any errors occur they are reported in '/tmp/bacclnt.err'.

The column format is shown in Table 30. Lines beginning with a hash ('#') sign are comment lines. The example values in Table 30 setup a client mapping named "Lamp Room 302", which writes (mapping type 2) the present value of the local object AI,4 to the remote object AO,1 on the device with the instance number 17801.

Column	Field	Example	Description
A	Description	Lamp Room 302	User-defined description of this client mapping. Can be left empty. Don't use commas or semi-colons in the text!
B	Local Object-Type	AI	The BACnet object type of the local server object (AI, AO, AV, BI, BO, BV, MI, MO, MV, ACCM, LOOP)
C	Local Object Instance Number	4	The object instance number of the above object.
D	Remote Device Instance	17801	The device object instance number of the remote BACnet device
E	Remote Object-Type	AO	The BACnet object type of the remote server object (AI, AO, AV, BI, BO, BV, MI, MO, MV, ACCM, LOOP)
F	Remote Object Instance Number	1	The object instance number of the above object.
G	Map Type	2	Defines the type of the mapping: 0=Poll, 1=COV, 2=Write, 3=Value
H	Interval	60	Defines the poll interval in seconds for poll/value mappings and the COV lifetime in seconds for COV mappings. Note: In previous versions this column was also used to specify the write priority for write mappings. This usage of column H is deprecated and column I should be used to specify priority.
I	Priority	8	For write and value mappings this defines the write priority (1..16). Omit this field or set it to '-1' to write w/o priority.
J	Local Property ID	45	Specifies the property ID of the local object, which is mapped to the remote object. If omitted, the Present_Value of the local object is mapped.
K	Remote Property ID	45	Specifies the property, which is written/read on the remote object. If omitted, the remote property ID is the Present_Value.

Table 30: CSV Columns of the BACnet Client Mappings File.

18.3.12 EDE Export of BACnet Objects

The BACnet server object configuration of the device is accessible as a set of CSV files following the EDE format convention. They can be downloaded via FTP from the directory '/data/ede' on the device. The files are

- lgate.csv: This is the main EDE sheet with the list of BACnet objects.
- lgate-states.csv: This is the state text sheet. For each state text reference in the main sheet, a line contains the state texts for this multi-state object.
- lgate-types.csv: This is the object types text sheet. The file contains a line for each object type number. Note, that lines for standard object types can be omitted.
- lgate-units.csv: This is the unit text sheet. The file contains a line for each engineering unit enumerator value. Note that lines for standard units can be omitted.

18.4 SNMP Interface

The Simple Network Management Protocol (SNMP) is a common protocol for monitoring and managing devices. SNMP is an "Internet-standard protocol" and is defined by the Internet Engineering Task Force (IETF). It is typically used in IT environments for server, network and supply management and monitoring.

SNMP allows querying status and statistics data from devices and also allows devices to alarm network management applications using SNMP traps. A managed device contains an SNMP agent which communicates with a management system using UDP. The SNMP agent holds collects and provides its data items in a tree. The data provided by an SNMP agent is defined by Management Information Bases (MIBs). These define the names and data types of the management data. Every data item is assigned an object ID (OID). A device can support an arbitrary number of MIBs, such as CPU statistics or network traffic statistics.

18.4.1 SNMP Features

LOYTEC devices supporting SNMP share these common features:

- Read-only access for SNMP version 2C and 3
- Standard MIBS: SNMPv2-MIB, SNMPv2-SMI, RFC1213-MIB, IF-MIB, IP-MIB, DISMAN-EVENT-MIB, HOST-RESOURCES-MIB, SNMP-FRAMEWORK-MIB, SNMP-MPD-MIB, SNMP-USER-BASED-SM-MIB, SNMP-VIEW-BASED-ACM-MIB,
- Option to expose OPC data points to SNMP.
- Option to create a device-specific MIB file.
- Option to send traps to a management system.

18.4.2 Configuration

The SNMP agent can be configured in the Web UI and in the configuration software. Figure 352 shows the Web interface. The settings in the configuration software are similar.

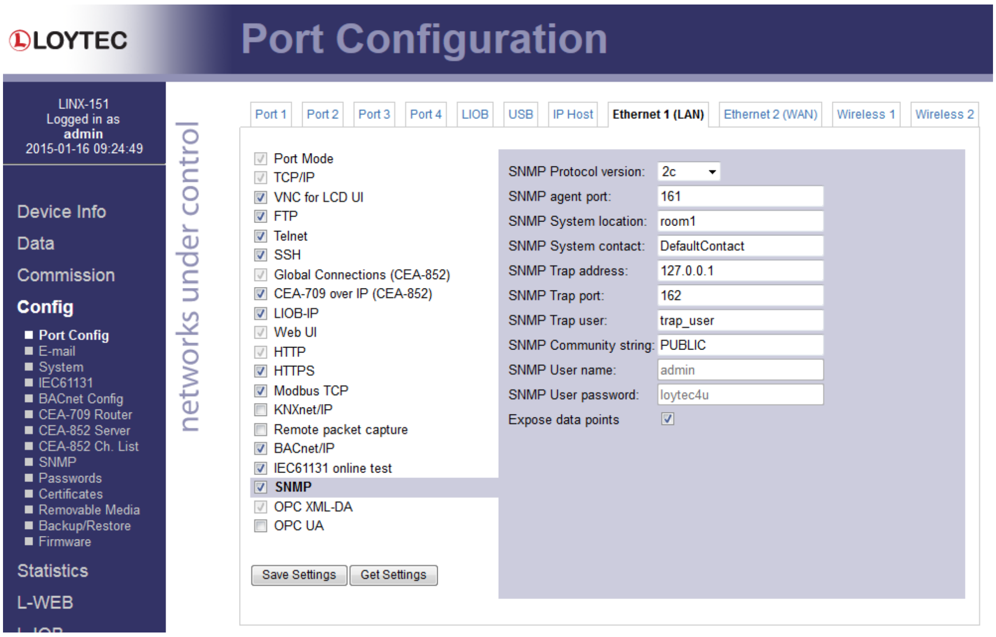


Figure 352: SNMP configuration page

The following settings are used to configure the SNMP agent:

- **SNMP Protocol version**: This setting selects between version 2C, 3 and 2C+3. Protocol version 2C is more common, but lacks encrypted authentication.
- **SNMP agent port**: This select the UDP port on which the SNMP agent listens. It is recommended to keep this port at its default setting, port 161.
- **SNMP System location:** This defines the value of the `SNMPv2-MIB:: sysLocation` OID. It is used to locate a device via SNMP.
- **SNMP System contact:** This defines the value of the `SNMPv2-MIB::sysContact` OID. It is used to identify the responsible contact persion for the deivce
- **SNMP Trap address:** This setting defines the destination IP address to which traps (alarms) are sent.
- **SNMP Trap port:** This setting defines the destination UDP port to which traps (alarms) are sent.
- **SNMP Trap user:** This setting defines the user name when sending traps (SNMP v3)
- **SNMP Community string:** This defines the (read) community string used for SNMP v2c.
- **SNMP User name:** This defines the user name required to access the SNMP agent (SNMP v3)
- **SNMP User password:** This defines the user password required to access the SNMP agent (SNMP v3).
- **Expose data points:** This switch allows to access data points exposed to OPC also to be accessed via SNMP.

18.4.3 Exposing Data Points to SNMP

The SNMP agent allows exposing data points to SNMP. It considers every data point which is exposed via OPC also to be exposed via SNMP.

As SNMP has several restrictions on what can be represented, the following mappings are made:

- **Binary data points**. Binary data points are mapped to the INTEGER type. FALSE is mapped to 0, TRUE is mapped to 1 and an invalid value is mapped to -1.

- **Analog data points**: SNMP has no standard way to represent floating point values, so their values are mapped to the STRING type. A value of "--" identifies an invalid value

- **Multistate data points**: Multistate data points are mapped to the STRING data type. Their values are represented by the multi-state text labels.

SNMP variable names have to be unique within their MIB, so data points with the same name in different folders are made unique by the following name scheme: `dpNNNNXUUUUUUUU`, e.g. `dpFreeMemoryX00000003`. NNNN is the data point name with all forbidden characters removed (only a-z, A-Z and 0-9 is allowed). UUUUUUUU is replaced with the unique ID of the data point.

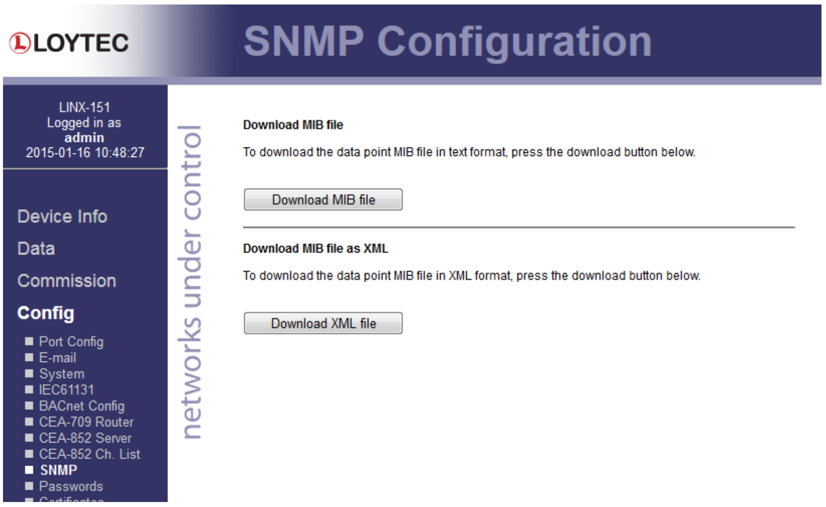


Figure 353: Downloading device-specific MIB files

Figure 353 shows the Web UI page which allows downloading the device specific MIB file. The "Download MIB file" buttons generates a MIB file which can be used by a network management tool. The "Download XML file" button generates an XML-encoded representation of the MIB contents.

Note that the MIB files are dependent on the data point configuration, so that changes in the data point configuration will change the MIB contents.

18.4.4 Alarming

The SNMP agent can send a trap if an alarm occurs in a generic alarm server. To connect the generic alarm server to the SNMP agent, it has to report to the special SNMP technology alarm server. The configuration steps in the **Create Alarm Server** dialog are shown in Figure 354.

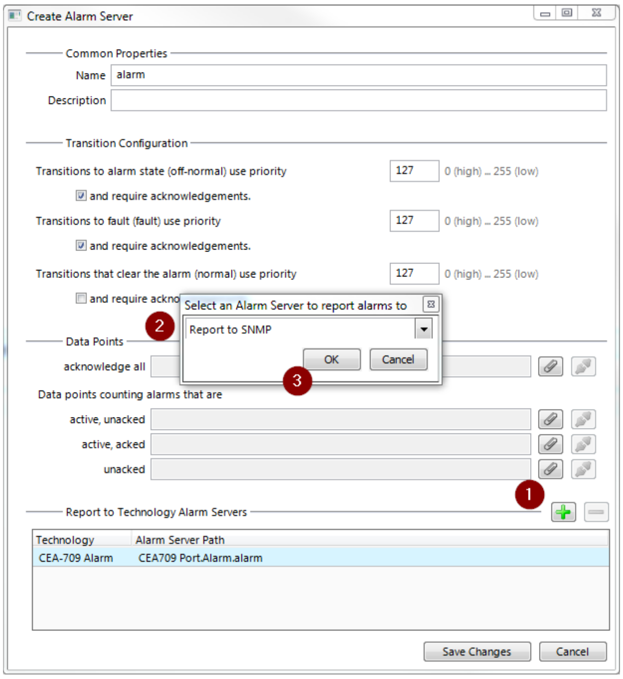


Figure 354: Report to SNMP

18.4.5 SNMP Security

As SNMP provides access to internal device information which could be exploited for an attack, SNMP should be used only in internal, non-critical environments.

SNMP Version 2C uses unencrypted authentication and payload. The community string is transmitted in clear text and can be easily extracted from captured network traffic.

SNMP Version 3 supports encrypted authentication and payload encryption. LOYTEC devices support only authentication. The password is not transmitted in clear text then.

LOYTEC devices do not support write accesses via SNMP.

19 Network Media

19.1 FT

The device's FT port is fully compatible to the parameters specified by LONMARK for this channel. FT ports can also be used on Link Power (LP-10) channels. However, the device does not provide the power supply for Link Power channels.

When using the Free Topology Segment feature of the FT, only one termination (Figure 355) is required and can be placed anywhere on the free topology segment. Instead of building the termination, one can order the L-Term module (LT-33) from LOYTEC, which can be used to properly terminate the bus.

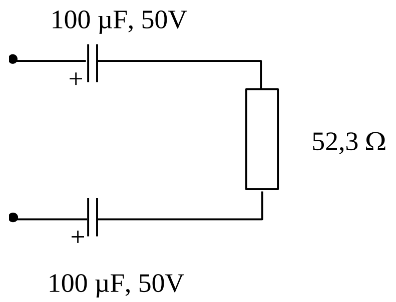


Figure 355: FT Free Topology Termination.

In a proper bus topology, two terminations are required (Figure 356). These terminations need to be placed at each end of the bus. Here, also L-Term modules can be used at either end.

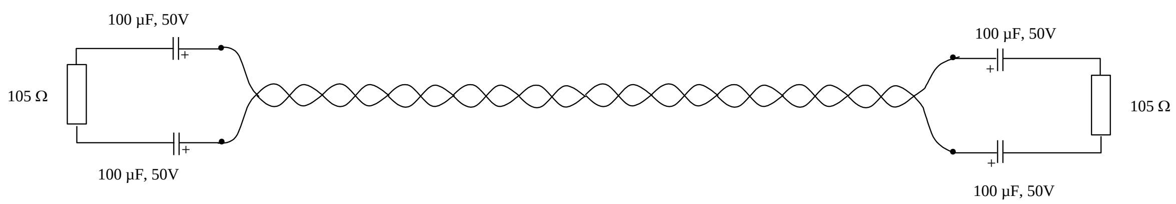


Figure 356: Termination in an FT Bus Topology.

19.2 M-Bus

The device uses the RS-232 console interface for the connection to an external M-Bus transceiver (repeater). The M-Bus specifies no special topology requirement, though it is not advised to use a ring topology. A maximum of 250 M-Bus slave devices can be connected to the bus, in fact, the external bus transceiver can have a lower limit of connected devices. Please refer to the datasheet of the transceiver used for more detailed information. The usual cabling is a standard telephone wire (JYStY N*2*0.8 mm). The maximum distance between a slave and the repeater is 350 meters at Baud rates from 300-9600 Baud; by limiting the lower Baud rates and using fewer slaves, this limit can be increased. Additionally, it must be ensured that the bus voltage does not fall below 12 V. The maximum cable length of the system must not exceed 1000 m (maximum cable capacitance of 180 nF).

19.3 Modbus RS-485

The Modbus RS-485 port of the device is an electrical interface in accordance to EIA-485. The topology of the Modbus RS-485 consists of a trunk cable, along which the devices are connected either directly or via short derivation cables. Without repeater a maximum of 32 slave devices with full RS-485 unit load can be connected to the bus. If RS-485 transceivers of slave devices are assured to pose only 1/2, 1/4 or even 1/8 unit load, a maximum of 64, 128, 256 slaves can be operated without a repeater, respectively. Alternatively, a repeater may be used. In any case, a maximum of 247 slaves can be addressed.

Each Modbus RS-485 network segment must be properly terminated with an LT-04 network terminator connected at each of the two ends of the segment media. Some Modbus slave devices might require a biasing terminator to guarantee a defined level on the wire when being idle. Use an LT-B4 instead of the LT-04 on one end in this case. Figure 357 shows an example Modbus RS-485 network configuration including biasing network termination.

The maximum length of the cabling depends on the Baud rate used. Without repeater a maximum length of 1000 meters is possible at 9600 Baud. According to the Modbus standard the derivation cabling must never exceed 20 m.

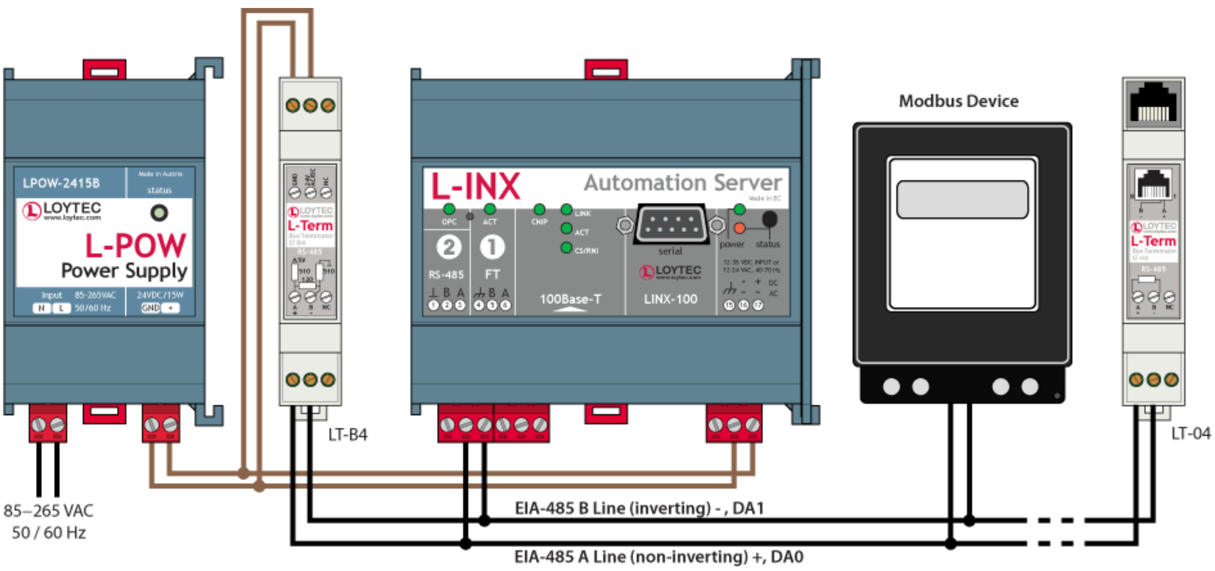


Figure 357: Modbus RS-485 network

19.4 MS/TP

MS/TP is an RS-485 protocol and usually needs three wires (negative, positive, and reference). Polarity must be connected correctly. When using 2-wire MS/TP, earth ground must be connected to the negative terminal of the power supply. Never connect the positive terminal of the power supply to earth ground! See Section 4.9 for wiring instructions. Each MS/TP network segment must be properly terminated. Use an LT-04 network terminator connected at each of the two ends of the segment media.

The RS-485 transceiver of the device represents a full-load on the RS-485 bus. Consequently, a minimum of 31 devices are supported on the MS/TP channel. More devices may be possible, if they represent half-load or quarter-load. Please consult the third-party documentation. If more MS/TP devices need to be connected, use an RS-485 repeater to separate them electrically.

Logically, the MS/TP bus supports up to 255 devices. Each MS/TP device must be assigned a unique MAC address. Up to 127 MS/TP masters can be connected. Make sure, that the Max_Master setting includes the highest MS/TP master MAC address.

For operation of some slower devices on the MS/TP network it is recommended to set the following properties of the device object to fine-tune communication on the network:

- APDU_Timeout = 60000 (1 min).
- APDU_Segement_Timeout = 40000 (40 sec).
- Optionally, disable MS/TP slave proxy if not needed in order to optimize bandwidth usage: Slave_Proxy_Enable = { False }.

19.5 Redundant Ethernet

19.5.1 Ethernet Cabling Options

The L-INX models 12X/15X/22X and the LGATE-950 are equipped with two Ethernet ports, which are connected to an internal Ethernet switch. This allows for advanced cabling options to reduce cabling costs or to increase network resilience. For this discussion, the term *upstream* is used to designate the direction towards the network, which the devices are connected to. Likewise, the term *downstream* is used to designate devices more distant to the network which the devices are connected to.

Redundant cabling options are enabled by the Rapid Spanning Tree Protocol (RSTP) which is implemented in most managed switches. Please note, that this is a feature of the switch, not of the L-INX or the L-GATE, so that LOYTEC cannot give a guarantee that this will work with a particular switch model. In no case redundant cabling options will work with unmanaged switches. The older Spanning Tree Protocol (STP) should not be used for this type of application, as it converges too slowly.

Star topology: In the most basic setup, a device is connected to an Ethernet switch with one cable. This is called a star cabling because all devices are connected to a common upstream device. In this setup, the cable and the switch are single point of failures.

Chain topology: Because the L-INX/L-GATE itself acts as an Ethernet switch, this device can be connected to a chain. This is a special form of the star topology. Its advantage is the reduced cabling costs. The disadvantage is the connection loss to downstream devices when an upstream device is powered-off, reset or removed. Also, the Ethernet bandwidth (100 MBit/s) is shared among all members of the chain. The last device has one unused Ethernet port, as it is not allowed to create Ethernet loops without STP. The recommended maximum number of daisy-chained devices is 20.

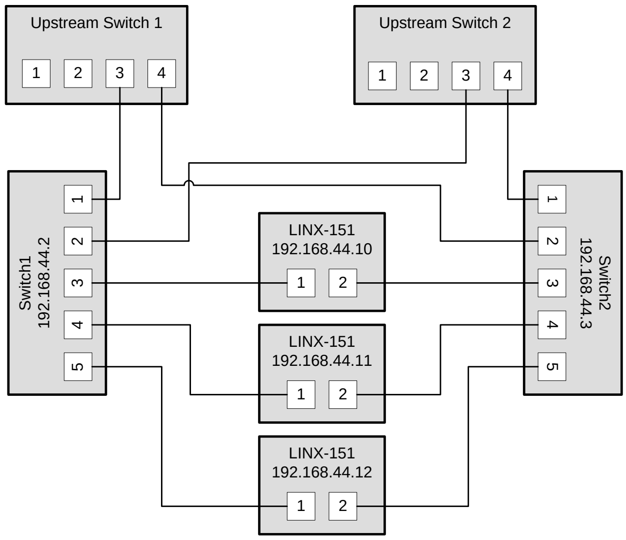


Figure 358: Fully redundant Ethernet topology

Fully redundant topology: Both Ethernet ports are connected to a different upstream switch. Thus, a single cable or upstream switch problem can be tolerated. This topology requires RSTP. In Figure 358, the LINX-151 with IP addresses 192.168.44.10 to 192.168.44.12 are connected in this way. This connection scheme increases switch and cabling costs, but increases network resilience. Note that the upstream network is connected via the lowest-numbered ports. If this is not possible, the ports need to be configured to the lowest STP port priority value (which is the highest priority).

Ring topology: In this setup, the devices are connected in a chain and each end of the chain is connected to a different upstream switch. This topology requires RSTP. If a single device is powered off, the RSTP will automatically recalculate the spanning tree so that all other devices in the chain are reachable. Only if two devices are power-off at the same time, the devices between them will not have an Ethernet connection. In Figure 359, the L-INX devices with IP addresses from 192.168.44.10 to 192.168.44.12 are connected in this way. The recommended maximum number of daisy-chained devices is 20.

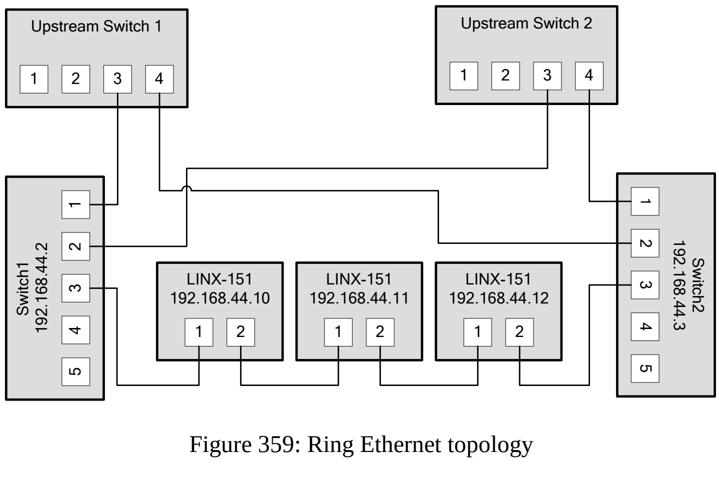


Figure 359: Ring Ethernet topology

19.5.2 Upstream Options

In case of redundant switches, there are two possible upstream topologies:

Single upstream connection: Switch1 (or Switch2, but not both) is connected to the upstream network while Switch2 only provides a redundant path to the Loytec devices. The redundant path is created by a direct Ethernet cable between Switch1 and Switch2 which needs to be plugged into a lower-numbered port than the L-INX devices are connected to. If this is not possible, the STP port priority for the cross-connection cable needs to be set to a low value. The RSTP domain should be restricted to Switch1 and Switch2. This can be done by enabling a BPDU filter on the port on Upstream Switch 1. This will block all RSTP packets to enter the upstream network. A sample setup for this topology is shown in Figure 360.

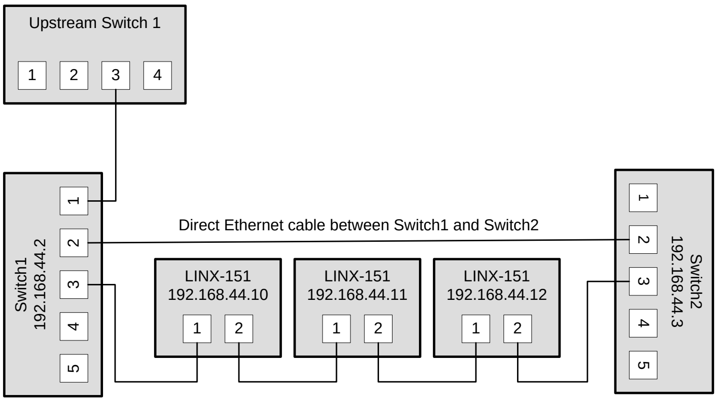


Figure 360: Single upstream connection.

Redundant upstream connection: Switch1 and Switch2 are both connected to the upstream network, either to two ports on the same switch or to two redundant upstream switches. In this case, RSTP is needed to ensure a loop-free topology between the upstream switches, Switch1 and Switch2, so the RSTP domain includes the upstream network and the chained L-INX/L-GATE devices. The configuration of Switch1 and Switch2 need to ensure that they are not selected as the root bridge. If possible device communication should be bound to a separate VLAN and MSTP (Multiple Spanning Tree Protocol) should be employed to isolate the spanning tree operations. This topology is shown in Figure 358.

19.5.3 Preconditions

For the fully redundant and ring topology, the following preconditions have to be met:

- The upstream switches have to support the Rapid Spanning Tree Protocol (RSTP), as defined in IEEE 802.1w.
- The upstream switches have to provide a broadcast storm filter.
- Two distinct switches are required for each end of the device chain.
- Both upstream switches are connected to the same Ethernet network.

19.5.4 Switch Settings

The switches which connect the devices to the network need the following settings. Note that these are only recommendations or starting points. Each network with redundant connections needs testing and verification to prevent network loops.

- The STP bridge must be enabled.
- The STP bridge priority should be set to the minimum (61440), so that these switches are not elected as root bridges.
- The bridge mode should match the upstream bridge modes, preferable 802.1s or 802.1w.

If the upstream network uses RSTP, the timing parameters of the upstream networks must be used. Else the timing parameters should be set to minimum values for fast convergence:

- Bridge max age time: 6 seconds
- Hello time: 1 seconds
- Forward delay: 4 seconds
- All ports connected to Ethernet rings have to be configured as NON-EDGE ports, so that the RSTP can detect loops
- The switches should be configured to block broadcast storms. A recommended rate is 5% or 3000 packets/seconds.

The upstream switches need the following configuration:

- If a single upstream connection is used, the connected port on the upstream switch should have BPDU filtering enabled.
- If redundant upstream connections are used, the connected ports on the upstream switches should have a BPDU root guard enabled.

19.5.5 Testing

When the switches are configured and the devices are connected, the following tests are recommended. These tests are important to confirm that the STP changes due to topology changes to not interfere with the rest of the network.

- Check that no broadcast storms are sent into the upstream network by capturing traffic between Switch1, Switch2 and the Upstream switch. This test should be done continuously, especially during switch and device power cycles.
- Check that all devices can be reached (ICMP ping).

Execute these tests for these conditions:

- Power up all switches and devices. Wait until all devices are up, then test.
- Power-off Switch1. Wait approx. 10 seconds, then test.
- Power-on Switch2, power-off Switch1. Wait until Switch2 has booted, then test.
- Power-on Switch1. Wait until Switch1 has booted, then test.
- Reboot all L-INX and L-GATE devices. Wait until the devices have booted, then test.
- Remove a single Ethernet cable. Wait approx. 10 seconds, then test. This test should be repeated for different cables. Make sure that at least the following connections are tested:
    - The connection between Switch1 and the L-INX directly connected to Switch1.
    - The connection between Switch2 and the L-INX directly connected to Switch2.
    - A connection in the L-INX chain which is not connected directly to either Switch1 or Switch2.

19.5.6 Example switch configuration

The following example shows the configuration commands for Switch1, Switch2 and the upstream switch (HP Procurve syntax) in the setup shown in Figure 358.

Upstream switches:

```
config
spanning-tree
spanning-tree priority 8
spanning-tree 3,4 root-guard
spanning-tree hello-time 1
spanning-tree forward-delay 4
spanning-tree maximum-age 6
exit
```

Switch1 and Switch2:

```
config
spanning-tree
spanning-tree priority 15
spanning-tree 1,2 port-priority 0
spanning-tree 3-5 port-priority 8
spanning-tree hello-time 1
spanning-tree forward-delay 4
spanning-tree maximum-age 6
exit
```

19.6 WLAN

19.6.1 Introduction

Devices supporting the LWLAN-800 wireless adapter can be connected to IEEE 802.11 wireless networks. The following operation modes are supported:

- **Client mode (separate network)**: The WLAN client connected to an existing access point. The firewall of the WLAN interface can be configured to provide only a subset of the services of the device. For example, the WLAN interface could expose the Web UI, but not BACnet communication.

- **Access point mode (separate network)**: In the isolated access point mode, a client can connect to the wireless network created by the device. The device will assign an IP address to the client and will redirect all traffic to itself. This mode is used to configure a device with a mobile device.

- **Access point mode (bridged)**: In the bridged access point mode, a client can connect to the access point and also can use the network devices on the bridged Ethernet device. In this mode, the DHCP server is deactivated to avoid interference with an existing DHCP server in the Ethernet network.

- **Mesh point (separate network)**: This mode is used to create an IEEE 802.11s mesh network. Mesh points communicate with other mesh points in their radio vicinity and automatically choose the best route. Mesh networks can be used to extend the range of a wireless network or to create redundant radio links.

- **Mesh point (bridged)**: This mode is like the mesh point mode and also bridges the mesh point to an Ethernet network. Thus devices in the Ethernet network can communicate with devices in the mesh network. Only one mesh point should be in the bridged mode to avoid network loops.

The LWLAN-800 interface can use two WLAN functions at the same time. This can be used for advanced setups, like:

- Wireless 1 is used as an access point for configuring the device, while the Wireless 2 interface is used to participate in a mesh network.

- Wireless 1 is used as a bridged access point for configuring the device and the devices on the Ethernet network while Wireless 2 connects to another wireless network to reach a remote device.

However, there are restrictions when using both interfaces at the same time:

- Both functions need to use the same radio band.

- Both functions need to use the same channel.

19.6.2 802.11s Mesh Networking

WLAN client and access point modes are similar to other devices using 802.11 wireless networks. This section explains the features and benefits of the 802.11s network.

A mesh network removes the roles of clients and access points. Every node in a mesh network can send and receive data, as in a normal wireless network. However, every mesh node also routes packets to other mesh nodes. It observes the signal strength to all reachable nodes and distributes this information to other mesh nodes. Thus, the mesh network can transmit data between nodes with are not in their radio vicinity. In this case, a path between sender and receiver is selected and the intermediate nodes transmit the packet over several hops.

As the signal strenght and thus the range of a node can change over time, as well as nodes can be added and removed, the best path can change. The 802.11s routing protocol takes this into account and changes paths dynamically.

802.11s also provides strong encryption using the AuthSAE (Simultaneous Authentication of Equals) protocol, so that each pair of mesh nodes use an encrypted link. It is resistant to passive, active and dictionary attacks, given a strong pre-shared key.

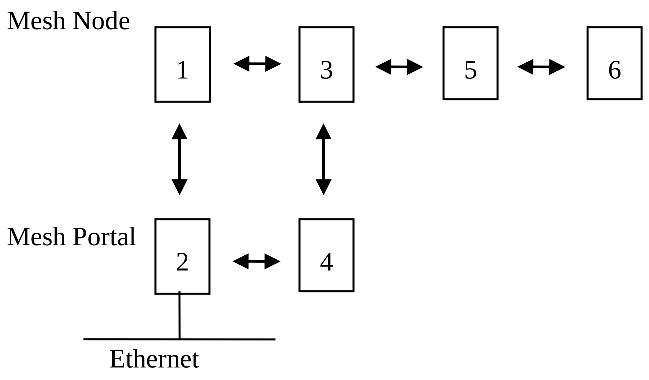


Figure 361: Mesh Networking

Figure 361 shows the roles of mesh nodes and possible links. Mesh point 1 can communicate with point 2 and point 3. It learns that the mesh point 2 is the mesh portal, so all traffic leaving the mesh network is automatically routed towards mesh point 2.

Mesh point 4 has mesh point 2 and 3 in its radio vicinity, but cannot communicate directly with mesh point 1. So mesh points 1 to 4 have two ways to reach each other and can tolerate the failure of a single node. This makes a mesh network resilient to node failure or fading radio links.

Mesh point 6 is an example on how mesh networks can be used to extend radio range. If point 2 communicates with point 6, there are two possible paths: 2-4-5-6 and 1-3-5-6. It selects the better path and mesh point 5 will extend the network range.

This example shows that every additional mesh point can make the network more resilient to failures or can extend the range far beyond the range of a single radio.

19.6.3 Hardware Installation

Connect the LWLAN-800 interface to the device with a USB cable, and then power the device. Do not remove the interface during operation.

The LWLAN-800 supports two antennas which should be mounted outside any metallized housing.

20 Firmware Update

The L-INX and L-GATE firmware support remote upgrade over the network and the serial console.

To guarantee that the device is not destroyed due to a failed firmware update, the firmware consists of two images:

1. The fallback image,
2. the primary image.

The fallback image cannot be changed. Thus, if the update of the primary image fails or the image is destroyed by some other means, the fallback image is booted and allows reinstalling a valid primary image. When the device boots up with the fallback image, the all port LEDs are flashing red.

The following firmware images are available for the different device models:

- LINX-10X/11X/20X/21X: linx_10x_11x_20x_21x_x_y_z.dl,
- LINX-12X/15X/22X/LGATE-95X: linx_12x_15x_22x_lgate95x_x_y_z.dl,
- LGATE-900: lgate900_x_y_z.dl.

20.1 Firmware Update via the Configurator

The primary image can be updated using the Configurator. For this purpose, it is recommended to have the device connected to the Ethernet and to have a valid IP configuration (see Section 5.2.2). The Configurator must be installed (see Section 7.1).

To Update the Firmware using the Configurator

1. Start the Configurator from the Windows Start menu: Start → Programs → LOYTEC LINX Configurator → LOYTEC LINX Configurator.
2. Select the menu: **Connection → Connect via FTP**. This opens the FTP connection dialog as shown in Figure 362.

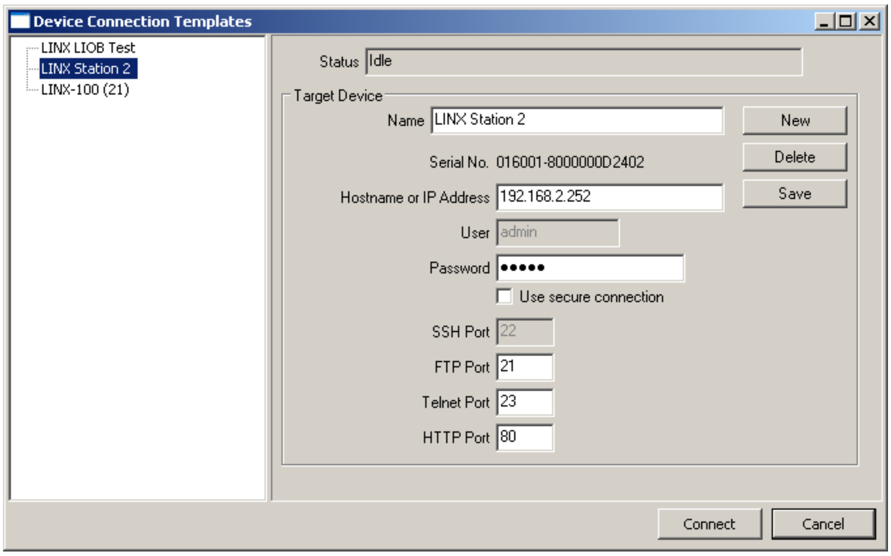


Figure 362: FTP connection dialog.

3. In the connection dialog, enter the IP address of the device as well as the admin user's password. The default password is 'loytec4u' (older firmware versions used 'admin'). This can be changed via the Web interface (see Section 5.1) and reset via the console UI (see Section 21.2.2).

4. If the device uses other port settings than the standard settings or the device is operated behind a NAT router, adapt the FTP and Telnet ports accordingly.

5. Click on **Connect**.

Note: *Alternatively, one can also connect via LNS. A firmware upgrade over an FT-10 channel, however, needs a lot more time to complete than over IP.*

---

6. Optionally, check for updates by selecting the menu **Help → Check for updates …**. This function checks for new firmware and Configurator versions.

7. Select the menu: **Firmware → Update …**

8. This opens the **Firmware Update** dialog as shown in Figure 363. Click on the button ... and select the firmware image.

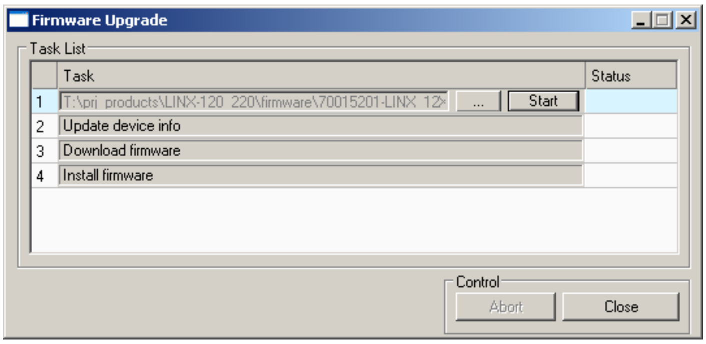


Figure 363: Firmware Update dialog of the Configurator.

9. Click on **Start** and observe the download progress.

10. When the download is complete, a dialog appears. Click **OK**.

11. In the Firmware Update dialog, click **Close**.

12. The device's firmware has now been successfully upgraded.

20.2 Firmware Update via the Console

On device models with the console connector it is possible to upgrade the device over this interface. To download the firmware via the console interface, the device must be connected to the RS-232 port (EIA-232) of a PC via its console interface as described in Section 21.2.1. You will need the LOYTEC serial upgrade tool (LSU Tool), which can be downloaded from our homepage at www.loytec.com.

Please make sure that the console shows the main menu. Otherwise navigate to the main menu or simply reset the device.

To Upgrade via the Console

1. Double click on the *.dlc file that comes with the new firmware package. This should start the LSU Tool and load the firmware image referenced in the dlc file. Please note that the dlc file and the dl file must be stored in the same folder. The start window of the LSU tool is shown in Figure 364.

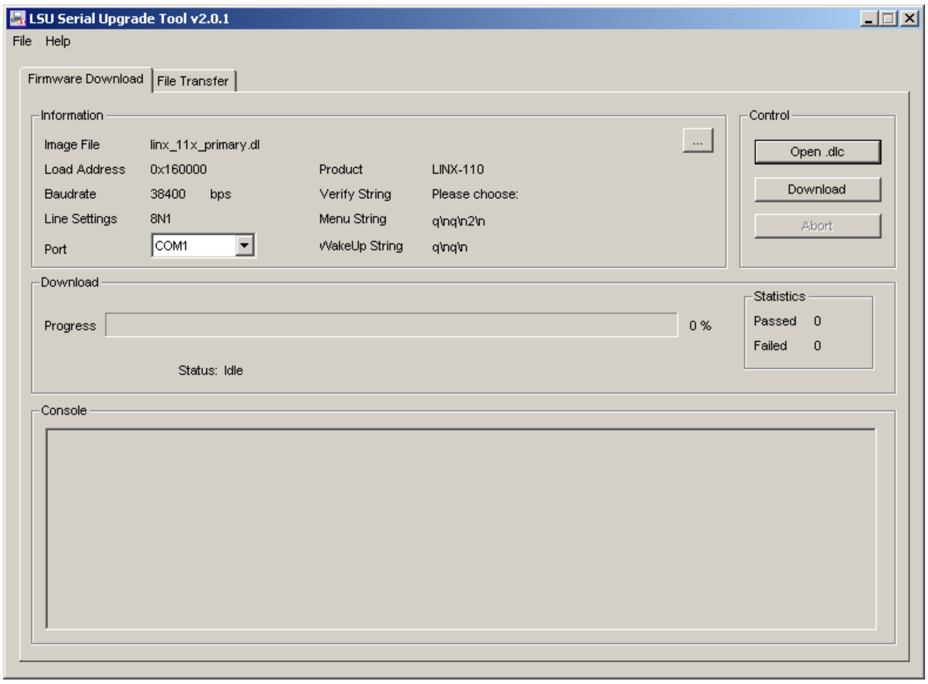


Figure 364: LSU Serial Upgrade Tool in Idle Mode.

2. If the device is not connected to COM1 you can change the port to COM2, COM3, or COM4. Make sure that the product shown under "Product" matches the device you are upgrading. Press **Download** to start the download. A progress bar as shown in Figure 365 can be seen.

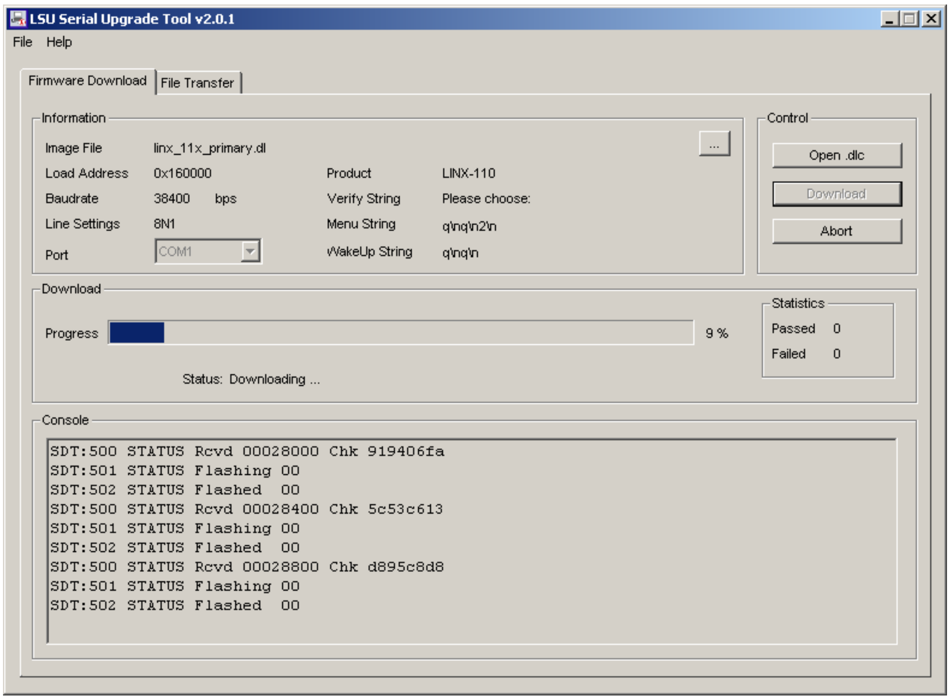


Figure 365: Progress Bar during Firmware Download.

3. If the upgrade is successful, the following window appears (Figure 366).

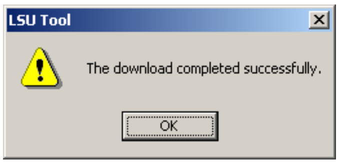


Figure 366: Successful Firmware Upgrade.

4. Double check that the new firmware is executed by selecting ‘1’ and pressing **Enter** in the console window. This will bring up the device information which shows the current firmware version.

20.3 Firmware Update via the Web Interface

The device’s firmware can also be upgraded using the Web interface. This option can be found in the **Config** menu under the **Firmware** item. For more details see Section 5.2.28.

21 Troubleshooting

21.1 Technical Support

LOYTEC offers free telephone and e-mail support for the L-INX product series. If none of the above descriptions solves your specific problem please contact us at the following address:

LOYTEC electronics GmbH
Blumengasse 35
A-1170 Vienna
Austria / Europe

e-mail : ***support@loytec.com***
Web : ***http://www.loytec.com***
tel : ***+43/1/4020805-100***
fax : ***+43/1/4020805-99***

or

LOYTEC Americas Inc.
N27 W23957 Paul Road
Suite 103
Pewaukee, WI 53072
USA

e-mail: ***support@loytec-americas.com***
Web: ***http://www.loytec-americas.com***
tel: ***+1 (512) 402 5319***
fax: ***+1 (262) 408 5238***

or

LOYTEC Asia Corporation Ltd.
16F.-3, No. 155, Zhongyang Rd
Xindian District
New Taipei City 23150
Taiwan

e-mail: ***support-asia@loytec.com***
tel: ***+886 (2) 8913 7838***
fax: ***+886 (2) 8913 7830***

21.2 Statistics on the Console

21.2.1 Connecting to the Console

Use a PC terminal program with the communication settings set to 38,400 bps / 8 data bits / no parity / 1 stop bit / no handshake. To connect COM1 of the PC to the Console on the device, use a standard null-modem cable with full handshaking. Power up the device or press **Return** if the device is already running. The menu shown in Figure 367 should appear on the terminal.

```
Device Main Menu
================

[1]  Show device information
[2]  Serial firmware upgrade
[3]  System configuration
[4]  CEA-709 configuration
[5]  IP configuration
[6]  CEA-852 device configuration
[7]  CEA-852 server configuration
[8]  Reset configuration (factory defaults)
[9]  Device statistics
[c]  Modbus cofiguration

[a]  Data Points

[0]  Reset device

Please choose:
```

Figure 367: Console Main Menu.

21.2.2 Reset configuration (load factory defaults)

Select item ‘8’ in the console main menu. This menu item allows resetting the device into its factory default state. The menu appears as shown in Figure 368.

```
Reset Configuration Menu
========================

[1]  Reset everything to factory defaults
[3]  Reset all passwords
[4]  Clear data point configuration

[q]  Quit

Please choose:
```

Figure 368: Reset to Factory Defaults Menu.

Select option ‘1’ to reset the entire device to factory defaults (including error log, configuration files, passwords etc.). Select option ‘3’ to reset all passwords (Web interface, FTP server etc.) to factory defaults.

Select option ‘4’ to clear all configured data points, such as CEA-709 network variables or user registers. This effectively clears the entire port configuration. The device must be rebooted to let the changes take effect.

Note: *This option does not reset the configuration of the built-in CEA-709 router. The nodes connected by the router are still reachable after clearing the data point configuration.*

21.2.3 Device Statistics Menu

Select ‘9’ from the device main menu to get to the device statistics menu. This menu holds relevant information regarding the device statistics of the device. This section describes those statistics, which are not available on the Web UI. The device statistics menu is shown in Figure 369. Use this menu only for debugging purposes. There is no need to access this menu if the network is running smoothly.

```
Statistics Menu
===============

[4]  Show IP statistics
[8]  Show DPAL statistics
[9]  Show Reg DPAL statistics
[b]  BBMD communications test

[q]  Quit

Please choose:
```

Figure 369: Device Statistics Menu on the Console.

21.2.4 IP statistics

A sample console output is shown in Figure 370.

```
*********** INTERFACE STATISTICS ************
***** lo0 *****
Address:127.0.0.1
Flags: Up Loopback Running Multicast
Send queue limit:50   length:0    Dropped:0
***** eth0 *****
Address:192.168.0.2     Broadcast Address:192.168.0.255
Flags: Up Broadcast Running Simplex Multicast
Send queue limit:50   length:0    Dropped:0
Network Driver Stats for CS8900 :
        rx ready len -        50       rx loaded len -          0
          rx packets -       931          tx packets -        165
            rx bytes -     78480            tx bytes -      13627
       rx interrupts -       931       tx interrupts -        165
          rx dropped -         0          rx no mbuf -          0
       rx no custers -         0  rx oversize errors -          0
       rx crc errors -         0     rx runt errors -          0
    rx missed errors -         0               tx ok -        165
       tx collisions -         0       tx bid errors -          0
  tx wait for rdy4tx -         0           tx rdy4tx -          0
  tx underrun errors -         0          tx dropped -          2
          tx resends -         0      int swint req -       2094
      int swint res -       2094         int lockup -          0
          interrupts -      3189

************ MBUF STATISTICS ************
mbufs: 512    clusters:  64    free:  14
drops:   0       waits:   0  drains:   0
      free:461          data:51          header:0           socket:0
       pcb:0          rtable:0           htable:0           atable:0
    soname:0           soopts:0           ftable:0           rights:0
    ifaddr:0          control:0          oobdata:0

************ IP Statistics ************
             total packets received         922
 datagrams delivered to upper level         922
    total ip packets generated here         158

Destination     Gateway/Mask/Hw    Flags     Refs     Use Expire
Interface
default         192.168.0.1        UGS          6       0      0 eth0
62.178.55.77    192.168.0.1        UGH          0       1   3606 eth0
62.178.95.96    192.168.0.1        UGH          0       1   3606 eth0
81.109.145.243  192.168.0.1        UGH          0       1   3606 eth0
81.109.251.36   192.168.0.1        UGH          0       1   3606 eth0
127.0.0.1       127.0.0.1          UH           0       0      0 lo0
130.140.10.21   192.168.0.1        UGH          1       6      0 eth0
192.168.0.0     255.255.255.0      U            0       0      3 eth0
192.168.0.1     00:04:5A:26:96:1F  UHL          7       0   1722 eth0
213.18.80.166   192.168.0.1        UGH          1     148      0 eth0
************ TCP Statistics ************

************ UDP Statistics ************
                total input packets         924
               total output packets         158

************ ICMP Statistics ************
```

Figure 370: IP Statistics.

The IP statistics menu has the additional feature of displaying any IP address conflicts. If the device’s IP address conflicts with another host on the network, the banner shown in Figure 371 is displayed.

```
WARNING: Conflicting IP address detected!
         IP address 10.125.123.95 also used by device with MAC address
         00 04 5A CC 10 41!

Clear IP conflict history (y/n):
```

Figure 371: IP Address Conflict.

As useful information, the MAC address of the conflicting host is shown. If the information about this conflict shall be cleared, enter ‘y’. If ‘n’ is selected, the conflict will show up again the next time this menu is entered.

21.2.5 BBMD Communications Test

This statistics menu provides a simple test for the user to determine, which of the IP addresses in the BDT are reachable over IP. The test uses a simple ping method on all IP addresses of the BDT. A sample result is shown in Figure 372. IP addresses, which reply to the ping request are shown as 'OK'. Others, that suffer from an error show 'FAILED' including a comment on what the problem was.

```
BBMD Communications Test
============================

Address                    Result  RTT(ms)  Comment
------------------------------------------------------------------------
10.102.77.77:47808         OK      2
10.102.77.78:47808         FAILED  n/a      No ping reply.
10.102.77.79:47808         OK      1
10.102.77.80:47808         OK      1
```

Figure 372: BBMD communications test.

21.3 Packet Capture

21.3.1 Configure Remote Packet Capture

Remote packet capture is able to capture packets on the Ethernet port and on the MS/TP port. The MS/TP remote packet capture option is only available, if the MS/TP port is enabled on the device (see Section 5.2.17). To enable the remote packet capture feature, go to the **Ethernet** port configuration and enable **Remote packet capture** as shown in Figure 373.

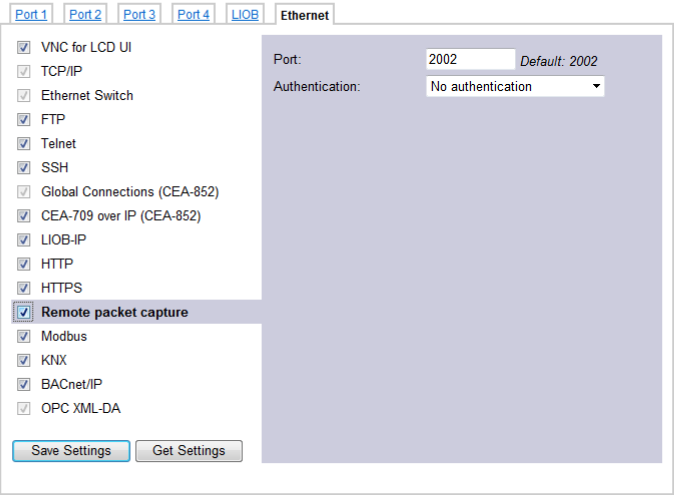


Figure 373: Remote packet capture port configuration.

The default **Port** setting may be changed to the desired port. Normally, this can be left at its default. If **No authentication** is selected, the device will allow incoming capture connections without requiring any credentials. If **Username and Password** is selected as authentication method, the client Wireshark will be required to provide valid credentials before the capture session can be started. Note, that only the users **admin** and **operator** are allowed to connect if this authentication method is selected.

Click the **Save Settings** button to save the configuration. The changes take effect and do not require to reboot the device. The remote capture can also be disabled again without a reboot.

21.3.2 Enable Local Capture

The device provides a local capture feature. With local capture enabled the device logs packets to an internal ring buffer. The log can be downloaded from the Web interface. To verify that the device is set up correctly, go to **Statistics → Packet capture** as shown in Figure 374**.**

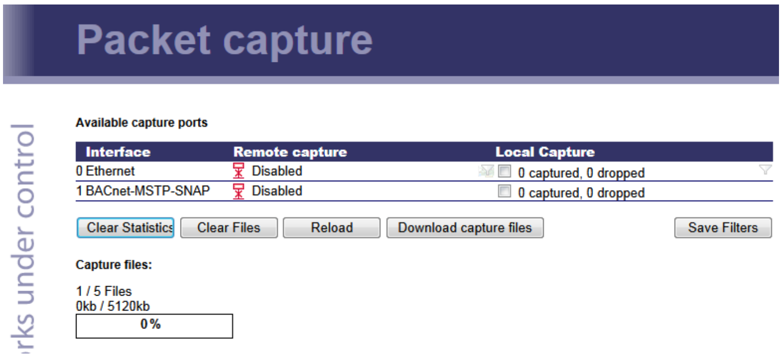


Figure 374: Packet capture statistics.

Verify that the Ethernet and optionally the MS/TP capture ports are listed in the **Available capture ports** table and that the **Remote capture** status for these ports reads **Disconnected**. If the MS/TP port is not listed on a device that has an MS/TP port, make sure that the MS/TP port is enabled in the port configuration.

To log offline without a Wireshark attached to the device, click the check box **Local Capture**. The device will then start capturing packets and stores them in a ring buffer. The log file can be downloaded by clicking on the button **Download capture files**. This stores a

ZIP archive of the packet capture to your local hard drive. Capture files can be cleared by clicking **Clear Files**. After a reboot all local capture files are lost.

For local Ethernet capture additional capture filters can be added to narrow down the amount of logged packets to those of interest. Select the line Ethernet port line and enter a basic filter expression at the bottom of the page. Then click on **Add** and add more filters. Finally click on **Save Filters** to store and activate the local capture filters. Figure 375 shows an example filter for packets with source IP address 192.168.24.100.

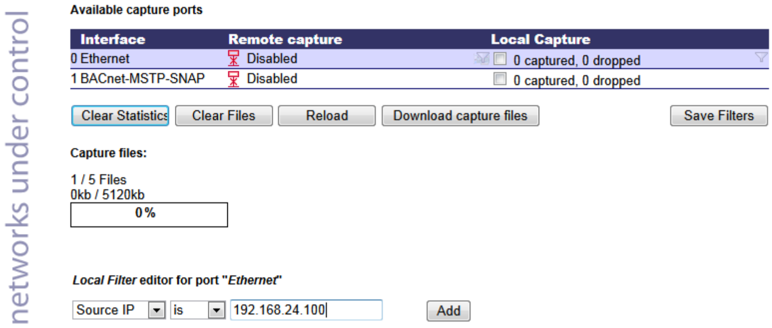


Figure 375: Adding local Ethernet capture filters.

21.3.3 Run Wireshark Remote Capture

The remote packet capture requires the use of Wireshark 1.6.11 with WinPCAP 4.1.2. Please update your Wireshark installation to this version or use a newer Wireshark version.

To add a remote capture port

1. Open Wireshark and choose the menu **Capture → Options…** . This opens the **Capture Options** dialog as shown in Figure 376.

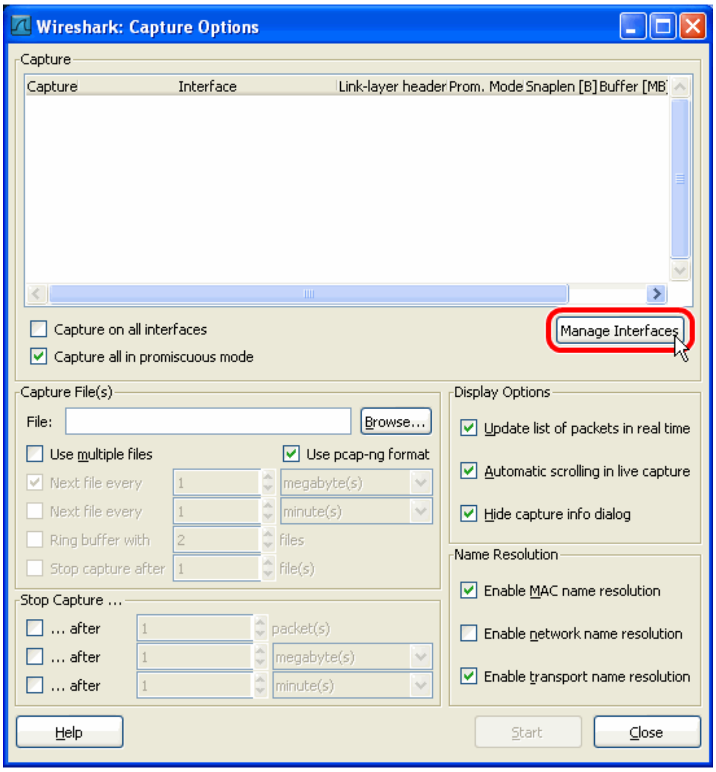


Figure 376: Wireshark Capture Options Dialog.

2. Click the **Manage Interfaces** button to open the **Add new interfaces** dialog.

3. Select the **Remote Interfaces** tab and click **Add** as shown in Figure 377.

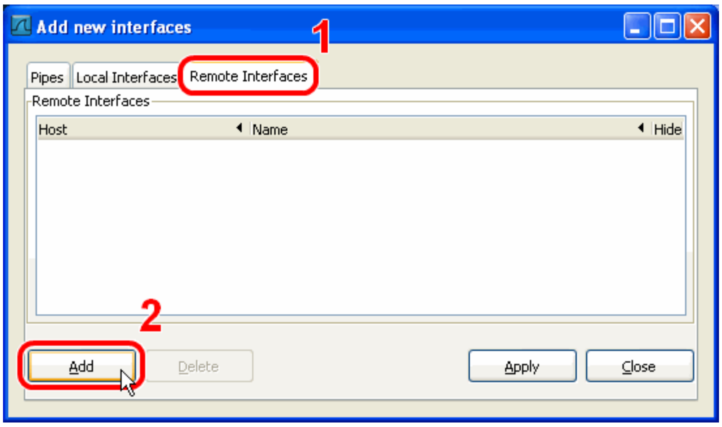


Figure 377: Wireshark Add New Interfaces Dialog.

4. Enter the correct settings for **Host** and **Port** (default 2002) and, if authentication is enabled, enter **Username** and **Password** in the corresponding fields as shown in Figure 378.

5. Note that only the users **admin** and **operator** are allowed to connect.

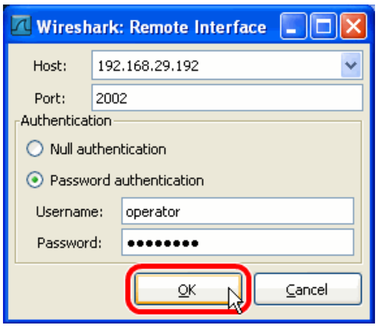


Figure 378: Wireshark Remote Interface Dialog.

6. Click **OK** to retrieve the interface list from the device.

7. If the connection to the device was established successfully, the **Remote Interfaces** list will be updated with information about all capture ports available on the device as shown in Figure 379.

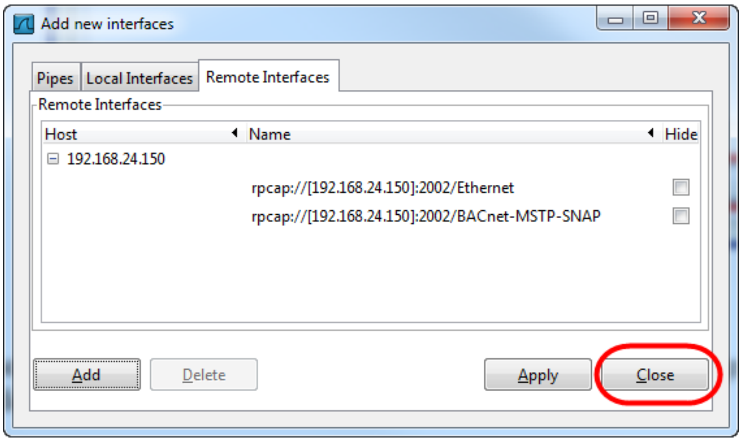


Figure 379: Added new interface to Wireshark.

8. Close the **Add new interfaces** and **Capture Options** dialogs to return to the main window.

To Start a Remote Capture

1. Select the created remote interface from the interface list in the main window. It is named 'Raw Ethernet traffic' for remote Ethernet dapture and 'SNAP encapsulated BACnet MS/TP traffic' for remote MS/TP capture.

2. Click the **Start** button as shown in Figure 380.

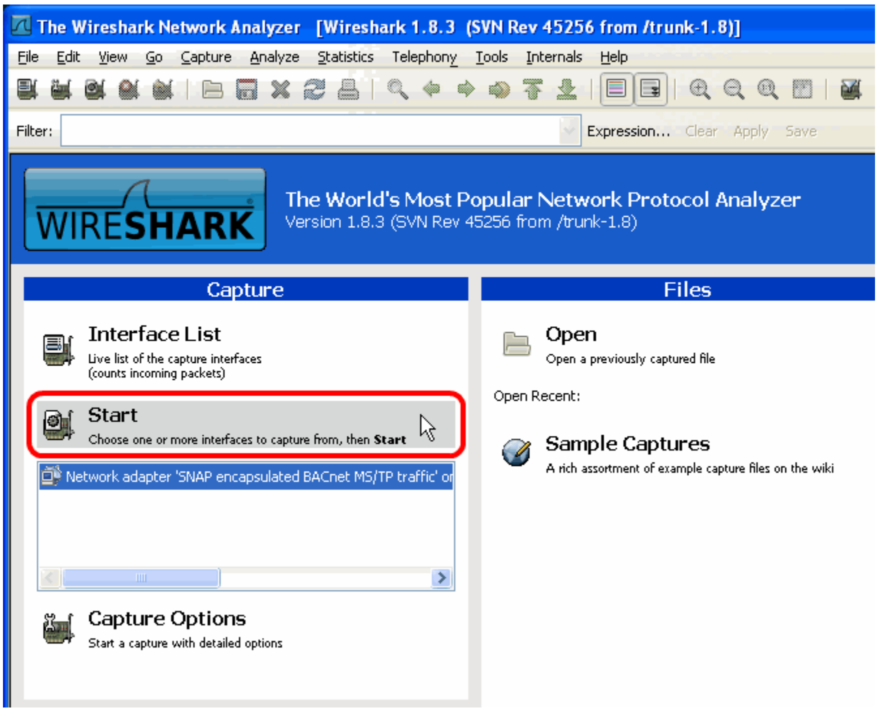


Figure 380: Start Remote Capture in Wireshark.

3. Wireshark will attempt to establish a connection to the device and, if successful, start displaying packets. An example capture is shown in Figure 381.

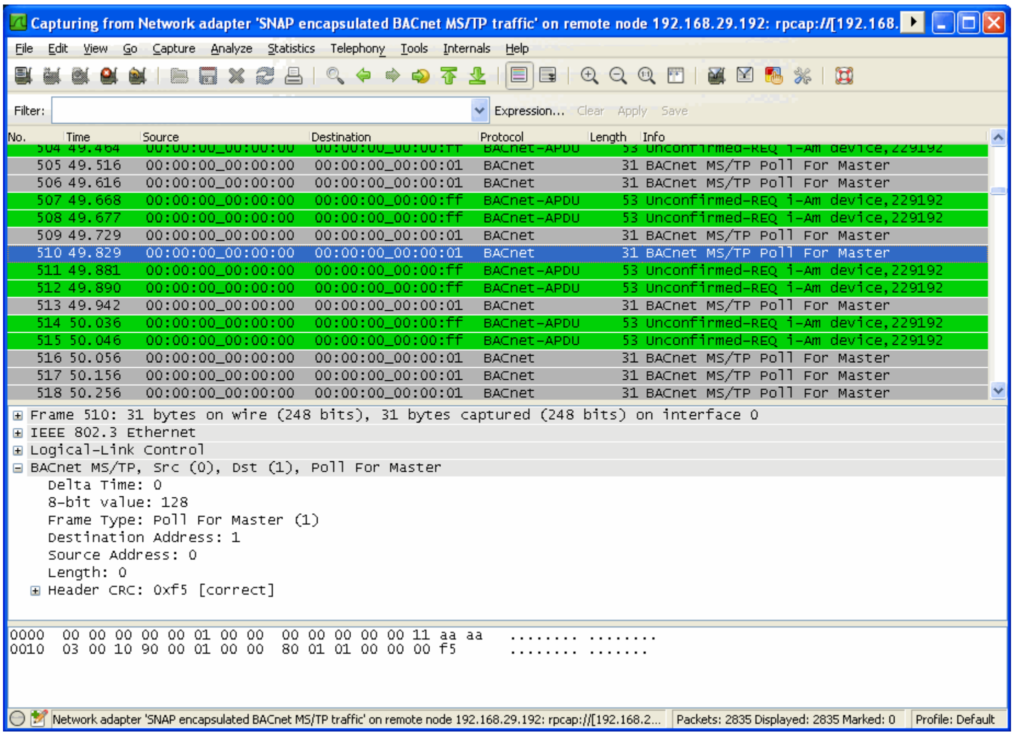


Figure 381: Example MS/TP remote capture in progress.

22 Application Notes

22.1 The LSD Tool

Please refer to application note "AN002E LSD Tool" for further information about the LOYTEC system diagnostics tool for the LINX-10X.

22.2 Use of Static, Dynamic, and External NVs on a Device

Please refer to application note "AN009E Changing Device Interface in LNS" for more information on the static NV interface, XIF files, device templates and the use of static, dynamic, and external NVs on LOYTEC gateway products.

23 Security Hardening Guide

This guide contains security-relevant information for operating the product on IT networks. The information refers to the firmware version and the instructions found in the previous chapters of this User Manual.

23.1 Installation Instructions

Install the device over the Web interface:

- Set up the basic device functions and protocol settings as described in Section 3.2.
- Disable the FTP, and Telnet servers in the IP port configuration as described in Section 5.2.4.

Connect a serial console cable:

- Connect to the console as described in Section 21.2.1.
- Go to menu [3] system configuration.
- Disable the Web server in option [9].
- Save the settings by hitting [x] for exit and save.

23.2 Firmware

The device is equipped with one piece of software. This is the firmware image and its related firmware version. The firmware is distributed as a downloadable file. The device can be upgraded by placing the firmware image onto the device using the procedure described in Chapter 20.

23.3 Ports

This Section lists all ports, which may be used by the device. The ports are default settings for their respective services. If not stated otherwise, the ports can be changed.

Required Ports:

- 80 tcp: This port is opened by the Web server and the OPC XML-DA server. It can be disabled if OPC XML-DA is not required. The port can be changed.
- 1628 udp/tcp: This is the data exchange port for CEA-852 (LON over IP). It is required for the primary function of the device to exchange control network data between

routers over the IP network. Each device needs this port open. The port can be changed.

- 1629 udp/tcp: This is the configuration server port of CEA-852. Exactly one device in the system needs this port open. Other devices register with the configuration server to form the IP-852 channel list. The port can be changed.
- 47808 udp: This is the data exchange port for BACnet/IP. It is required for the primary function of the device to exchange control network data between routers over the IP network. Each device needs this port open. The port can be changed.

Optional ports not necessary for the primary product function. They can be disabled as described in the installation instructions in Section 23.1:

- 21 tcp: This port is opened by the FTP server. The port can be changed and disabled.
- 22 tcp: This port is opened by the SSH server. The port can be changed and disabled.
- 23 tcp: This port is opened by the Telnet server. The port can be changed and disabled.
- 161 tcp: This port is opened by the SNMP server. This port is disabled by default. The port can be changed.
- 443 tcp: This port is opened by the secure Web server for HTTPS. It can be disabled.
- 5900 tcp: This port is opened by the VNC server, if it is enabled. This port is disabled by default. The port can be changed.
- 502 tcp: This port is opened, if Modbus TCP is configured in slave mode. This port is disabled by default. The port can be changed.
- 3671 udp: This port is opened by KNXnet/IP, if KNX is enabled on the Ethernet interface. This port is disabled by default. The port can be changed.
- 1630 udp/tcp: This port is used by the CEA-709 RNI and for the remote LPA. The port can be changed and disabled.
- 2048 tcp: This port is opened by the logiCAD online test. It cannot be changed. The service can be disabled but the port will remain open.
- 16028/16029 udp: These ports are opened for LIOB-IP on the device. These ports cannot be changed. They can be disabled.
- 2002 tcp: This port is opened by the Wireshark protocol analyzer front-end. This port is disabled by default. The port can be changed.
- 4840 tcp: This port is opened by the OPC UA server. This port is disabled by default. The port can be changed.

23.4 Services

Required services:

- CEA-852 (LON over IP): Primary function of the device. This service is in accordance with the standard ANSI/CEA-852-B.
- BACnet/IP: Primary function of the device. This service is in accordance with the standard ANSI/ASHRAE 135-2010.
- OPC XML-DA: This Web service provides access to data points over the OPC XML-DA standard.

Optional services not necessary for the primary product function. They can be disabled as described in the installation instructions in Section 23.1:

- HTTP: Web server. It provides a Web-based configuration UI. The Web UI can be disabled after setting up the device. The Web service is also used for the Configurator connection for configuration, firmware upgrade, and access to the log file.
- HTTPS: Secure Web server. It provides a Web-based configuration UI using HTTPS. It is also used for a secure Configurator connection.
- SSH: SSH server. It provides secure access to the device console menu over the network.
- FTP and Telnet: The FTP and Telnet server is used for connection to the device by the Configurator for configuration, firmware upgrade, and access to the log file. On devices without SSH these services must be enabled during device configuration.
- VNC: The VNC server can be used for remote access to the LCD display on devices that have it. The service is disabled by default.
- Modbus TCP: A Modbus TCP server is running when Modbus TCP is operated in slave mode. In all other cases this service is not needed.
- KNXnet/IP: A KNXnet/IP server is running if KNX is enabled on the Ethernet port. In all other cases this service is not needed.
- RNI: This service provides the remote network interface (RNI) function. It is also used by the remote LPA feature. If these features are not needed the service can be disabled.
- logiCAD online test: This service is used by the L-logiCAD programming tool for online debugging of IEC61131 programs. It is enabled by default on L-INX devices that have the IEC61131 logic kernel. The service can be disabled.
- LIOB-IP: This service is used by the L-IOB host function to operate LIOB-IP I/O modules. This service is enabled by default on all L-INX devices. The service can be disabled.
- OPC UA: This secure service provides access to data points over the OPC UA standard. The service is disabled by default.
- SNMP: SNMP server. It provides network management information on the device used by standard IT tools. The service is disabled by default.
- Wireshark front-end: The Wireshark protocol analyzer may connect to this service and retrieve online protocol analyzer logs. The service is disabled by default.

23.5 Logging and Auditing

The device contains a log file, which can be read out over FTP or the Web server. This log contains information when the device started and when crucial communication errors occur. Other information such user log-on are not logged as they are not part of the primary services of this device.

Logged events:

- Time of the last power-on reset of the L-INX/L-GATE device.
- Time and version of the last firmware upgrade.
- Time when the device configuration has been cleared or the device was reset to factory defaults.
- Commission of the CEA-709 node/router.
- Static errors in the device and data point configuration.
- System overload situations as one-time log messages since last power-on.
- Crucial communication errors as they occur.

24 Specifications

24.1 Physical Specifications

24.1.1 LINX-100//101/110/111/200/201/210/211, LGATE-900

Operating Voltage	12 – 35 VDC or 12 – 24 VAC ±10 %
Power Consumption	typ. 3 W
In rush current	up to 950 mA @ 24 VAC
Operating Temperature (ambient)	0°C to +50°C
Storage Temperature	–10°C to +60°C
Humidity (non condensing) operating	10 to 90 % RH @ 50°C
Humidity (non condensing) storage	10 to 90 % RH @ 50°C
Enclosure	Installation enclosure 107 mm wide, DIN 43 880
Environmental Protection	IP 40 (enclosure); IP 20 (screw terminals)
Installation	DIN rail mounting (EN 50 022) or wall mounting

24.1.2 LINX-102/103/112/113/202/203/212/213 and LGATE-902

Operating Voltage	12 – 35 VDC or 12 – 24 VAC ±10 %
Power Consumption	typ. 2.5 W
In rush current	up to 950 mA @ 24 VAC
Operating Temperature (ambient)	0°C to +50°C
Storage Temperature	–10°C to +60°C
Humidity (non condensing) operating	10 to 90 % RH @ 50°C
Humidity (non condensing) storage	10 to 90 % RH @ 50°C
Enclosure	Installation enclosure 107 mm wide, DIN 43 880

Environmental Protection	IP 40 (enclosure); IP 20 (screw terminals)
Installation	DIN rail mounting (EN 50 022) or wall mounting

24.1.3 LINX-12X/15X/22X, LGATE-95X

Operating Voltage	24 VDC or 24 VAC ±10 %
Power Consumption	typ. 2.5 W
In rush current	up to 950 mA @ 24 VAC
Operating Temperature (ambient)	0°C to +50°C
Storage Temperature	–10°C to +60°C
Humidity (non condensing) operating	10 to 90 % RH @ 50°C
Humidity (non condensing) storage	10 to 90 % RH @ 50°C
Enclosure	Installation enclosure 159 mm wide, DIN 43 880
Environmental Protection	IP 40 (enclosure); IP 20 (screw terminals)
Installation	DIN rail mounting (EN 50 022) or wall mounting

24.2 Resource Limits

24.2.1 L-INX Models

Table 31 and Table 32 specify the resource limits of the different L-INX models.

Model / Limits	100/101	102/103	110/111	112/113	120/121	150/151
Total number of data points	10,000	10,000	10,000	10,000	30,000	30,000
OPC Tags	2,000	2,000	500	2,000	10,000	10,000
User Registers	1,000	1,000	1,000	1,000	2,000	2,000
NVs (static, dynamic)	1,000	1,000	1,000	1,000	2,000	2,000
External NVs	1,000	1,000	1,000	1,000	2,000	2,000
Alias NVs (ECS and legacy mode)	1,000	1,000	1,000	1,000	2,000	2,000
Address table entries/legacy	512/ 15	512/ 15	1,000/ 15	1000/ 15	1,000/ 15	1,000/ 15
LONMARK Calendar objects	1 (25 calendar patterns)					
LONMARK Scheduler objects	100 (max. AST configuration size 384KB, 64 data points per scheduler)					
LONMARK Alarm Servers	1	1	1	1	1	1
BACnet objects (analog, binary, multi-state)	n/a	n/a	n/a	n/a	n/a	1,000
BACnet client mappings	n/a	n/a	n/a	n/a	n/a	5,000
BACnet scheduler objects	n/a	n/a	n/a	n/a	n/a	100
BACnet calendar objects	n/a	n/a	n/a	n/a	n/a	25
BACnet notification classes	n/a	n/a	n/a	n/a	n/a	32
BDT max recommended	n/a	n/a	n/a	n/a	n/a	100
KNX Communication Objects (per interface)	n/a	250	n/a	250	1000	1000
Trend Logs	256	256	256	256	512	512
Total trended data points	256	256	256	256	1000	1000
Total aggregated size	6MB	60MB	6MB	60MB	60MB	60MB
E-mail templates	100	100	100	100	100	100
Math objects	100	100	100	100	100	100
Alarm Logs	10	10	10	10	10	10
Modbus data points	2,000	2,000	2,000	2,000	2,000	2,000
M-Bus data points	1,000	1,000	1,000	1,000	1,000	1,000
EnOcean data points	n/a	250	n/a	250	1,000	1,000
Connections (local)	1,000	1,000	1,000	1,000	2,000	2,000
Connections (global)	250	250	250	250	250	250
L-WEB Clients (concurrent)	15	32	15	32	32	32
L-IOB Modules	8	8	8	8	24	24

Table 31: Resource limits of different L-INX models

Limits \ Model	200/201	202/203	210/211	212/213	220/221
Total number of data points	10,000	10,000	10,000	10,000	30,000
OPC Tags	2,000	2,000	500	2,000	10,000
User Registers	1,000	1,000	1,000	1,000	2,000
NVs (static, dynamic)	n/a	n/a	n/a	n/a	n/a
External NVs	n/a	n/a	n/a	n/a	n/a
Alias NVs (ECS and legacy mode)	n/a	n/a	n/a	n/a	n/a
Address table entries/legacy	n/a	n/a	n/a	n/a	n/a
LONMARK Calendar objects	n/a	n/a	n/a	n/a	n/a
LONMARK Scheduler objects	n/a	n/a	n/a	n/a	n/a
LONMARK Alarm Servers	n/a	n/a	n/a	n/a	n/a
BACnet objects (analog, binary, multi-state)	750	750	750	750	1,000
BACnet client mappings	750	750	750	750	5,000
BACnet scheduler objects	100	100	100	100	100
BACnet calendar objects	25	25	25	25	25
BACnet notification classes	32	32	32	32	32
BDT max recommended	100	100	100	100	100
KNX Communication Objects (per interface)	n/a	250	n/a	250	1000
Trend Logs	256	256	256	256	512
Total trended data points	256	256	256	256	1000
Total aggregated size	6MB	60MB	6MB	60MB	60MB
E-mail templates	100	100	100	100	100
Math objects	100	100	100	100	100
Alarm Logs	10	10	10	10	10
Modbus data points	2,000	2,000	2,000	2,000	2,000
M-Bus data points	1,000	1,000	1,000	1,000	1,000
EnOcean data points	n/a	250	n/a	250	1,000
Connections (local)	1,000	1,000	1,000	1,000	2,000
Connections (global)	250	250	250	250	250
L-WEB Clients (concurrent)	15	32	15	32	32
L-IOB Modules	8	8	8	8	24

Table 32: Resource limits of different L-INX models (cntd.)

24.2.2 L-GATE Models

Table 33 specifies the resource limits of the different L-GATE models.

Model / Limits	900	902	95X
Total number of data points	10,000	10,000	30,000
OPC Tags	500	2,000	5,000
User Registers	1,000	1,000	2,000
NVs (static, dynamic)	1,000	1,000	2,000
External NVs	1,000	1,000	2,000
Alias NVs (ECS and legacy mode)	1,000	1,000	2,000
Address table entries/legacy	512/15	512/15	1,000/15
LONMARK Calendar objects	1 (25 calendar patterns)		
LONMARK Scheduler objects	100 (max. AST configuration size 384KB, 64 data points per scheduler)		
LONMARK Alarm Servers	1	1	1
BACnet objects (analog, binary, multi-state)	750	750	1,000
BACnet client mappings	750	750	1,000
BACnet scheduler objects	100	100	100
BACnet calendar objects	25	25	25
BACnet notification classes	32	32	32
BDT max recommended	100	100	100
KNX Communication Objects (per interface)	n/a	250	1000
Trend Logs	256	256	512
Total trended data points	256	256	1000
Total aggregated size	2MB	60MB	60MB
E-mail templates	100	100	100
Math objects	100	100	100
Alarm Logs	10	10	10
Modbus data points	n/a	250	2,000
M-Bus data points	n/a	250	1,000
EnOcean data points	n/a	250	1,000
Connections (local)	1,000	1,000	2,000
Connections (global)	250	250	250
L-WEB Clients (concurrent)	15	32	32
L-IOB Modules	n/a	n/a	n/a

Table 33: Resource limits of different L-GATE models

24.3 Removable Media

24.3.1 LINX-12X/15X/22X, LGATE-95X

SD Card	microSD form factor, max 8GB, standard density or SDHC (no SDXC), optionally with or without partition table, uses first primary partition

25 References

[1] L-IP User Manual 6.0, LOYTEC electronics GmbH, Document № 88065911, September 2012.

[2] LIP-ME201 User Manual 4.9, LOYTEC electronics GmbH, Document № 88073506, September 2013.

[3] NIC User Manual 4.2, LOYTEC electronics GmbH, Document № 88067217, April 2013.

[4] LWEB-802/803 User Manual 2.0, LOYTEC electronics GmbH, Document № 88074213, June 2014.

[5] LWEB-900 User Manual 1.3, LOYTEC electronics GmbH, Document № 88081504, March 2014.

[6] L-VIS User Manual 5.0, LOYTEC electronics GmbH, Document № 88068519, June 2014.

[7] LIOB-10x/x5x User Manual 5.0, LOYTEC electronics GmbH, Document № 88078510, April 2014.

26 Revision History

Date	Version	Author	Description
2008-12-23	3.0	STS	Initial revision V3.0 for LINX-20X 3.0
2009-02-20	3.2	PP	Added Section 5.2 Connections, Chapter 10 M-Bus.
2009-05-14	3.3	PP	Updated for LINX-20X firmware release 3.3, added Chapter 11 Modbus, added Section 6.1.5. Behavior on Value Changes, added Section 6.1.6 Custom Scaling, added Section 7.7.2. Create Connections from a CSV File
2009-11-16	3.4	STS	Updated for LINX-20X firmware release 3.4. Section 6.3.3 Withdraw and embedded exceptions. Section 7.3.2 renumber, filter, include subfolders. Updated Section 7.3.3 BACnet project settings, updated Section 7.6.1 Scan for BACnet objects. Section 7.8.2 Replace data points in e-mail template. Added Section 7.9.7 Configure embedded exceptions. Section 7.13 added COV delta to math objects, replace data points. Added Section 12.1 M-Bus cabling, added Section 12.2 Modbus cabling.
2010-04-30	3.5	STS	Updated for LINX-20X firmware release 3.5. Removed Section on Console UI and added console statistics as Section 13.2. Added Section 4.2.3 port configuration Web UI. Added Section 6.2.5: Managing Multistate Maps. Section 6.3 Project settings: added device configuration tab. Section 6.11.4 Added alarm log fill level notification. Section 7.2: New L-WEB project manager. Section 9.5.1 M-Bus secondary address scanning, new activation in project settings. Section 10.3.1 Modbus speeds 1200, 2400, 4800. Section 10.4.1 Modbus slave, device config in project settings. Section 16.1.2 Resource Limits: 2000 OPC tags.
2010-10-18	4.0	STS	Merged manuals for LINX-10X, 20X, 11X, 21X. Updated for firmware version 4.0.
2011-04-22	4.1	STS	Updated for L-INX firmware 4.1. Section 5.1.8 Documented system registers. Section 5.3 Added math object function list. Section 5.4.1 Describe alarm types. Section 5.5.3 Model number range. Section 6.3 Updated project settings. Added Section 6.6.6: Backup/Restore via Configurator. Section 6.10.3 Create connections via drag-and-drop. Added section 6.15.3: RemoteTrend Logs. Added Section 10.5.6 M-Bus Device Capabilities. Added Section 11.5.3: Data point creation with online test. Added Section 11.5.7: Create Modbus slave data points. Section 12.5: Updated workflows. Added Section 14.4: Redundant Ethernet.
2011-08-26	4.2	STS	Updated for L-INX firmware 4.2. Added Sections 4.2.14, 4.2.15 for BACnet 1-hop BDT and BACnet ACL. Section 4.2.16 Update on SSL support. Added Section 4.4.2 OPC ACL. Added Section 5.1.10 Structures. Section 5.4.4

Date	Version	Author	Description
			Updated for trend backup on SD cards. Sections 5.1.2, 5.5.7, 5.6.5, 9.1.4 Added description of dynamic polling. Section 6.2 Updated for new property view, description of color coding. Section 6.13 Added alarm fields in e-mails. Added Section 11.5.8 Structured Modbus Data Points. Added Section 16.2.5 BBMD Communications Test.
2011-11-25	4.3	STS	Updated for L-INX firmware 4.3. Section 4.2.13: Description of new APDU settings. Added Section 6.9.9 Read active priority. Added Section 6.10.5 Global connections. Section 11.2: Updated on Modbus parity settings. Section 11.5.9: documented frame type damaged. Section 12.2: Describe installation of logiCAD softlock license.
2012-01-13	4.4	STS	Updated for L-INX/L-GATE firmware 4.4. Integrated L-GATE manual. Section 3.4.4: Description of green PLC LED. Added Section 4.2.6 VNC configuration. Section 4.2.8: Description for separate IP channel for global connections. Added Section 4.3.8 BACnet bindings statistics. Added Section 5.1 universal gateway. Added Section 6.8.6 Configure user-defined function blocks. Added Section 6.10.6 Automatic generation of BACnet objects. Section 14.5 Usage of biasing terminator LT-B4.
2012-04-12	4.5	STS	Update for L-INX/L-GATE firmware 4.5. Section 4.3.7: Enhanced BACnet MS/TP statistics. Added Section 5.2.11 Property Relations. Updated Section 5.5.1 Alarming. Added Section 5.10 Storage of User Documenation. Updated Section 6.2.6 for managing favorites. Added Section 6.2.7 Managing property relations. Updated Section 6.3.3 for unique NV names option. Updated Section 6.13.1 for generic alarm server. Updated Section 6.13.2 for extended alarm conditions. Updated Section 6.13.4 for linear mode of alarm logs. Added Section 6.13.5 Multi-edit of alarm conditions. Added Chapter 12 KNX. Added Section 13.6.7 Using Structured Data Point Members.
2012-07-01	4.6	STS	Update for L-INX/L-GATE firmware 4.6. Section 6.3.8 Info tab in project settings. Section 6.9.7 Import Server Objects from an EDE File. Section 6.9.11 Write with Priority. Section 10.7.10 M-Bus Device Replacement. Section 11.5.1 Updated multi-read limit. Section 12.5.2 Group import settings, naming rules in KNX Project Settings. Updated Section 13.6.3 with structured data types. Section 13.6.5 Using Retain Variables. Section 13.6.9 BACnet Server Objects in logiCAD. Chapter 19 Security Hardening Guide.
2012-10-03	4.7	STS	Section 4.2.25 SSH Server. Section 4.2.26 HTTPS Server. Section 5.2.8 Secure Mode system registers.
2012-12-14	4.8	STS	Update for L-INX/L-GATE firmware 4.8. Added LGATE-951 model. Added Section 3.4.7 on EXT LED. Added Section 5.2.12 Convertible Engineering Units. Added Section 5.4.2 Multi-Slot Connections. Added Section 5.4.3 Automatic Generation and Templates. Updated Section 5.5.3 for scheduling of events. Added Section 5.6.9 CEA-709 Data Points in Connections. Added Section 5.7.6 BACnet Data Points in Connections. Updated Section 6.2.4 Property View with convertible units. Updated Section 6.6.3 Create user registers with read or write data points. Added Section 6.10.4 Create a Multi-Slot Connection. Added Section 6.10.5 Create a Math Block Adaptor. Updated Section 6.10.8 by adding

Date	Version	Author	Description
			description of auto-generate preview dialog. Added Section 6.10.9 Create an Auto-Generate Template. Added Section 6.10.10 Managing Connection Resources. Updated Section 6.12 with new schedule configuration. Updated Section 6.14 by adding aligned interval trends. Updated Section 12.6.4 for pre-installed KNX types. Added Section 13.2.2 Hardlock License. Added Section 17.3 MS/TP Remote Packet Capture.
2013-08-08	4.9	STS	Update for L-INX/L-GATE firmware 4.9. Section 4.2.15 MS/TP Configuration: Added slave proxy enable. Added Section 4.2.16 BACnet Recipients. Added Section 4.2.17 BACnet Time Master. Added Section 4.2.20 BACnet Slave Proxy. Updated Section 4.2.24 Certificate Management. Reorganized Section 4.3 Data Management. Added Section 4.4.5 Global Connections Statistics. Added Section 5.5.6 Historic Filters. Section 5.7.3 BACnet Schedulers and Calendars: Added preset labels. Section 5.7.4 BACnet Trend Logs: Updated to revision 12. Added Section 5.10 Regular Expressions. Section 6.2.5 Managing Multistate Maps updated. Section 6.2.9 BACnet Properties: Updated new properties. Section 6.3.2 Data Point Naming Rules: Added LNS sub-systems. Updated Section 6.6.3 Create User Registers as value data points. Updated Section 6.9.4 Create a Client Mapping. Updated Section 6.10.9 Create an Auto-Generate Template with variable placeholders. Added Section 6.17 Historic Filters configuration. Added Section 9.4.7 OPC Bridge Limits. Section 11.4.5 Modbus Properties updated. Section 13.2.1 Added text on license upgrade. Added Section 13.4.6 I/O Driver Settings. Updated Section 14.2.1 NV Import File documentation. Updated Section 17.3 Remote Packet Capture with Ethernet capture. Added Section 20.3 Removable media support information.
2014-03-31	5.0	STS	Update for L-INX/L-GATE firmware 5.0. New chapter 2 What's new in L-INX. Added Section 4.7.7 Modbus LED. Section 5.2.24 updated for installing signed cert and CA cert. Section 5.2.25 Firmware Update over Web UI. Section 5.3.1 Data point bread-crumb navigation and live update. Section 5.3.2 Trend log overview page. Section 6.2.2 Receive timeout definition, background polling. Section 6.2.11 Added scheduler property relations. Section 6.4.3 Placeholders in auto-generate templates. Section 6.5.2 Updated parameter merge dialog. Section 6.5.3 Added property relations to scheduler. Section 6.5.5 Placeholders in e-mail templates. Section 6.5.6 Historic filter difference mode. Section 7.3.2 Data point naming rules updated. Section 7.3.5 BACnet keep OWS settings added. Section 7.6.1 Connection dialog updated. Section 7.9.11 Extended to priority read data points. Section 7.10.1 Added bi-directional connection. Section 7.12 updated for generic schedulers. Section 7.16.1 Add trigger calculation in math object. Added Section 14.4.7 PLC conflicts. Updated references of Chapter 22.
2015-01-22	5.1	STS	Update for L-INX/L-GATE firmware 5.1. Section 2.1 New in L-INX/L-GATE 5.1.0 added. Section 1.7 EnOcean added. Section 5.1 Device Information and Account Management updated. Section 5.2.4 IP Configuration updated. Section 5.2.5 Using Multiple IP Ports added. Section 5.2.6 IP Host Configuration updated. Section 5.2.7 WLAN Configuration added. Section 5.2.20 BACnet Restart Notifications added.

Date	Version	Author	Description
			Section 5.3.2 Trend chart view added. Section 5.4 Commission Web UI for BACnet added. Section 6.2.10 Structures for favorites documented. Section 6.5.3 Schedule default "silent" documented. Section 7.3.1 General settings updated. Section 7.6.1 Device connection dialog updated. Section 7.8.3 Using Feedback Data Points updated for integrated value feedback. Section 7.9.12 Duplicate BACnet Devices with Data Points added. Section 7.10.10 Create a Complex Auto-Generate Template added. Section 7.11 E-mail Templates format string documented. Section 11.4.3 Commission Web UI for M-Bus added. Section 11.4.4 M-Bus Statistics updated. Section 12.2.2 Dynamic Polling specified for Modbus master. Section 12.3.3 Commission Web UI for Modbus added. Section 12.3.4 Modbus statistics updated. Section 12.5.1 Modbus Single Read/Single Write documented. Chapter 14 EnOcean added. Section 16.3.11 Added new columns to client mapping CSV. Section 16.4 SNMP Interface added. Section 17.3 Modbus RS-485 unit loads documented. Section 17.6 WLAN added. Removed LOPC-BR800 documentation.
2015-09-04	5.3.0	STS	Update for L-INX/L-GATE firmware 5.3. Section 1.9, 4.1, 4.2, 4.9 Added new models. Section 5.3.1 Data Points - edit structured values. Section 5.6 Documentation. Section 5.7.2 LWEB-802 Config. Section 6.2.13 Explain unit system definitions. Section 6.4.3 Automatic Generation and Templates - %{folder_descr} added. Section 6.4.5 Forward Delay. Section 6.7.1 Specify network unit behavior of BACnet objects. Section 7.2.4 Added SI and U.S. unit properties. Section 7.2.10 BACnet Properties - Added Notify Type, Notification Class, Event Enable, specified how to make server object name writeable. Section 7.3.1 Added unit system settings. Section 7.6.7 Create Projects for SI and U.S. Units. Section 7.16.1 Create a Math Object - multi-select and create. Section 7.18 Automated Data Point Creation. Section 12.5.3 Modbus data Point test using online test added. Section 13.4.3 KNX Protocol Analyzer. Added Chapter 15 OPC client. Added Chapter 16 ekey. Section 17.6.10 Custom Protocol Support. Section 18.1.4 Data Point Template CSV File. Chapter 24: Added limits for new models.