
LIOB-x8x

L-IOB™ I/O Controller

User Manual

LOYTEC electronics GmbH



Contact

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

Version 4.9

Document № 88080305

LOYTEC MAKES AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR IN ANY COMMUNICATION WITH YOU, AND LOYTEC SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THIS PRODUCT IS NOT DESIGNED OR INTENDED FOR USE IN EQUIPMENT INTENDED FOR SURGICAL IMPLANT INTO THE BODY OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, FOR USE IN FLIGHT CONTROL OR ENGINE CONTROL EQUIPMENT WITHIN AN AIRCRAFT, OR FOR ANY OTHER APPLICATION IN WHICH IN THE FAILURE OF SUCH PRODUCT COULD CREATE A SITUATION IN WHICH PERSONAL INJURY OR DEATH MAY OCCUR.

LOYTEC MAKES NO REPRESENTATION AND OFFERS NO WARRANTY OF ANY KIND REGARDING OF ANY THIRDPARTY COMPONENTS MENTIONED IN THIS MANUAL.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of LOYTEC.

LC3020™, L-Chip™, L-Core™, L-DALI™, L-GATE™, L-INX™, L-IOB™, LIOB-Connect™, LIOB-FT™, L-IP™, LPA™, L-Proxy™, L-Switch™, L-Term™, L-VIS™, L-WEB™, L-ZIBI™ and ORION™ stack are trademarks of LOYTEC electronics GmbH.

LonTalk®, LONWORKS®, Neuron®, LONMARK®, LonMaker®, i.LON®, and LNS® are trademarks of Echelon Corporation registered in the United States and other countries.

Contents

1	Introduction	14
1.1	Overview	14
1.2	Scope.....	14
2	Quick-Start Guide	15
2.1	Hardware Installation.....	15
2.2	Commissioning or BACnet Setup	15
2.3	Getting Started with the L-INX Configurator.....	15
2.4	Configuration of the I/Os.....	16
2.5	Getting started with logiCAD.....	17
3	Hardware Installation	22
3.1	Enclosure.....	22
3.2	Product Label	22
3.3	Mounting.....	23
3.4	Terminal Layout.....	23
3.4.1	LIOB-180.....	24
3.4.2	LIOB-181.....	25
3.4.3	LIOB-182.....	26
3.4.4	LIOB-183.....	27
3.4.5	LIOB-184.....	28
3.4.6	LIOB-480/580.....	29
3.4.7	LIOB-481/581.....	30
3.4.8	LIOB-482/582.....	31
3.4.9	LIOB-483/583.....	32
3.5	Power Supply and Wiring	33
3.5.1	CEA-709 Network Connection in Free Topology.....	33
3.5.2	CEA-709 Network Connection in Bus Topology.....	33
3.5.3	CEA-852 Network Connection	34
3.5.4	BACnet/IP Network Connection.....	35
3.5.5	Extension of LIOB-48x/58x controllers with LIOB-45x/55x device.....	35
3.6	LED signals	35
3.6.1	Status LED of LIOB-18x	35
3.6.2	Status LED of LIOB-48x	35
3.6.3	Status LED of LIOB-58x	36
3.7	Status Button	36
4	LCD Display	37

4.1	Main Page.....	37
4.2	I/O Page.....	38
4.3	Manual / Quick Edit Mode	39
4.4	I/O Configuration	39
4.5	LIOB-IP Page (LIOB-48x/58x).....	40
5	Web Interface (LIOB-48x/58x)	41
5.1	Device Information and Account Management	41
5.2	Device Configuration.....	43
5.2.1	System Configuration	43
5.2.2	Backup and Restore.....	44
5.2.3	Port Configuration	45
5.2.4	IP Configuration	46
5.2.5	VNC Configuration	46
5.2.6	CEA-852 Device Configuration (LIOB-48x)	47
5.2.7	Global Connections Configuration	48
5.2.8	BACnet Configuration (LIOB-58x).....	49
5.2.9	BACnet/IP Configuration (LIOB-58x).....	50
5.2.10	BACnet Recipients	50
5.2.11	BACnet Time Master	51
5.2.12	E-mail Configuration	51
5.2.13	IEC61131 Configuration	52
5.3	Data Management.....	53
5.3.1	Data Points	53
5.3.2	Trend	55
5.3.3	Scheduler.....	56
5.3.4	Calendar	57
5.3.5	Alarm.....	58
5.4	Device Statistics.....	59
5.4.1	System Log.....	59
5.4.2	IP Statistics	59
5.4.3	CEA-852 Statistics	60
5.4.4	Enhanced Communications Test	61
5.4.5	Global Connections Statistics	62
5.4.6	CEA-709 Statistics (LIOB-48x)	63
5.4.7	OPC Server Statistics Page.....	63
5.4.8	BACnet Bindings Statistics (BACnet Mode)	64
5.4.9	Scheduler Statistics Page	65
5.4.10	Alarm Log Page.....	65
5.4.11	Packet Capture.....	66

5.5	L-WEB	66
5.5.1	Installation	66
5.5.2	LWEB-802 Configuration.....	67
5.5.3	Access Control List.....	67
5.6	L-IOB Host and Local I/Os	68
5.6.1	LIOB-IP Bus.....	68
5.6.2	LIOB-IP Installation Page.....	68
5.6.3	L-IOB Overview Page.....	69
5.6.4	L-IOB I/O Test Page.....	70
5.7	Reset, Contact, Logout	71
6	Concepts	72
6.1	Device Installation	72
6.2	LONMARK® Device Mode (LIOB-18x/48x)	72
6.3	Data Points	73
6.3.1	Overview.....	73
6.3.2	Timing Parameters	73
6.3.3	Default Values	74
6.3.4	Persistency	74
6.3.5	Parameters.....	75
6.3.6	Behavior on Value Changes.....	75
6.3.7	Custom Scaling	76
6.3.8	System Registers	76
6.3.9	User Registers	77
6.3.10	Structures	78
6.3.11	Property Relations.....	78
6.3.12	Convertible Engineering Units.....	79
6.4	Math Objects (LIOB-48x/58x)	80
6.4.1	General Properties.....	80
6.4.2	Usage Hints.....	80
6.4.3	Function List.....	81
6.5	Connections	82
6.5.1	Local Connections.....	82
6.5.2	Multi-Slot Connections and Automatic Generation	83
6.5.3	Global Connections (LIOB-48x/58x).....	83
6.6	AST Features	84
6.6.1	Alarming	84
6.6.2	Historical Alarm Log (LIOB-48x/58x).....	85
6.6.3	Scheduling.....	86
6.6.4	Trending (LIOB-48x/58x).....	88

6.6.5	E-mail (LIOB-48x/58x).....	88
6.6.6	Historic Filters.....	88
6.7	I/O Technology.....	90
6.7.1	I/O Configuration.....	90
6.7.2	STId Card Reader Mode.....	98
6.7.3	I/O Data Points.....	99
6.7.4	Default I/O Configuration.....	101
6.8	CEA-709 Technology (LIOB-18x/48x).....	102
6.8.1	CEA-709 Data Points.....	102
6.8.2	Static Interface Changes.....	102
6.8.3	Limitations for Local CEA-709 Schedulers.....	103
6.8.4	Limitations for CEA-709 Alarm Servers.....	103
6.8.5	Limitations for Local CEA-709 Trends (LIOB-48x).....	103
6.8.6	Dynamic Polling in CEA-709.....	103
6.8.7	CEA-709 Data Points in Connections.....	103
6.9	BACnet Technology (LIOB-58x).....	104
6.9.1	BACnet Data Points.....	104
6.9.2	BACnet Alarming.....	104
6.9.3	BACnet Schedulers and Calendars.....	105
6.9.4	BACnet Trend Logs.....	105
6.9.5	Dynamic Polling in BACnet.....	106
6.9.6	BACnet Data Points in Connections.....	106
6.9.7	Native BACnet Objects for I/Os.....	106
6.10	IEC61131 Variables.....	109
7	The Configurator.....	110
7.1	Installation.....	110
7.1.1	Software Installation.....	110
7.1.2	Registration as an LNS Plug-In.....	110
7.1.3	CEA-709 Operating Modes.....	111
7.2	Data Point Manager.....	112
7.2.1	Folder List.....	112
7.2.2	Network Port Folders.....	114
7.2.3	Data Point List.....	114
7.2.4	Property View.....	115
7.2.5	Managing Multistate Maps.....	117
7.2.6	Organizing Favorites.....	118
7.2.7	Managing Property Relations.....	119
7.2.8	CEA-709 Properties (LIOB-18x/48x).....	120
7.2.9	BACnet Properties (LIOB-58x).....	120

7.3	Project Settings.....	121
7.3.1	General.....	122
7.3.2	CEA-709 Settings (LIOB-18x/48x)	123
7.3.3	AST Settings	123
7.3.4	BACnet Settings (LIOB-58x).....	125
7.3.5	System Settings	126
7.3.6	Project Information	127
7.4	Workflows in CEA-709 (LIOB-18x/48x).....	128
7.4.1	Involved Configuration Files	128
7.4.2	Configure with LNS	128
7.4.3	Configure without LNS	129
7.4.4	Device Replacement	129
7.4.5	Adding the Device to LNS	130
7.4.6	Replace a device in LNS	132
7.5	Workflows in BACnet (LIOB-58x).....	134
7.5.1	Involved Configuration Files	134
7.5.2	Engineer Online	134
7.5.3	Engineer Offline.....	135
7.6	Using the L-INX Configurator	136
7.6.1	Starting Stand-Alone.....	136
7.6.2	Uploading the Configuration.....	138
7.6.3	Create User Registers.....	139
7.6.4	Configuration Download.....	140
7.6.5	Upload the System Log	141
7.6.6	Backup and Restore	142
7.7	CEA-709 Configuration (LIOB-18x/48x).....	143
7.7.1	Starting as an LNS Plug-In.....	143
7.7.2	Create Static NVs.....	143
7.7.3	Configuration Download over LNS	144
7.7.4	Enable Legacy NM Mode	146
7.7.5	Build XIF for Port Interface.....	146
7.8	Advanced CEA-709 Configuration (LIOB-18x/48x).....	146
7.8.1	Install Unconfigured Devices (LIOB-18x).....	146
7.8.2	Using Feedback Data Points	148
7.8.3	Working with UNVTs.....	148
7.8.4	Configure User-Defined Function Blocks.....	149
7.9	BACnet Configuration (LIOB-58x).....	149
7.9.1	Scan for BACnet Objects.....	149
7.9.2	Import from EDE File	150

7.9.3	Use Imported BACnet Objects	151
7.9.4	Create a Client Mapping	151
7.9.5	Create Server Object	152
7.9.6	Export Server Objects to an EDE File	153
7.9.7	Import Server Objects from an EDE File	154
7.9.8	Map other Properties than Present_Value	154
7.9.9	Enable International Character Support.....	155
7.9.10	Read the Active Priority	155
7.9.11	Write with Priority.....	156
7.10	Connections	156
7.10.1	Create a New Connection	156
7.10.2	Create Connections from a CSV File	158
7.10.3	Modify Connections	158
7.10.4	Connection Overview	160
7.10.5	Create a Global Connection (LIOB-48x/58x)	160
7.11	E-mail Templates (LIOB-48x/58x)	161
7.11.1	Create an E-mail Template	161
7.11.2	Trigger E-mails.....	162
7.11.3	Attachments.....	163
7.11.4	Limit E-mail Send Rate	164
7.12	Local Schedule and Calendar	164
7.12.1	Create a local Calendar.....	164
7.12.2	Create Calendar Pattern.....	165
7.12.3	Create a Local Scheduler.....	165
7.12.4	Configure Scheduled Data Points	166
7.12.5	Configure Scheduled Events.....	167
7.12.6	Configure Exception Days.....	169
7.12.7	Configure Control Data Points	170
7.12.8	Using the SNVT_tod_event.....	171
7.12.9	Using the Local Scheduler.....	171
7.13	Local Alarming	171
7.13.1	Create an Alarm Server	171
7.13.2	Create an Alarm Condition.....	172
7.13.3	Deliver Alarms via E-mail (LIOB-48x/58x).....	174
7.13.4	Create an Alarm Log (LIOB-48x/58x)	175
7.13.5	Multi-Edit Alarm Conditions.....	176
7.14	Local Trending (LIOB-48x/58x).....	177
7.14.1	Create a Local Trend	177
7.14.2	Configure Trended Data Points	178

7.14.3	Trend Triggers	179
7.14.4	Download Trend Data in CSV Format.....	180
7.14.5	Deliver Trend Data via E-mail.....	181
7.14.6	Technology Trends (LIOB-58x)	181
7.15	Math Objects (LIOB-48x/58x)	182
7.15.1	Create a Math Object	182
7.15.2	Editing a Math Object.....	183
7.16	Historic Filters.....	184
7.16.1	Create Historic Filters	184
7.16.2	Managing Historic Filter Resources.....	185
7.17	I/Os	185
7.17.1	Configure I/Os	185
7.17.2	Manage I/O Configurations.....	186
7.17.3	Using I/O Data Points	187
7.17.4	Printing Labels	188
8	OPC Server (LIOB-48x/58x)	189
8.1	XML-DA OPC Server.....	189
8.1.1	Access Methods	189
8.1.2	Data Points.....	190
8.1.3	AST Objects.....	193
8.1.4	OPC Groups.....	196
8.2	Using L-WEB.....	196
8.2.1	Create a new L-WEB Project.....	196
8.2.2	Start a Graphical L-WEB Design.....	197
8.2.3	Organize L-WEB Projects	198
8.3	Using the OPC Bridge.....	199
8.3.1	Software Installation	199
8.3.2	Configure the Bridge Locally.....	199
8.3.3	Export OPC Servers for another PC.....	200
8.3.4	Import OPC Servers Using the Configurator	200
8.3.5	Manually Configure Servers	201
8.3.6	Bridge Timing Parameters	202
8.3.7	Bridge Limits	202
8.4	Using Custom Web Pages	203
9	IEC 61131	204
9.1	Overview	204
9.2	Installing logiCAD.....	204
9.2.1	Softlock License.....	205
9.2.2	Hardlock License	206

9.3	IEC61131 Project Files	207
9.4	Working with logiCAD	208
9.4.1	Managing Variables.....	210
9.4.2	Build and Download the IEC61131 Program.....	211
9.4.3	Usage of NVs, Technology Converters (LIOB-18x/48x).....	212
9.4.4	IEC61131 Program Cycle Time.....	213
9.4.5	CPU Overload.....	214
9.4.6	I/O Driver Settings.....	215
9.5	Workflows	215
9.5.1	Starting with Data Points.....	215
9.5.2	Starting with logiCAD.....	217
9.5.3	Pre-compiled IEC61131 Program.....	223
9.6	Additional Features	224
9.6.1	Force Update Functionality.....	224
9.6.2	Using UNVT variables (LIOB-18x/48x).....	224
9.6.3	Create Your Own Data Type.....	225
9.6.4	Using Persistent Data Points and Markers.....	225
9.6.5	System Registers, System Time.....	226
9.6.6	Code Protection.....	226
9.6.7	Using Structured Data Point Members.....	226
9.6.8	BACnet Server Objects (LIOB-58x).....	227
10	Operating Interfaces	229
10.1	Common Interface (LIOB-48x/58x)	229
10.1.1	Schedule and Calendar XML Files.....	229
10.1.2	Trend Log CSV File.....	229
10.1.3	Alarm Log CSV File.....	230
10.2	CEA-709 Interface (LIOB-18x/48x)	231
10.2.1	NV Import File.....	231
10.2.2	Node Object.....	232
10.2.3	Real-Time Keeper Object.....	233
10.2.4	Calendar Object.....	233
10.2.5	Scheduler Object.....	233
10.2.6	PLC Objects.....	233
10.3	BACnet Interface (LIOB-58x)	233
10.3.1	Device Object.....	233
10.3.2	Device Name and ID.....	234
10.3.3	Device Information.....	235
10.3.4	Object Database.....	235
10.3.5	Protocol Parameters.....	236

10.3.6	Diagnostics.....	236
10.3.7	Date & Time	237
10.3.8	Time Master	237
10.3.9	Backup & Restore	238
10.3.10	Client Mapping CSV File	238
10.3.11	EDE Export of BACnet Objects	239
11	Firmware Update.....	240
12	Troubleshooting.....	242
12.1	Technical Support	242
12.2	Remote Packet Capture (LIOB-48x/58x)	242
12.2.1	Configure Remote Packet Capture	242
12.2.2	Run Wireshark Remote Capture	243
13	Application Notes.....	246
13.1	External Power Supply (not using LPOW-2415A).....	246
13.2	Physical Connection of Inputs.....	246
13.2.1	Connection of Switches.....	246
13.2.2	Connection of S0 Pulse Devices (Meters).....	247
13.2.3	Connection of Voltage Sources to Universal Inputs	248
13.2.4	Connection of Voltage Sources to Digital Inputs	248
13.2.5	Connection of 4-20mA Transmitters to Universal Inputs	249
13.2.6	Connection of Resistive Sensors	249
13.2.7	Connection of STId Card Readers	250
13.3	Physical Connection of Outputs.....	250
13.3.1	6A Relays with one External Fuse	250
13.3.2	6A Relays on LIOB-xx2 using Separate Fuses	251
13.3.3	16A and 6A Relays on LIOB-xx3.....	251
13.3.4	External Relays and Inductive Loads.....	251
13.3.5	1A Triacs	252
13.3.6	Analog Outputs	252
14	Security Hardening Guide.....	254
14.1	Installation Instructions.....	254
14.2	Firmware.....	254
14.3	Ports	254
14.4	Services.....	255
14.5	Logging and Auditing	255
15	Specifications	257
15.1	I/O Specification	257
15.1.1	UI - Universal Input	257
15.1.2	DI - Digital Input, Counter Input (S0-Pulse).....	257

15.1.3 AO - Analog Output	258
15.1.4 DO - Digital Output.....	258
15.1.5 PRESS - Pressure Sensor.....	258
15.2 Internal Translation Tables	259
15.3 Specification for LIOB-18x Models.....	260
15.4 Specification for LIOB-48x Models.....	261
15.5 Specification for LIOB-58x Models.....	262
15.6 Resource Limits.....	263
16 References	264
17 Revision History	265

Abbreviations

AST	Alarming, Scheduling, Trending
BACnet	Building Automation and Control Network
CEA-709	Protocol standard for LONWORKS networks
CEA-852	Protocol standard for CEA-709 over IP networks
COV	Change-Of-Value
CP.....	Configuration Property
CS.....	Configuration Server that manages CEA-852 IP devices
DHCP.....	Dynamic Host Configuration Protocol, RFC 2131, RFC 2132
I/O	Input/Output
LIOB	LOYTEC I/O bus
MAC	Media Access Control
NAT	Network Address Translation, see Internet RFC 1631
NTC	Negative Temperature Coefficient (Temperature Sensor)
NV.....	Network Variable
OPC.....	Open Process Control
PTC	Positive Temperature Coefficient (Temperature Sensor)
SCPT.....	Standard Configuration Property Type
SNVT	Standard Network Variable Type
UCPT	User-Defined Configuration Property Type
UI.....	User Interface

1 Introduction

1.1 Overview

The L-IOB I/O controllers LIOB-18x, LIOB-48x, and LIOB-58x combine physical inputs and outputs (I/Os) with programmable logic (IEC61131) and provide application specific data points through a CEA-709 or BACnet connection. The data points are also subject to AST (alarming, scheduling, and trending) functions of the L-IOB I/O controller. The I/O controllers come in different models with different I/O configurations, including analog outputs, digital outputs, digital inputs, and universal inputs, which can be freely configured.

Initial configuration of the I/O controllers is accomplished through the L-INX Configurator software. The logic application is developed using the provided IEC-61131 compliant design tool. Parameterization of I/Os, self-test, manual modes, override values, etc. can also be performed on the L-IOB LCD display.

1.2 Scope

This document covers LIOB-18x, LIOB-48x, and LIOB-58x controllers with firmware version 4.9 (or higher) and the L-INX Configurator version 4.9 (or higher).

2 Quick-Start Guide

This chapter shows step-by-step instructions on how to configure the L-IOB I/O controller for a simple application.

2.1 Hardware Installation

A LIOB-18x I/O controller is connected to a CEA-709 network using the FT port of the L-IOB device. A LIOB-48x I/O controller is connected to a CEA-852 network using the Ethernet/IP port of the L-IOB device. A LIOB-58x I/O controller is connected to a BACnet network using the Ethernet/IP port of the L-IOB device. In all cases, the device must be powered e.g. using an LPOW-2415A power supply.

2.2 Commissioning or BACnet Setup

For LIOB-18x/48x models, use the appropriate L-IOB template of your LNS™ based network management tool (e.g. NL-220™ or LonMaker™) to create and commission a L-IOB device in the database. Once the device is created, the 'Configure' function of the network management tool can be used to invoke the Configurator Software.

For LIOB-58x models, the initial IP and BACnet setup must be done in the LCD UI (see Section 4.1) or Web UI (see Section 5.2).

2.3 Getting Started with the L-INX Configurator

Before setting up a working IEC61131 program, the data points of the L-IOB device need to be set up. These can be data points of I/Os, network variables, registers, etc. 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 1.
2. When the device is online, connect to the device by clicking on the LNS or device connect speed button as indicated by the red rectangle in Figure 1.

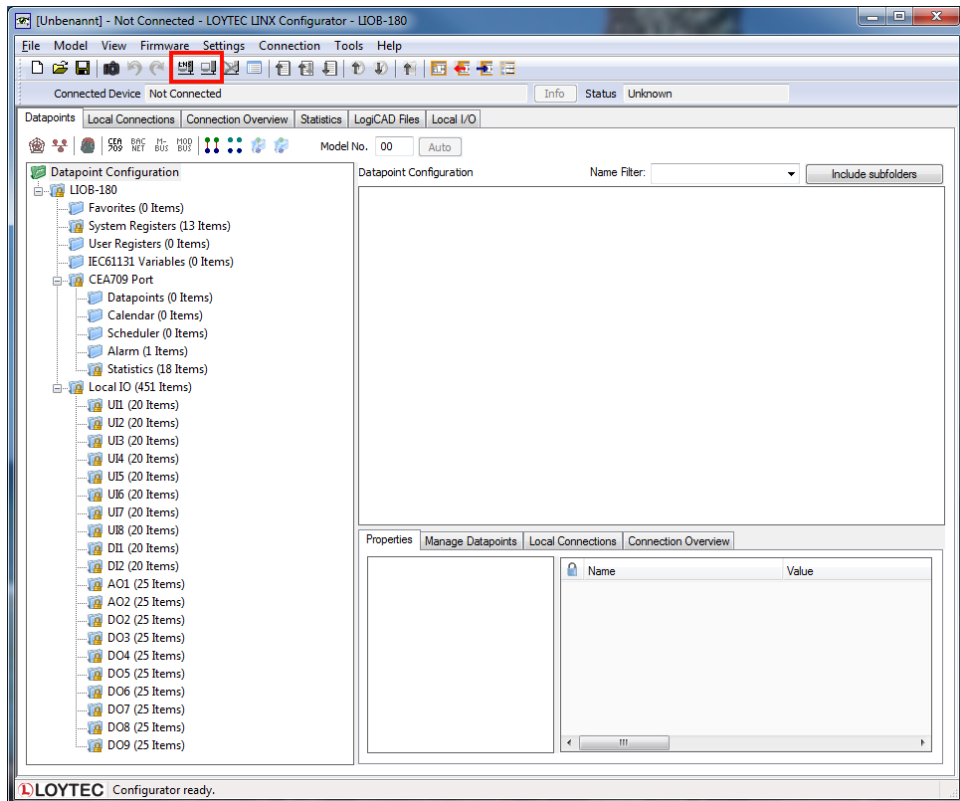


Figure 1: L-INX Configurator main screen

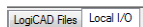
- For detailed information on how to create data points etc. please refer to Chapter 7.

2.4 Configuration of the I/Os

The L-INX Configurator uses a separate tab to configure the I/Os. The I/O configuration can be done off-line and is shown in the following steps.

To Configure I/Os

- For LIOB-18x models, select the **Local I/O** tab. For LIOB-48x/58x models, select the **L-IOB** tab and then **LIOB-LOCAL**.



- The I/Os available on that L-IOB device are displayed in the **Inputs / Outputs** table.

Inputs / Outputs

Nr.	TerminalNr	Terminal	Name	Hardware type
1	1	UI1	UI1	IN Analog/Digital
2	2	GND12	GND UI1-UI2	IN Analog/Digital
3	3	UI2	UI2	IN Analog/Digital
4	4	UI3	UI3	IN Analog/Digital

- To adapt the I/O name, double-click the name in the **Name** column and edit it, e.g., 'RoomTemp'.

Nr.	TerminalNr	Terminal	Name	Hardware type
1	1	UI1	RoomTemp	IN Analog/Digital

- Select (or multi-select) an I/O in the **Inputs/Outputs** list and observe the **Object parameters** list below. These parameters can be used to configure the I/O.

Object parameters

Nr.	DP	Create	OPC	PLC In	PLC Out	Parameter name	Parameter value	Unit	Range	Description
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Name	RoomTemp			Terminal name
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	HardwareType	IN Analog/Digital			Terminal type
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SignalType	Voltage 0-10V			Type of the input/output signal

- On the **Datapoints** tab, the data points for the I/Os have been created. These data points can be used, e.g., in the logiCAD IEC61131 program. For inputs the data point L1_x_UIy_**Input** will be used to read an input value and for outputs the data point L1_x_DOy_**Output** will be used to set an output value.

2.5 Getting started with logiCAD

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

- L-logiCAD setup package. This package installs the logiCAD software, which is needed to design PLC programs for the device.
- L-INX Configurator. This software is required to configure the device to provide the necessary data points to the PLC and integrate the device into the network.
- 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 Sections 7.1 and 9.2.

To Start a logiCAD Project

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



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

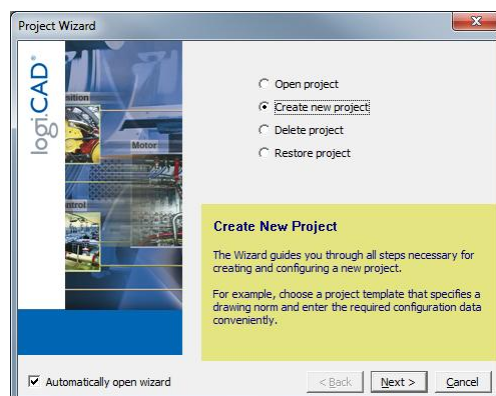


Figure 2: logiCAD project wizard

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

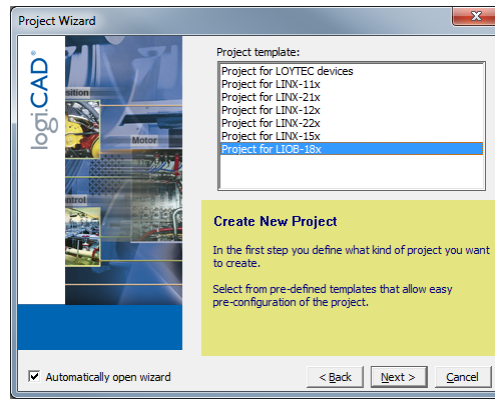


Figure 3: Available project templates

4. Select the project template for the L-IOB device (e.g. LIOB-18x).

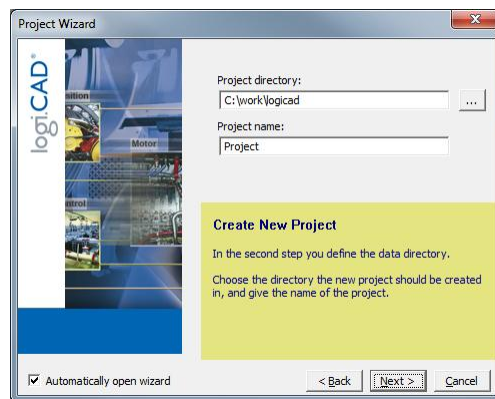


Figure 4: Project name and path

5. Specify the name of the project and the path where to store the project files, see Figure 4.

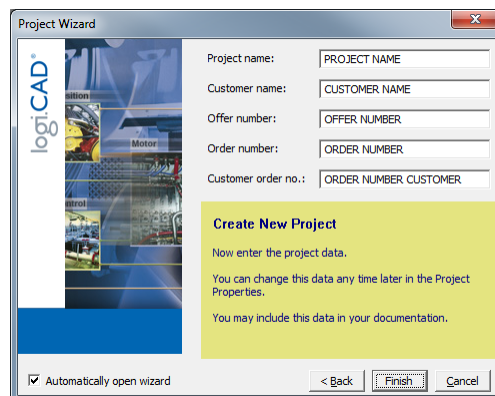


Figure 5: Additional information

6. After specifying additional information the new project is created by pressing the **Finish** button.
7. As shown in Figure 6 below, expand the tree element **Functionplans** and double click **Plan_1** in order to start editing the plan.

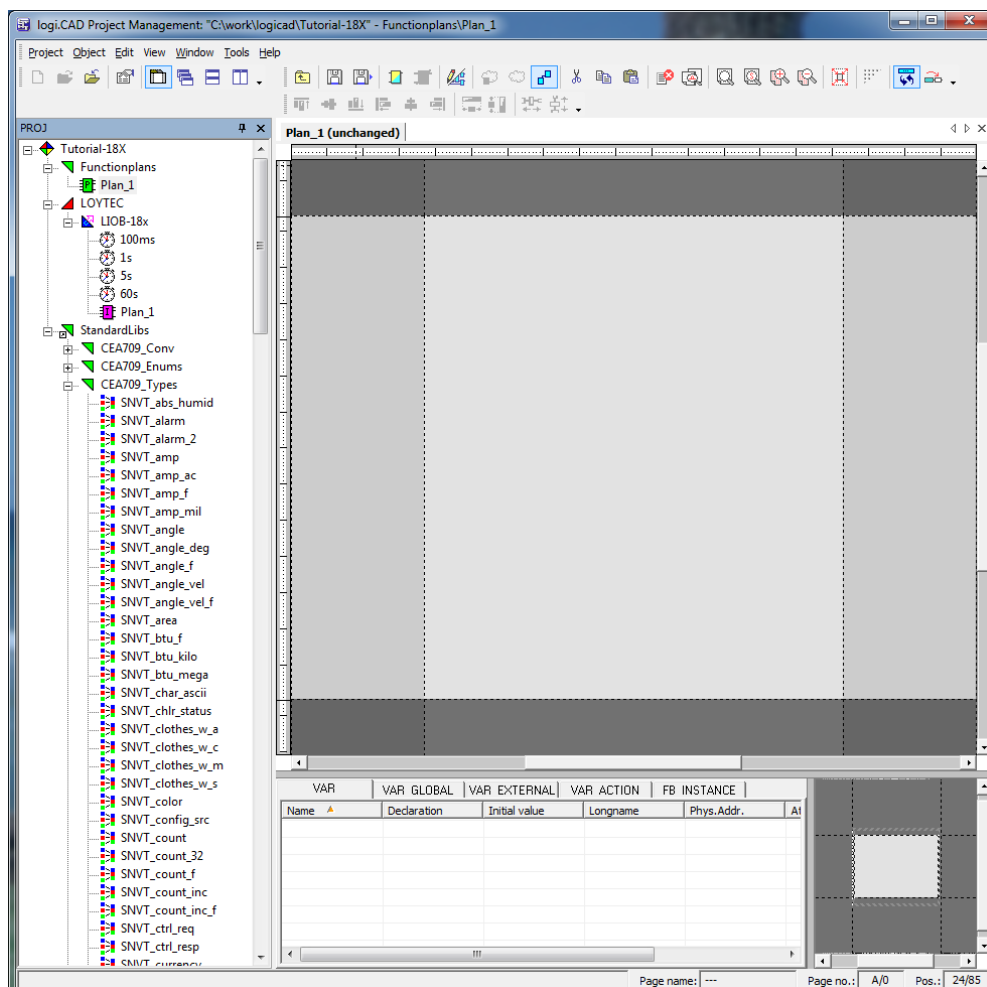


Figure 6: Edit Plan_1

8. In the new LogiCAD project there are no external variables yet. To expose data points from the L-IOB device to the logic program, activate the **PLC** check box of the corresponding data points in the Configurator, e.g., an I/O data point, a network variable, a BACnet server object, or a user register.

Datapoint Name	No.	PLC	Direction	Register Name
L1_1_UI2_Input_Read	1	<input checked="" type="checkbox"/>	In	L1_1_UI2_Input
L1_1_UI2_IOStatus_Read	2	<input type="checkbox"/>	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. 'Local IO' for all local I/Os that are exposed as a PLC variable. An example is shown in Figure 7.

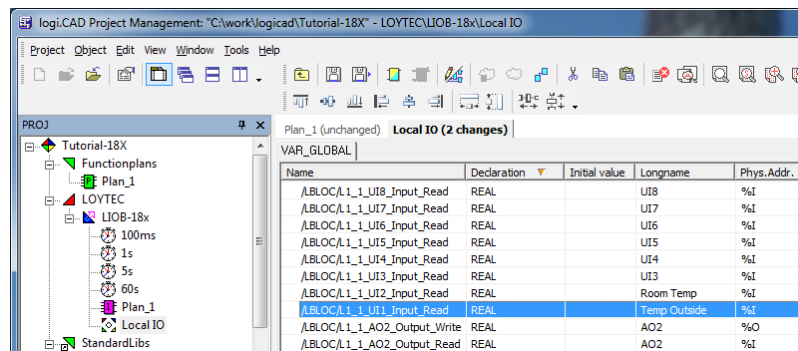


Figure 7: Exposed PLC data points appear in LogiCAD

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 8.

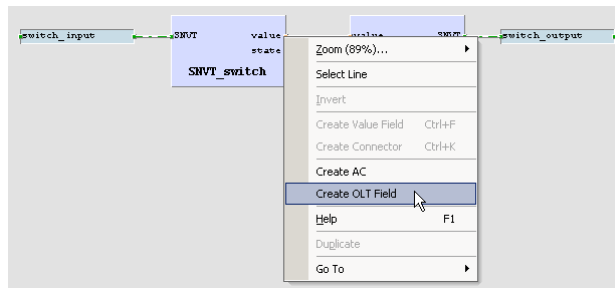


Figure 8: Create online test fields

13. Place the fields above and below the drawing as shown in Figure 9, then press the **Save** button to save your changes.

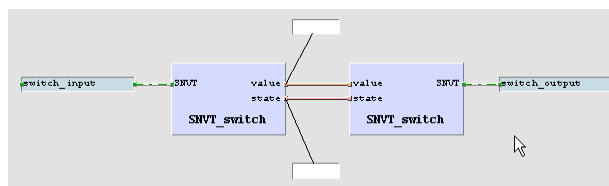


Figure 9: Online Test fields

14. Finally, open the context menu of the **LIOB-18x** (48x/58x) 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 **LIOB-18x** and select **Download** from the context menu. A connection dialog will appear and ask for the type of connection and additional information.

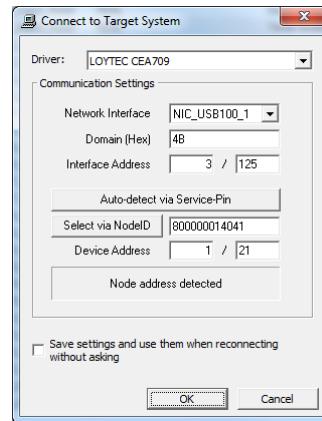


Figure 10: IEC61131 program download

16. Select the LOYTEC CEA709 communication driver and enter the requested communication settings as described in Section 9.4.2. Start the download process by pressing the **OK** button.
17. After the download completed, check the PLC Status in the LCD display. If it is not “Running”, the logic is not yet running. Reboot the device to start the IEC61131 program.

3 Hardware Installation

3.1 Enclosure

The L-IOB enclosures are 107 mm wide for DIN rail mounting, following DIN 43 880. Figure 11 shows the LIOB-180 enclosure as an example.

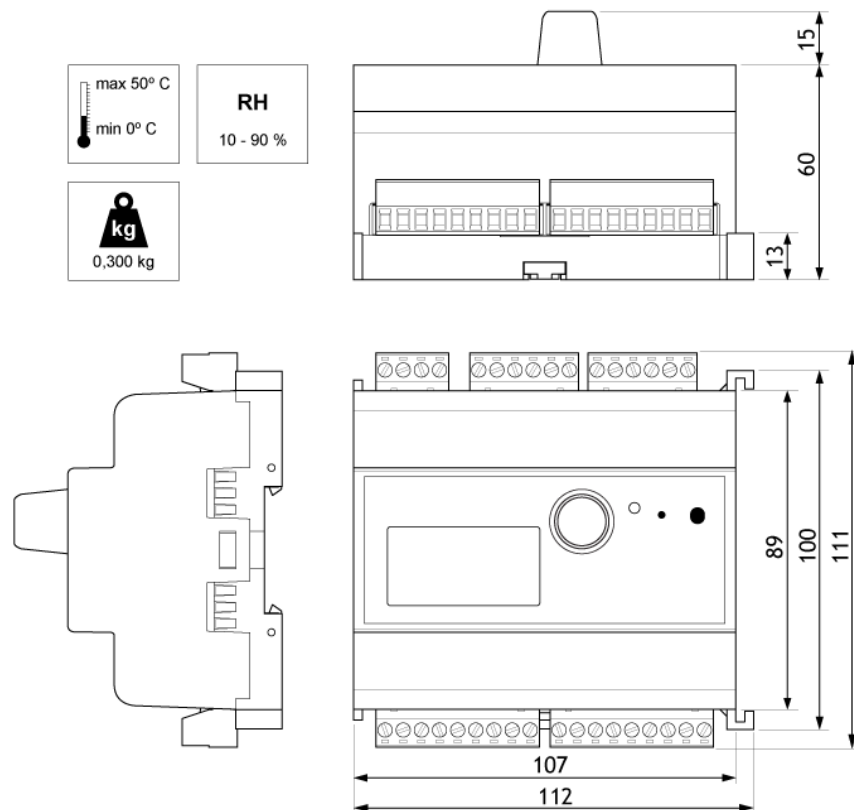


Figure 11: LIOB-180 Enclosure (dimensions in mm)

3.2 Product Label

The product label on the side of the L-IOB I/O controller contains the following information (see Figure 12):

- L-IOB order number (e.g.: LIOB-180),
- Date Code, which defines the production week and year,

- Serial number with bar-code (SER#),
- Node ID of the device. Models with an Ethernet/IP interface additionally contain the MAC address on the label.

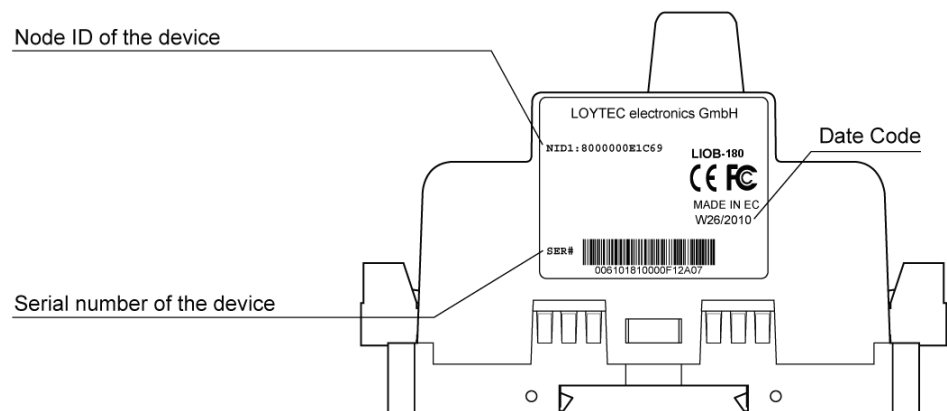


Figure 12: L-IOB product label

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

3.3 Mounting

All L-IOB devices come prepared for mounting on DIN rails following DIN EN 50 022. The devices can be mounted in any position. However, an installation place with proper airflow must be selected to ensure that the L-IOB’s temperature does not exceed the specified range (see Section 15.3).

3.4 Terminal Layout

The L-IOB device provides pluggable screw terminals to connect to sensor and actuator hardware. The screw terminals can be used for wires of a maximum thickness of 2.5 mm²/AWG13. Refer to Section 15.1 for the specification of all I/O hardware types. The next sections show the terminal layouts for the different L-IOB models.

3.4.1 LOIB-180

The terminal layout of the LOIB-180 is shown in Figure 13.

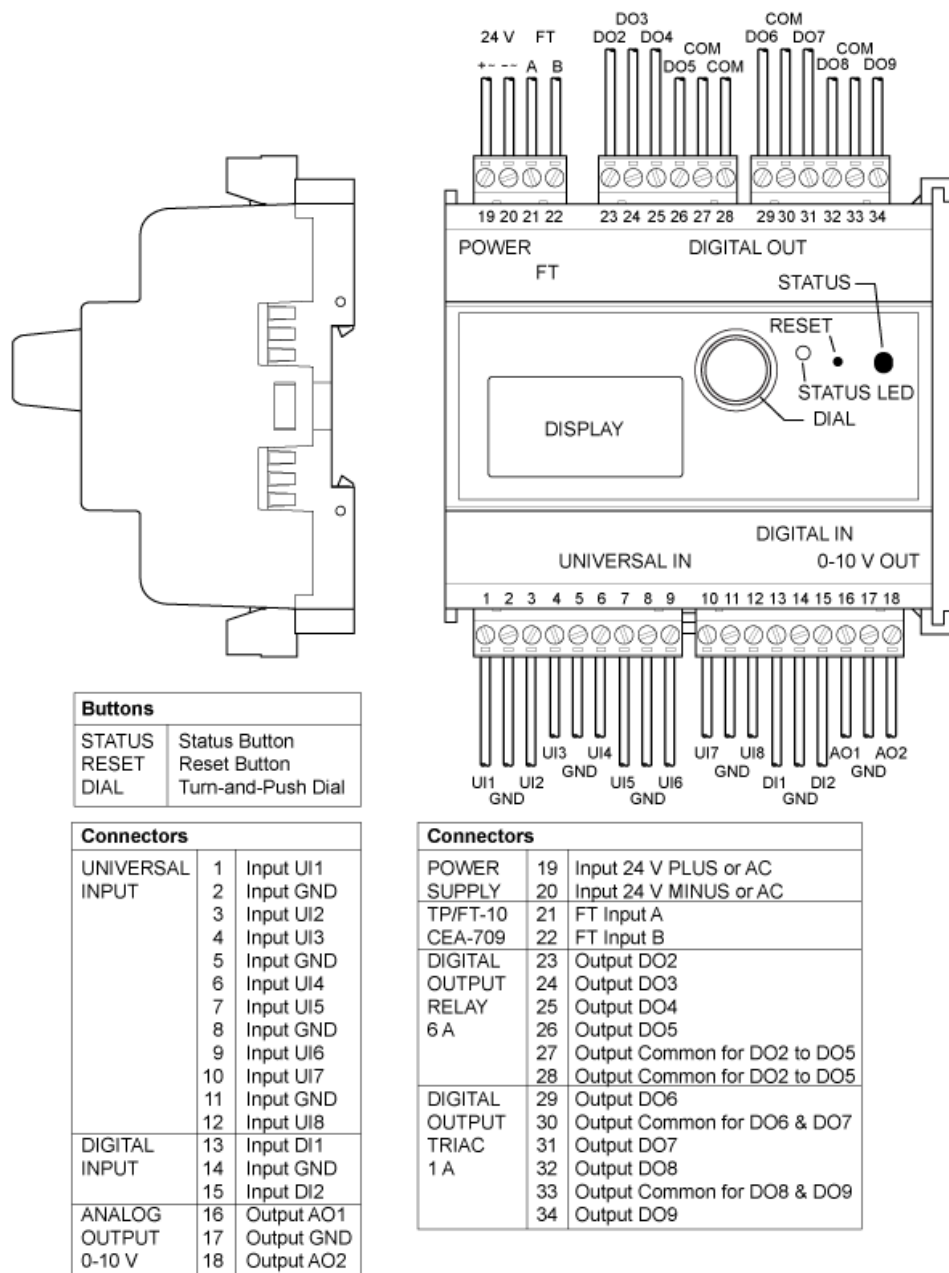


Figure 13: LOIB-180 terminal layout

3.4.2 LIOB-181

The terminal layout of the LIOB-181 is shown in Figure 14.

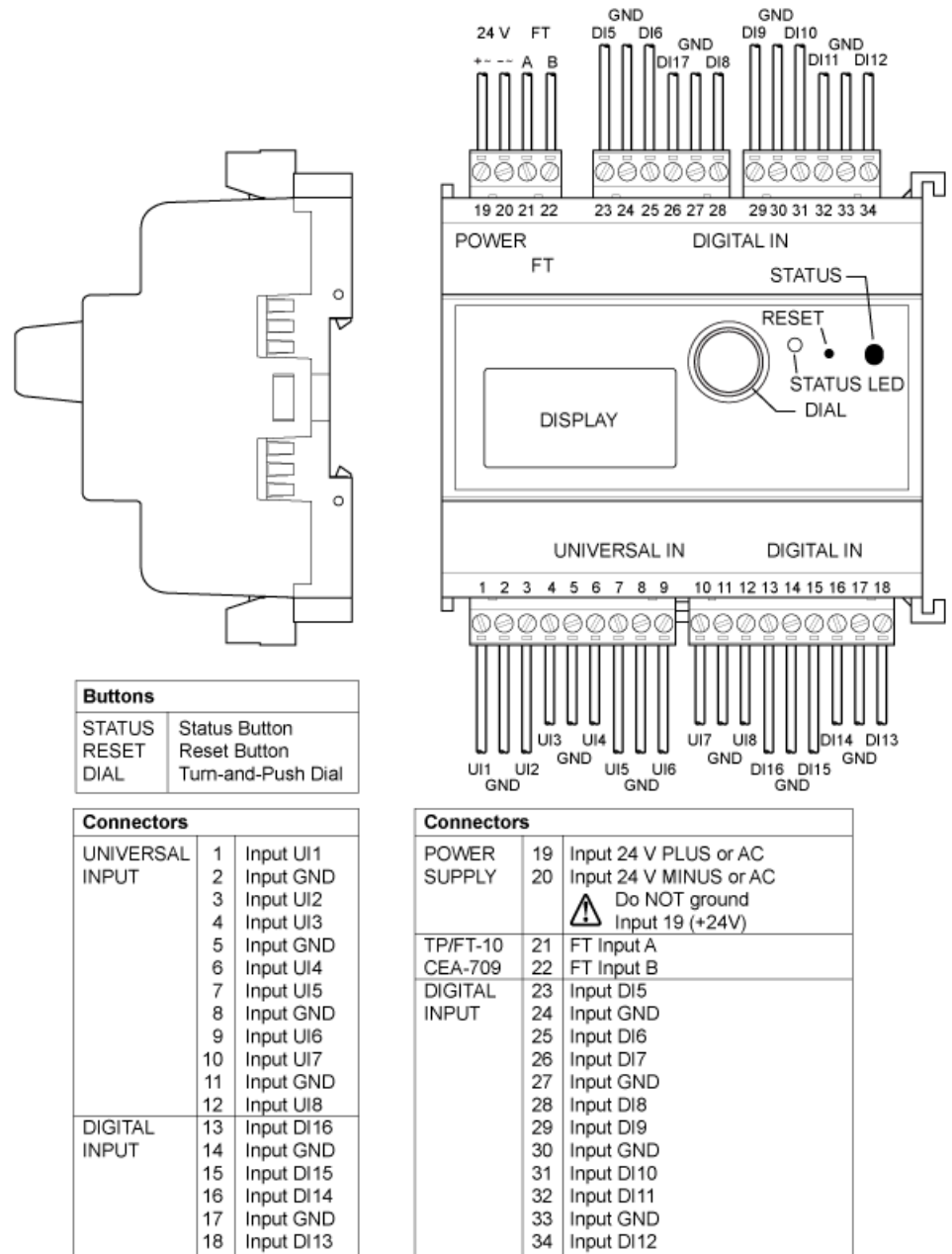


Figure 14: LIOB-181 terminal layout

3.4.3 LIOB-182

The terminal layout of the LIOB-182 is shown in Figure 15.

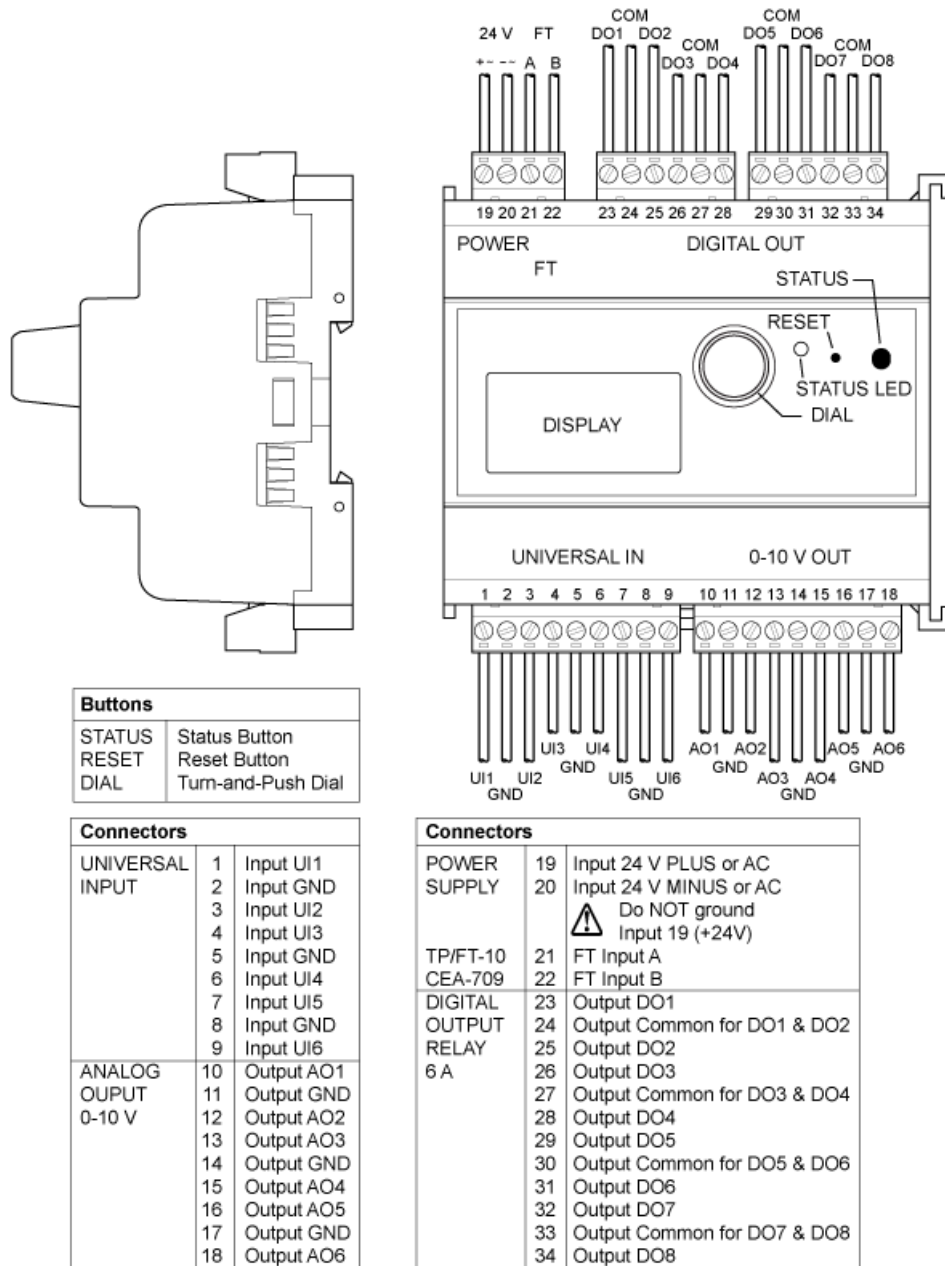


Figure 15: LIOB-182 terminal layout

3.4.4 LIOB-183

The terminal layout of the LIOB-183 is shown in Figure 16.

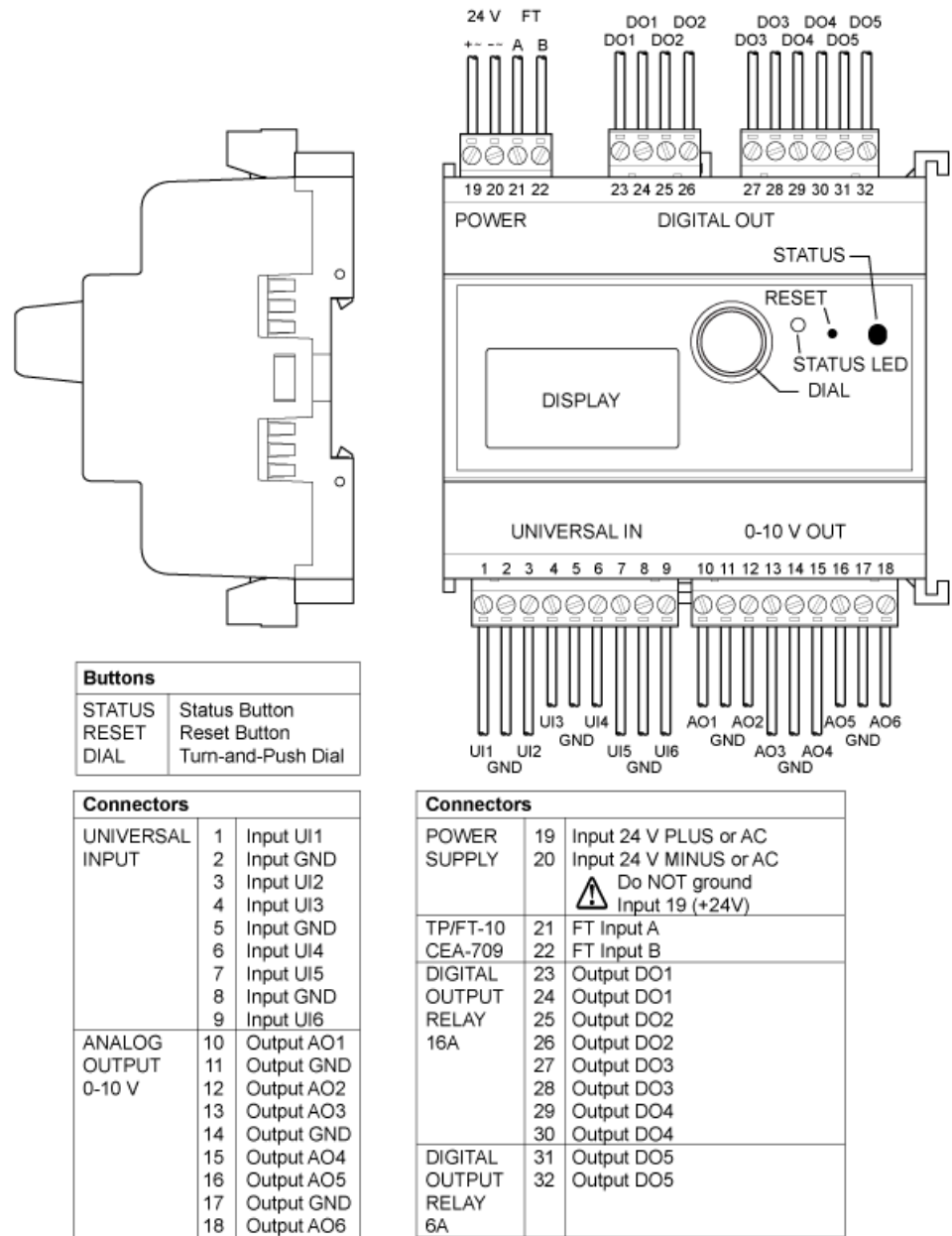


Figure 16: LIOB-183 terminal layout

3.4.5 LIOB-184

The terminal layout of the LIOB-184 is shown in Figure 17.

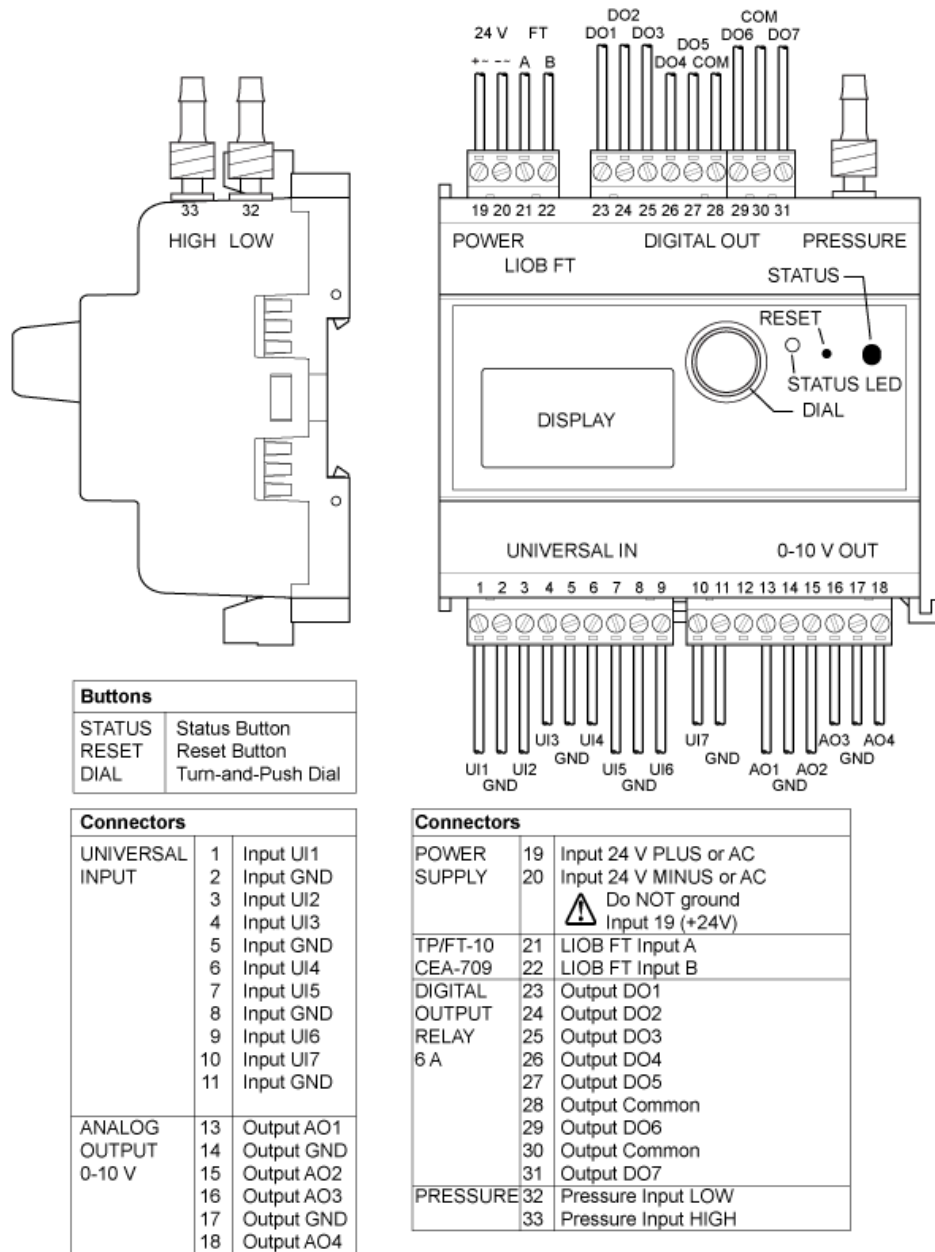


Figure 17: LIOB-184 terminal layout

3.4.6 LIOB-480/580

The terminal layout of the LIOB-480/580 is shown in Figure 18.

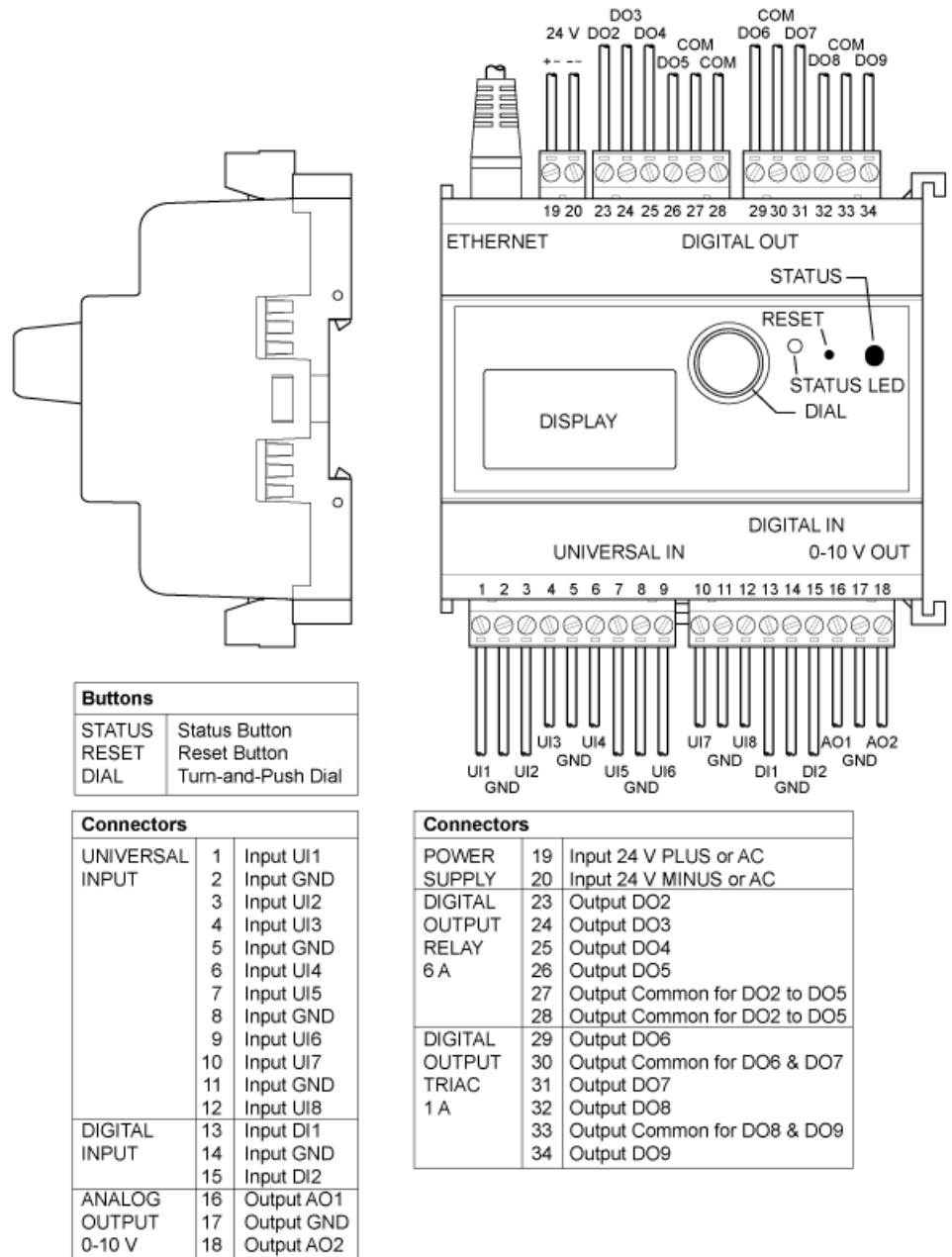


Figure 18: LIOB-480/580 terminal layout

3.4.7 LIOB-481/581

The terminal layout of the LIOB-481/581 is shown in Figure 19.

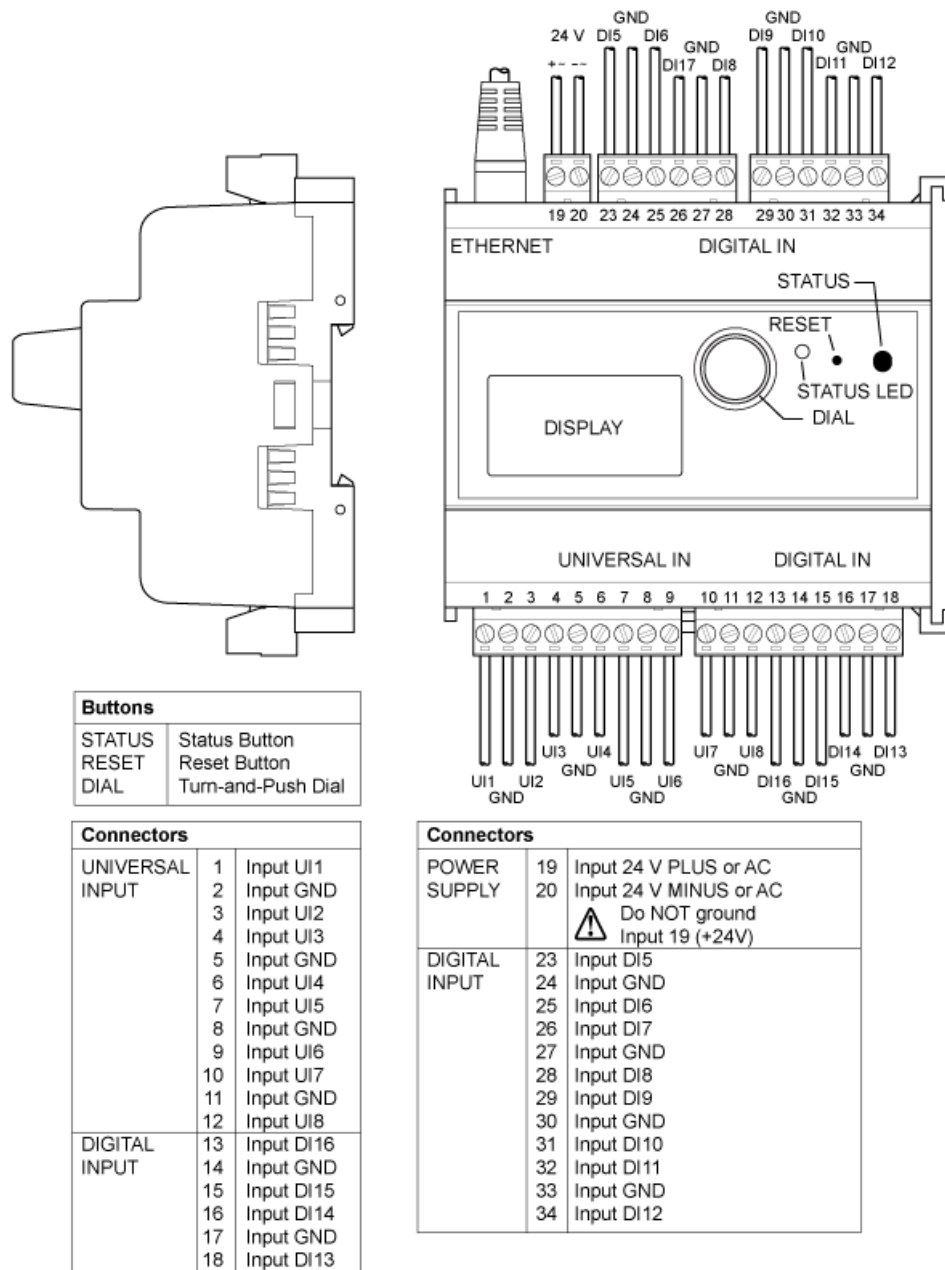


Figure 19: LIOB-481/581 terminal layout

3.4.8 LIOB-482/582

The terminal layout of the LIOB-482/582 is shown in Figure 20.

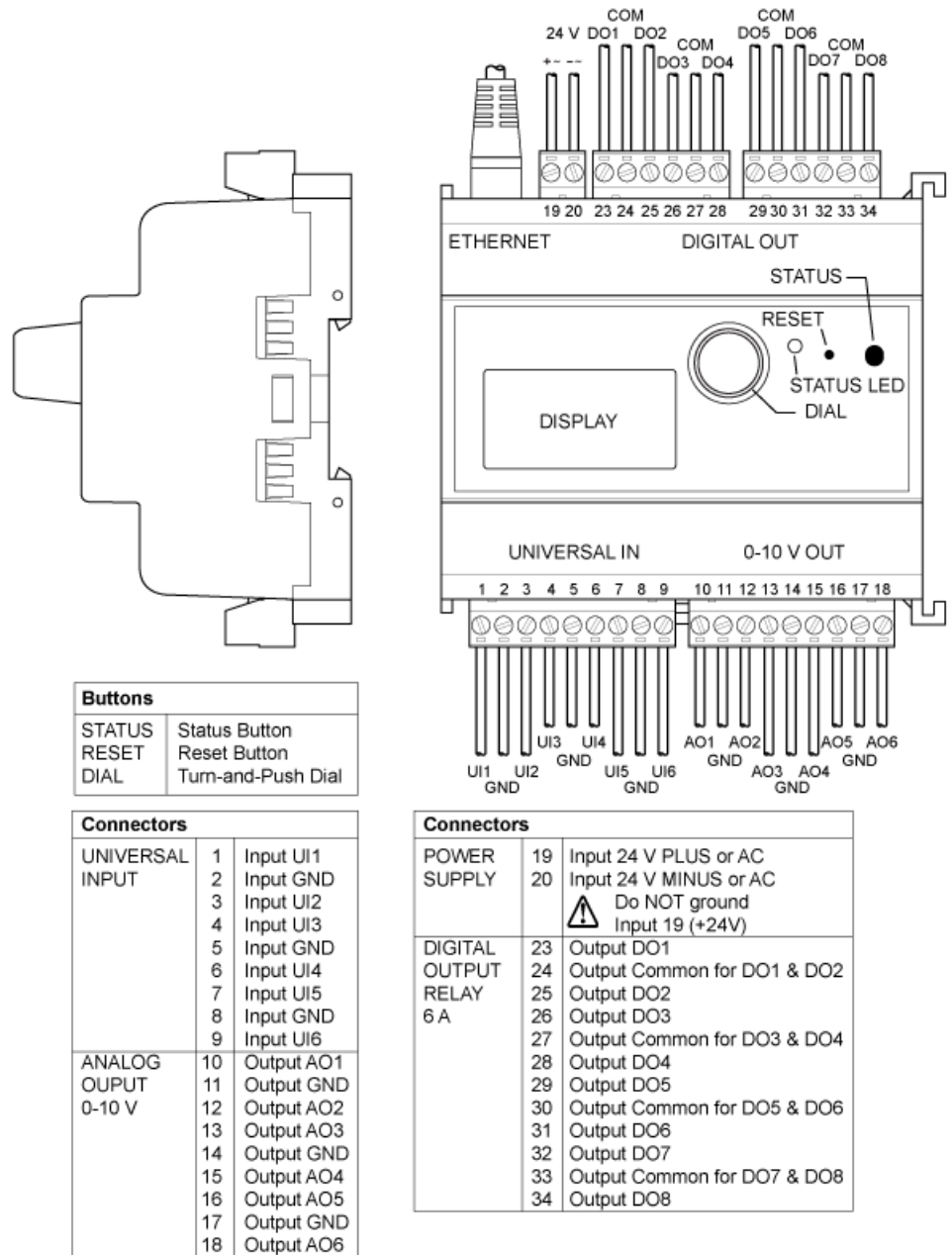


Figure 20: LIOB-482/582 terminal layout

3.4.9 LIOB-483/583

The terminal layout of the LIOB-483/583 is shown in Figure 21.

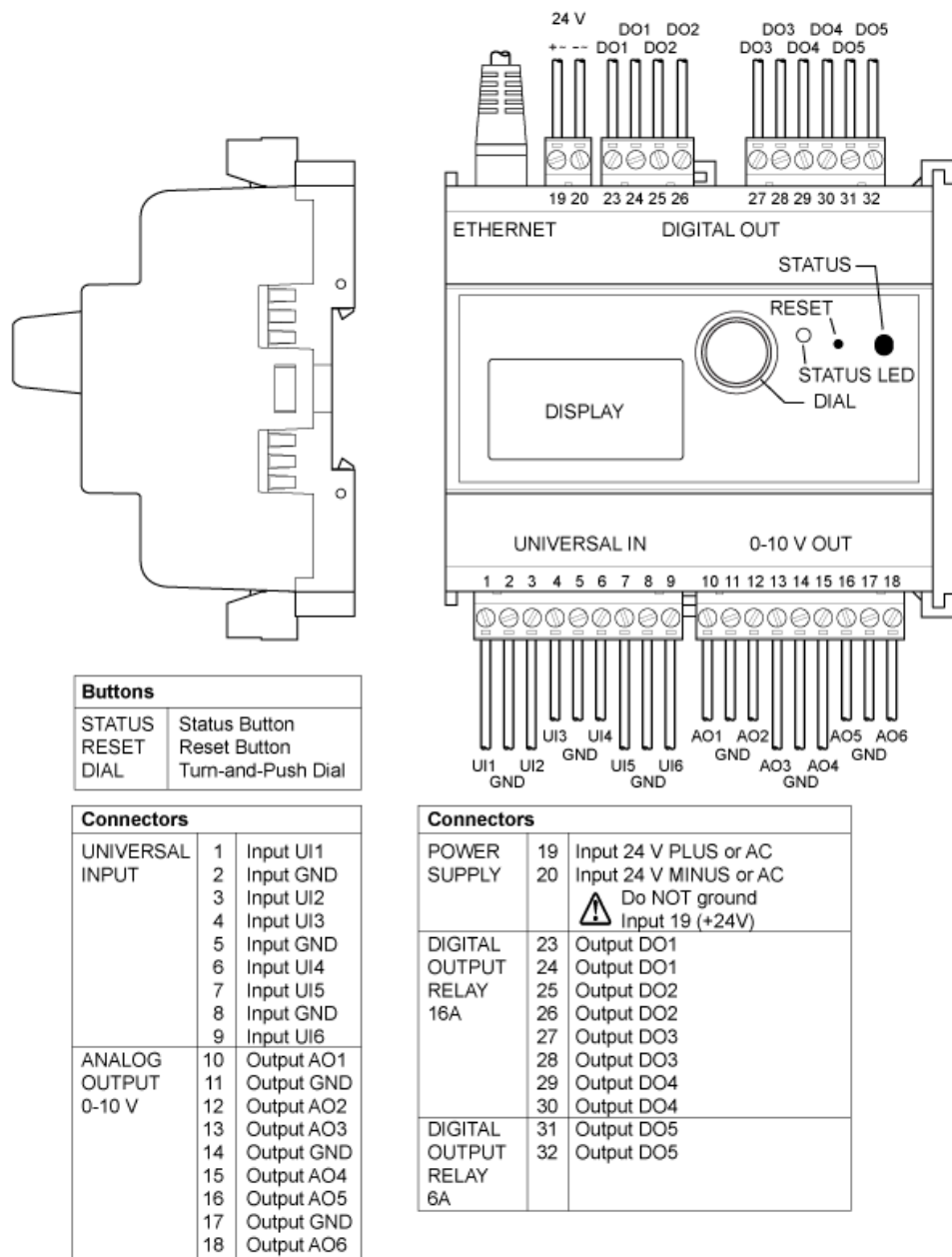


Figure 21: LIOB-483/583 terminal layout

3.5 Power Supply and Wiring

There are four ways of connecting LIOB-18x/48x/58x devices along with all other nodes in a network:

- CEA-709 Free Topology (LIOB-18x),
- CEA-709 Bus Topology (LIOB-18x),
- CEA-852 (LIOB-48x),
- BACnet/IP (LIOB-58x).

All rules of CEA-709 / CEA-852 / LONMARK® devices or BACnet devices regarding network installation, management, and maintenance apply. For using external (non-LOYTEC) power supplies, please refer to Section 13.1. For connecting sensors and actuators to the L-IOB I/Os, please refer to Sections 13.2 and 13.3. The next sections describe the different power supply and cabling variants in detail.

3.5.1 CEA-709 Network Connection in Free Topology

Figure 22 shows the connection of LIOB-18x controllers in free topology, which can be used for cable lengths of up to 500 m between any two devices. As an example for other nodes in the network, a L-INX device is connected on top.

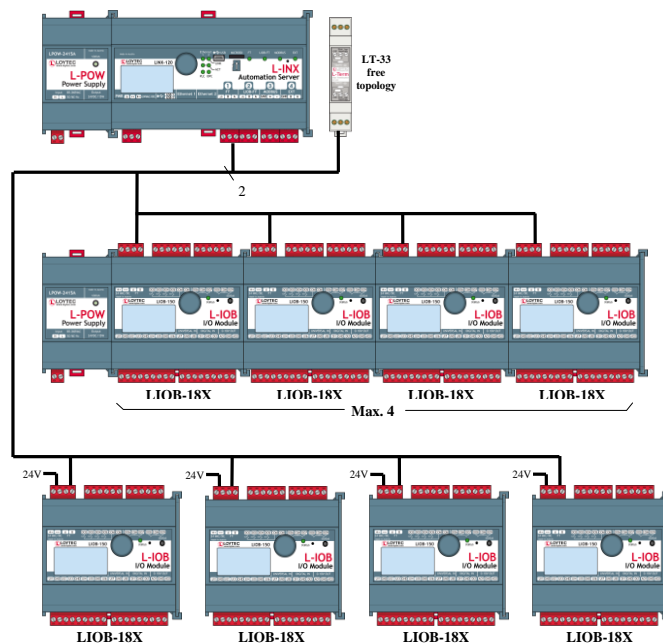


Figure 22: LIOB-18x in Free Topology

The L-IOB devices can either be powered by L-POWs (middle part of Figure 22) or other 24 V power supplies (bottom part of Figure 22). One LT-33 terminator (free topology terminals) must be placed somewhere in the network.

3.5.2 CEA-709 Network Connection in Bus Topology

Figure 23 shows the connection of LIOB-18x controllers in bus topology, which must be used for a cable length of above 500 m. As an example for other nodes in the network, a

L-INX device is connected on the top end of the bus. The L-IOB devices can either be powered by L-POWs (middle part of Figure 23) or other 24 V power supplies (bottom part of Figure 23). An LT-33 terminator (bus topology terminals) must be placed at both ends of the bus.

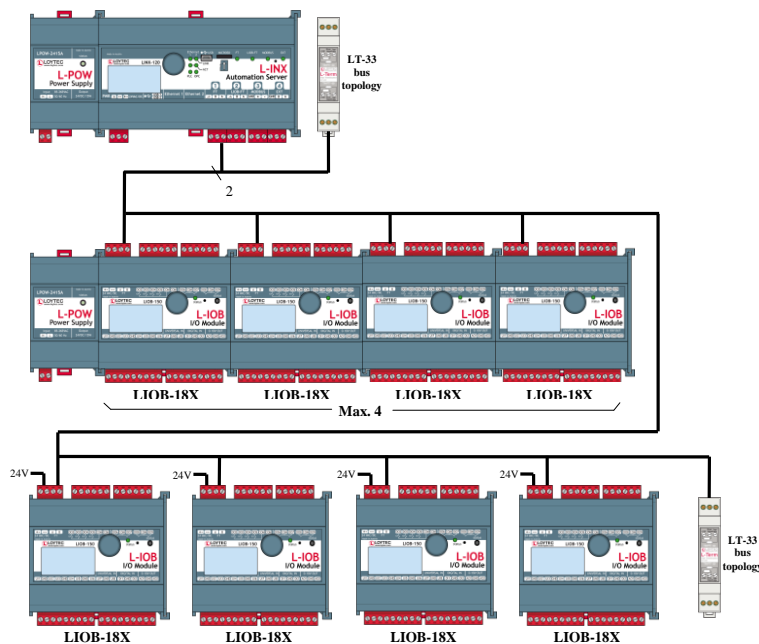


Figure 23: LIOB-18x in Bus Topology

3.5.3 CEA-852 Network Connection

Figure 24 shows the connection of LIOB-48x controllers over Ethernet/IP. The L-IOB devices can either be powered by L-POWs (middle part of Figure 24) or other 24 V power supplies (bottom part of Figure 24). The L-INX device at the top is an example of a CEA-852 configuration server, which is required in every CEA-852 network.

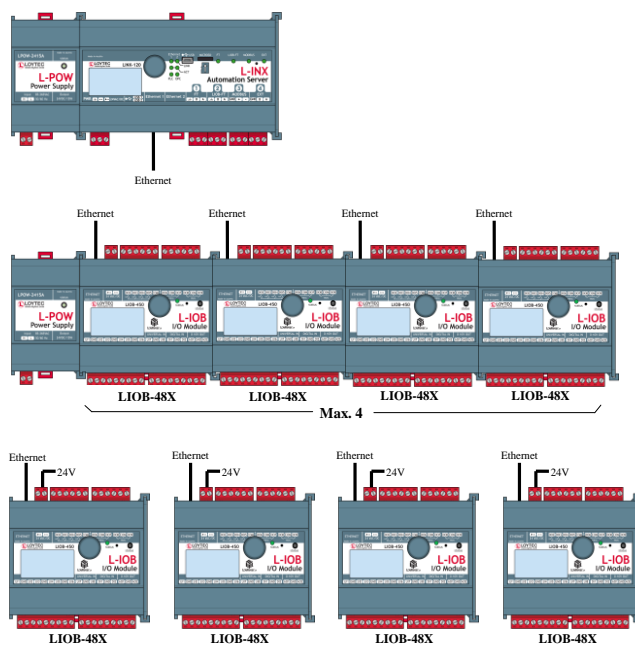


Figure 24: LIOB-48x Connection

3.5.4 BACnet/IP Network Connection

Figure 25 shows the connection of LIOB-58x controllers over Ethernet/IP. The L-IOB devices can either be powered by L-POWs (top part of Figure 25) or other 24 V power supplies (bottom part of Figure 25).

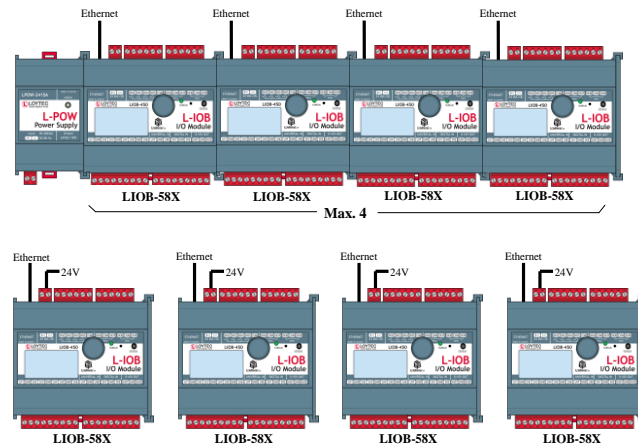


Figure 25: LIOB-58x Connection

3.5.5 Extension of LIOB-48x/58x controllers with LIOB-45x/55x device

The I/Os of LIOB-48x and LIOB-58x controllers can be extended with one LIOB-45x or LIOB-55x device (in L-INX mode) by using the LIOB-IP bus of the controller. Please refer to the LIOB-10x/x5x User Manual [2] for how to connect a LIOB-45x/55x device to a L-IOB host (in this case the LIOB-48x/58x controller).

3.6 LED signals

The L-IOB device is equipped with a three-color status LED (see Figure 13) which indicates the current state of the L-IOB device.

3.6.1 Status LED of LIOB-18x

The meaning of the LED signals for the LIOB-18x models is listed in Table 1.

Behavior	Description	Comment
OFF	Online	The L-IOB device is online.
Flickering GREEN	Traffic	The L-IOB device receives or transmits packets.
ORANGE	Manual Mode	At least one I/O is in manual mode.
RED	Error	An error has occurred (e.g. a sensor is disconnected or signals an error).
GREEN flashing at 0.5 Hz	Offline	The L-IOB device is offline.
RED flashing at 0.5 Hz and "LIOB Fallback" shown in LCD UI	Fallback Override	The primary firmware image is corrupt and the L-IOB has booted the fallback image. In this case, the firmware must be upgraded again.

Table 1: Status LED Patterns of LIOB-18x

3.6.2 Status LED of LIOB-48x

The meaning of the LED signals for the LIOB-48x models is listed in Table 2.

Behavior	Description	Comment
GREEN	Online	The L-IOB device is online.
Flickering GREEN	Traffic	The L-IOB device receives or transmits packets.
ORANGE	Manual Mode or no CS	At least one I/O is in manual mode or the configuration server cannot be contacted.
RED	Error	An error has occurred (e.g. a sensor is disconnected or the configuration server has rejected the device).
GREEN flashing at 0.5 Hz	Offline	The L-IOB device is offline.
ORANGE flashing at 0.5 Hz	Offline, no CS	The L-IOB device is offline and the configuration server cannot be contacted.
RED flashing at 0.5 Hz	No CEA-852 Config	The CEA-852 port is not configured. The device must be added to a CEA-852 IP channel.
RED flashing at 0.5 Hz and "LIOB Fallback" shown in LCD UI	Fallback Override	The primary firmware image is corrupt and the L-IOB has booted the fallback image. In this case, the firmware must be upgraded again.

Table 2: Status LED Patterns of LIOB-48x

3.6.3 Status LED of LIOB-58x

The meaning of the LED signals for the LIOB-58x models is listed in Table 3.

Behavior	Description	Comment
OFF	No Traffic	No packets are received or transmitted.
Flickering GREEN	Traffic	The L-IOB device receives or transmits packets.
ORANGE	Manual Mode	At least one I/O is in manual mode.
RED	Error	An error has occurred (e.g. a sensor is disconnected).
RED flashing at 0.5 Hz and "LIOB Fallback" shown in LCD UI	Fallback Override	The primary firmware image is corrupt and the L-IOB has booted the fallback image. In this case, the firmware must be upgraded again.

Table 3: Status LED Patterns of LIOB-58x

3.7 Status Button

The L-IOB device is equipped with a status button (see Figure 13). When pressing the status button shortly during normal operation of the device, a service pin message (LIOB-18x/48x) or I-Am message (LIOB-58x) is sent out, the LCD is reset, and the LCD backlight is switched on.

The status button can also be used to switch the device back to factory default state. Press the button and power-cycle the device. Keep the button pressed until the Status LED illuminates orange permanently. Release the button within five seconds from that time on to reset the device to factory defaults.

4 LCD Display

The L-IOB device is equipped with an LCD display and jog dial for monitoring, testing, and configuration purposes. The LCD backlight is automatically switched off after 30 minutes of idle time (no jog dial or status button activity).

4.1 Main Page

The main page of the LCD UI of a LIOB-18x is shown in Figure 26. It displays the device's name, CPU load, system temperature, supply voltage, and Programmable Logic (PLC) status. In the row where the CPU load is displayed, there is a language symbol on the right side, where the LCD language can be chosen. Observe that changing the language requires a system reboot. In the LIOB-48x/58x LCD UI, the IP address and Ethernet status is shown instead of the PLC status.

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 **I/O** »» menu is described in Sections 4.2, 4.3, and 4.4. The **Datapoints** »» menu allows browsing through the data points on the device.

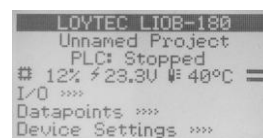


Figure 26: Main Screen of the LCD UI

LIOB-48x/58x devices can additionally host one LIOB-45x/55x device in LIOB-IP mode. In this case, there will be an additional **LIOB-IP** menu on the main screen which is described in Section 4.5.

The **Device Settings** »» menu allows configuring basic device settings. Navigate e.g. to the **Device Management** »» sub-menu, which is displayed in Figure 27.

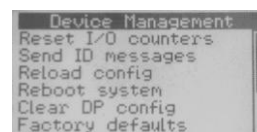


Figure 27: Device Management Menu on the LCD UI

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

- **TCP/IP Setup (LIOB-48x/58x):** IP Configuration page (IP-Address, etc.).
- **CEA-709 over IP (LIOB-48x):** CEA-852 setup page. Here the user can setup the LIOB-48x device to become part of a CEA-852 (CEA-709 over IP) channel. Further settings can be made in the Web UI, see Section 5.2.6.
- **Send ID messages:** send out a service pin message (LIOB-18x/48x) or I-Am message (LIOB-58x).
- **Reboot system:** By choosing this item, the device performs a full reboot.
- **Clear DP config:** By choosing this item, the user can clear the device's entire data point configuration.
- **Factory Defaults:** By choosing this item, the user can reset the entire device to its factory default.
- **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:** changes the display contrast.
- **Language:** changes the LCD language. Observe that this requires a device reboot.
- **Reset I/O counters:** resets all I/O counters like pulse count values.
- **LONMARK Device Mode (LIOB-18x/48x):** the LIOB-18x/48x device is configured to behave like a LONMARK® certified LIOB-15x/45x device, see Section 6.2.

4.2 I/O Page

The I/O page of the L-IOB LCD display is shown in Figure 28. The top and bottom rows show the direction, state, and operating mode (without a letter = Auto, **M** = Manual, **O** = Override, **D** = Disabled) of all I/Os.

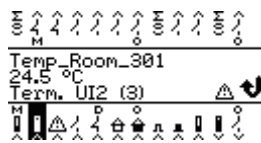


Figure 28: L-IOB LCD Display Main Page

The I/O state is shown as one of the following icons:

- **Switch Icon:** used for I/Os which generate a digital value as well as Inputs in Switch Mode,
- **Bar Icon:** used for I/Os which generate an analog value,
- **Pulse Icon:** used for pulse count inputs,
- **House Icon:** used for occupancy inputs,
- **Exclamation Mark:** shown for disconnected sensors or sensors which indicate an error,
- **Check Icon:** shown for sensors which indicate correct operation,

- **COM Icon:** used for all common terminals of relays and triacs.
- **CD Icon:** used for STId card reader code signals.
- **DT Icon:** used for STId card reader data signals.
- **CK Icon:** used for STId card reader clock signals.

Refer to Section 6.7 for details on I/O types, configuration, and operating modes.

When an I/O is selected, the middle of the main page shows the I/O name, current value, terminal name, and terminal number. To the right, the exit icon and I/O state icon is shown. When the exit icon is selected, the middle of the main page shows the device name. The device state icon shows an exclamation mark if at least one I/O shows an exclamation mark. Otherwise it shows the checked icon to indicate that all I/Os are operating correctly.

By turning the jog dial, the user can cycle through all I/Os. This can be used to get a quick overview of all I/O states. Observe that the common terminal icons (COM) only show the name and information of that terminal. No further configuration is possible for common terminals.

4.3 Manual / Quick Edit Mode

If the jog dial is pushed shortly on an I/O in manual mode (**M**), the quick edit mode is entered, which allows changing the I/O value by turning the jog dial. By pushing the jog dial shortly again, the quick edit mode is left. When pushing the jog dial shortly on an I/O in auto mode (normal mode without special letter), one can quickly switch to both manual mode *and* quick edit mode by turning to the jog dial. The manual mode (along with the quick edit mode) can be left again by pushing the jog dial for at least one second. The manual mode can also be setup in the corresponding I/O configuration page (see Section 4.4).

If an input is in manual mode, the physical input from the connected sensor is ignored and the user can setup a simulated input value to be used in the logic application. This can be used e.g. to test the behavior of the application in the L-IOB device depending on certain input values. If an output is in manual mode, the value coming from the logic application is ignored and the user can set the value for the actuator connected to the physical output. This can be used to physically test the connected actuator.

It is possible that changing the manual value is restricted via a PIN code. In this case, the user will be requested to enter a pin code before the value can be changed. The pin code only needs to be entered once except when the device is not operated manually for more than 30 minutes.

4.4 I/O Configuration

If the jog dial is pushed for at least one second on any I/O, the configuration page for that I/O is entered, which allows viewing and changing configuration properties of the I/O. The properties which can be changed are enclosed by angle brackets (“<”, “>”). By turning the jog dial, the user can cycle through the configuration properties. If the jog dial is pushed shortly on a property, the edit mode is entered, which allows changing the property by turning the jog dial. By pushing the jog dial again, the edit mode is left.

It is possible that changing properties is restricted via a PIN code. In this case, the user will be requested to enter a pin code before a value can be changed. The pin code only needs to be entered once except when the device is not operated manually for more than 30 minutes.

To leave the configuration page, the user must turn the jog dial until the title line (I/O name) is selected and then push the jog dial. Alternatively, the jog dial can also be pushed anywhere on the page for at least 1 second.

Observe that depending on the hardware type, signal type, and interpretation of an I/O, the list of configuration properties varies. Refer to Section 6.7 for detailed information on the different configuration properties. In addition to the configuration properties described there, for some I/Os, a “RawValue” property will be displayed. It shows the physical measured value for inputs (e.g. the resistance of an NTC) resp. the physical value for outputs (e.g. the actual voltage for analog outputs). Note that this information can be used for debugging sensors or actuators but is not available in the form of data points.

For counting inputs, two additional configuration options are available: ‘Pulse Count Reset’ and ‘Count Start Value’. Using ‘Pulse Count Reset’, the counter can either be reset to 0 or to the value set in ‘Count Start Value’.

4.5 LIOB-IP Page (LIOB-48x/58x)

LIOB-48x/58x devices can additionally host one LIOB-45x/55x device in LIOB-IP mode. This requires firmware version 4.8 or higher on the LIOB-48x/58x. The corresponding LCD page can be entered from the main page via **LIOB-IP** or **Device Settings** »» **LIOB-IP** »».

Even without any configuration, the LIOB-IP bus can be scanned to check if a LIOB-45x/55x device is connected to the L-IOB host. In the **LIOB-IP** page, select the item **Scan LIOB bus** to scan for an attached LIOB-45x/55x device. At the end of the scan process, the LCD will show the detected L-IOB device and its status resp. error state. By pushing the jog dial on the detected L-IOB device, some configuration properties of the device are displayed.

If a configuration is downloaded to the L-IOB host using the Configurator software, a configuration run is started automatically and if the configuration matches the physically attached LIOB-45x/55x device, it should go online. The configuration run can also be started manually at any time by selecting **Configure LIOBs** in the **LIOB-IP** page.

The attached LIOB-45x/55x Device can be enabled and disabled by pushing the jog dial on the device (shown in the **LIOB-IP** page) and choosing **Enable** or **Disable**. To activate the new setting, a configuration run must be started afterwards by choosing **Configure LIOBs** as described above.

The LCD Display of a connected LIOB-45x/55x device can be remotely accessed by pushing the jog dial on the device (shown in the **LIOB-IP** page) and choosing **Remote Display**. To leave the remote display mode again, either push and hold the jog dial for at least 10s and then release it or exit via the **Remote LCD Access** menu item in the device configuration page of the LIOB-45x/55x device.

5 Web Interface (LIOB-48x/58x)

The LIOB-48x/58x models come with a built-in Web server and a Web interface to configure the device and extract statistics information. The Web interface allows configuring the IP settings, CEA-852, BACnet, and other 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 Windows 7 or Windows 8, 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 29.

The device information page shows information about the device and the current firmware version. Below, it shows operational parameters, such as free memory, CPU load, system temperature and supply voltage. The project information area shows details on the currently loaded project. The IEC61131 Info area shows the current status of the logic program. The L-IOB Info area shows the status of the local I/Os and LIOB-IP bus.

The page also includes 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

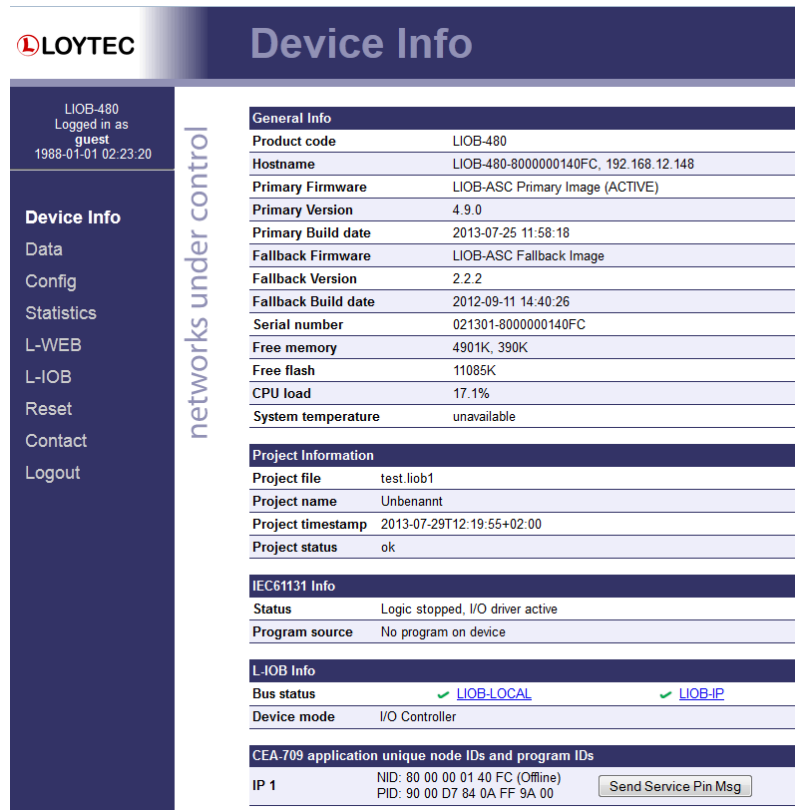


Figure 29: Device Information Page

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 30. Enter the default administrator password 'loytec4u' and select **Login**.

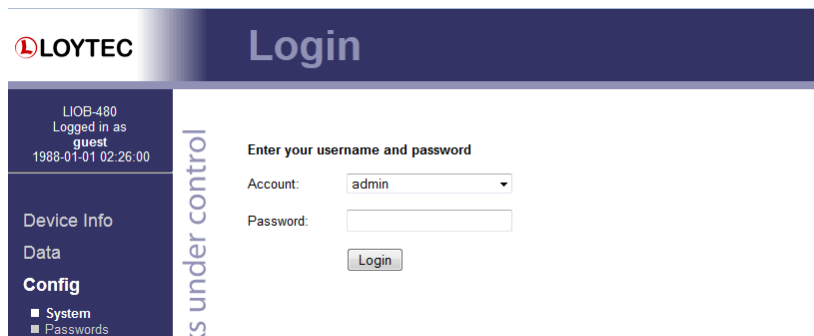


Figure 30: Enter 'loytec4u' as the default administrator password

The Config menu opens. Click on **Passwords** in the Config menu, which opens the password configuration page as shown in Figure 31. 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.

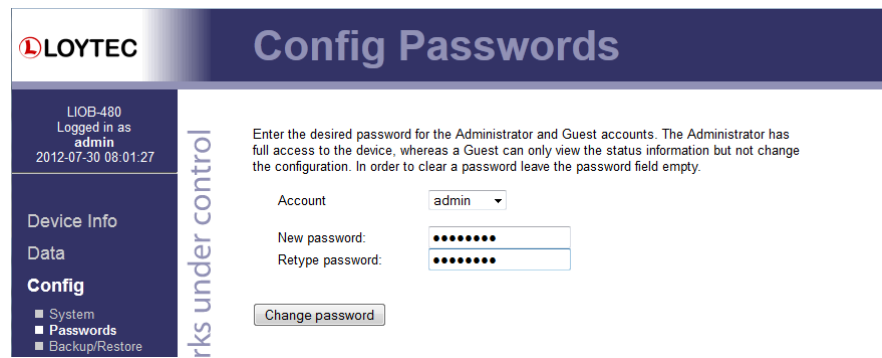


Figure 31: Password Configuration Screen

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 32. 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** (LIOB-58x) or **LonMark** (LIOB-48x). In the **auto** mode, the device switches to the first external time source that is discovered. Possible external time sources are NTP, BACnet, or 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 5.2.4).

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 10.3.8). 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 32: 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

The section **System Parameters** allows defining the displayed units and CSV separator. The displayed units can be either primary or alternative. The primary unit is the primary engineering unit of a data point. The alternative display unit can be configured. This setting decides, which units are displayed on the data point Web pages. 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-801/900 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 33 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,
- I/O configuration and parameters,
- AST settings,
- L-WEB configuration and custom Web pages,
- IEC61131 logic program and retain variables.

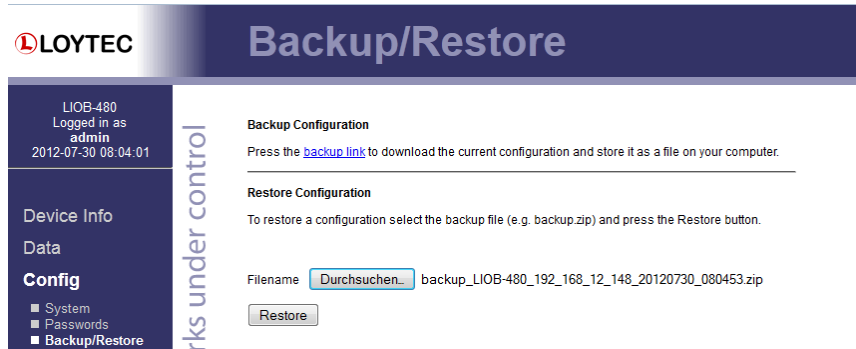


Figure 33: Backup/Restore page

5.2.3 Port Configuration

This menu allows configuring the device’s communications ports. For the Ethernet communication port, a configuration tab is provided by the Web UI, as shown in Figure 34. It 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.

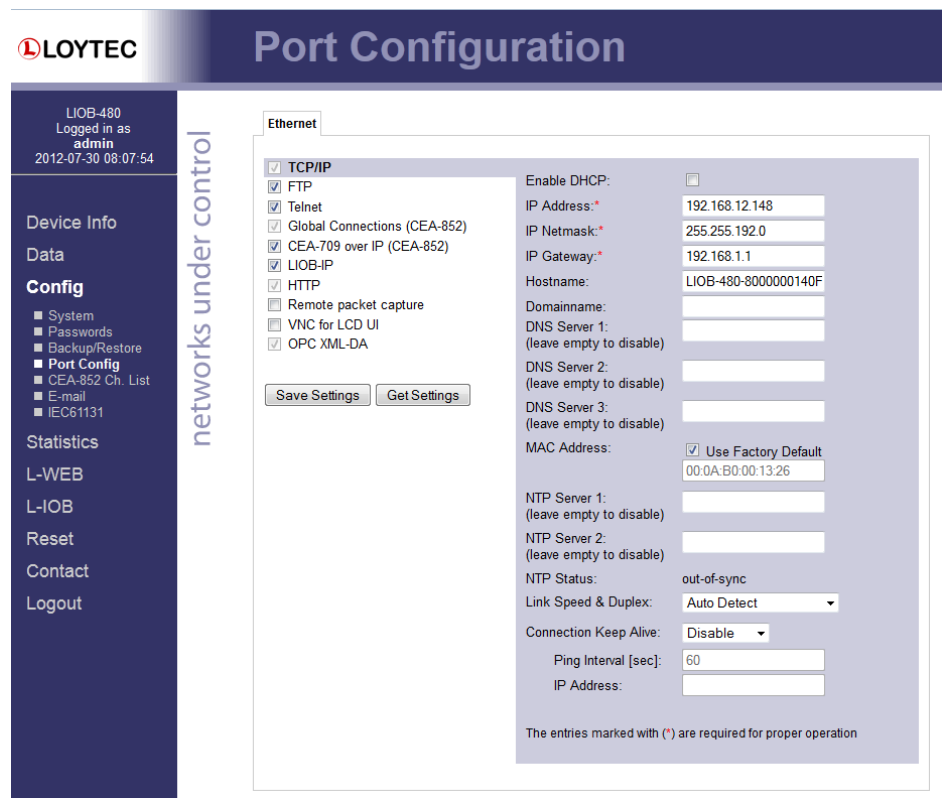


Figure 34: Port Configuration Page

When selecting a protocol, 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 34. The mandatory IP settings, which are needed to operate 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.

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. Further, you can configure up to 3 Domain Name Servers.

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.

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**).

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.

Other standard protocols that are bound to the 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 35. The FTP server is used for instance to update the firmware or to upload a new data point configuration. Note that HTTP for the Web server can only be disabled using the device configuration of the Configurator.

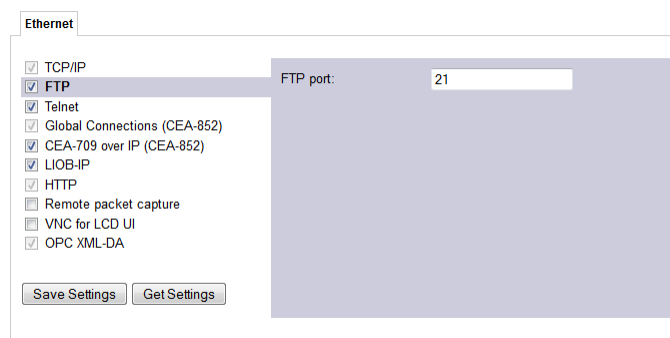


Figure 35: FTP server configuration on the Ethernet port

5.2.5 VNC Configuration

LIOB-48x/58x devices 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 36. 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’.

The screenshot shows the 'Ethernet' configuration page. On the left, there is a list of checkboxes for various protocols: TCP/IP, FTP, Telnet, Global Connections (CEA-852), CEA-709 over IP (CEA-852), LIOB-IP, HTTP, Remote packet capture, VNC for LCD UI (checked), and OPC XML-DA. Below this list are 'Save Settings' and 'Get Settings' buttons. On the right, there is a settings box for VNC with the following fields: VNC port (5900), Max VNC clients (1), VNC password (loytec4u), and Admin pin code (0000).

Figure 36: VNC Configuration

5.2.6 CEA-852 Device Configuration (LIOB-48x)

For LIOB-48x devices, the **CEA-709 over IP (CEA-852)** protocol is enabled by default, since it is the only means to communicate in a control network. The CEA-852 protocol settings are displayed in the settings box on the right-hand side as shown in Figure 37. 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 unit of the device and sends its configuration.

The screenshot shows the 'Ethernet' configuration page with the 'CEA.709 over IP (CEA-852)' checkbox selected. The settings box on the right contains the following fields: Config server address (192.168.12.30), Config server port (1629), Config client port (1628), Device name (LIOB-480), Channel mode (Standard), Pri. SNTP server, Sec. SNTP server, Channel timeout [ms] (off), Escrow timeout [ms] (64), Aggregation timeout [ms] (16), MD5 authentication (off), MD5 secret (16 bytes, hex) (unknown), Location string (unknown), NAT Address (checked Automatic NAT), and Multicast Address (leave empty to disable). A warning message is displayed: 'Warning: This internet connection is insecure. All data will be transmitted in clear text. Use the console interface to change the MD5 secret in a secure way.'

Figure 37: 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-IOB device and the IP channel. The field

Config client port represents the IP port of the L-IOB's CEA-852 unit. 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 to learn more about NAT configuration.

In the field **Device name** the user can enter a descriptive name for the L-IOB, 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 unit. 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 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 field **Escrow timeout** defines how long the CEA-852 unit on the L-IOB 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 LIOB.

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 text 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 unit on the L-IOB 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 unit of the L-IOB 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.

5.2.7 Global Connections Configuration

The CEA-852 device used for global connections (see Section 6.5.2) 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 38. For LIOB-48x devices, the settings are shared with the **CEA-709 over IP** settings, if that protocol is enabled.

Otherwise (as well as for LIOB-58x devices), the CEA-852 device is configured on this tab as described in Section 5.2.6.

For a LIOB-48x device, 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 5.2.6. 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.

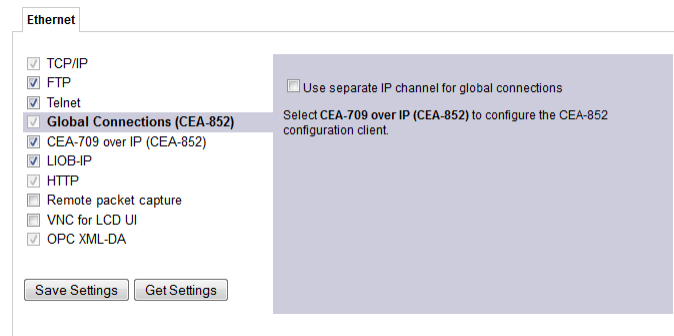


Figure 38: Global Connections Configuration Page

5.2.8 BACnet Configuration (LIOB-58x)

Figure 39 shows the BACnet device configuration page of a LIOB-58x. 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 inter-network.*

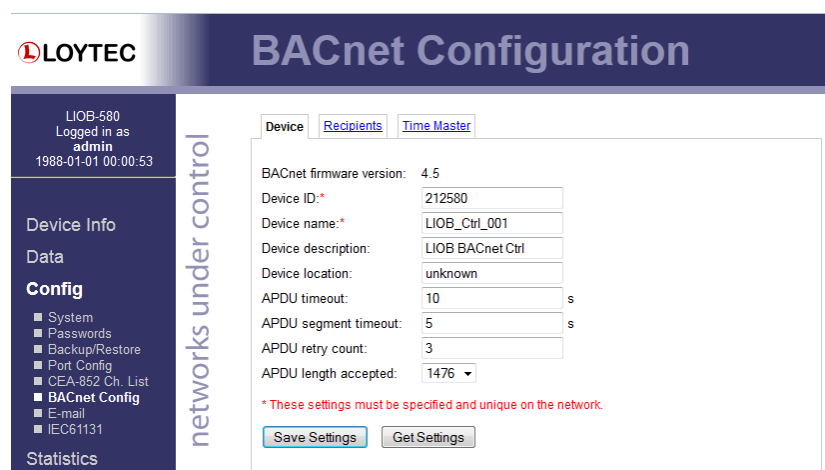


Figure 39: BACnet Device Configuration

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".

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.

5.2.9 BACnet/IP Configuration (LIOB-58x)

For LIOB-58x devices, the **BACnet/IP** protocol is enabled by default, since it is the only means to communicate in a control network. The protocol is available on the Ethernet port. The BACnet/IP protocol settings are displayed in the settings box on the right-hand side as shown in Figure 40. 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.

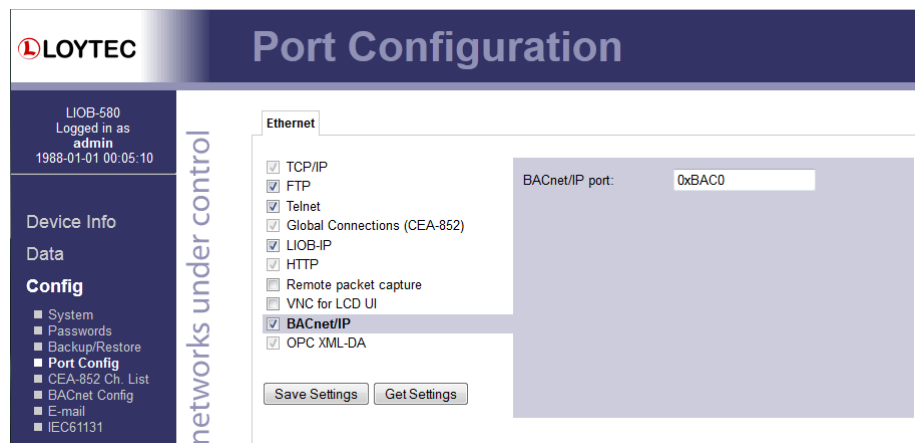


Figure 40: BACnet/IP Configuration

5.2.10 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 41. 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.

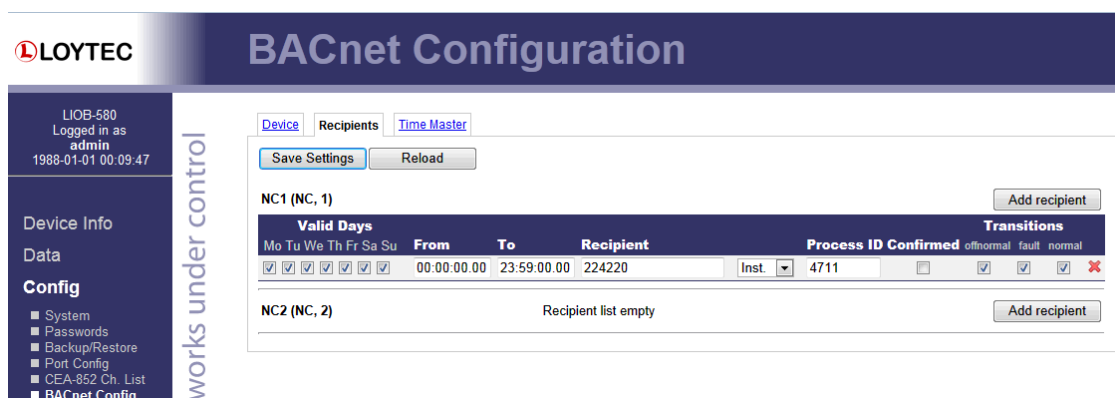


Figure 41: BACnet Recipients Configuration

5.2.11 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 42) 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 10.3.8 for more information on the settings for time sync interval, interval offset and align intervals.

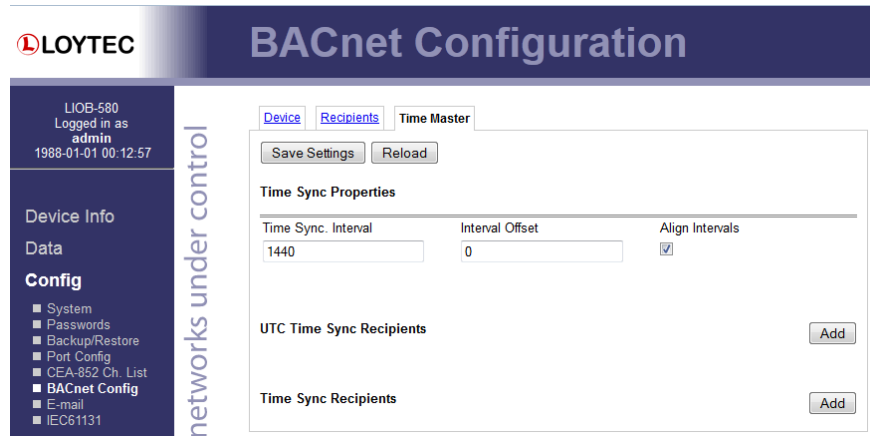


Figure 42: BACnet Time Master Configuration

5.2.12 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 43.

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.

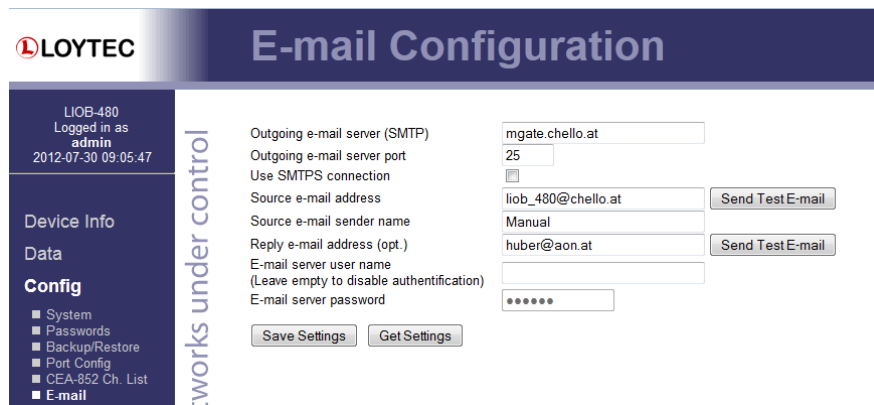


Figure 43: E-mail Configuration Page

5.2.13 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 44. After downloading the IEC61131 program the PLC kernel is restarted with the new 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.

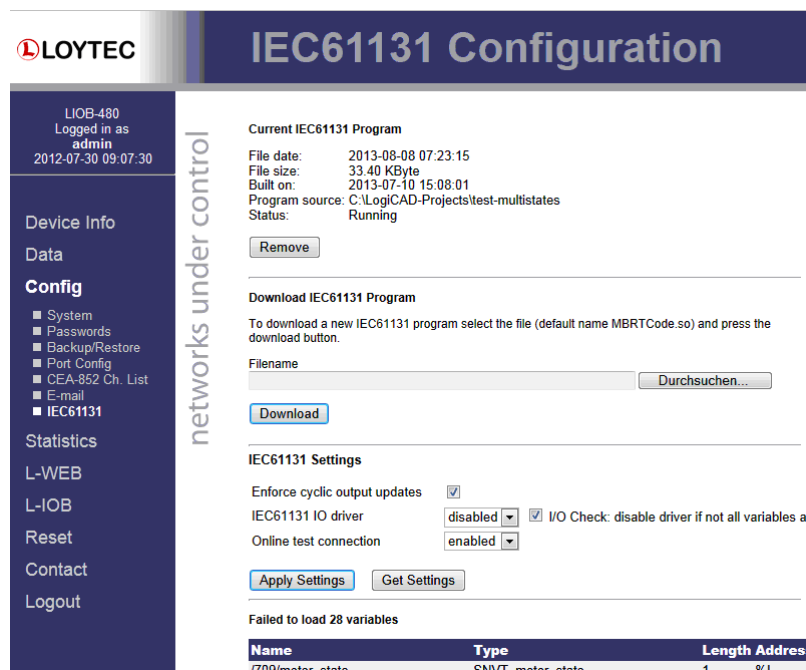


Figure 44: IEC61131 Configuration Page

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 **Remove** button. The device needs to reboot after this action.

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 9.4.6.

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 45. 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 and all levels below. Thus, one can get an easy overview of all data points.

The data point list displays the data point name, direction, type, current value, and data point state. 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 4. Values can be directly edited in the list where the pencil symbol appears.

The screenshot shows the LOYTEC web interface for 'Data Points'. On the left is a navigation menu with options like 'Device Info', 'Data', 'Config', 'Statistics', 'L-WEB', 'L-IOB', 'Reset', 'Contact', and 'Logout'. The 'Data' section is expanded to show 'Data Points', 'Trend', 'Scheduler', 'Calendar', and 'Alarm'. The main content area is titled 'Data Points' and features a tree view on the left with folders like 'ROOT (ROOT OBJECT)', 'Favorites', 'System Registers', 'User Registers', 'IEC61131 Variables', 'Alarm', 'Trend', 'CEA709 Port', 'Local IO', and 'LIOB-IP'. A 'Reload' button is present above the table. The table, titled 'Folder: /System Registers/', lists various data points with columns for Name, Dir, Type, State, and Value. The values include numerical data, strings, and binary states.

Name	Dir	Type	State	Value
System Time	input	analog	normal	1343632665
CPU Load	input	analog	normal	15.5
Free Memory	input	analog	normal	5015120
Free Flash	input	analog	normal	11348112
Supply Voltage	input	analog	invalid value	--
System Temp	input	analog	invalid value	--
Application Vendor	input	analog	invalid value	--
Authentication Code	output	analog	invalid value	--
Authentication Result	input	binary	normal	FALSE
Serial Number	input	string	normal	021301-8000000140FC
Firmware Version	input	string	normal	4.9.0
TZ Offset	input	analog	normal	7200
Device Status	input	string	normal	<?xml version="1.0" encoding=&qu...
MAC Address	input	user	normal	000AB0001326
Device IP Address	input	string	normal	192.168.12.148
Device IP Port	input	analog	normal	80
Ethernet Link Mask	input	multistate	normal	Eth 1
Hostname	input	string	normal	LIOB-480-8000000140FC

Figure 45: 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.
offline (config)	The data point has a value but it is not reflected on the network due to a configuration error (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.
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 e.g. a data point is not used in the configuration.

Table 4: 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 46. The **Status** field is discussed in Table 4. 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 6.3.2 for a closer discussion on timing parameters. The **Native Info** field displays detailed information on the underlying technology object.

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.

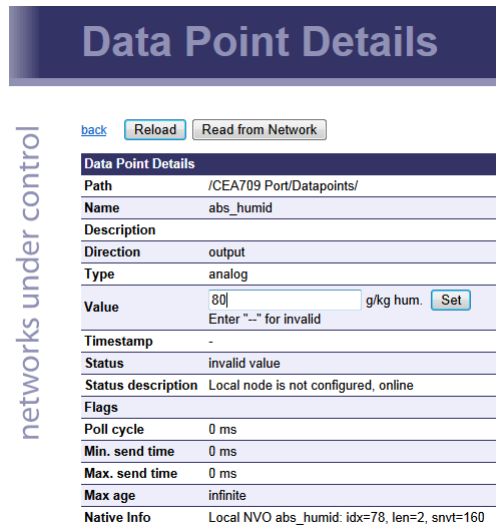


Figure 46: Data point details page

5.3.2 Trend

The Web interface provides a configuration page to re-configure 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). 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 47.

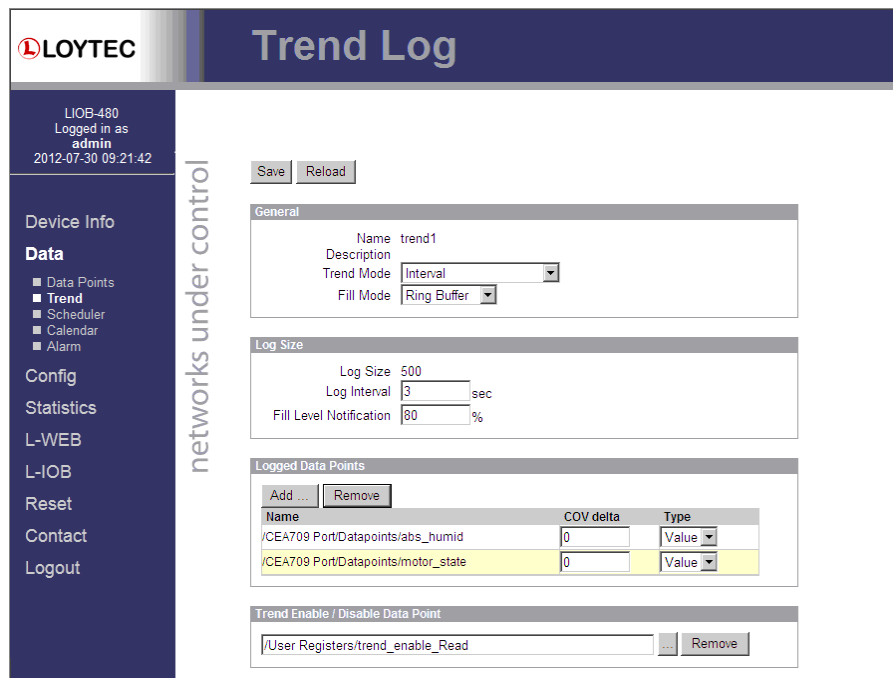


Figure 47: Trend log configuration page

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.

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 48.

The **effective period** defines when this schedule shall be in effect. Leave **From** and **To** at ‘*.*.*’ to make this schedule always in-effect. Otherwise enter dates, such as ‘30.1.2000’. 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 48, 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 49. 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.

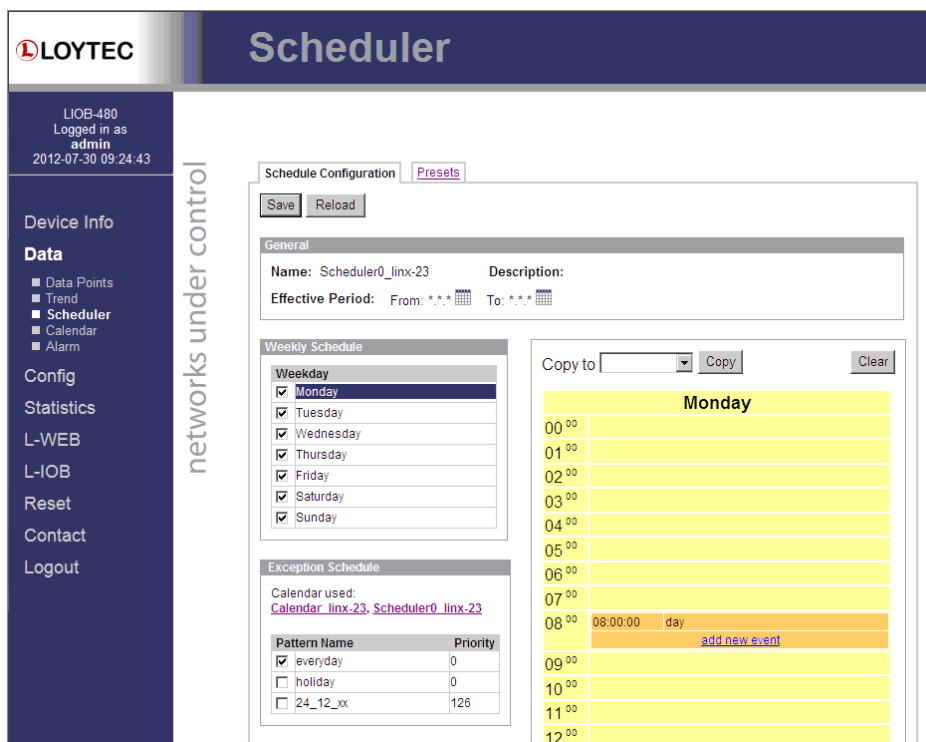


Figure 48: Schedule Configuration Page

Data Point Description	Data Type	default	day
temp1	double	0	20 <input type="button" value="Delete"/>

Figure 49: 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.

On local schedulers 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 50. 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 7.12.4.

Figure 50: 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 51.

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'`.

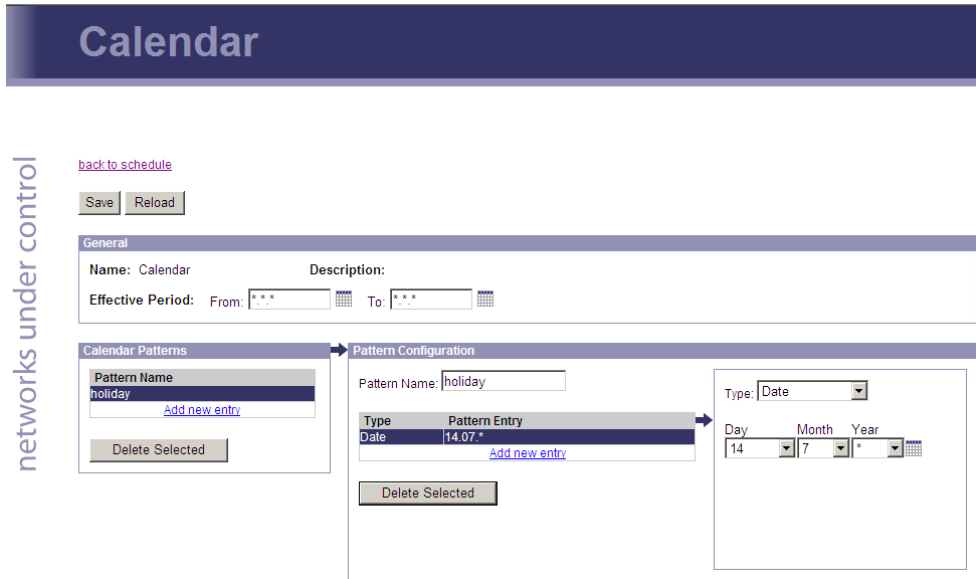


Figure 51: Calendar Configuration Page

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 51, the calendar pattern **Holidays** is selected.

In the **Pattern Configuration** box, the calendar pattern’s name can be edited. It also lists the entries. New entries can be added by clicking **Add new entry**. Existing entries can be selected and edited in the box on the right-hand side. In the example in Figure 51, 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 6.6.3 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 52.

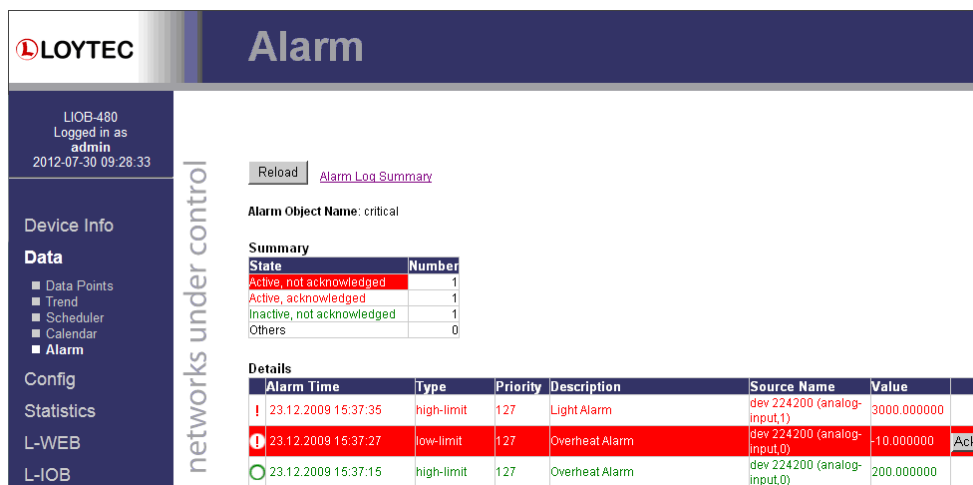


Figure 52: Alarm Summary Page

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 5.4.10 for the alarm log Web interface.

5.4 Device Statistics

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

5.4.1 System Log

The System Log page prints all messages stored in the system log of the device. An example is shown in Figure 53. This log data is important for trouble-shooting. It contains log entries for reboots and abnormal operating conditions. When contacting LOYTEC support, please have a copy of this log ready.

The screenshot shows the LOYTEC System Log interface. The top header includes the LOYTEC logo and the title 'System Log'. Below the header, there are three buttons: 'Clear system log', 'Update system log', and 'Save system log'. The left sidebar contains a navigation menu with the following items: 'LIOB-480 Logged in as admin 2012-07-10 09:33:33', 'Device Info', 'Config', 'Statistics' (with sub-items: System Log, IP, CEA-852, Enh. Comm. Test, CEA-709, OPC Server, Scheduler, Alarm Log), 'L-WEB', 'LIOB', and 'Reset'. The main content area displays a vertical list of log entries, each starting with a timestamp and followed by a message. The messages include system boot information, scheduling mode, boot process completion, and firmware updates. A vertical watermark 'networks under control' is visible on the left side of the log entries.

Figure 53: System Log Page

5.4.2 IP Statistics

Figure 54 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).

LOYTEC Statistics IP

LIOB-480
Logged in as
admin
2012-07-10 09:38:12

Device Info
Config
Statistics

- System Log
- IP
- CEA-852
- Enh. Comm. Test
- CEA-709
- OPC Server
- Scheduler
- Alarm Log

L-WEB
LIOB
Reset
Contact
Logout

networks under control

Update IP statistics

Ethernet Interface Statistics	
Rx packets	0
Rx bytes	3582049422
Rx dropped	0
Rx no clusters	0
Rx CRC errors	0
Rx missed errors	0
Tx collisions	0
Tx resends	0
Tx packets	0
Tx bytes	28711574
Rx no mbuf	0
Rx oversize errors	0
Rx runt errors	0
Tx ok	0
Tx dropped	0

TCP/IP Statistics	
Total IP packets received	17867496
Total IP packets generated here	353088
Total TCP packets sent	348666
Total TCP data packets sent	191816
Total TCP data bytes sent	21122716
Total TCP packets received	311858
Total UDP packets received	17554279
Total UDP packets sent	4393

Update IP statistics

Figure 54: IP Statistics Page

5.4.3 CEA-852 Statistics

The CEA-852 statistics page displays the statistics data of the CEA-852 unit, the global connections, and the LIOB-IP channel on the device (restrictions apply depending on L-IOB model). The upper part of the CEA-852 statistics page is depicted in Figure 55. 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.

The screenshot displays the 'CEA-852 Statistics' page. On the left, there is a navigation sidebar with options like 'Device Info', 'Config', 'Statistics', 'L-WEB', 'LIOB', 'Reset', 'Contact', and 'Logout'. The 'Statistics' section is expanded, showing a list of items including 'System Log', 'IP', 'CEA-852', 'Enh. Comm. Test', 'CEA-709', 'OPC Server', 'Scheduler', and 'Alarm Log'. The main content area is titled 'networks under control' and contains two tables of statistics for 'CEA-709 over IP (CEA-852)'. The first table, 'CEA-852 Device Statistics', lists various metrics such as 'Seconds since cleared', 'Date/Time of clear (GMT)', 'No. of registered members', and various packet counts. The second table, 'Extended CEA-852 Device Statistics', lists 'Session ID', 'SNTP synchronized', 'Number of CR member infos', 'Current channel routing mode', and 'Message alloc count'.

CEA-852 Device Statistics	
Seconds since cleared	773598082
Date/Time of clear (GMT)	Fri Jan 1 00:00:17 1988
No. of registered members	4
LT Packets received	0
LT Bytes received	unknown
LT Packets sent	4
LT Bytes sent	unknown
IP Packets sent	0
IP Bytes sent	0
IP Packets received	4
IP Bytes received	168
IP Packets data sent	0
IP Packets data received	0
LT Stale packets	0
RFC Packets sent	153
RFC Packets received	153
Avg. aggregation to IP	1
Avg. aggregation from IP	1
UDP Packets sent	2177
TCP Packets sent	0
Multi-cast Packets sent	0

Extended CEA-852 Device Statistics	
Session ID	0x47156f1a
SNTP synchronized	no
Number of CR member infos	3
Current channel routing mode	CR
Message alloc count	0

Figure 55: Part of the CEA-852 Statistics Page

5.4.4 Enhanced Communications Test

The Enhanced Communications Test allows testing the CEA-852 communication path between the CEA-852 unit (resp. global connection unit, LIOB-IP channel) on the L-IOB device 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 56.

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 5.

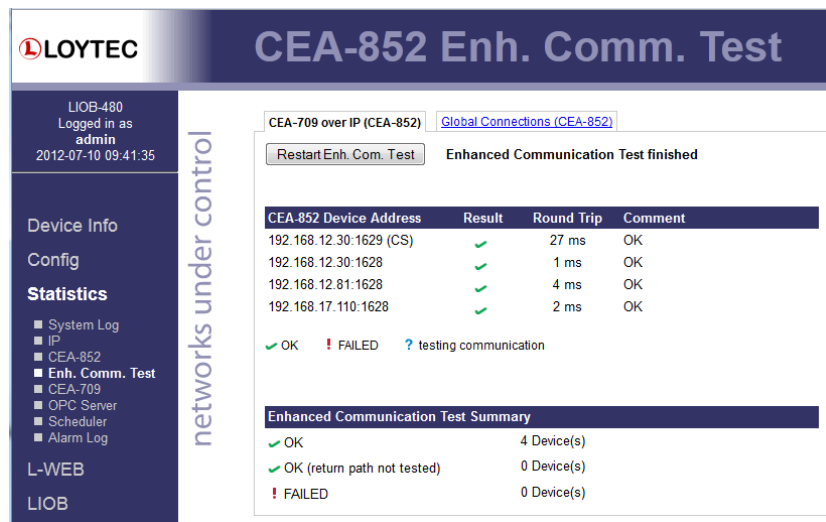


Figure 56: Enhanced Communication Test Output

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 5: Possible Communication Problems

5.4.5 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 57. 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.

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.

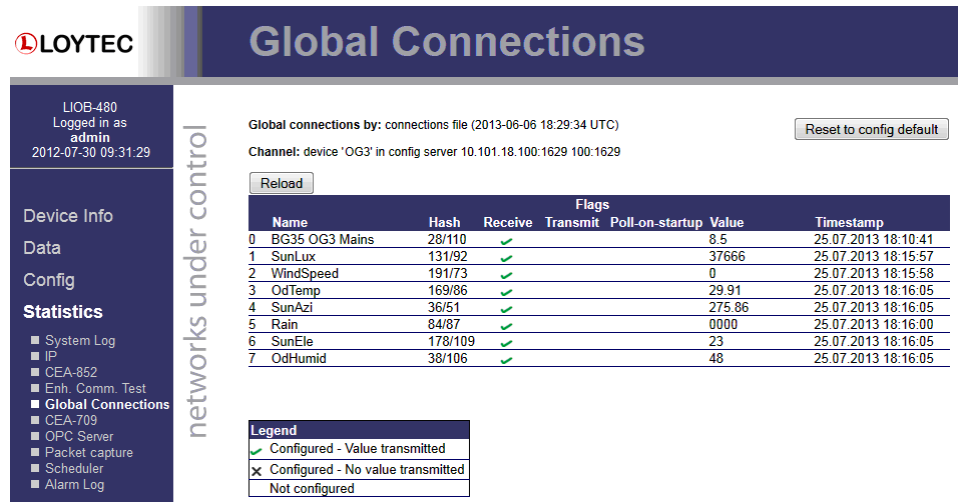


Figure 57: Global Connections Statistics

5.4.6 CEA-709 Statistics (LIOB-48x)

The CEA-709 statistics page displays statistics data of the CEA-709 port on the device as shown in Figure 58. This data can be used to troubleshoot networking problems. To update the data, click on the button **Update CEA-709 statistics**.

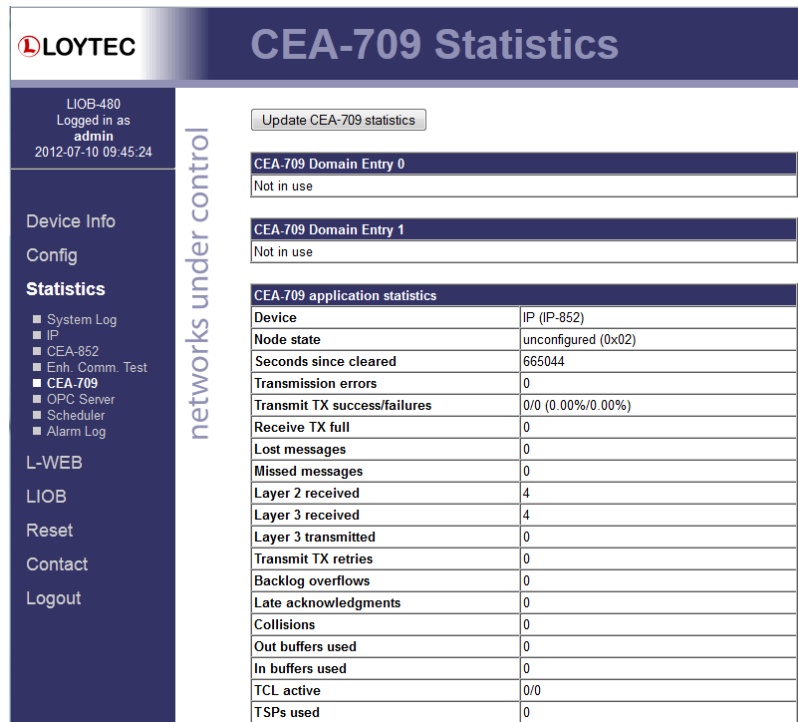


Figure 58: CEA-709 Statistics Page

5.4.7 OPC Server Statistics Page

The OPC 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 59. Clicking on the **Update OPC 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.

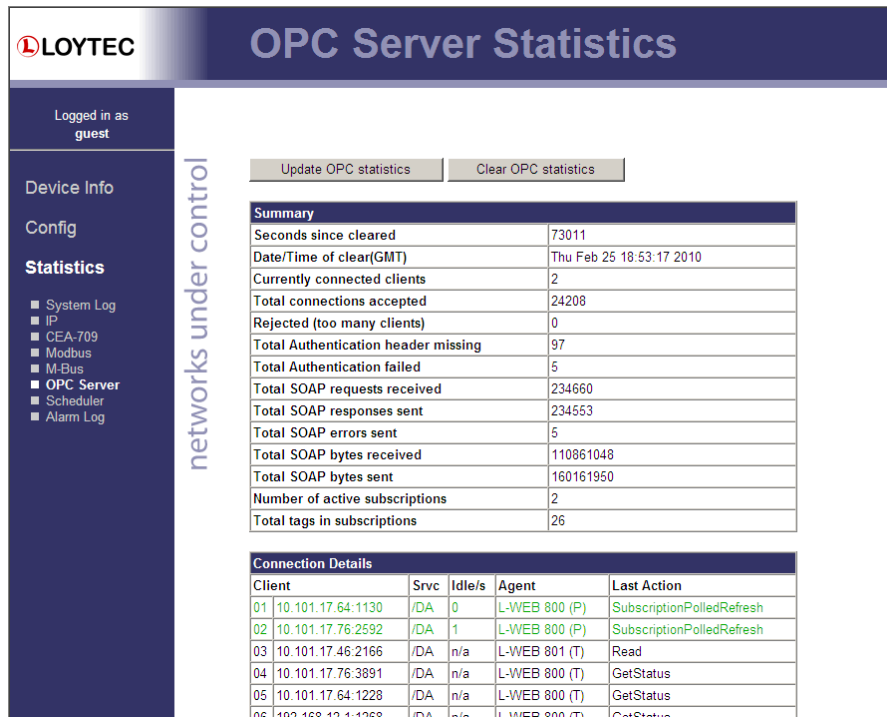


Figure 59: OPC Server Statistics Page

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 **Srv** 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.

5.4.8 BACnet Bindings Statistics (BACnet Mode)

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 60 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.

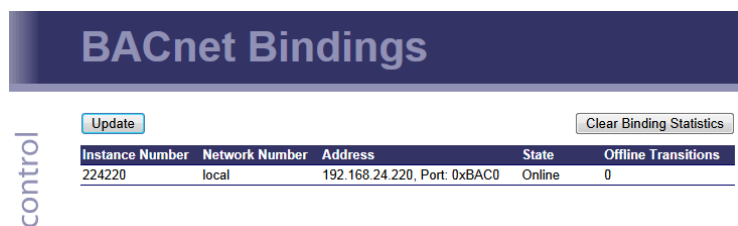


Figure 60: BACnet bindings statistics page

5.4.9 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 61.

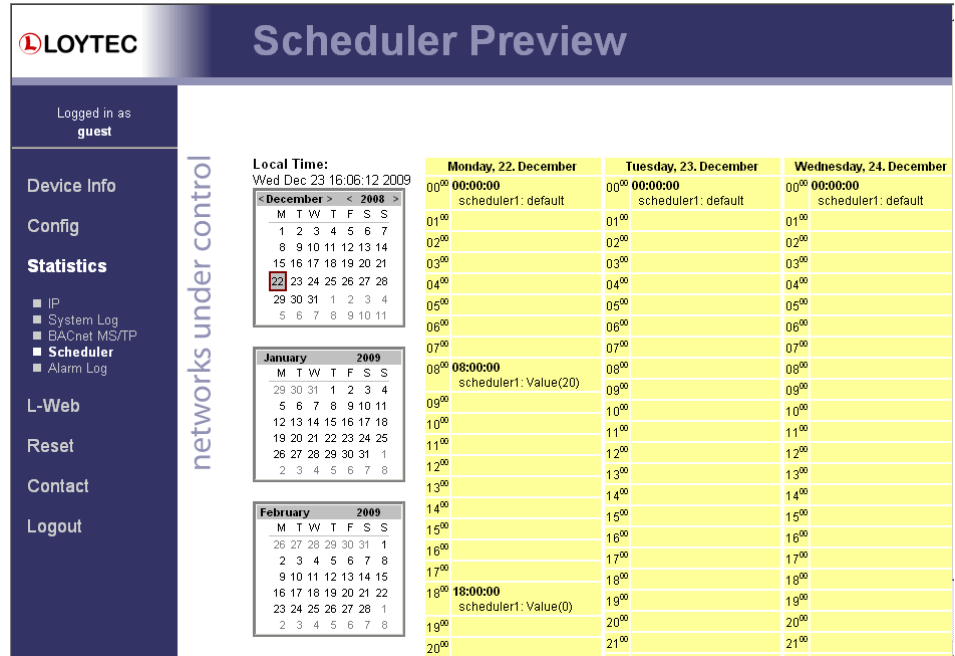


Figure 61: Scheduler Statistics Page

5.4.10 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 62.

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).

Alarm Log																																																																																																																								
<div style="display: flex; justify-content: space-between;"> Reload Upload Alarm Log Clear Alarm Log </div> <p>Alarm log name: Alarm Log Attached alarm objects: <i>critical</i></p> <table border="1"> <thead> <tr> <th>Event Time</th> <th>State</th> <th>Type</th> <th>Priority</th> <th>Description</th> <th>Source Name</th> <th>Value</th> <th>Ack Source</th> <th>Alarm Time</th> <th>Clear Time</th> <th>Ack Time</th> </tr> </thead> <tbody> <tr> <td>23.12.2009 15:43:07</td> <td>acknowledged</td> <td>high-limit</td> <td>127</td> <td>Overheat Alarm</td> <td>dev 224200 (analog-input,0)</td> <td>200.000000</td> <td>admin@new:LINX-200</td> <td>23.12.2009 15:37:15</td> <td>23.12.2009 15:37:18</td> <td>23.12.2009 15:43:07</td> </tr> <tr> <td>23.12.2009 15:43:07</td> <td>acknowledged</td> <td>low-limit</td> <td>127</td> <td>Overheat Alarm</td> <td>dev 224200 (analog-input,0)</td> <td>-10.000000</td> <td>admin@new:LINX-200</td> <td>23.12.2009 15:37:27</td> <td>23.12.2009 15:43:02</td> <td>23.12.2009 15:43:07</td> </tr> <tr> <td>23.12.2009 15:43:02</td> <td>acknowledge pending</td> <td>low-limit</td> <td>127</td> <td>Overheat Alarm</td> <td>dev 224200 (analog-input,0)</td> <td>-10.000000</td> <td></td> <td>23.12.2009 15:37:27</td> <td>23.12.2009 15:43:02</td> <td></td> </tr> <tr> <td>23.12.2009 15:42:53</td> <td>acknowledged</td> <td>high-limit</td> <td>127</td> <td>Light Alarm</td> <td>dev 224200 (analog-input,1)</td> <td>3000.000000</td> <td>admin@new:LINX-200</td> <td>23.12.2009 15:37:35</td> <td>23.12.2009 15:42:53</td> <td>23.12.2009 15:38:04</td> </tr> <tr> <td>23.12.2009 15:38:04</td> <td>acknowledged active</td> <td>high-limit</td> <td>127</td> <td>Light Alarm</td> <td>dev 224200 (analog-input,1)</td> <td>3000.000000</td> <td>admin@new:LINX-200</td> <td>23.12.2009 15:37:35</td> <td></td> <td>23.12.2009 15:38:04</td> </tr> <tr> <td>23.12.2009 15:37:35</td> <td>active</td> <td>high-limit</td> <td>127</td> <td>Light Alarm</td> <td>dev 224200 (analog-input,1)</td> <td>3000.000000</td> <td></td> <td>23.12.2009 15:37:35</td> <td></td> <td></td> </tr> <tr> <td>23.12.2009 15:37:27</td> <td>active</td> <td>low-limit</td> <td>127</td> <td>Overheat Alarm</td> <td>dev 224200 (analog-input,0)</td> <td>-10.000000</td> <td></td> <td>23.12.2009 15:37:27</td> <td></td> <td></td> </tr> <tr> <td>23.12.2009 15:37:18</td> <td>acknowledge pending</td> <td>high-limit</td> <td>127</td> <td>Overheat Alarm</td> <td>dev 224200 (analog-input,0)</td> <td>200.000000</td> <td></td> <td>23.12.2009 15:37:15</td> <td>23.12.2009 15:37:18</td> <td></td> </tr> <tr> <td>23.12.2009 15:37:15</td> <td>active</td> <td>high-limit</td> <td>127</td> <td>Overheat Alarm</td> <td>dev 224200 (analog-input,0)</td> <td>200.000000</td> <td></td> <td>23.12.2009 15:37:15</td> <td></td> <td></td> </tr> </tbody> </table>											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		
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 62: 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.

5.4.11 Packet Capture

The packet capture feature allows configuring and running a remote packet capture for the Ethernet port. Please refer to Section 12.2 for more information on how to configure remote packet capture with Wireshark.

5.5 L-WEB

5.5.1 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 63). Clicking on **Install** will download the installer for L-WEB and start the installation process. See Section 8.2 for more information on working with the L-WEB visualization.

L-WEB Project List

LIOB-480
Logged in as
admin
2012-07-10 09:49:19

Device Info

Config

Statistics

L-WEB

- Project List
- LWEB-802 Config
- ACL Configuration

Install the LOYTEC LWEB-800 Visualization software on your PC

Available L-WEB projects

Name	Last modified	Size (Bytes)	
project2.lweb	29.06.2012 15:31:29	7331	
project1.lweb2	29.06.2012 15:41:04	7230	

Figure 63: L-WEB Project List Page

5.5.2 LWEB-802 Configuration

Here the user can setup the location of the LWEB-802 application as shown in Figure 64. Refer to the LWEB-802 manual for more information.

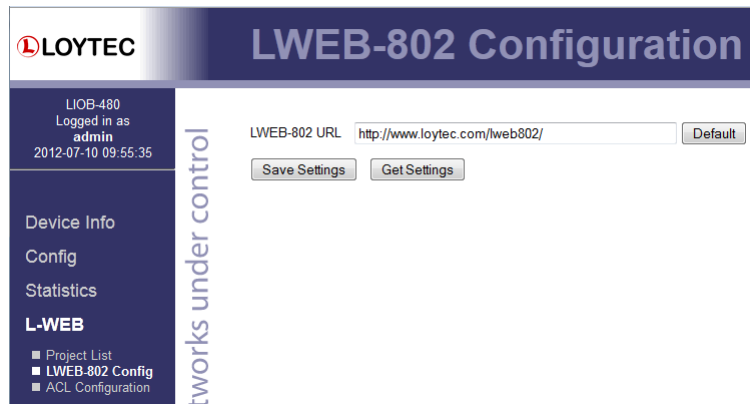


Figure 64: LWEB-802 Configuration Page

5.5.3 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 65.



Figure 65: 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 65 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".

5.6 L-IOB Host and Local I/Os

A LIOB-48x/58x can act as a L-IOB Host for a LIOB/45x/55x device, providing a Web interface to configure, operate, and test the connected L-IOB device. It also provides statistics information about the connected L-IOB device. All L-IOB I/O controllers are additionally equipped with local I/Os, which are also covered by this section.

5.6.1 LIOB-IP Bus

To be able to connect a LIOB-45x/55x device over the LIOB-IP bus, the firmware version of the LIOB-48x/58x must be 4.8 or higher. After a firmware upgrade, the LIOB-IP support must be enabled in the Web-UI (menu "L-IOB / Upgrade"). The LIOB-IP bus will then be enabled by default, as shown in Figure 66. It is possible to disable the LIOB-IP bus entirely by un-checking the **LIOB-IP** checkbox. The corresponding UDP/TCP ports 16028 and 16029 will then not be open anymore. Observe that the LIOB-IP bus only acts as a virtual medium, connecting the LIOB-45x/55x device over Ethernet/IP. The rest of the configuration must be done as described in Section 5.6.2.

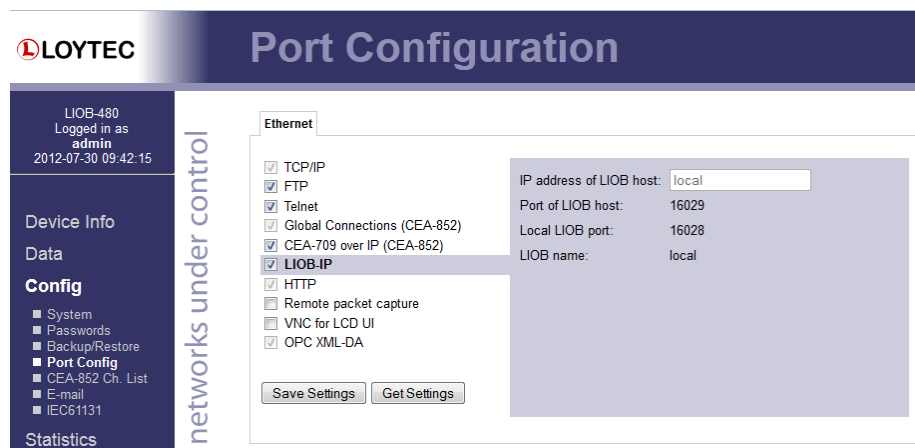


Figure 66: LIOB-IP Bus

5.6.2 LIOB-IP Installation Page

The **LIOB Installation** page for the LIOB-IP bus is shown in Figure 67.



Figure 67: LIOB-IP Device Installation and Scan

Observe that the Station ID must be set to 1 in the LCD UI of the connected LIOB-45x/55x device. Should the connected device not be online, try scanning (**Scan**) the LIOB-IP bus and saving the LIOB-IP configuration again (**Save Settings**).

5.6.3 L-IOB Overview Page

The **LIOB Overview** page provides a quick overview of all L-IOB devices and their I/Os. This page is available for the local I/Os as well as the connected LIOB-45x/55x device. Figure 68 shows the Overview page for the local I/Os. With **Reset All Count Values**, the counter inputs as well as the run hours and energy count values of all outputs can be reset.

The screenshot shows the LIOB Overview page for a LIOB-580 device. The user is logged in as 'admin' at 2012-12-14 17:09:04. The page is titled 'LIOB Overview' and has a sidebar with navigation options. The main content area shows a table of I/Os with columns for I/O Name, Term, Type, Value, Mode, and Status. The table lists 16 I/Os, including temperature sensors (UI), digital inputs (DI), and analog/digital outputs (AO, DO). The 'Value' column shows live values, and the 'Mode' column shows the operating mode (Auto, Manual, Override, Disabled). A 'Reset All Count Values' button is visible above the table.

I/O Name	Term	Type	Value	Mode	Status
UI_Custom_NTC	UI1	IN Analog/Digital 10V	25.30 °C	Auto	OK
UI_PT1000	UI2	IN Analog/Digital 10V	-1.12 °C	Auto	OK
UI_NTC10K	UI3	IN Analog/Digital 10V	20.22 °C	Auto	OK
UI_NTC1K8	UI4	IN Analog/Digital 10V	25.23 °C	Auto	OK
UI_Ni1000	UI5	IN Analog/Digital 10V	-2.03 °C	Auto	OK
UI_Linear	UI6	IN Analog/Digital 10V	28.85 °C	Auto	OK
UI_Frequency	UI7	IN Analog/Digital 10V	0.00 Hz	Auto	OK
UI_Trans_Table	UI8	IN Analog/Digital 10V	39.96 °C	Auto	OK
DI_Digital	DI1	IN Digital	OPEN	Auto	OK
DI_Pulse_Cnt	DI2	IN Digital	0.00	Auto	OK
AO_Digital	AO1	OUT Analog 12V	OFF	Auto	OK
AO_Analog	AO2	OUT Analog 12V	0.00 V	Auto	OK
DO_Digital	DO2	OUT Relais 6A	OPEN	Auto	OK
DO_PWM	DO3	OUT Relais 6A	0.00 %	Auto	OK
manual	DO4	OUT Relais 6A	OPEN	Manual	OK
override	DO5	OUT Relais 6A	OPEN	Override	OK
disabled	DO6	OUT Triac	OPEN	Disabled	OK
unused	DO7	OUT Triac	OPEN	Auto	OK
unused	DO8	OUT Triac	OPEN	Auto	OK

Figure 68: LIOB Overview

The live value and a few properties are shown for each I/O of the L-IOB device. The operating mode of each I/O can be changed in the **Mode** column. The **Value** column always shows the corresponding effective value (e.g. the override value in override mode). In case of manual operating mode (see Section 6.7.1.9), the manual value is displayed and can be changed in the **Value** column.

By clicking on the link in the **I/O Name** column, a detailed view of each I/O can be invoked, see Figure 69. Please refer to Section 6.7 for a detailed description of all I/O properties. After changing configuration properties, the user must click **Save Settings** to activate the new configuration in the L-IOB device.

LIOB-480, Terminal UI1	
Name	Temp
Input	33.20 °C
ManualValue	33.20 °C
HardwareType	IN Analog/Digital 10V
SignalType	Resistance
Interpretation	NTC10K
OperatingMode	Manual
LinkCfgVers	1.00
IOStatus	OK
SIUnit_OnText	°C
USUnit_OffText	°F
Resolution	0.10 °C
MultUS	1.80
OffsUS	32.00
OverrideValue	0.00 °C
DefaultValue	0.00 °C
Offset	0.00 °C
MinValue	-273.15 °C
MaxValue	100.00 °C
COV	0.00 °C
MinSendTime	0.00 s

Figure 69: L-IOB I/O Details

5.6.4 L-IOB I/O Test Page

The **LIOB I/O Test** page provides the possibility of documenting tests of the connected actuators and sensors. This page is available for the local I/Os as well as the connected LIOB-45x/55x device. Figure 70 shows the L-IOB I/O test page of the local I/Os.

I/O Name	Terminal	Test Result	Test Date	Test Comment
LOCAL I/O				
UI_Custom_NTC	UI1	Not Tested	2012-12-13 22:43:15	No sensor available yet
UI_PT1000	UI2	Not Tested	2012-12-13 22:43:19	No sensor available yet
UI_NTC10K	UI3	OK	2012-12-13 22:43:22	Was tested with actual sensor
UI_NTC1K8	UI4	OK	2012-12-13 22:43:26	Was tested with actual sensor
UI_Ni1000	UI5	OK	2012-12-13 22:43:28	Was tested with actual sensor
UI_Linear	UI6	Not Tested	2012-12-13 22:43:31	No sensor available yet
UI_Frequency	UI7	NOT OK	2012-12-13 22:43:35	Sensor defect

Figure 70: L-IOB I/O Test

For each I/O, a **Test Result** (Not Tested, OK, NOT OK) can be chosen. The **Test Date** will then be set automatically but can be manually changed afterwards. An additional **Test Comment** can be entered. This data will be stored persistently in the LIOB-48x/58x until it is explicitly cleared or the type of the connected L-IOB device changes because of a new host configuration. To explicitly clear the test information, the buttons **Clear Tests** or **Clear All Tests** are used. The test information can also be exported to a CSV file by using **Export Device (CSV)** or **Export All (CSV)**.

5.7 Reset, Contact, Logout

The menu item **Reset** allows three essential operations:

- Rebooting the device from a remote location.
- Clearing 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.
- Resetting all parameters from a remote location. This option leaves the base configuration intact but deletes all parameter values written by LWEB-821/900.

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

This chapter describes the basic concepts behind the installation, configuration, and data points of LIOB-18x/48x/58x devices. Observe that not all properties and data points will be available in all configuration instances. Some properties e.g. can only be seen or setup at configuration time, others only at run time.

6.1 Device Installation

For LIOB-58x devices, the first step after hardware installation and IP configuration is to configure the BACnet interface, see Sections 5.2.8 and 5.2.9.

For LIOB-48x devices, the first step after hardware installation and IP configuration is to add them to a CEA-852 channel, see Section 4.1 or Section 5.2.6.

The I/Os of LIOB-48x and LIOB-58x controllers can be extended with one LIOB-45x or LIOB-55x device (in L-INX mode) by using the LIOB-IP bus of the controller. Please refer to the LIOB-10x/x5x User Manual [2] for how to connect a LIOB-45x/55x device to a L-IOB host (in this case the LIOB-48x/58x controller).

Both the LIOB-18x and LIOB-48x devices must be installed and commissioned like any other CEA-709 / LONMARK® node. Please consult the documentation of your CEA-709 network management tool on how to perform these tasks.

For LNS™ based network management tools, the LOYTEC L-INX Configurator Software acts as an LNS™ plug-in to configure the LIOB-18x/48x devices. It also installs the needed template for the LIOB-18x/48x models when it is registered as a plug-in in the network management software. Both off- and online installation of the L-IOB devices is supported.

For Non- LNS™ network management tools, a LOYTEC NIC such as the NIC-USB100 or NIC852 is needed for configuration of LIOB-18x/48x devices. The devices must be configured (using the CEA-709 connection method of the Configurator Software) before installing and commissioning them in the network management tool.

6.2 LONMARK® Device Mode (LIOB-18x/48x)

LIOB-18x/48x I/O Controllers in a certain configuration are LONMARK® certified. To setup this special configuration, the LONMARK® device mode must be setup in the LCD UI, see Section 4.1. The LIOB-18x/48x device will reboot and from now on behave exactly like the corresponding LIOB-15x/45x I/O module in LONMARK® device mode. If e.g. a LIOB-180 device is switched to LONMARK® device mode, it will behave like a LIOB-150 I/O module (in LONMARK® device mode). To switch the device back, go to the device information and configuration page (toothed wheel icon) in the L-IOB LCD UI and switch

the device mode back to 'I/O Controller'. Please refer to the LIOB-10x/x5x User Manual [2] for more information about the LIOB I/O modules in LONMARK® device mode.

6.3 Data Points

6.3.1 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.

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, 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.

6.3.2 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:

- **Polcycle** (input): 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): This is a variation on the poll cycle. When receive timeout is enabled, the data point actively polls the source unless it receives an update. For example, if poll cycle is set to 10 seconds and an update is received every 5 seconds, no extra polls are sent. The receive timeout setting cannot be used for feedback data points.

- **Poll-on-startup** (input): 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 an option that some network technologies offer. With static polling the poll cycle is used to permanently poll values over the network. This is required for data points that require constant value updates a fixed poll cycle (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 L-WEB). When the data is not 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.

6.3.3 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.

6.3.4 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.

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.

6.3.5 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. In addition, some I/O configuration properties are implicitly classified as parameters. 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-821/900 master parameter editor 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-821/900 Master Parameter Editor manual.

When changing parameters on the device or via the master parameter editor, they are out of sync with their default values in the configuration. The Configurator provides a mechanism to resolve this conflict and to merge those changed parameters back into the configuration. This is accomplished in the parameter merge dialog when uploading or downloading the configuration (see Figure 71). 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.

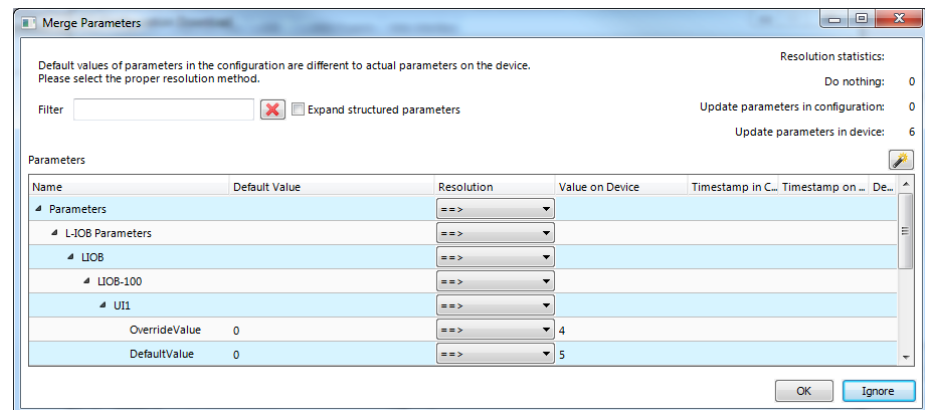


Figure 71: Parameter merge dialog

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.

6.3.6 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 zero, all updates are considered.

6.3.7 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.

6.3.8 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 (LIOB-48x/58x only). 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:

- **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.
- **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 9.6.6 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".
- **Firmware Version:** This register is a *string* data point. It displays the device's firmware version as an ASCII string. Example: "4.1.0".
- **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.
- **MAC Address (LIOB-48x/58x):** This register is a *user* data point. It displays the device's MAC address as an array of 6 hexadecimal Bytes. Example: 000AB001D1E4.
- **Device IP Address (LIOB-48x/58x):** 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 (LIOB-48x/58x):** This register is an *analog* data point. It displays device's HTTP port as an integer value. Example: 80.
- **Ethernet Link Mask (LIOB-48x/58x):** This register is a *multistate* data point. It displays the link information of the Ethernet port. Example: "Eth 1".
- **Hostname (LIOB-48x/58x):** This register is a *string* data point. It displays the host name, which has been configured in the IP settings. Example: "my_liob".

6.3.9 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, two data points are generated for each register, one for writing the register (output) and one for reading the register (input). 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*.

6.3.10 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 72. 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*.


example2_Read	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	In
byte_0	1.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	In
byte_1	1.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	In

Figure 72: 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.

6.3.11 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 73. They are marked with a link symbol . When hovering with the mouse over the link symbol, a bubble help appears describing the property relation.





reg_bool_fb_alarm_Read	7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In	
feedbackValue -> User Registers.master_feedback_Read	7.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In	
enableAlarm	7.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Value	
inAlarm	7.3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Value	

Figure 73: 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 73. 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 7.2.7.

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. CEA-709). 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.
- **historicFilter:** This property relation exists for data points that have at least one historic filter assigned (see Section 6.6.6).

6.3.12 Convertible Engineering Units

Analog data points possess the engineering units property that defines the physical unit of the underlying scalar value, e.g., "kg". The engineering unit is displayed with 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 those units the Configurator can offer compatible units of the respective other unit system, of different magnitude or different usage. Important properties of convertible units when used are:

- An alternative display unit can be configured that will be displayed on the Web UI of the device.
- 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.

6.4 Math Objects (LIOB-48x/58x)

6.4.1 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, \dots, 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: $(v1 + v2) * \text{sqrt}(\text{pow}(v3, 0.1))$

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.

6.4.2 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 $v1=5, v2=\text{invalid}, v3=3$ 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.

6.4.3 Function List

The currently supported math function calls are listed in Table 6.

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

Function	Return Value
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 by $\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$. It is defined for every real number except for no positive integers. For nonnegative integer 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 6: Available math functions

6.5 Connections

6.5.1 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.

6.5.2 Multi-Slot Connections and Automatic Generation

Although the L-IOB I/O Controllers are capable of multi-slot connections and auto-generate, these features will usually not be required in their applications. For more information on multi-slot connections and auto-generate, please refer to the L-INX User Manual [3].

6.5.3 Global Connections (LIOB-48x/58x)

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.

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 Sections 5.2.6 and 5.2.7.

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 6.3.2). 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 6.3.2), 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 7.10.5. Note, that the number of configurable global connections on a device is limited (see Section 15.6).

6.6 AST Features

The LIOB-18x models are only capable of (local) alarming and scheduling, while the LIOB-48x/58x models feature the complete AST functionality, as described in the following sections. For getting remote access to the local alarms, schedules, and calendars on a LIOB-18x device, a LOYTEC device capable of remote AST access (e.g. L-INX) is needed, since the LIOB-18x models do not have a Web UI (no Ethernet interface). Refer e.g. to the L-INX user manual [3] for information on remote AST access.

6.6.1 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 6.3.11). 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 CEA-709), 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.

6.6.2 Historical Alarm Log (LIOB-48x/58x)

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.

6.6.3 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 }).

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 74 for an example of a schedule in a given calendar week.

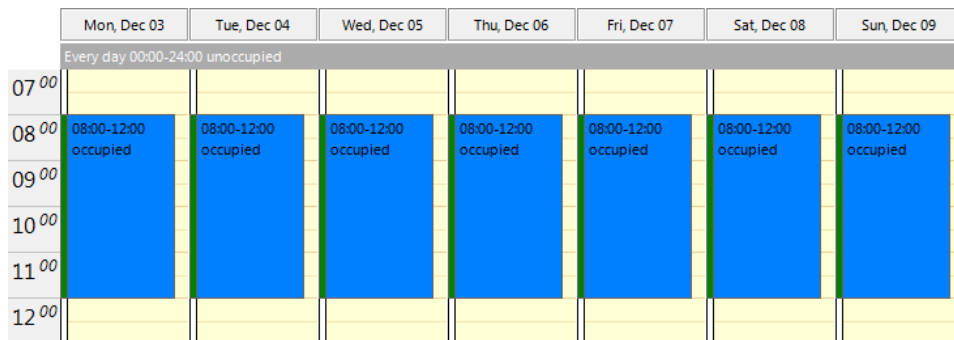


Figure 74: Example of a recurring event in a schedule

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 75. 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 7.10 to learn about special behavior of the respective networking technology.

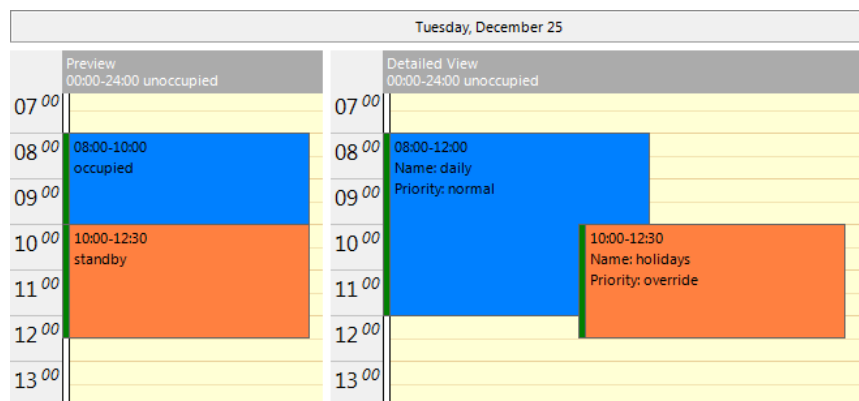


Figure 75: Example with overlapping events and different priorities

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 ahead into the future and predicting when the next scheduled state will occur. There are two data points involved: the time-to-next-state is a counter in minutes to the next scheduled event, and the next-state data point is the state of the next scheduled event. This information can be used by controllers to optimize their algorithms (e.g., pre-heat a room for the scheduled occupancy state). Use the `SNVT_tod_event` in CEA-709 to accomplish this task.

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.

6.6.4 Trending (LIOB-48x/58x)

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.

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.

6.6.5 E-mail (LIOB-48x/58x)

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.

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 5.2.12). 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.

6.6.6 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 relation 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), 0..24 samples ago,
- Yearly value at HH:MM:SS on MM/DD 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 (previous day at midnight). This is shown in Figure 76. Those historic filter relations can be processed in a math object to calculate the consumption of the previous day.

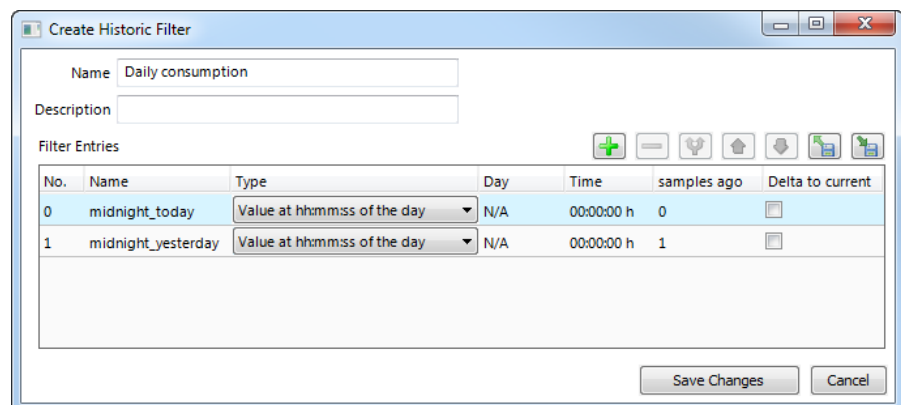


Figure 76: Example historic filters for daily consumption

For calculating the difference between the current value and any historic value, the filter definition can be configured to delta mode. This is a shortcut to creating a math object processing the historic filter data point and the underlying data point. The results of the math object for the previous day and the consumption to-the-hour of the current day are available in data points, which can be visualized, alarmed 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 a 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 7.16.

6.7 I/O Technology

6.7.1 I/O Configuration

Each I/O in a L-IOB device has certain configuration properties which are specific to that I/O. Not all listed properties are available for all I/Os, dependent on the hardware type and certain other configuration properties. The following sections describe the I/O properties and their dependencies.

6.7.1.1 Name

The name property is available for all I/Os. It is the user defined name of the I/O (e.g.: 'Temperature1').

6.7.1.2 HardwareType

The hardware type property is available for all I/Os. The following hardware types are available (dependent on the L-IOB model):

- **IN Analog/Digital:** universal analog/digital input (UI) which can be configured to measure resistance, voltage, or current (with or without internal shunt). Observe that this hardware type is also used for the internal pressure sensor (e.g. 'PRESS' in LIOB-184). In this case, 0V refers to 0P (Pascal) and 10V to 500P.
- **IN Digital:** digital S0 input (DI).
- **OUT Analog:** analog 0-10 V output (AO).
- **OUT Relay 6A:** digital 6 A relay output (DO).
- **OUT Relay 16A:** digital 16 A relay output (DO).
- **OUT Triac:** digital 1 A triac output (DO).

This hardware type property can not be configured of course. Refer to Section 15.1 for a detailed specification of the different I/O hardware types.

6.7.1.3 SignalType

The signal type property is available for I/Os of hardware type "IN Analog/Digital". The following signal types can be configured:

- **Resistance:** measures resistance of about 1 k Ω to 100 k Ω . A value bigger than 500 k Ω is detected as a disconnected sensor.
- **Voltage 0-10V:** measures voltage from 0 to 10 V.
- **Current 4-20mA:** measures current from 4 to 20 mA. A value smaller than 3.5 mA is detected as a disconnected sensor. An external shunt of 249 Ω must be used for correct measurement.
- **Voltage 2-10V:** measures voltage from 2 to 10 V. A value smaller than 1.75 V is detected as a disconnected sensor.
- **Current 4-20mA Shunt:** measures current from 4 to 20 mA. A value smaller than 3.5 mA is detected as a disconnected sensor. No external shunt is required. This setting is only available on some universal inputs which have an internal shunt, see Section 15.3. Observe that changing the signal type to this setting may result in changing the setting

on other universal inputs too. In this case, a Configurator message will inform the user of the changes.

6.7.1.4 Interpretation

The interpretation property is available for all I/Os. Depending on the hardware type and signal type, the following interpretations can be configured:

- **CustomNTC**: This interpretation is only available for universal inputs (hardware type “IN Analog/Digital”) with signal type “Resistance”. It is used for connecting a custom NTC temperature sensor to the input. The parameters of the NTC can be setup as described in Section 6.7.1.24.
- **PT1000**: This interpretation is only available for universal inputs (hardware type “IN Analog/Digital”) with signal type “Resistance”. It is used for connecting a PT1000 temperature sensor to the input.
- **NTC10K**: This interpretation is only available for universal inputs (hardware type “IN Analog/Digital”) with signal type “Resistance”. It is used for connecting an NTC10K temperature sensor to the input.
- **NTC1K8**: This interpretation is only available for universal inputs (hardware type “IN Analog/Digital”) with signal type “Resistance”. It is used for connecting an NTC1K8 temperature sensor to the input.
- **Ni1000**: This interpretation is only available for universal inputs (hardware type “IN Analog/Digital”) with signal type “Resistance”. It is used for connecting an Ni1000 temperature sensor to the input.
- **Linear**: This interpretation is only available for universal inputs (hardware type “IN Analog/Digital”). It is used to perform a linear transformation from a physical input value (resistance, voltage, or current, see Section 6.7.1.3) to the actual value, the sensor is supposed to measure (e.g. temperature, see Section 6.7.1.5). The input range is specified by the signal type:
 - Resistance: 0 ... 10 k Ω
 - Voltage 0-10V: 0 ... 10 V
 - Current 4-20mA: 4 ... 20 mA
 - Voltage 2-10V: 2 ... 10 V
 - Current 4-20mA Shunt: 4 ... 20 mA

The output range is specified by MinValue and MaxValue, see Section 6.7.1.20. In case of signal type “Voltage 2-10V” e.g. a measured value of 2 V would be transformed into MinValue and a measured value of 10 V would be transformed into MaxValue.

- **Frequency**: This interpretation is available for all inputs. It is used to measure the frequency of the digital input resp. universal input in digital mode. The period used for measurement is specified by the MinSendTime parameter, see Section 6.7.1.22.
- **Translation Table**: This interpretation is only available for universal inputs (hardware type “IN Analog/Digital”). It is used to perform a transformation from a physical input value (resistance, voltage, or current, see Section 6.7.1.3) to the actual value, the sensor is supposed to measure (e.g. temperature, see Section 6.7.1.5) using a transformation table. The table can be selected with TransTable, see Section 6.7.1.23. The tables can be setup for each L-IOB device as explained at the end of Section 7.17.2.

- **Frequency Table:** This interpretation is available for all inputs. It is used to perform a transformation from a frequency value (measured as described above) to the actual value, the sensor is supposed to measure (e.g. velocity, see Section 6.7.1.5) using a transformation table. The table can be selected with TransTable, see Section 6.7.1.23. The tables can be setup for each L-IOB device as explained at the end of Section 7.17.2.
- **Physical Unit Count:** This interpretation is available for all inputs. It is used to count in a certain physical unit. The unit is setup with DataType and SIUnit_OnText, see Sections 6.7.1.5 and 6.7.1.15. The increment for each pulse is setup using the Resolution property, see Section 6.7.1.16.
- **Digital:** This interpretation is available for all I/Os. In case of an analog output, the off and on output values are specified with OffValue and OnValue, see Section 6.7.1.26.
- **Pulse Count:** This interpretation is available for all inputs. It is used to count pulses of the digital input or universal input in digital mode. It is also used for connecting the code signal of STId card readers, see Section 6.7.2. Like in the “Physical Unit Count” interpretation, a unit and an increment can be setup for calculating a value of a certain physical unit. However, in case of the “Pulse Count” interpretation, this calculation is only used for display on the L-IOB LCD UI. The data point remains a 32-bit counter.
- **Occupancy:** This interpretation is available for all inputs. See Section 6.7.1.25 for details.
- **Clock:** This interpretation is available for all interrupt-capable inputs. It is used for connecting the clock signal of STId card readers, see Section 6.7.2.
- **Card Data:** This interpretation is available for all inputs. It is used for connecting the data signal of STId card readers, see Section 6.7.2.
- **Analog:** This interpretation is only available for analog outputs (hardware type “OUT Analog”). It is used to output a voltage between 0 and 12 V. In case of using data type “Percentage” (see Section 6.7.1.5), the output value (in percent) is scaled using the MinValue and MaxValue properties (see Section 6.7.1.20). An output value of 50% would e.g. translate into an actual voltage of just in the middle between MinValue and MaxValue, an output value of 100% would translate into a voltage of MaxValue. Note that an output value of 0% is still always translated into 0V. As soon as the output value is slightly raised above 0% however, the voltage jumps to MinValue.
- **PWM:** This interpretation is available for all outputs. It is used to generate a pulse width modulated output signal. The period is setup with PWMPeriod, see Section 6.7.1.27. In case of an analog output, the off and on output values are specified with OffValue and OnValue, see Section 6.7.1.26. The output value (in percent) is scaled using the MinValue and MaxValue properties (see Section 6.7.1.20). An output value of 50% would e.g. translate into an actual pulse width of just in the middle between MinValue and MaxValue, an output value of 100% would translate into a pulse width of MaxValue. Note that an output value of 0% is still always translated into zero pulse width (always off). As soon as the output value is slightly raised above 0% however, the pulse width jumps to MinValue. This scaling is typically used to correctly control slow actuators like heating valves. If a valve e.g. requires a pre-heating time of 1 min, using a PWM period of 10 min, the MinValue can be set to 10% to compensate the pre-heating time. For further information on optimal control of your actuator in PWM mode, please refer to the corresponding data sheet. When the output value changes during a PWM period, the new value will be applied in the current period if possible. If the value decreases, the physical output will be switched off earlier, or immediately, if the newly defined point in time has already elapsed. If the value increases, the physical output will be switched off later, if it was still switched on when the value changes. Otherwise, the new value will be applied starting with the next period. When a PWM output is in

manual operating mode (see Section 6.7.1.9), a new period is started immediately whenever the manual value is changed.

- **Fading:** This interpretation is available for all analog outputs. It is equal to the “Analog” interpretation except for the behavior on a value change. The output is not immediately set to the new value but instead a slow fading from the current value to the new value is performed. The transition time used for fading from the current value to the new value (fading time) is setup in the property **DeadTime**, see Section 6.7.1.6. Since this time is constant, the transition speed depends on the difference between current and new value.
- **Ramping:** This interpretation is available for all analog outputs. It is equal to the “Fading” interpretation except that the **DeadTime** property (see Section 6.7.1.6) specifies the time it takes to ramp from the minimum value to the maximum value (ramping time). This way, the transition speed between current and new value is always equal, independent of the difference between the two values.

6.7.1.5 DataType

The data type property is available for all inputs with interpretation “Linear”, “Translation Table”, “Frequency Table”, “Physical Unit Count”, “Digital”, “Occupancy”, and “Switch Mode”, as well as analog outputs with interpretation “Analog”, “Fading, or “Ramping”.

6.7.1.6 DeadTime for Fading and Ramping

For interpretation “Fading”, the **DeadTime** property specifies the fading time from the current value to the new value. For interpretation “Ramping”, the **DeadTime** property specifies the ramping time from the minimum to the maximum value (independent of current and new value).

6.7.1.7 IOFunc, GroupNumber, and DeadTime for Interlocked Mode

The IO function, group number, and dead time properties are available for all digital outputs. If **IOFunc** is set to “Interlocked”, the **GroupNumber** property can be used to form different interlocked groups. Further, for each interlocked output, the **DataType** can be chosen between “Switch” and “Duration”. The digital outputs belonging to one interlocked group have the following characteristics:

- **DataType “Switch”:**
 - If “true” is written to the output value, the output is switched on permanently.
 - If “false” is written to the output value, the output is switched off immediately.
- **DataType “Duration”:**
 - If a positive output value is written, it is interpreted as a period (in [ms]) in which the output shall be switched on. After that period, the output is automatically switched off again.
 - If 0 is written to the output value, the output is switched off immediately.
 - If a negative output value is written, the output is switched on permanently.
- Whenever an output is switched on (as described above), it is checked before if another output in the same interlocked group is already switched on. In this case, the other output is switched off immediately. Then, for a period setup with **DeadTime**, all outputs

of the group remain off. After that, finally the new output is switched on (either for a certain period of permanently).

The interlocked mechanism can be used e.g. for sunblind motors where it must be ensured that the up and down motors are never active at the same time.

6.7.1.8 IOFunc, GroupNumber, and SubGroupNumber for Card Reader Mode

The IO function, group number, and sub group number properties are available for all inputs used for connecting the signals of an STId card reader, see Section 6.7.2.

6.7.1.9 OperatingMode, OverrideValue, and DefaultValue

The operating mode property is available for all I/Os. For inputs, the **OperatingMode** property has the following meaning:

- **Disabled:** The physical input is disabled and the data point is set to **DefaultValue**. This can be used to disable unused inputs.
- **Auto:** The input measures the value from the connected sensor.
- **Override:** The physical input is disabled and the data point is set to **OverrideValue**.
- **Manual:** The physical input is disabled and the data point is set by the user on the L-IOB LCD UI (see Chapter 4). This can be used to simulate input values for the logic application.

The **DefaultValue** is also used for inputs, when no sensor value has yet been read from the physical input or a sensor error is detected. For outputs, the **OperatingMode** property has the following meaning:

- **Disabled:** The output value set by the logic application is ignored and the physical output (as well as the feedback value) is set to **DefaultValue**.
- **Auto:** The physical output (and feedback value) is set as requested by the logic application.
- **Override:** The output value set by the logic application is ignored and the physical output (as well as the feedback value) is set to **OverrideValue**. This can be used e.g. for providing a constant supply voltage to a sensor.
- **Manual:** The output value set by the logic application is ignored and the physical output (as well as the feedback value) is set by the user on the L-IOB LCD UI (see Chapter 4). This can be used to test actuators.

6.7.1.10 Persistent Flag

The persistent flag is available for all outputs. It specifies the behavior of the L-IOB outputs after a power cycle. If it is set, the output is set to the last stored value. If it is cleared, the output is set to the DefaultValue (see Section 6.7.1.9).

6.7.1.11 Invert Flag

The invert flag is available for all I/Os in digital/PWM mode or where the input is used to detect resp. count pulses. In the first case, the invert flag is used to invert the input or output. In the second case, the flag specifies whether the pulse shall be detected at the positive or negative edge. Observe that if the SignalType is set to "Resistance" (see Section 6.7.1.3), the Invert flag is set by default. This is because the ON state ("CLOSED") refers to a small resistance value whereas the OFF state ("OPEN") refers to a big resistance value.

6.7.1.12 AnalInvert Flag

The analog invert flag is available for all voltage or current inputs with interpretation “Linear” or “Translation Table” (see Section 6.7.1.4) and for all outputs with interpretation “Analog”, “Fading”, or “Ramping”. It inverts the analog range between the MinValue and the MaxValue. For a 0-10V sensor with linear interpretation e.g., 0V would result in a live value of MaxValue and 10V results in the MinValue.

6.7.1.13 Sqrt Flag

The square root flag is available for all voltage, current, or pressure inputs with interpretation “Linear” or “Translation Table” (see Section 6.7.1.4). It performs a square root operation on the signal range. The formulas for the different sensor signal types are listed below:

- 0-10V Sensor or Pressure Sensor: $Usqrt = \sqrt{U * 10V}$
- 2-10V Sensor: $Usqrt = 2V + \sqrt{(U - 2V) * 8V}$
- 4-20mA Sensor: $Isqrt = 4mA + \sqrt{(I - 4mA) * 16mA}$

Values below the signal range (below 0V, 2V, or 4mA) are not converted. The *Usqrt* or *Isqrt* value is then used instead of the *U* or *I* value for further calculation (linear transformation or translation table). The square root flag can be used e.g. to easily calculate a flow value instead of a differential pressure.

6.7.1.14 PulseTime Flag

The pulse time flag is available for all inputs with interpretation “Pulse Count” or “Physical Unit Count” (see Section 6.7.1.4). It activates the pulse time data point.

6.7.1.15 SIUnit_OnText and USUnit_OffText

The SI unit / on text and US unit / off text properties are available for all I/Os. In case of analog values, the **SIUnit_OnText** property specifies the unit in SI mode, in case of digital values, it specifies the ON text (shown when the digital I/O is active). In case of analog values, the **USUnit_OffText** property specifies the unit in US mode, in case of digital values, it specifies the OFF text (shown when the digital I/O is inactive). Using the Configurator (**Project Settings**), the user can choose between SI and US units.

6.7.1.16 Resolution

The resolution property is available for all I/Os with an analog data type. It specifies the resolution of the value display in the L-IOB LCD UI as well as the step width for the L-IOB jog dial when manually setting an analog value or property of that I/O. For the interpretations “Physical Unit Count” and “Pulse Count”, it also specifies the increment for each pulse, see Section 6.7.1.4.

6.7.1.17 MultUS and OffsUS

The US Multiplier and Offset properties are available for all I/Os with an analog data type and are calculated automatically. Using the Configurator, the user can choose between SI and US units. In case of US units, the I/O values displayed in the L-IOB LCD UI are converted using these properties ($ValueUS = ValueSI * MultUS + OffsUS$). The US unit is setup in the **USUnit_OffText** property, see Section 6.7.1.15. Observe that the properties setup in the Configurator as well as the connected data points will still always carry SI units.

6.7.1.18 DisplayOnSymbol and DisplayOffSymbol

The display on/off symbol properties are available for inputs with Interpretation “Digital” (see Section 6.7.1.4). They are used to specify the display of the two digital states ON and OFF. For both these states, the following symbols can be chosen:

- **OPEN**: open switch symbol
- **CLOSED**: closed switch symbol
- **OK**: check symbol
- **ERROR**: exclamation mark symbol

6.7.1.19 Offset

The offset property is available for inputs with an analog data type. It is added at the end of the calculation to the (already processed) value. This way, e.g. temperature sensors can be calibrated in each room without changing the common application.

6.7.1.20 MinValue and MaxValue

The minimum and maximum value properties are available for all I/Os with an analog data type (except for Interpretation “Physical Unit Count”) as well as for inputs with interpretation “Digital”, “Occupancy”, or “Switch Mode”. They are used for:

- scaling inputs with interpretation “Linear” (see Section 6.7.1.4),
- scaling outputs with interpretation “Analog”, “PWM”, “Fading”, or “Ramping” (see Section 6.7.1.4),
- scaling the bar icon in the L-IOB LCD UI (see Section 4.1),

6.7.1.21 COV and MaxSendTime

The **Change-Of-Value** property is available for all inputs with an analog data type. It is used to specify a minimum delta value required to trigger a data point update. If the input value only changes within a range of $-\text{COV}$ to $+\text{COV}$, no update is generated. In any case, after **MaxSendTime** an update is generated to be sure to get the latest value at least at some point in time. If COV is set to 0, every input value change generates an update of the attached data point.

Observe that the **MaxSendTime** property is available for all I/Os (analog and digital) to act as a heartbeat function. For outputs, it is applied both on the output value (heartbeat to external L-IOB device) and feedback value (heartbeat from L-IOB device or local I/O). Whenever an update is sent from an I/O because of a max send time, the connected data point (input or feedback value) is updated too, even if there is no value change. In cases where this behaviour is not desired, the “Analog Point COV Increment” or “Only notify on COV” property of the corresponding host data point must be set. See Section 7.2 for where these data point settings can be made.

6.7.1.22 MinSendTime

The minimum send time property is available for all I/Os. It specifies the minimum time that has to pass before a new update of an input or output feedback value is generated. If MinSendTime is set to 0, all changes of the inputs or output feedbacks immediately generate updates. For inputs with interpretation “Frequency” or “Frequency Table” (see Section 6.7.1.4), the MinSendTime property also specifies the period used for frequency measurement.

6.7.1.23 TransTable

The translation table property is available for all inputs with interpretation “Translation Table” or “Frequency Table” (see Section 6.7.1.4). It specifies the translation table which shall be used for the translation. The configuration of translation tables is explained at the end of Section 7.17.2.

6.7.1.24 NTC_Rn, NTC_Tn, and NTC_B

The Rn, Tn, and B properties are available for inputs with interpretation “Custom NTC” (see Section 6.7.1.4). Tn is specified in degree Celsius. Rn is the resistance of the NTC temperature sensor at the temperature Tn. The temperature value is calculated using the formula:

$$T = B * (Tn + 273.16 \text{ degC}) / (B + \ln(R / Rn) * (Tn + 273.16 \text{ degC})) - 273.16 \text{ degC}.$$

T is the calculated temperature in degree Celsius and R is the measured resistance of the NTC temperature sensor.

6.7.1.25 HoldTime and DebounceTime

These properties are available for all inputs where the Interpretation is set to “Occupancy”. The HoldTime is also available for the data signal of STId card readers, see Section 6.7.2. The DebounceTime is also available for inputs with interpretation “Digital”.

In interpretation “Occupancy”, the occupied state is detected using a sensor which creates pulses whenever the room is occupied. These pulses are converted to an occupancy value in the following way:

- When a pulse is detected, the value changes to OCCUPIED and stays there for at least the **HoldTime**.
- When new pulses are detected during the **HoldTime**, the timer for the **HoldTime** is started again.
- When no pulse is detected within the **HoldTime**, the state goes back to UNOCCUPIED.
- From this moment on, all pulses from the sensor are ignored for as long as the **DebounceTime** lasts. This is useful e.g. when light is switched off due to detecting the UNOCCUPIED state which leads to new pulses of the sensor and in turn leads to switching on the light again. To break this loop, the debounce time is used.

In case of interpretation “Digital”, the **DebounceTime** is used for spike suppression. It specifies the maximum duration of an unwanted spike to be suppressed.

6.7.1.26 OffValue and OnValue

The off and on value properties are available for analog outputs in digital mode. They specify the physical values (voltages) to be output for the OFF (inactive) and ON (active) state.

6.7.1.27 PWMPeriod

The PWM period property specifies the period (in seconds) for outputs with Interpretation “PWM” (Pulse Width Modulation), see Section 6.7.1.4.

6.7.1.28 NominalPower

The nominal power property is available for all outputs. It specifies the nominal resp. average expected power consumption of the device or appliance (e.g. lamp) connected to the output. For digital outputs it specifies the average power when the output is activated

(e.g. relay closed), for analog outputs it specifies the average power when the output is set to 10 V. The nominal power property is used to calculate the energy count data point of the output.

6.7.2 STId Card Reader Mode

The L-IOB devices support STId card readers running the ISO2 protocol. Refer to Section 15.3 and the following sections for information on which L-IOB models support STId card readers. The card readers have three signals which must be connected to inputs on the L-IOB device, as explained in the following sections.

6.7.2.1 Code Signal

The code signal of the card reader must be connected to a universal or digital input of the L-IOB device. The following properties must be set for that input:

- SignalType (if universal input): “Voltage 0-10V”.
- Interpretation: “Pulse Count”.
- IOFunc: “Cardreader ISO2”.
- GroupNumber: must be equal for all three signals of one card reader.
- SubGroupNumber: will automatically be set to “Code”.
- Invert: checked if signal is low active, check data sheet of card reader.

In its live value, this L-IOB input will represent the number of codes read from the card reader.

6.7.2.2 Data Signal

The data signal of the card reader must be connected to a universal or digital input of the L-IOB device. The following properties must be set for that input:

- SignalType (if universal input): “Voltage 0-10V”.
- Interpretation: “Card Data”.
- IOFunc: will automatically be set to “Cardreader ISO2”.
- GroupNumber: must be equal for all three signals of one card reader.
- SubGroupNumber: will automatically be set to “Data”.
- Invert: depends on implementation of card reader, check data sheet.
- HoldTime: specifies the time, a new code remains in the L-IOB data point. After this time, the data point is set back to “0xBF ...”, which means “not available”. If the HoldTime is set to 0, the last read value remains until a new value is read.

In its live value, this L-IOB input will represent the read code in a 20-byte array (40 decimal digits).

6.7.2.3 Clock Signal

The clock signal of the card reader must be connected to an interrupt-capable universal or digital input of the L-IOB device. The corresponding interpretation “Clock” will only be

available in the Configurator software, if the input is interrupt-capable. The following properties must be set for that input:

- **SignalType** (if universal input): “Voltage 0-10V”.
- **Interpretation**: “Clock”.
- **IOFunc**: will automatically be set to “Cardreader ISO2”.
- **GroupNumber**: must be equal for all three signals of one card reader.
- **SubGroupNumber**: will automatically be set to “Clock”.
- **Invert**: checked if signal is low active, check data sheet of card reader.

The live value of that L-IOB input will not deliver any useful data.

6.7.3 I/O Data Points

This section describes the I/O data points available in a LIOB-18x/48x/58x. For data points which can be read and written, value data points will be created. For each I/O, the most basic data points are created automatically (input value, output value, feedback). In the Configurator, the user can choose to create lots of additional data points on the common and I/O level.

6.7.3.1 Common I/O Data Points

The common local I/O data point names are preceded with ‘L1_1’. There is only one common local I/O data point, which is not assigned to a specific I/O:

- **ManualMode**: flag is set if at least one I/O is in manual mode.

If a LIOB-48x/58x device is extended by a LIOB-45x/55x module over the LIOB-IP bus, the common I/O data points for the LIOB-45x/55x module are preceded with ‘L2_1’, e.g. ‘L2_1_**ProductCode**’. The following common I/O data points exist for a connected LIOB-45x/55x device:

- **ProductCode**: the product code of the connected LIOB-45x/55x, e.g. ‘LIOB-450’.
- **DeviceName**: name of the LIOB-45x/55x device.
- **NID**: unique Node ID of the LIOB-45x/55x device.
- **StationID**: must always be 1.
- **PinCode**: pin code for the LCD UI of the LIOB-45x/55x device.
- **ErrorMask**: last error of the LIOB-45x/55x device.
- **CfgExists**: flag is set if configuration for this device exists (always set).
- **Enabled**: flag is set if this device is enabled.
- **Online**: flag is set if this device is online.
- **NotDetected**: flag is set if this device could not be detected.
- **EnableUpgrade**: flag is set if firmware upgrade is allowed for this device.

- **AlternativeUnit:** flag is set if this device shows US units (instead of SI units).
- **ManualMode:** flag is set if at least one I/O is in manual mode.
- **FMWVersion:** firmware version of this device.
- **FMWTimestamp:** firmware build time stamp (date and time) of this device.
- **Serial:** serial number of this device.
- **LastRebootTimestamp:** date and time of last reboot of this device.
- **SystemTemp:** current system temperature of the device.
- **SystemVoltage:** current supply voltage of this device.
- **CPUload:** current CPU load of this device.

6.7.3.2 I/O specific Data Points for Inputs

The I/O specific data point names are preceded with 'L1_1' or 'L2_1' and the I/O terminal name, e.g. "L1_1_UI5_Input".

- **Input:** current value of the input as measured and interpreted by the L-IOB device, if the operating mode is set to "Auto". This data point will go out-of-service, if the sensor is detected disconnected. If the operating mode is set to a mode different than "Auto", the input value will be set to the corresponding manual, override, or default value, see Section 6.7.1.9.
- **IOStatus:** current status of input. This data point will go from "OK" to "Disconnected", if the sensor is disconnected.
- **PulseTime:** time between the last two pulses for signal interpretation "Pulse Count" or "Physical Unit Count" (see Section 6.7.1.4). This can be used to quickly detect a change of the derivative of the physical value, e.g. a change of the electric power if the measured physical value is electric energy. Note that the PulseTime data point must be activated via the PulseTime flag, see Section 6.7.1.12.
- **OperatingMode:** see Section 6.7.1.9.
- **OverrideValue:** see Section 6.7.1.9.
- **DefaultValue:** see Section 6.7.1.9.
- **Offset:** see Section 6.7.1.19.
- **MinValue:** see Section 6.7.1.20.
- **MaxValue:** see Section 6.7.1.20.
- **HoldTime:** see Section 6.7.1.25.
- **DebounceTime:** see Section 6.7.1.25.
- **PulseCountInit:** when this data point is written, the pulse count is reset to the written value. This applies to inputs with Interpretation "Pulse Count" or "Physical Unit Count", see Section 6.7.1.4.

6.7.3.3 I/O specific Data Points for Outputs

The I/O specific data point names are preceded with 'L1_1' or 'L2_1' and the I/O terminal name, e.g. "L1_1_DO4_Output".

- **Output:** current value of the output as set by the logic application.
- **Feedback:** the feedback data point will always be set to the current physical value of the output. See Section 6.7.1.9 for details.
- **IOStatus:** current status of output. This data point always has the value "OK".
- **RunHours:** total time the digital output was switched on.
- **EnergyCount:** energy consumption of the device or appliance (e.g. lamp) connected to the output, see also Section 6.7.1.28.
- **SwitchCycles:** Number of switching cycles of the digital output. This is useful for estimating the expected remaining life time of a relay. Observe that this data point is never reset, not even by a reset to factory defaults, see Section 3.7.
- **OperatingMode:** see Section 6.7.1.9.
- **OverrideValue:** see Section 6.7.1.9.
- **DefaultValue:** see Section 6.7.1.9.
- **Offset:** see Section 6.7.1.19.
- **MinValue:** see Section 6.7.1.20.
- **MaxValue:** see Section 6.7.1.20.
- **PWMPeriod:** see Section 6.7.1.27.
- **RunHoursInit:** when this data point is written, the RunHours data point is reset to the written value.
- **EnergyCntInit:** when this data point is written, the EnergyCount data point is reset to the written value.
- **NominalPower:** see Section 6.7.1.28.

6.7.4 Default I/O Configuration

Every L-IOB device is shipped with a default I/O configuration, which can be used right away. This configuration is dependent on the hardware type (see Section 6.7.1.2) of the corresponding I/O:

- **Universal Inputs (UI):** measure voltage between 0V and 10V. The interpretation is 'Linear' with MinValue 0V and MaxValue 10V.
- **Digital Inputs (DI):** act as switch inputs. The interpretation is 'Digital'.
- **Pressure Sensors (PRESS):** measure pressure between 0P and 500P (Pascal). The interpretation is 'Linear' with MinValue 0P and MaxValue 500P.
- **Analog Outputs (AO):** deliver a voltage between 0V and 10V according to the given percentage level. The interpretation is 'Analog' with MinValue 0V and MaxValue 10V.

- **Digital Outputs (DO):** relays or triacs close when the given value goes active and open when the value goes inactive. The interpretation is ‘Digital’.

6.8 CEA-709 Technology (LIOB-18x/48x)

6.8.1 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).

The CEA-709 NVs on the device are created in the form of static NVs. 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 6.8.2).

Based on the NV, 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”.

6.8.2 Static Interface Changes

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 device in the LNS database and restores the previous bindings and dynamic NVs. If the device 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.

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 77.



Figure 77: Setting a model number range in for a project

6.8.3 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 78.

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 78: 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 7.3.3). 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.

6.8.4 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.

6.8.5 Limitations for Local CEA-709 Trends (LIOB-48x)

Local CEA-709 trend objects support trending multiple data points in all trend modes, interval, COV, and trigger. 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.

6.8.6 Dynamic Polling in CEA-709

External input NVs in CEA-709 rely on polling. Static polling can be configured by setting the poll cycle (see Section 6.3.2). In addition, this technology also supports dynamic polling. If the L-WEB requires a refresh on those NVs, polling is activated at the requested rate. If those data points go out of scope, the polling on the external input NVs falls back to the static poll interval. If no static polling is needed at all, the poll cycle 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 poll cycle, no change of the poll cycle is made at run-time.

6.8.7 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.

6.9 BACnet Technology (LIOB-58x)

6.9.1 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 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.

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.

6.9.2 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, value objects (BI, BV), and for multi-state input, value objects (MSI, MSV). Since there is no value source for the

Feedback_Value property available, alarm conditions cannot be defined for binary output (BO) and multi-state output (MSO) objects.

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.

6.9.3 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 79. 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 79: 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.

6.9.4 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 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.

6.9.5 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 poll cycle (see Section 6.3.2). In addition, this technology also supports dynamic polling on devices, that do not support COV. If the L-WEB requires a refresh on those client mappings, polling is activated at the requested rate. If those data points go out of scope, the polling on the remote BACnet object falls back to the static poll interval.

If no static polling is needed at all, the poll cycle 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.

6.9.6 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**.

6.9.7 Native BACnet Objects for I/Os

L-IOB I/Os can optionally be directly exposed to BACnet server objects. In this case, each L-IOB I/O object is represented by one BACnet object. The mapping does more than a regular data point connection for the sensor/actuator value. All relevant I/O configuration

properties are mapped to corresponding BACnet properties and reflect actual input/output values, I/O status, override/manual values, operating mode, inversion and adhere to all prescribed BACnet functions.

The type of the created BACnet server object depends on the type of the live value I/O data point. If e.g. a universal input (UI) is used for measuring an analog value, the type of the live value (**Input** data point) will be Double and thus, an Analog Input (AI) BACnet object will be created. Table 7 shows all possible native BACnet object types for I/Os.

I/O	I/O Live Value Type	BACnet Object	Feedback Object
DI/UI	Double	Analog Input (AI) or Accumulator	-
DI/UI	Boolean	Binary Input (BI)	-
DI/UI	LIOB/MagCard	Analog Input (AI)	-
AO/DO	Double	Analog Output (AO)	Analog Input (AI)
AO/DO	Boolean	Binary Output (BO)	-

Table 7: Native BACnet object types for I/Os

A property relation ‘native’ (see Section 6.3.11) is added to the original L-IOB input or output data point, which links to the created native BACnet data point. For an AO object, an additional AI object is generated as the feedback value object. The data point of the AO object has a ‘feedbackValue’ property relation that points to the feedback AI object. For BO objects, a separate feedback object is not required, since a feedback value is already included in the BACnet BO object.

For the STId Card Reader mode (see Section 6.7.2), the I/O live value type of the Card Data input is an array of 40 nibbles which contains the last read Card ID in BCD encoding (LIOB/MagCard). In this case, the first n BCD digits that form a number which can be displayed by a BACnet float are written to the Present_Value of the created BACnet AI object. An ASCII version of the entire BCD code is written to the Description property of the BACnet object.

For the “Pulse Count” interpretation of inputs (see Section 6.7.1.4), it is possible to choose between an analog input or an accumulator object in the Configurator. The main difference is that accumulator objects can precisely represent 32-bit meter data while analog objects suffer from a loss of resolution.

When using native BACnet objects for L-IOB outputs, the BACnet priority array concept is used in the I/O operating mode ‘Auto’ for determining the physical output value. Other I/O operating modes (‘Override’, ‘Manual’, ‘Disabled’, see Section 6.7.1.9) bypass the BACnet priority array. The physical value is determined by the override value or manual value in this case. The following L-IOB live value and configuration property data points, which can be changed at run-time, are exposed to BACnet properties:

- **Input:** For L-IOB inputs in ‘Auto’ mode, the input value is written to the Present_Value property of the BACnet input object. When switching from another operating mode to ‘Auto’, the current input value is also written.
- **Output:** The output value written by the controller application to the BACnet output object is reflected at the automatic priority in the Priority_Array of the BACnet object. The resulting Present_Value of the BACnet object is then written to the physical output in ‘Auto’ mode.
- **OperatingMode:**
 - **Auto:** The BACnet object is in normal operation as described above. Out_of_service and OVERRIDDEN flags of the BACnet object are cleared.

- **Override (Input):** The BACnet input object is set to out-of-service. The Present_Value no longer reflects the physical L-IOB input value. The L-IOB override value is coupled to the Present_Value and vice-versa. Writing the Out_Of_Service property over the BACnet network can turn the 'Override' mode on or off (except if in 'Manual' or 'Disabled' mode).
 - **Override (Output):** The BACnet output object is set to out-of-service. The Present_Value no longer writes to the L-IOB output value. The L-IOB override value is coupled to priority slot '1' and vice-versa. It is removed from priority slot '1' when returning to 'Auto' mode. Writing the Out_Of_Service property over the BACnet network can turn the 'Override' mode on or off (except if in 'Manual' or 'Disabled' mode).
 - **Manual:** The BACnet object is set OVERRIDDEN. The Present_Value reflects the manual value and is decoupled from the L-IOB input value (input) or Priority_Array (output). Out_Of_Service is not set. The Out_Of_Service, Present_Value, and Reliability properties are made read-only and can no longer be written over BACnet. This mode cannot be modified over the BACnet network.
 - **Disabled:** The BACnet object is set out-of-service, the OVERRIDDEN flag is set, and the reliability is set to 'no fault detected'. The Out_Of_Service, Present_Value, and Reliability properties are made read-only and can no longer be written over BACnet. This mode cannot be modified over the BACnet network.
- **OverrideValue:** In 'Override' mode this value is written to the Present_Value of the BACnet object at priority '1'. When entering 'Override' mode, the current override value is written. When leaving 'Override' mode, NULL is written at priority '1' for output objects and nothing is done for input objects. For input objects in 'Override' mode, the override value is updated by the Present_Value when written over BACnet.
 - **ManualValue:** In 'Manual' mode, the L-IOB manual value is written to the Present_Value property of the BACnet object. The Present_Value cannot be written over BACnet in 'Manual' mode.
 - **DefaultValue:** The L-IOB default value is written to the Relinquish_Default property of the BACnet object and vice-versa, if it exists.
 - **Invert:** This L-IOB parameter is written to the Polarity property of BACnet BO and BI objects and vice-versa.
 - **I/OStatus:** The I/O status is reflected in the Reliability property of the BACnet object:
 - NO_FAULT_DETECTED: If the I/O does not report any error.
 - NO_OUTPUT/NO_SENSOR: If an output or sensor failure is detected.
 - COMMUNICATION_ERROR: If the L-IOB host reports a communication error.
 - UNRELIABLE_OTHER: For all other problems.
 - **Feedback:** For L-IOB outputs in digital mode, the feedback value is written to the Feedback_Value property of the BACnet BO object. For L-IOB outputs in analog mode, the feedback value is written to the dedicated BACnet AI object, which is pointed to by the 'feedbackValue' property relation.
 - **MinValue, MaxValue, Resolution:** For analog BACnet objects and BACnet Accumulator objects, these L-IOB configuration properties are written to the corresponding BACnet properties when they change. For a BACnet Accumulator object, the L-IOB Resolution is mapped to the BACnet Scale property. The BACnet properties are read-only from the BACnet network.
 - **I/O Name and Description:** Initially, the BACnet server object name and description are both set to e.g. 'LI_1_UI1' (for UI1), where the preamble is 'LI_1' for local I/Os

and 'L2_1' for an attached LIOB-55x device. In the Configurator, the BACnet server object name and description can later be modified manually or synchronized with the current I/O name and description.

- **PulseCountInit:** If the BACnet object type is Accumulator and this L-IOB configuration property is written (to reset the pulse counter), the value is also written to the Value_Set property of the BACnet Accumulator object and vice versa.

6.10 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 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.

7 The Configurator

This Chapter gives step-by-step instructions on how to commission a L-IOB I/O controller, create a data point configuration in the L-INX Configurator, and how to manage and use those data points.

7.1 Installation

7.1.1 Software Installation

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, Service Pack 8 or LNS TE SP5, OpenLNS (for LNS mode),
- Windows XP, Windows 2003 Server, Windows Vista, Windows 7, Windows 2008 Server, Windows 8 (64 bit), Windows 2012 Server.

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 LIOB-18x/48x shall be configured using LNS-based tools (e.g., NL220 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 80).

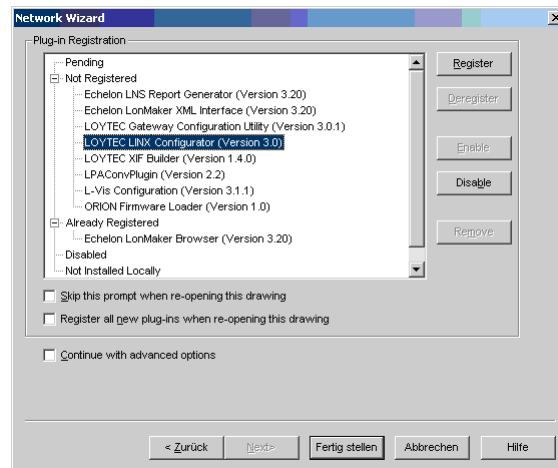


Figure 80: Select the Plug-in to be registered

3. Click **Register**. The Configurator now appears in the **Pending** list.
4. Click **Finish** to complete the registration. Device templates for LIOB-18x and LIOB-48x are added automatically and the 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.

5. Under **LonMaker** → **Network Properties** → **Plug-In Registration** make sure that the **LOYTEC LINX Configurator (Version X.Y)** shows up under **Already Registered**.

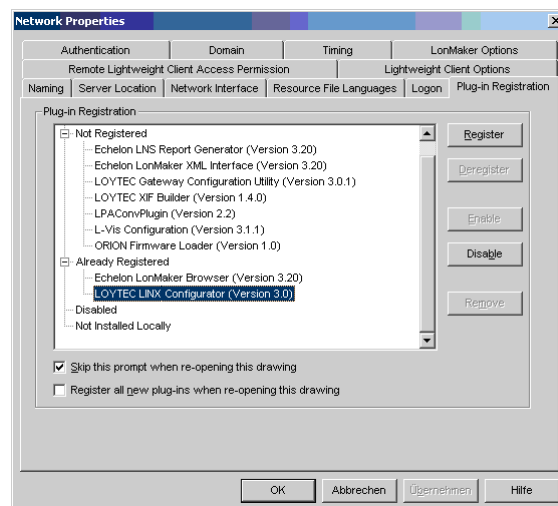


Figure 81: Check that the L-INX Configurator is properly registered

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 82. 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 82),
- The data point list (number 2 in Figure 82),
- And a property view (number 3 in Figure 82).

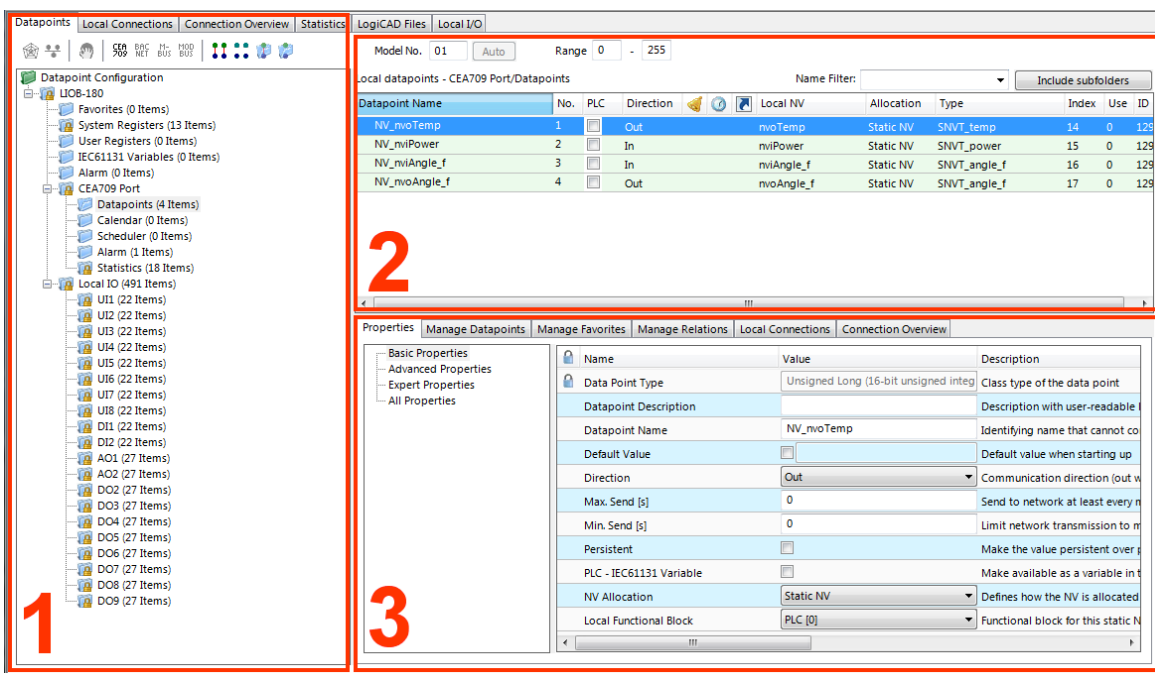


Figure 82: Data Point Manager Dialog

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 (LI0B-58x):** This folder has a number of sub-folders for different import methods:
 - **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.

- **LIOB-18X/48X/58X:** This is the device folder of the LIOB. 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 are available:
 - **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 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.
 - **Alarm:** This folder contains generic alarm servers. These provide technology-independent alarms. Any data point can be alarmed using a generic alarm server.
 - **IEC61131 Variables:** This folder contains the variables from the IEC61131 logic if created from the logic tool.
 - **CEA709 Port (LIOB-18x/48x):** 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 (LIOB-58x):** This folder contains data points, schedulers, calendars, trend logs, statistics, remote data points, and data points of native BACnet objects (for I/Os) of the BACnet network technology. See Section 7.2.2.
 - **Local IO:** This folder contains the data points of the local I/Os. See Section 7.17.3.
 - **LIOB-IP:** This folder contains the data points of the optionally connected LIOB-45x/55x device. See Section 7.17.3.
- **Global Objects (LIOB-48x/58x):** 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.

7.2.2 Network Port Folders

Each network port folder (CEA709/BACnet) 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 (LIOB-48x/58x):** 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.

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 havin "val" in the name.

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 (LIOB-48x/58x) 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 (LIOB-48x/58x) 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 pressing the **New** button to the right of the list or via the **New** command in the context menu. A plus  sign in the list indicates that the data point contains sub-points. Clicking on the plus  sign expands the view.

For the alarming and scheduling (AST) features, there are columns which display icons for data points that are attached to an AST function. See Table 8 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 8: 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 codings are described in Table 9.





Color	Coded Information
 ao1 (orange)	Data point is user-created and already has already been downloaded to and exists on device.
 ao2 (green)	Data point is user-created and new. It hasn't yet been downloaded to the device and is therefore not yet usable on 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 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 9: 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.

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 (LIOB-48x/58x):** 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-821/900 master parameter editor. A parameter data point is also persistent. See Section 6.3.5.
- **PLC in - Logic Variable:** If enabled, the data point will be usable in the IEC61131 PLC program as an input variable.
- **PLC out - Logic Variable:** If enabled, the data point will be usable in the IEC61131 PLC program as an output variable.
- **Use Polycle value as:** For input data points, this property defines whether the input shall use a receive timeout or be constantly polling. See Section 6.3.2.
- **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 6.3.2.
- **Polycle:** For input data points, this property defines the poll cycle in seconds. Set this property to 0 to disable polling. See Section 6.3.2.
- **Receive Timeout:** For input data points, this property defines the receive timeout in seconds. Set this property to 0 to disable polling. See Section 6.3.2.
- **Min Send:** For output data points, this property defines the min send time in seconds. See Section 6.3.2.
- **Max Send:** For output data points, this property defines the max send time in seconds. See Section 6.3.2.
- **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 6.3.7.
- **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 6.3.7.
- **Custom Scaling Factor, Custom Scaling Offset:** These properties only exist, if linear scaling is enabled. They apply to analog data points only. See Section 6.3.7.
- **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 6.3.4.
- **Default Value:** This property defines a default value (see Section 6.3.3). 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 6.6.6.

- **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.
- **Unit:** For analog data points this property contains the definition of an engineering unit of the scalar value, e.g., “kg”. 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 6.3.12).
- **Unit for display:** If the data point has a convertible unit, an alternative unit for display can be chosen. The data point value on the Web UI is converted to and displayed in the display unit (e.g., °F instead of °C). For a non-convertible 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 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 83.

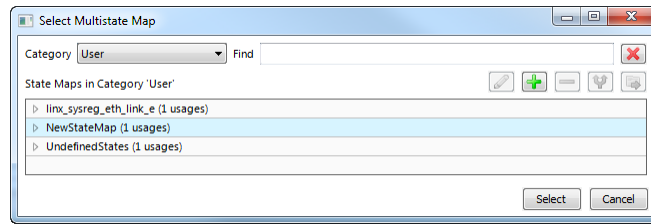


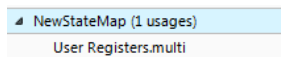

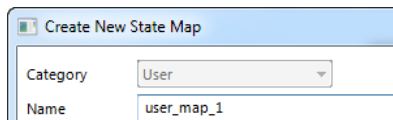


Figure 83: Select multistate maps in the multistate map manager

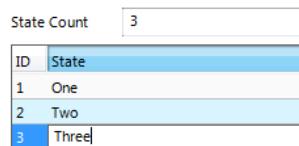
2. Select a **Category** and an existing state map in the **State Maps** list and click on **Select**. Maps that are fixed and cannot be changed are marked with a lock symbol .
3. Expand a state map line to see where this state map is used. Select the usage and click the **Go to Data Point** button . 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.

7.2.6 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:

Favorites - Favorites				
Link Name	OPC	PLC	Link Path	ID
my_name_humid_link	<input checked="" type="checkbox"/>	<input type="checkbox"/>	LINX-120.CEA709 Port.Datapoints.abs_humid	10FE

The **link name** can be edited to something different than 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 84. 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.

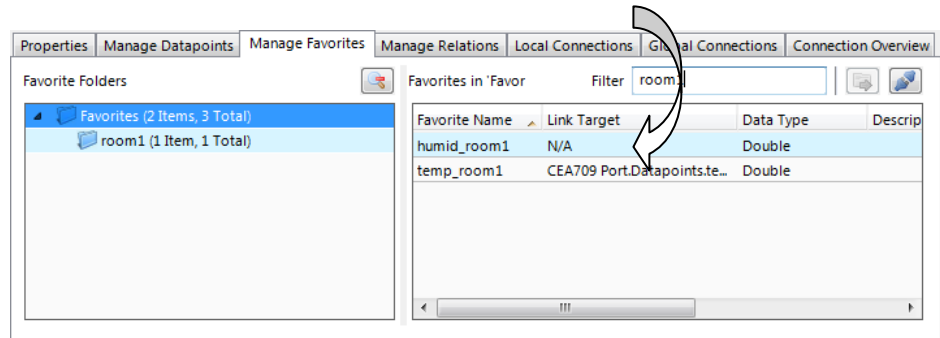





Figure 84: Manage favorites tab

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.

7.2.7 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 85. 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.

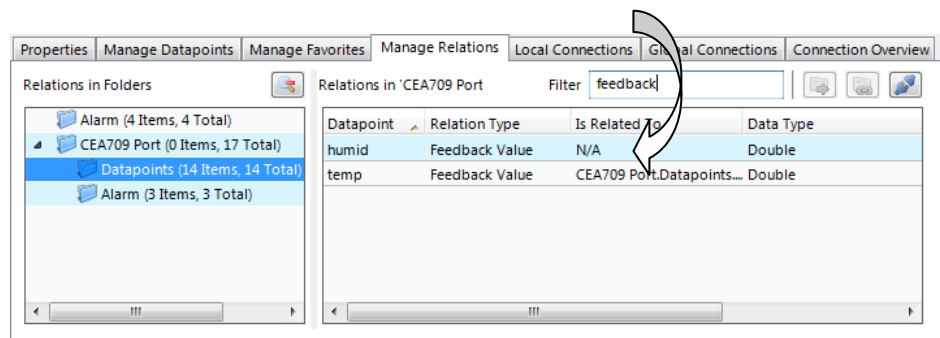



Figure 85: Manage relations tab

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.

7.2.8 CEA-709 Properties (LIOB-18x/48x)

Apart from the common data point properties discussed in Section 7.2.4 the data points of the CEA-709 technology have additional properties. Only the ones applicable for LIOB-18x/48x devices are listed here.

- **NV Allocation:** This property defines how a data point shall be allocated on the device. Only “Static NV” is available for LIOB-18x/48x devices.
- **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.
- **Local/Remote NV Name:** This property specifies the programmatic name of the NV.
- **Local/Remote Functional Block:** This property specifies the function block of the NV.
- **Local/Remote NV Flags:** This property specifies the NV flags.
- **Retry Count:** For external NVs, this property defines the retry count. The default is 3.
- **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.

7.2.9 BACnet Properties (LIOB-58x)

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.
- **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.
- **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 6.9.1).
- **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.

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 86 contains settings independent of the technology port. The settings relevant for LIOB-18x/48x/58x devices are:

- **Project Name:** This setting allows entering a descriptive name for the project.
- **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 and the device settings as defined in the **System Settings** tab (see Section 7.3.5).

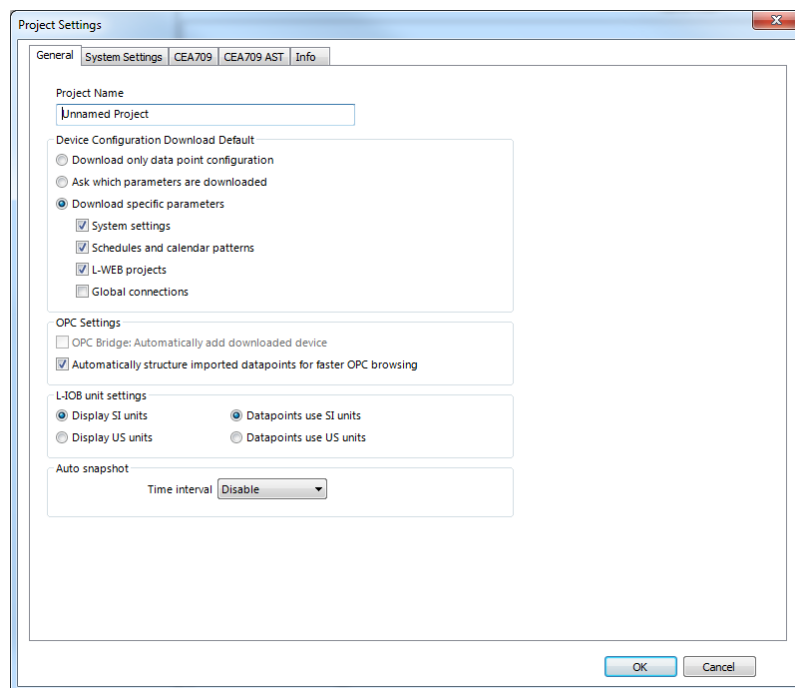



Figure 86: General Project Settings

- **Automatically add downloaded device to the OPC Bridge (LIOB-48x/58x):** This option is only available, if LOPC-BR800, the LOYTEC OPC bridge software, is installed on the same PC. If this setting is enabled, the device which a configuration is downloaded to, is automatically added to the OPC server list in the bridge. For more information on using LOPC-BR800 refer to Section 8.3.
- **Automatically structure imported data points for faster OPC browsing (LIOB-48x/58x):** 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.
- **L-IOB Unit Setting:** The ‘Display SI units’ / ‘Display US units’ setting defines whether SI units or US units are displayed in the LCD and Web UI. The ‘Datapoints use SI units’ / ‘Datapoints use US units’ setting defines whether actual processing of data points is done in SI units or US units.
- **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.

7.3.2 CEA-709 Settings (LIOB-18x/48x)

The CEA-709 configuration tab as shown in Figure 87 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.
- **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 unused 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 floating point SNVTs over fixed-point types, e.g. a SNVT_temp_f will be generated instead of a SNVT_temp.

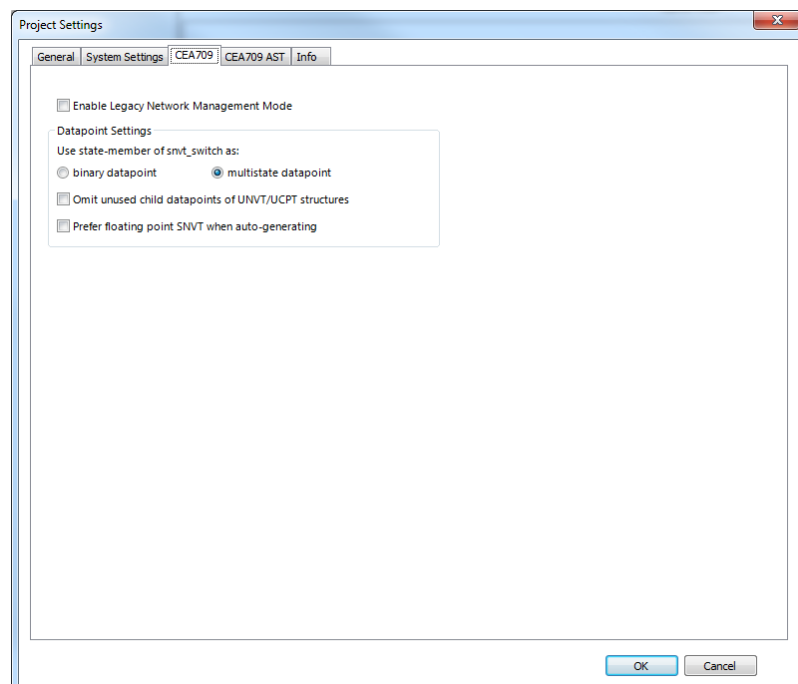


Figure 87: CEA-709 Project Settings

7.3.3 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.

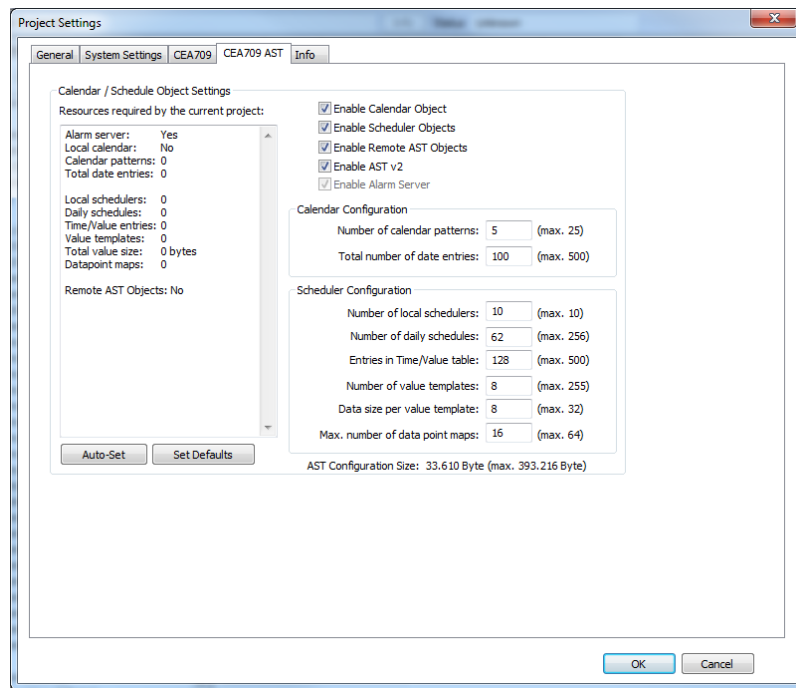


Figure 88: CEA-709 AST Project Settings

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.

7.3.4 BACnet Settings (LIOB-58x)

The BACnet configuration tab as shown in Figure 89 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).
- **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.

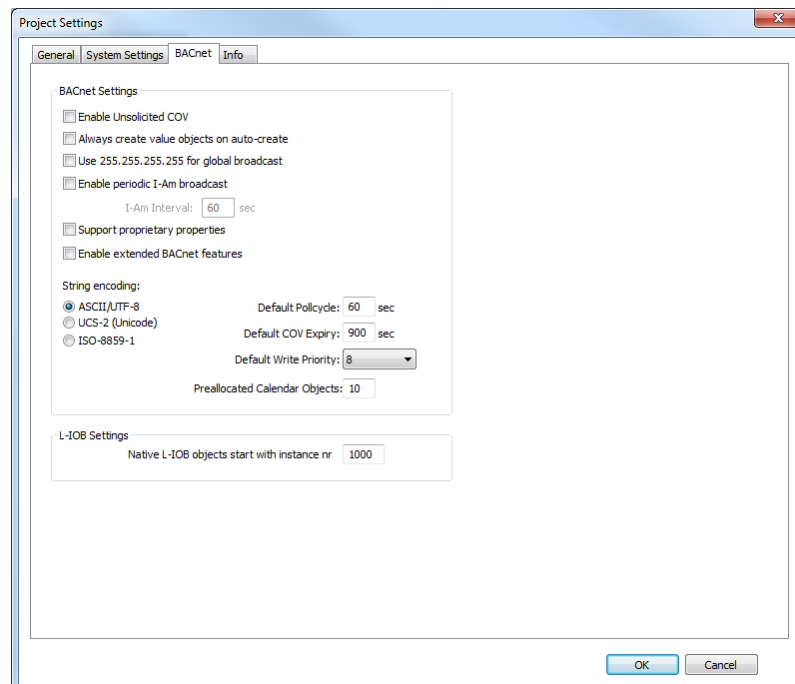


Figure 89: BACnet Project Settings

7.3.5 System Settings

This tab is shown in Figure 90. It 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.

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.

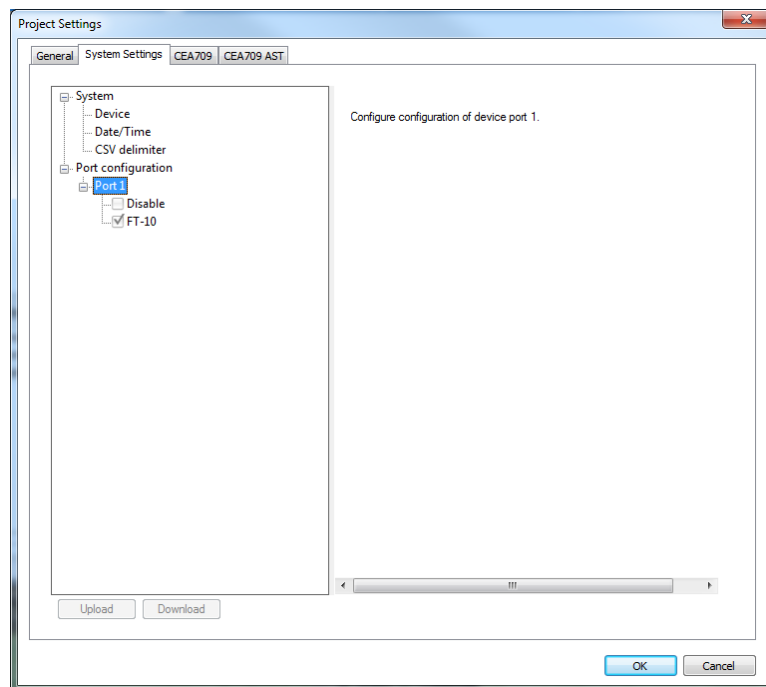


Figure 90: Device Configuration Settings

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 take effect only after a reboot of the device.

Important! *After downloading the device settings from this dialog, the device needs to be rebooted to let the changes take effect.*

7.3.6 Project Information

This tab is shown in Figure 91. 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.

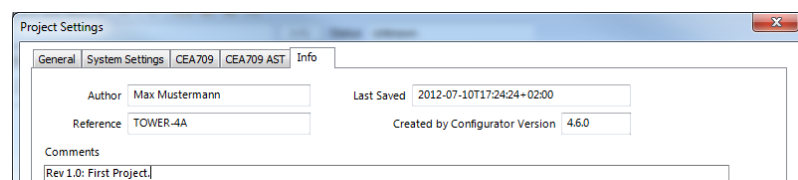


Figure 91: Info tab in project setting

7.4 Workflows in CEA-709 (LIOB-18x/48x)

This section discusses a number of work flows for configuring a LIOB-18x/48x in different use cases in addition to the simple use case in the quick-start scenario (see Chapter 2). 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 7.4.2)
- Non-LNS network management tool (see Section 7.4.3)

7.4.1 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 device online. There exists a standard XIF file for the LIOB-18x models (LIOB-18X_V1_FT-10.xif) and one for the LIOB-48x models (LIOB-48X_V1_IP-852.xif).
- L-INX Configurator project file: This file contains all ports, data points, and connections of a project. The file ends with “.liob1” and stores all relevant configuration data and is intended to be saved on a PC to backup the device’s data point configuration.

7.4.2 Configure with LNS

The flow diagram in Figure 92 shows the steps that need to be followed in order to configure the device in a network with LNS 3.x.

First, the device must be added to LNS (see Section 7.4.5). Then the L-INX Configurator must be started in plug-in mode to configure the device (see Section 7.7.1). In the Configurator, create the required static NVs (see Section 7.7.2). Finally, the configuration needs to be downloaded to the device via LNS (see Section 7.7.3). 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).

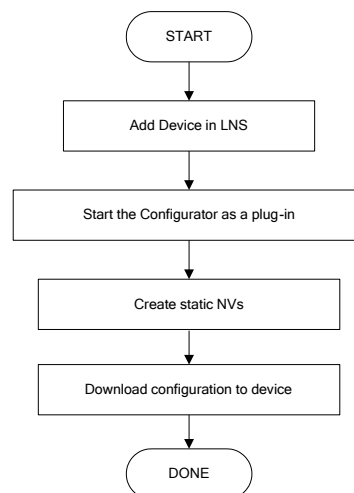


Figure 92: Basic design-flow with LNS

To add more NVs when all bindings are in place and the device is already being used, simply repeat the steps described above. The Configurator software will back up the bindings, download the new configuration, and re-create the bindings again.

7.4.3 Configure without LNS

The flow diagram in Figure 93 shows the steps that need to be followed in order to configure the device without LNS 3.x.

Start the Configurator in stand-alone mode and connect to the device via the device connect method (see Section 7.6.1). Create the required static NVs (see Section 7.7.2).

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 7.7.4). 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 7.7.5). 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).

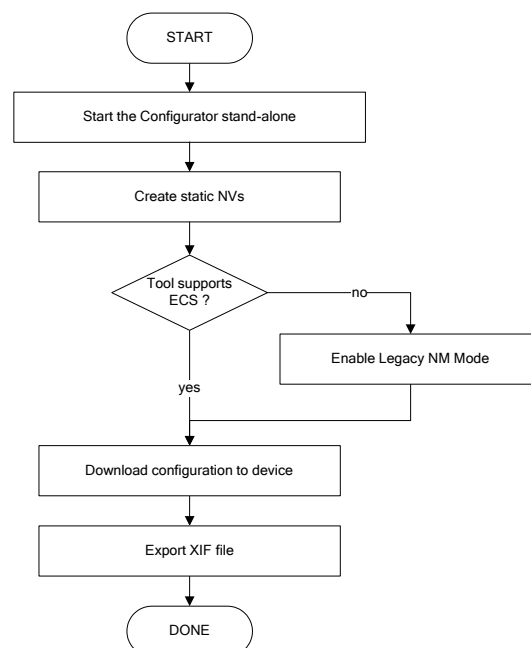


Figure 93: Basic design-flow without LNS

To use the device in the non-LNS management tool, commission the device using the exported XIF file and create the bindings.

When changing the static interface of a running device configuration with existing bindings, 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 6.8.2).

7.4.4 Device Replacement

A device can be replaced in the network by another unit. This might be necessary if a hardware defect occurs. The work flow is depicted in Figure 94.

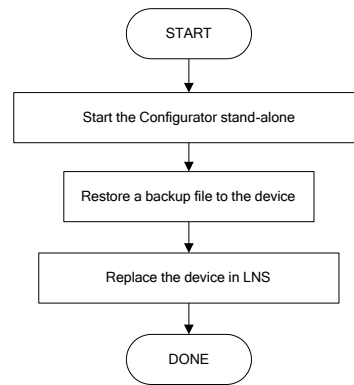


Figure 94: Basic work flow to configure a replacement device

Start the Configurator software stand-alone and connect via the device connect 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 and bindings are restored. The device is again configured online and fully functional in the CEA-709 network.

If using an LNS-based tool, the device needs to be replaced in that tool at some later point in time (see Section 7.4.6) 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.

7.4.5 Adding the Device to LNS

To configure the 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 LIOB-18x/48x to your database.

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 95.

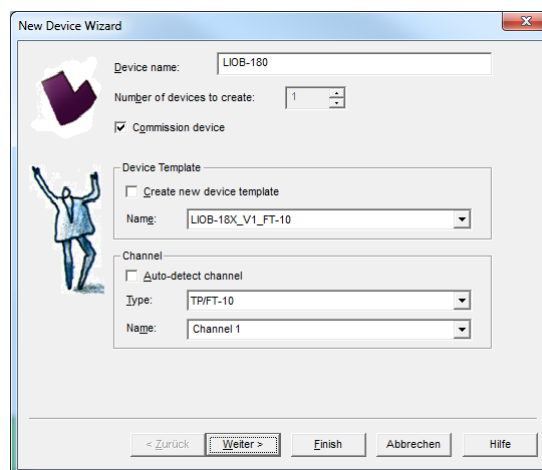


Figure 95: Create a new device in the drawing

2. Select **Commission Device** if the device is already connected to the network.
3. In the **Device Template** group box select the existing device template of the LIOB-18x (“LIOB-18X_V1_FT-10”) or LIOB-48x (“LIOB-48X_V1_IP-852.xif”).

4. Select the channel, which the device is connected to and click **Next**.
5. The following dialog shown in Figure 96 appears, click **Next**.

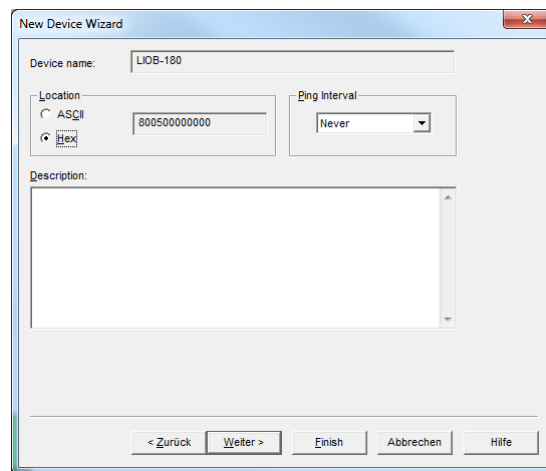


Figure 96: Leave defaults for Location

6. Check Service Pin as the device identification method as shown in Figure 97 and click **Next**.

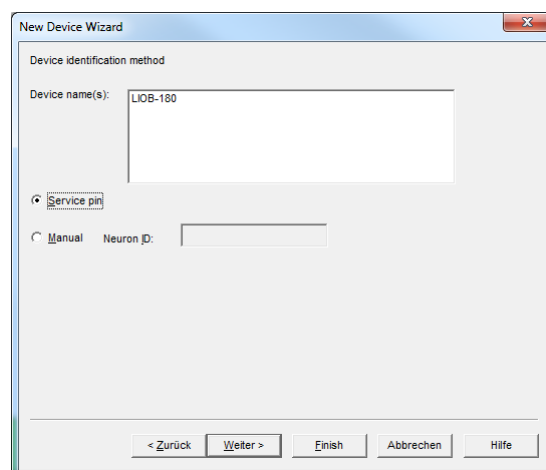


Figure 97: Use Service Pin

7. Click **Next** in the following screens until you get to the final dialog shown in Figure 98.
8. If the device is already on-net, select **Online**.

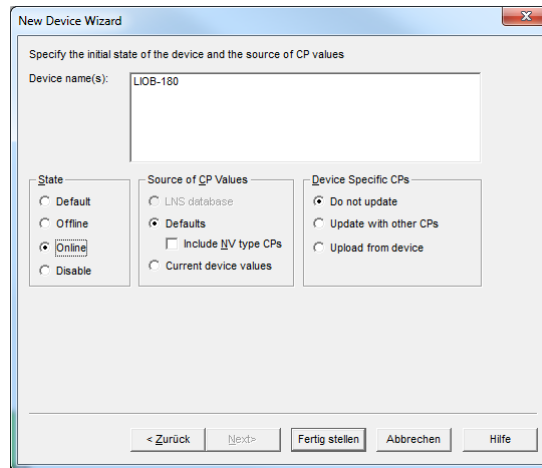
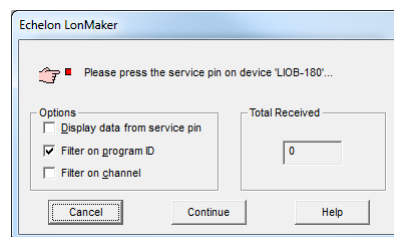


Figure 98: Final dialog

- Click **Finish**. A dialog will prompt to press the service pin.



- Finally, you should get the device added to your drawing as depicted in Figure 99.

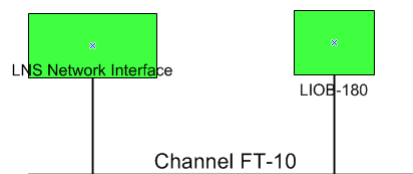


Figure 99: The L-IOB device has been added to the drawing

7.4.6 Replace a device in LNS

This Section describes how to replace a L-IOB device in your LNS database. The example discusses a LIOB-180 but is general to all models. The description refers to LonMaker TE. Let's assume there is a device 'LIOB-180' in the LNS database as shown in Figure 99.

To Replace a Device in LonMaker TE

- Select the device and right-click on the device shape.
- Select **Commissioning** → **Replace....** This opens the LonMaker Replace Device Wizard as shown in Figure 100.

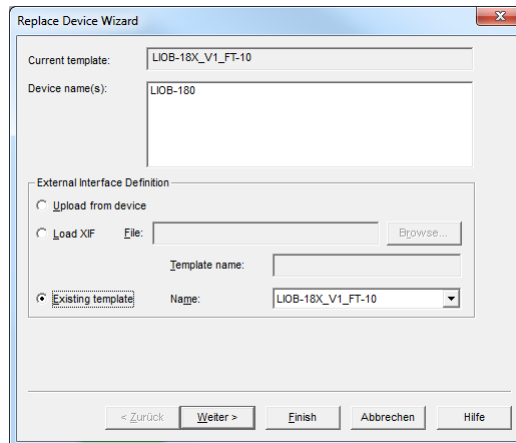


Figure 100: LonMaker replace device wizard

3. Choose the existing device template and click **Next**.
4. In the following window shown in Figure 101 click **Next**.

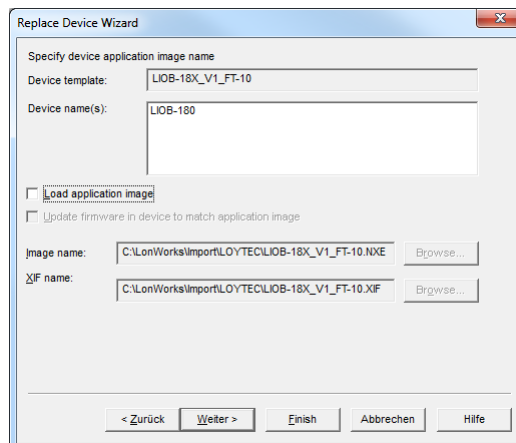


Figure 101: Click Next without loading an application image

5. Then select **Online** as shown in Figure 102 and click **Next**.

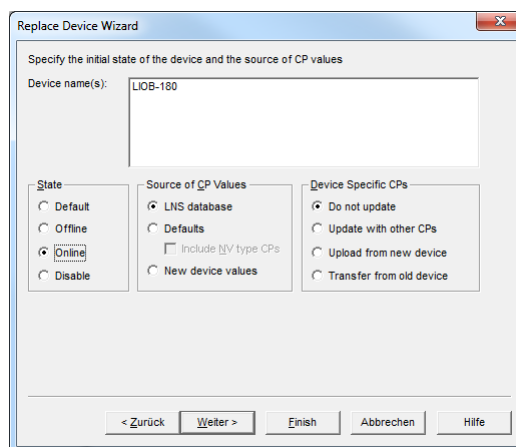


Figure 102: Select online state

6. Select the **Service pin** method and click on **Finish** as shown in Figure 103.

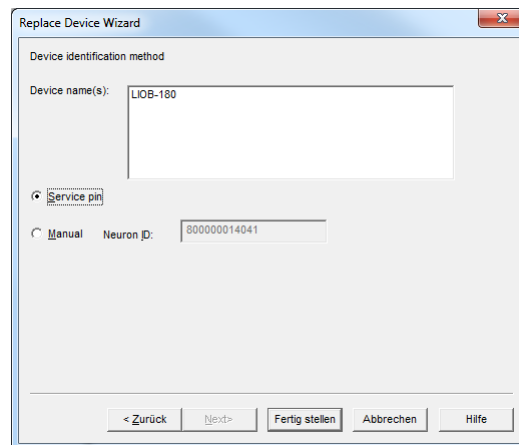


Figure 103: Select Service Pin and click Finish

- Then the service pin requestor opens as shown in Figure 104. Press the status button on the replacement device.

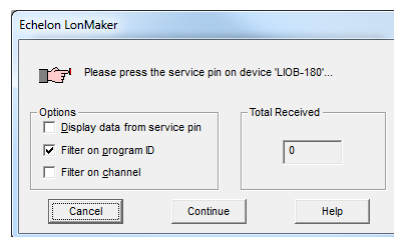


Figure 104: Wait for the service pin from the device

- After the service pin has been received, LonMaker commissions the replacement device and installs the bindings.

7.5 Workflows in BACnet (LIOB-58x)

This section discusses a number of work flows for configuring a LIOB-58x device in different use cases in addition to the simple use case in the quick-start scenario (see Chapter 2). 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.

7.5.1 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. The file ends with “.liob1”. It stores all relevant configuration data and is intended to be saved on a PC to backup the 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 Engineer Online

The flow diagram in Figure 105 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.

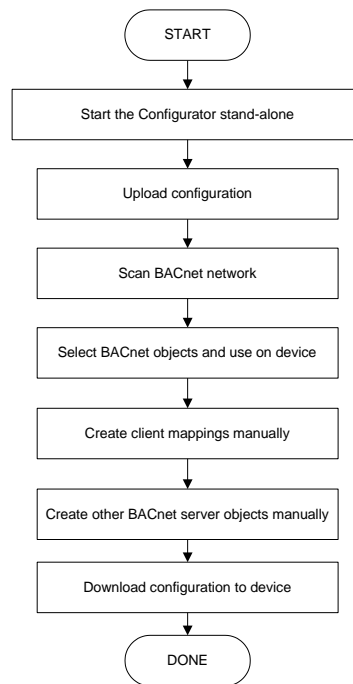


Figure 105: Basic work flow to engineer on-line

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).

7.5.3 Engineer Offline

The flow diagram in Figure 106 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.

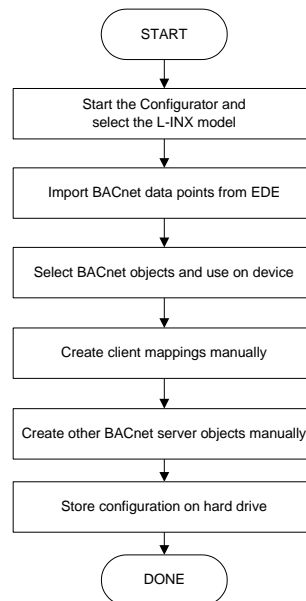


Figure 106: Basic work flow to engineer off-line

Start the Configurator in stand-alone mode and select the appropriate L-IOB 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).

7.6 Using the L-INX Configurator

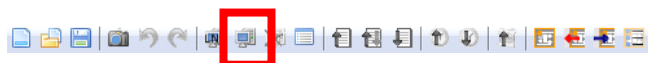
7.6.1 Starting Stand-Alone

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 device is not yet connected to the network, go to the **Model** menu and select the model to be configured. If the device is already connected to the network it is recommended to connect the Configurator online to the device.

To Connect to a LIOB-18x Stand-Alone

1. Select the device 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 107 opens containing the managed device connection templates.

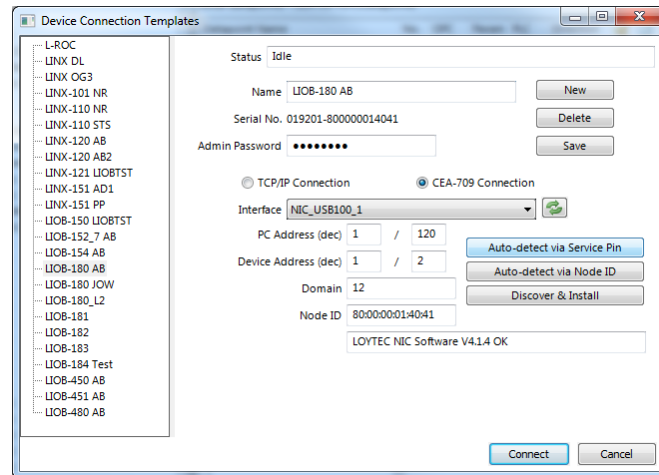


Figure 107: CEA-709 connection dialog

2. To add a new device connection, click on **New** or select an existing connection in the tree on the left-hand side.
3. Choose 'CEA-709 Connection' and select a LOYTEC network interface, e.g. 'NIC_USB100_1'. Refer to the NIC User Manual [1] for more information on LOYTEC network interfaces.
4. Choose a CEA-709 address for the PC and enter the address of the device to connect to. If the device is already commissioned, it can also be detected automatically via Service Pin or Node ID using the corresponding buttons.
5. If the device has not yet been commissioned, it can be discovered and installed using the corresponding button. Refer to Section 7.8.1 for more information.
6. Click on **Save** to store that connection.
7. Click on **Connect**. This establishes the connection to the device.

To Connect to a LIOB-48x/58x Stand-Alone

1. Select the FTP 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 107 opens containing the managed device connection templates.

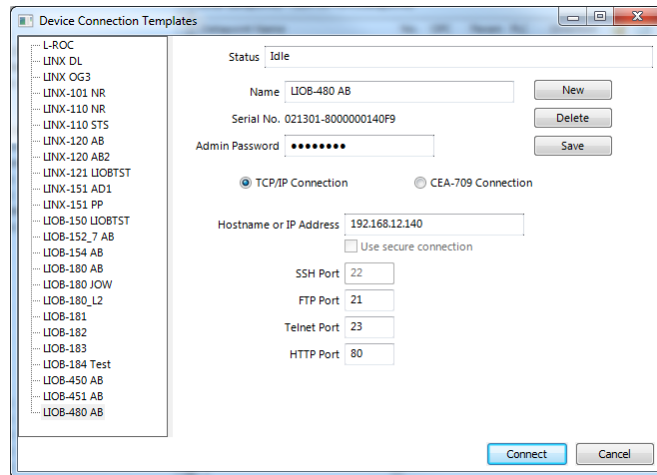


Figure 108: TCP/IP connection dialog

2. To add a new device connection, click on **New** or select an existing connection in the tree on the left-hand side.
3. Enter the host name or IP address of the device. Enter the admin password. The default password is 'loytec4u' (older firmware versions used 'admin').
4. If your device is located behind a NAT router or firewall, you may change the FTP, Telnet, SSH, and HTTP ports to your needs for accessing the device.
5. Click on **Save** to store that connection.
6. 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, schedules, etc.

To Upload a Configuration

1. Click on the **upload** button



in the tool bar. The configuration upload dialog opens up as shown in Figure 109.

2. Click on the button **Start** to start the transfer. This will upload the configuration of all ports.

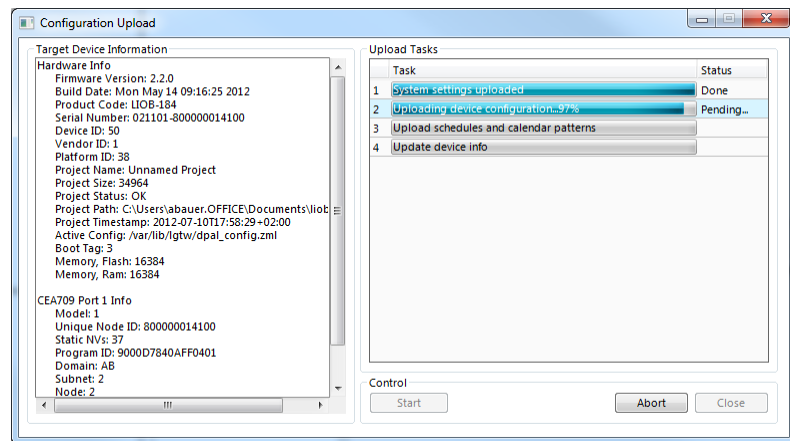


Figure 109: Configuration upload dialog

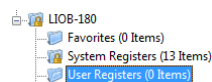
3. When the project settings of the configuration being uploaded specify to ask which specific parameters shall be uploaded, check the needed information and proceed.
4. If parameters are selected to be uploaded (LIOB-48x/58x) and their values have changed on the device, the parameter merge dialog appears. For resolving the reported conflicts refer to Section 6.3.5.

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 110.

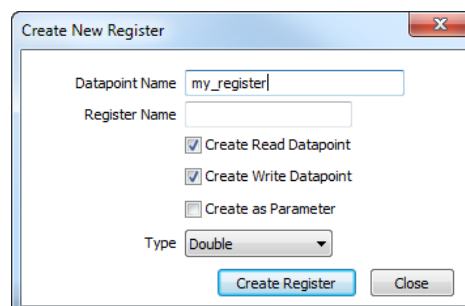

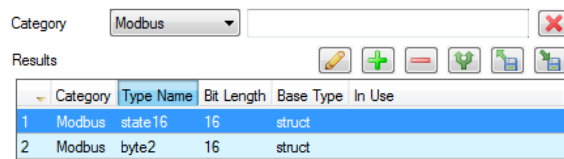


Figure 110: Create a user register

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. 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.
5. 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.
6. Select a **Type**. Available are “Double”, “Boolean”, “Signed Integer”, “String”, “Variant” or “User”.
7. 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.
8. 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**.



9. Click **Create Register**.
10. 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.

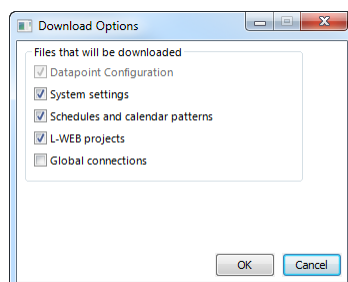
To Download a Configuration

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 111.

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. You can choose to reset all I/O counter values on the device with the checkbox **Reset Counters**.

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. 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 6.3.5.
6. When the download process has finished, a notification window appears, which has to be acknowledged by clicking **OK**.

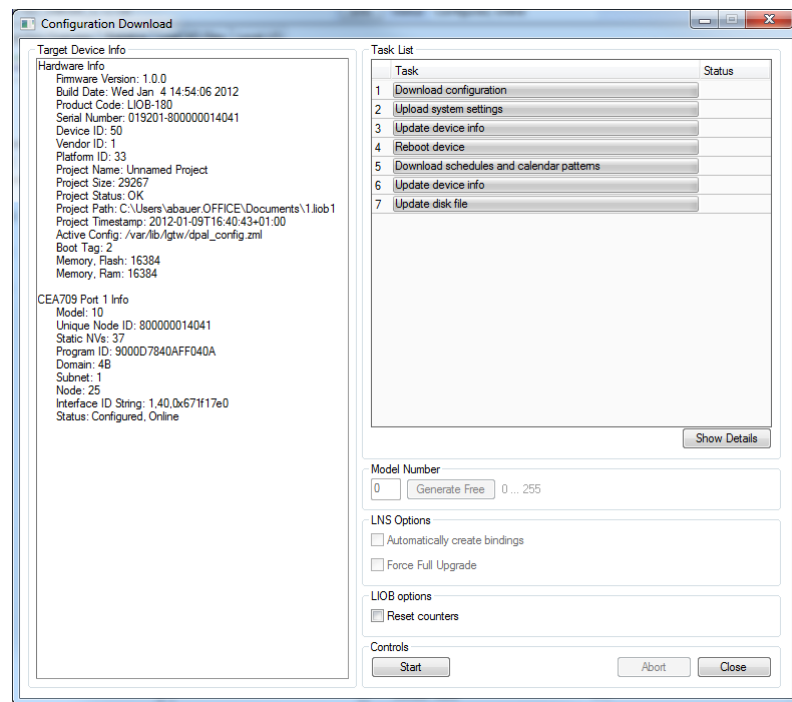


Figure 111: Configuration Download Dialog

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 process, last shutdown reason) or errors at run-time. This file is important for trouble-shooting and can 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 112 opens showing the upload progress.

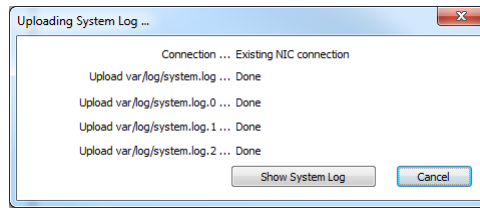


Figure 112: Upload system log dialog

- When the upload is finished, click on **Show System Log**. The system log window appears as shown in Figure 113. Observe that the LIOB-18x/48x/58x will not show correct time stamps unless time synchronization is setup.

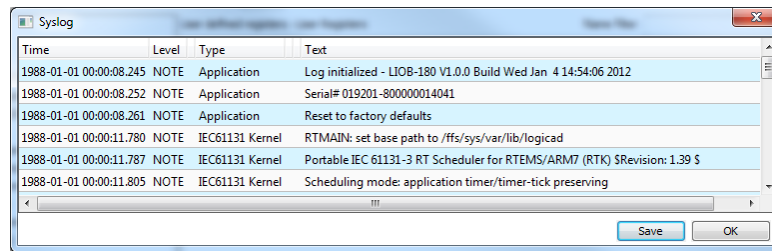


Figure 113: System log window

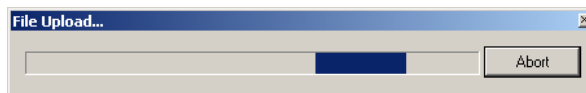
- Click on **Save** to store the system log into a file on your local hard drive.

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

- Connect to the device (see Section 7.6.1).
- Choose the menu **Tools → Backup Device Configuration ...**
- A file requestor opens. Choose a location to store the ZIP file of the device backup. The suggested file name contains the creation date.
- Click on **Save**. The backup is being uploaded.



To Restore a Backup

- Choose the menu **Tools → Restore Device Configuration ...**
- In the file requestor choose a backup ZIP file and click **Open**.
- The Configurator restores and reboots the device. The process is complete when the device has finished rebooting.

7.7 CEA-709 Configuration (LIOB-18x/48x)

7.7.1 Starting as an LNS Plug-In

In LonMaker the plug-in is started by right-clicking on the LIOB-18x/48x device shape or one of the functional blocks and selecting **Configure...** from the pop-up window.

In NL220, the Plug-in is started by right clicking on the LIOB-18x/48x 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 LIOB-18x/48x device and selecting the **LOYTEC LINX Configurator** in the **Start PlugIn** sub menu.

A window similar to what is shown in Figure 114 should appear.

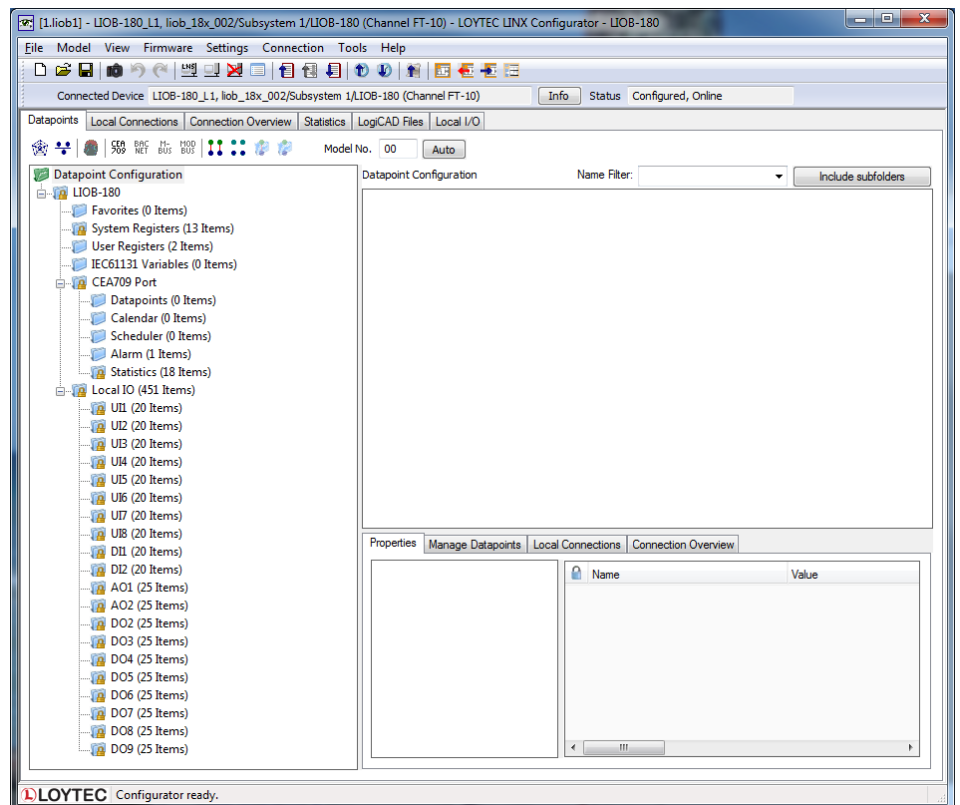


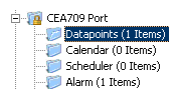
Figure 114: L-INX Configurator main window

7.7.2 Create Static NVs

The LOYTEC device can be configured to change its static interface and boot with a new one. Static NVs can be created manually in the CEA709 port folder.

To Create Static NVs

1. Select the **Datapoints** folder under the CEA709-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 115.

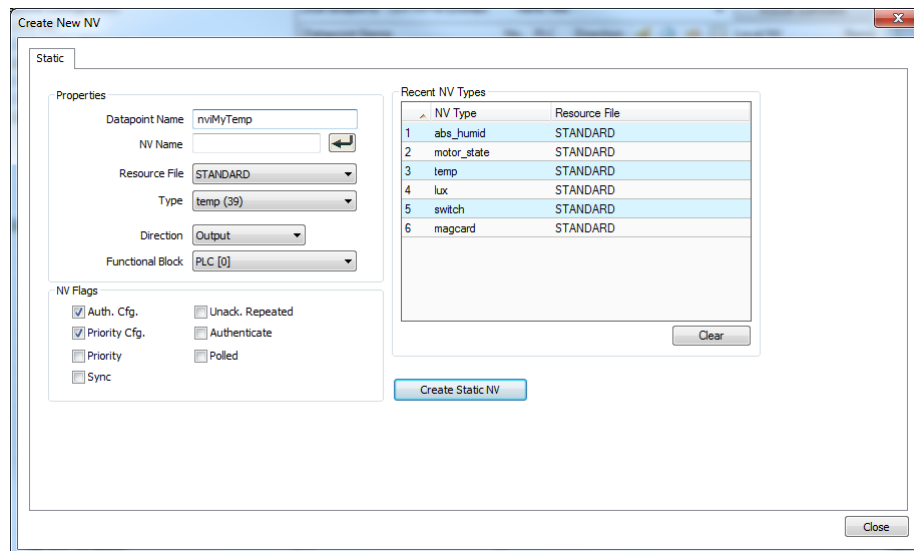


Figure 115: Create a static NV manually

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. Choose a functional block, where this static NV shall be located in.
7. Click **Create Static NV**. The static NV is created and appears in the data point list.
8. 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 6.8.2). 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.
9. Click **Close**.

7.7.3 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 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 6.8.2).

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 116.

2. If no bindings shall be generated, deselect the **Automatically create bindings** checkbox indicated by the red circle in Figure 116.
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 116.
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**.

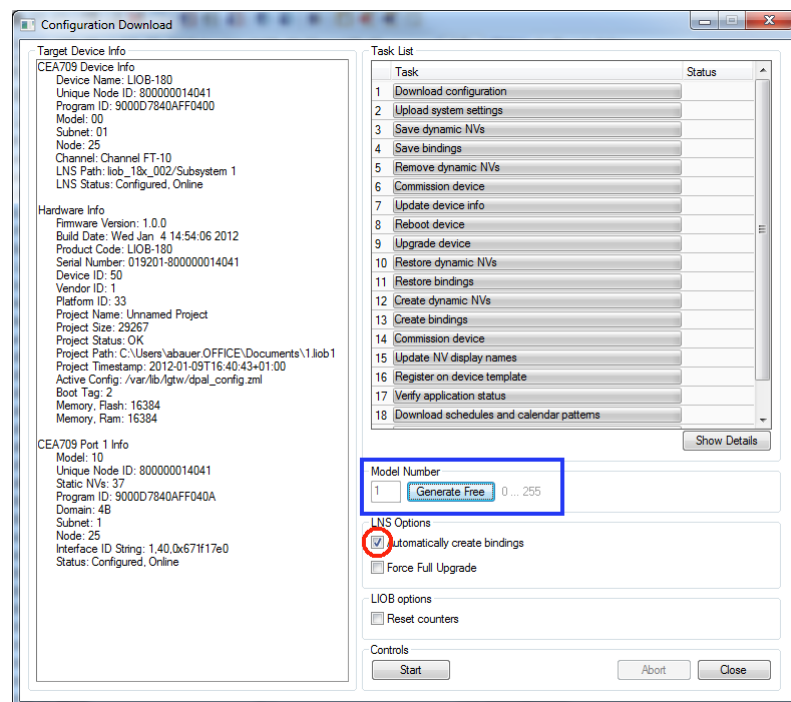


Figure 116: Configuration Download Dialog via LNS

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).

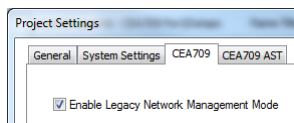
Normally, the Configurator software optimizes the download process by not executing certain LNS operations, if not necessary. For example, only those bindings are deleted and re-created, which correspond to real changes in the interface. The user can check the **Force Full Upgrade** option to clean and re-do all steps.

7.7.4 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**.



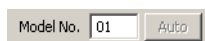
4. Click **OK**.
5. Download the configuration to activate the change.

7.7.5 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 "liob-18X_1.xif".

7.8 Advanced CEA-709 Configuration (LIOB-18x/48x)

7.8.1 Install Unconfigured Devices (LIOB-18x)

Like all CEA-709 devices, LIOB-18x devices must be installed by a network management tool (e.g., LNS-based tool) to be available for communication. If no network management tool is available, the LIOB-18x device can also be installed using the Configurator to get access to it. For LIOB-48x devices, this will usually not be necessary since the Configurator can always get access via TCP/IP.

To Install LIOB-18x Devices

1. Open the CEA-709 management dialog by clicking on the **Discover & Install** button of the dialog appearing after clicking the **Connect to device** speed button. The dialog CEA709 Management as shown in Figure 117 is shown.

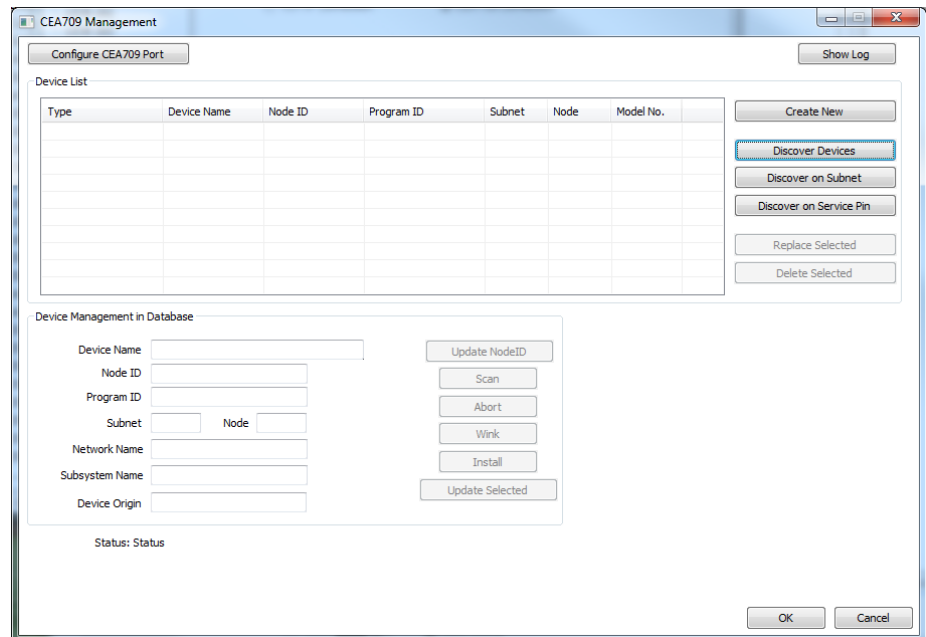


Figure 117: Discover & Install in CEA709 Management dialog

2. Discover devices by using one of the **Discover** buttons on the right side of the CEA-709 management dialog. Click **Done**.

Type	Device Name	Node ID	Program ID	Subnet	Node	Model No.
LIOB-18x (FT-10)		800000014041	9000D7840AFF040A	1	25	10

3. Select the device to install in the **Device List** of the CEA-709 management dialog.
4. Click the **Install** button. This opens the **Install Devices** dialog as shown in Figure 118.

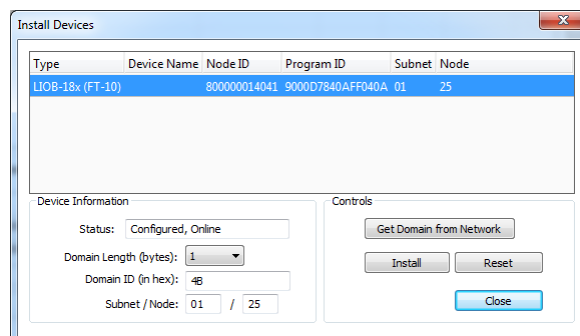


Figure 118: Install devices dialog

5. Select the device to be installed.
6. Enter the domain information or click **Get Domain from Network** and press a service pin.

7. Enter a subnet and node address and click **Install**. Click **Close**.
8. Check that the LIOB-18x device to connect to is selected in the **Device List** of the CEA-709 management dialog and click **OK**.
9. You can now connect to the device by clicking **Connect**.

7.8.2 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. 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.

To Create a Feedback Data Point

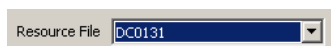
1. Select an output data point in the data point list of the **CEA-709 Port** folder, e.g. 'nvoHumid101'.
2. Right-click and choose **Create Feedback-Point** from the context menu.
3. A new input data point is created, having '_fb' appended to the original name, e.g., 'nvoHumid101_fb'. Note, that the 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.

7.8.3 Working with UNVTs

This device supports network variable types (UNVTs). 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, 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 7.7.2.
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.

7.8.4 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 ('PLC'). 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 6.8.2).

To Configure Function Blocks

1. Select the menu **Tools** → **Manage LonMark Objects ...** . The dialog Manage LonMark Objects as shown in Figure 119 appears.

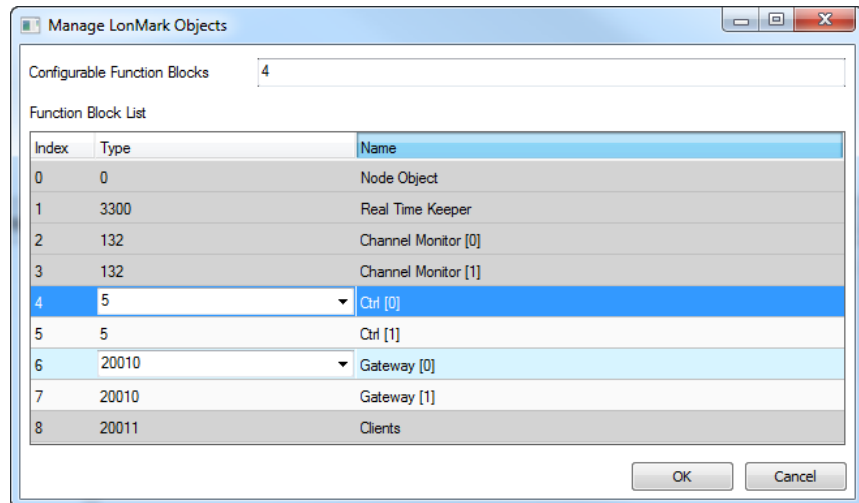


Figure 119: Manage LONMARK Objects

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 (LIOB-58x)

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**



- Right-click on that folder and select **Scan BACnet Network...**. This opens the BACnet Network Scan dialog as shown in Figure 120.

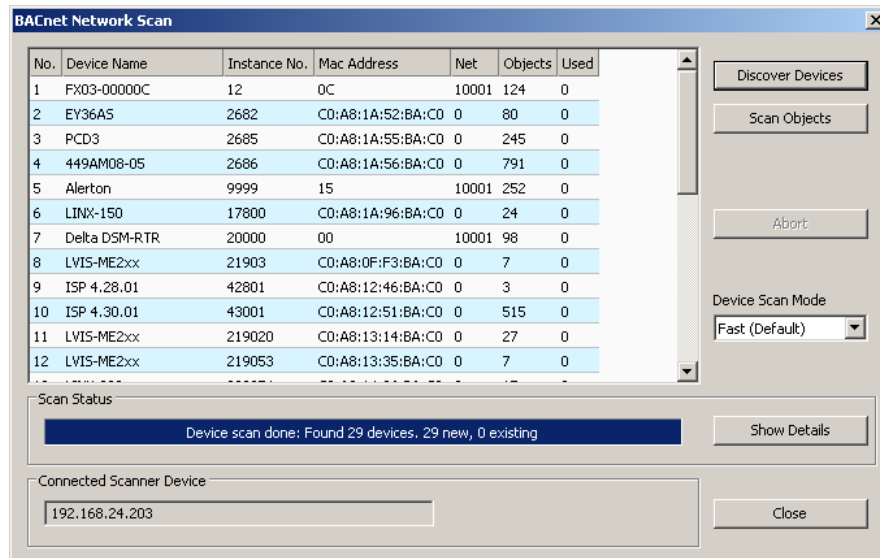


Figure 120: BACnet network scan dialog

- 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.
- 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.
- 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.
- 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 7.3.4).

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 10.3.11. 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.

Mapped Property	Present_Value
Client Map	LVIS-ME2xx (139), BI 1, Present_Value, Auto, Expiry 900 sec / Poll 60 sec

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.

- 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 121.

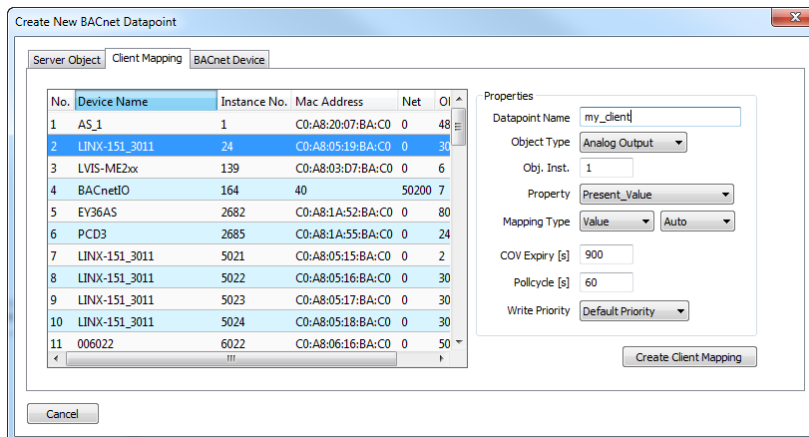


Figure 121: Create Client Mapping Dialog

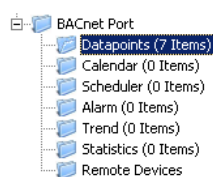
- Select the tab **Client Mapping**.
- Choose a target device in the list of known devices. Enter a **Data Point Name**, choose an **Object Type**, and edit the target object instance number. Then select the **Mapping Type**. For read client mappings edit the **COV expiry** or **Pollcycle** setting. For write client maps edit the **Write Priority**. For value client maps edit both. When finished click **Create Client Mapping**.

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

- Select the **Datapoints** folder under the **BACnet Port** folder.



- 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 122.

Figure 122: 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 **Engineering Units** for analog objects. 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.
5. 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 10.3.11.

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 123.

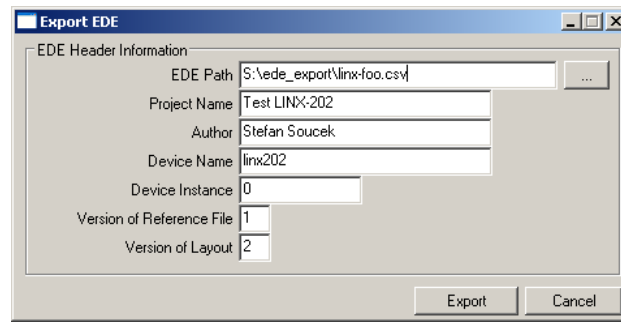



Figure 123: EDE Export Dialog

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**.

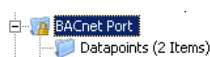
7.9.7 Import Server Objects from an EDE File

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 10.3.11. 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 124.

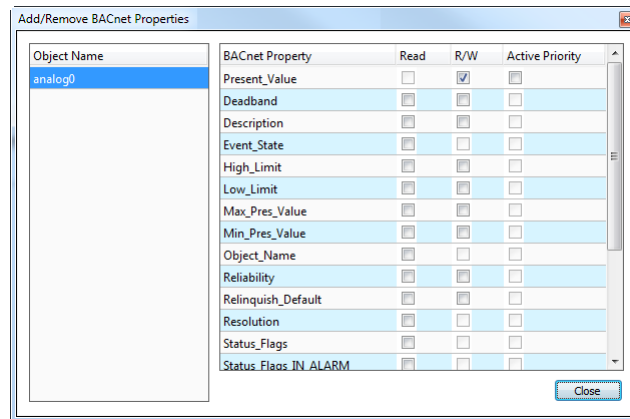


Figure 124: Dialog for adding/removing BACnet properties

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.

[-] AI1	1	<input type="checkbox"/>	<input type="checkbox"/>	Out	AI1	Analog Input
[-] High_Limit	1.1	<input type="checkbox"/>	<input type="checkbox"/>	Value	AI1	Analog Input
[-] Low_Limit	1.2	<input type="checkbox"/>	<input type="checkbox"/>	Value	AI1	Analog Input

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 7.3.4).
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.

- 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.
- For the Present_Value select the additional box Active Priority.



To Read the Active Priority of a Feedback Value

- Select a write client mapping.
- Right-click on the data point and select **Create Priority Feedback Point** from the context menu.
- 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 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 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 Create a Priority Write Data Point

- Select the commandable BACnet server object.
- Right-click and select **Create Priority Write Point...** from the context menu.
- A dialog pops up and prompts for the write priority. Note that write priority can also be changed later.
- The new priority write data point appears below the original BACnet server object data point.

Datapoint Name	No.	OPC	Param	PLC	Direction
AO1	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In
AO1_pri4_Write	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Out

7.10 Connections

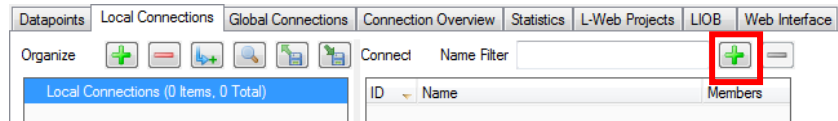
7.10.1 Create a New Connection


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 is distributed to all output data points. A status change of an output data point is propagated back to all input data points. All data points in the connection must be of a compatible type.


To manually create a new connection

1. Click on the **Local Connections** tab




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.

ID	Name	Members
1000	New Connection	0

2. 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.

3. Now the connection tab contains the new connection and on the right-hand side the list of data points in that connection as shown in Figure 125.

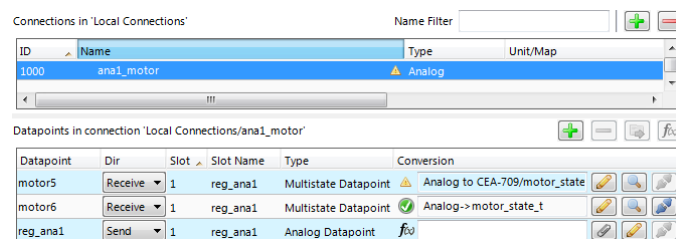


Figure 125: Connection tab with a connection and data points

4. 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.

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 126.
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 126.

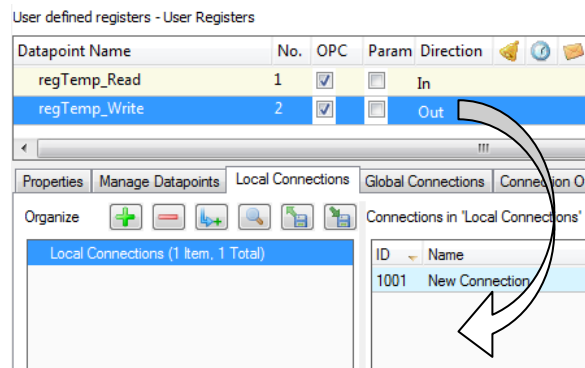


Figure 126: Create a connection with drag-and-drop

- To add data point to that connection, drop the new data point into the empty area in the connected data points list below as shown in Figure 126.

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.

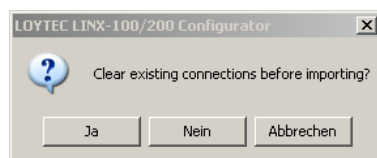
To Create Connection from a CSV File

- Select the menu **Tools** → **Export Local Connections ...**
- Select an appropriate file name and export.
- Edit the connections CSV file. An example is shown in Figure 127.

```
#connection_csv_ver,1
#ConnectionName,HubDPName,TargetDPName
Ai0,LIOB-480.CEA709 Port.ai0,LIOB-480.User Registers.abs_humid1
Ai1,LIOB-480.CEA709 Port.ai1,LIOB-480.User Registers.abs_humid2
Ai2,LIOB-480.CEA709 Port.ai2,LIOB-480.User Registers.abs_humid3
Ai3,LIOB-480.CEA709 Port.ai3,LIOB-480.User Registers.abs_humid4
```

Figure 127: Example Connection CSV File

- Select the menu **Tools** → **Import Local Connections ...**
- 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.




- Choose the file to import and click **Ok**.
- When the import has completed, optionally view the log to check, which connections have been added, modified, and deleted.

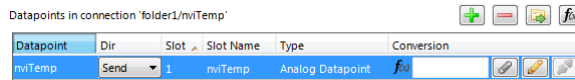
7.10.3 Modify Connections

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 128.
3. Select an existing connection.
4. Drag the selected data point and drop it into the empty area of the **Datapoints in connection** list area as shown in Figure 128. This adds the data point to the selected connection.

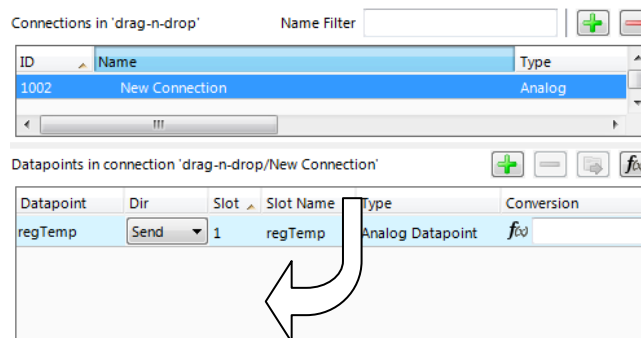

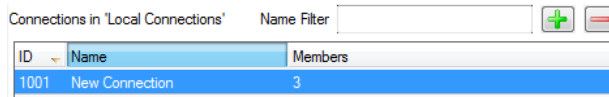


Figure 128: Modify connections in the properties view

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 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 129.

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 129: Connections Summary

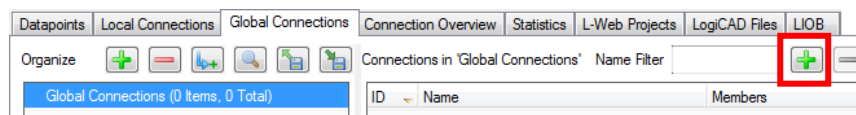
7.10.5 Create a Global Connection (LIOB-48x/58x)


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 6.5.2.

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




in the main window and press the speed button  **Create new Connection**. A new connection is added to the connection list.

2. 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.

ID	Name	Members
1000	outdoorTemp	1



- 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.
- Change the direction by modifying **send** or **receive**. For changing multiple data points use multi-select.

Datapoint	Dir	Type
temp_Write	Send	Analog Datapoint

- 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 .

Properties of connection

Name	Value	Description
Connection Name	outdoorTemp	Name of the c
Max. Send [s]	0	Send to netw
Min. Send [s]	0	Limit network

- To export the definitions of the created global connections, click the button  **Export connections to disk** and choose the XML format.
- 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.

7.11 E-mail Templates (LIOB-48x/58x)

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 5.2.12).

To Create an E-mail Template

- Under the **Global Objects** folder, select the **E-mail Configuration** sub-folder.



- Right-click and select **New E-mail Template ...** from the context menu.
- In the **Configure E-mail Template** dialog, which is shown in Figure 130 enter the **To** address and the **Subject**. Optionally, **Cc** and **Bcc** addresses can be specified.

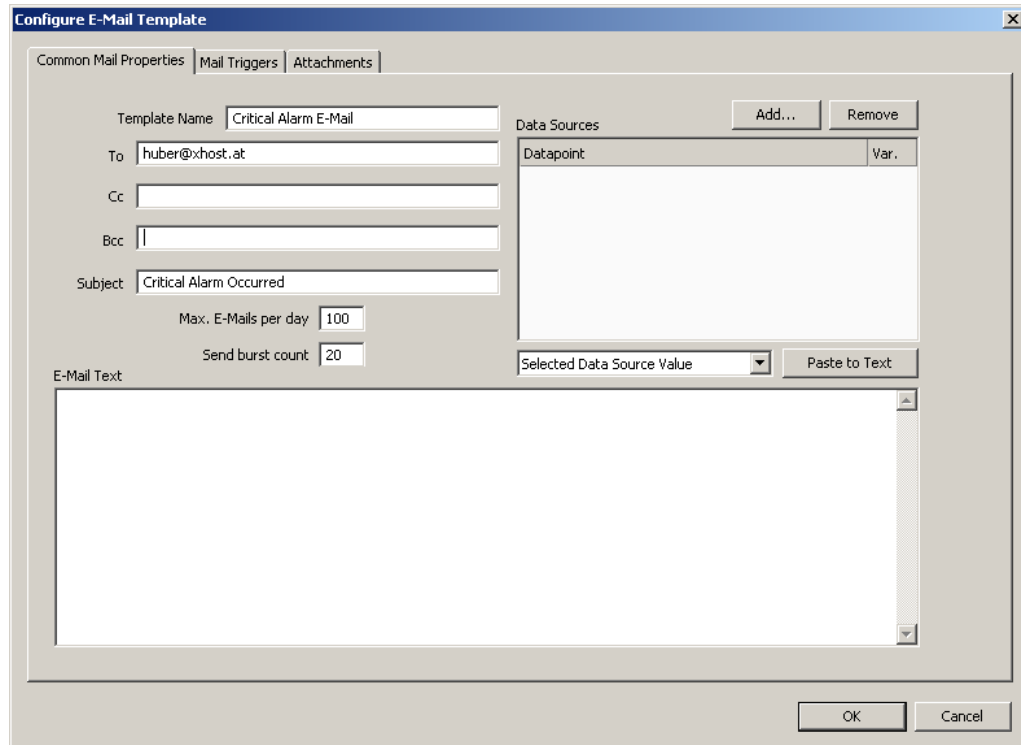
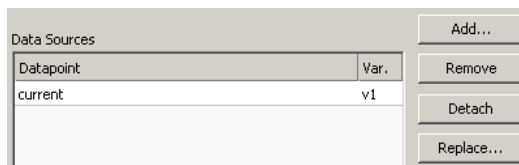


Figure 130: Configure E-Mail Template Dialog

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. 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.



8. A place holder `% {v1}` for the data point value appears now in the e-mail text.
9. 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

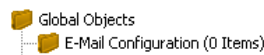
E-mail templates are used to assemble and transmit e-mails when certain trigger conditions occur. For an e-mail template, one or more trigger conditions can be defined. The e-mail will be sent, when one of the trigger conditions is activated. Depending of the trigger data point type, the trigger conditions 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 triggered.


To Create an E-mail Trigger

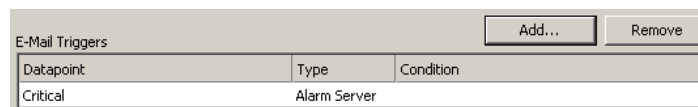
1. Under the **Global Objects** folder, select the **E-mail Configuration** sub-folder.



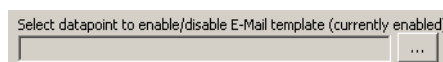
2. Right-click and select **Configure E-mail Template ...** from the context menu.
3. Change to the **Mail Triggers** tab.

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.

Attachment	Device File Path
system.log	/var/log/system.log

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.

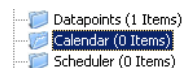
7.12 Local Schedule and Calendar

7.12.1 Create a local Calendar

As the first step, the required data points must be created. A calendar must be created, if the schedules shall work with exception days, such as “Holidays”. If it suffices for schedules to define daily schedules for normal weekdays only, no calendar needs to be created.

To Create a Local Calendar

1. Under the port folder, select the **Calendar** sub-folder to create a calendar.



2. Right-click in the data point list view and select **New local Calendar**
3. In the Create New Calendar dialog box (as shown in Figure 131) enter Name and Description of the calendar. Note, that a BACnet calendar does not have the **Effective Period** settings.

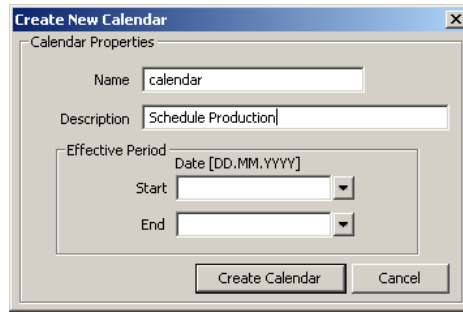


Figure 131: Create New Calendar dialog box

4. Click **Create Calendar**. The calendar appears now in the data point list view.

7.12.2 Create Calendar Pattern

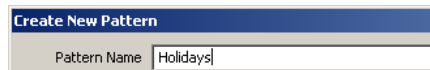
When a local calendar is used, it needs to be configured with calendar patterns. A calendar pattern represents a class of days such as “Holidays”. The calendar patterns can then be used in a schedule to define daily schedules for exception days. The available calendar patterns should be created when 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. Select an existing calendar data point.

No.	Direction	Calendar Name
1	In	calendar

2. Right-click and select **Create Calendar Pattern...**
3. Enter a Pattern Name in the **Create Calendar Pattern** dialog



4. Click **Create Pattern**. The dialog closes and the calendar pattern appears beneath the calendar data point.

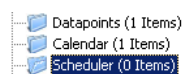
Local calendar objects							
	No.	Direction	Calendar Name	Index	Func. Block	Use	ID
	1	In	calendar			0	1030
	1.1		Holidays				1032

7.12.3 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 port folder, select the Scheduler sub-folder to create a scheduler.



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.

7.12.4 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 6.9.3).

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**.

Datapoint	Description	Location	Group	Default

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.*

- 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
NW_bac_lonCtrlInvo12_temp	temp	1	0.00	21.00	16.00

- 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
NW_bac_lonCtrlInvo12_temp	temp	1	0.00	21.00	16.00
NW_bac_lonCtrlInvo13_temp	temp	1	0.00	21.00	16.00

- 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.*

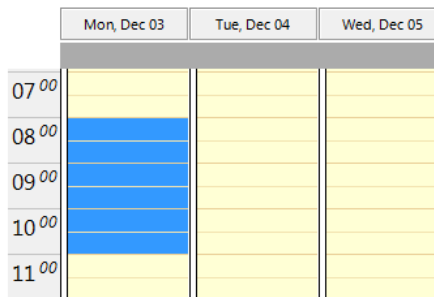
7.12.5 Configure Scheduled Events

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

- Open the **Configure Schedule** dialog and click on the **Configuration** tab (see Section 7.12.4).
- 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.



- 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.

Event

Name

Value Priority

Start End Duration

- Choose an **Event type**, which defines how the event is recurring.

Event type

One-time

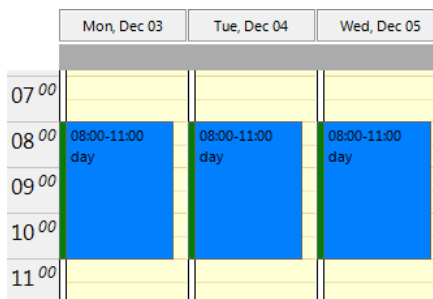
Daily

Weekly

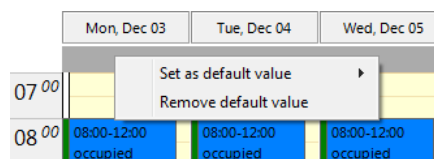
Start

End

- The click **OK**. The new event appears in the daily planner.



- 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.
- Right-click on an event in order to edit it. You can choose to modify it, change its color, or delete it.
- 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 6.8.3).

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.

7.12.6 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

No.	Direction	Calendar Name	Index	Func. Block	Use	ID
1	In	calendar			1	1030
1.1		Holidays				1032

3. Right-click and select **Configure Pattern ...** in the context menu.
4. The **Configure Pattern** dialog appears as shown in Figure 132. 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.

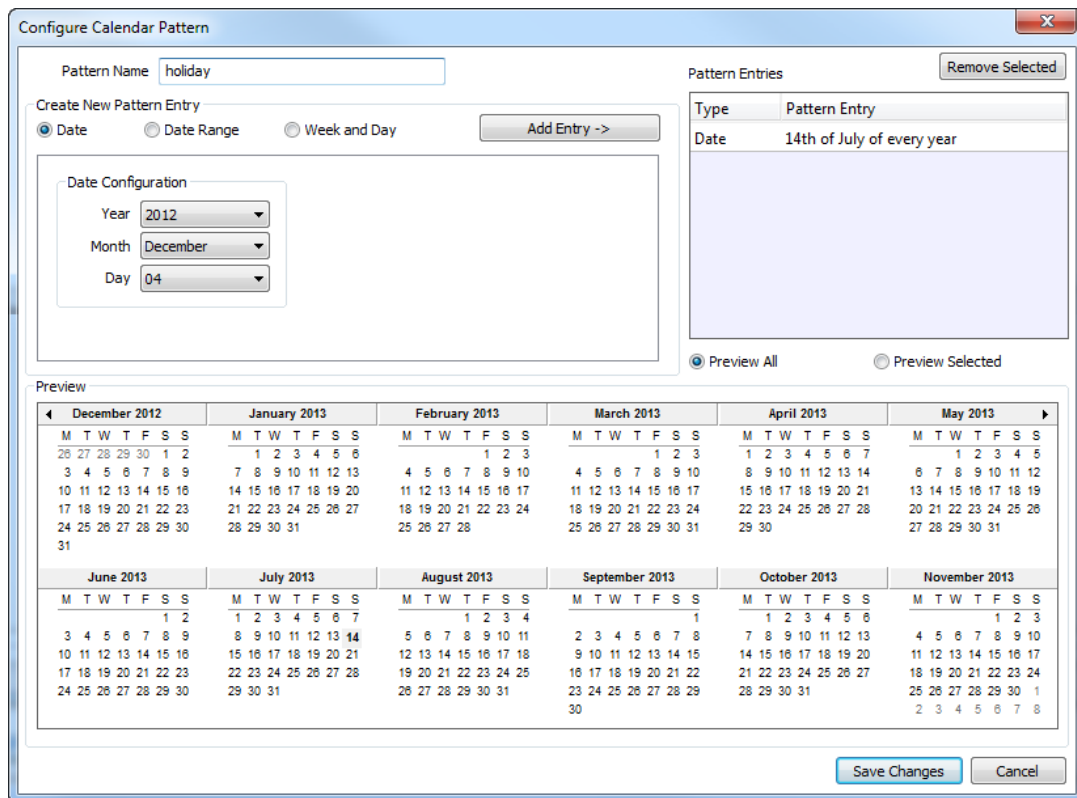


Figure 132: Configure Calendar Pattern Dialog

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**.*

7.12.7 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 7.12.4.
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.

7.12.8 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 7.12.4.
3. All three parts of the SNVT_tod_event are scheduled.

No.	OPC	Direction		Datapoint Name
1	<input checked="" type="checkbox"/>	Out		nvoTodEvent
1.1	<input checked="" type="checkbox"/>	Out		current_state
1.2	<input checked="" type="checkbox"/>	Out		next_state
1.3	<input checked="" type="checkbox"/>	Out		time_to_next_state

7.12.9 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.

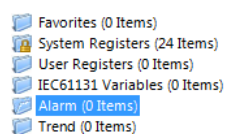
7.13 Local Alarming

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 ...**

- In the **Create New Alarm Server** dialog box (as shown in Figure 133) enter **Name** and **Description** of the alarm server.



Figure 133: Create New Alarm Server dialog box

- For alarm transitions you may define, which require acknowledgement and at which priority they are reported.
- You may attach data points for storing alarm counters. These will be linked using the respective property relations (see Section 6.3.11).
- 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.
- Click **Create**. The alarm server appears now in the data point list view.
- 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 134. 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 7.2.7). 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 7.2.7).
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.

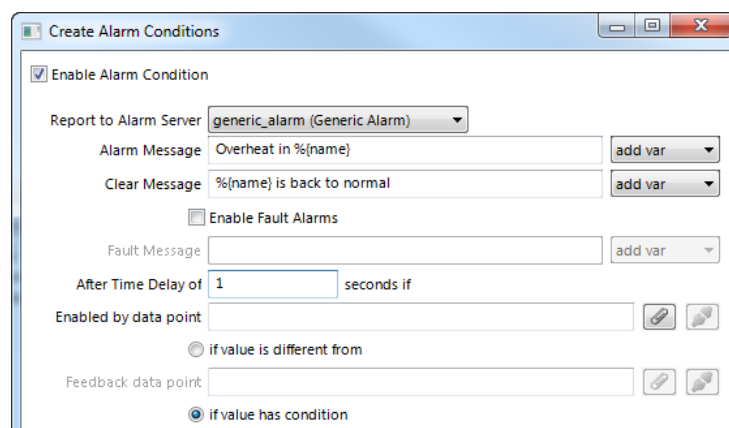


Figure 134: Common settings for an alarm condition

10. For an analog feedback condition fill in the alarm condition as shown in Figure 135. 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 7.2.7).

Figure 135: Condition for an Analog Feedback Alarm

- For an analog value condition fill in the alarm condition as shown in Figure 136. 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 7.2.7).

Figure 136: Alarm Condition for an Analog Value Condition

- For a binary data point define an alarm value in the alarm condition as shown in Figure 137. Select the **Alarm Value** which triggers the alarm.

Figure 137: Alarm Condition for a Binary Data Point

- For a multi-state data point define the alarm values in the alarm condition as shown in Figure 138. Select the states in the list **Not Alarmed** and move them to **Alarm on States** by clicking the arrow buttons.

Figure 138: Alarm Condition for a Multi-State Data Point

- 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 (LIQB-48x/58x)

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 5.2.8). 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.

Mail Triggers		
Datapoint	Type	Condition
Critical	Alarm	-

5. In the **Manage Trigger Conditions** list put a check mark on alarm conditions that shall invoke the transmission of the e-mail.

Manage Trigger Conditions	
Enabled Conditions	
<input checked="" type="checkbox"/>	State to Active
<input type="checkbox"/>	State to Ackd-Active
<input type="checkbox"/>	State to Ack-Pending
<input type="checkbox"/>	State to Ackd
<input type="checkbox"/>	State to Inactive
<input checked="" type="checkbox"/>	Offnormal-Alarms
<input checked="" type="checkbox"/>	Lowlimit-Alarms
<input checked="" type="checkbox"/>	Highlimit-Alarms
<input checked="" type="checkbox"/>	Fault-Alarms
<input checked="" type="checkbox"/>	Buffer-Alarms

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.

Data Sources	
Datapoint	Var.
LINX-120.CEA709 Port.Alarm.Critical	v1

Alarm value to subject Paste

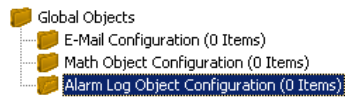
7.13.4 Create an Alarm Log (LIQB-48x/58x)

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.

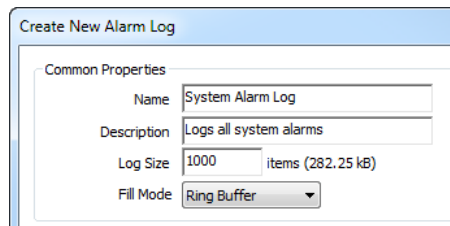
An alarm log can log transitions of one or more alarm objects. Its size is configurable. The alarm log is a ring buffer. As soon as its size limit is reached, the oldest alarm log records are overwritten by new alarm transitions.

To Create an Alarm Log

1. Under the **Global Objects** folder, select the **Alarm Log Object Configuration** sub-folder.



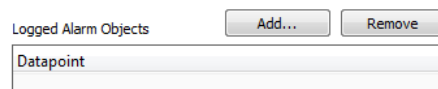
- In the data point list right-click and select **New Alarm Log ...** from the context menu.



- In the **New Alarm Log** dialog enter a **Name** for the alarm log. Optionally enter a **Description**.
- Enter a **Log Size**, which defines how many transitions are resident in the alarm log.
- 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.
- Define a percentage for **Fill Level Notification**, which can be used to trigger the transmission of E-Mails.




- Click on the button **Add...** on top of the **Logged Alarm Objects** list.



- 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.
- 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 7.2.7).
- 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 139. 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.

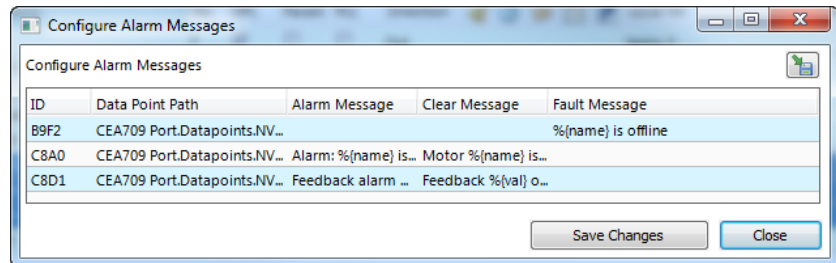


Figure 139: Spreadsheet for multi-edit of alarm messages

7.14 Local Trending (LIOB-48x/58x)

7.14.1 Create a Local Trend

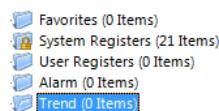
The value of a data point can be logged over time. This is referred to as trend data. To generate trend data a trend object has to be created. The trend data is stored in a data logger file. This file can be downloaded via FTP in binary or CSV format (see Section 10.1.2).

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 140) enter a name and optionally a description for the trend log object.

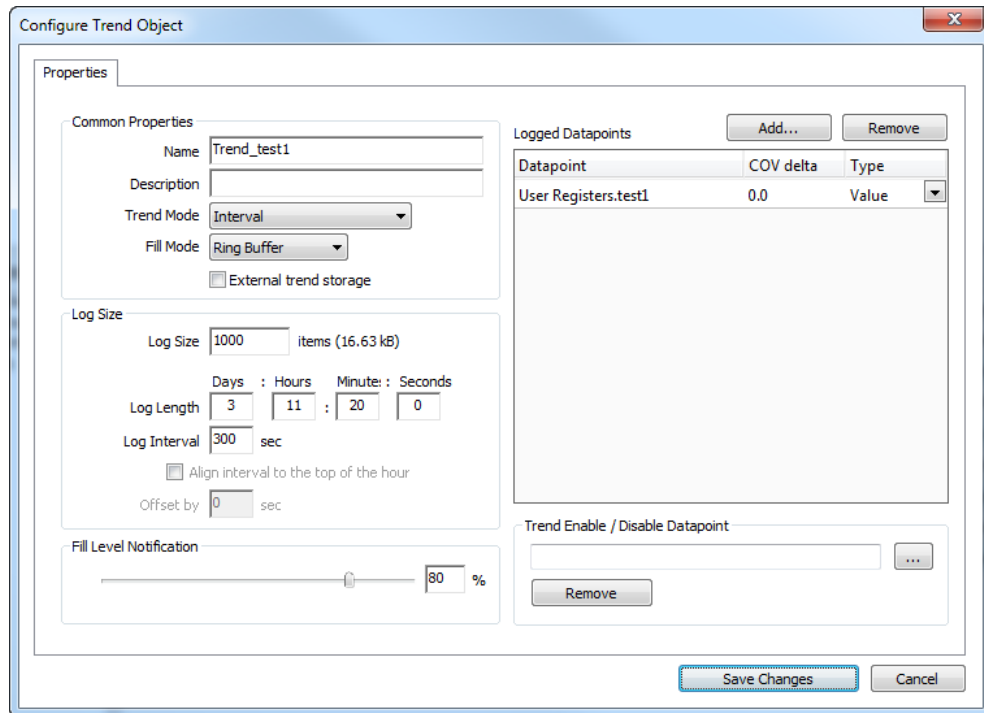


Figure 140: Basic Trend Object Configuration

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.

No.	Direction	Trend Name	Use	ID
1	Out	TestTrend	0	1014

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.

Logged Datapoints		
Datapoint	COV delta	Type
abs_humid1	0.0	Value

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!

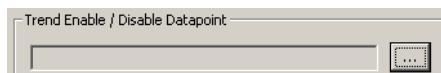
Data points can also be attached to a trend by selecting a data point in the data point manager, drag it onto a trend object and drop it on the trend object.

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.

No.	Direction	Trend Name	Use	ID
1	Out	TestTrend	0	1014

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.

Trend Triggers			Add...	Remove
Datapoint	Type	Condition		
state	Value Update	-		

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 10.1.2). 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 141. This would result in the trend CSV file `'/data/trend/out_temp_107C.csv'`.

No.△	Direction	Trend Name	Object Name	Obj Type	Instance	Alloc	Use	ID
1	Out	out_temp	out_temp	Trend Object	26	50	0	107C

Figure 141: 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 5.2.12). 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.

E-Mail Triggers		
Datapoint	Type	Condition
TestTrend		Fill Level

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.

Attachments			
Attach File		TestTrend_1014.csv	<input type="button" value="Add"/>
Attachment	Device File Path	Add Datetime	<input type="button" value="Remove"/>
TestTrend_1014.csv	/tmp/uid/trend/1014.csv	<input checked="" type="checkbox"/>	

7. Click **OK** to complete the e-mail template configuration.

7.14.6 Technology Trends (LIOB-58x)

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 6.9.4).

7.15 Math Objects (LIOB-48x/58x)

7.15.1 Create a Math Object

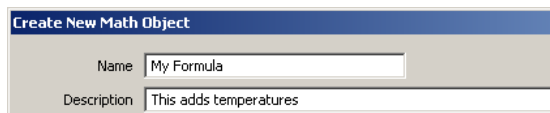
Math objects are advanced application objects that can execute mathematical operations on data points. A math object takes a number of input data points (variables v_1, v_2, \dots, v_n) and calculates a result value according to a specified formula. When configuring a math object, the input data points, 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.



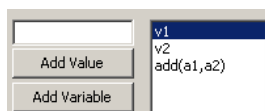
2. Right-click and select **New Math Object ...** from the context menu.
3. In the Create New Math Object dialog, enter a name and optionally a description for the math object.



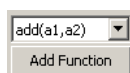
4. Attach input data points by clicking the **Add Input DP** button.

Var.	Input Datapoint	Datapoint Path	COV delta
v1	nviTemp1	LINX-100.CEA709 Port.Datapoints	0.0
v2	nviTemp2	LINX-100.CEA709 Port.Datapoints	0.0

5. In the data point selector dialog, select the input data points and click **OK**. The data points appear as v_1, v_2 , etc.
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.
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. The resulting formula is displayed at the bottom of the dialog. Alternatively, the formula can be entered there.

Output =

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.

Output Datapoint	Datapoint Path
my_register_Write	LINX-100.User Registers

11. In the data point selector dialog select the output data points and click **OK**.
12. To create the math object click **Create**.

7.15.2 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.

Math Objects				
Datapoint Name Filter: <input type="text"/>				
No.	Direction	Name	Description	ID
1		My Formula	This adds temperatures	1000



3. Right-click and select **Configure Math Object ...** from the context menu.
4. Edit the math object as described in Section 7.15.1.
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**.


7.16 Historic Filters

7.16.1 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 6.6.6.

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 entry 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'.

No.	Name	Type	Day	Time	samples ago	Delta to current
0	midnight_today	Value at hh:mm:ss of the day	N/A	00:00:00 h	0	<input type="checkbox"/>




6. To duplicate an entry click on the duplicate button . On the duplicate modify the settings accordingly.

No.	Name	Type	Day	Time	samples ago	Delta to current
0	midnight_today	Value at hh:mm:ss of the day	N/A	00:00:00 h	0	<input type="checkbox"/>
1	midnight_yesterday	Value at hh:mm:ss of the day	N/A	00:00:00 h	1	<input type="checkbox"/>

7. For getting the difference of an historic value to the current value check the box **Delta to current**.





No.	Name	Type	Day	Time	samples ago	Delta to current
0	midnight_to_now	Value at hh:mm:ss of the day	N/A	00:00:00 h	0	<input checked="" type="checkbox"/>

8. Click on **Save Changes** and select the created filter. For each filter entry defined, a historicFilter property relation is created under the base data point(s).

energyCount	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Value	
historicFilter (midnight_to_now)	1.1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In	
historicFilter (midnight_today)	1.2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In	
historicFilter (midnight_yesterday)	1.3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In	

7.16.2 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.


7.17 I/Os

The LIOB-18x/48x/58x models are equipped with local Inputs and Outputs to directly connect to sensors and actuators. Additionally, a LIOB-48x/58x device can be extended with the I/Os of one LIOB-45x/55x device. The following sections describe how to configure these I/Os.

7.17.1 Configure I/Os

At first the I/Os have a default configuration. To adapt the I/Os to the specific needs in a project they can be freely configured. Note that certain changes to the I/O configuration must be committed before proceeding with other tasks. This can be done explicitly by clicking **Commit member changes** . Otherwise, the user will be requested to commit by the Configurator. Changes can be canceled by clicking **Cancel member changes** .

To Configure I/Os

- For LIOB-18x models, select the **Local I/O** tab. For LIOB-48x/58x models, select the **L-IOB** tab and then **LIOB-LOCAL**. For adding a LIOB-45x/55x device to a LIOB-48x/58x model, select **LIOB-IP** in the **L-IOB** tab and use the **Add Device(s)** button .
- The **L-IOB** tab displays the I/Os available on the device in the **Inputs / Outputs** list.

Inputs / Outputs					
Nr	TerminalNr	Terminal	Name	Hardware type	DP count
1	1	UI1	UI1	IN Analog/Digital 10V	22
2	2	GND12	GND UI1-UI2	IN Analog/Digital 10V	
3	3	UI2	UI2	IN Analog/Digital 10V	22
4	4	UI3	UI3	IN Analog/Digital 10V	22

- To adapt the I/O name, double-click the name in the **Name** column and edit it, e.g., 'RoomTemp'.

Nr	TerminalNr	Terminal	Name	Hardware type
1	1	UI1	RoomTemp	IN Analog/Digital




- Select (or multi-select) an I/O in the **Inputs/Outputs** list and observe the **Object parameters** list below. These parameters can be used to configure the I/O.

Object parameters									
Nr	DP Create	OPC	PLC In	PLC Out	Parameter name	Parameter value	Unit	Range	Description
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Name	RoomTemp			Terminal name
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	HardwareType	IN Analog/Digital			Terminal type
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SignalType	Voltage 0-10V			Type of the input/output signal

- For example, change the **SignalType** of the input UI1 to 'Resistance'.

- Note that the object configuration properties may change when changing the signal type or interpretation. For detailed information on available configuration properties and their dependencies refer to Section 6.7. In this example you may now edit the **Interpretation** to 'NTC10K', which will allow connecting an NTC temperature sensor to the input.






SignalType	Resistance
Interpretation	NTC10K


- Observe that the **Object parameters** table lists all I/O configuration properties and also life values that can be exposed as L-IOB data points. These L-IOB data points in turn can be exposed as OPC data points (LIOB-48x/58x) as well as input and/or output data points in the logic program. Only the basic data points of each I/O (input, output, feedback) are created by default. Using the checkboxes in the **DP Create** column of the **Objects parameters** table, additional configuration properties and life values can be exposed as L-IOB data points. Using the checkboxes in the **OPC**, **PLC In**, and **PLC Out** columns, it is possible to select which of the created data points shall be exposed as OPC data points, logic program input data points, and logic program output data points.
- For LIOB-58x devices, native BACnet objects (see Section 6.9.7) can be created using the **BACnet Object** checkbox. For inputs with interpretation "Pulse Count", it is possible to choose between the **BACnet Object Type** "Analog Input" and "Accumulator". The corresponding data points will appear in the **LIOB-58X / BACnet Port / Datapoints** folder of the **Datapoints** tab. The BACnet server object names and descriptions will initially receive unique I/O names as described in Section 6.9.7. They can be set to the current I/O names and descriptions by using the button **Set BACnet Names** .
- I/Os of the same hardware type can be copied using the **Copy selected object**  and **Paste to selected object(s)**  buttons.

7.17.2 Manage I/O Configurations

I/O configurations can be managed by the user. E.g. a new template may be generated from the I/O configuration, which can be re-used in other projects.

To Manage I/O Configurations

- Select the **L-IOB** tab.
- To export a new template for a L-IOB configuration click the **Export device template** button . In the file save dialog, enter a new L-IOB template file name and store it. This file can then be used in a different project using the **Import Template** button .
- Existing configurations can be updated with new L-IOB templates by using the **Upgrade Template** button . This can be used e.g. to migrate new functions which were not yet available in earlier template versions. It is highly recommended to backup the original project before upgrading templates. After upgrading the template, a log is displayed, which shows all information about the upgrade process. Please carefully review this log before using the updated configuration.
- Another way of managing I/O configurations is using terminal configurations. A terminal configuration can be created from an I/O object in the **Inputs / Outputs** list by using the **Export object to disc** button  next to the list. In this case, only the configuration of this I/O is stored in an XML file. Terminal configurations can be imported (to one or multiple I/Os of the same hardware type) by using the **Import object from disc** button .

5. These terminal configurations can also be used to configure all I/Os of the L-IOB device using a CSV file. Please see the corresponding “Manage L-IOB Configurations” section in the LIOB-10x/x5x User Manual [2] for more information.
6. To setup translation tables for certain I/O interpretation modes, click the **Edit translation tables** button .

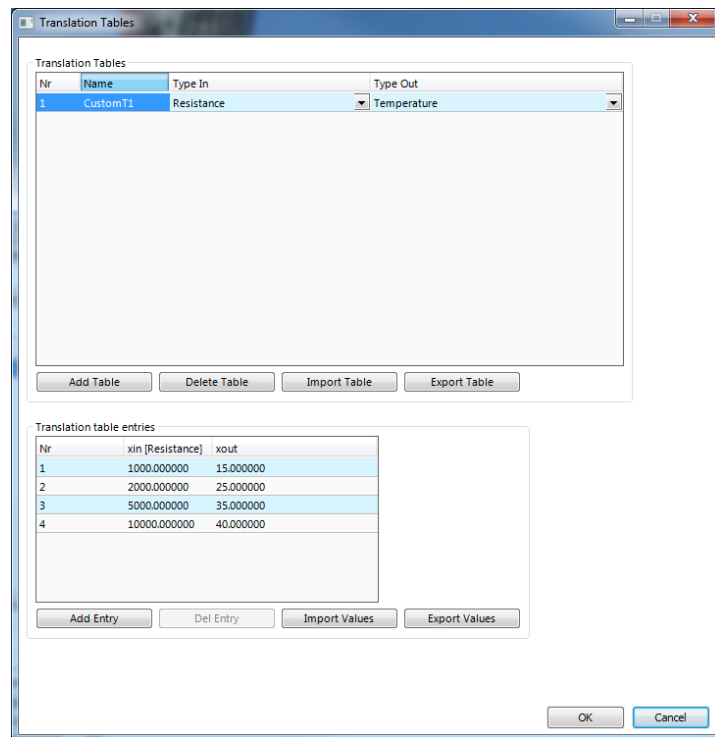


Figure 142: L-IOB Translation Tables

7. You can add and delete tables using the buttons **Add table** and **Delete table**. The data types for the **xin** and **xout** columns of the table can be setup in **Type In** and **Type Out**. In the **Translation table entries** list, you can enter the **xin** and **xout** values. The translation is done (with linear interpolation) from measured **xin** values to **yout** values which are used for the L-IOB data points. Further, the table values can be exported and imported (CSV files) with **Import Values** and **Export Values**. Entire tables (including table name and type) can be exported and imported with **Import Table** and **Export Table**. Tables of some standard temperature sensors are installed with the Configurator in the subfolder “LIOB\tables”.

7.17.3 Using I/O Data Points

The I/Os are exposed to data points on the device. These data points comprise the present value of the I/O as well as status and configuration data. The corresponding folders in the data point manager tab of the Configurator are named **Local IO** and **LIOB-IP**, see Figure 143.

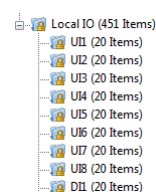


Figure 143: IO folder in the data point manager

Under this folder there exist sub-folders for each I/O on the controller. When selecting any of the folders, the data point list shows all data points in the scope of that folder. The data points on the **Local IO / LIOB-IP** folders provide general I/O data. The data points of the I/O sub-folders expose the present value as well as status and configuration data specific to the I/O. Refer to Section 6.7.3 for more information on I/O data points.

The I/O data points can be used in the configuration just as all other data points. To expose an I/O directly as a variable to the PLC select the **PLC** check box. Note that I/O data points cannot be directly subject to alarming. For doing so, native data points (CEA709) that support alarm conditions must be created and put in a connection with the I/O data point.

7.17.4 Printing Labels

For documentation purpose, I/O labels can be printed and glued to the top and bottom of the L-IOB enclosure. It is recommended to use **Avery L6031REV** labels for printing.

To Print the L-IOB labels

1. Start the Configurator and open the LIOB-18x/48x/58x project to print labels for.
2. Select menu **File / Print**.
3. This opens the Print View dialog as shown in Figure 144. Select **Export LIOB Labels** on the left side.

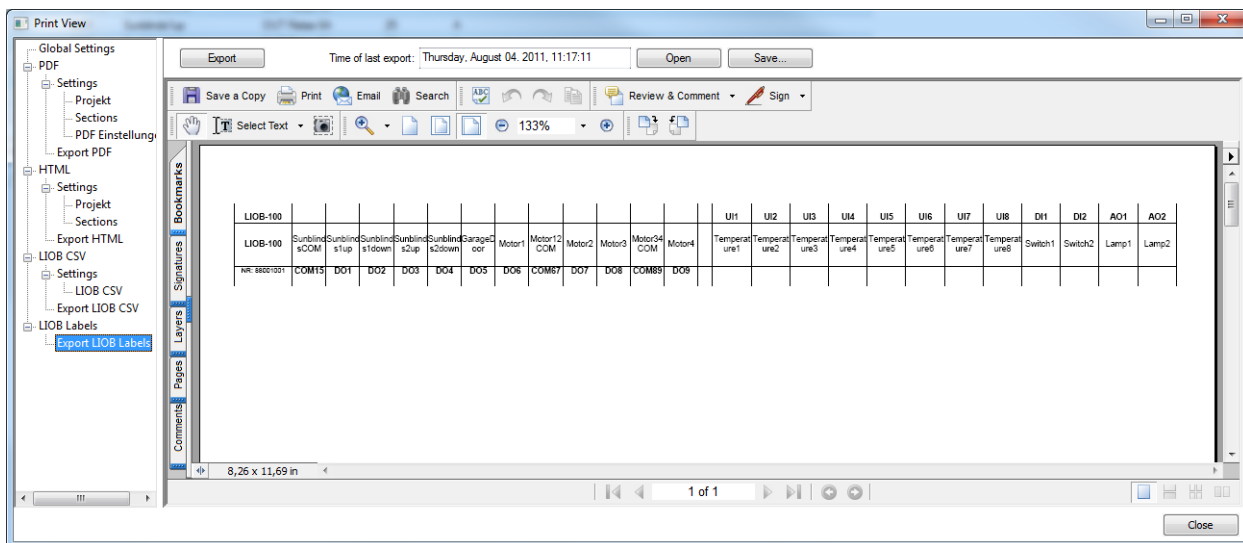


Figure 144: Print View

4. Click on **Export** to generate a PDF file of the labels.
5. Click on **Open** to open the file in a separate PDF reader window. Click on **Save** to save the PDF file.
6. Print the labels directly from the PDF reader embedded in the print view or from the opened / saved PDF file.

8 OPC Server (LIOB-48x/58x)

8.1 XML-DA OPC Server

8.1.1 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 7.3.5) or in the L-Config tool (see NIC User Manual [1]).

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.

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 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 8.2.

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 145.

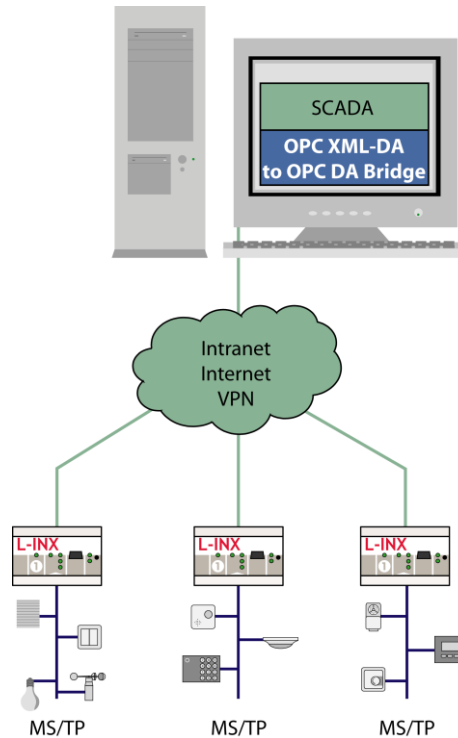


Figure 145: Using a XML-DA/DCOM bridge

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 OPCBR-800 can be purchased and installed on any PC. This is discussed in Section 8.3.

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 8.4 for more information.

8.1.2 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 146.

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*.

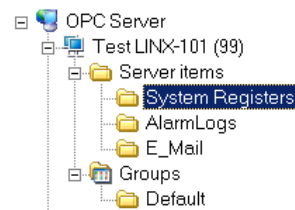


Figure 146: Client browsing the OPC tag namespace

8.1.2.1 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.

8.1.2.2 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.

8.1.2.3 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.

8.1.2.4 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.

8.1.2.5 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.

8.1.2.6 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.

8.1.3 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.

8.1.3.1 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.
- **CallItemPath** (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 8.1.3.2.

8.1.3.2 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.

8.1.3.3 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.

8.1.3.4 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 serve as placeholders for the central data logger to store its URI. The tag CentralDL0 is intended for the primary, CentralDL1 for the secondary central data logger. The tags are stores in non-volatile memory and retain their values over a power-on reset. For more information on the central data logger refer to the LWEB-801/900 User Manual.

8.1.3.5 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.

8.1.4 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 6.3.2).

8.2 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 Servers on a Windows PC.

The L-WEB software uses the standardized OPC XML-DA Web service to communicate between L-WEB and remote 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 LIOB-48x/58x devices 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.

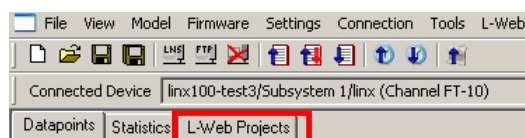
Starting from the data point configuration, the user can create an L-WEB project. The L-WEB project contains the data point configuration of the Web service interface and a graphical design for the L-WEB user interface. For more information on creating graphical designs using the L-VIS/L-WEB Configurator software refer to the L-VIS User Manual.

8.2.1 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 147.

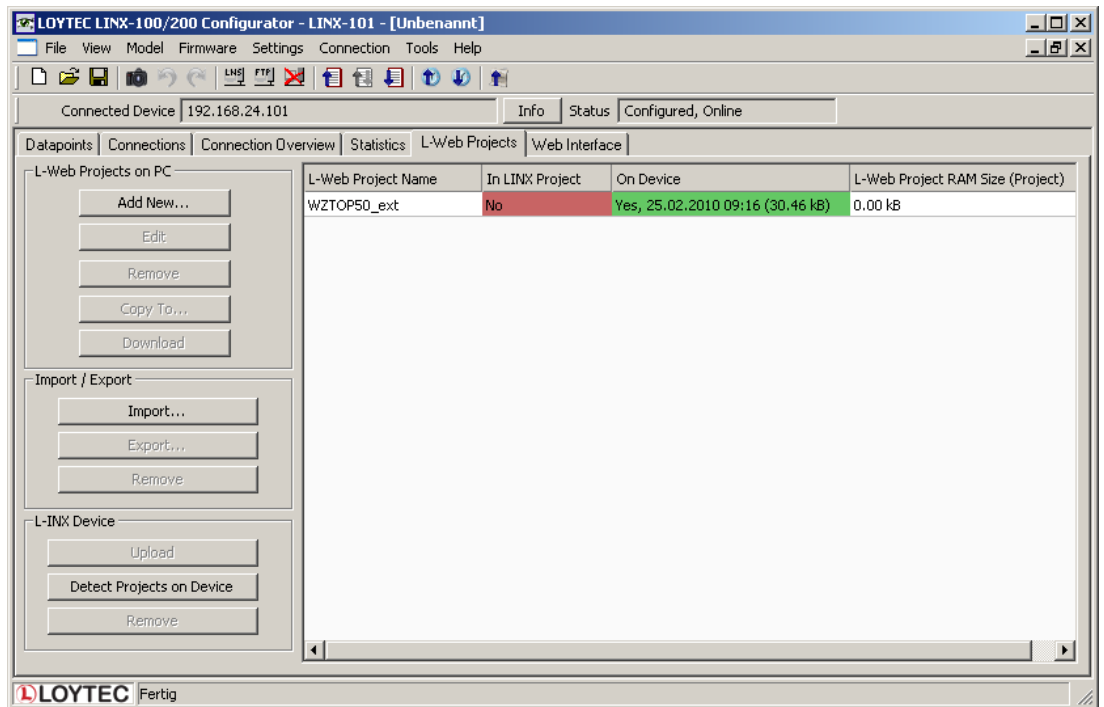
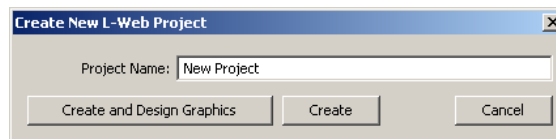


Figure 147: L-WEB Projects Tab

3. Click on **Add New ...**
4. Enter a new **Project Name**.



5. Click on **Create**. The new project appears in the projects list.

L-Web Project Name	In LINX Project	On Device	L-Web Project RAM Size (Project)
New Project	No	No	0.00 kB

8.2.2 Start a Graphical L-WEB Design

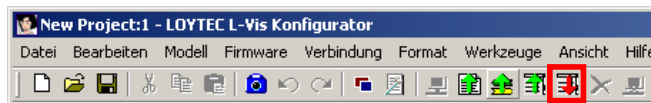
The L-WEB graphical design tool is started from within the L-WEB projects tab. The graphical design for the L-WEB project is created in the L-VIS design tool (L-VIS/L-WEB Configurator). The data point configuration created in the Configurator project is available for the L-WEB project and its graphical design.

To Start a Graphical Design

1. Select the **L-WEB Projects** tab.
2. Select an L-WEB project.
3. Click **Edit**.



- This opens the L-VIS graphical design tool. Complete the graphical design in the tool and click the **Write Project to Device** speed button



- 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.

8.2.3 Organize L-WEB Projects

L-WEB projects can be organized within the 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

- Connect to the device as described in Section 7.6.1.
- Select the **L-WEB Projects** tab.
- 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 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.

- 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

- 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

- 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

- 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.
- 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.

8.3 Using the OPC Bridge

8.3.1 Software Installation

The LOYTEC OPC Bridge software LOPC-BR800 is installed as a separate application on a PC. A license for the LOYTEC OPC Bridge software must be purchased separately. With one software license for the OPC Bridge, multiple devices can be accessed. It is not recommended to operate more than 5 devices with one bridge license. With the OPC Bridge software installed, each configured LOYTEC device appears as a separate COM/DCOM server. The OPC Bridge software can be used with LOYTEC OPC servers only. The bridge won't connect to third-party OPC servers.

System requirements:

- Windows XP, Windows 2003 Server, Windows Vista, Windows 7, Windows 2008 Server.

The OPC Bridge software is available via download from the LOYTEC Web site www.loytec.com. For installing the software, a registration code must be purchased.

To Install the OPC Bridge

- Double click on 'lopc-br800_x_y_setup.exe' and follow the installation steps.
- When asked, type in the bridge's registration code. Click **Next**.
- Finalize all remaining installation steps by clicking **Next** and **Finish**.

When the installation is complete, the OPC Bridge software is available under **Programs** → **LOYTEC OPC Bridge**.

8.3.2 Configure the Bridge Locally

If the bridge software is installed on the same PC where the Configurator software is used to configure the OPC server, the server information can be automatically made available as a COM OPC server. This is done when the configuration is downloaded into the device.

To Configure the Bridge Locally

- Open the Configurator software and configure the device as described in Section 7.6.
- In the Configurator menu go to **Settings** → **Project settings** This opens the project settings dialog on the tab **General** as shown in Figure 148.

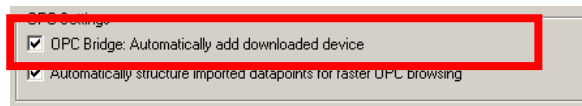


Figure 148: Enable bridge export in the project settings

3. Put a check mark on **OPC bridge: Automatically add downloaded device**.
4. Click **OK**.
5. Downloading the configuration also makes the OPC server on the device available as a COM OPC server through the local bridge.

8.3.3 Export OPC Servers for another PC

If the bridge software is not installed locally on the same PC where the device configuration is created, it must be exported to make the OPC server information available. The exported file can then be transferred to the OPC bridge.

To Export the Bridge Configuration

1. Open the Configurator.
2. Connect to the device, which shall be available in the bridge.
3. Select the menu **File → Export to OPC Bridge Configuration...**
4. In the **OPC Bridge Device Properties** dialog as shown in Figure 149, add information which is not default, i.e., Min update rate, Wait time and Hold Time (see Section 8.3.6). If the operator user has a non-default password, enter the password.
5. Click **Export to File**.

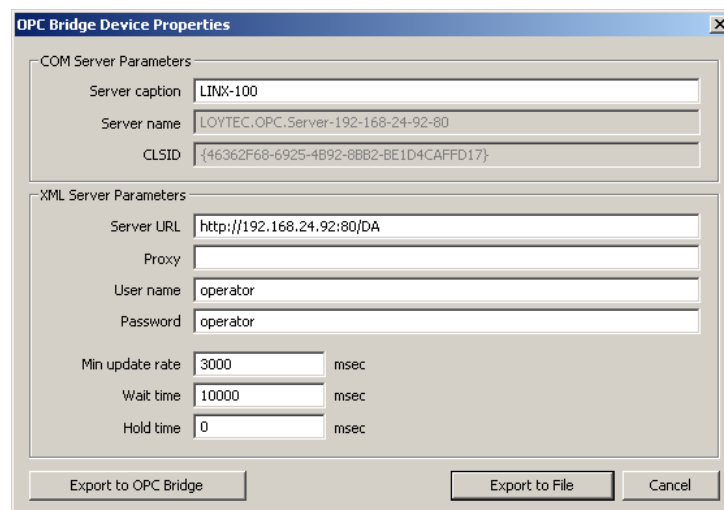


Figure 149: Bridge Export dialog

8.3.4 Import OPC Servers Using the Configurator

When using OPC server information exported by the Configurator, the exported server definition must be imported to the OPC bridge. This can also be done using the Configurator software. The Configurator software must be installed on the same PC as the OPC bridge for doing so.


To Import a Server Definition

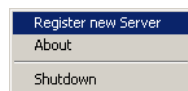
1. Open the Configurator software.
2. Select the menu **File → Import OPC Bridge Configuration...**
3. In the file requestor select the bridge configuration XML file, which has previously been exported and click **OK**.
4. The **OPC Bridge Device Properties** dialog displays the imported bridge information.
5. Click **Export to Bridge** to add the respective COM server to the bridge.
6. Click **Save**.

8.3.5 Manually Configure Servers

The OPC bridge configuration can also be edited manually. The same procedure is also applicable to verify imported OPC server definitions in the bridge software. After adding a server definition in the bridge software, the device's OPC server will be available as a COM OPC server through the bridge.

To Configure a Server in the Bridge

1. Start the OPC bridge from the Windows Start menu under Programs → LOYTEC OPC Bridge → OPC Bridge manual Setup.
2. In the system tray, right-click on the bridge icon .
3. In the context menu select **Register new Server**.



4. In the Register Server dialog click **Add**.
5. A new server entry is added. Enter the information on **Server caption** (this is displayed), **Server name** (this is the COM object name), and the **Server URL** (the URL of the LOYTEC device) as shown in Figure 150.

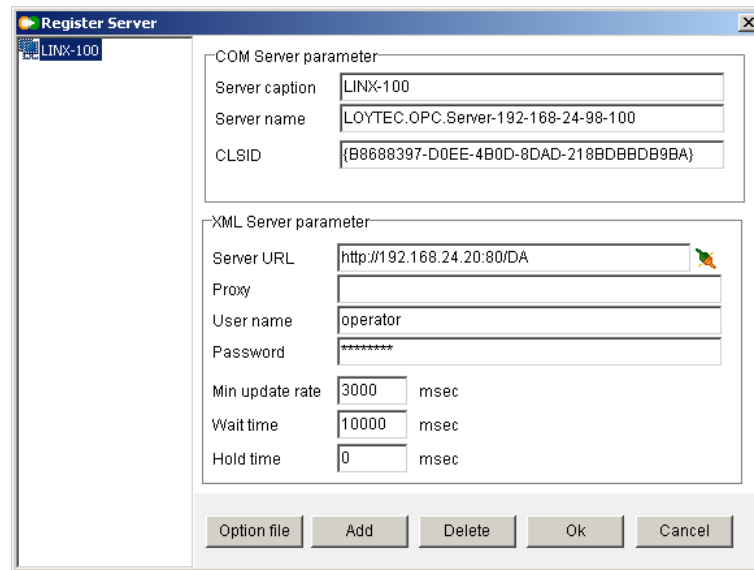



Figure 150: Register Server dialog in the bridge software

6. Additionally, enter the “operator” as the **User name** and its **Password** on the device. Enter the default timing parameters (min update rate 3 sec, wait time 10 sec, hold time 0). For information on the timing parameters refer to Section 8.3.6.
7. To test the connection to the device, press the button .
8. Click **OK**.

8.3.6 Bridge Timing Parameters

The OPC bridge uses three timing parameters to operate on the OPC XML-DA side. The defaults for those parameters are good for most cases. For specific use cases, however, they might need to be adapted.

- **Minimum Update Rate:** This parameter defines the rate at which the LOYTEC OPC server will update its value caches. In general value changes could occur much more frequently than the value caches are updated. Therefore, the minimum update rate determines the granularity at which updates may be notified to the OPC client. If an OPC client specifies an update rate less than this value for a group of subscribed data points, the group will not be updated faster. If the group specifies a slower update rate, however, the device will be polled less often. Lowering this parameter gives better responsiveness to data changes but also increases the XML-DA traffic and impacts overall performance. The recommended default is 3 seconds.
- **Wait Time:** This parameter specifies how long the OPC bridge shall wait for value updates before issuing the next polled refresh to the device. This time can be higher than the update rate because changes are notified within the wait time as soon as they occur on the device. The recommended default is 10 seconds.
- **Hold Time:** This parameter specifies that changes will be held back by the OPC server for the given amount of time before they are notified to the client. Normally, changes in the server are notified at the update rate’s granularity. In rare cases this can be used to hold back changes beyond the update rate. The recommended default is 0.

8.3.7 Bridge Limits

One installed instance of the LOYTEC OPC bridge software LOPC-BR800 can serve multiple devices. These appear as separate OPC COM/DCOM servers on the PC. The actual number of devices is limited depending on the configured timing parameters.

Regarding the proposed settings in Section 8.3.6 up to 5 concurrently opened devices are supported.

8.4 Using Custom Web Pages

Custom Web pages can also be developed for the LIOB-48x/58x. 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/xmla/1.0/>.

Any Web content, including scripts, applications or static Web pages can be stored directly on the device'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.

9 IEC 61131

To design the IEC61131 program which should run on the 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 is 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.

9.1 Overview

The PLC in the 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 L-IOB IOs. Figure 151 depicts the usage of data points for IEC61131 programs at the example of using CEA-709 network variables.

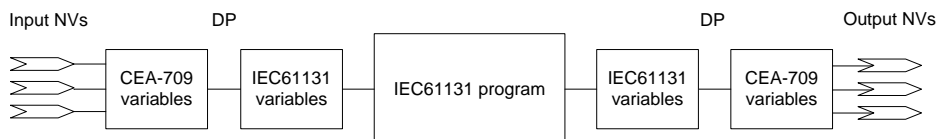


Figure 151: Usage of data points in IEC61131

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.

9.2 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 device.
2. L-INX Configurator. This software is required to configure the 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 related software packages. These packages include the template project for the device, the required software to build IEC61131 programs for the device, and required extensions to interface to CEA-709 networks. Follow the instructions of the installer to install logiCAD.

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.

9.2.1 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 152. In this dialog press the button **Download Softlock License**.

Important!

When using Windows 7 or Windows 8, 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.

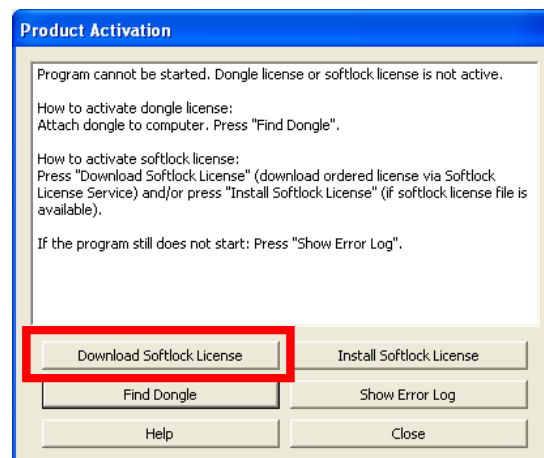


Figure 152: logiCAD Product Activation

Enter the license data provided with the product information on the Web site form as shown in Figure 153. 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.

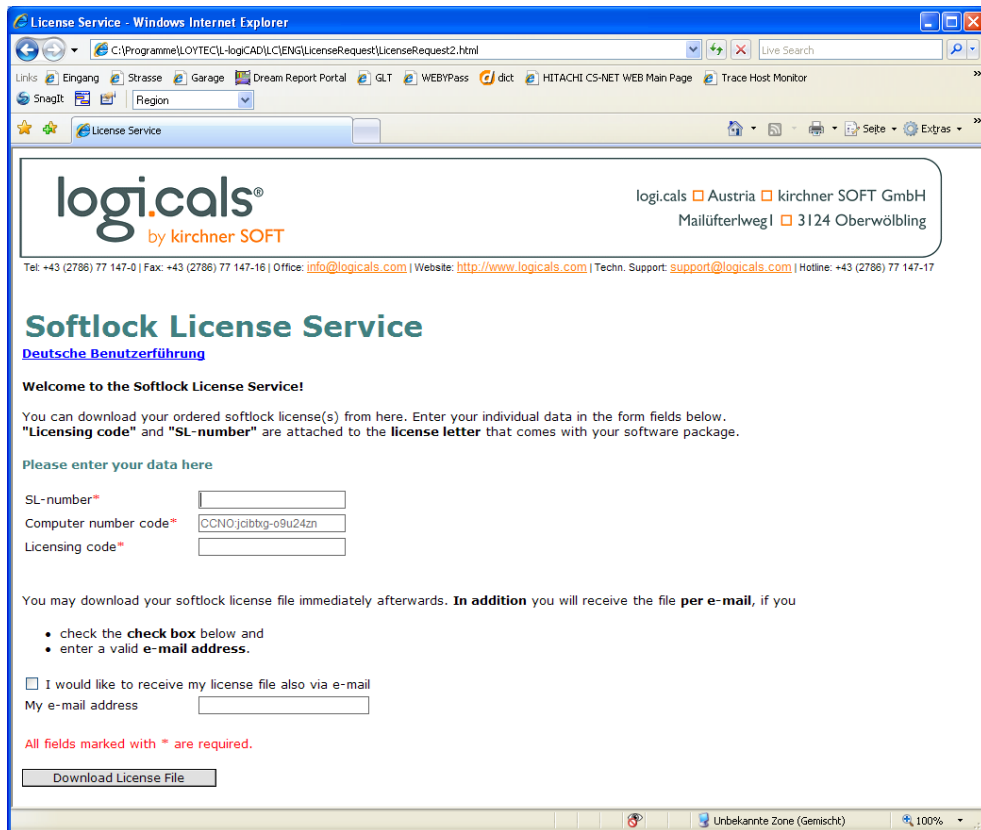


Figure 153: logiCAD softlock license Web form

Then install the license using the **Install Softlock License** button in the product activation dialog as indicated in Figure 154.

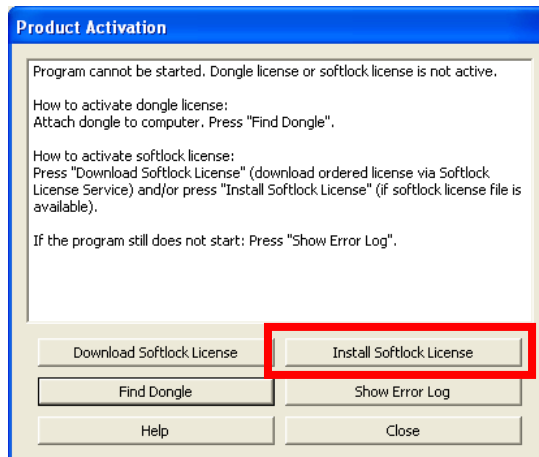


Figure 154: logiCAD softlock installation

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.

9.2.2 Hardlock License

The hardlock license is needed to run a copy of logiCAD in a virtual machine on the PC. It needs to be purchased 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 155 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**.

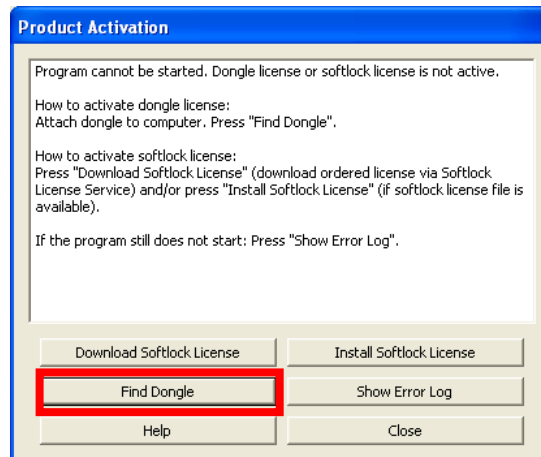


Figure 155: logiCAD Product Activation with Hardlock

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!*

9.3 IEC61131 Project Files

In the L-INX Configurator select the **LogiCAD Files** tab to attach an IEC61131 program and a logiCAD project to the project. The tab is shown in Figure 156.

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 device,
- Required connections,
- IEC61131 program.

The logiCAD project directory can also be attached to the configuration file, in order to include the logiCAD project sources from which the program was compiled.

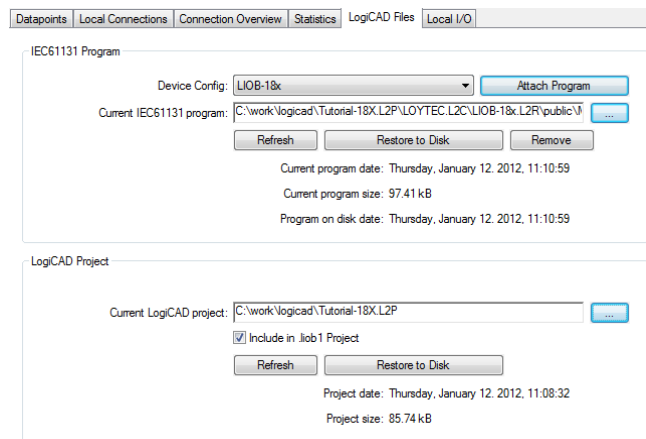


Figure 156: IEC61131 Project Files

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.

9.4 Working with logiCAD

For using the device with logiCAD, a predefined project template for the LIOB-18x/48x/58x must be used. Hence, when creating a new project e.g. for a LIOB-180, select the project template “Project for LIOB-18x”, see Figure 3. For additional information how to create, delete and manage projects please refer to the logiCAD online help.

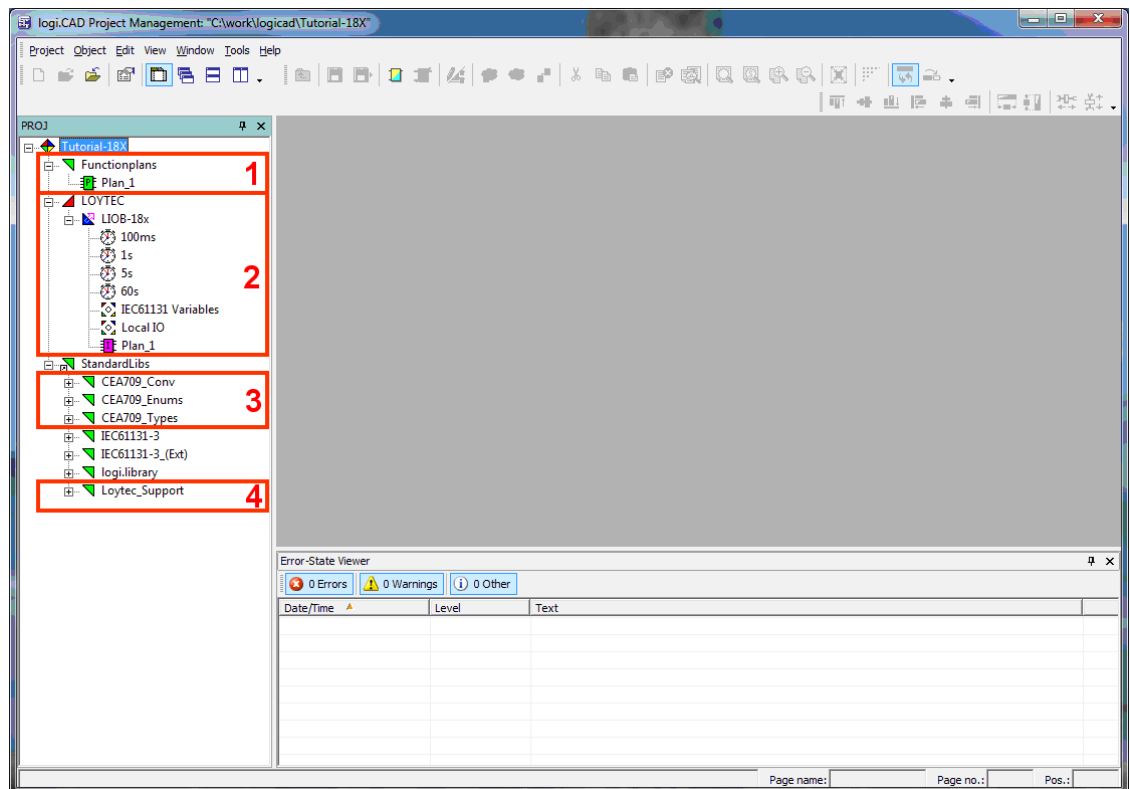


Figure 157: LOYTEC specific extensions

Figure 157 shows the standard project for the LIOB-18x/48x/58x including all LOYTEC-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/LIOB-18x** represents the device. The folder **LOYTEC** represents a configuration containing one LIOB-18x/48x/58x 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 LIOB-18x/48x/58x template, a program instance of the ‘Plan_1’ is already defined. To be able to transfer IEC61131 variables from the device to the IEC61131 program a global variables object within the **LIOB-18x**, **LIOB-48x**, or **LIOB-58x** folder is required, see Section 9.4.1 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 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 project, which perform this conversion. See Section 9.4.3 for details.
4. For designing programs that support the force update functionality (see Section 9.6.1), or designing user-defined *Technology Mapper* (see Section 9.6.2), additional function blocks are required. These blocks are located within the **Loytec_Support** directory.

All LOYTEC-specific add-ons are provided using function blocks. Hence, in the following, all samples are based on designs using function blocks.

9.4.1 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 device 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 other technologies) and the PLC program is depicted in Figure 158.

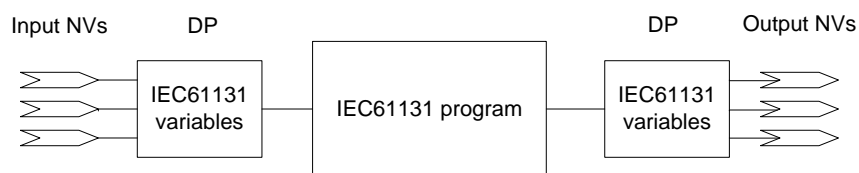


Figure 158: Connecting IEC61131 variables

The place where global variables are created on the device is shown in Figure 159.

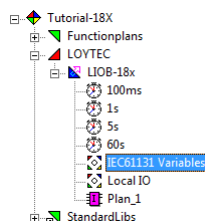
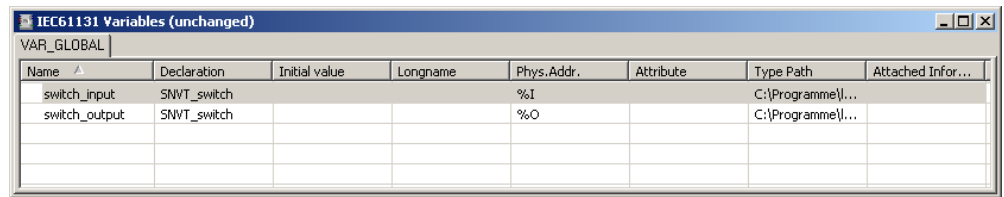


Figure 159: Global Variables Object

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 7.3.6.



Name	Declaration	Initial value	Longname	Phys.Addr.	Attribute	Type Path	Attached Infor...
switch_input	SWT_switch			%I		C:\Programme\...	
switch_output	SWT_switch			%O		C:\Programme\...	

Figure 160: Variables defined in global variables object

Figure 160 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.*

9.4.2 Build and Download the IEC61131 Program

IEC61131 programs, designed using logiCAD, must be cross compiled in order to run on the 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 LIOB-18x/48x/58x resource, see Figure 159, 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 9.4.5 for details.

After successfully finished code generation, the IEC61131 application can be downloaded to the device. Right click on the LIOB-18x resource and select download.

The IEC61131 program is downloaded to a LIOB-18x device via CEA-709:

- **CEA-709:** Select the network interface to use and fill out all other fields, see Figure 161. Alternatively, select the network interface and press Auto-detect via Service-Pin. Then press the service pin on the device. Note that this connection method requires an installed LOYTEC network interface.

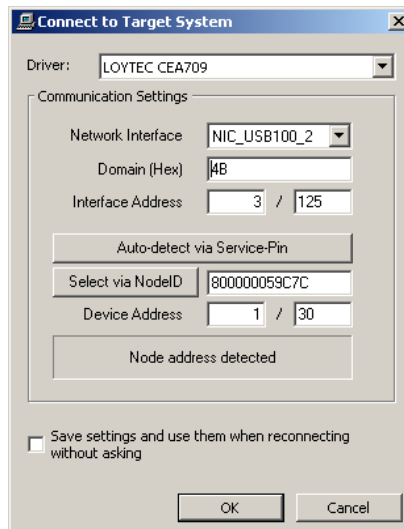


Figure 161: Connect via CEA-709

The IEC61131 program is downloaded to a LIOB-48x device via TCP/IP or CEA-852:

- **TCP/IP:** Enter the IP address of the LIOB-48x device. Do not change the default communication port (2048). This is the easiest and fastest way to connect to the device.
- **CEA-852 (CEA-709 over IP):** If for some reason, downloading directly via TCP/IP is not possible (e.g. because the required communication port is not accessible over the network), the download can also be done over CEA-852. Select CEA-709 as described above and choose the NIC852 which is a member of the same CEA-852 channel as the LIOB-48x device.

Important: *To be able to communicate with the device via CEA-709 or CEA-852, the device must be commissioned.*

The IEC61131 program is downloaded to a LIOB-58x device via TCP/IP:

- **TCP/IP:** Enter the IP address of the LIOB-58x device. Do not change the default communication port (2048).

9.4.3 Usage of NVs, Technology Converters (LIOB-18x/48x)

To use 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 172 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 162 shows the three possibilities how to use NVs within an IEC61131 program.

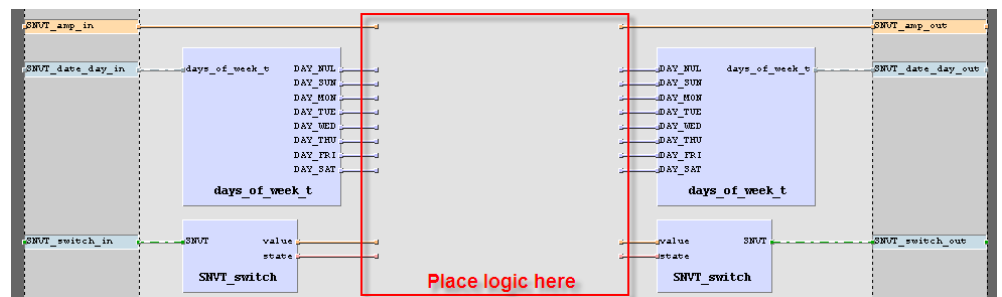


Figure 162: Usage of NVs

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.

9.4.4 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 159 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 0 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 163.

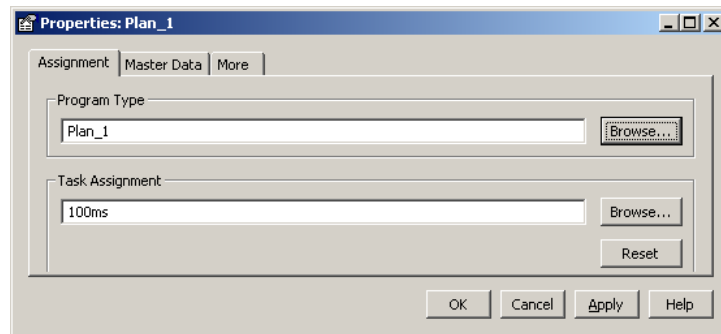


Figure 163: Task assignment

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 template project use the configured cycle time as task name.

9.4.5 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 device.
- 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 on the LCD UI, see Section 4.1. 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.

9.4.6 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 5.2.13).

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 164). If the I/O driver is disabled, it can be temporarily enabled again on the Web interface until the next reboot.

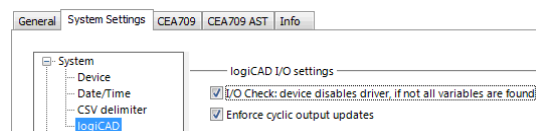


Figure 164: System settings for I/O check

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.

9.5 Workflows

9.5.1 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 165 presents the basic steps of this workflow. To follow the steps refer to the quick-start description of Section 2.5.

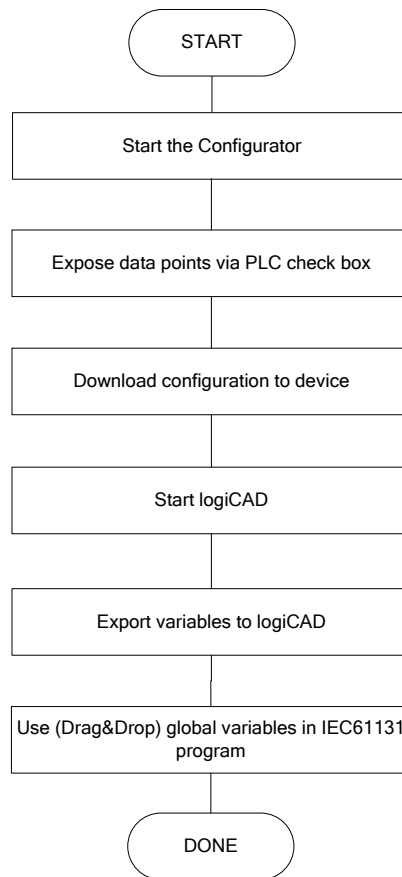


Figure 165: Start with network based information

It is assumed that there are already network data points (CEA-709, I/O, 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. Additionally 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 166. The external variables (see Section 9.5.2) are automatically created when adding the global variable to the Functionplan.

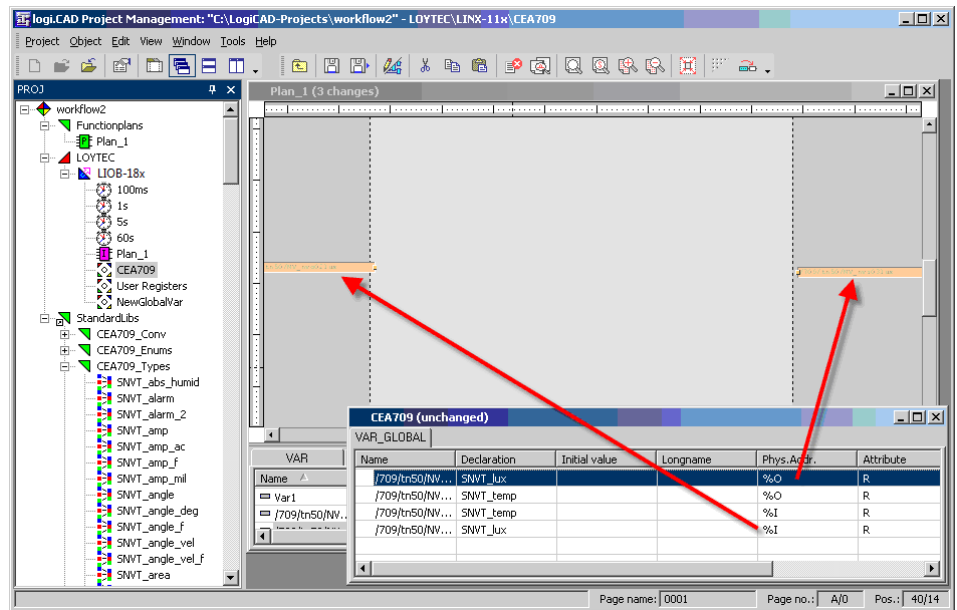


Figure 166: Adding global variables to Functionplans

After adding function blocks that perform the expected work, the IEC61131 program is ready to compile and download (see Section 9.4.2).

9.5.2 Starting with logiCAD

This section introduces a workflow how to develop a new IEC61131 program from scratch using logiCAD. Figure 167 shows the necessary steps to perform. For starting a new logiCAD project refer to the quick-start of Section 2.5.

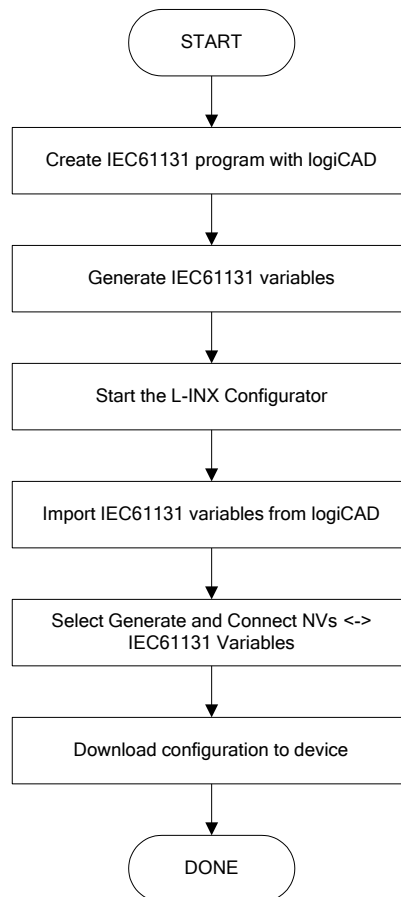


Figure 167: Starting with logiCAD

After creating a new project for a LIOB-18x/48x/58x resource and opening the Functionplan Plan_1 an empty input area is shown, similar to Figure 168.

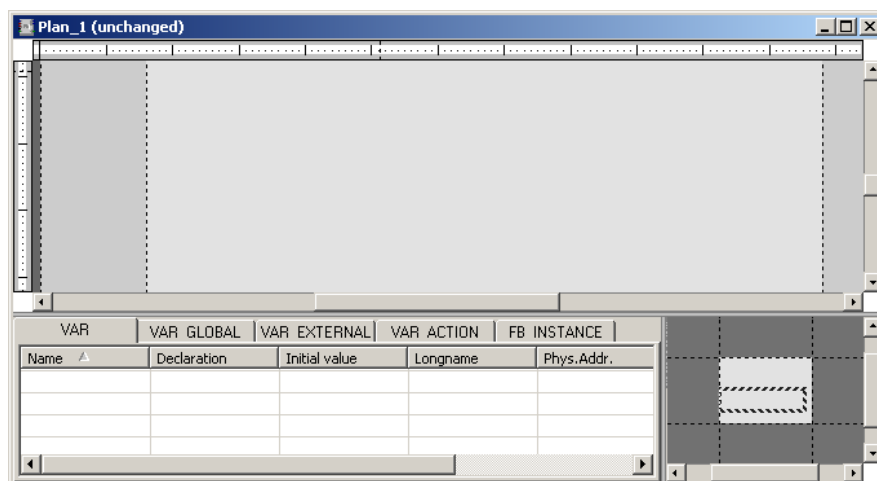


Figure 168: Start new function plan

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 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 169.

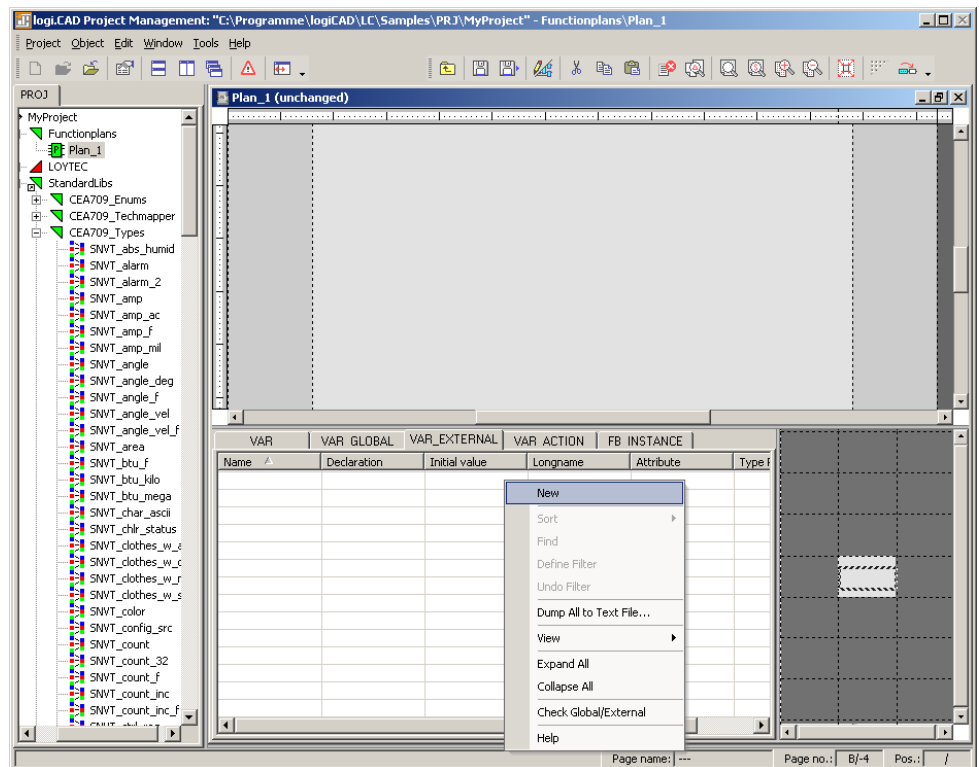


Figure 169: New external variable

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 170. Finally, the new variable is added by pressing the **Add** button.

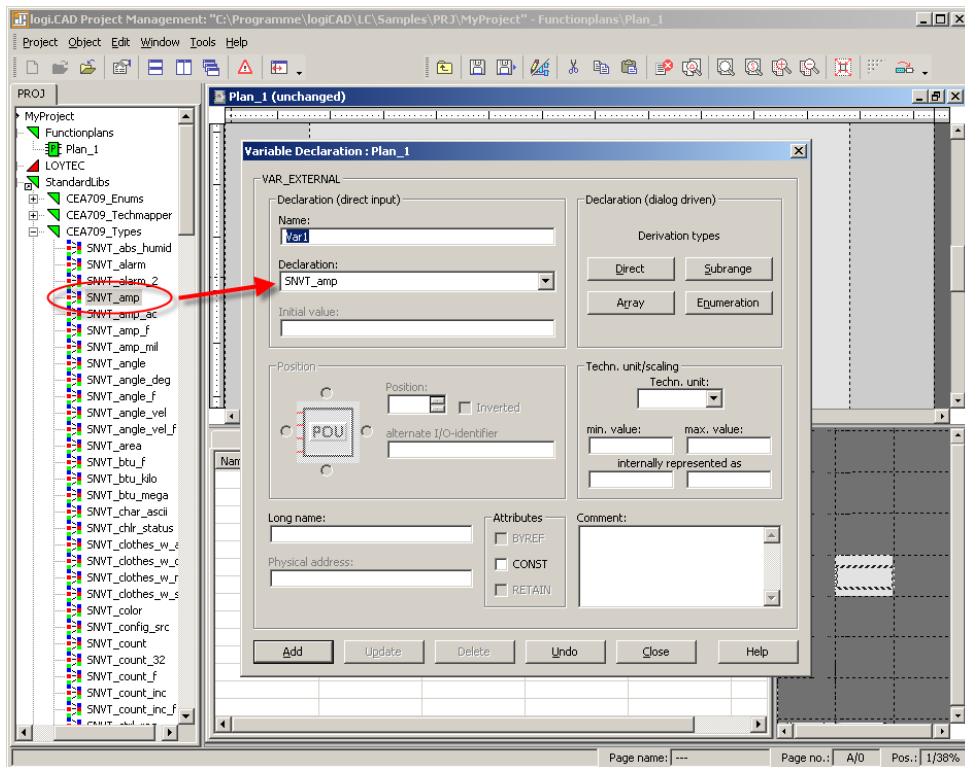


Figure 170: External variable type declaration

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.

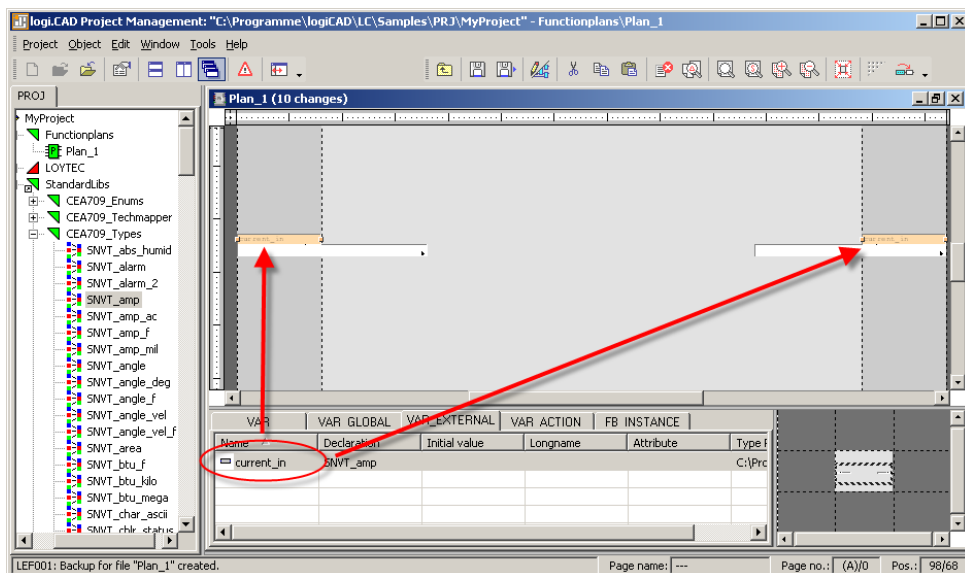


Figure 171: Drag-and-drop external variable to drawing area

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 172 for a sample configuration.

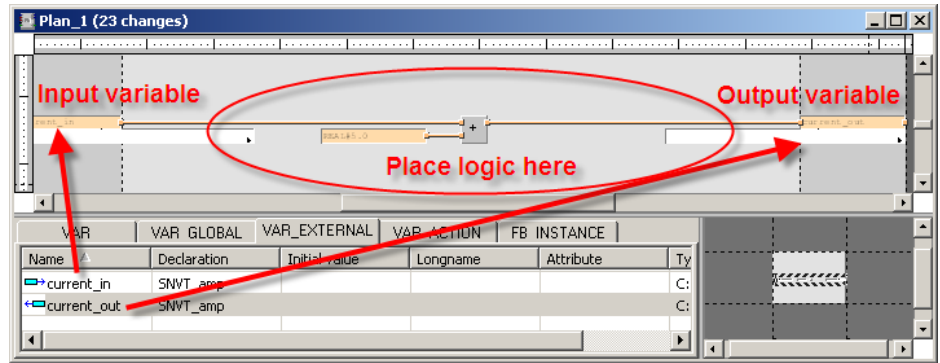


Figure 172: Use external variables

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.

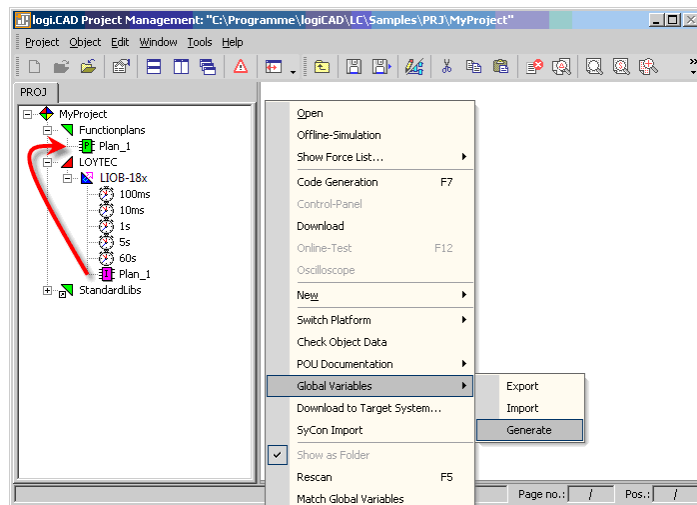


Figure 173: Auto-create global variables

To start the creation of global variables, based on external variables, **Save** all changes, then right click the LI0B-18x/48x/58x device and select **Global Variables** → **Generate**. Then the selected resource is parsed and every program instance found is checked for external variables. In Figure 173 the type instance Plan_1 refers to the Functionplan ‘Plan_1’, as defined in Figure 172. 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.

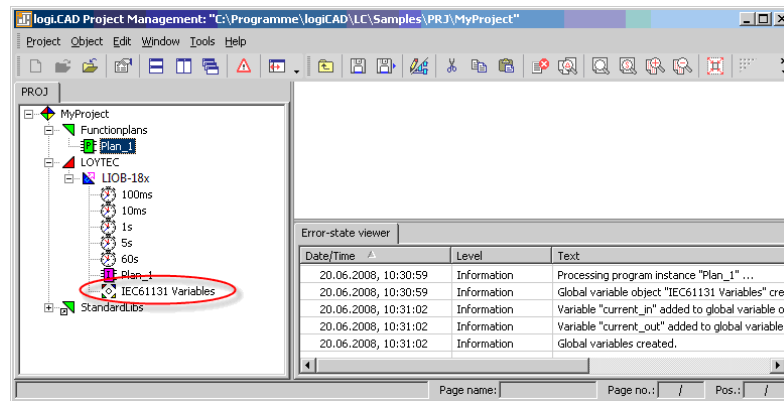


Figure 174: Created global variables object

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 174 for details.

The global variables are created based on the following rules:

- The direction of the variable is determined based on the graphical representation, shown in Figure 175.

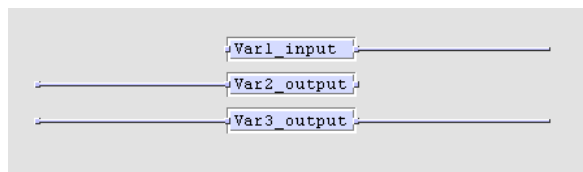



Figure 175: Direction of global variables

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 9.6.1) or as marker (see Section 9.6.4). 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 9.4.2 for details.

Based on the above created global variables corresponding IEC61131 data points are created on the device. 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 176 shows the result of the import process.

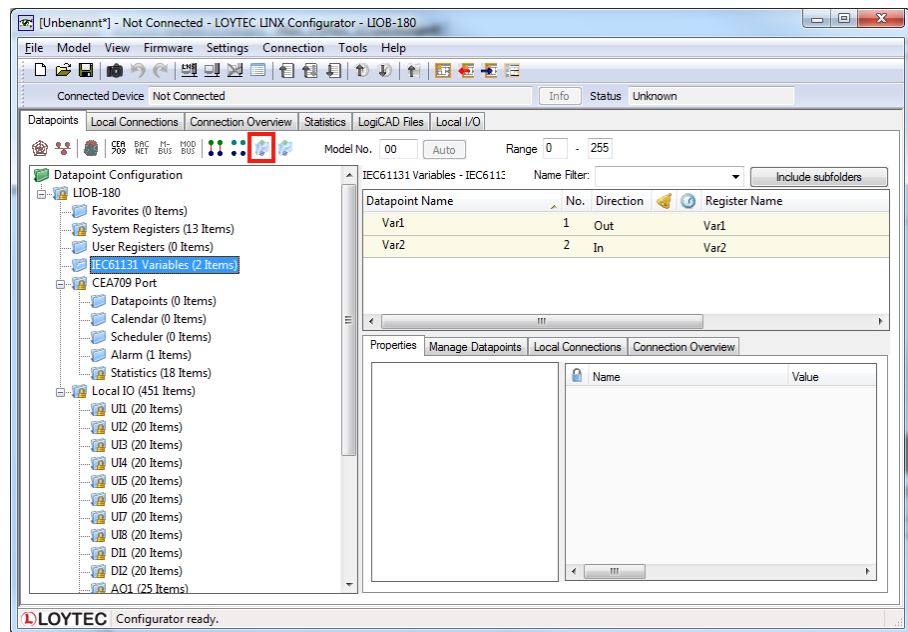


Figure 176: Connect IEC61131 variables

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 device. See Chapter 7 for a general description of the L-INX Configurator.

After rebooting the device, the IEC61131 program is up and running. Check the PLC LED for potential overload.

9.5.3 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 9.5.1. 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 Configurator project. If the network interface for the device 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 7.10 for details on creating connections manually.

Finally, after downloading the configuration and rebooting the device, the IEC61131 program can be downloaded to the device via the L-INX Configurator. After a final reboot the device loads and executes the IEC61131 program.

9.6 Additional Features

9.6.1 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.

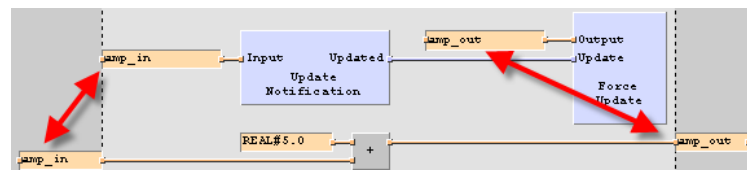


Figure 177: Force update

Figure 177 depicts how to use the force update functionality. Besides the part of the IEC61131 program that defines the calculations to perform (lower half of Figure 177), 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.

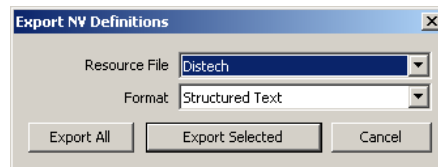
Important: *Every global variable connected to the update notification or force update block must be connected via the right-hand side terminal!*

9.6.2 Using UNVT variables (LIOB-18x/48x)

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 LONMARK resources files.

To generate type definitions 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.

9.6.3 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 device. 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), and simple arrays.

While the Configurator will be able to automatically determine the required data point size for simple arrays (like ARRAY [1..16] OF INT) and create a suitable register data point during the IEC61131 variable import, the data size of custom structures cannot be determined automatically at the moment, 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.

9.6.4 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 6.3.4). There is no difference in handling global variables connected to persistent data points or to non-persistent data points. Global variables connected to persistent data points are marked with retain attribute in logiCAD and 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 set an input variable as persistent in logiCAD, open the global variables object containing the appropriate variable, double click the variable, set the **retain** check box and press the **Update** button. Now export the global variables and import them to the L-INX Configurator. After downloading the new IEC61131 point configuration to the device, the data point is set to persistent. Setting the data point persistent in the L-INX Configurator, export the data points and import them to logiCAD is also possible.

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 177. As a result it is not possible to create a global variables object holding markers automatically. The procedure as described in Section 9.5.1 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.

9.6.5 System Registers, System Time

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 7.10).

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.

9.6.6 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.

9.6.7 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 9.4.3) 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.



Datapoint Name	No.	OPC	Param	PLC	Direction
mod1_Read	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In
byte_0	1.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	In
byte_1	1.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	In

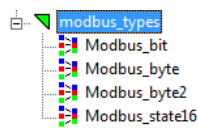
2. Export the variables to logiCAD. The exposed structure members appear for example as 'UR/mod1_Read.byte_0' in the 'User Registers' global variables block.
3. Export the types for the respective type category in the menu **Tools** → **Manage Structured Types...**. This opens the structured types management dialog.
4. Choose the type **Category**, e.g. 'Modbus'.

Category

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.

Disk Repository Path 

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.



9.6.8 BACnet Server Objects (LIOB-58x)

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 I/Os 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 178 shows an example, where the PLC uses the input ‘/BAC/AI2.Present_Value_Read’ to get the sensor values from AI2.

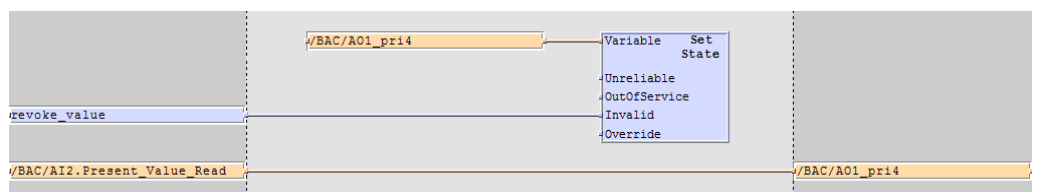


Figure 178: PLC with BACnet objects

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 178 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 178 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 sensor 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.

10 Operating Interfaces

10.1 Common Interface (LIOB-48x/58x)

10.1.1 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 141. 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/>

10.1.2 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 141. 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 <code>logger_entry</code> 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 10: 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.

10.1.3 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 10.1.2.

10.2 CEA-709 Interface (LIOB-18x/48x)

10.2.1 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 11. The columns are separated by commas ',' or semi-colons ';'. Which separator is used can be configured in the Web UI (LIOB-48x, 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 11: CSV Columns of the NV Import File

10.2.2 Node Object

The LIOB-18x/48x provides 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 time is set. 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.
- LIOB-48x: nvoIpAddress (*SNVT_str_asc*): This NV can be used to poll the IP address of the device. It does not send updates.

10.2.3 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).

10.2.4 Calendar Object

When the scheduler objects are enabled in the project settings, the device includes the standard LONMARK calendar object.

10.2.5 Scheduler Object

When the scheduler objects are enabled in the project settings, the device includes the configured number of standard LONMARK scheduler objects.

10.2.6 PLC Objects

The device contains eight proprietary PLC objects called 'PLC'. The names and counts can be changed in the Configurator software, see Section 7.8.4. These objects 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 PLC blocks.

10.3 BACnet Interface (LIOB-58x)

10.3.1 Device Object

The BACnet interface provides one device object as shown in Table 12. 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_Recipients	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

Table 12: Properties of the Device Object

10.3.2 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 inter-network-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 5.2.8). 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 5.2.8). For the Device object, this name shall be unique within the BACnet inter-network.

Object_Type (Read-Only). The object's type. For the Device object, the value of this property is 8 (= DEVICE).

10.3.3 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 "LIOB-580" or "LIOB-581".

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 5.2.8). 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 5.2.8).

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.

10.3.4 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.

10.3.5 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 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.

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.

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.

10.3.6 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.

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.

10.3.7 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).

10.3.8 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 13 outlines how these properties interact.

Time_Synchronization_Interval	Align_Intervals	Periodic Time Synchronization Event At...
Multiple of 1440 (minutes), i.e., multiple of one day	TRUE	<i>Interval_Offset</i> minutes after midnight, every (<i>Time_Synchronization_Interval</i> /1440) days
Multiple of 60 (minutes) but <i>not</i> multiple of 1440 (minutes), i.e., multiple of one hour	TRUE	<i>Interval_Offset</i> minutes from the current * hour, every (<i>Time_Synchronization_Interval</i> /60) hours
Multiple of 1440 (minutes), i.e., multiple of one day	FALSE	<i>Interval_Offset</i> minutes from the current * minute, every (<i>Time_Synchronization_Interval</i> /1440) days
Multiple of 60 (minutes), but <i>not</i> multiple of 1440 (minutes), i.e., multiple of one hour	FALSE	<i>Interval_Offset</i> minutes from the current * minute, every (<i>Time_Synchronization_Interval</i> /60) hours
Neither multiple of 60 or 1440, but greater than zero	TRUE or FALSE	<i>Interval_Offset</i> minutes from the current * minute, every <i>Time_Synchronization_Interval</i> minutes
Zero	TRUE or FALSE	Never

Table 13: 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 13 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.

10.3.9 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 be 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.

10.3.10 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 14. Lines beginning with a hash ('#') sign are comment lines. The example values in Table 14 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)
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)
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
H	Interval/ Priority	8	Defines the poll interval in seconds for poll mappings and the COV lifetime in seconds for COV mappings. For write mappings this defines the write priority (1..16). Omit this field or set it to '-1' to write w/o priority.

Table 14: CSV Columns of the BACnet Client Mappings File

10.3.11 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.


11 Firmware Update

The L-IOB firmware supports remote upgrade over the CEA-709 interface (LIOB-18x), the CEA-852 interface (LIOB-48x), and over IP (LIOB-48x/58x). To guarantee that the L-IOB is not destroyed due to a failed firmware update, the L-IOB firmware consists of two images:

1. fallback image,
2. 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 L-IOB device boots up with the fallback image, the status LED is flashing red. The primary image can be updated using the Configurator Software. See Section 7.1 on how to install the Software.

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 to device** or **Connection** → **Connect via LNS**. See Sections 7.6.1 and 7.7.1 for setting up the connection.
3. In the connection dialog click on **Connect**.
4. Optionally, check for updates by selecting the menu **Help** → **Check for updates** . This function checks for new firmware and Configurator versions.
5. Select the menu: **Firmware** → **Update**.
6. This opens the Firmware Update dialog as shown in Figure 179. Click on the button  and select the firmware image.

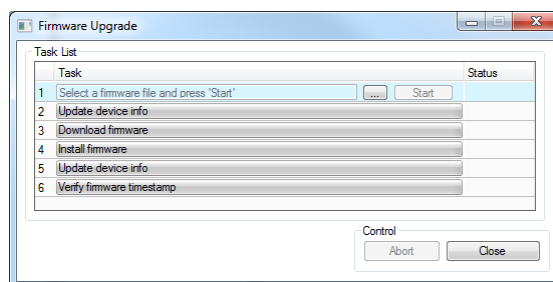


Figure 179: Firmware Update dialog of the Configurator

7. Click on **Start** and observe the download progress.
8. When the download is complete, a dialog appears. Click **OK**.
9. In the Firmware Update dialog, click **Close**.
10. The device's firmware has now been successfully upgraded.

12 Troubleshooting

12.1 Technical Support

LOYTEC offers free telephone and e-mail support for the L-IOB 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.
N27W23957 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

12.2 Remote Packet Capture (LIOB-48x/58x)

12.2.1 Configure Remote Packet Capture

Remote packet capture is able to capture packets on the Ethernet port. To enable the remote packet capture feature, go to the **Config / Port Config** Web page and enable Remote packet capture as shown in Figure 180.

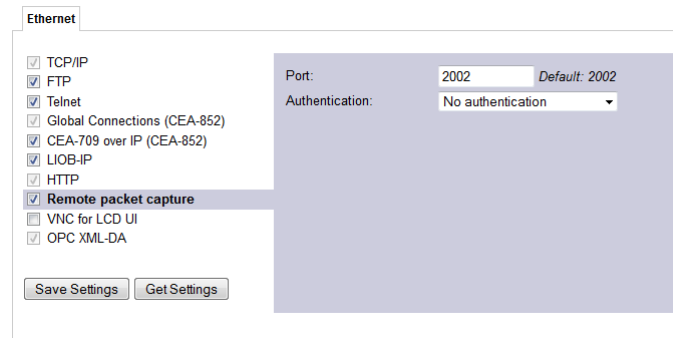


Figure 180: Remote packet capture port configuration.

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.

12.2.2 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 181.

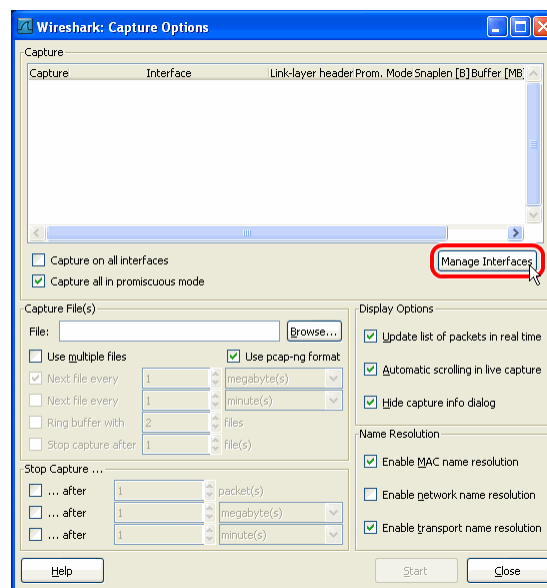


Figure 181: Wireshark Capture Options Dialog.

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 182.

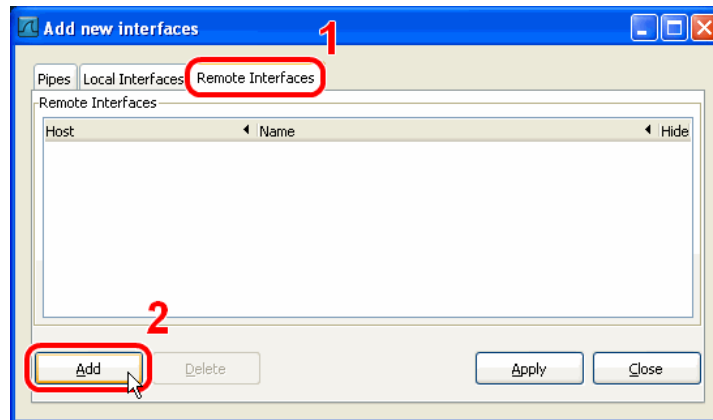


Figure 182: Wireshark Add New Interfaces Dialog.

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 183.
5. Note that only the users **admin** and **operator** are allowed to connect.

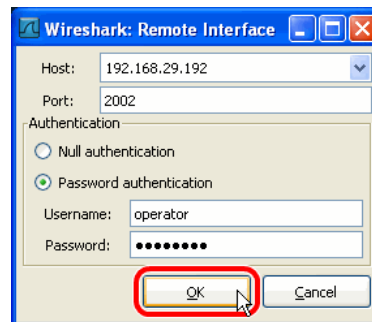


Figure 183: Wireshark Remote Interface Dialog.

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 184.

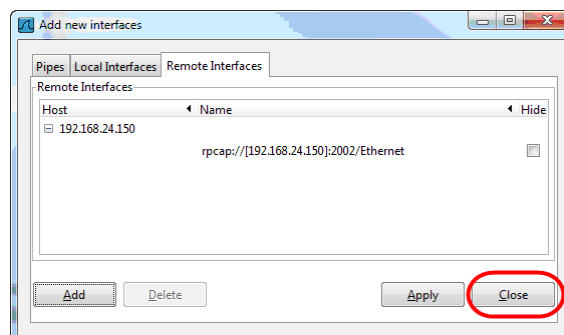


Figure 184: Added new interface to Wireshark.

- Close the **Add new interfaces** and **Capture Options** dialogs to return to the main window.

To Start a Remote Capture

- Select the created remote interface from the interface list in the main window. It is named 'Ethernet'.
- Click the **Start** button as shown in Figure 185.

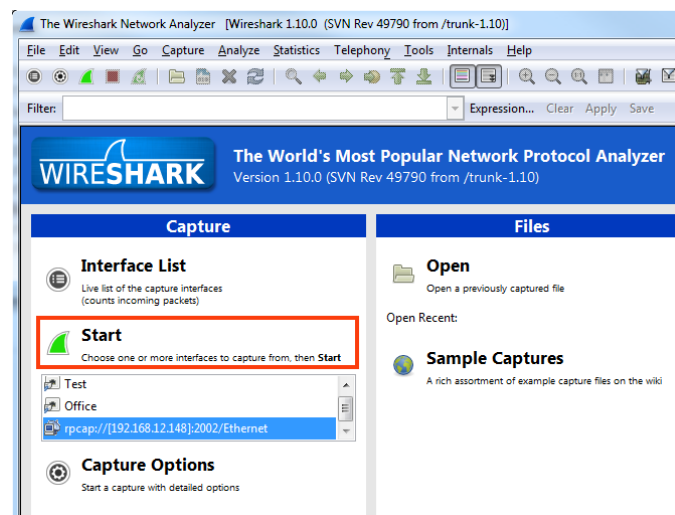


Figure 185: Start Remote Capture in Wireshark.

- Wireshark will attempt to establish a connection to the device and, if successful, start displaying packets. An example capture is shown in Figure 186.

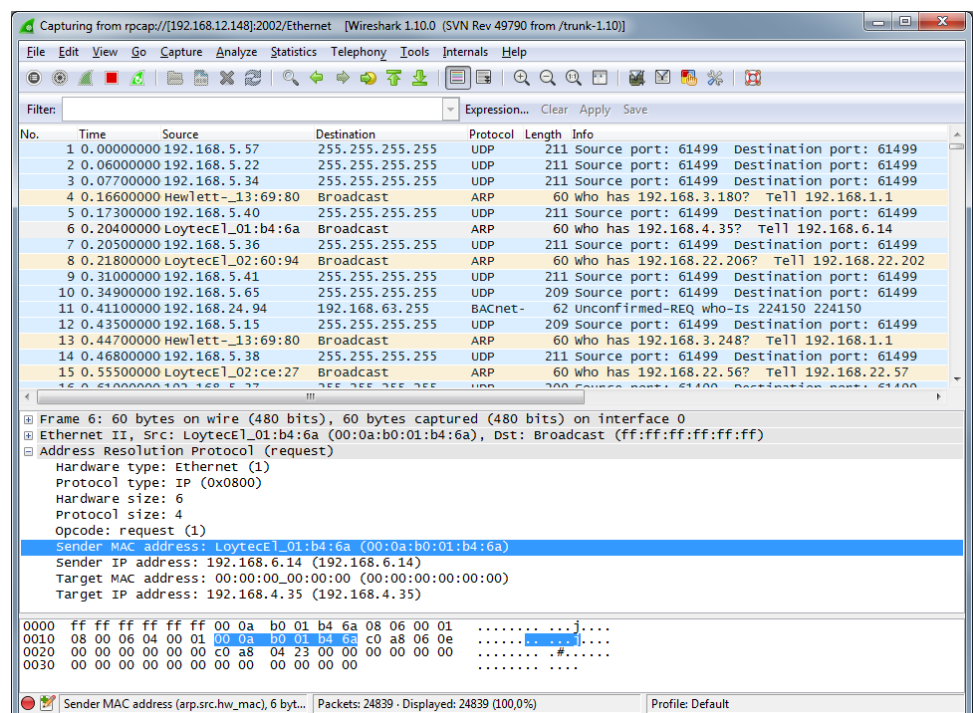


Figure 186: Example remote capture in progress.

13 Application Notes

13.1 External Power Supply (not using LPOW-2415A)

When using a non-LOYTEC power supply (see Figure 187), the following rules must be complied with:

- Consistent polarity must be maintained when connecting LOYTEC I/O controllers and modules to the transformer. That is, the ‘- ~’ terminal of each I/O controller and each I/O module must be connected to the same terminal on the secondary side of the transformer.
- The I/O controllers and modules are half-wave rectified. Connecting two half-wave rectified devices to the same transformer without maintaining polarity will cause short circuit.
- The GND terminals of the I/O controller or module are internally wired to the ‘- ~’ terminal. Therefore, if powering I/O controllers and modules with the same transformer, it is again essential to maintain polarity. Failure to do so will result in short circuit and/or damaged device.
- If the transformer output must be grounded, connect the ‘- ~’ terminal to earth ground.

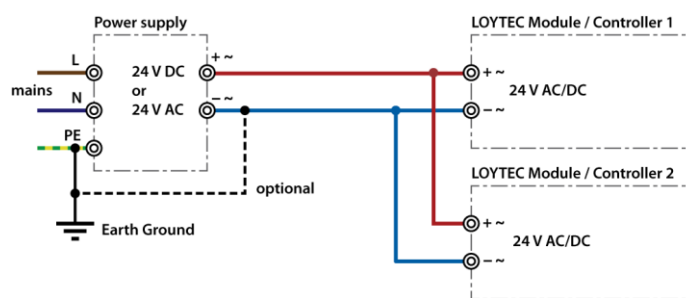


Figure 187: External power supply

13.2 Physical Connection of Inputs

13.2.1 Connection of Switches

On- or off-switches can either be connected to the DIs (Digital Inputs) or to the UIs (Universal Inputs) in digital interpretation.

13.2.1.1 Switch connected to a DI

A switch can be directly connected to a digital input as shown in Figure 188.

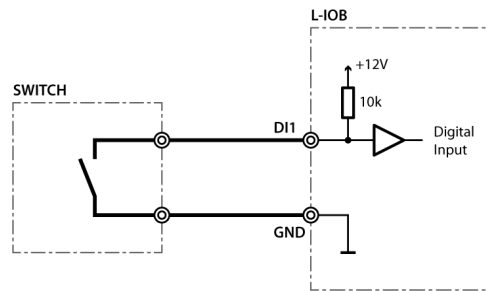


Figure 188: Switch connected to DI

The digital inputs (DI) recognize the following digital signals according to the connected resistance (switch):

Resistance of Switch	Status
< 6.8 k Ω	Closed Switch
> 10 k Ω	Open Switch

13.2.1.2 Switch connected to a UI

A switch can be directly connected to a universal input with signal type resistance as shown in Figure 189.

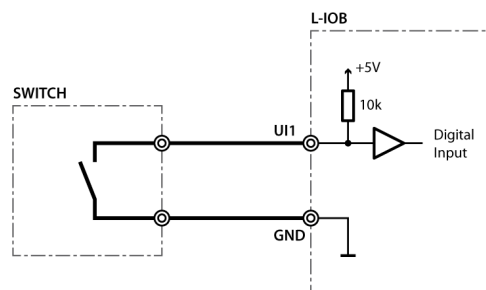


Figure 189: Switch connected to UI

UIs recognize the following digital signals according to the input resistance (switch):

Resistance Switch	Status
< 1.9 k Ω	Closed Switch
> 6.7 k Ω	Open Switch

13.2.2 Connection of S0 Pulse Devices (Meters)

S0 pulse meters must be connected to digital inputs (DI) as shown in Figure 190.

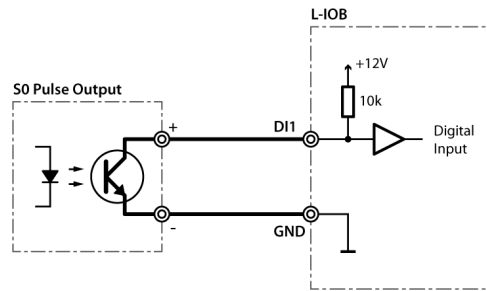


Figure 190: S0 pulse meter connected to DI

13.2.3 Connection of Voltage Sources to Universal Inputs

The Universal Input (UI) provides voltage measurement both if used as an analog or digital input. The signal type must be configured to 'Voltage 0-10V' or 'Voltage 2-10V' in both cases.

13.2.3.1 Voltage Source connected to UI with Analog Interpretation

Figure 191 shows the connection of a voltage source to a universal input in analog mode.

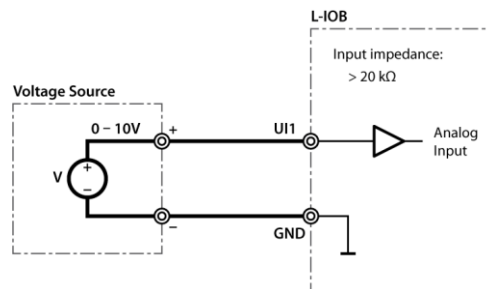


Figure 191: Voltage source on UI in analog mode

13.2.3.2 Voltage Source connected to UI with Digital Interpretation

Figure 192 shows the connection of a voltage source to a universal input in digital mode. In this case, the voltage source acts as a switch with the depicted low and high levels.

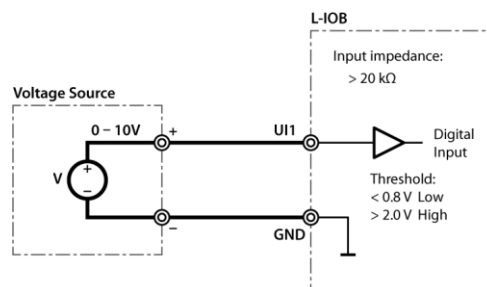


Figure 192: Voltage source on UI in digital mode

13.2.4 Connection of Voltage Sources to Digital Inputs

Figure 193 shows the connection of a voltage source to a digital input. In this case, the voltage source acts as a switch with the depicted threshold.

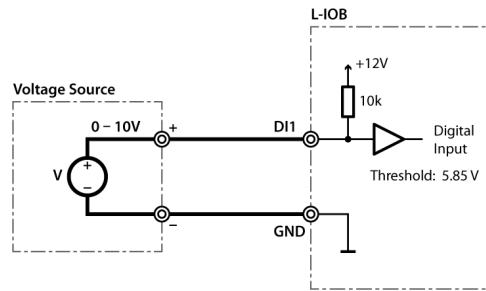


Figure 193: Voltage source on DI

13.2.5 Connection of 4-20mA Transmitters to Universal Inputs

13.2.5.1 4-20mA Transmitter connected to UI with Internal Shunt

Some universal inputs have an internal shunt which can be activated (in pairs with another UI) in the Configurator software (signal type 'Current 4-20mA Shunt'). Which UIs are equipped with shunts is documented in Section 15.3 and the following Sections. Figure 194 shows the connection of a 4-20mA transmitter to a universal input with internal shunt.

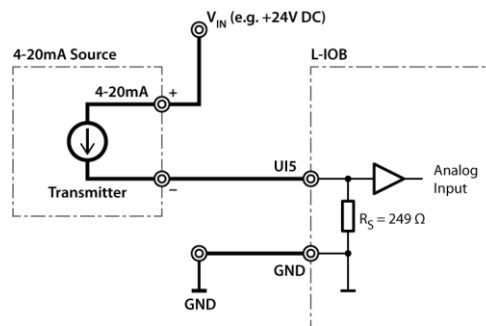


Figure 194: 4-20mA transmitter with internal shunt on UI

13.2.5.2 4-20mA Transmitter connected to UI with External Shunt

On universal inputs, which do not have an internal shunt, an external shunt must be used as shown in Figure 195. The signal type must be set to 'Current 4-20mA' in the Configurator software.

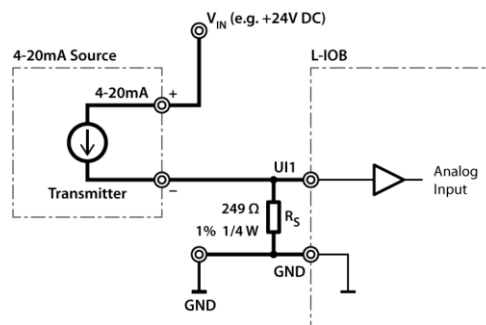


Figure 195: 4-20mA transmitter with external shunt on UI

13.2.6 Connection of Resistive Sensors

Figure 196 shows the connection of resistive sensors to the universal inputs with a temperature sensor as an example. Sensors in the resistance range of 1 kΩ to 100 kΩ can be measured. The signal type must be set to 'Resistance' in the Configurator software.

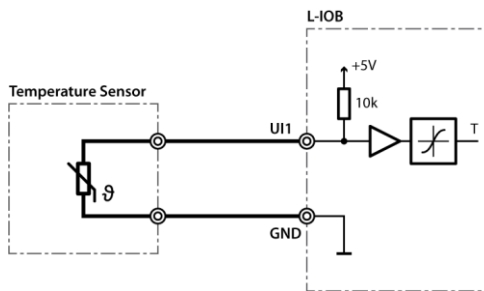


Figure 196: Temperature measurement on UI

13.2.7 Connection of STId Card Readers

Figure 197 shows the connection of an STId card reader to three L-IOB inputs (UIs or DIIs). Observe that the clock signal must be connected to an interrupt-capable input of the L-IOB device. More information on STId card readers can be found in Section 6.7.2.

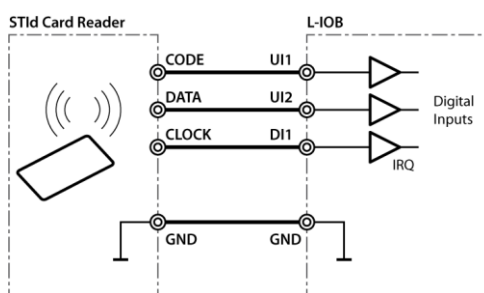


Figure 197: STId card reader

13.3 Physical Connection of Outputs

13.3.1 6A Relays with one External Fuse

The total current of all used 6A relays must be restricted to 6A, if more than two relays share a common (COM) terminal. The wiring shown in Figure 198 can be used for all L-IOB models with common terminals.

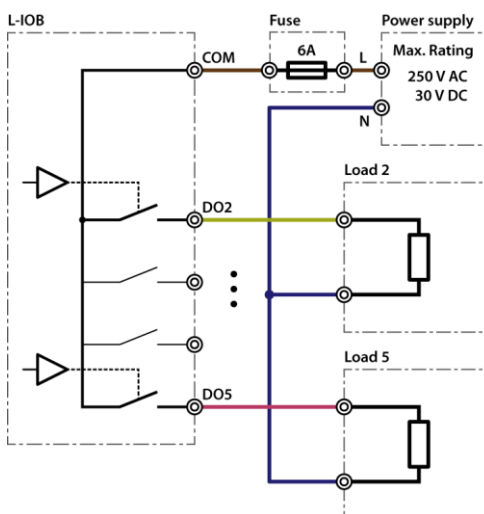


Figure 198: 6A relays with one external fuse

13.3.2 6A Relays on LIOB-xx2 using Separate Fuses

Figure 199 shows the wiring of the 6A relays for the LIOB-182/482/582 models using separate fuses. In this case, two relays share one common terminal (COM).

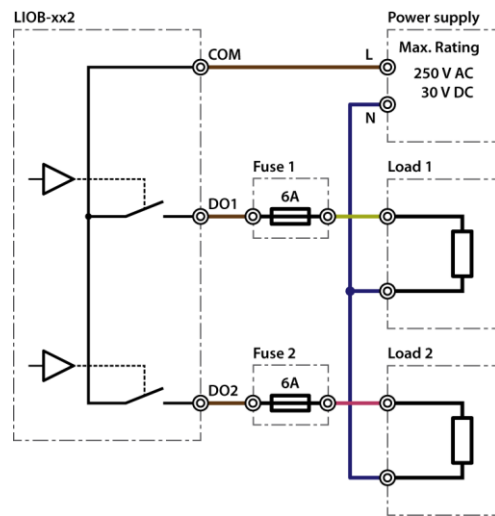


Figure 199: LIOB-182/482/582 6A relays

13.3.3 16A and 6A Relays on LIOB-xx3

The 16A and 6A relays on the LIOB-183/483/583 models all have two separate terminals per relay. There are no common (COM) terminals. This means that a 16A (or 6A) fuse must be wired to one of the two terminals of each relay, as shown in Figure 200.

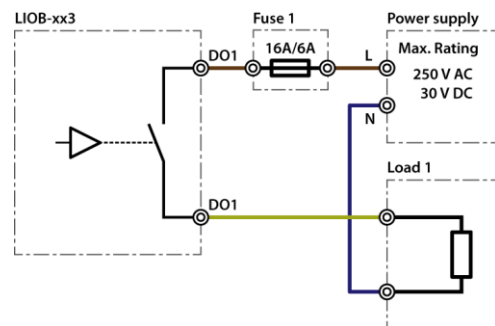


Figure 200: LIOB-183/483/583 16A/6A relays

13.3.4 External Relays and Inductive Loads

When controlling an external relay or inductive load using a L-IOB relay, either an integrated suppressor circuit must be used for the inductor, or a free-wheeling diode, a varistor, RC circuit, etc. must be installed to suppress voltage peaks and sparking due to switching off inductive circuits. It is recommended to use diodes that are part of the 1N400x family and to place them close to the relay, as shown in Figure 201. Figure 202 shows the connection of a 230V relay with a varistor.

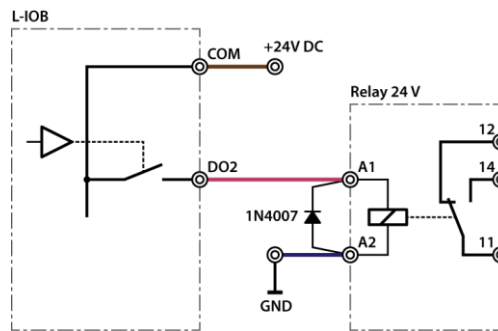


Figure 201: Suppressor circuit with free-wheeling diode

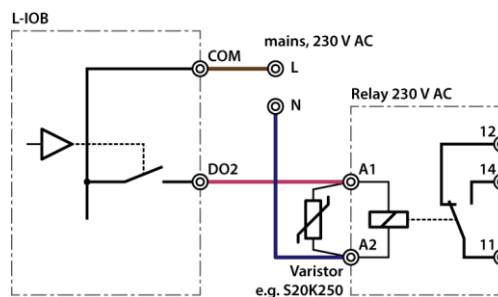


Figure 202: Suppressor circuit with varistor

13.3.5 1A Triacs

Figure 203 shows the wiring of the 1A Triac Outputs.

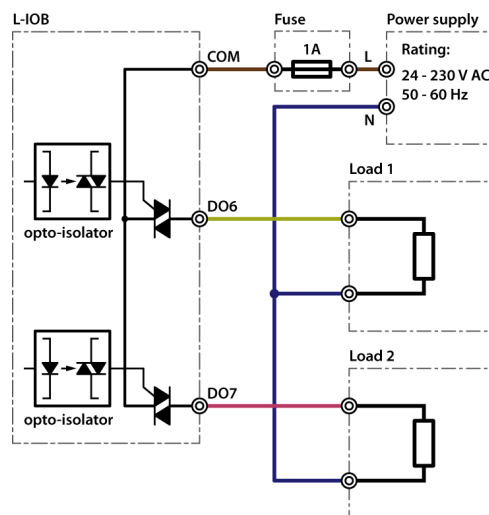


Figure 203: 1A Triacs

13.3.6 Analog Outputs

Figure 204 shows the wiring of the analog outputs (AO). Observe that the analog outputs are labeled '0-10V OUT' but are in fact capable of delivering over 11V.

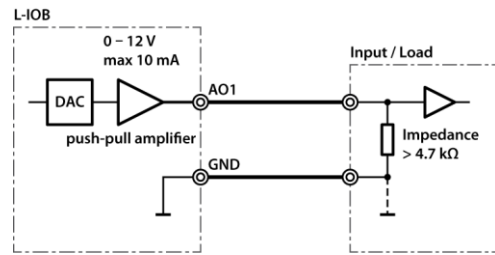


Figure 204: Analog outputs

The input impedance of the connected load must be greater than or equal to 4.7 kΩ for linear output.

14 Security Hardening Guide

This guide contains security-relevant information for operating the LIOB-48x/58x on IT networks. The information refers to the firmware version and the instructions found in the previous chapters of this User Manual.

14.1 Installation Instructions

Install the device over the Web interface:

- Set up the basic device functions and protocol settings as described in Chapter 2.
- Disable the FTP, and Telnet servers in the IP port configuration as described in Section 5.2.4.

14.2 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 11.

14.3 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. 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.
- 47808 udp (LIOB-58x): 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 14.1:

- 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.
- 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.
- 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.

14.4 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 (LIOB-58x): 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 14.1:

- HTTP: Web server. It provides a Web-based configuration UI. The Web UI can be disabled after setting up the device.
- 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.
- logiCAD online test: This service is used by the L-logiCAD programming tool for online debugging of IEC61131 programs. It is enabled by default. 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.

14.5 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 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.
- 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.

15 Specifications

15.1 I/O Specification

15.1.1 UI - Universal Input

UIs are universal inputs for four different input types. They correspond to class 1 with +/- 1 % accuracy, have an input voltage range of 0 to 10V, and can withstand up to 30 V. The ADC resolution is 16 bits. Galvanically isolated sensors resp. switches must be connected. Universal inputs can be configured as:

- **Binary Input (Digital Input):** input impedance > 20 kΩ, sampling period 10 ms.
 - In voltage mode, the threshold values are < 0.8 V for low level and > 2 V for high level.
 - In resistance mode, the threshold values are < 1.9 kΩ for low level and > 6.7 kΩ for high level.

Between the threshold values, the resulting level of the UI is not defined.

- **Voltage Metering 0-10 V:** input impedance > 20 kΩ, sampling period < 1 s, resolution 0.1 V.
- **Current Loop 4-20 mA:** input impedance 249 Ω, sampling period < 1 s. An internal shunt of 249 Ω is available for some universal inputs. Otherwise, an external resistor of 249 Ω must be used as a shunt.
- **Resistance Measurement:** input impedance 10 kΩ, sampling period < 1 s. Resistors in the range of 1 kΩ to 100 kΩ can be measured.

The average sampling period p of analog inputs depends on the number of active (non-disabled) universal inputs n that are configured in analog mode. The formula for p is:

$$p = n * 125ms$$

This means if e.g. only two UIs are configured as analog inputs, a new sample is taken every 250ms (on average) for each of the two inputs. The UIs configured as digital inputs are unaffected (sampling period always 10ms) by this formula.

15.1.2 DI - Digital Input, Counter Input (S0-Pulse)

DIs are fast binary inputs, which can also be used as counter inputs (S0). They follow the S0 specification for electric meters and have a sampling period of 10 ms. They change state at

a load of 195 Ω between the DI terminal and GND. Galvanically isolated sensors resp. switches must be connected.

15.1.3 AO - Analog Output

AOs are analog outputs with a signal range of 0 to 10 V (up to 12 V), a resolution of 10 bits, and a maximum output current of 10 mA (short circuit proof). The accuracy over the whole range is +/- 100 mV.

15.1.4 DO - Digital Output

The following digital outputs are available:

- Relay 6A Output: Switching capacity 6 A, 250 VAC resp. 30 VDC.
- Relay 16A Output: Switching capacity 16 A, 250 VAC resp. 30 VDC.
- TRIAC Output: Switching capacity 1 A, 24 to 230 VAC.

When connecting an air gap switch to a L-IOB relay, a quenching circuit like a varistor or RC element must be used.

15.1.5 PRESS - Pressure Sensor

These inputs represent differential pressure sensors which measure pressures from 0 - 500 Pascal. They are equipped with two 3/16" (4.8 mm) hose connectors.

15.2 Internal Translation Tables

The L-IOB devices offer fixed internal translation tables for easy configuration of some temperature sensors. The xin/xout values of these tables are listed in Table 15.

xout: Temp. [°C]	xin: Resistance [Ω]			
	PT1000	NTC10K	NTC1K8	Ni1000
-30	882.2	176680	24500	842
-20	921.6	96970	14000	893
-10	960.9	55300	8400	946
0	1000.0	32650	5200	1000
10	1039.0	19900	3330	1056
20	1077.9	12490	2200	1112
25	1097.4	10000	1800	1141
30	1116.7	8060	1480	1171
40	1155.4	5320	1040	1230
50	1194.0	3600	740	1291
60	1232.4	2490	540	1353
70	1270.0	1750	402	1417
80	1308.9	1260	306	1483
90	1347.0	920	240	1549
100	1385.0	680	187	1618
120	1460.6	390	118	1760

Table 15: Internal translation table values

15.3 Specification for LIOB-18x Models

Dimensions [mm]	107 x 100 x 75 (L x W x H)				
Operating Temperature (ambient)	0°C to +50°C				
Storage Temperature	-10°C to +85°C				
Humidity (non condensing) operating / storage	10 to 90 % RH @ 50°C				
Environmental Protection	IP 40 (enclosure); IP 20 (screw terminals)				
Power Supply	24 VDC / 24 VAC ±10 %				
Installation	DIN rail mounting (EN 50 022)				
Interface	1 x CEA-709/FT				
Types	LIOB-180	LIOB-181	LIOB-182	LIOB-183	LIOB-184
Power Consumption [W]	1.7 / 2.6 (all Relays on)	1.7	1.7 / 2.7	1.7 / 2.5	1.7 / 2.5
Universal Input (UI)	8	8	6	6	7 + 1 x Pressure Sensor
Digital Input (DI)	2	12	-	-	-
Analog Output (AO)	2	-	6	6	4
Digital Output (DO)	8 (4 x Relay, 4 x Triac)	-	8 (8 x Relay)	5 (4 x Relay 16 A, 1 x Relay 6 A)	7 (5 x Relay, 2 x Triac)
Digital output specification	Relay: 6 A Triac: 1 A @ 24-230 VAC	-	Relay: 6 A	Relay: 16 A and 6 A	Relay: 6 A Triac: 1 A @ 24-230 VAC
Internal shunt available for current measurement	UI5 & UI6, UI7 & UI8	UI5 & UI6, UI7 & UI8	UI3 & UI4, UI5 & UI6	UI3 & UI4, UI5 & UI6	UI5 & UI6, UI7
Supported STId card readers	3	2	-	-	1

15.4 Specification for LIOB-48x Models

Dimensions [mm]	107 x 100 x 75 (L x W x H)				
Operating Temperature (ambient)	0°C to +50°C				
Storage Temperature	-10°C to +85°C				
Humidity (non condensing) operating / storage	10 to 90 % RH @ 50°C				
Environmental Protection	IP 40 (enclosure); IP 20 (screw terminals)				
Power Supply	24 VDC / 24 VAC ±10 %				
Installation	DIN rail mounting (EN 50 022)				
Interface	1 x IP-852				
Types	LIOB-480	LIOB-481	LIOB-482	LIOB-483	LIOB-484
Power Consumption [W]	1.7 / 2.6 (all Relays on)	1.7	1.7 / 2.7	1.7 / 2.5	not yet available
Universal Input (UI)	8	8	6	6	not yet available
Digital Input (DI)	2	12	-	-	not yet available
Analog Output (AO)	2	-	6	6	not yet available
Digital Output (DO)	8 (4 x Relay, 4 x Triac)	-	8 (8 x Relay)	5 (4 x Relay 16 A, 1 x Relay 6 A)	not yet available
Digital output specification	Relay: 6 A Triac: 1 A @ 24-230 VAC	-	Relay: 6 A	Relay: 16 A and 6 A	not yet available
Internal shunt available for current measurement	UI5 & UI6, UI7 & UI8	UI5 & UI6, UI7 & UI8	UI3 & UI4, UI5 & UI6	UI3 & UI4, UI5 & UI6	not yet available
Supported STId card readers	3	2	-	-	not yet available

15.5 Specification for LIOB-58x Models

Dimensions [mm]	107 x 100 x 75 (L x W x H)				
Operating Temperature (ambient)	0°C to +50°C				
Storage Temperature	-10°C to +85°C				
Humidity (non condensing) operating / storage	10 to 90 % RH @ 50°C				
Environmental Protection	IP 40 (enclosure); IP 20 (screw terminals)				
Power Supply	24 VDC / 24 VAC ±10 %				
Installation	DIN rail mounting (EN 50 022)				
Interface	1 x BACnet/IP				
Types	LIOB-580	LIOB-581	LIOB-582	LIOB-583	LIOB-584
Power Consumption [W]	1.7 / 2.6 (all Relays on)	1.7	1.7 / 2.7	1.7 / 2.5	not yet available
Universal Input (UI)	8	8	6	6	not yet available
Digital Input (DI)	2	12	-	-	not yet available
Analog Output (AO)	2	-	6	6	not yet available
Digital Output (DO)	8 (4 x Relay, 4 x Triac)	-	8 (8 x Relay)	5 (4 x Relay 16 A, 1 x Relay 6 A)	not yet available
Digital output specification	Relay: 6 A Triac: 1 A @ 24-230 VAC	-	Relay: 6 A	Relay: 16 A and 6 A	not yet available
Internal shunt available for current measurement	UI5 & UI6, UI7 & UI8	UI5 & UI6, UI7 & UI8	UI3 & UI4, UI5 & UI6	UI3 & UI4, UI5 & UI6	not yet available
Supported STId card readers	3	2	-	-	not yet available

15.6 Resource Limits

Table 16 specifies the resource limits of the different L-IOB models.

Model Limits	18x	48x	58x
Total number of data points	2000	2000	2000
OPC Tags	-	200	200
User Registers	1000	1000	1000
IEC61131 Variables	1000	1000	1000
CEA-709 NVs	200	200	-
CEA-709 Alias NVs	200	200	-
CEA-709 Address table entries/non-ECS	256/15	256/15	-
BACnet Server Objects	-	-	200
BACnet Client Mappings	-	-	200
Calendar Patterns	25	25	25
Scheduler objects	10 (max. AST configuration size 384KB, 64 data points per scheduler)		10
Alarm Servers	1	1	32
Trend Logs	-	50	50
Total trended data points	-	100	100
Total aggregated size	-	6MB	6MB
E-mail templates	-	20	20
Math objects	-	20	20
Alarm Logs	-	5	5
Connections (local)	200	200	200
Connections (global)	-	100	100
L-WEB Clients (concurrent)	-	8	8
Extension with external L-IOB Modules	-	1 x LIOB-45x	1 x LIOB-55x

Table 16: Resource limits of different L-IOB models

16 References

- [1] NIC User Manual, LOYTEC electronics GmbH, Document № 88067217, May 2013.
- [2] LIOB-10x/x5x User Manual, LOYTEC electronics GmbH, Document № 88078509, August 2013.
- [3] L-INX User Manual, LOYTEC electronics GmbH, Document № 88073016, August 2013.

17 Revision History

Date	Version	Author	Description
2012-01-12	1.0	AB	Initial Version
2012-02-03	1.1	AB	Added Interpretations Fading and Ramping in Section 6.7.1.4, Added Section 6.7.1.6: DeadTime for Fading and Ramping, Added Section 6.7.2: STId Card Reader Mode.
2012-07-11	2.2	AB	Added LIOB-184 model and LIOB-48x models, Corrected LED patterns in Section 3.5.4, Added counting input options in Section 4.4. Corrected I/O specifications in Section 15.1, Added Chapter 14: Security Hardening Guide, Added Section 15.6: Resource Limits, Added Section 6.2: LONMARK® Device Mode, Added remarks about remote AST access for LIOB-18x in Section 6.6.
2012-12-14	4.8	AB	Added LIOB-58x (BACnet/IP) models, LIOB-48x/58x devices can be extended by one LIOB-45x/55x device, Added Section 5.6.4: L-IOB I/O Test Page, Updated contact information in Section 12.1, Added Section 15.2: Internal Translation Tables, Some updates and corrections of specifications in Chapter 15, Added applications notes for connecting I/Os in Chapter 13, Brought up-to-speed with L-INX user manual 4.8.
2013-08-01	4.9	AB	Added number of external L-IOB devices in Section 15.6, PWM period starts immediately when manual value changes, Section 6.7.1.4, Physical unit display for pulse count interpretation in Section 6.7.1.4, Added AnaInvert and Sqrt flags in Sections 6.7.1.12 and 6.7.1.13, Added switch data type for interlocked mode in Section 6.7.1.7, Only the most basic data points are created by default, Section 6.7.3, Added DP Create, OPC, and PLC flags in Section 7.17.1, MaxSendTime now also works from L-IOB host to device, Section 6.7.1.21, Improved description of MaxSendTime function in Section 6.7.1.21, Some translations tables are now included in the Configurator, Section 7.17.2, Faster sampling for analog inputs in Section 15.1.1, Added spike suppression in Section 6.7.1.25, Some general improvements regarding basic data points, Added I/O extension with LIOB-45x/55x devices in Section 3.5.5,

Date	Version	Author	Description
			Updated all screenshots, Added BACnet Accumulator object in Sections 6.9.7 and 7.17.1, Brought up-to-speed with L-INX user manual 4.9.