L-Proxy™

User Manual

LOYTEC electronics GmbH, Vienna

This page is intentionally left blank!

Contact

LOYTEC electronics GmbH
Stolzenthalergasse 24/3
A-1080 Vienna
AUSTRIA/EUROPE
support@loytec.com
http://www.loytec.com

Version 2.2

Document No. 88065704

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.

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.

L-Chip™, LC7093™ and L-PROXY™ are trademarks of LOYTEC electronics GmbH.

LonTalk©, LonWorks© and Neuron©, LonMaker and LNS are trademarks of Echelon Corporation registered in the United States and other countries.

Contents

1 Introduction

1.1 Overview

The L-Proxy is a high performance network infrastructure component for EIA-709 networks. It is the first device that allows network variable binding across multiple domains, SNVT translation, address table expansion for EIA-709 nodes, firewall security etc. The L-Proxy provides up to **five** communication ports and it translates packets between these ports. Its unique architecture very much supports well defined interfaces between various subsystems like HVAC, lighting, and security in a typical building.

The basic operation of L-Proxy is to take an input network variable on one side, interpret the data in the variable and transmit a corresponding output network variable on the other side. This concept is used to solve a great amount of today's communication problems in EIA-709 networks as explained in the following chapters.

L-Proxy is configured over the network with an easy to use configuration tool (plug-in) for LNS based network installation tools.

The L-Proxy is used for:

- Binding of network variables across multiple domains (up to 5 with a single device)
- Overcome the 15 destination addresses limit by expanding the address table in Neuron Chip based network nodes (up to 384 address table entries on each port)
- Save group addresses by breaking up groups into individual NV updates
- Translation between different SNVT types
- Firewall for EIA-709 networks
- Supports up to 384 network variables on each port
- Supports up to 512 alias-network variables on each port
- Supports up to 384 address table entries on each port
- Provides current time from built-in real-time clock

1.2 Scope

This document covers L-Proxy devices with firmware version 2.1. See Section 11 for differences between the different L-Proxy firmware versions.

2 What is a Proxy?

Webster's Revised Unabridged Dictionary (1913) lists the following entries for *Proxy*:

Proxy \Prox"y\, n.; pl. Proxies. [Contr. from procuracy. Cf. Proctor.]
1. The agency for another who acts through the agent; authority to act for another, esp. to vote in a legislative or corporate capacity.
2. The person who is substituted or deputed to act or vote for another.
3. A writing by which one person authorizes another to vote in his stead, as in a corporation meeting.

L-Proxy follows this definition. It acts on behave of somebody else. For L-Proxy the "somebody else" is another node in the network. The L-Proxy hardware offers 5 individual network ports as shown in Figure 1. One can think of L-Proxy as being 5 individual nodes that are somehow internally connected where each network port acts as an individual node. Whenever an input network variable on L-Proxy is being updated, L-Proxy updates a corresponding output network variable. Also if a network variable poll is received on the output network variable this poll can be forwarded to the input network variable. It doesn't matter if the output network is on the same port (node) or on one of the other four ports (nodes) as the input network variable. Since the 5 ports represent 5 individual nodes, which can be configured in different domains, it is now possible to bind network variables across domains.

Figure 1 L-Proxy offers five network ports, which represent 5 individual nodes.

We have listed use cases for L-Proxy that are typical in network installations. We use these use cases to explain the fundamental operation of L-Proxy. Remember that input and output network variables can reside on the same port (node) or different ports (nodes), in the same domain or in different domains. The behavior of L-Proxy is always the same.

2.1 Update-Update Use Case

Figure 2 shows an Update-Update use case. In this use case the input network variable of the L-Proxy is bound to an output network variable of Node 1. This output network variable is periodically updated. On the output side of L-Proxy the corresponding output network variable is also bound to an input network variable on Node 2, which is waiting for updates.

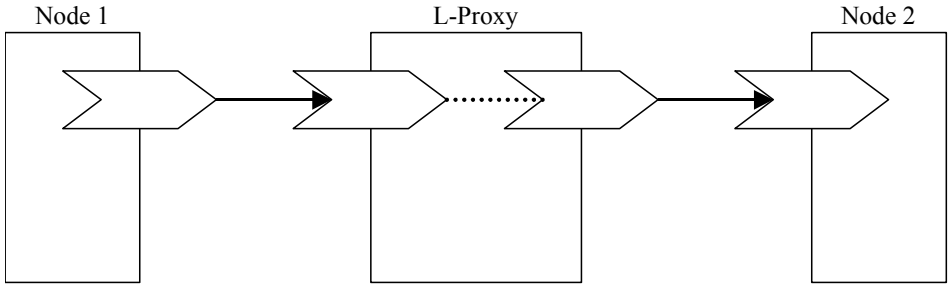


Figure 2: Update-Update use case

2.2 Poll-Poll Use Case

Figure 3 shows a Poll-Poll use case. In this use case the input network variable of L-Proxy is bound to an output network variable of Node 1. However, this output network variable on Node 1 is declared polled and never updated. Hence, Node 1 waits for incoming polls. On the output side of L-Proxy the corresponding output network variable is not bound. Node 2 periodically polls this network variable.

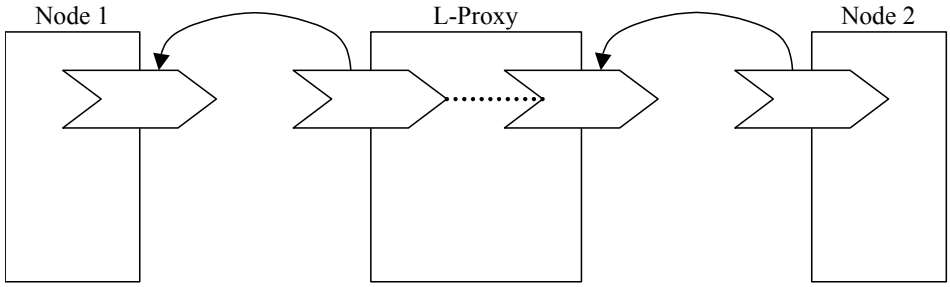


Figure 3: Poll-Poll use case

2.3 Update-Poll Use Case

Figure 4 shows an Update-Poll use case. In this use case the input network variable on L-Proxy is bound to an output network variable on Node 1. This output network variable on Node 1 is periodically updated. On the output side of L-Proxy the corresponding output network variable is not bound. Node 2 periodically polls this network variable.

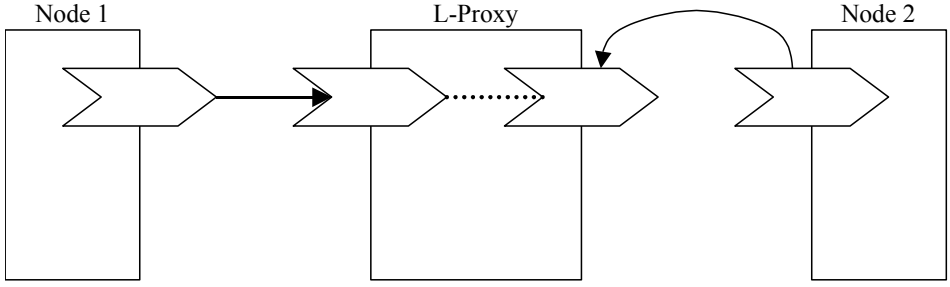


Figure 4: Update-Poll use case

2.4 Poll-Update Use Case (Active Poll)

Figure 5 shows a Poll-Update use case. In this use case the input network variable on L-Proxy is bound to an output network variable on Node 1. However, this output network variable on Node 1 is declared polled and never updated; Node 1 waits for incoming polls. On the output side of L-Proxy the corresponding output network variable is bound to an input network variable on Node 2, which is waiting for updates. This use case is also called active poll use case since L-Proxy actively polls the output network variable on Node 1.

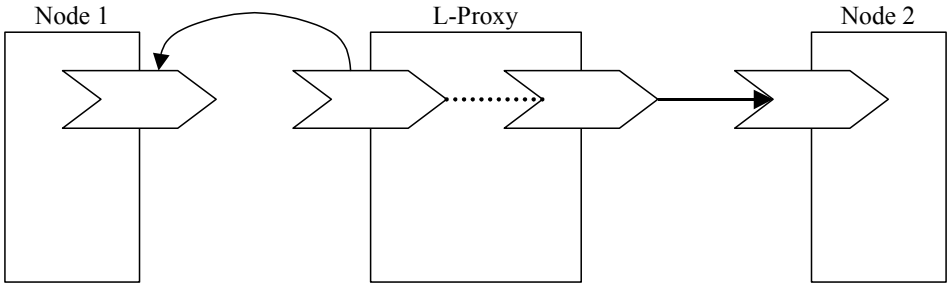


Figure 5: Poll-Update use case (active poll).

2.5 Use Case Combinations

There can be combinations of the different use cases. An example where Update-Update and Update-Poll are combined for one NV is shown in Figure 6. On the input side of L-Proxy an output NV is bound to the input NV of L-Proxy. Updates are received periodically. On the output side of L-Proxy an input network variable on Node 2 is bound to the output NV on L-Proxy (Update-Update use case). Further, an OPC server polls the same output NV periodically (Update-Poll use case).

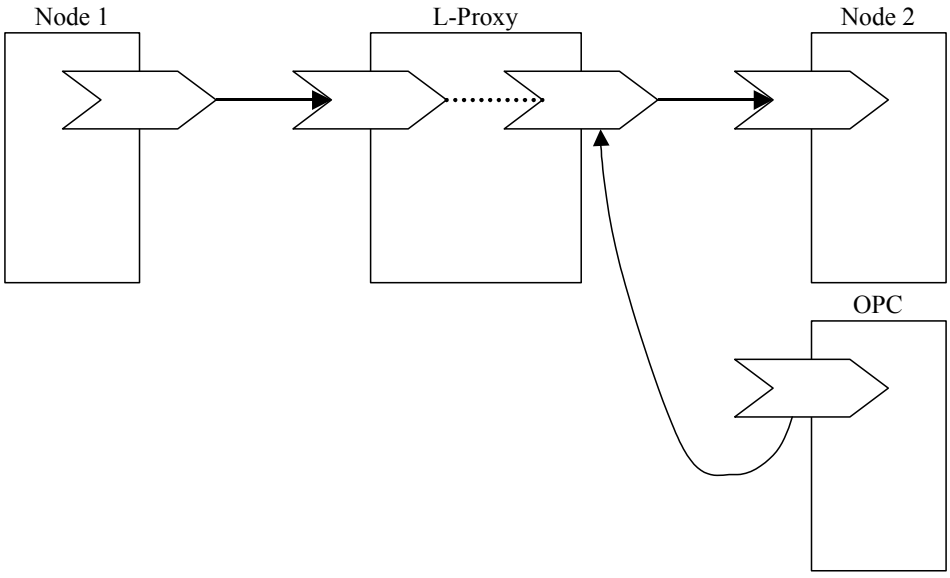


Figure 6: Combination of two basic use cases Update-Update and Update-Poll.

Another possible example for such a combination (Update-Update and Update-Poll) is shown in Figure 7. This use case is a variation of the Update-Update use case. The only difference is, that Node 2 does not wait for the first update to arrive after power-up or reset. Rather, it polls for an initial value after it was started and waits for updates thereafter.

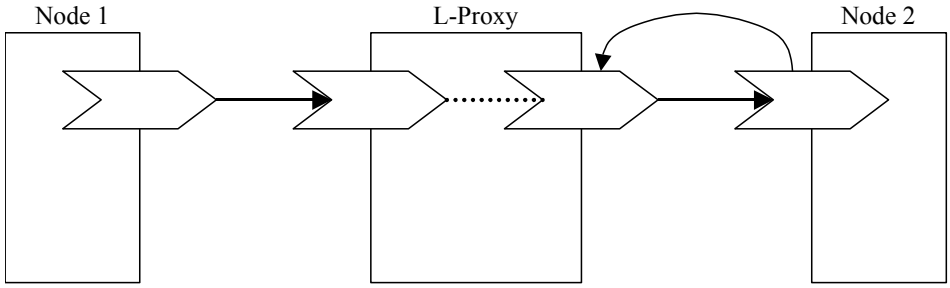


Figure 7: Combination where Node 2 gets an initial value through polling.

2.6 Poll Cache

If Node 2 doesn't receive network variable updates from the source node (Node 1) but rather polls Node 1 to update its input network variables, L-Proxy supports different network variable caching strategies in order to support a wide variety of desired behaviors.

Figure 8 shows the network variable *cache* memory inside L-Proxy. The cache can be used in different operating modes.

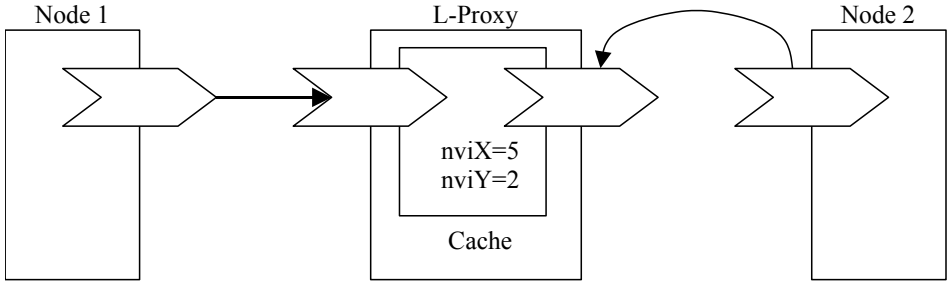


Figure 8: Poll cache inside L-Proxy. The cache is updated with every NV update and every poll response.

2.6.1 Passive Poll

Passive poll means that polling a network variable on Node 1 (see Figure 8) is always initiated by the destination node (Node 2) and never by L-Proxy itself. Passive poll mode supports the following cache strategies:

No Cache: Polls from Node 2 are directly forwarded to Node 1. A poll response is sent to Node 2 after the poll response was received from Node 1. If Node 1 doesn't respond to the poll request also L-Proxy doesn't respond to the poll request from Node 2. This behavior can be used to detect "dead" (not responding) nodes. This mode is typically used in poll-poll or update-update use cases.

Infinite Cache: The value in the cache never expires. Polls from Node 2 are never forwarded to Node 1 except after power-up when the cache value is not valid. The cache content is updated with network variable updates (no polls) from Node 1. Only after power-up of L-Proxy, when the cache entry is not valid, L-Proxy will forward poll requests from Node 2 to Node 1 until a valid value is stored in the cache.

Enable: The cache is enabled and a timeout in seconds must be specified. If the value in the cache is current (cache timeout has not expired since last update) the poll request from Node 2 reads the value from the cache. If the cache is expired the poll request from Node 2 will be forwarded to Node 1 and the L-Proxy will not respond to Node 2 until the poll response from Node 1 has been received. This mode can be used to detect dead nodes while keeping the network traffic low.

The Infinite-Cache and Enable Cache mode are typically used in update-poll or mixed with update-update use cases.

2.6.2 Active Poll

Active poll means that L-Proxy actively polls the source node (Node 1) and sends updates on the output network variables. The active poll can either occur only once at system startup or periodically at a specified poll rate. If in addition to the active polls from L-Proxy the destination node (Node 2) or some other node polls the source node (Node) 1, L-Proxy supports two strategies. This mode is used for the poll-update use case.

Forward polls: The poll request from Node 2 is interleaved with the active polls form L-Proxy in order to update the cache value right away. The poll response is sent to Node 2 after the poll response has been received from Node 1.

Use value from last poll cycle: The poll request from Node 2 returns the current value from the cache (last active poll cycle). The cache value is not updated.

Active polls can be used if a node cannot send updates (e. g. electrics meter) and the destination node (e. g. SCADA system) cannot poll a value. L-Proxy can then periodically poll the electric meter and send an NV update to the SCADA system.

2.6.3 Poll Attribute of NV

If an L-Proxy input network variable must poll the corresponding output network variable on the source node (Node 1) the input network variable poll attribute **must be set** when the network variable is created. In LonMaker set the appropriate flag as shown in Figure 9.

If polling is not used, don’t set the poll attribute for the dynamically created input NV!

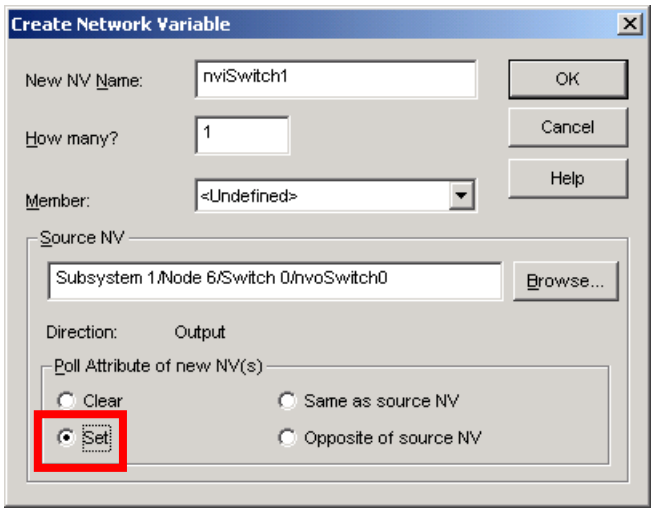


Figure 9 Set the poll attribute for the input network variable that needs to poll an output NV.

2.6.4 Poll strategy for multiple input network variables

In the case that multiple input network variables are connected to an output NV, the first input network variable that has the polled attribute set is polling its corresponding output NV. In the example in Figure 10 the poll request would only be forwarded to Node 1.

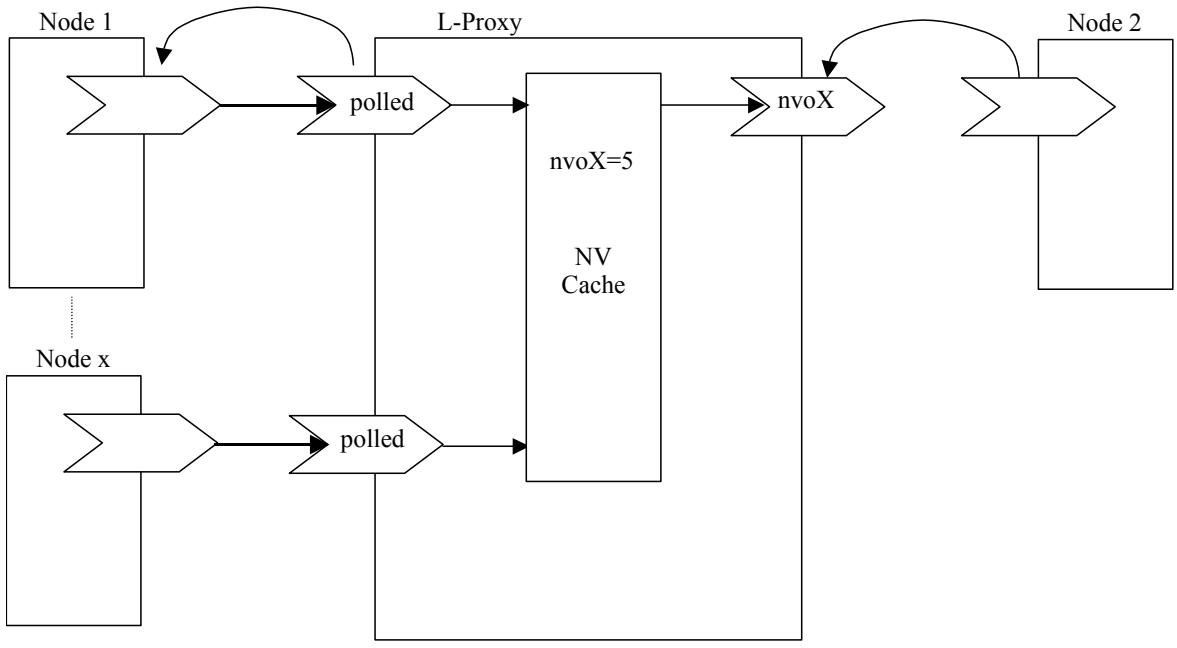


Figure 10 The first input NV with the polled attribute set updates the cache value.

2.7 SNVT Conversion

The L-Proxy starting with firmware version 2.0 supports automatic SNVT conversion for scalar types. In general there are the following requirements for SNVT conversion:

A. **Integer to Float conversion (I2F/F2I)**: This conversion type converts an integer value (signed or unsigned) to a float value (signed or unsigned) or vice versa. The type of both values must be a SNVT with the same SI unit (e.g. SNVT_temp to SNVT_temp_f).

B. **Integer to Integer conversion (I2I)**: This conversion type converts an integer value (signed or unsigned) to another integer value (signed or unsigned) or vice versa. The type of both values must be a SNVT with the same SI unit (e.g. SNVT_temp to SNVT_temp_p).

C. **Enumeration or Structure conversion**: This type converts one enumeration value to another enumeration value or one structure value to another one (e.g. SNVT_switch to SNVT_state). For each such a conversion a separate conversion function must be defined. This function may require some parameters (e.g. which bit in SNVT_state is to be converted). (currently not implemented)

Note! Currently L-Proxy supports Integer to Float and Integer to Integer conversion but does not support Enumeration or Structure conversion.

In case the conversion results in a value out of the range of the destination SNVT type, the constant for an invalid value is used, if such a value exists. Otherwise, the minimum resp. maximum value is used depending on which direction the valid rang of the SNVT type was exceeded. If the source value for a conversion is the constant for an invalid value, the destination NV value is set to 0.

Firmware version 2.0 supports the following SNVT conversions:

- SNVT_amp (1), SNVT_amp_ac (139), SNVT_amp_f (48), SNVT_amp_mil (2)
- SNVT_angle (3), SNVT_angle_deg (104), SNVT_angle_f (49)
- SNVT_angle_vel (4), SNVT_angle_vel_f (50), SNVT_rpm (102)
- SNVT_btu_f (67), SNVT_btu_kilo (5), SNVT_btu_mega (6)
- SNVT_count (8), SNVT_count_f (51), SNVT_count_inc (9), SNVT_count_inc_f (52)
- SNVT_density (100), SNVT_density_f (101)
- SNVT_elec_kwh (13), SNVT_elec_whr (14), SNVT_elec_whr_f (68)
- SNVT_flow (15), SNVT_flow_f (53), SNVT_flow_mil (16), SNVT_flow_p (161)
- SNVT_freq_f (75), SNVT_freq_hz (76), SNVT_freq_kilohz (77), SNVT_freq_milhz (78)
- SNVT_grammage (71), SNVT_grammage_f (72)
- SNVT_length (17), SNVT_length_f (54), SNVT_length_kilo (18), SNVT_length_micr (19), SNVT_length_mil (20)
- SNVT_lev_cont_f (55), SNVT_lev_percent (81)

- SNVT_mass (23), SNVT_mass_f (56), SNVT_mass_kilo (24), SNVT_mass_mega (25), SNVT_mass_mil (26)
- SNVT_ph (125), SNVT_ph_f (126)
- SNVT_power (27), SNVT_power_f (57), SNVT_power_kilo (28)
- SNVT_ppm (29), SNVT_ppm_f (58)
- SNVT_press (30), SNVT_press_f (59), SNVT_press_kilo (113)
- SNVT_pwr_fact (98), SNVT_pwr_fact_f (99)
- SNVT_res (31), SNVT_res_f (60), SNVT_res_kilo (32)
- SNVT_sound_db (33), SNVT_sound_db_f (61)
- SNVT_speed (34), SNVT_speed_f (62), SNVT_speed_mil (35)
- SNVT_temp (39), SNVT_temp_diff_p (147), SNVT_temp_f (63), SNVT_temp_p (105)
- SNVT_turbidity (143), SNVT_turbidity_f (144)
- SNVT_vol (41), SNVT_vol_f (65), SNVT_vol_kilo (42), SNVT_vol_mil (43)
- SNVT_volt (44), SNVT_volt_ac (138), SNVT_volt_f (66), SNVT_volt_kilo (46), SNVT_volt_mil (47)

3 Electrical and Mechanical Installation

3.1 Enclosure

The L-Proxy enclosure is 9 TE (1 TE = 17.5 mm) wide for DIN rail mounting, following DIN 43 880 (see Figure 11).

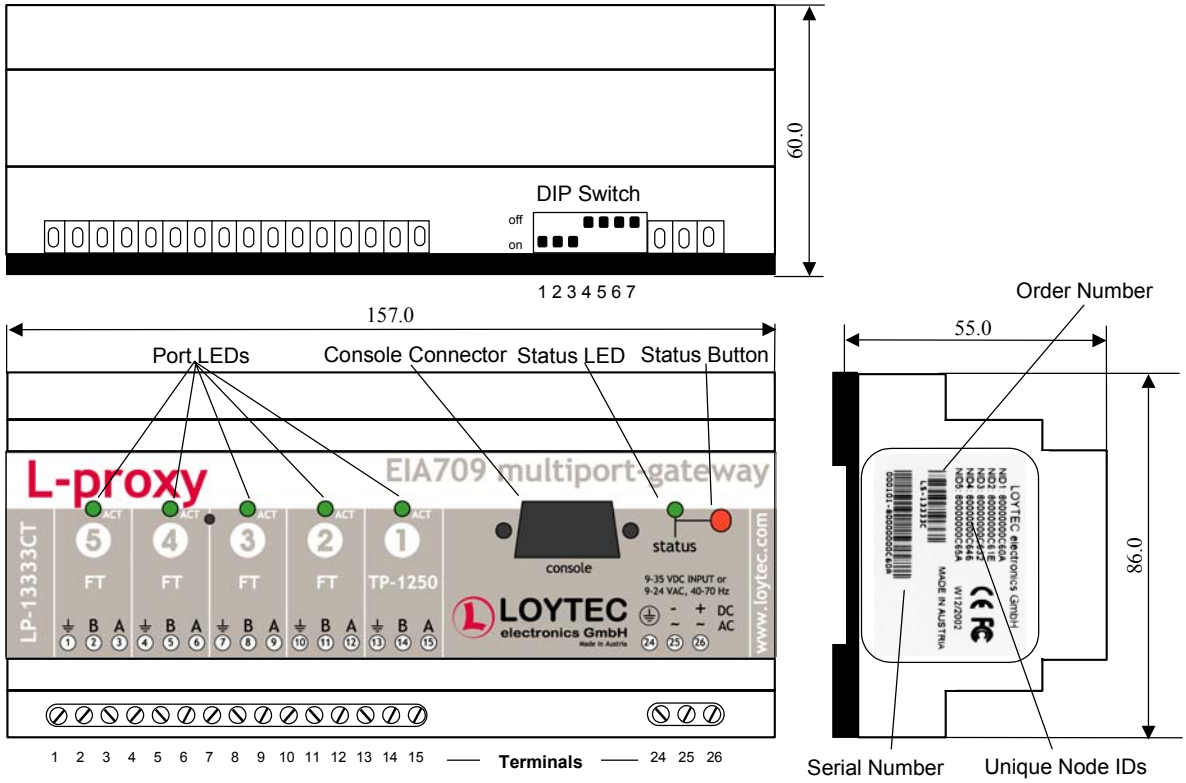


Figure 11: L-Proxy enclosure (dimensions in mm)

3.2 Product Label

The product label on the right side of the L-Proxy contains the following information (see Figure 11):

- L-Proxy order number with bar-code (Code 128, e.g. LP-13333CT)
- Serial number with bar-code (Code 128)
- Unique node ID of each port (NIDx)

An additional label is also supplied with the L-Proxy for documentation purposes.

3.3 Mounting

The device comes prepared for mounting on DIN rails following DIN EN 50 022. For wall mounting the L-Proxy is delivered with a piece of DIN rail matching the L-Proxy enclosure, which can be mounted on the wall. Then the L-Proxy is snapped on the rail.

The device can be mounted in any position. However, an installation place with proper airflow must be selected to ensure that the L-Proxy temperature does not exceed the specified range (see Section 11).

3.4 DIP Switch Settings

The L-Proxy uses 7 switches to select the mode of operation. For details see Table 1 and Chapter 7.

DIP Switch #	Function	Factory Default Settings
1	reserved	ON
2	reserved	ON
3	Channel auto detection On/Off (RS-485 ports only)	ON
4	Backbone arbitration On/Off	OFF
5	Station ID Bit 0 (LSB)	OFF
6	Station ID Bit 1	OFF
7	Station ID Bit 2 (MSB)	OFF

Table 1: DIP switch settings

3.5 Power Supply

The L-Proxy can either be DC or AC powered (Table 2).

Terminal	Function	Note
24	Main Earth Ground	
25, 26	Power Inputs	9-35 VDC or 9-24 VAC ± 10%

Table 2: Power Terminals

Note! Do not ground one of the power supply wires on terminal 25 and 26 (see Figure 13)!

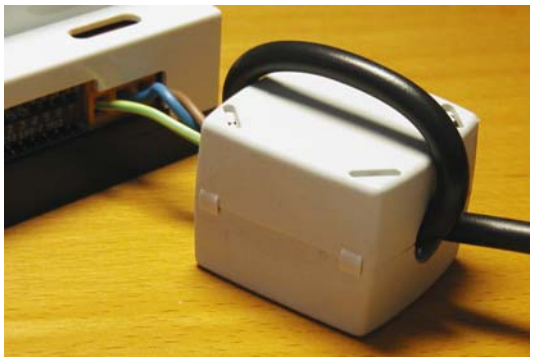


Figure 12: Attach the ferrite to the power cord

Attach the ferrite that comes with the L-Proxy to the power cord as shown in Figure 12. Make sure the power cord passes the ferrite twice.

The following power supplies are recommended for use with the L-Proxy:

Manufacturer: IDEC IZUMI CORPORATION
Manufacturer part number: PS5R-A12
Description: Power Supply, 12V, 7.5W, UL 508, CSA C22.2 No.14, EN60950, 100-240VAC
LOYTEC order number: LS-PS7W

Note: Switched power supplies like the IDEC IZUMI PS5R-A12 might interfere with power-line communication. If you are using power-line communication we strongly recommend a linear power supply or have the switched power supply tested against interference with power-line communication signals. The IDEC power supply is NOT recommended for use with power-line communication.

3.6 Connection Diagrams

The L-Proxy provides screw terminals to connect to the network as well as to the power supply. The screw terminals can be used for wires having a maximum thickness of 1.5 mm^2/AWG12.

3.6.1 LP-13333CT

Terminal	Function
1	Earth Ground
2, 3	EIA-709 A, B of FT-10 Channel Port 5
4	Earth Ground
5, 6	EIA-709 A, B of FT-10 Channel Port 4
7	Earth Ground
8, 9	EIA-709 A, B of FT-10 Channel Port 3
10	Earth Ground
11, 12	EIA-709 A, B of FT-10 Channel Port 2
13	Earth Ground
14, 15	EIA-709 A, B of TP-1250 Channel Port 1
24	Main Earth Ground
25, 26	Power Supply

Table 3: L-Proxy Terminals LP-13333CT

3.7 Wiring

Every network segment connected to the L-Proxy needs to be terminated according to the rules found in the specification of the transceiver (see Chapter 3.8).

Important: All used and unused ports must be properly terminated. For unused ports, it is recommended to use a 100 Ohm 0.25 W resistor between terminals A and B as termination (see Figure 13).

Important: All Earth ground terminals must be connected to the main Earth ground terminal 24. When using shielded network cables only one side of the cable should be connected to ground. Thus, the shield must be connected to earth ground either at the L-Proxy terminals or somewhere else in the network, but never at more than one place (see Figure 13)!

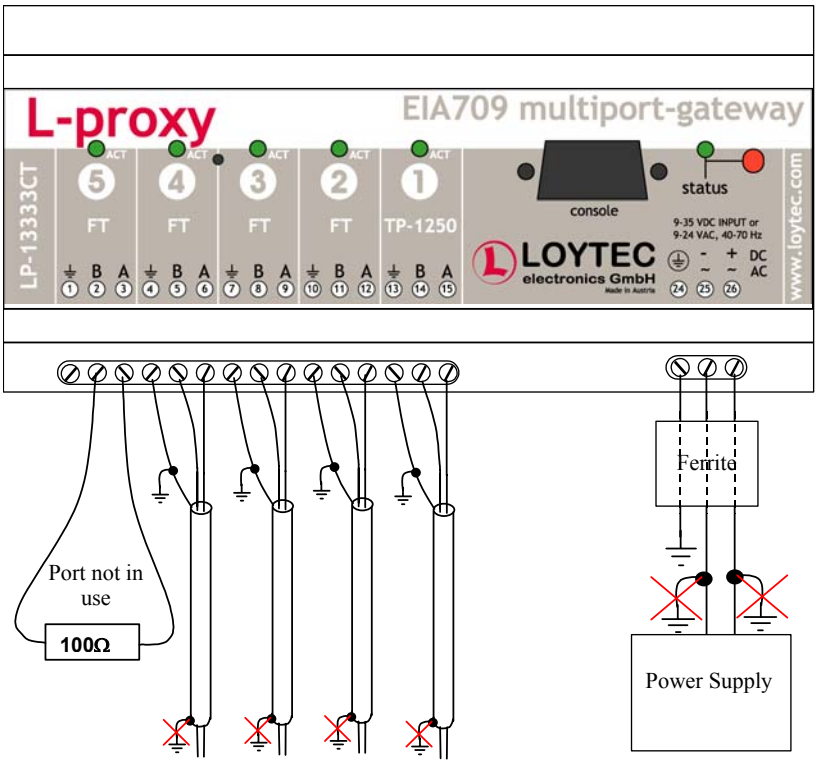


Figure 13: Connecting the L-Proxy (LP-13333CT)

Figure 13 shows an L-Proxy where Port 5 is not used.

3.8 Communication Media

3.8.1 TP-1250

The TP-1250 uses transformers for galvanic isolation. The topology of a TP-1250 network is a bus. Thus, both ends of the bus cable need to be terminated with a termination network as shown in Figure 14.

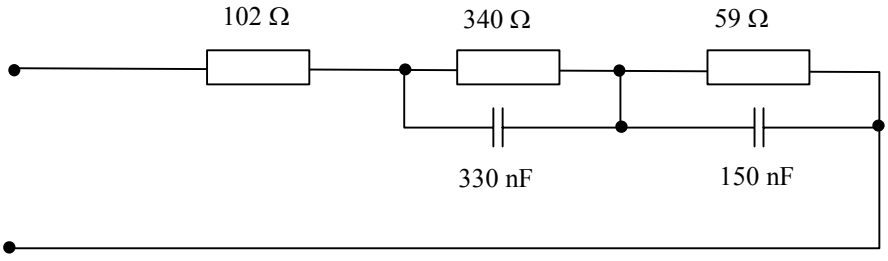


Figure 14: TP-1250 Termination Network

If backbone mode is disabled, the L-Proxy TP-1250 ports are fully compatible to the parameters specified by LonMark for this channel (TP/XF-1250). If backbone mode is enabled, proprietary channel parameters are used. In this case no Neuron Chip based nodes or other nodes with standard TP-1250 communication parameters are permitted on the same channel.

3.8.2 FT-10

The L-Proxy FT-10 ports are fully compatible to the parameters specified by LonMark for this channel. FT-10 ports can also be used on Link Power (LP-10) channels. However, the L-Proxy does not provide the power supply for Link Power channels.

When using the Free Topology Segment feature of the FT-10, only one termination (Figure 15) is required and can be placed anywhere on the free topology segment.

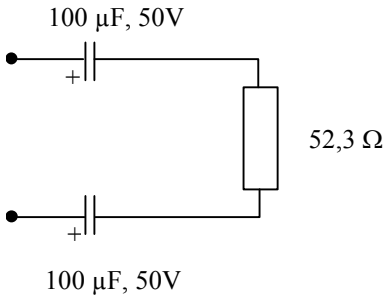


Figure 15: FT-10 Free Topology Termination

In a double terminated bus topology, two terminations are required (Figure 16). These terminations need to be placed at each end of the bus.

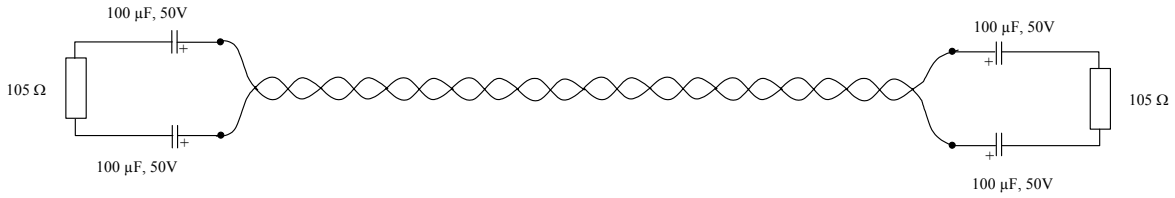


Figure 16: Termination in an FT-10 Bus Topology

4 L-Proxy Configuration Utility (Plug-In)

The L-Proxy configuration utility is used to create the internal connections between input and output network variables on L-Proxy. It is also used to specify configuration properties when network variables are polled rather than updated. The configuration utility is installed as a plug-in tool for all LNS based network management tools.

System requirements:

- LNS 3, Service Pack 7 or higher
- Network management tool that supports dynamic network variables (e.g. LonMaker 3.1 or higher, Newron NL-220, Spega Alex 3)
- Windows XP, Windows 2000, Windows NT® 4.0, Windows Me, and Windows 98.

4.1 Installation

To install the configuration utility double click on Setup and follow the installation steps.

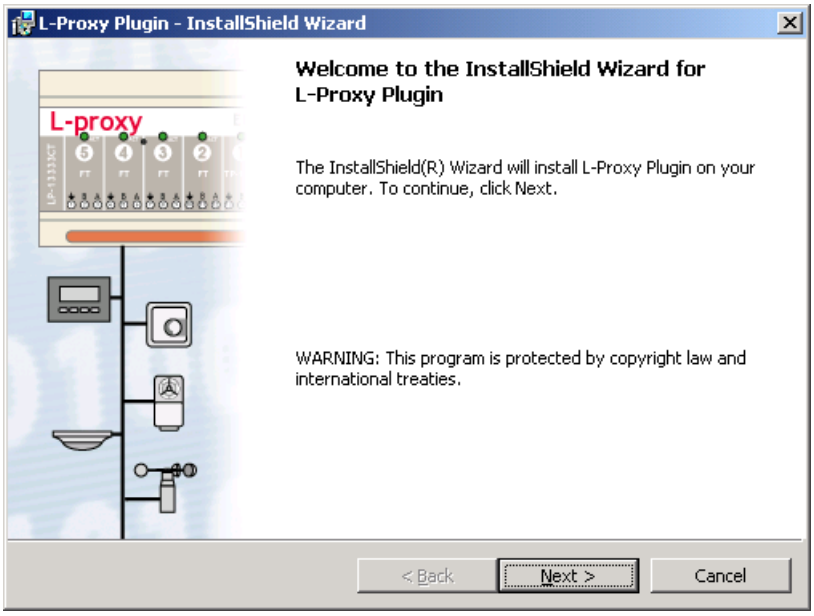


Figure 17 L-Proxy Plug-in welcome screen.

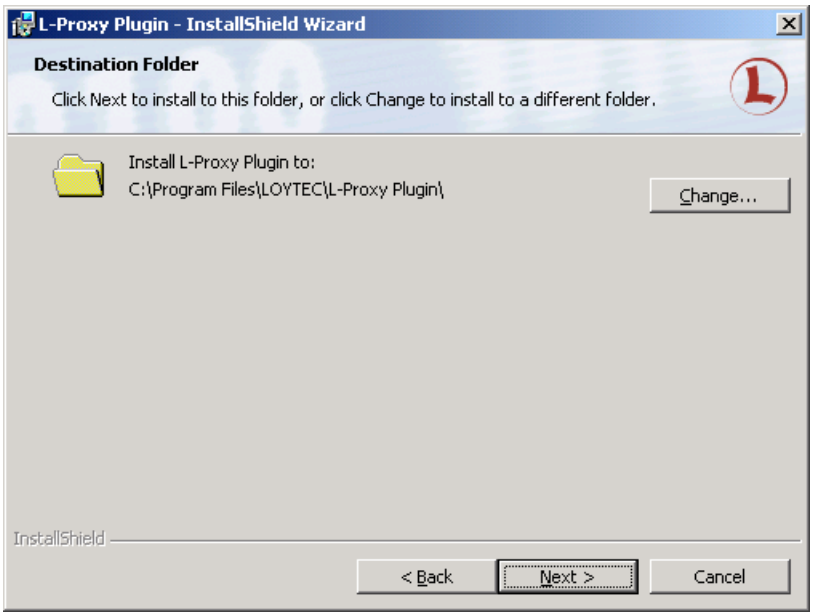


Figure 18 Choose the destination directory.

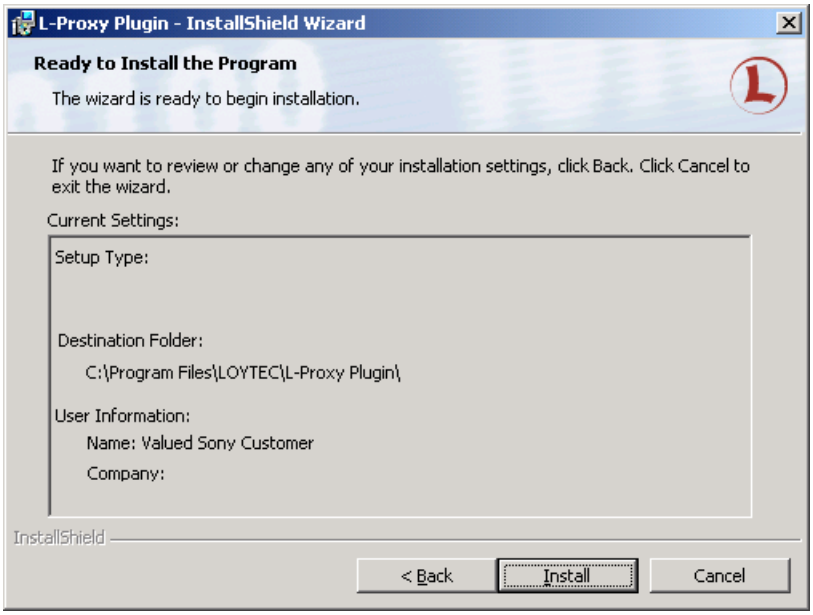


Figure 19 Click on Install to install the Plug-in.

4.2 Register in LonMaker

After successfully installing the L-Proxy configuration utility the program must be registered as a plug-in in LonMaker. Open LonMaker and create a new network. When the Plug-in Registration Dialog window pops up select the **L-Proxy Configuration Plugin** from the list of "Not Registered Plug-Ins". Device templates for L-Proxy are added automatically and XIF files are copied into the LNS import directory.

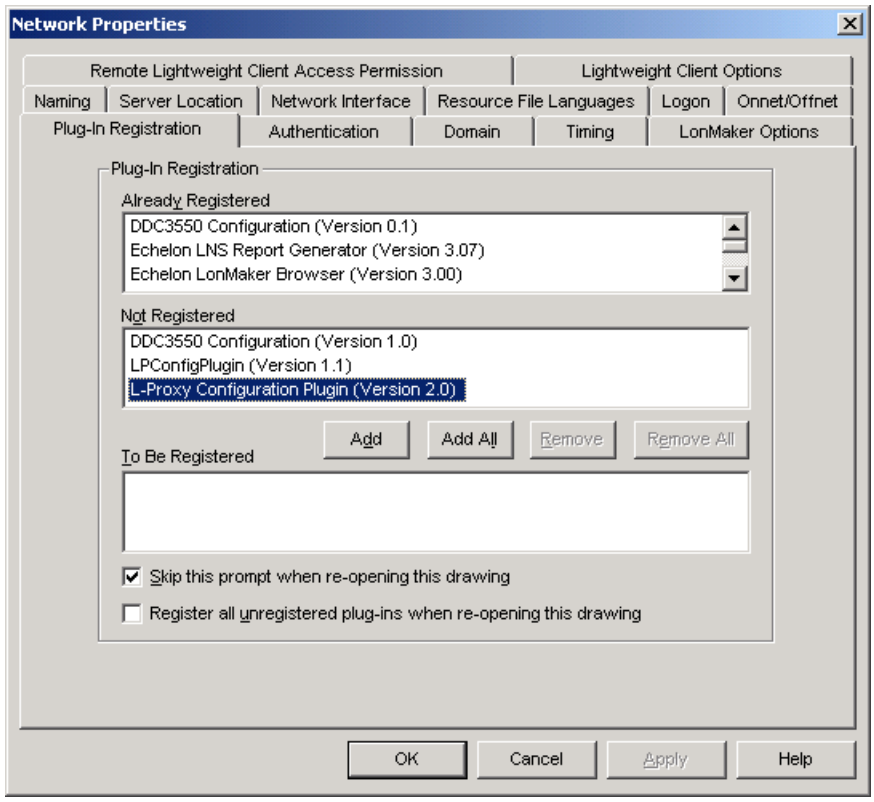


Figure 20 Select the Plug-in to be registered and click Add.

Click *Add*

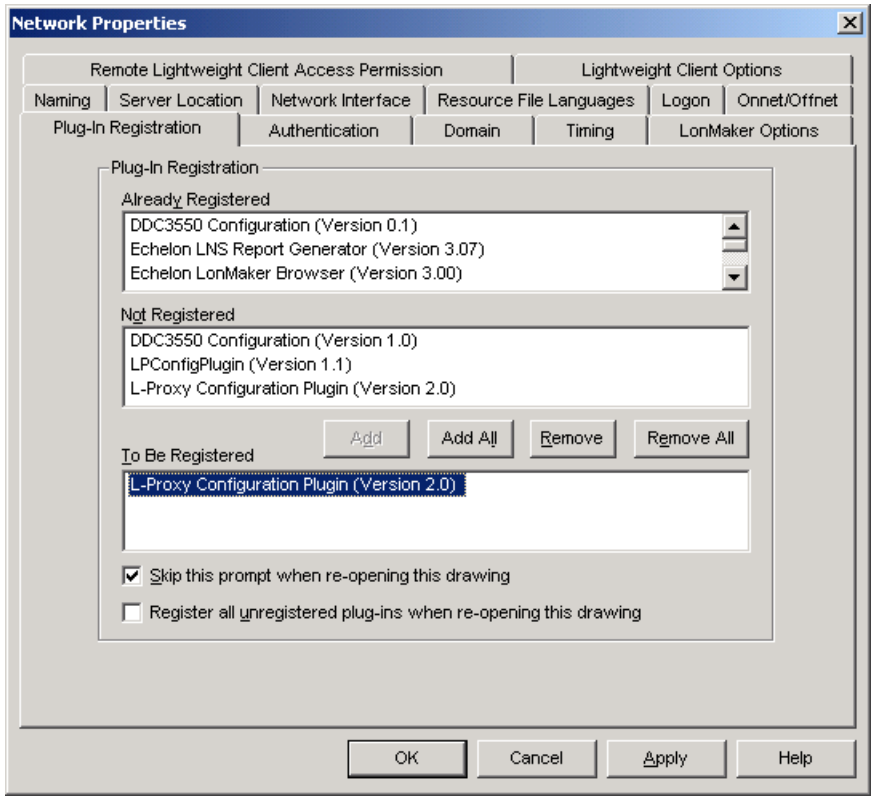


Figure 21 Click OK to register the L-Proxy configuration plug-in.

Click *OK*

Note! *If you are using multiple databases (projects) make sure you have registered the plug-in in each project.*

Under LonMaker => Network Properties => Plug-In Registration make sure that the **L-Proxy Configuration Plugin** shows up under "Already Registered".

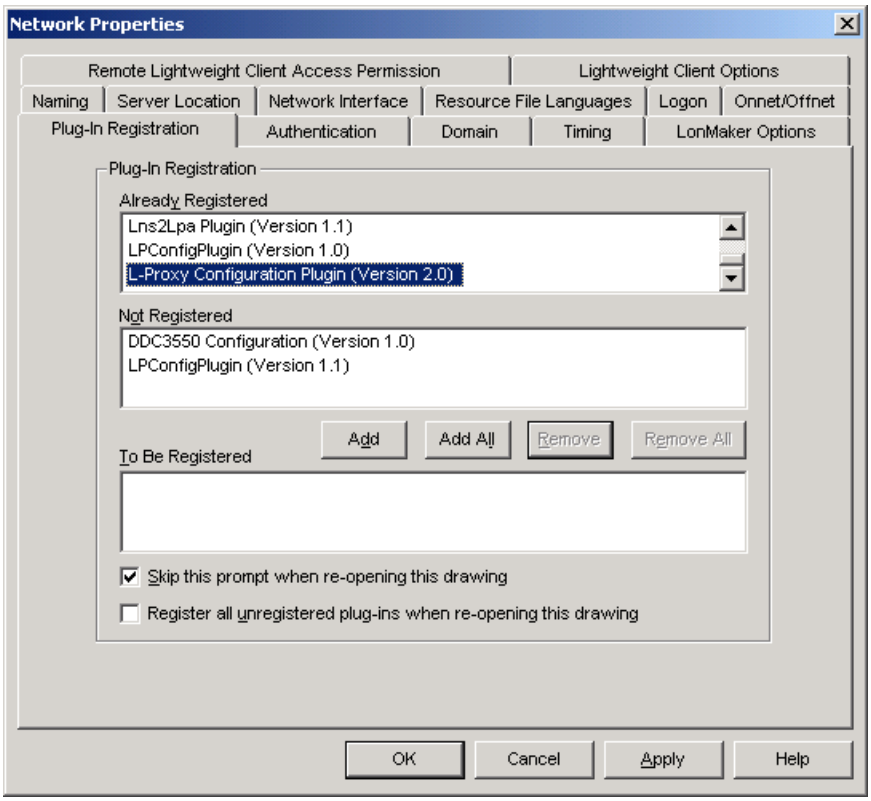


Figure 22 Double check that the L-Proxy plug-in is properly registered.

4.3 Register in NL-220

After successfully installing the L-Proxy configuration utility the program must be registered as a plug-in in NL220. Open NL220 and create a new project. When the project is opened, make sure the option **Simple command string** is checked in the **PlugIns** menu. If not checked, check it by selecting the option. Then use the option **Register plugins** in the **PlugIns** menu.

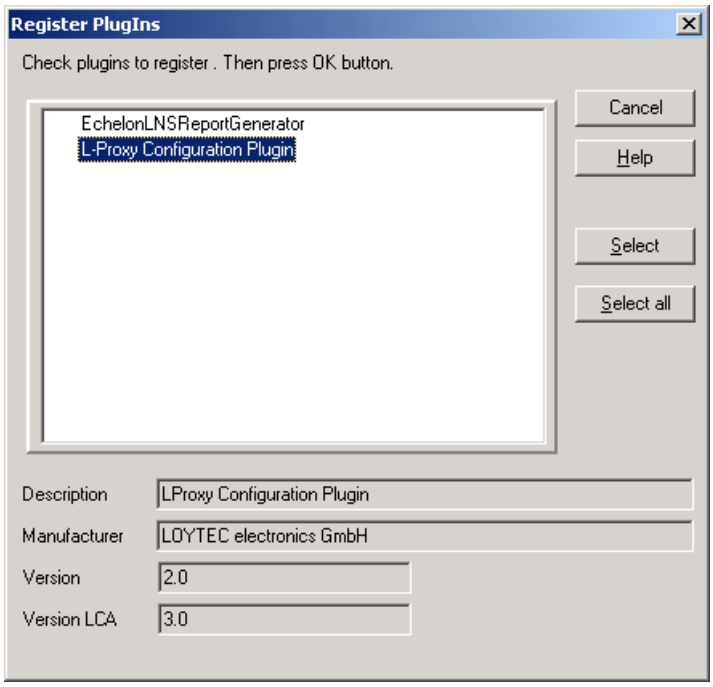


Figure 23 Check the Plug-in to be registered and click **OK**.

Click *OK*

Note! *If you are using multiple databases (projects) make sure you have registered the plug-in in each project.*

Select again the option **Register plugins** in the **PlugIns** menu. Make sure the L-Proxy Configuration Plugin is preceded by a ✔ sign.

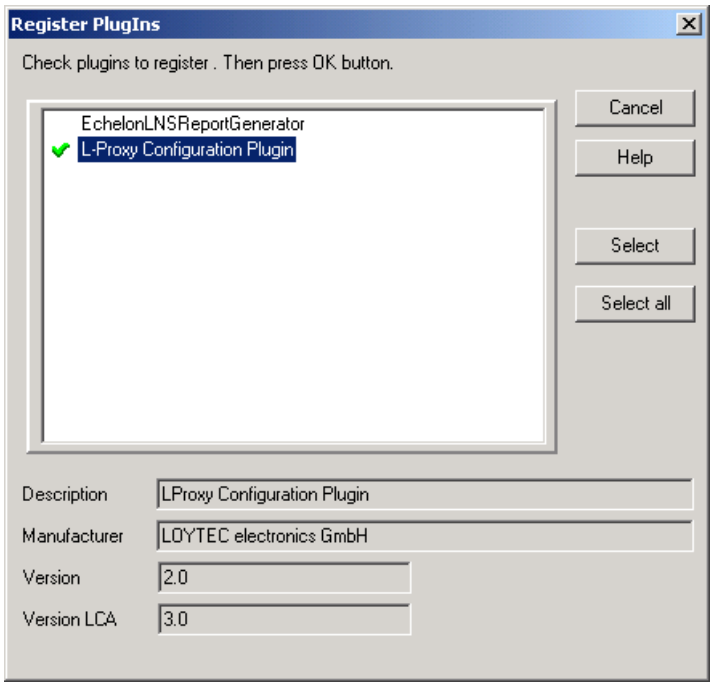


Figure 24 Double check that the L-Proxy plug-in is properly registered.

4.4 Register in Alex

After successfully installing the L-Proxy configuration utility the program must be registered as a plug-in in Alex. Open Alex and create a new project or open the project to which a L-Proxy device should be added. In the tree view, perform a right click on the new project and select **Registriere PlugIn** and select **L-Proxy Configuration Plugin.** The registration process of the plugin also generates the L-Proxy device templates automatically.

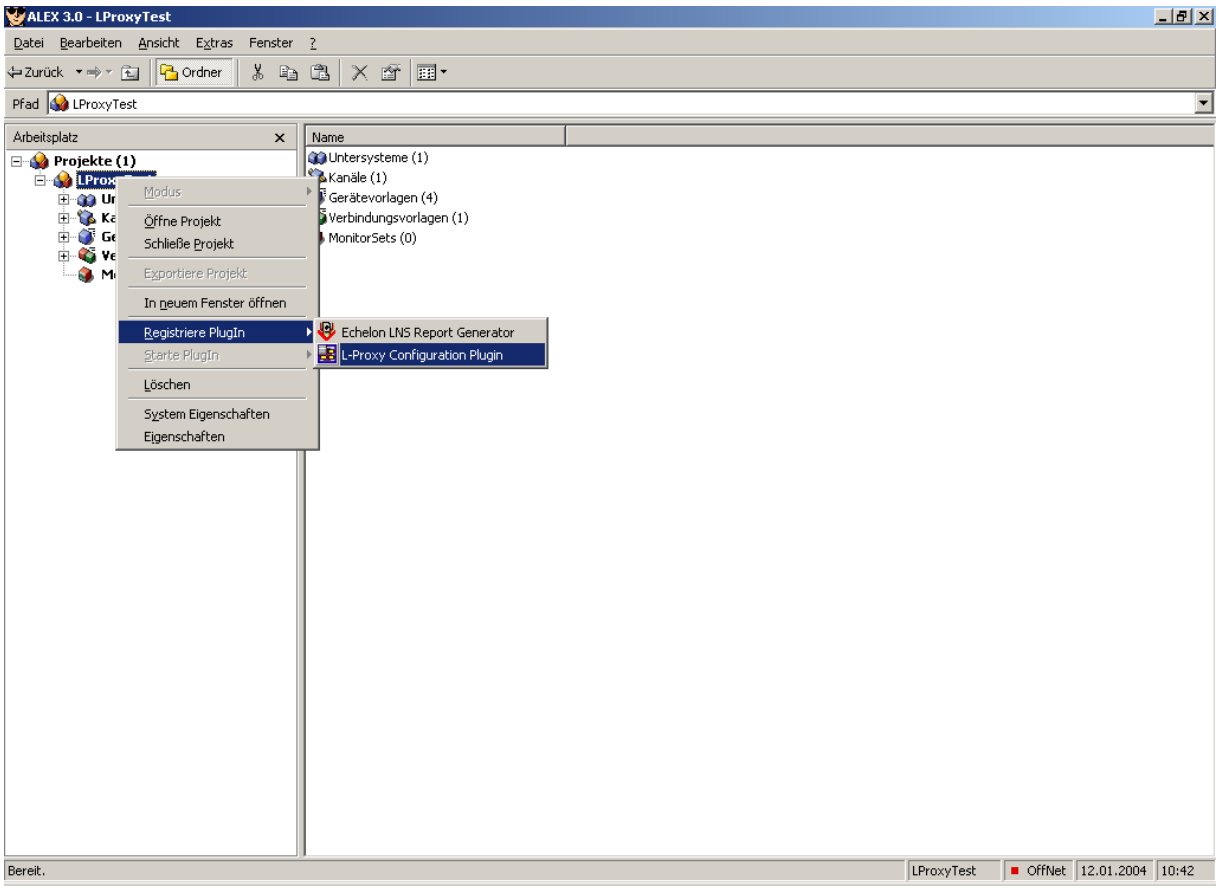


Figure 25 Register the L-Proxy Configuration Plugin

Note! *If you are using multiple databases (projects) make sure you have registered the plug-in in each project.*

4.5 Operating modes of the configuration utility

The L-Proxy configuration utility 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 configuration tool.

4.5.1 On-line mode

This is the preferred method to use the configuration utility. 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 to download the L-Proxy configuration into the device.

4.5.2 Off-line mode

The off-line mode can be used to add the device using the device templates, extract the port interface definition and to make the internal connections.

4.5.3 Stand-alone mode

The L-Proxy configuration utility can also be executed as a standalone program. In this operating mode one can create the L-Proxy internal connections (see Section 5.6.4) but it is not possible to extract the interface definition (see Section 5.6.1). This mode is very useful for the engineer who makes the time consuming bindings (connections) between the different L-Proxy ports since he doesn't need a network management tool like e.g. NL-220, LonMaker or Alex.

5 L-Proxy in a Network

This Chapter gives step-by-step instructions on how to commission L-Proxy, create multiple *Proxy* functional blocks, create input and output network variables, and make L-Proxy internal bindings. We have shown the configuration steps using LonMaker 3.1, NL-220 and Alex 3 but other LNS based network management tools can be used as well to install and configure L-Proxy.

5.1 Network Buffers

The L-Proxy can handle packets from the network with a maximum length of 256 bytes. There is no explicit limit in the network buffer counts.

5.2 Design Flow with L-Proxy

The flow diagram in Figure 26 shows the steps that need to be followed in order to install L-Proxy in a network. After adding one L-Proxy port one or more functional block must be created and the input and output network variable must be created dynamically. Then the binding with the other nodes in the network can be carried out. After all network variables have been created one must start the L-Proxy configuration utility Plug-In and extract the port definition for every installed L-Proxy port.

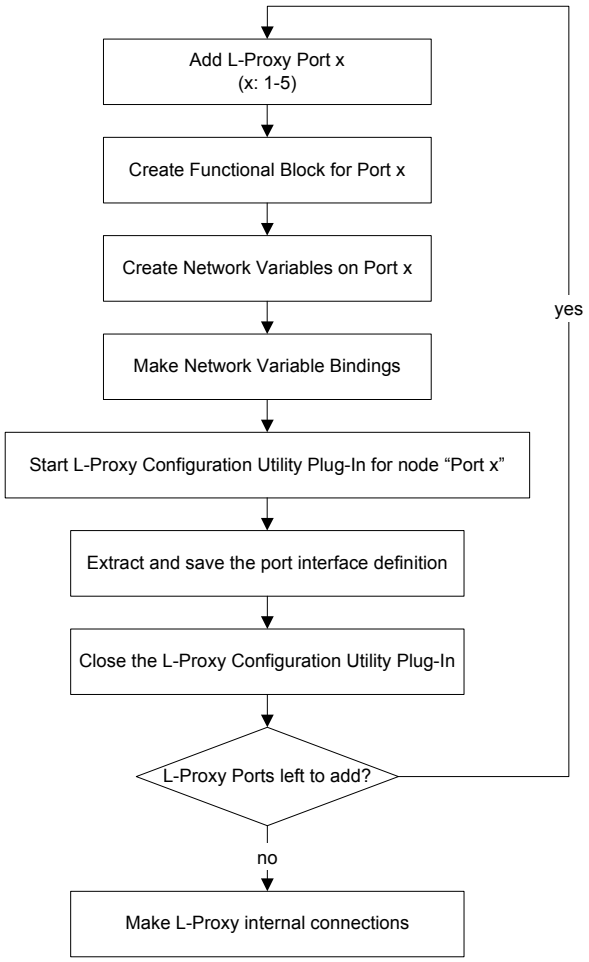


Figure 26 Basic design-flow on how to add and bind L-Proxy in a network.

After all interfaces from the L-Proxy ports in use have been extracted the L-Proxy internal connections must be created. Therefore the L-Proxy Plug-In can be started on any of the 5 ports (nodes). First the port interface definition files *name.lpi* created in a previous step must be loaded. Then the L-Proxy internal connections can be created. This configuration can be saved in an L-Proxy project file *projectname.lpp*. Before downloading the new L-Proxy configuration into L-Proxy all L-Proxy ports in use must be commissioned. If the L-Proxy device is online the new configuration can now be downloaded from the plug-in.

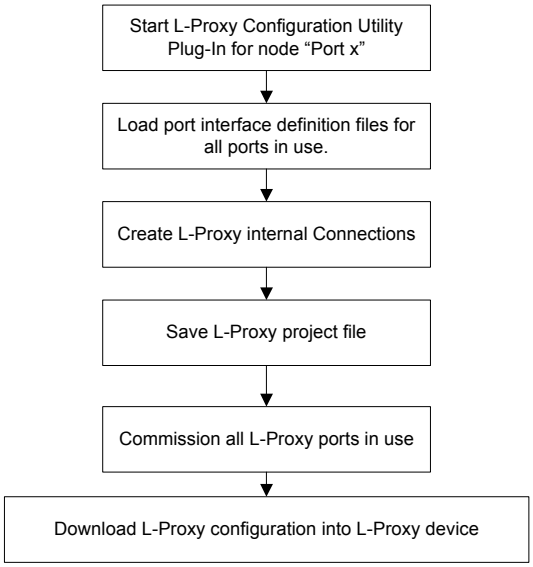


Figure 27 Creating L-Proxy internal connections.

5.3 Adding L-Proxy

This paragraph shows the basic steps that need to be followed in order to add L-Proxy in a network management tool. Before you continue please make sure that you have installed the L-Proxy configuration utility as explained in Chapter 4.

5.3.1 Adding L-Proxy in LonMaker

Drag a new device shape onto the drawing area. In the dialog from Figure 28 specify a device name for the new device e.g. *L-Proxy1 Port 2* in case you have multiple L-Proxies in the network.

Select: *Commission Device*

Click *Next*

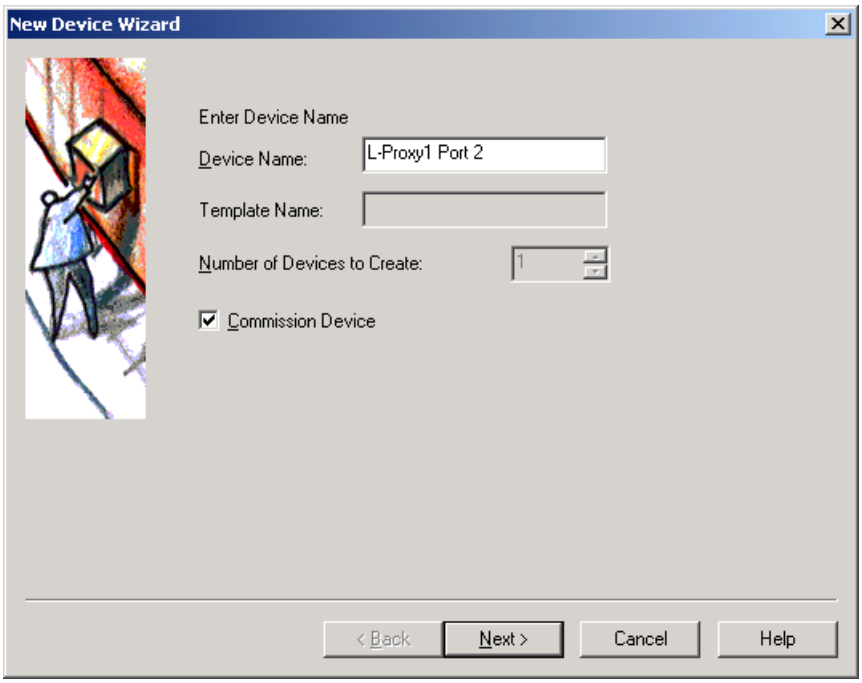


Figure 28 Adding a new L-Proxy port (Port 2) to the network.

In the next dialog window select *Existing Template* and select the proper template name as shown in Figure 29. Please use the proper template for FT10 and for the TP-1250 network port.

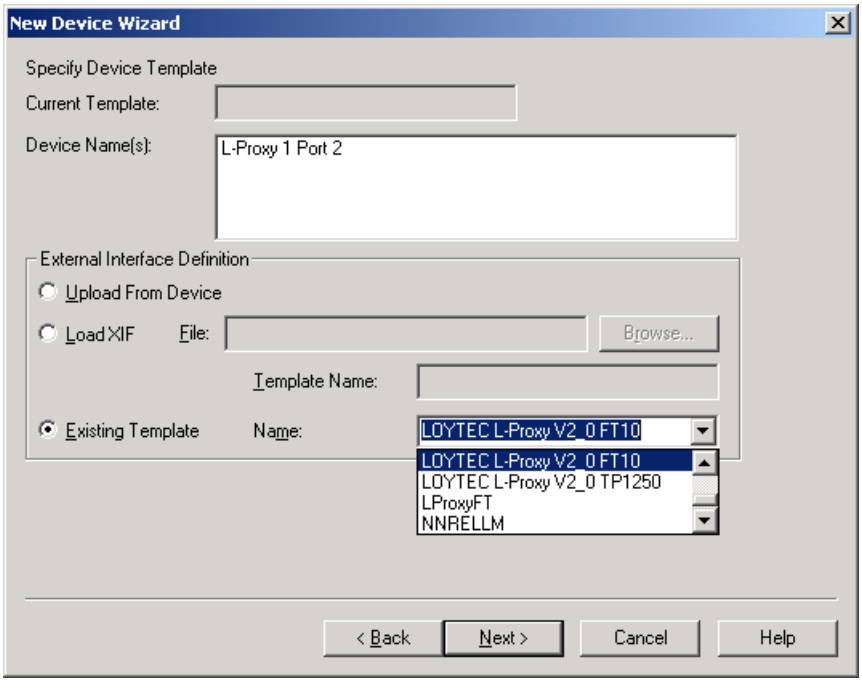


Figure 29 Specify the following Template Names.
FT port (Port 2-5): LOYTEC L-Proxy V2_0 FT10
TP-1250 port (Port 1): LOYTEC L-Proxy V2_0 TP1250

Click *Next*

Select the channel to which L-Proxy Port 2 should be connected.

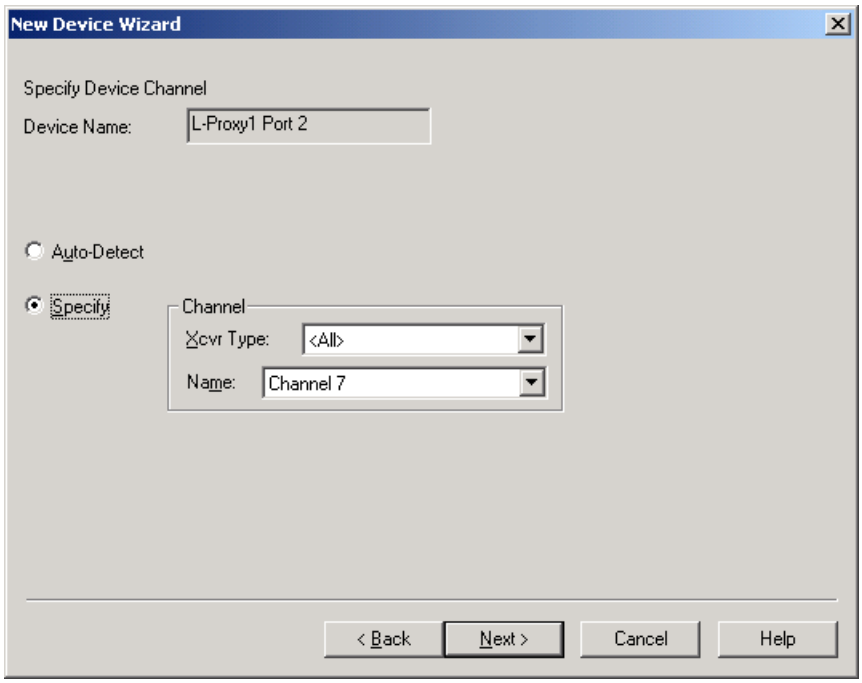


Figure 30 Select the proper channel for L-Proxy Port 2.

Click *Next*

Click *Next*

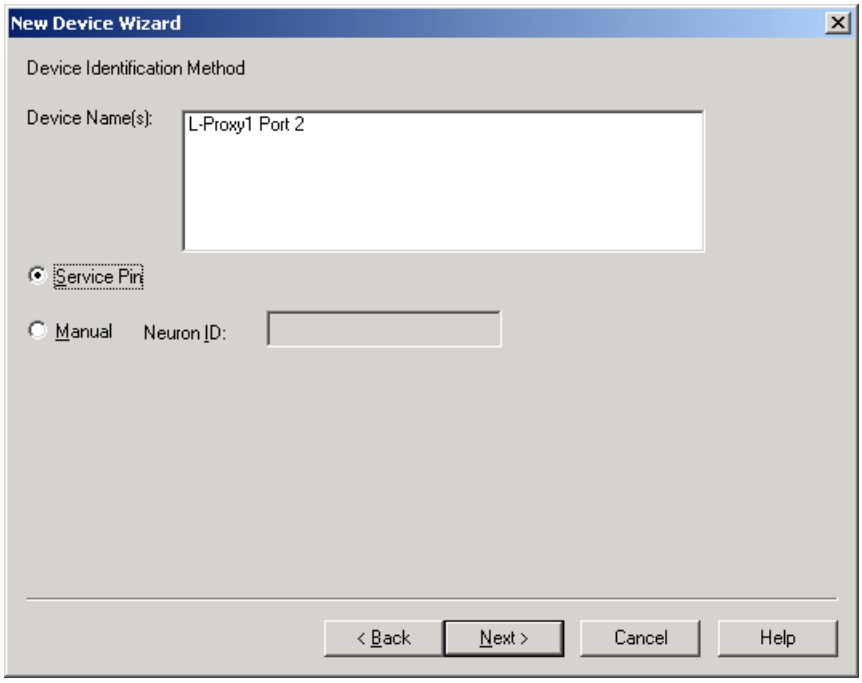


Figure 31 Select Service Pin or Manual. The Neuron ID of the 5 ports are printed on a label that comes with L-Proxy.

Click *Next*

Click *Next*

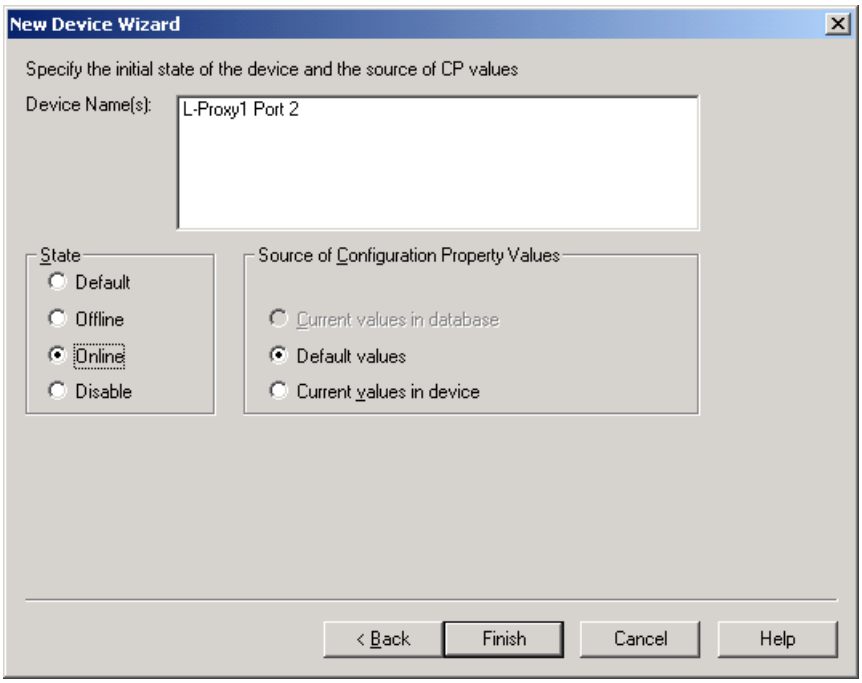


Figure 32 Choose State: Online

Click *Finish*

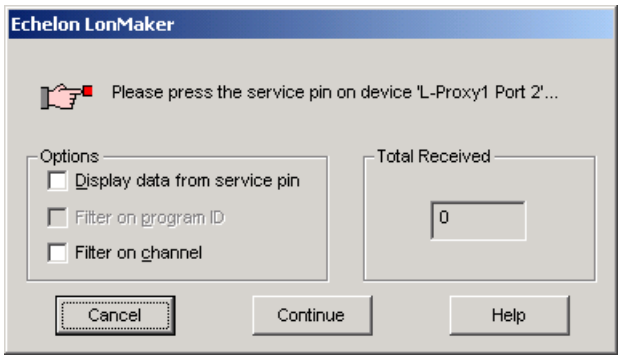


Figure 33 In order to use the Service Pin on L-Proxy one must press and hold the status button until the port LED for the desired port lights up yellow. Release the button and L-Proxy sends out the service pin message for this port.

In some installations the message in Figure 34 might appear. Click Yes, OK, and Continue to finish the installation.

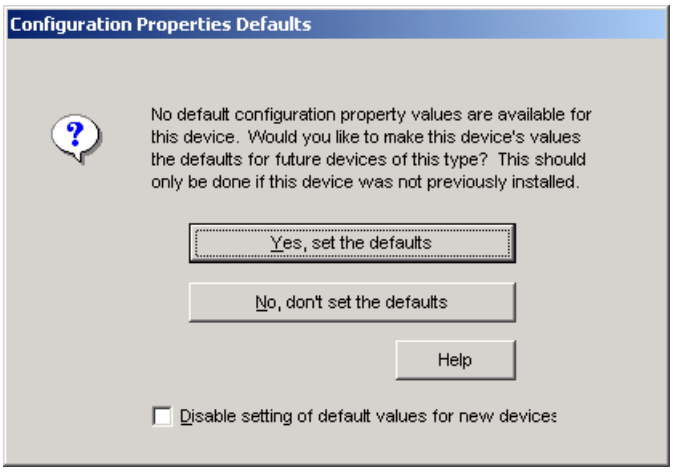


Figure 34 Click on Yes to continue.

Your L-Proxy node should now appear in the LonMaker drawing area.

5.3.2 Adding L-Proxy in NL-220

Make sure you are in the **All subsystems** tree. For this, the button  in **Trees toolbar** must be pressed (if not click on it).

Right click on the subsystem you want to add the device in and select the option **New node in** <…>

In the dialog from Figure 35 specify a device name for the new device e.g. *L-Proxy1 Port 2* in case you have multiple L-Proxies in the network.

Check the option **Create node from a device template**.

Select the proper device template for the L-Proxy Port. An FT port (Port 2-5) uses the LOYTEC L-Proxy V2_0 FT10 template and the TP-1250 port (Port 1) the LOYTEC L-Proxy V2_0 TP1250 template.

Click *OK*

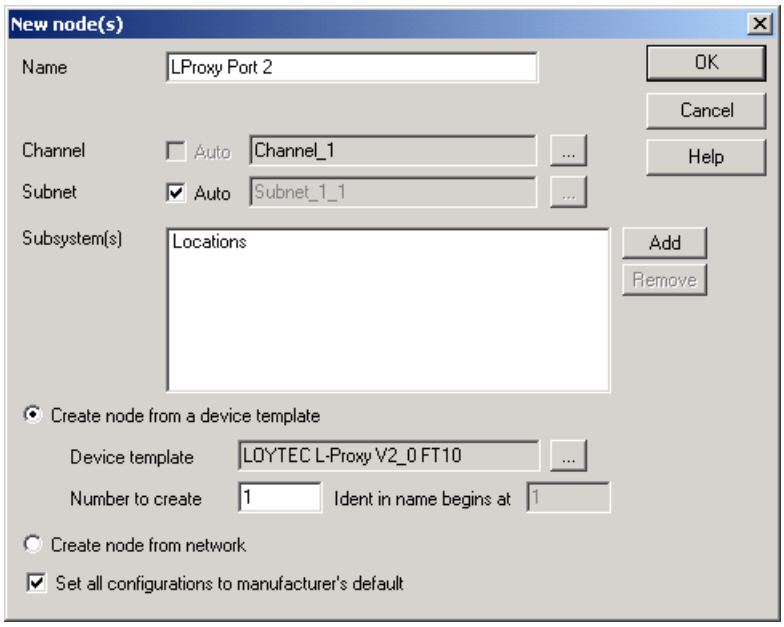


Figure 35 Adding a new L-Proxy port (Port 2) to the network.

Note : You may select the channel manually by un-checking **Auto** and clicking on ... . You will be able to select the channel you want to create the device in. If **Auto** is checked then **NL220** will try to automatically determine the channel of the device from the network.

Your L-Proxy node should now appear in the NL220 tree as shown in Figure 36. Its icon must be .

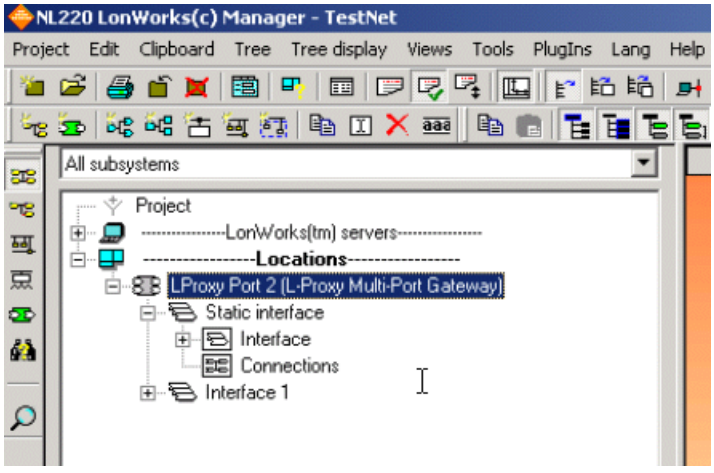


Figure 36 L-Proxy Interface.

5.3.3 Adding L-Proxy in Alex

In the tree view, go to the subsystem to which the L-Proxy port should be added. Select Geräte and choose Neues Gerät. The Dialog in Figure 35 is shown. Enter the device name (e.g. L-Proxy Port 1) and choose the correct device template for the L-Proxy. Make sure to slect the device template according to the L-Proxy port you are adding (TP-1250 version for port 1, FT-10 version for all other ports). If required, enter the Neuron ID or press the Service Pin and select *OK* to close the dialog and add the L-Proxy port to the project.

Figure 37 Adding a new L-Proxy port (Port 1) to the network.

5.4 Creating Functional Blocks/Virtual Interfaces

After adding the L-Proxy node at least one but up to five functional block/virtual interfaces must be created for the proxy object.

5.4.1 Creating Functional Blocks in LonMaker

Drag the Functional Block shape onto the drawing area. A dialog like the one in Figure 38 appears.

Under Device Name select *L-Proxy Port 2* *(we use Port 2 in this example)*

Under Functional Block Name select *Proxy 1* *(or Proxy 2, or Proxy 3, ...)*

Click *Next*

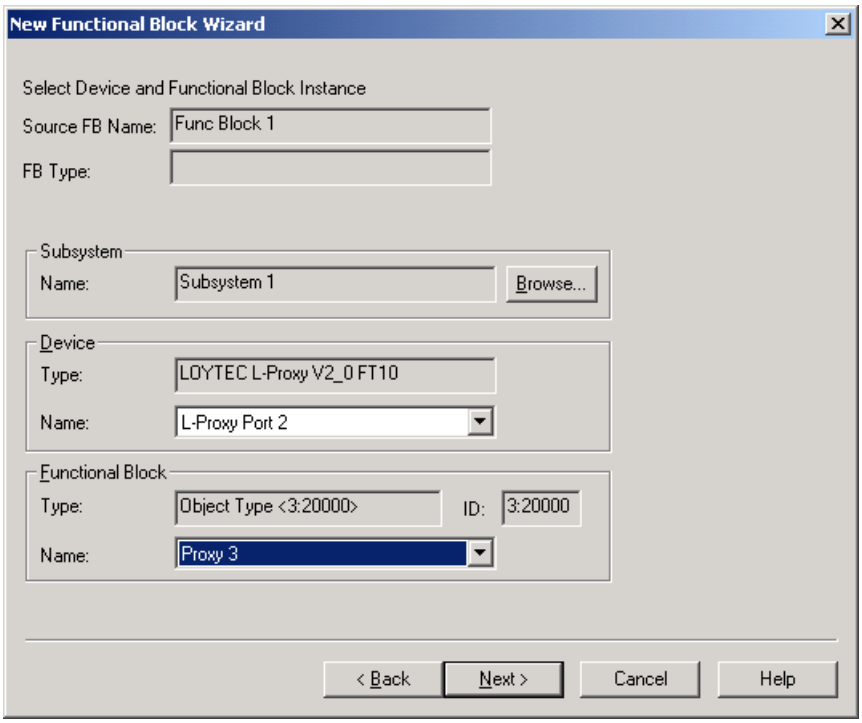


Figure 38 Select *L-Proxy Port 2* and *Proxy 1,2,3,4,5* in order to create a functional block for the Proxy object on Port 2.

Under FB Name select *Proxy*

Click *Finish*

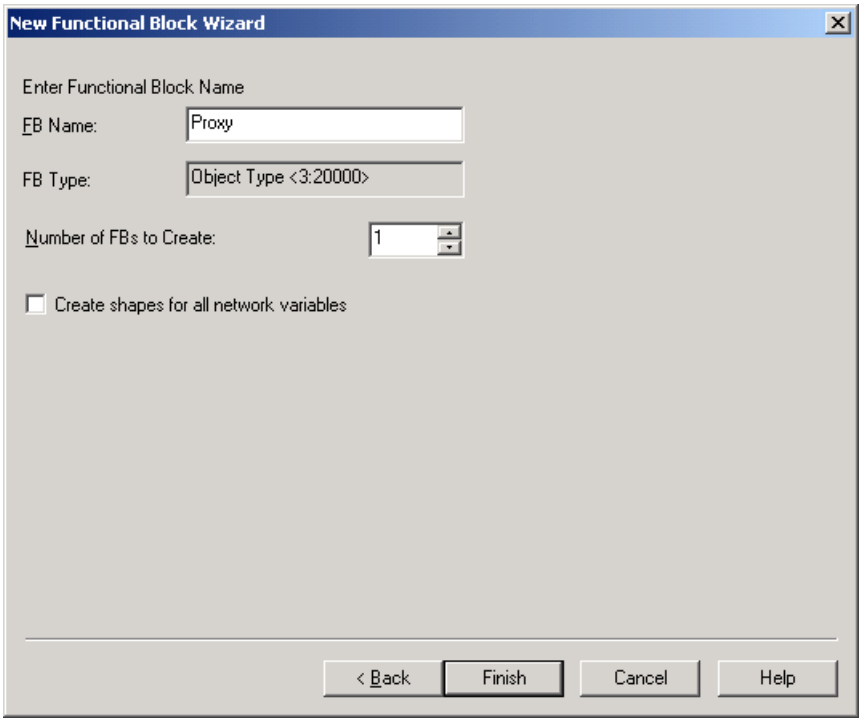


Figure 39 Enter the FB Name for L-Proxy Port 2.

An empty functional block as shown in Figure 40 appears in the drawing area.

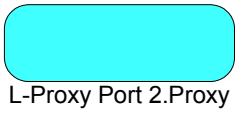


Figure 40 Empty functional block for L-Proxy Port 2.

You can repeat the above steps to create up to 5 Functional blocks on every L-Proxy port. The different functional blocks for one L-Proxy port are only for logical grouping in the LonMaker drawing.

You must repeat these steps for all L-Proxy ports that are connected to the network.

5.4.2 Creating Virtual Interfaces in NL-220

After adding the L-Proxy node a virtual interface must be created for the device. Right click on the device and select the option **New interface …** in the **Virtual interfaces** sub menu.

In the next window enter the name of the interface, for example **Interface 1**. You may create as many interfaces as needed (for example to sort network variables by functions).

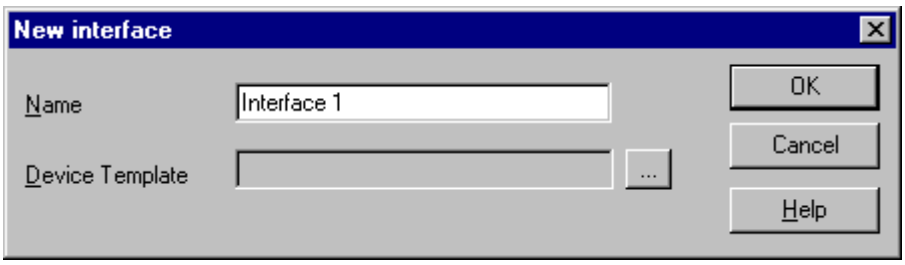


Figure 41 Creating a virtual interface in L-Proxy device

Click *OK*

The virtual interface must now appear in the L-Proxy device in the tree. Open the **L-Proxy1 Port2** device in the tree (click on the ⊞). You must be able to see a Static interface (containing non-dynamic network variables) and the virtual interface you created **Interface 1**.

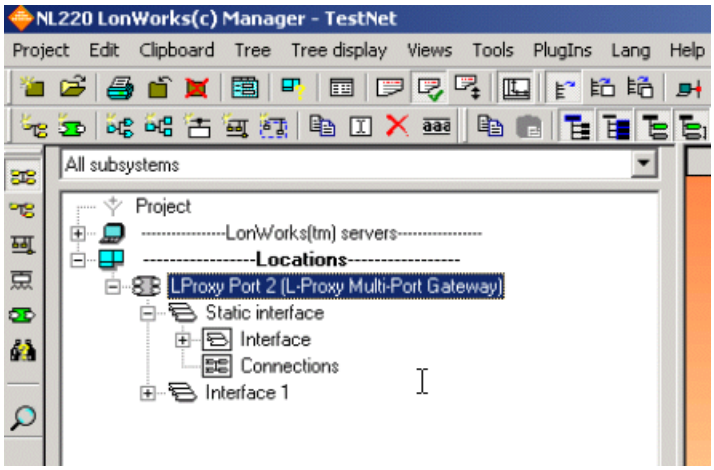


Figure 42 Interfaces of the L-Proxy device in tree.

You can repeat the above steps to create up to 5 virtual interfaces on every L-Proxy port. The different virtual interfaces for one L-Proxy port are only for logical grouping of the NVs.

You must repeat these steps for all L-Proxy ports that are connected to the network.

5.4.3 Creating Virtual Interfaces in Alex

After adding the L-Proxy node a virtual interface must be created for the device. In the tree view, select **Virtuelle Funktionseinheiten** of the newly created device. Click on *Neue virtuelle Funktionseinheit*

In the dialog, enter the name of the interface, for example **Proxy**. You may create as many interfaces as needed (for example to sort network variables by functions).

Figure 43 Creating a virtual interface in L-Proxy device

Click *OK*

5.5 Dynamic Network Variables

After creating the functional block/virtual interface for the Proxy object we need to create the input and output network variables for the Proxy object. L-Proxy uses dynamic network variables. Dynamic network variables are created during system configuration and can be added and deleted at any time whereas static network variables are created at compile time of the application program and cannot be changed afterwards.

Note: Always close the L-Proxy configuration utility when creating, adding, or deleting network variables in the Proxy functional block.

5.5.1 Creating a new dynamic network variable

Dynamic network variables are created by selecting the counterpart of the desired network variable binding. The dynamic network variable will inherit its properties from its counterpart e.g. the SNVType. Figure 44 shows a simple network comprising 3 nodes and the newly created Proxy functional block.

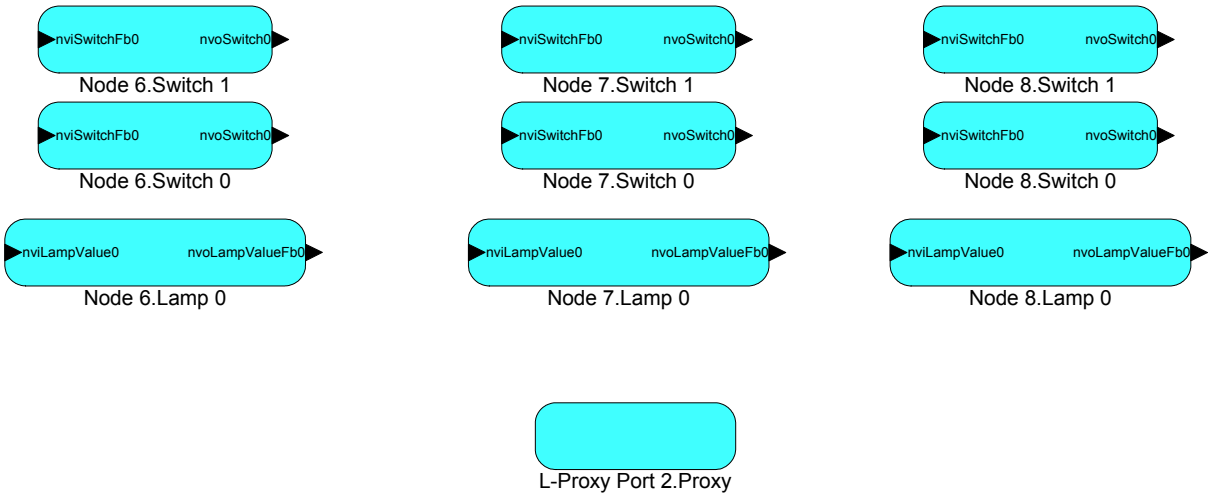


Figure 44 Simple network comprising 3 nodes and one L-Proxy Port.

Let's assume we want to connect network variable nvoSwitch0 on Node 6 to L-Proxy. This means we need to create an input network variable in the Proxy object with the same type as nvoSwitch0. Please follow the steps below.

5.5.1.1 Adding dynamic NVs in LonMaker

Drag the Input Network Variable shape onto the Proxy functional block. A new window appears (see Figure 45).

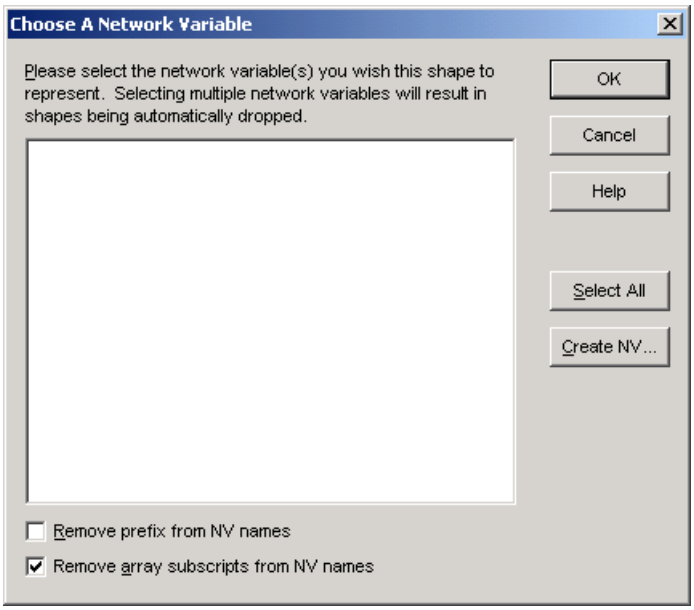


Figure 45 Add an input network variable to the Proxy functional block.

Click on *Create NV*

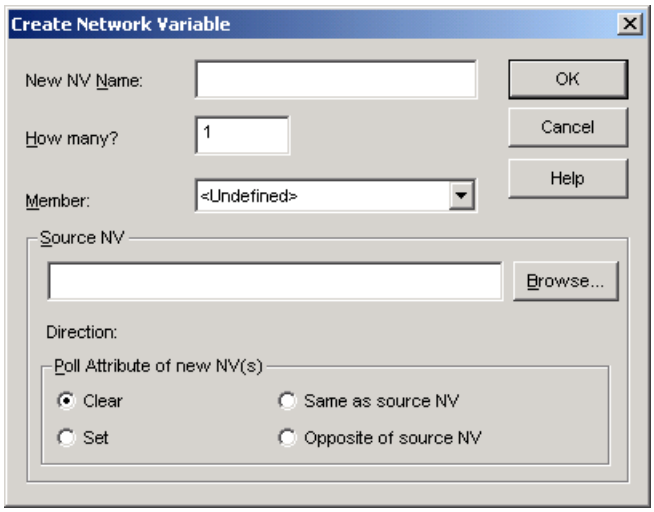


Figure 46 Dialog to specify the properties of the new network variable. If you want to create multiple copies of the same network variable you can specify the number in the *How many?* Field. This feature is especially useful to create multiple copies of the same output network variable.

Click on *Browse…*

and select the counterpart network variable nvoSwitch0 on Node 6 for this connection.

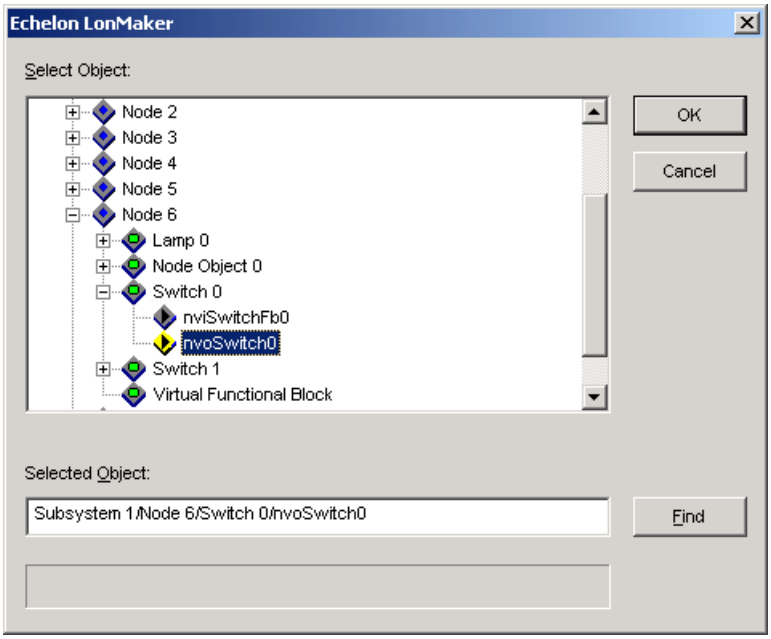


Figure 47 Select the counterpart NV on Node 6.

Click *OK*

A new name is already suggested for this newly created network variable. In our case we keep the *New NV Name: nviSwitch0* as shown Figure 48.

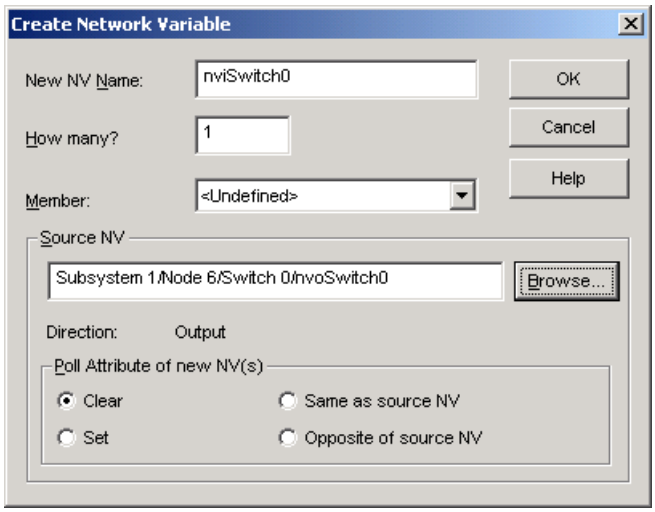


Figure 48 We keep the suggest name *nviSwitch0* for the new network variable.

Make sure that the Poll Attribute of the new NV is set properly. Set it to *Clear* if the NV is updated and set it to *Set* if the NV is polled. If the NV is polled both the source and the destination NV must be declared polled. Also if the NV is updated both the source and the destination NV must have the Poll Attribute cleared.

Click *OK*

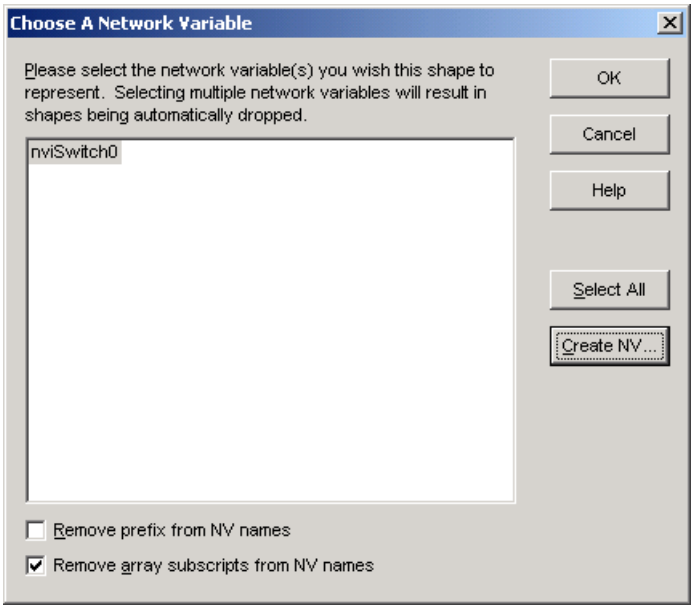


Figure 49 We have created a new input network variable in the Proxy functional block.

Click *OK*

After the network drawing updates, it shows the newly created input network variable in the Proxy functional block.

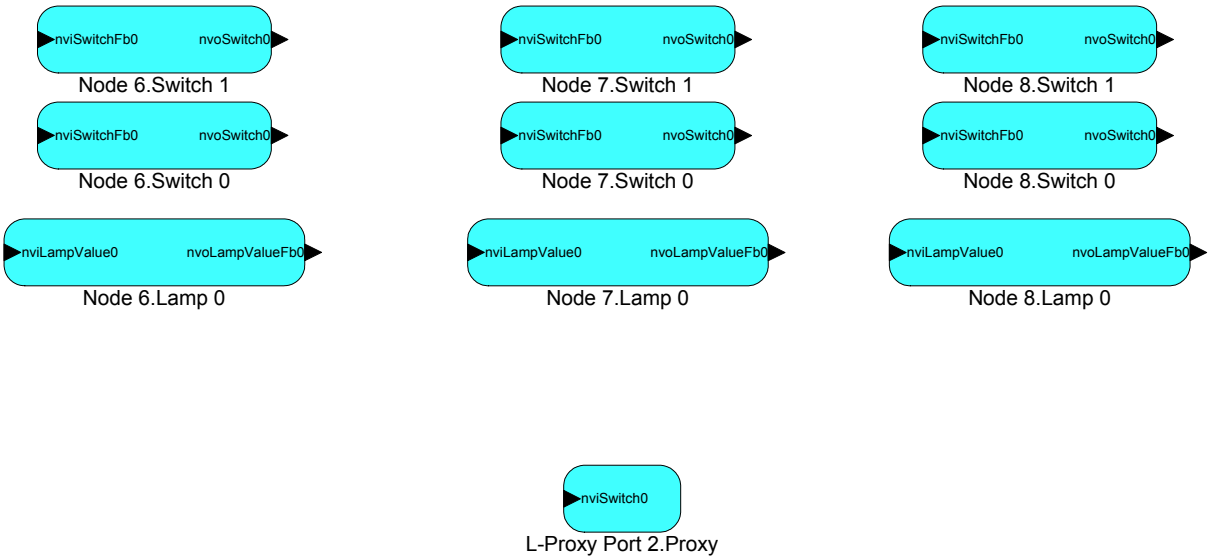


Figure 50 Updated network diagram that shows the new input network variable in the Proxy functional block.

5.5.1.2 Adding dynamic NVs in NL-220

Drag the counterpart (Node 6.nvoSwitch0) Network Variable in the tree onto the virtual interface **Interface 1** of the L-Proxy device.

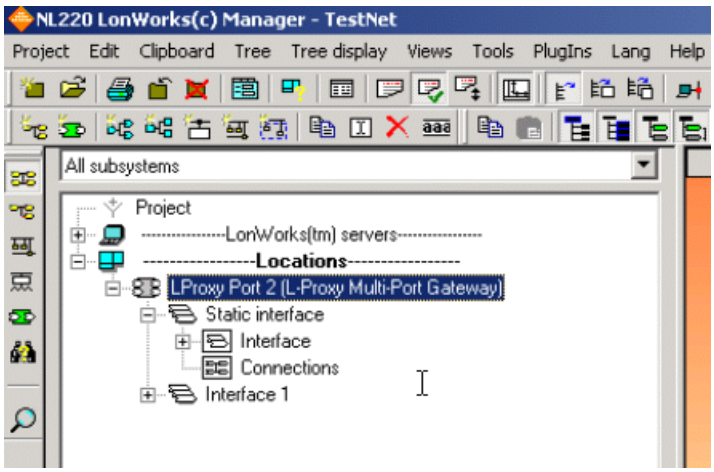


Figure 51 Dragging the counterpart network variable onto Virtual Interface 1.

This operation opens the connection window in working view (right view) and a new window pops up (see Figure 52).

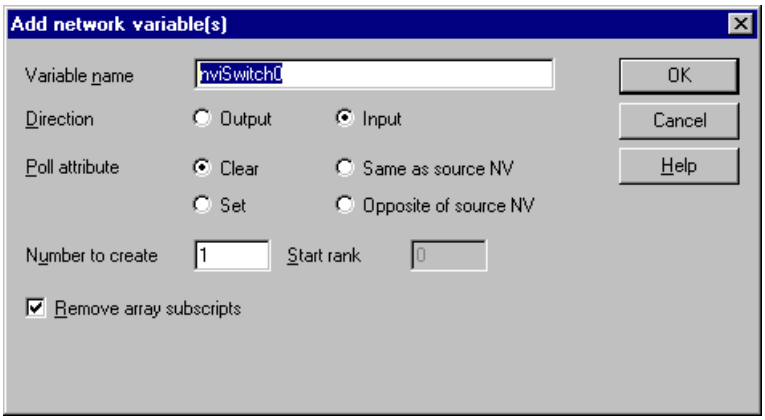


Figure 52 Add an input network variable to the L-Proxy virtual interface Interface 1.

This dialog is used to specify the properties of the new network variable. If you want to create multiple copies of the same network variable you can specify the number in the *Number to create* Field. This feature is especially useful to create multiple copies of the same output network variable. When creating multiple identical NVs please make sure that the "Remove array subscripts" is checked. Otherwise these NVs are not properly displayed in the L-Proxy configuration plug-in utility.

A new name is already suggested for this newly created network variable. In our case we keep the *Variable Name: nviSwitch0*.

Make sure that the Poll Attribute of the new NV is set properly. Set it to *Clear* if the NV is updated and set it to *Set* if the NV is polled. If the NV is polled both the source and the destination NV must be declared polled. Also if the NV is updated both the source and the destination NV must have the Poll Attribute cleared.

Click on *OK*

Now the tree shows the newly created input network variable in the virtual interface **Interface 1** (you must open the virtual interface in the tree to display the network variable).

As shown in Figure 53 the connection window in the work view (right view) includes now the nvoSwitch0 of Node 6 and the nviSwitch0 of the L-Proxy device. You can create or cancel the connection.

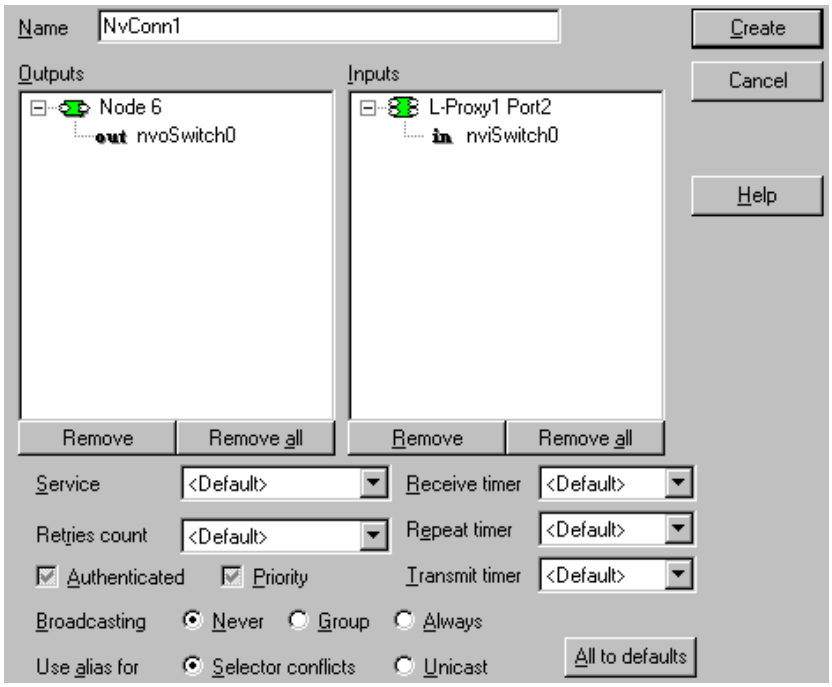


Figure 53 Connection window with the newly created L-Proxy variable and the Node 6 output variable.

5.5.1.3 Adding dynamic NVs in Alex

Select **Netzwerkvariablen** in the tree view of one of the virtual interfaces of the L-Proxy device. Click on *Neue Netzwerkvariable*.

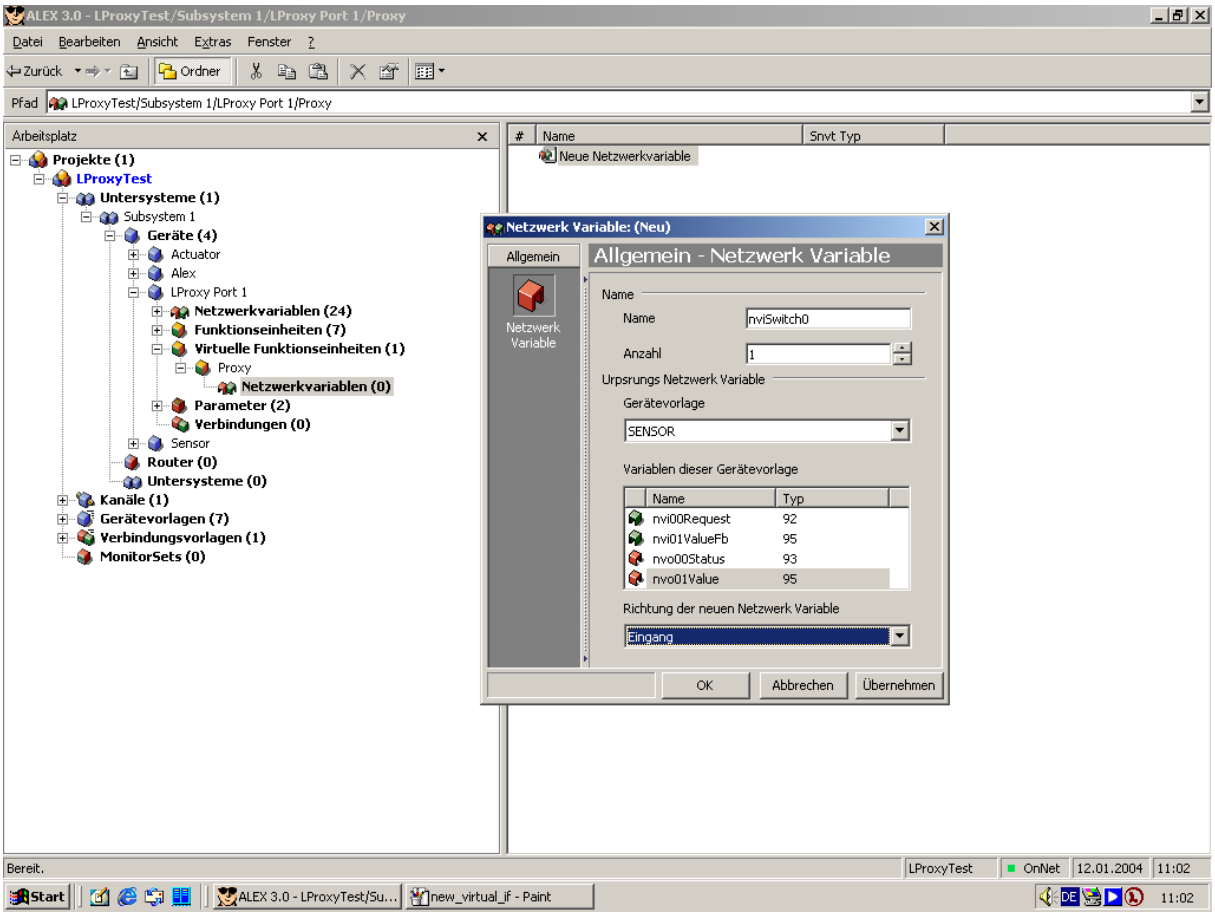


Figure 54 Add an input network variable to the L-Proxy virtual interface Proxy.

Enter the name for the network variable and the count how many network variables of this type should be created. In Gerätevorlage, slect the device template of the node which hosts the network variable to which the L-Proxy network variable will be bound later. Select the counterpart network variable in the list and specify the correct direction (Eingang for an input network variable). Click *OK*.

Repeat this for all network variables which will be used on the L-Proxy.

5.5.2 Modifying network variables

Dynamic network variables in the Proxy functional block or the virtual interface can be added and deleted without loosing an existing binding in LonMaker, NL-220 or Alex. The L-Proxy configuration utility will perfectly deal with the added or deleted network variables as explained in Section 5.6. When deleting or adding NVs the L-Proxy configuration utility must be closed. The design flow is as shown in Figure 55.

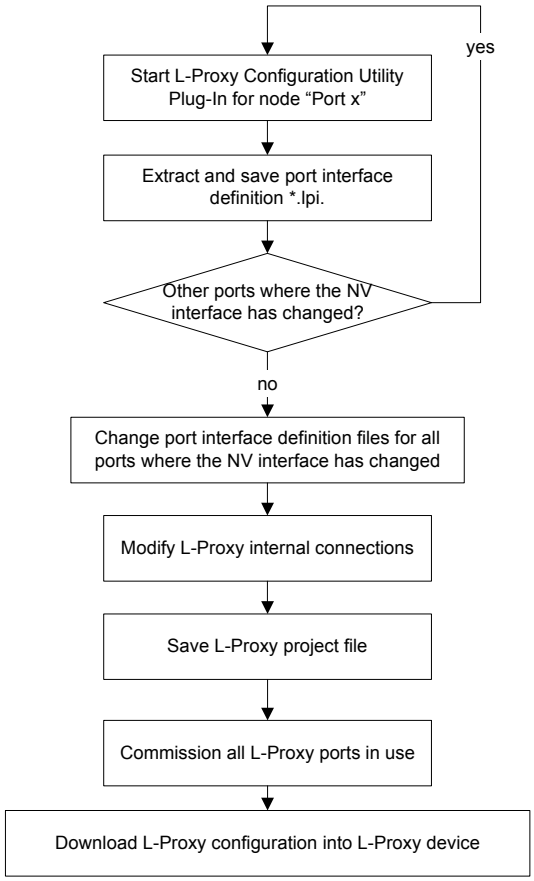


Figure 55 Design flow when network variables have been added, deleted, or modified.

5.6 Using the L-Proxy configuration utility

L-Proxy comes with a configuration utility to create the L-Proxy internal connections (bindings). This user friendly intuitive LNS Plug-In is a powerful tool to create the L-Proxy internal connections. If you follow the few steps explained below you will have L-Proxy configured in no time.

In LonMaker the Plug-in is started by right clicking on the L-Proxy device shape or the L-Proxy functional block and selecting *Configure*... from the pop-up window.

In NL-220 the Plug-in is started by right clicking on the L-Proxy node, then selecting the Option **L-Proxy Configuration Plugin** in the **PlugIns** sub menu.

In Alex the Plug-in is started by right clocking on the L-Proxsy device and selecting the **L-Proxy Configuration Plugin** in the **Starte PlugIn** sub menu.

A window similar to what is shown in Figure 56 should appear.

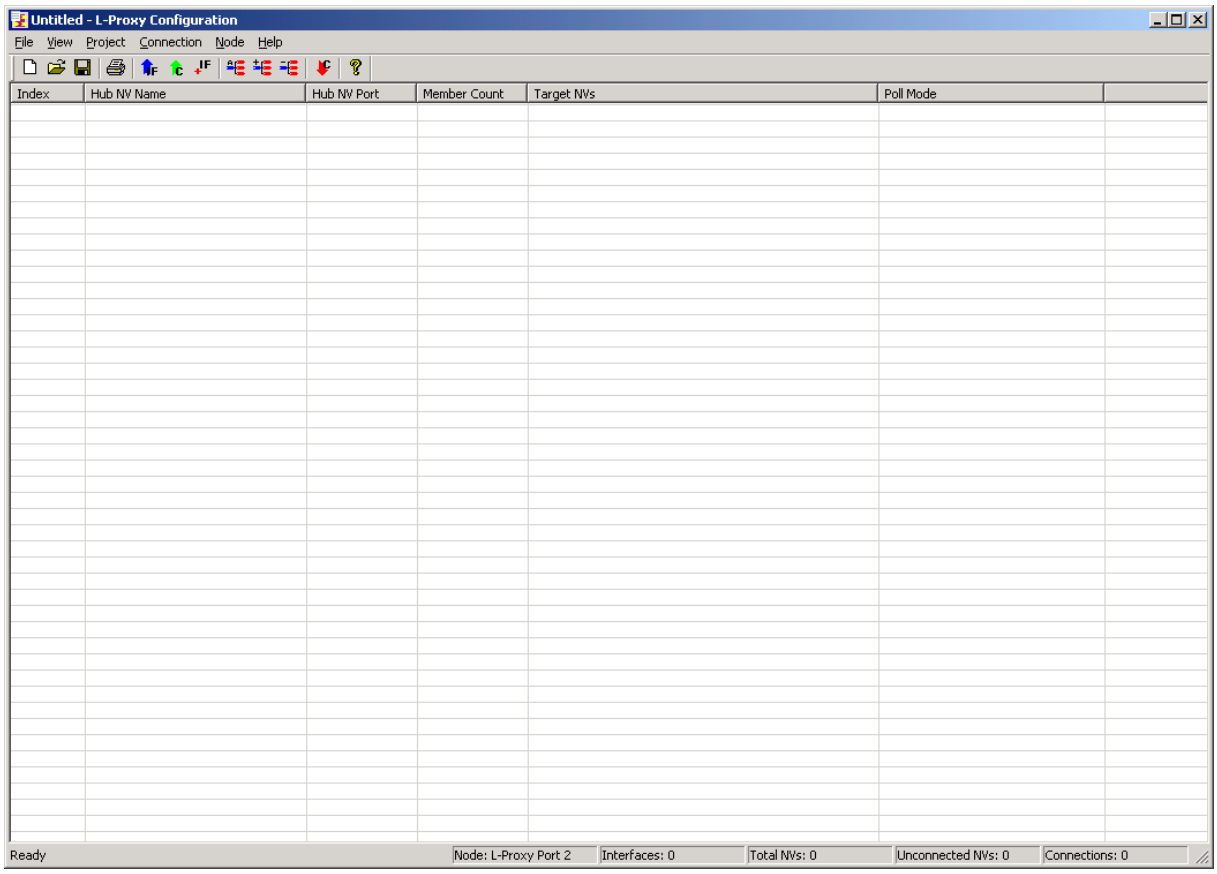
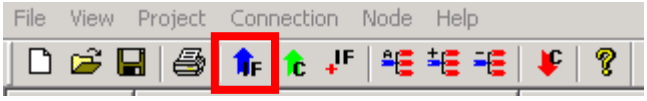


Figure 56 L-Proxy configuration utility main window.

5.6.1 Extracting the Network Variable Interface when L-Proxy is not online

As a first step the network variable interface must be uploaded from every L-Proxy node (Port). If the network management tool is not online with the L-Proxy device one should

Select Project => Load Port Interface from Node or

click on  in the tool bar.

This will open a file requestor that asks for the file name for this port interface definition.

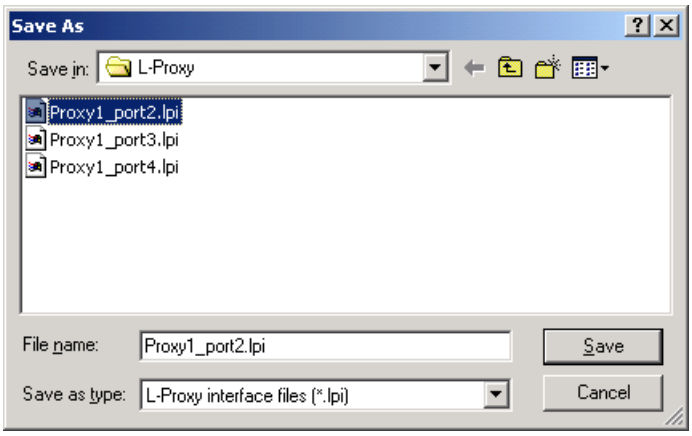


Figure 57 Specify a file name for the port interface definition file.

Give the file a meaningful name like in the example in Figure 57. Close the L-Proxy Plug-In and repeat this step for all other L-Proxy ports. If the ports are used in different domains (different LNS databases) you need to open the different LNS databases and extract the port interface definitions for the L-Proxy port in this LNS database (domain). If multiple engineers are working on a project in different LNS databases they need to extract the port interface definition after they have created all network variables on L-Proxy and send the port interface file to the engineer who makes the connections between the different ports.

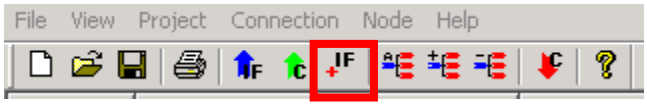
If the network variable interface changes at a later date please extract the interface definition file again and continue with the next step.

Note! *The L-Proxy configuration utility plug-in must be started for each L-Proxy Port for which the interface is being extracted. This is the case if multiple L-Proxy ports are used in the same LNS database.*

5.6.2 Configure Port Interface

After all port interface definition files have been created, open the L-Proxy configuration utility plug-in for the port, which you have chosen to be the "maintenance port" for this L-Proxy device. Note, that "maintenance port" doesn't mean this is a special port, we only want to use this nomenclature to refer to the port on which the L-Proxy configuration is downloaded into the device. It is also the Port to which we are currently physically connected.

Select Project => Configure Port Interface or

click on  in the tool bar.

This will open a dialog that asks for the file names of the previously extracted port interface definitions.

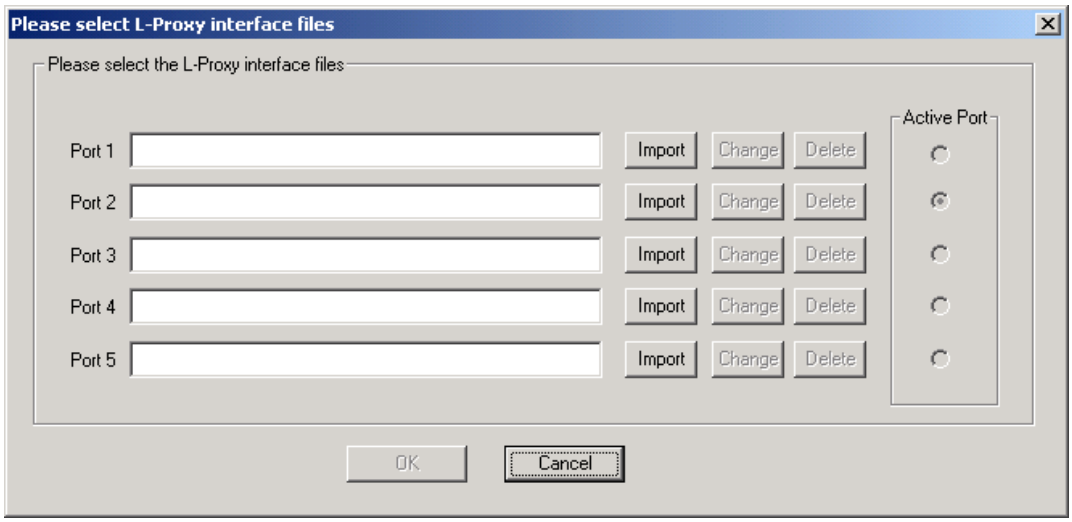


Figure 58 Specify the port interface definition files for up to 5 L-Proxy ports. The right most column indicates the L-Proxy port to which the network management tool is connected if LonMaker is On-Net.

Click *Import* and select the interface definition files

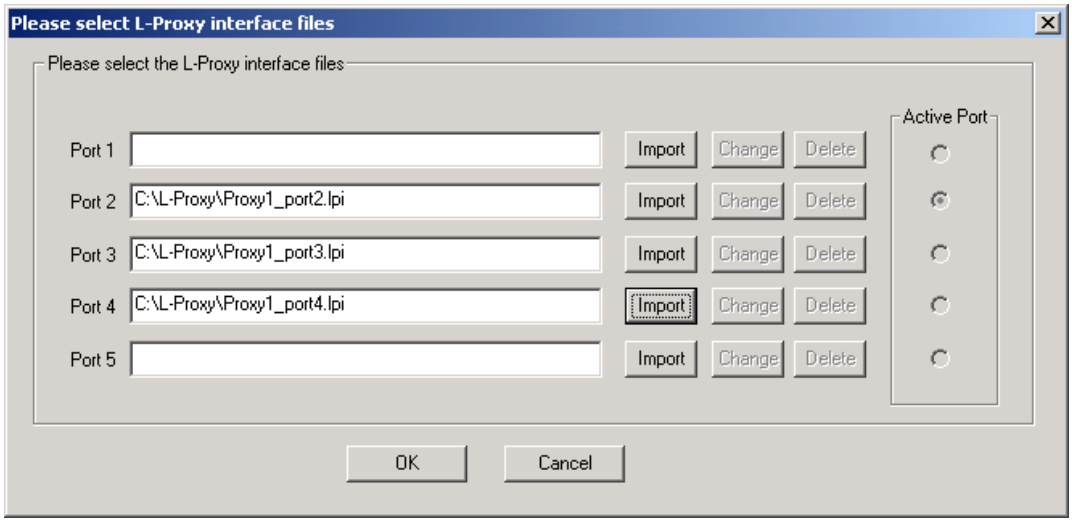


Figure 59 Specify the port interface definition files for up to 5 L-Proxy ports.

Click *OK*

If you need to change the interface definition file at a later date or want to remove one port click on Change or Delete to carry out the desired operation.

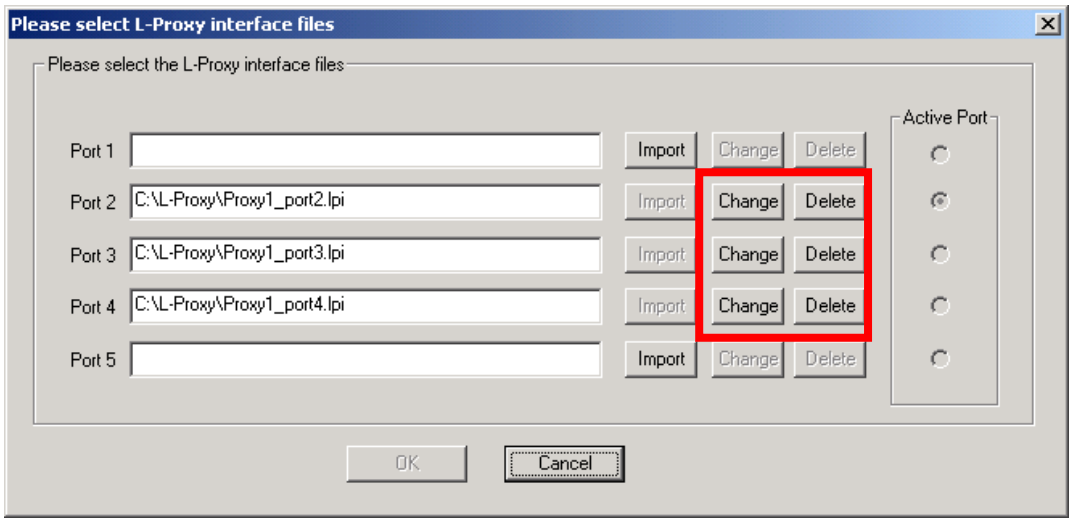


Figure 60 Changing or deleting an interface definition file in the port list.

We have now extracted the interface definitions for the following network. Note, that Port 2 on L-Proxy has 4 output network variables of the same type nvoLampValue0, nvoLampValue1, nvoLampValue2, and nvoLampValue4.

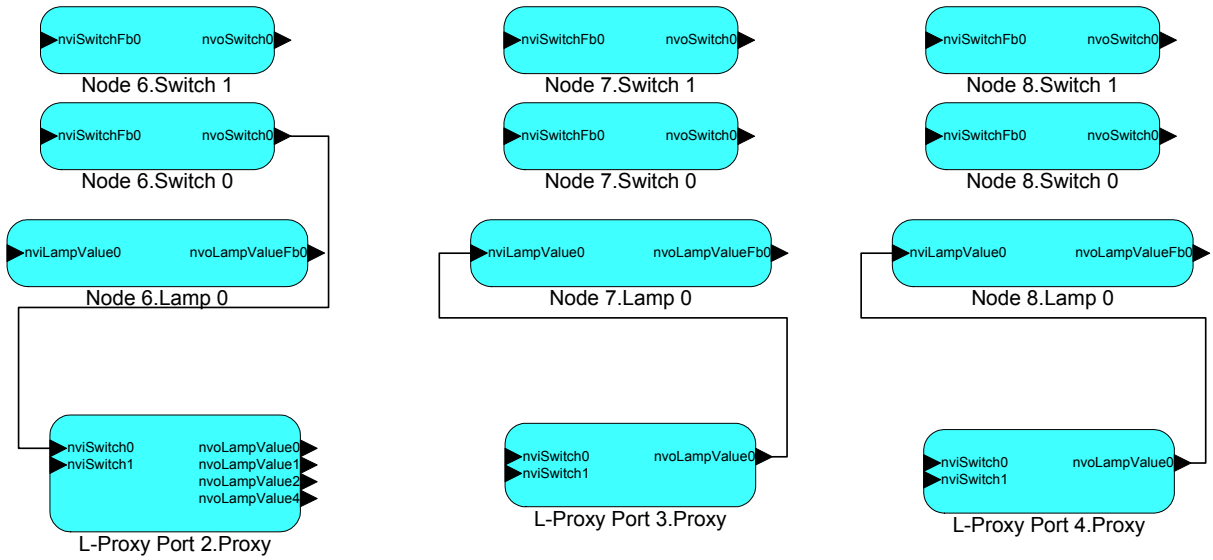
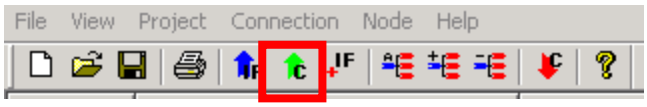


Figure 61 Network variable interface definitions for 3 L-Proxy ports (Port 2 – Port 4).

5.6.3 Extracting the Network Variable Interface when L-Proxy is online

If the L-Proxy device is online with the network management tool one can simply select

Node => Upload Configuration or

click on  in the tool bar

in order to upload the network variables from **all** L-Proxy ports together with the internal connections. In this situation the steps in Section 5.6.1 and 5.6.2 can be skipped.

After a successful upload the following window appears.

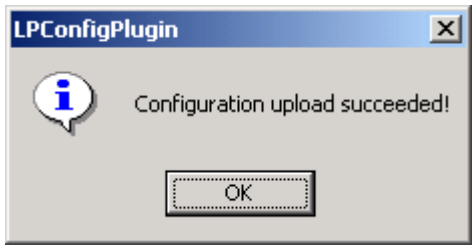


Figure 62 Successful configuration-upload message.

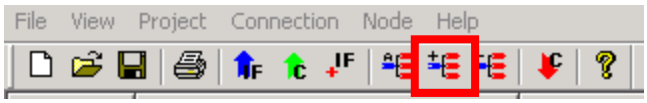
5.6.4 Add new connection

After extracting the network variable interface definitions and assigning the interface definitions files with the L-Proxy ports in the previous section or simply uploading the configuration from an L-Proxy that is online we are now ready to create the L-Proxy internal connections between the different input and output network variables. Note, that it doesn't matter if the network variables that must be connected reside on the same port, on different ports, in the same domain, or in different domains.

A connection is a sort of L-Proxy internal binding between input and output network variables on L-Proxy. A connection always consists of *one* source (HUB) network variable and *one or multiple* target network variables. HUB NVs can be input or output network variables.

To create a new connection select

Connection => New or

click on 

in the tool bar to create a new connection.

This will bring up the connection window as shown in Figure 63. The top left window shows the available Hub Network Variables. The two bottom windows show the available Target Network Variables and the Selected Target Network Variables.

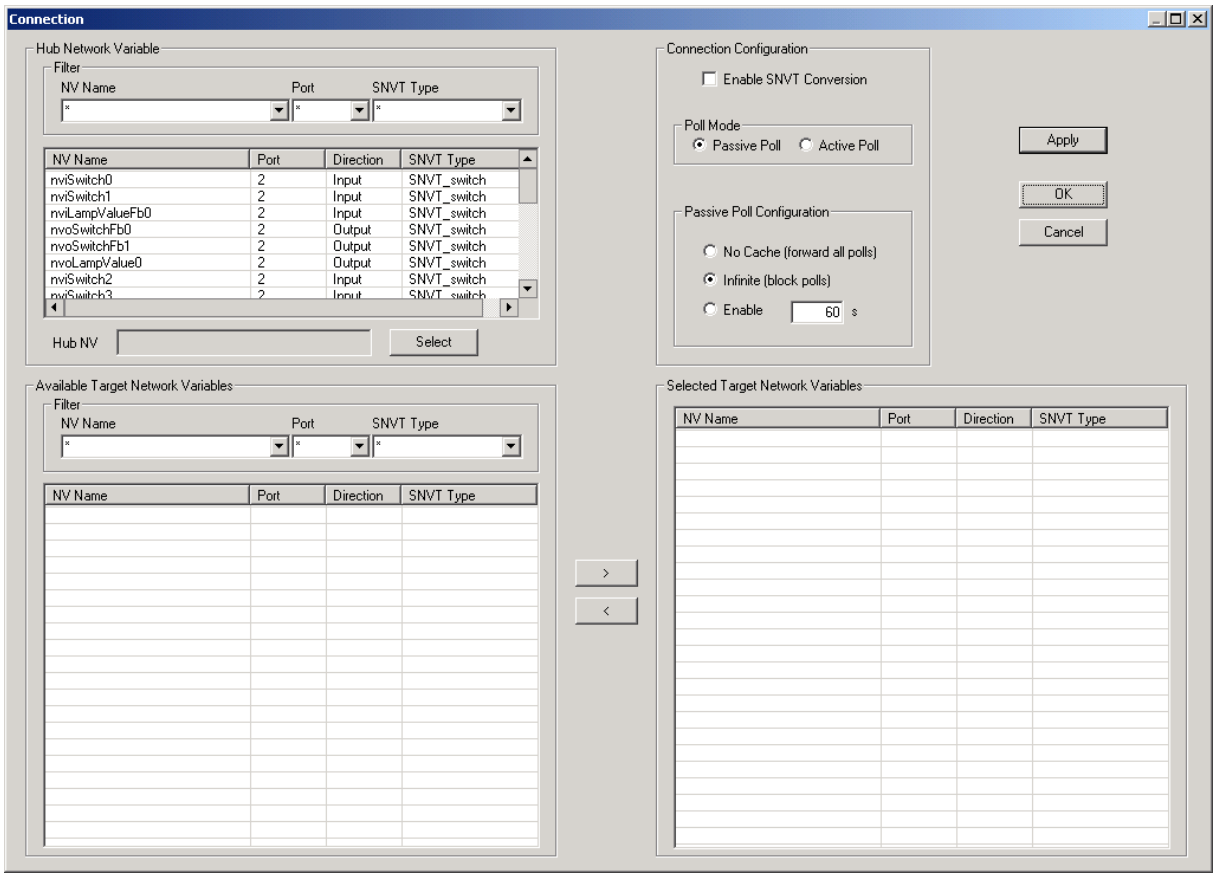


Figure 63 Connection dialog window.

Double-click on nviSwitch0 on Port 2 in the Hub Network Variable window. The *Available Target Network Variable* window now lists all possible target network variables. If there are no valid target network variables available the windows remains empty.

Note! *Valid target NVs are network variables that have the same type (SNVT) as the Hub NV, or if SNVT conversion is checked, a compatible type!*

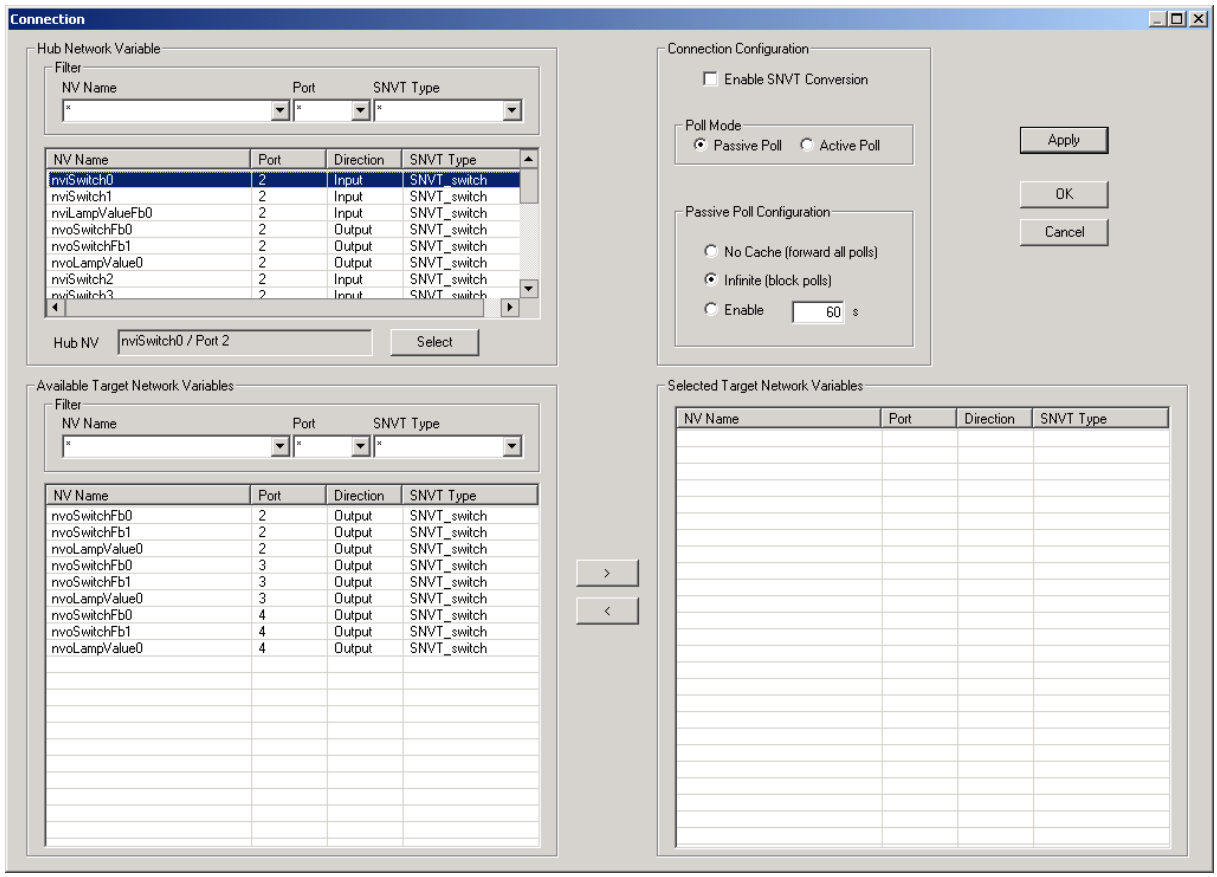


Figure 64 After selecting a hub network variable the available target network variables are listed in the “Available Target Network Variables” window.

Select from the list of available target NVs the ones that should be connected with the hub NV nviSwitch0 / Port 2. Press the “>” button to add the selection to the target NV list or the “<” button to remove an NV from the target NV list.

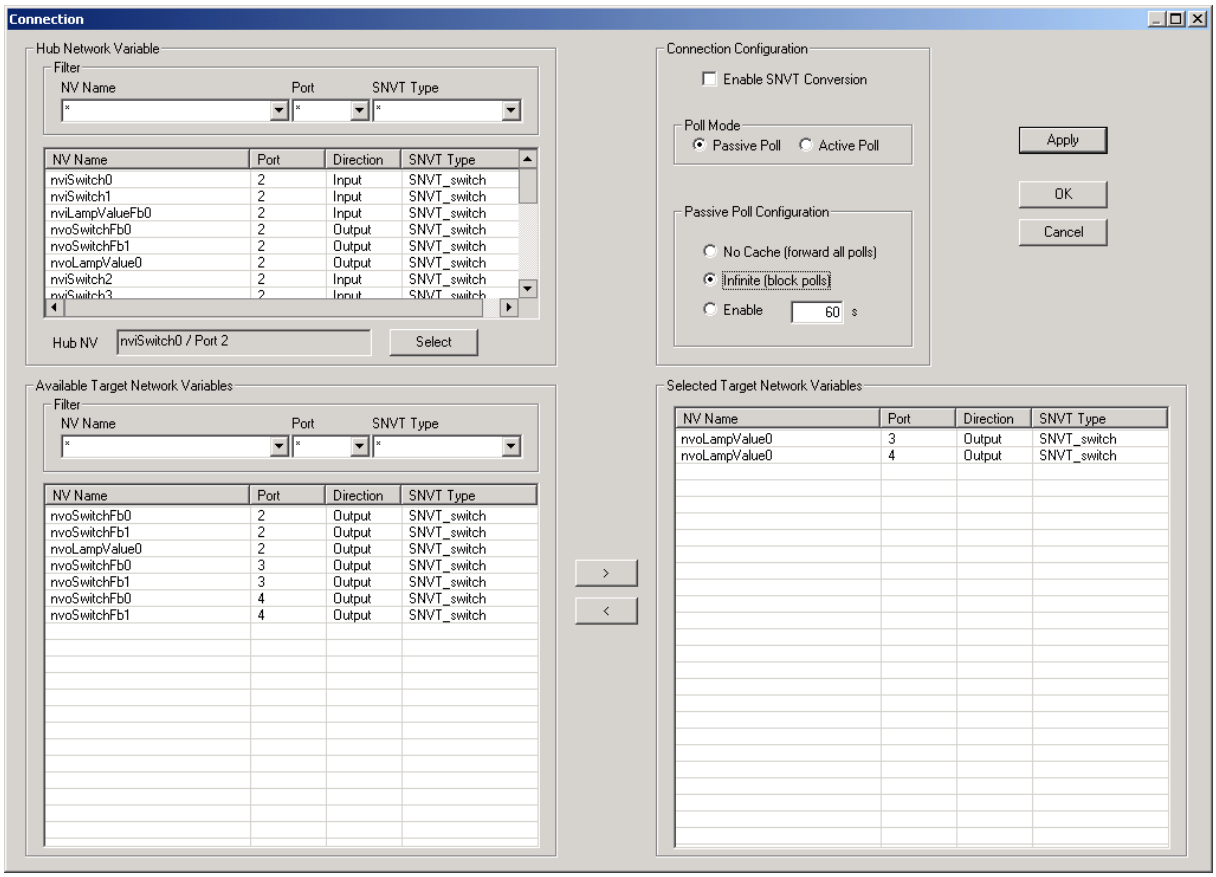


Figure 65 The target NVs for this connection have been selected.

Click *OK*

to save the new connection. The new connection connects the input NV nviSwitch0 on Port 2 with the output NV nvoLampValue0 on Port 3 and nvoLampValue0 on Port 4.

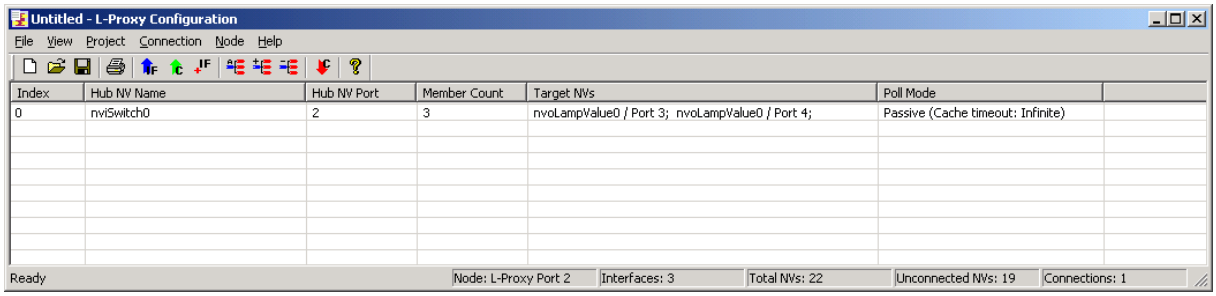
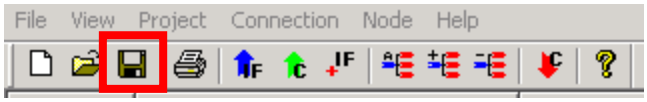


Figure 66 We have created a new connection inside L-Proxy.

Before downloading the new configuration into L-Proxy we should save the project

File => Save or

click on  in the tool bar.

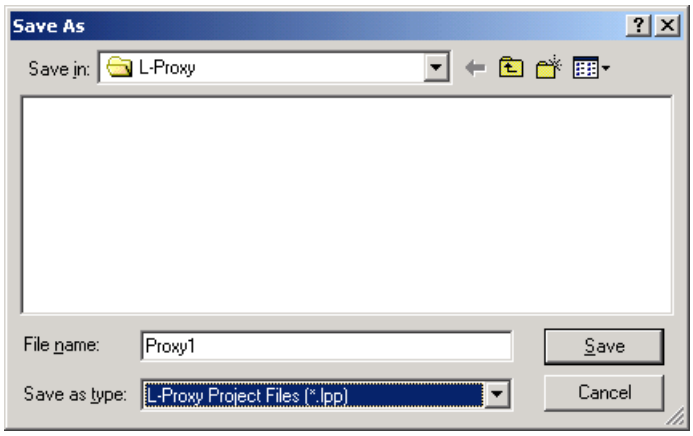
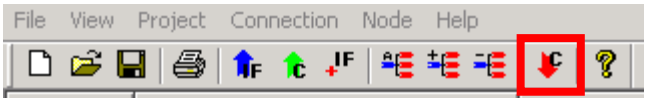


Figure 67 Saving the L-Proxy configuration in a project file.

Note! Before downloading the L-Proxy configuration all ports in use on L-Proxy must be commissioned.

We can now select

Node => Download Configuration or

click on  in the tool bar

to download the new configuration into L-Proxy. If the download succeeds the following message appears.

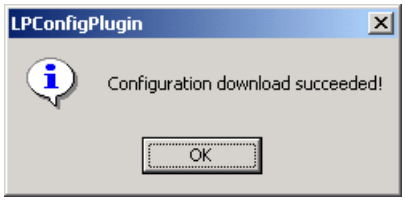


Figure 68 Successful configuration download into L-Proxy.

If you get a Configuration Download failed message the port interface definition files don't match with the port interface on the L-Proxy device. Please extract the port interface definition files for used ports again, import the port interface definitions files, and try the configuration download again.

Note: After commissioning one or multiple L-Proxy ports the L-Proxy configuration must be downloaded with the L-Proxy Plugin.

If we now flip the switch on Node 6 in Figure 69 the lamp on Node 7 and Node 8 will light up.

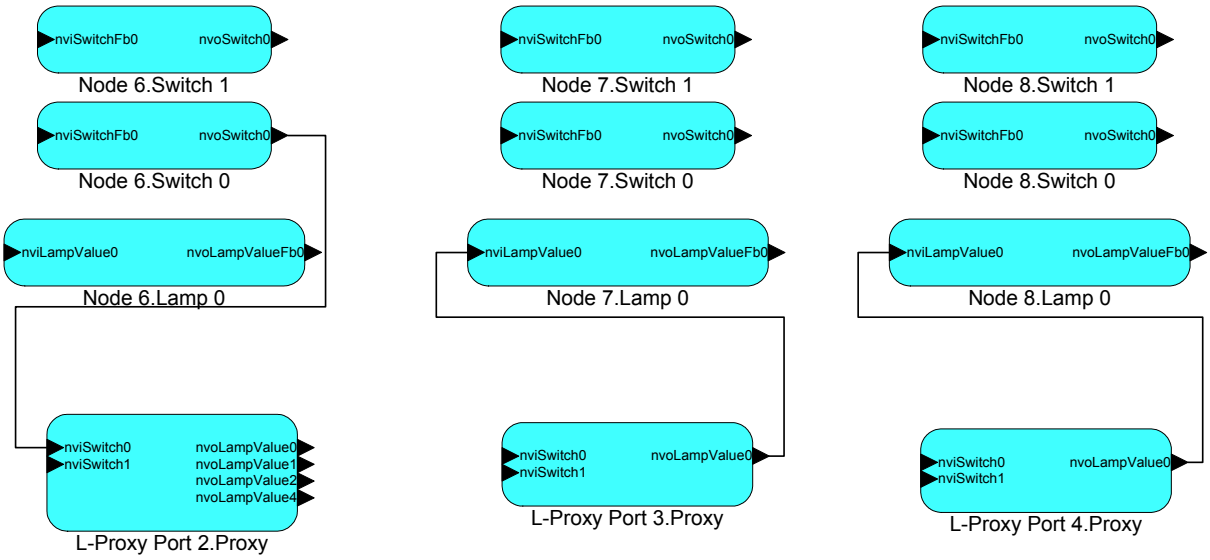


Figure 69 L-Proxy connects the switch on Node 6 (Port 2) with the lamp on Node 7 (Port 3) and Node 8 (Port 4).

5.6.5 Delete a connection

To delete a connection select the connection in the main window and select

Connection => Delete or right mouse-click on the connection and select Delete.

You can select multiple connections and then delete them together.

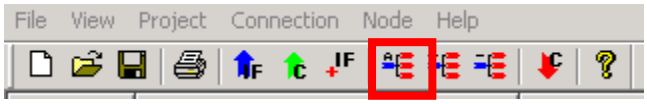
5.6.6 Edit a connection

To edit a connection in order to add or remove network variables or to change the poll configuration one can double click the connection in the main window.

5.6.7 Auto-connection

The auto-connection feature connects input and output network variables that have the same name after the prefixes nvi or nvo have been removed from their names.

Select Connection => Autoconnect or

click on 

in the tool bar

to auto connect NVs. Note, that the auto connect feature does not work if multiple input NVs **and** multiple output NVs with the same name after the prefixes exist. In this case you will see the following message displayed: No network variables for auto-connection available.

5.7 L-Proxy LonMark Objects

The L-Proxy has the LonMark objects as outlined in the following sections.

5.7.1 Node Object

The L-Proxy provides a node object as specified in [1]. Its object number is 0 (as for every node object). The object implements the following optional features as described in [1]:

- Time stamp input .
- LonMark File Transfer including random access
- Max send time configuration property (heart beat)
- Object versioning
- Node versioning

Further it has the following manufacturer specific features:

- The L-Proxy provides the current value of the on-board temperature sensor.
- The L-Proxy provides the current value of the input voltage on the power supply.
- The L-Proxy provides the port number for each port.

Thus, the object has the following network variables (values in brackets are the corresponding network variable indices):

- nviRequest (0)
- nvoStatus (1)
- nviTimeSet (2)
- nviFileReq (3)
- nvoFileStat (4)
- nviFilePos (5)
- nciMaxStsSendT (6)
- nciNodeMajVer (SCPTobjMajVer) (7)
- nviNodeMinVer (SCPTobjMinVer) (8)
- nciDevMajVer (SCPTdevMajVer) (9)
- nciDevMinVer (SCPTdevMinVer) (10)
- nvoSystemTemp (SNVT_temp, #39) (11) (maximum update rate is every 2,000 ms)
- nvoSupplyVolt(SNVT_volt, #44) (12) (maximum update rate is every 2,000 ms)

- nroPort (no SNVT, #0, 1 byte) (13) (constant)

The node object does not support requests. See the description of the other objects for a description of the supported requests and status. Note, that even though each port has its own node object, these node objects are not completely independent. If the time is updated on one port this will update the L-Proxy's on-board RTC and thus concerns the time on all ports. The heart-beat is set independent for each port. The status and request NVs on each port are independent, too. However, the status of the other objects might not be.

The L-Proxy Configuration Utility Plug-in can be used to control the real-time clock. From the plug-in main menu select

Node => Configure Node Object …,

which displays the dialog from Figure 70. This dialog shows the current system temperature, the supply voltage, the port number for which the plug-in was opened and it allows setting the update rate for the status information.

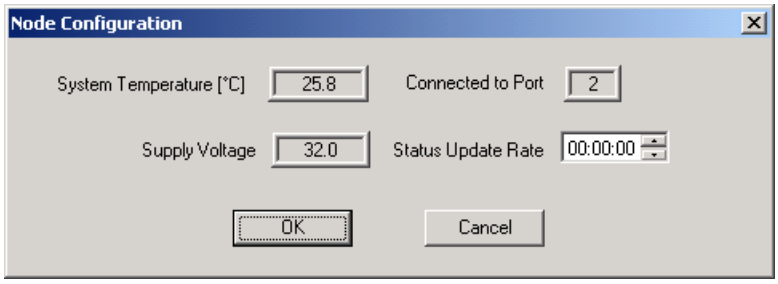


Figure 70 Node configuration window.

5.7.2 Real Time Keeper Object

The real time keeper object is based on the real time keeper functional profile as specified in [2]. Its object number is 1. The object implements the following optional features as described in [2]:

- Master operation configuration property
- Time broadcast configuration property
- Summer date and time configuration property
- Winter date and time configuration property
- Manual allowed configuration property (manual setting of time via console)
- Object versioning

Thus, the object has the following network variables (values in brackets are network variable index):

- nvoTimeDate (14)

- nciMasterSlave (15)
- nciUpdateRate (16)
- nciSummerTime (17)
- nciWinterTime (18)
- nciManAllowed (19)
- nciTimerMajVer (SCPTobjMajVer) (20)
- nciTimerMinVer (SCPTobjMinVer) (21)

The real time keeper object does support the following requests:

- RQ_NORMAL: No change of object status. Status network variable is updated with status of real time keeper object as with RQ_UPDATE_STATUS.
- RQ_UPDATE_STATUS: Updates node status network variable with current status of real time keeper object.
- RQ_REPORT_MASK: Reports status bits supported by real time keeper object (see below). Set report_mask to 1.

The real time keeper object can have the following status:

- out_of_service: Set to 1 if the RTC is out of operation.

The Real Time Keeper uses the on-board RTC to keep the time. Note, that even though each port has its own time-keeper object, these objects are not completely independent. Since only one RTC exists they will all report the same time. As a consequence, the values of nciManualAllowed, nciSummerTime and nciWinterTime will always be set to the same value. That is, if they are changed on one port, the other ports will also have their corresponding NVs set to the new value.

The L-Proxy Configuration Utility Plug-in can be used to control the real-time clock. From the plug-in main menu select

Node => Configure Clock …,

which displays the dialog from Figure 71.

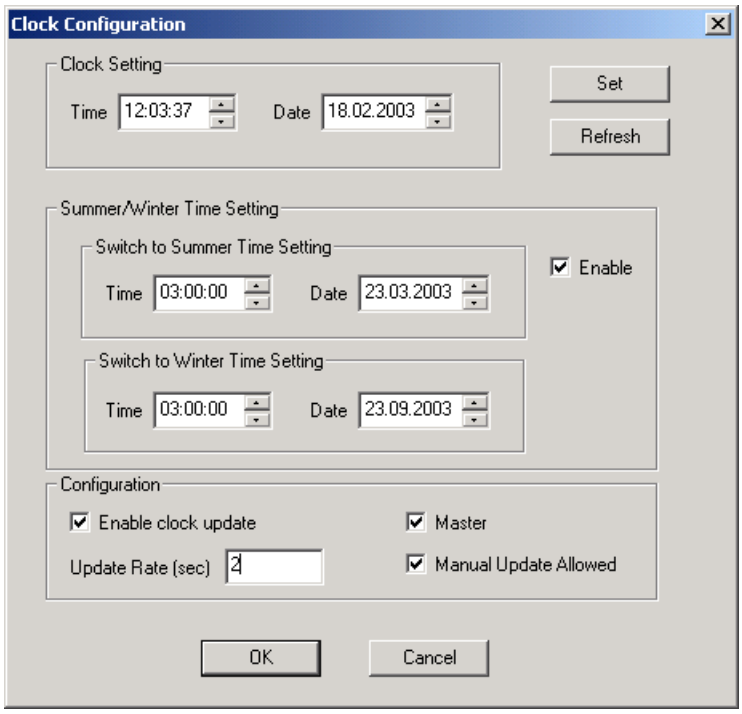


Figure 71 Control panel for real-time clock settings.

Under clock settings one can adjust the current date and time information. The Summer/Winter Time Setting section allows setting dates when the real-time clock should switch between summertime and wintertime and between wintertime and summertime. This setting must be manually adjusted every calendar year. The third section defines if the nvoTimeDate network variable should be periodically updated and what the update rate should be. The manual update allowed allows or disallows setting the clock via the console interface. The master flag allows to set the clock as a master clock or a slave clock as defined in [2].

5.7.3 Proxy Object

The proxy object is based on the generic controller object as specified in [1]. Its object number is 2. This object has only the following network variables for object versioning (values in brackets are network variable index):

- nciProxyMajVer (SCPTobjMajVer) (22)
- nciProxyMinVer (SCPTobjMinVer) (23)

The proxy object does support the following requests:

- RQ_NORMAL: Enables the proxy object. If it was disabled a consistency check of the network variable relaying configuration is performed.
- RQ_UPDATE_STATUS: Updates node status network variable with current status of proxy object.
- RQ_REPORT_MASK: Reports status bits supported by proxy object (see below). Set report_mask to 1.

- RQ_DISABLED: Disables the proxy object. No network variable relaying is performed anymore. That is, incoming updates are not forwarded and polls are not answered. All outstanding polls are terminated.
- RQ_ENABLE: Same as RQ_NORMAL.
- RQ_SELF_TEST: Perform consistency check of the network variable relaying configuration. During consistency check the object status is set to self_test_in_progress. If the consistency check fails programming_failed and fail_self_test is set.

The proxy object can have the following status:

- disabled: The object is disabled (see above).
- fail_self_test: The self test failed (see above).
- self_test_in_progress: The self test is in progress (see above).
- programming_fail: The consistency check of the network variable relaying configuration failed.

The purpose of the object is to do network variable relaying and SNVT translation. Thus, the behavior of the object is defined during the configuration phase by

- Defining dynamic network variables.
- Creating the connections between the dynamically created network variables.

6 Application Scenarios

This chapter shows some networking scenarios that are typical applications for L-Proxy. Please note that only few scenarios are shown right now and this chapter will be continuously updated with new application areas.

6.1 Address Table Extension

L-Proxy can be used to expand the number of address table entries available in a standard node that uses a Neuron Chip. If more than 15 destination addresses are required by the application one can use L-Proxy to overcome this limitation. Also if the node is a member of more than 15 groups L-Proxy can help to remove this restriction.

6.1.1 Saving entries in the address table

Figure 72 shows a simple network diagram comprising three Neuron Chip powered nodes. Each node contains four LonMark objects, a node object, an actuator object and two sensor objects. The actuator object controls a lamp; the two sensor objects are connected to switches.

The output network variable nvoSwitch0 on node 6 is bound to nviLampValue0 on node 7 and on node 8. The 2 bindings use up two entries in the address table.

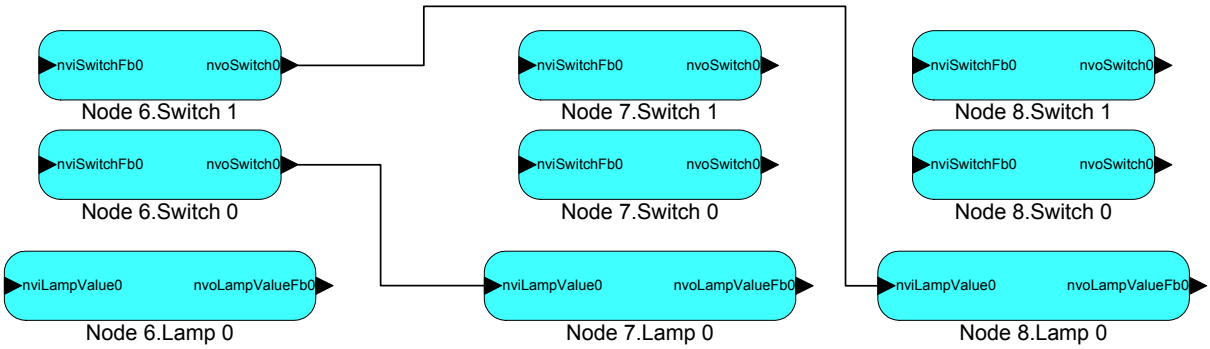


Figure 72 Two individual subnet/node addressed bindings use up 2 entries in the address table.

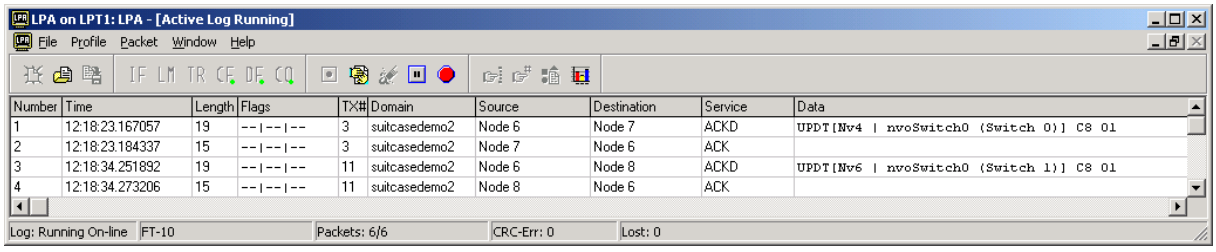


Figure 73 Protocol analyzer log file for the 2 subnet/node bindings shown in Figure 72. Node 6 first sends a packet to node 7 and then a packet to node 8.

L-Proxy can be used to make a single subnet/node addressed binding from the source node to L-Proxy. L-Proxy then completes the binding to the final destination nodes 7 and 8. In this case one entry in the address table was saved in the source node (node 6). Figure 74 shows the same network from above but now we added one port of L-Proxy (Port 2) to have a single destination address for node 6 thus saving an entry in the address table of node 6.

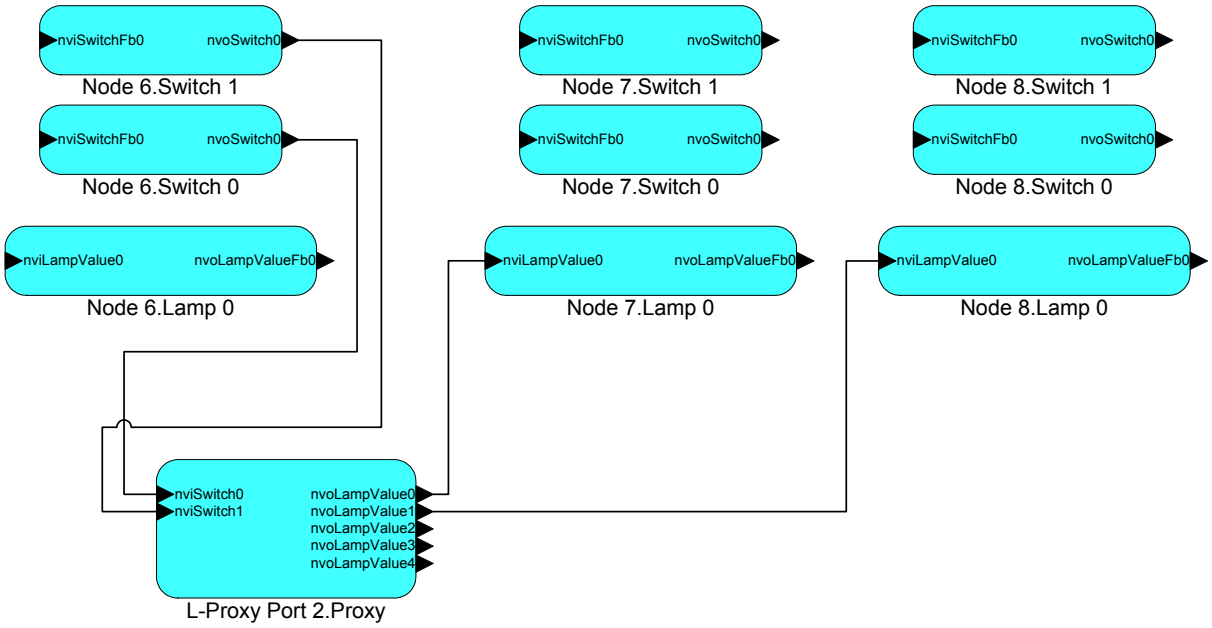


Figure 74 Node 6 only sends packet to L-Proxy Port 2 instead of sending message to nodes 7 and 8 directly. L-Proxy forwards the packet to the final destination nodes 7 and 8.

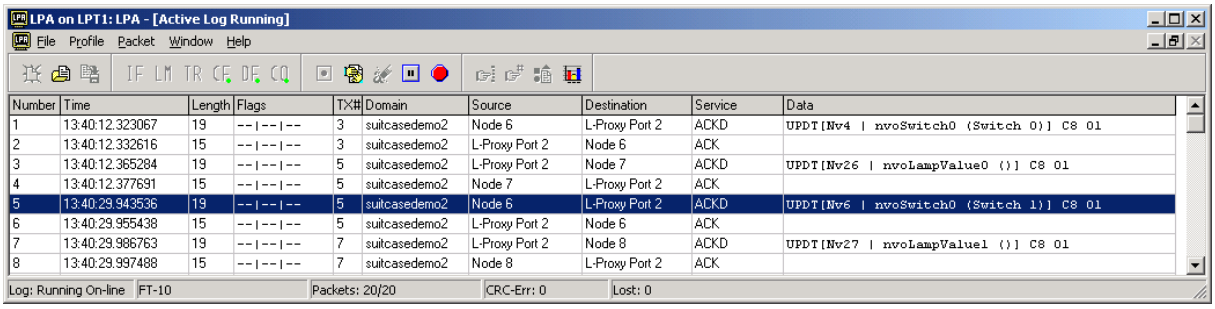


Figure 75 Protocol analyzer log file for the 2 subnet/node bindings shown in Figure 74. Node 6 first sends an update for nviSwitch0 to L-Proxy, which sends an update to node 7. Next node 6 sends an update for nviSwitch1 to L-Proxy, which send an update to node 8.

The following Figure 76 shows the L-Proxy internal bindings required to fulfill the application shown in Figure 74. The first entry connects the input network variable nviSwitch0 on Port 2 to the output network variable nvoLampValue0 also in Port 2. The second entry connects the input network variable nviSwitch1 on Port 2 to the output network variable nvoLampValue1 also in Port 2. Note that these internal bindings are not visible within NL-220, LonMaker or Alex and cannot be seen with a protocol analyzer on the network.

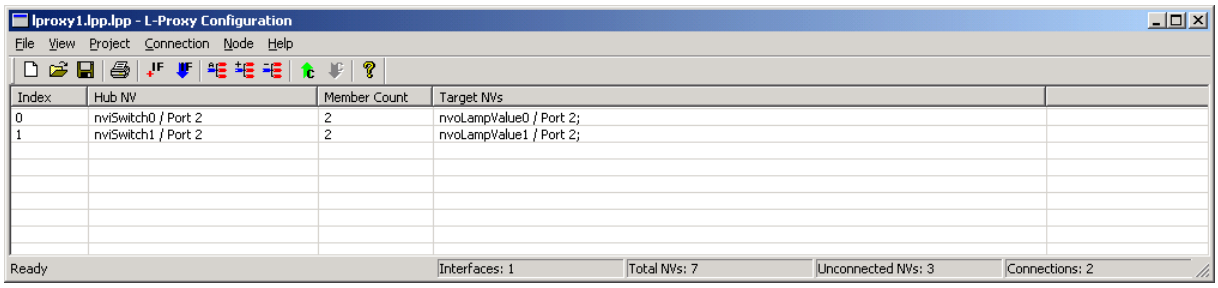


Figure 76 L-Proxy internal binding for the network shown in Figure 74.

6.1.2 Saving a group address

Figure 77 shows a simple network diagram comprising three Neuron Chip powered nodes. Each node contains four LonMark objects, a node object, an actuator object and two sensor objects. The actuator object controls a lamp, the two sensor objects are connected to switches.

The output network variable nvoSwitch0 on node 6 is bound to the input network variables nviLampValue0 on nodes 7 and 8. In this case NL-220 and LonMaker create a group with the 3 group members Node 6, Node 7, and Node 8. Figure 78 shows the log file from a protocol analyzer that clearly shows the acknowledged service using group addressing.

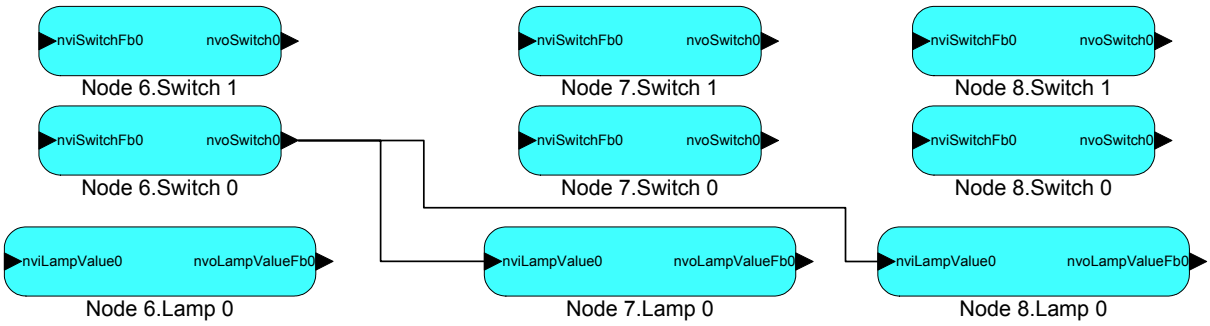


Figure 77 Acknowledged group binding between nodes 6, 7, and 8.

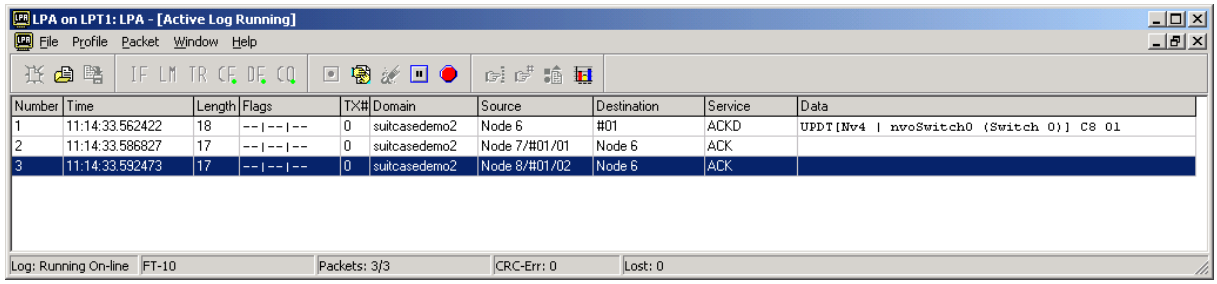


Figure 78 Protocol analyzer log file for the group binding shown in Figure 77. Node 6 uses acknowledged group binding to communicate with nodes 7 and 8.

L-Proxy can be used to break up the group into individual subnet/node addressed bindings, which saves valuable group addresses for group bindings with a higher member count. Figure

79 shows the same network from above but now we have added one port of L-Proxy (Port 2) to break up the group into individual subnet/node addressed messages.

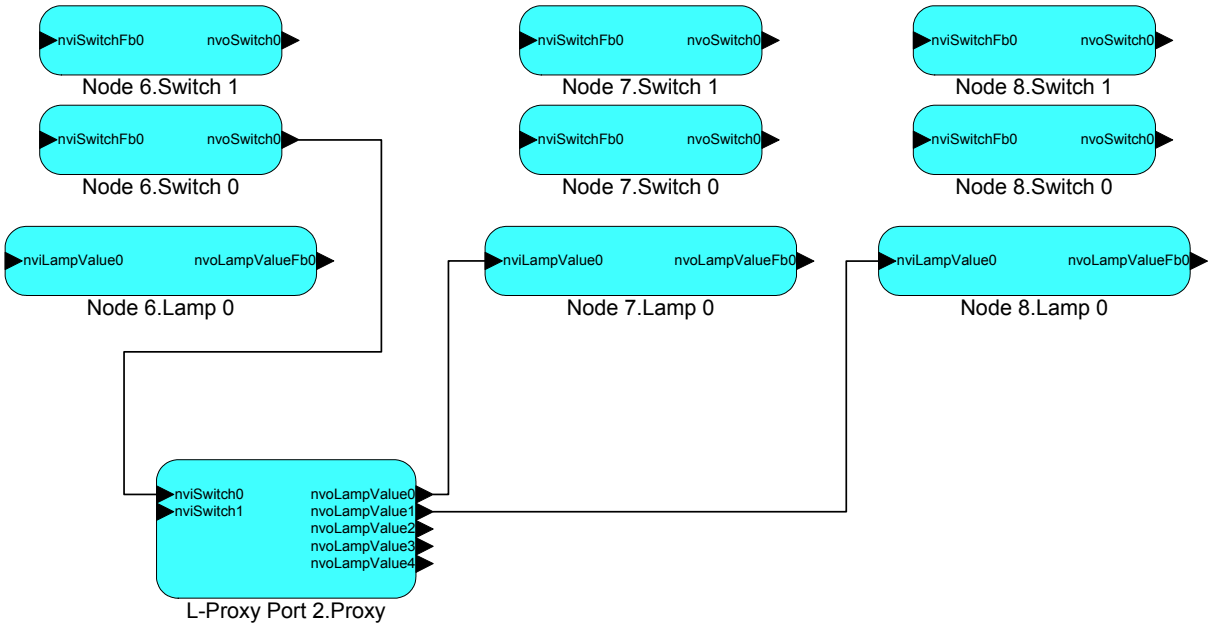


Figure 79 Acknowledged subnet/node addressed messages save a group address entry.

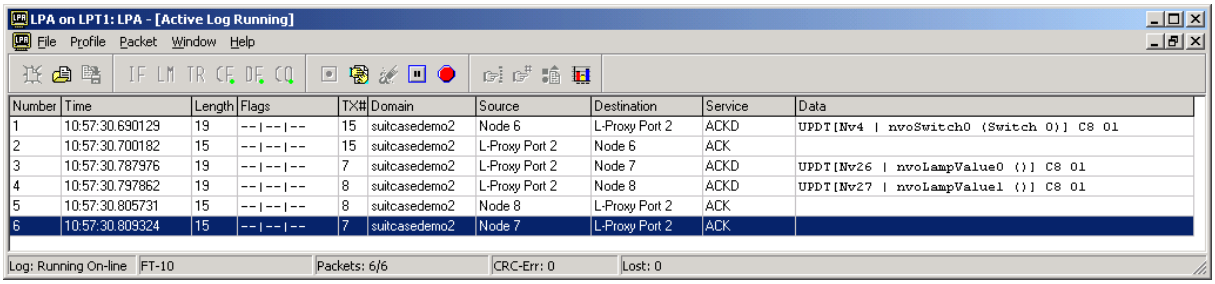


Figure 80 Protocol analyzer log file for the L-Proxy bindings shown in Figure 79. Node 6 uses acknowledged subnet/node binding to communicate with L-Proxy port 2. L-Proxy relays the information to nodes 7 and 8 using acknowledged subnet/node addressed messages.

The following Figure 81 shows the L-Proxy internal bindings required to fulfill the application shown in Figure 79. The first entry connects the input network variable nviSwitch0 on Port 2 to the output network variables nvoLampValue0 till nvoLampValue4 on Port 2. One can see that in this configuration the input network variable nviSwitch0 is actually connected to 5 identical output network variables named nvoLampValue0-4. Therefore up to 5 destination nodes can be bound to the 5 output network variables on L-Proxy. Note that these internal bindings are not visible within NL220/LonMaker/Alex and cannot be seen with a protocol analyzer on the network.

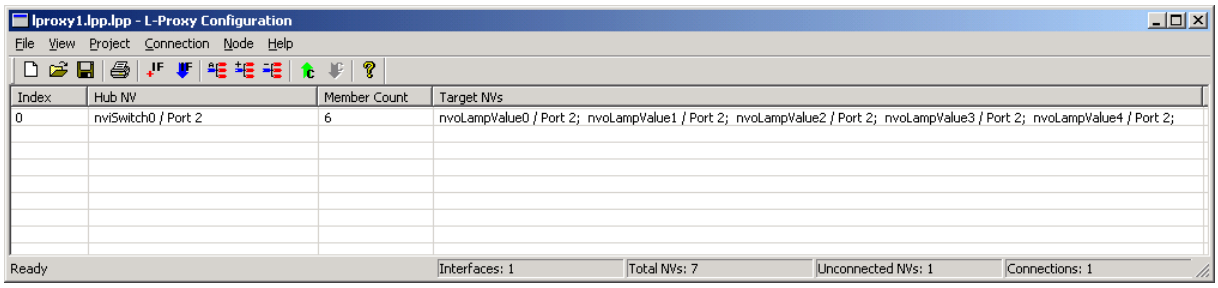


Figure 81 L-Proxy internal binding for the network shown in Figure 79.

6.2 Binding across Domains

Let's assume the network configuration from Table 4. We have 3 nodes Node 6, Node 7, and Node 8, which reside in three different domains. Each of the three nodes has a switch and a lamp object. The switch object is controlled by a switch connected to the node and the lamp object turns on and off a lamp.

Node name	NV name	Domain	Comment
Node 6	NvoSwitch0	4C4F59544543	SNVT_switch output NV
Node 6	NviLampValue0	4C4F59544543	SNVT_switch input NV
Node 7	NvoSwitch0	03	SNVT_switch output NV
Node 7	NviLampValue0	03	SNVT_switch input NV
Node 8	NvoSwitch0	505859	SNVT_switch output NV
Node 8	NviLampValue0	505859	SNVT_switch input NV

Table 4 Network configuration that spans 3 domains.

Let's assume that Node 6 in Domain 4C4F59544543 ("LOYTEC") must turn on the lamp connected to Node 7 in Domain 03 and the lamp connected to Node 8 in Domain 505889 ("PXY"). Furthermore the switch connected to Node 8 must turn on the lamp connected to Node 6 in Domain 4C4F59544543.

This would require a binding across domains, which is not supported in state-of-the-art network installations tools. One can use L-Proxy to make the connections between the 3 domains as explained below.

Lets add L-Proxy1 Port 2 to the network with DomainID 4C4F59544543 as shown in Figure 82. Next we have to create the dynamic input and output network variables on L-Proxy Port 2. Now we can bind the output NV nvoSwitch0 on Node 6 to the corresponding input NV nviSwitch0 on L-Proxy Port 2. We also bind the input NV nviLampValue0 on Node 6 to the output NV nvoLampValue0 on L-Proxy Port 2.

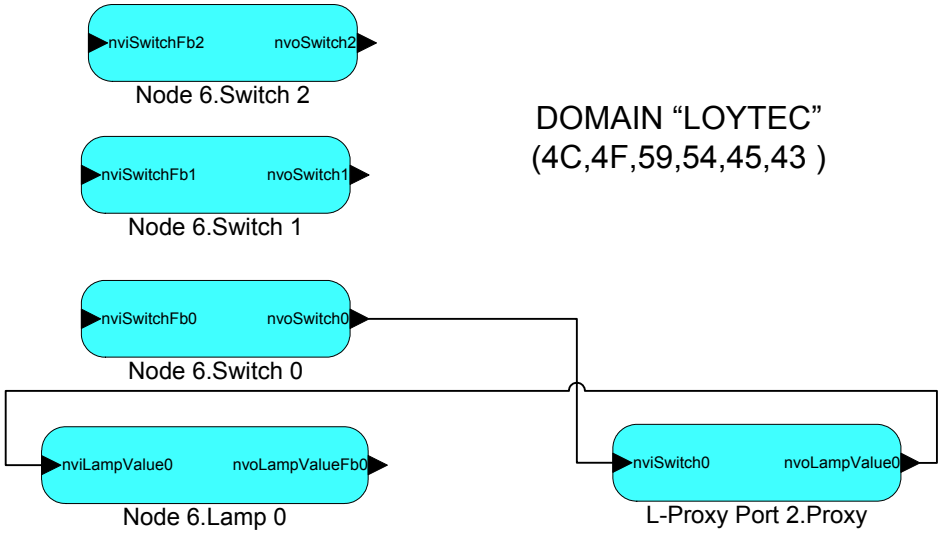


Figure 82 We bind the input and output NVs of Node 6 in domain 4C4F59544543 to the corresponding dynamically created NVs on L-Proxy Port 2.

Lets do similar bindings for the NVs on Node 7 and Node 8 as shown in Figure 83 and Figure 84. for L-Proxy Ports 3 and 4.

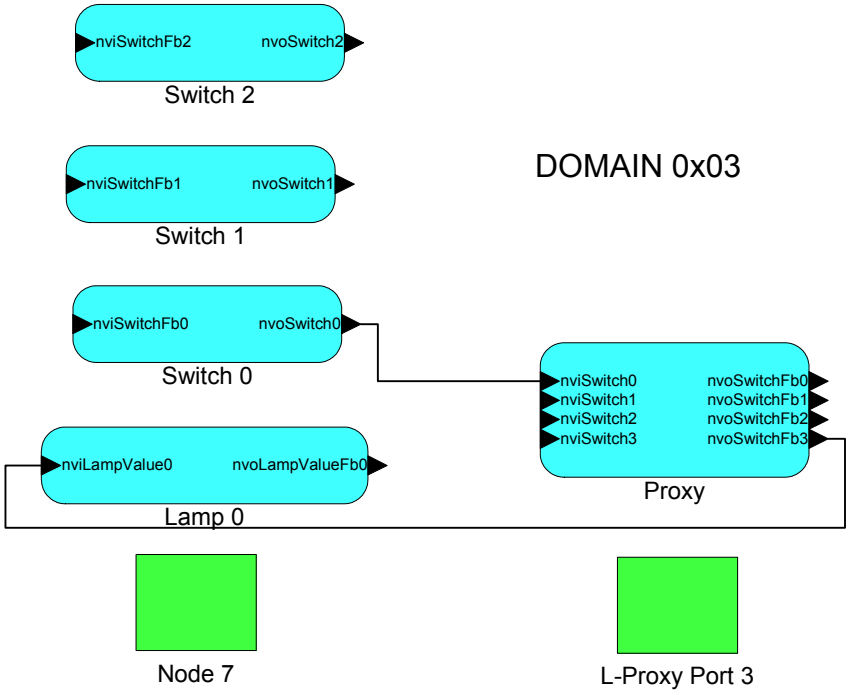


Figure 83 We bind the input and output NVs of Node 7 in domain 03 to the corresponding dynamically created NVs on L-Proxy Port 3.

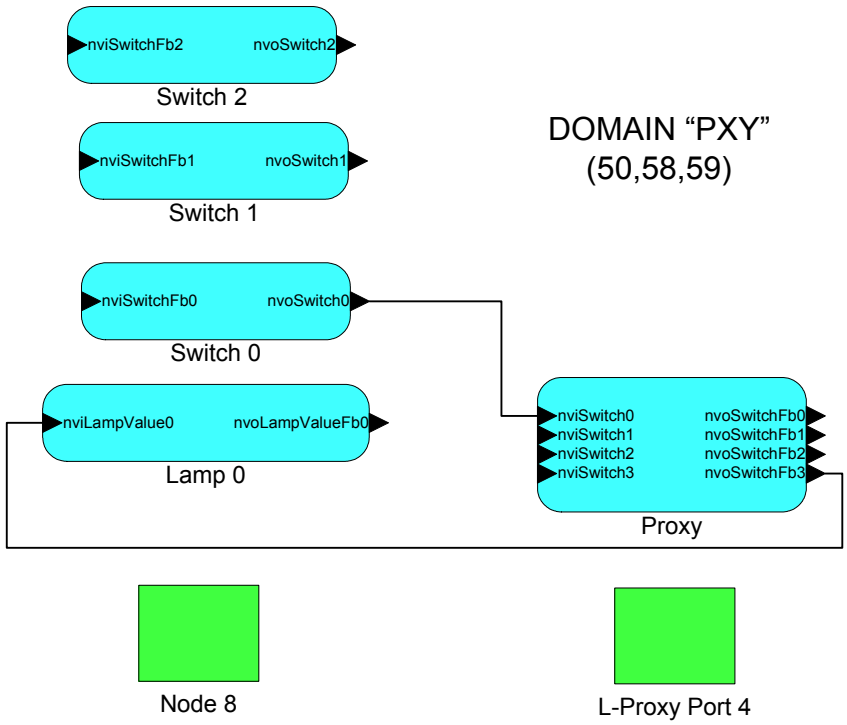


Figure 84 We bind the input and output NVs of Node 8 in domain 505859 to the corresponding dynamically created NVs on L-Proxy Port 4.

After extracting the port interface definitions for L-Proxy Ports 2-4 we can now create the Connections with the L-Proxy configuration utility. The results are shown in Figure 85. The first connection connects the switch on Node 6 (Port 2) with the lamp on Nodes 7 (Port 3) and 8 (Port 4). The second connection connects the switch on Node 8 (Port 4) with the lamp on Node 6 (Port 2).

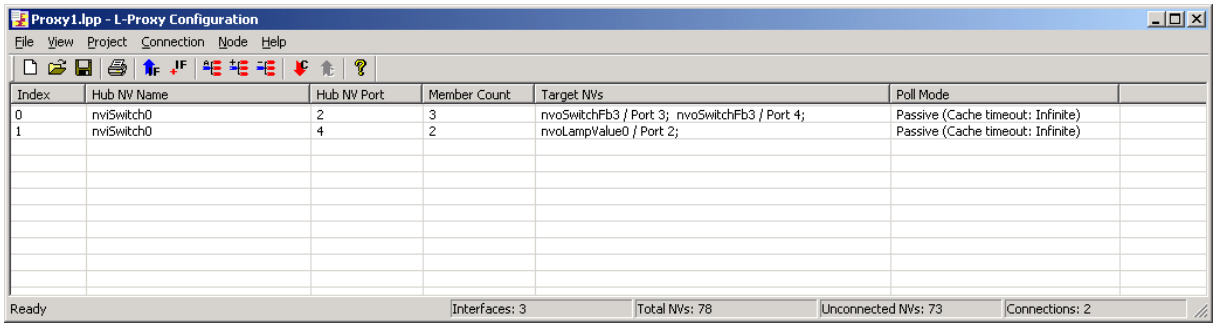


Figure 85 These 2 connections implement the desired functionality as specified: Node 6 in Domain 4C4F59544543 ("LOYTEC") must turn on the lamp connected to Node 7 in Domain 03 and the lamp connected to Node 8 in Domain 505889 ("PXY"). Furthermore the switch connected to Node 8 must turn on the lamp connected to Node 6 in Domain 4C4F59544543.

If we now flip the switch on Node 6 the protocol analyzer log file will show the results from Figure 86.

Packet #1: NV update sent from nvoSwitch0 on Node 6 to L-Proxy Port 2 nviSwitch0.

Packet #2: L-Proxy Port 2 sends an acknowledgement back to Node 6 in domain 4C4F59544543.

Packet #3: L-Proxy Port 3 forwards the NV update from “Node 6 nvoSwitch0” to Node 7 in domain 03.

Packet #4: L-Proxy Port 4 forwards the NV update from “Node 6 nvoSwitch0” to Node 8 in domain 505859.

Packet #5: Node 7 sends an acknowledgement back to L-Proxy Port 3.

Packet #6: Node 8 sends an acknowledgement back to L-Proxy Port 4.

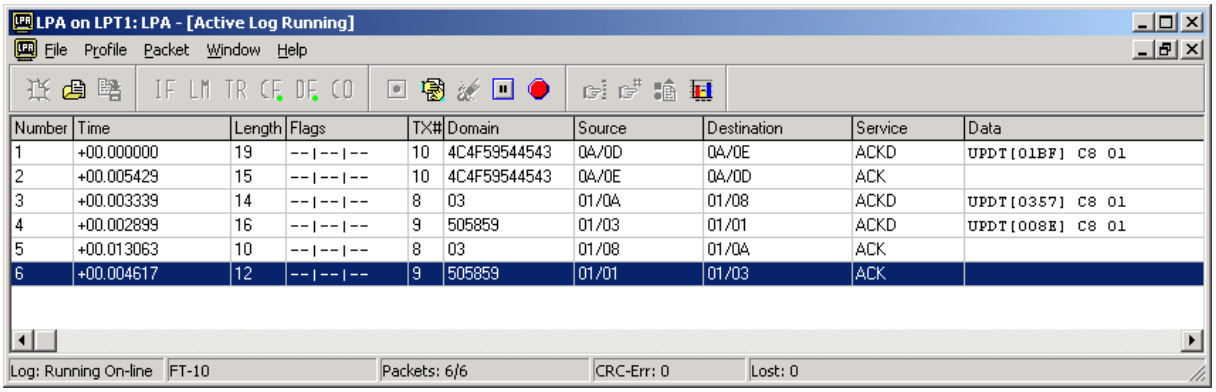


Figure 86 Turning on the switch on Node 6 creates the following packet stream in order to turn on the lamps on Node 7 and Node 8.

The second part of our desired network configuration is shown in Figure 87. When we flip the switch on Node 8 the lamp on Node 6 should turn on. The following packet streams can be seen on a protocol analyzer log output.

Packet #1: NV update sent from nvoSwitch0 on Node 8 to L-Proxy Port 4 nviSwitch0.

Packet #2: L-Proxy Port 4 sends an acknowledgement back to Node 8 in domain 505859.

Packet #3: L-Proxy Port 2 forwards the NV update from “Node 8 nvoSwitch0” to Node 6 in domain 4C4F59544543.

Packet #4: Node 6 sends an acknowledgement back to L-Proxy Port 2.

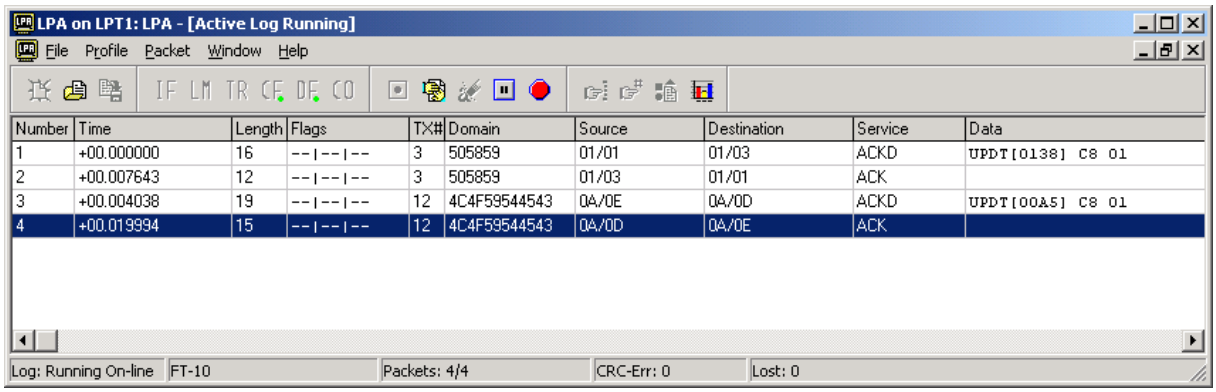


Figure 87 Turning on the switch on Node 8 creates the following packet stream in order to turn on the lamp on Node 6.

Note, in this example the 3 L-Proxy ports can either be connected to the same network cable (as shown in the log file in Figure 86) or they can be connected to individual network cables.

6.3 Poll-Update Proxy

Lets assume a scenario with a node that cannot send updates by itself but must be polled. On the other hand the destination of the information e.g. a SCADA system or OPC server cannot poll but must receive updates. L-Proxy connections can now be defined as “active poll”, which means that the source node is actively polled by L-Proxy at a defined timer interval and the updates are sent to the destination node. Note, that active poll is only supported with firmware version 2.0 and higher.

Lets assume that the output network variable nroPort in Figure 88 on node lproxy1 2 cannot send updates by itself but must be polled. In our case the lproxy1 3 polls this output network variable with the poll rate of 30s as defined in the Plug-in connection dialog.

In this example the poll mode is set to active poll and the poll configuration is set to Forward polls.

Forward poll means that if poll requests are received from the destination node (lproxy1 4) on the output side of L-Proxy, they are forwarded to the source node (lproxy1 2) in addition to the periodic polls generated by L-Proxy.

The second poll configuration option is “Use value from last poll cycle”, which means that additional poll requests from the destination node (lproxy1 4) are not forwarded to the source node but immediately return a response with the value of the network variable that was polled in the last active poll-cycle.

The poll-rate in seconds can be specified in the Poll rate field.

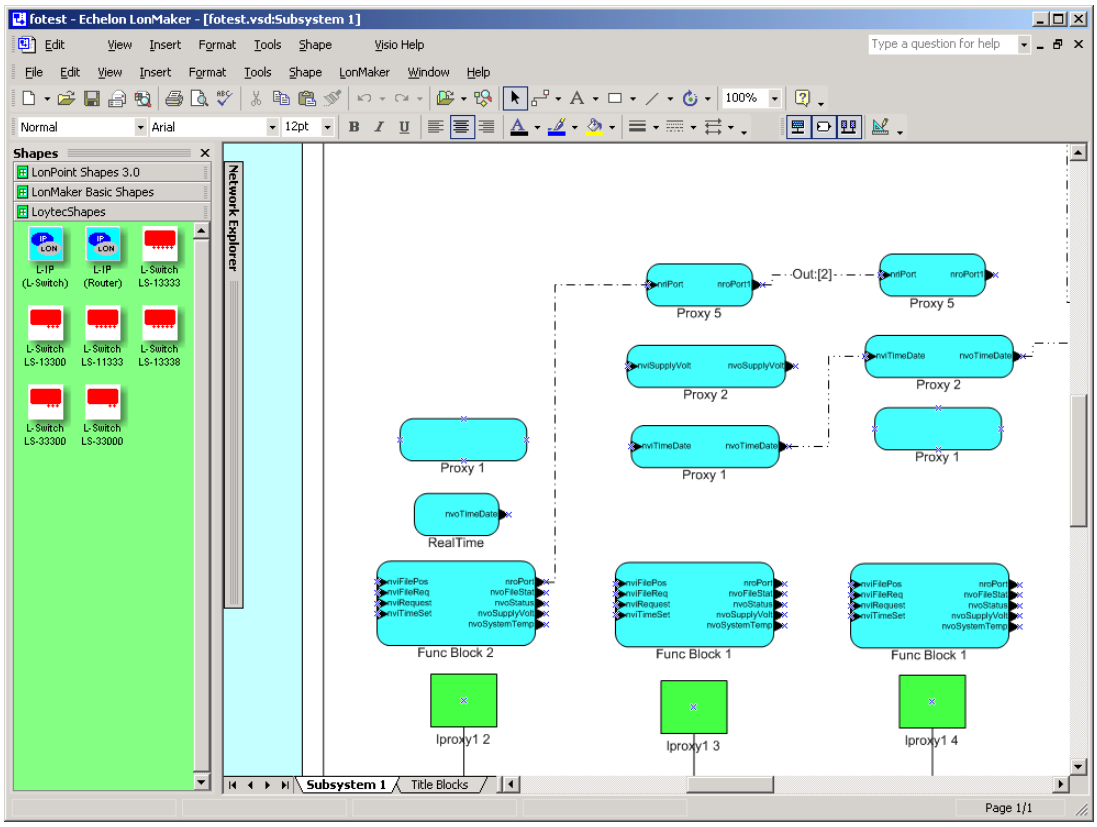


Figure 88 Active poll of nvo nroPort on node lproxy1 2 from nvi nriPort on node lproxy1 3.

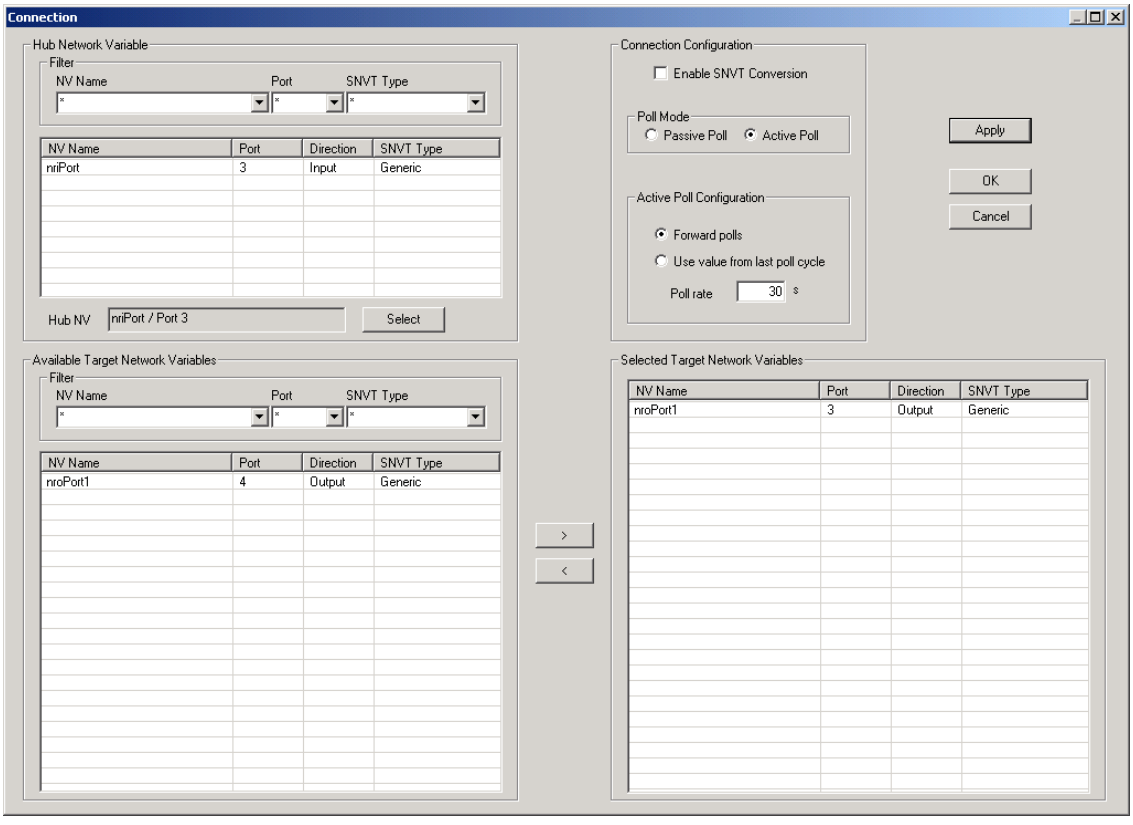


Figure 89 Active poll connection with a poll-rate of 30s.

6.4 Poll Once Initial Value Proxy

Some scenarios require that at power-up or after a reset the network variable values from the source node should be polled-once to obtain the current state of these network variables. In the L-Proxy configuration utility a poll-once flag can be set for each individual L-Proxy internal connection as shown in Figure 90. The poll rate will be automatically adjusted by L-Proxy to not saturate the network channel.

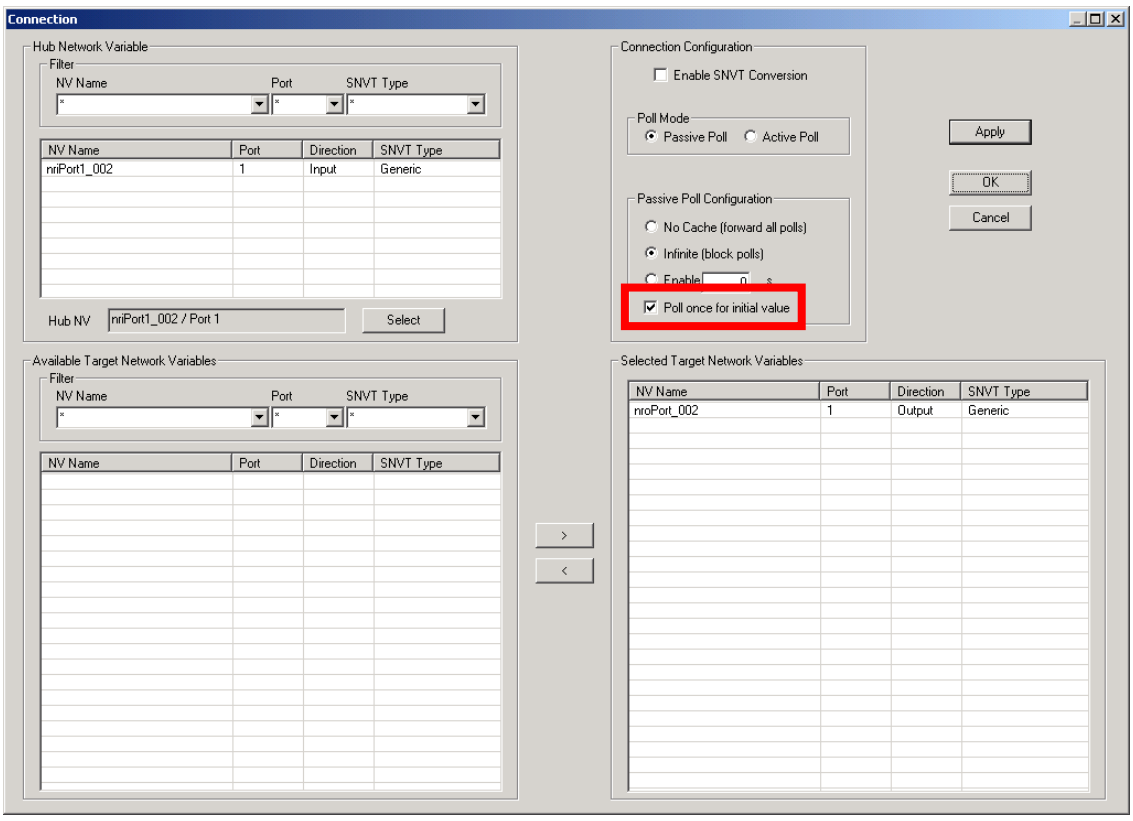


Figure 90 The poll once for initial value flag is set for this connection.

6.5 SNVT Translation

Translation between different scalar SNV-Types is a new feature starting with firmware version 2.0. L-Proxy can automatically convert between SNVTs of the same SI unit and if the SNVT is of scalar type.

The example in Figure 91 shows a node 1 that has network variables nvoAmpere with SNVT_amp and nvoTemperature with SNVT_temp. It also shows a node 2 with network variables nviAmpere_mil with SNVT_amp_mil and nviTemperature_f with SNVT_temp_f. Since the network variables are of different type on node 1 and on node2 these NVs cannot be bound directly. L-Proxy can act as a translator between the different network variable types as shown in Figure 91.

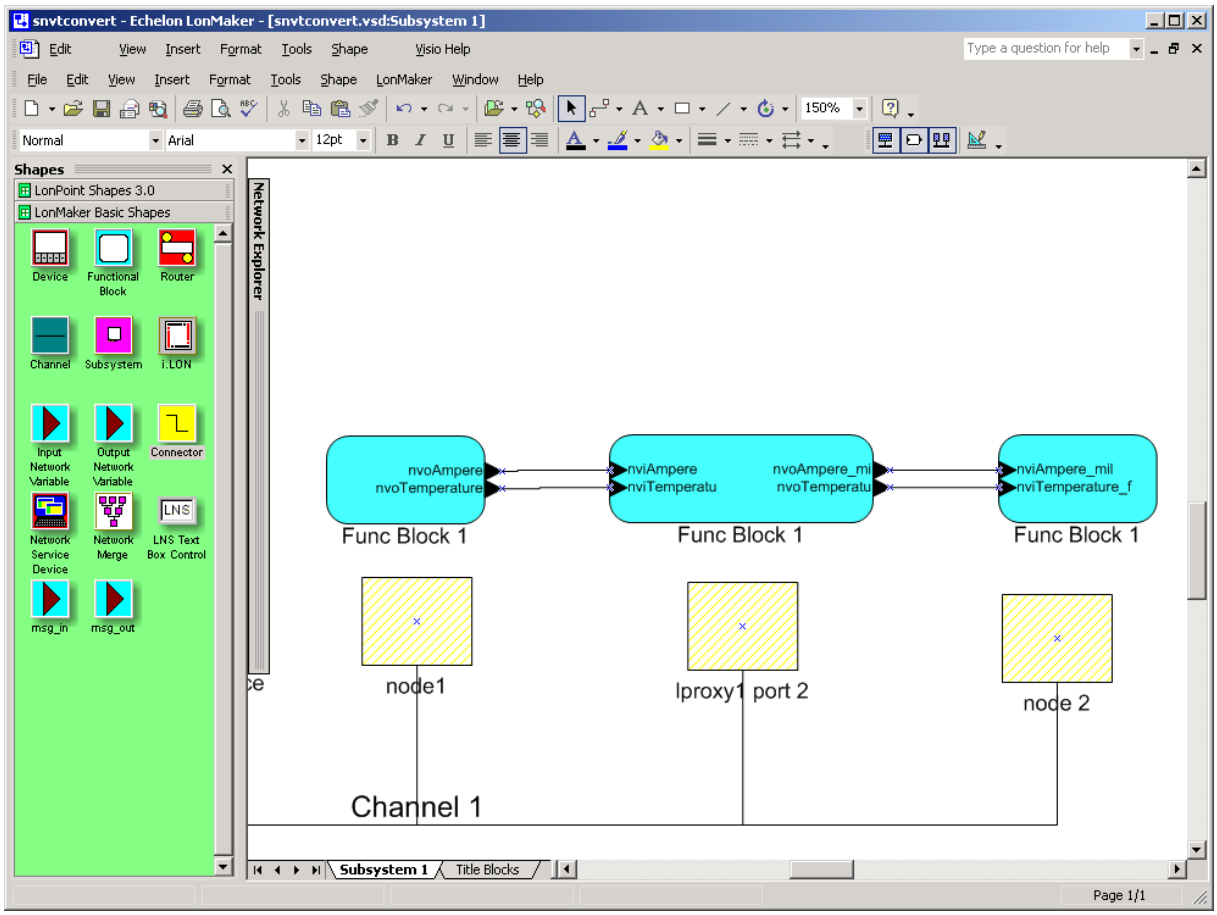
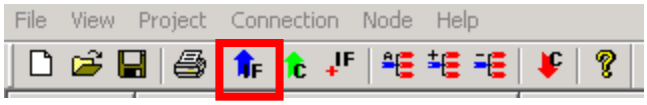
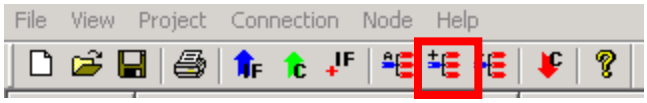


Figure 91 Example for scalar snvt translation.

The next step is to start the L-Proxy configuration plug-in on node lproxy1 port2 and to extract the network interface.



Create a new connection with



,

which opens the window shown in Figure 92.

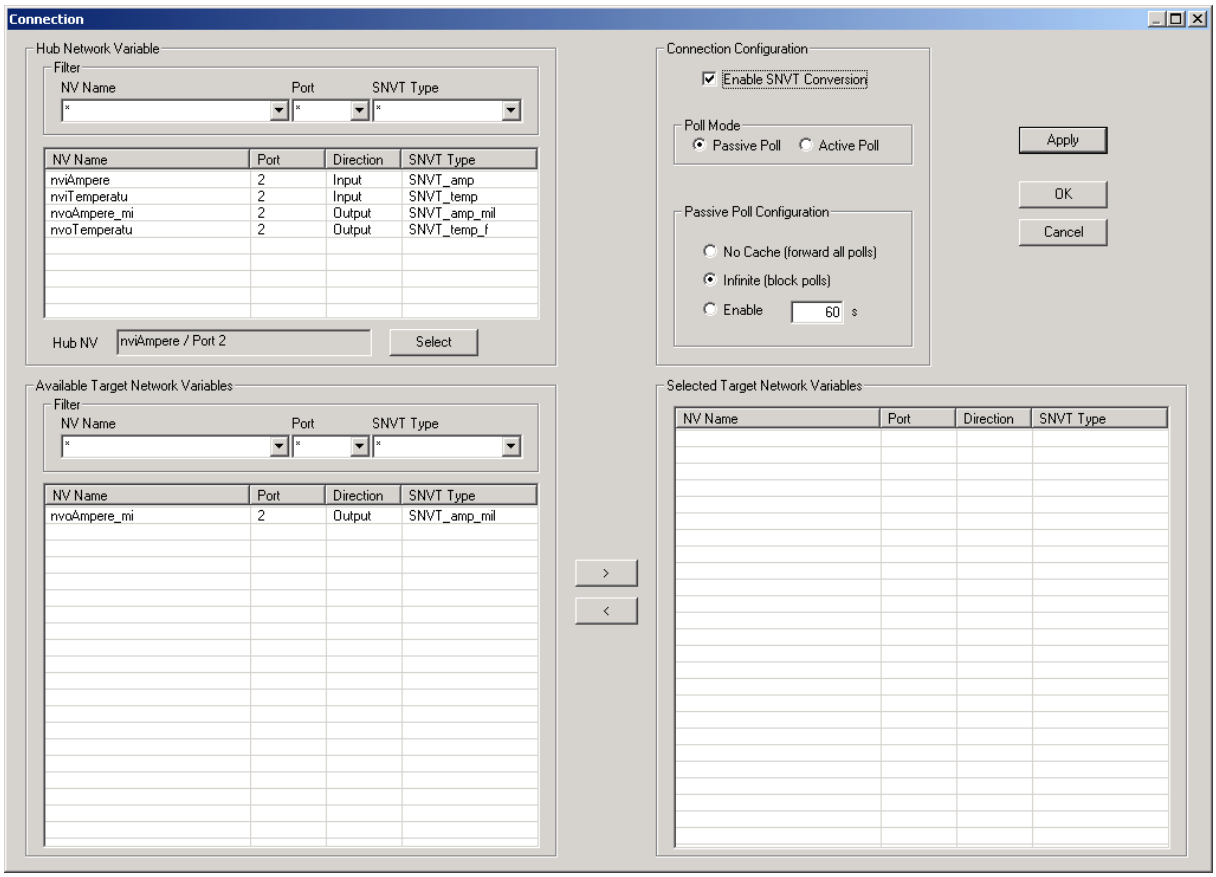


Figure 92 Enable SNVT Conversion to allow connections of NV with different type.

All source network variables are shown in the top left window. Select nviAmpere on Port 2 with a double-click. The nviAmpere appears in the Hub NV box but no network variables are shown in the Available Target Network Variables window. Please check the Enable SNVT Conversion box in order to active SNVT conversion. Suddenly the network variable nvoAmpere_mil appears in the Available Target Network Variables window. SNVT_amp and SNVT_amp_f are of the same scalar SI type, hence L-Proxy is able to make the conversion between the 2 types.

Click on > to select the target NV as shown in Figure 93. Click on OK to create the connection.

Now repeat the same steps for the second network variable of type SNVT_temp and SNVT_temp_f.

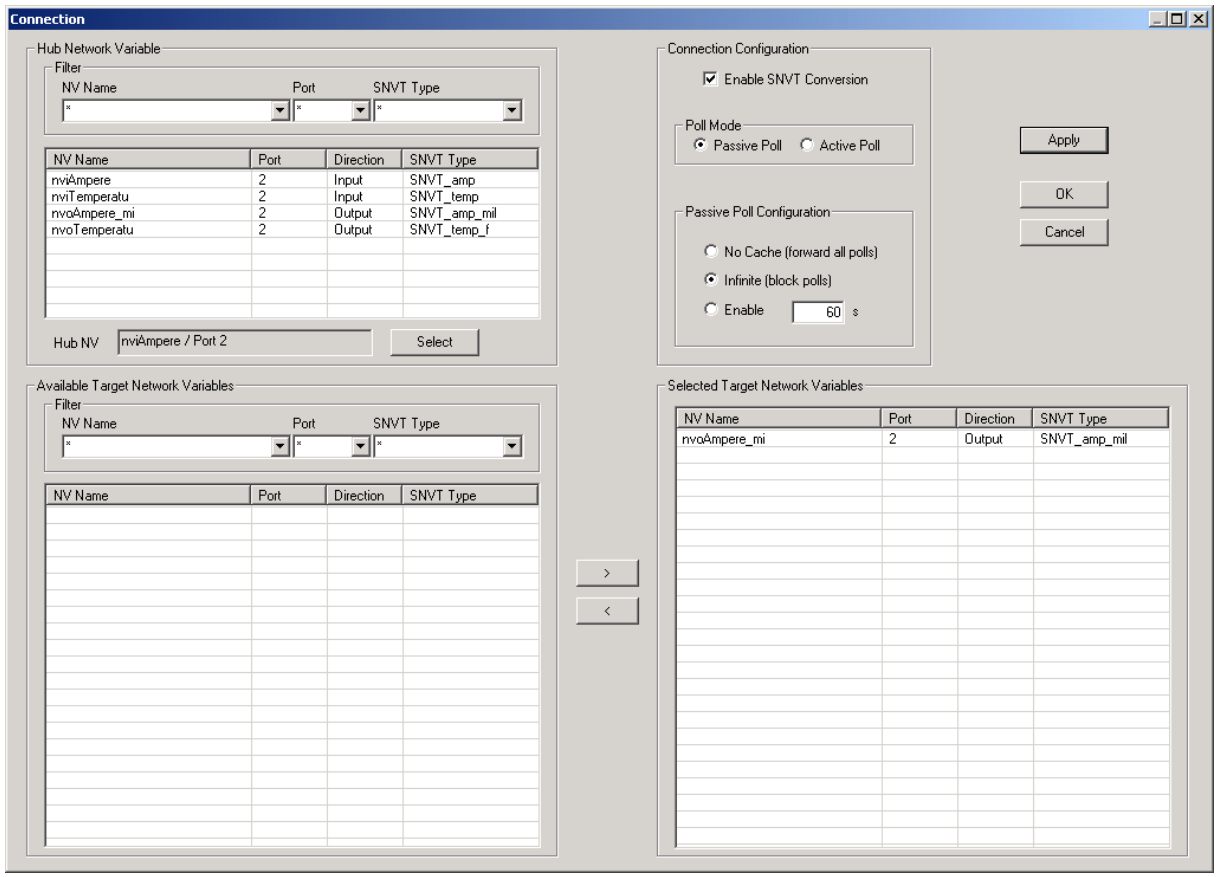


Figure 93 Create a connection between type SNVT_amp and type SNVT_amp_mil.

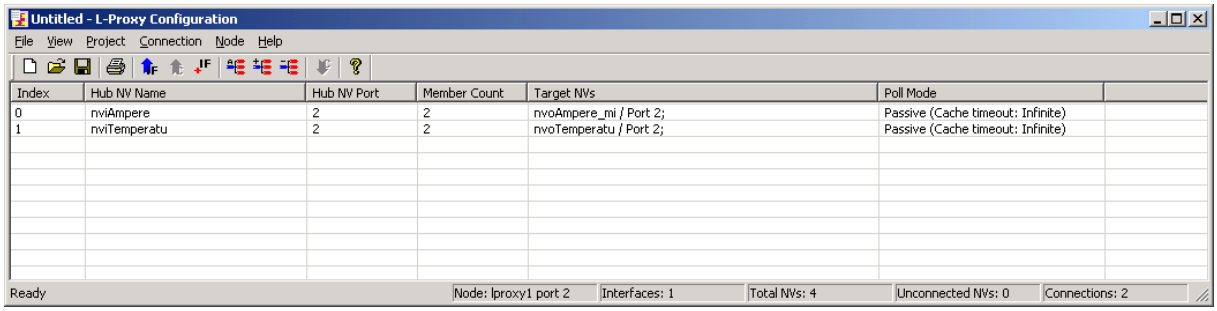


Figure 94 The two connections convert between the different network variable types used in the example in Figure 91.

7 User Interface

7.1 Status Button

The L-Proxy is equipped with a status button (see Figure 11). When pressing the status button shortly during normal operation of the L-Proxy, it sends a "Service Pin Message" on every port.

When pressing and holding the status button the port LEDs will light up orange one after the other. Releasing the status button while the port LED is orange will send a "Service Pin Message" only on the port where the status LED is lit.

Note that every L-Proxy port has its own unique node ID ("Neuron ID").

7.2 Wink action

When receiving a wink network management message all the port LEDs light up orange one after the other back and forth three times ("scanner light"). At the end of this cycle the port LED on which the wink command was received lights up three times.

7.3 LED Signals

7.3.1 Status LED

The L-Proxy is equipped with a two-color status LED (green and red, see Figure 11).

When power is applied to the L-Proxy, the status LED is green. During boot-up the status LED is used to signal error conditions (red).

After boot-up the status LED is used to signal write accesses to flash memory (switched to red for the duration of every write access).

7.3.2 Port LEDs

Each port on the L-Proxy has a three color LED (green, red and orange, see Figure 11). The port LEDs have the following behavior:

- If a port was not detected or an error occurred during detection, the associated port LED is permanently red.
- If a port is performing a bit-rate auto-detection, the associated port LED will be flashing orange at a frequency of 1 Hz.
- Whenever a network variable update, a poll, or a configuration message is received on a port, the associated port LED will flash green.

- If the L-Proxy fall-back image was booted, all port LEDs (see Section 8), except those where a bit-rate auto-detection is performed, are flashing red once per second.

7.4 Console

The L-Proxy is equipped with a serial interface to

- display the results of the self test
- allow advanced configuration via a console menu
- upgrade the L-Proxy firmware

To use the serial interface the console connector (see Figure 11) of the L-Proxy can be connected to the RS-232 port of a PC. Now, the PC can communicate with the L-Proxy using a standard terminal program with the communication settings set to 38,400 bps / 8 data bits / no parity / 1 stop bit.

7.4.1 Self Test

Whenever the L-Proxy comes out of reset it performs a self-test. If the self-test passed successfully, all port LEDs successively turn green for 0.5 seconds. If a failure occurs during self-test, the status LED is flashing red and the L-Proxy is reset.

The console output of a successful boot sequence on an L-Proxy with 5 ports on the console reads as follows:

```
LOYTEC electronics GmbH
www.loytec.com

Testing Board ID (0E)                                    Passed
Testing RAM                                              Passed
Testing boot loader                                      Passed
Testing fallback image                                   Passed
Testing primary image                                    Passed
Testing Flash                                            Passed

Loading primary image                                    Passed

Port 1 detected                                          Passed
Port 2 detected                                          Passed
Port 3 detected                                          Passed
Port 4 detected                                          Passed
Port 5 detected                                          Passed

L-PROXY(c)
LOYTEC electronics GmbH
Jul 21 2003 - V2.1.0
System has passed self-test and is active …
```

The duration of a successful boot sequence of an L-Proxy with 5 ports is typically 7 seconds.

7.4.2 Console Menu

After booting the L-Proxy displays the following console menu:

```
LOYTEC electronics GmbH
www.loytec.com

L-PROXY Configuration Menu
==========================

[1]  Show device information
[2]  Update firmware
[3]  System configuration
[4]  EIA-709 configuration
[8]  Reset configuration (factory defaults)
[0]  Reset L-Proxy

Please choose:
```

The menu items are described below:

1 - Show device information

This menu item shows some information about L-Proxy and the current firmware version.

2 - Update firmware

This menu item allows updating the L-Proxy firmware via the serial interface (console). See Section 8.2 for detailed instructions.

3 – System configuration

This menu item allows setting the date and time for the internal real-time clock.

4 – EIA-709 configuration

This menu item allows setting transceiver parameters for the EIA-709 network interface like bit-rate auto-detection, enable or disable the backbone mode, transceiver configuration. See Section for 7.4.2.1 details.

8 – Reset Configuration (factory defaults)

This menu item allows setting the L-Proxy into its factory default state. This clears all dynamically created network variables and all internal connections. Please select this menu when you move L-Proxy from one project into another project.

0 - Reset L-Proxy

This menu item resets the L-Proxy.

7.4.2.1 EIA-709 Configuration Menu

This menu allows changing the EIA-709 transceiver configuration, enable the backbone mode for TP-1250 transceivers, and enable bit-rate auto-detection for RS-485 transceivers.

```
EIA709 Configuration Menu
=========================
```

```
[1]  Port 1: XF/TP-1250 (1250 kBit)
     Port 2: FT-10
     Port 3: FT-10
     Port 4: FT-10
     Port 5: FT-10

[0]  Backbone mode configuration source: DIP SWITCH

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

Please choose:
```

Figure 95 EIA-709 configuration menu

7.4.2.1.1 Option 1 to 5 - Change transceiver configuration for Port 1 to 5

This menu item allows setting the default transceiver configuration for port 1 to 5 if there are different possible transceiver configurations.

For TP-1250 transceivers it is possible to set the transceiver settings to backbone mode. See Section 10.2 for an in-depth discussion of this feature.

For RS-485 transceivers it is possible to manually set the bit-rate or choose bit-rate auto-detection. If auto-detection is enabled the detected bit-rate is shown.

Please contact LOYTEC support (see Section 9.7) if you want to change the default transceiver configuration (e.g. PLT-22 in CENLEC or NON-CENELEC mode).

7.4.2.1.2 Option 9 - Bit-rate auto-detection configuration source (only if RS-485 port is present)

This menu item allows to set what decision element should be used to enable or disable the bit-rate auto-detection. If [1] DIP Switch is selected the value set on the DIP switch (see Table 1) is used to enable or disable the bit-rate auto-detection. If [2] Software is selected the setting made with Option 1 - 5 is used to enable or disable the bit-rate auto-detection.

7.4.2.1.3 Option 0 - Backbone mode configuration source (only if TP-1250 port is present)

This menu item allows to set what decision element should be used to enable or disable the L-Switch collision-less backbone mode (see Section 10.2). If [1] DIP Switch is selected the value set on the DIP switch (see Table 1) is used to enable or disable the backbone mode. If [2] Software is selected the setting made with Option 1 - 5 is used to enable or disable the backbone mode.

7.5 Network Diagnostics

The L-Proxy provides simple network diagnostics via its port LEDs.

The port LED is flashing green if a network variable update is received or transmitted and if a network variable poll is received or transmitted and if a new configuration is loaded into the L-Proxy.

8 Updating the L-Proxy Firmware

The L-Proxy firmware supports remote upgrade over the network and the serial console.

To guarantee that the L-Proxy cannot be destroyed due to a failed firmware update the L-Proxy firmware consists of two images:

1. Fall-back image
2. L-Proxy application image

The fall-back image is write protected in flash memory and provides everything needed to talk to the L-Proxy platform over the network. The L-Proxy application image is designed to be updated over the network whenever there is a need to do so.

The fall-back image makes sure that the L-Proxy comes up in a status where the maintenance software can at least talk to the L-Proxy platform and can download a new L-Proxy application image.

When the L-Proxy boots up with the fall-back image, all port LEDs are flashing red. In this state it does not forward any messages.

Note: All configuration settings and the domain information will be lost when the firmware is updated. One must re-commission all ports in use on L-Proxy and download the L-Proxy configuration after a firmware update.

8.1 Firmware Update via the Network

Basically firmware downloads can be performed on every L-Proxy port. However, since the L-Proxy is not based on a Neuron Chip a new firmware image cannot be downloaded with a standard tool. Rather, a designated tool, the LSD Tool (see Section 10.1), must be used. See the LSD Tool documentation for details on how to download a new L-Proxy firmware via the network.

8.2 Firmware Update via the Console

To download the firmware via the console the L-Proxy must be connected to the RS-232 port of a PC via its console interface as described in Section 7.4. You will need the LOYTEC serial upgrade tool (LSU Serial Upgrade Tool), which can be downloaded from our homepage at www.loytec.com.

Double click on the *.dlc file that comes with the new firmware package. This should start the LSU Tool and load the firmware image referenced in the dlc file. Please note that the dlc file and the dl file must be stored in the same folder. The start window of the LSU tool is shown in Figure 96.

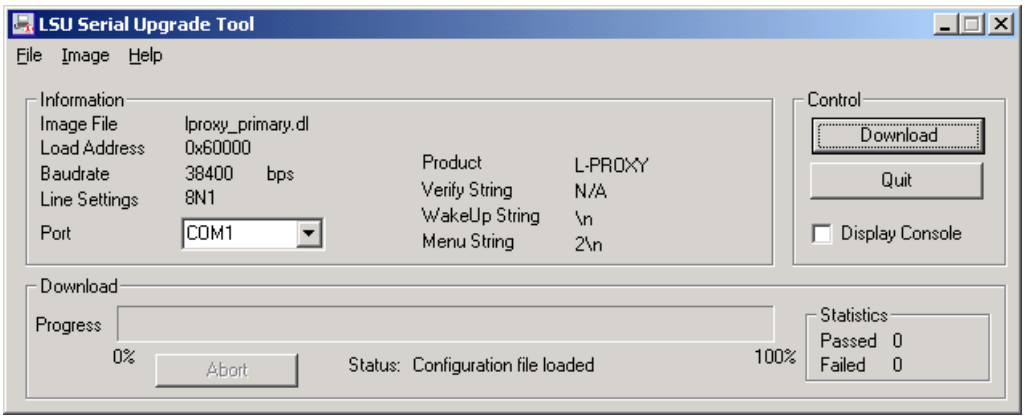


Figure 96 LSU Serial Upgrade Tool in idle mode.

If L-Proxy is not connected to COM1 you can change the port to COM1, COM2, COM3, or COM4. Make sure that the product shown under "Product" matches the device you are upgrading. Note that Figure 96 and Figure 97 do not necessarily show the proper product.

Press "Download" to start the download. A progress bar as shown in Figure 97 can be seen.

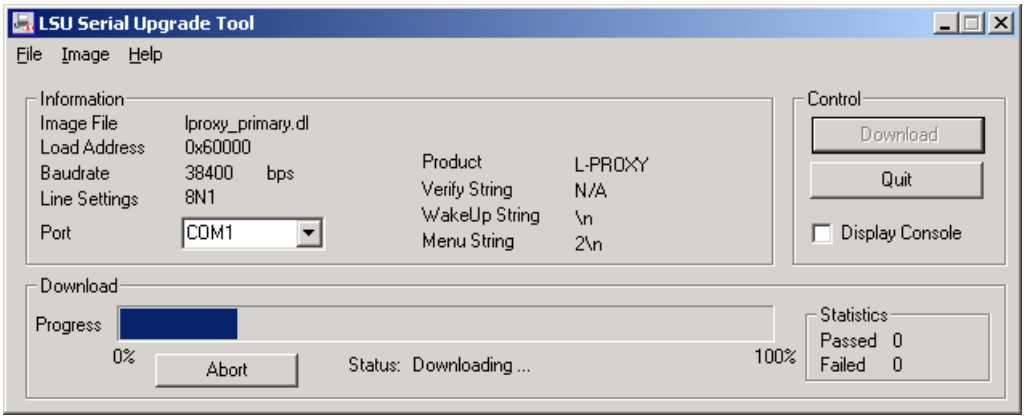


Figure 97 Progress bar during firmware download.

If the upgrade is successful the following window appears.

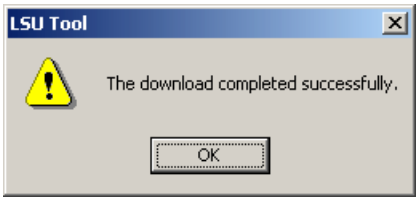


Figure 98 Successful firmware upgrade.

After successfully completing the firmware upgrade the L-Proxy must be reset. This can be accomplished by

a) Removing power for at least 5 seconds.

b) Pressing the reset button on the front panel with a small pin.

c) Issuing the Reset command in the user interface over the serial port.

In order to issue a reset command over the serial port one can open the console window in the LSU Tool ("Display Console" under Control), move the mouse pointer into the console window area and click the left mouse button. Now enter "0" on the keyboard and press Enter. This will issue a reset command and the L-Proxy will reset and execute the newly downloaded firmware image.

Double check that the new firmware is executed by selecting 1 and pressing Enter in the console window. This will bring up the device information, which shows the current firmware version.

8.3 Commissioning and Configuration after Firmware Download

After a firmware upgrade each of the 5 L-Proxy ports (nodes) must be *commissioned* in e.g. NL220 or LonMaker. Commissioning the port downloads the address (domain) information and creates the dynamic network variables. After all ports (nodes) have been commissioned the L-Proxy configuration utility (plug-in) must be opened and the project file *projectname.lpp* must be downloaded into L-Proxy (Node => Download Configuration).

8.4 Upgrade from Firmware Version 1.x to Version 2.x

After downloading the new firmware version into L-Proxy one must replace the device in LonMaker, NL-220 or Alex by selecting Replace as shown in Figure 99.

Note! Upgrading L-Proxy from version 1.x to 2.x will loose all bindings but the dynamically created network variables will be saved.

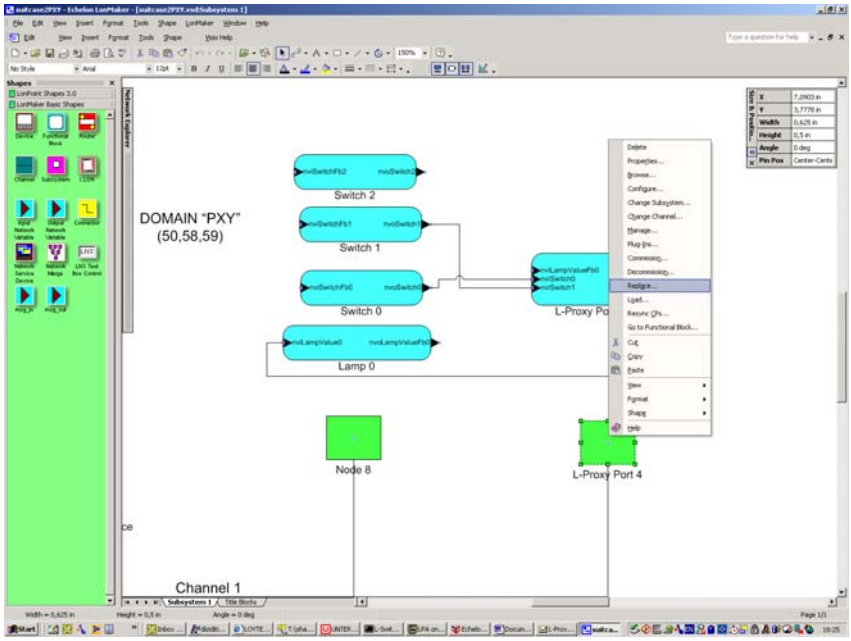


Figure 99 Replace the L-Proxy device.

Select the new device template for the FT-10 or the TP-1250 port.

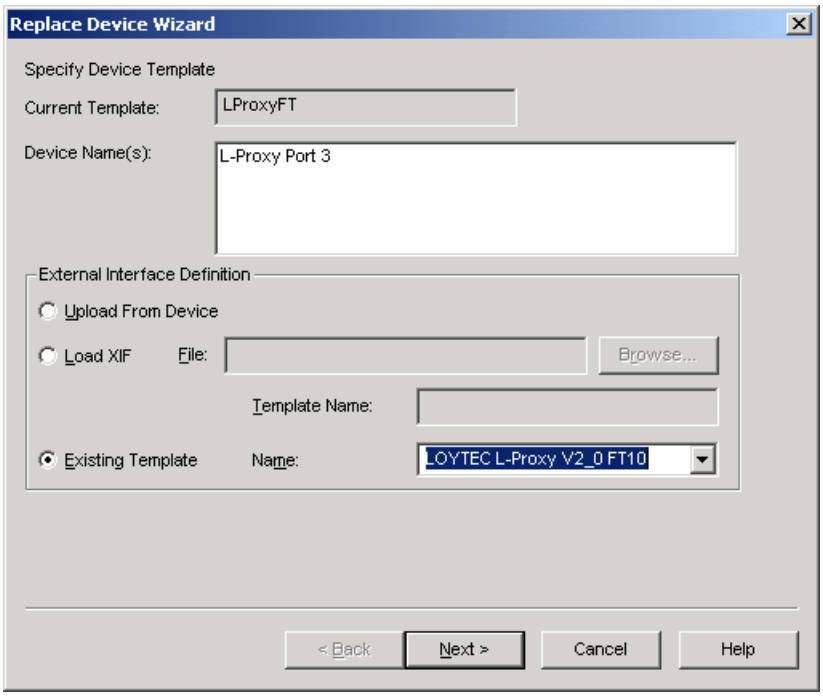


Figure 100 Select the new device template.

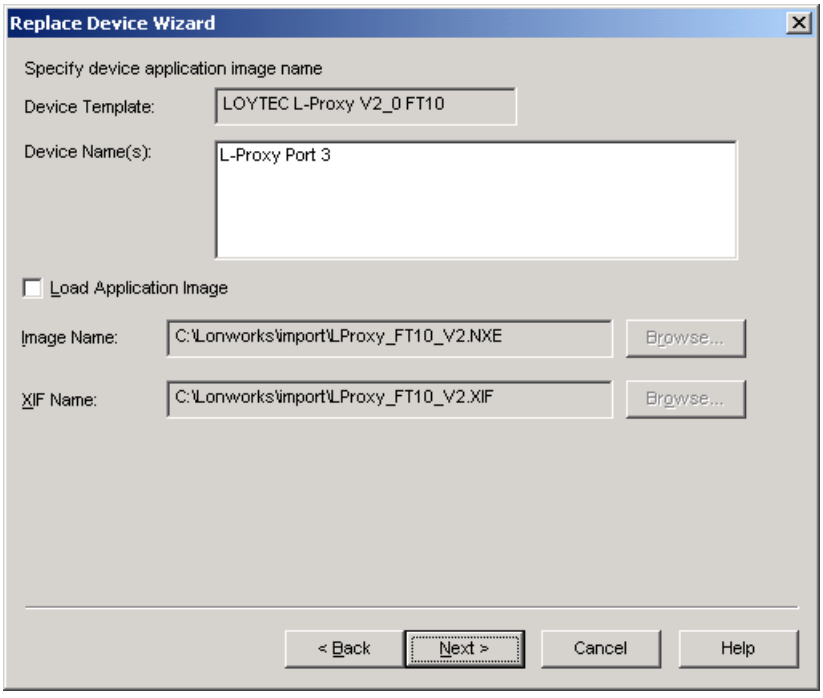


Figure 101 Click on Next.

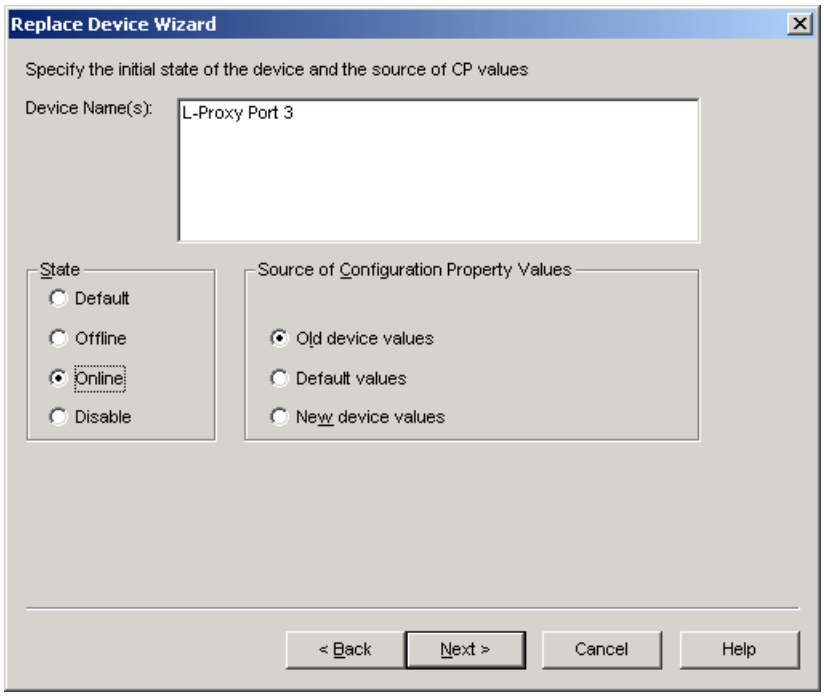


Figure 102 Select online and click Next.

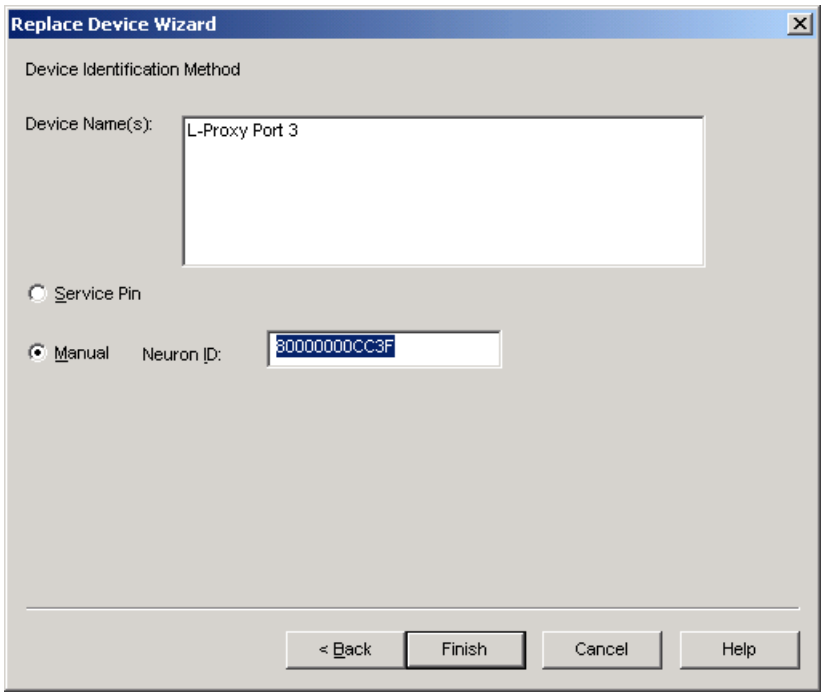


Figure 103 Use the Neuron ID already stored in the database.

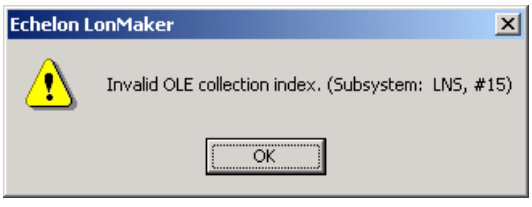


Figure 104 This warning might appear after commissioning the new L-Proxy port.

9 Troubleshooting

9.1 All port LEDs are flashing red

Problem

All port LEDs are flashing red at a rate of approx. once per second and the L-Proxy does not relay messages.

Explanation

Somehow the primary image was destroyed and the fall-back image was booted (see Section 8). This image does not support relaying messages. It only allows downloading a new firmware.

Solution

If this problem occurs because a firmware update was attempted (and failed somehow), simply retry downloading the new firmware image.

If no firmware update was attempted, please contact LOYTEC support (see Section 9.2).

9.2 LonMaker Warning appears during installation

Problem

When adding a new L-Proxy Port to the project the following warning messages might appear.

Figure 105 Click OK to continue.

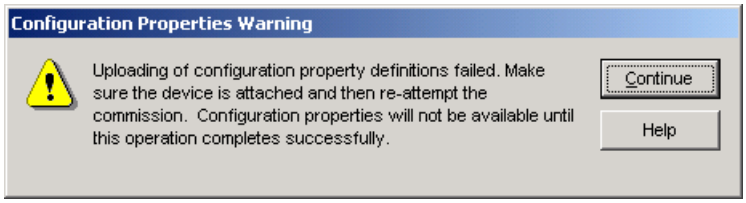


Figure 106 Click on Continue.

Explanation

There are still some dynamically created NVs in L-Proxy. This can happen if the node has been used in a different project and it hasn't been de-commissioned in the old project before it was moved to the new network.

Solution

Click on Yes and Continue. In most cases this will be enough to use L-Proxy. If L-Proxy is behaving strange one should reset the L-Proxy to its default configuration via the console interface (see Section 7.4.2)

9.3 NL220 Warning appears during installation

Problem

When adding a new L-Proxy Port to the project the following warning messages might appear.

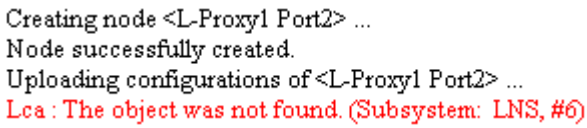


Figure 107 Warning when creating node

Explanation

There are still some dynamically created NVs in L-Proxy. This can happen if the node has been used in a different project and it hasn't been de-commissioned in the old project before it was moved to the new network.

Solution

Ignore the error. In most cases this will be enough to use L-Proxy. If L-Proxy is behaving strange one should reset the L-Proxy to its default configuration via the console interface (see Section 7.4.2)

9.4 Move L-Proxy into a new LNS project

Problem

When moving an L-Proxy device from one LonMaker/NL220/Alex project into a different LonMaker/NL220/Alex project one must delete the L-Proxy port in the old project before adding the L-Proxy device in a different project.

Explanation

If L-Proxy is moved from one LonMaker/NL220/Alex project into a different project LonMaker/NL220 must release all captured resources in L-Proxy. Resources are released when the L-Proxy Port is deleted from the LonMaker/NL220/Alex project.

Solution

Select the L-Proxy Port in the old LonMaker/NL220/Alex project and choose Delete/Remove from the context menue. Make sure that the device can be reached by LonMaker/NL220/Alex during the deletion process. Therefore LonMaker/NL220/Alex must be On-Net. This process must be followed for all L-Proxy ports that have been used.

As a second alternative the L-Proxy can be set into its factory default state through the console interface, see Section 7.4.2.

9.5 Node does not respond to poll request

Problem

The input NV on the L-Proxy is being polled by another node but L-Proxy does not respond to the poll message.

Explanation

If the L-Proxy internal connection consists of only one input NV and one or more output NVs then this input NV cannot be polled by another node. If the internal connection has 2 or more input NVs and the poll is received on the first input NV the poll will be forwarded on the second input NV.

If the internal connections uses active poll (active poll flag enabled) the input NV that outputs the active poll (first input NV if multiple input NVs exist) cannot be polled.

Solution

Always poll the source of the information (output NV on source node) and not the input NV of the L-Proxy.

9.6 L-Proxy doesn’t forward packets after commissioning

Problem

No packets are forwarded by L-Proxy after one or more ports have been re-commissioned.

Explanation

The L-Proxy configuration will be deleted when one or more ports are commissioned.

Solution

Always download the L-Proxy configuration after commissioning one or more L-Proxy ports.

9.7 Technical Support

LOYTEC offers free telephone and e-mail support for our L-Proxy product series. If none of the above descriptions solves your specific problem please contact us at the following address:

LOYTEC electronics GmbH
Stolzenthalergasse 24/3
A-1080 Vienna
Austria / Europe

email: ***support@loytec.com***
web: ***http://www.loytec.com***
tel: ***+43/1/40208050***
fax: ***+43/1/402080599***

10 Application Notes

10.1 The LSD Tool

Please refer to application note “AN002E LSD Tool” for further information about the LOYTEC system diagnostics tool for the L-Proxy.

10.2 L-Proxy Backbone Mode vs. a Standard TP-1250 Backbone

Please refer to application note “AN004E Backbone Mode” for further information on how to best utilize the high-speed backbone mode of the L-Proxy.

11 Firmware Versions

Table 5 shows the most important features available only in certain firmware versions depending on the firmware version.

Firmware Version Supported Features	1.0 Final 1	2.0 0	2.1.0
Console Menu (configuration & firmware update)	√	√	√
Binding across domains	√	√	√
Address table extension	√	√	√
Firewall	√	√	√
Password protection	√	√	√
Statistics information		√	√
SNVT translation		√	√
Real time clock		√	√
TP-1250 backbone mode	√	√	√
RS-485 bit-rate detection		√	√
Active Polls		√	√
LSD 2.0 support			√

Table 5: Available features depending on firmware version.

12 L-Proxy Feature Summary

- Supports binding of network variables across domain boundaries
- Supports the extension of a Neuron Chip address table
- Can be used as a firewall in EIA-709 networks
- Conversion between different scalar SNVT types of same SI type
- Provides current date and time from built-in real-time clock with battery backup
- Supports 384 network variables per port
- Supports 512 alias-network variables per port
- Supports 384 address table entries per port
- Comfortable and easy to use configuration utility
- Perfectly integrated with LNS based network management tools
- Relays packets of up to 256 bytes length
- Supports up to 10 domains (5 when used with LNS based management tools)
- Short propagation delays between ports
- Processes up to 500 packets/sec
- Supports firmware update and external configuration through each channel interface
- Supports multiple transceivers: FT-10/LPT-10, TP-1250, (TP-RS485, PLT-22 optional)
- Bit-rates between 300 bps and 2.5 Mbps
- TP-1250 ports can be used as collision-less, low-latency backbone
- Supply voltage and CPU temperature monitor
- 9-35 V DC / 9-24 V AC supply voltage
- 157 x 86 x 60 (L x W x H in mm) or 9 TE
- DIN-rail (EN 50 022) or wall mountable

13 Specifications

Operating Voltage	9-35 V DC or 9-24 V AC ±10%
Power Consumption	3 W typical
In-rush current	up to 1100 mA @ 24 VAC
Operating Temperature (ambient)	0°C to + 50°C
Storage Temperature	-10°C to +85°C
Humidity (non condensing) operating	10 to 90% RH @ 50°C
Humidity (non condensing) storage	90% RH @ 50°C
Enclosure	Installation enclosure 9 TE, DIN 43 880
Environmental Protection	IP 40 (enclosure); IP 20 (screw terminals)
Installation	DIN rail mounting (EN 50 022) or wall mounting

14 Version History

Date	Version	Author	Description
30-09-02	1.0	DL	Initial revision V1.0
10-21-02		DL	Added Section 9.4
11-07-2002	1.1	DL	Released Version 1.1
12-11-2002	1.2	DL	Added support for NL220 network management tool
02-14-2003	2.0	DL	Add Section 2.7 SNVT Conversion
02-14-2003	2.0	DL	Change screenshots for Plug-in version 2.0
02-14-2003	2.0	DL	Add XIF support in Section 5.3
02-14-2003	2.0	DL	Add Use Case Poll-Update Proxy in Section 6.3
02-14-2003	2.0	DL	in Section 6.4
02-14-2003	2.0	DL	Add Section 5.7 L-Proxy LonMark Objects
02-14-2003	2.0	DL	Change Section 7.4 Console
02-14-2003	2.0	DL	Change Chapter 10 to refer to the application notes
02-14-2003	2.0	DL	Add references in Chapter 15
02-18-2003	2.0	DL	Release Version 2.0 of the user manual
03-27-2003	2.1	DL	Change solution in Section 9.2 and Section 9.3
06-30-2003	2.1	DL	Add Section 9.5 Node does not respond to poll request
06-30-2003	2.1	DL	Add Section 9.6 L-Proxy doesn't forward packets after commissioning
08-14-2003	2.1	DL	Add Section 6.4 Poll Once Initial Value Proxy
01-12-2004	2.2	NR	Add description how to use L-Proxy in Alex

15 References

[1] LonMark Application Layer Interoperability Guidelines, Version 3.3, October 2002

[2] Real-Time Keeper Functional Profile, LonMark, Version 1.0, Profile 3300