Manual

sbc
SAIA BURGESS CONTROLS

LONWORKS®

with
Saia PCD® COSinus Systems

LONMARK®
SCHWEIZ

LON over IP - PCD7.R58x
LON FTT10 - PCD2.F2400 / PCD3.F240

Document 26-883 | Edition ENG02 | 2019-01-16

0 Content

1. Saia PCD overview of solutions

2 Philosophy and components of LON

6 The SBC Lon configurator

7 LON commissioning software

8 Error Handling

A Appendix

0.1 Document versions

Version	Published	Changed in	Remarks
EN01	2013-07-21	-	New document
ENG02	2019-01-16	Chapter A	New phone number (2015)

0.2 About this manual

See the section in the appendix in relation to some of the terms, abbreviations and the references used in this manual.

This manual and the books mentioned in the appendix are not exhaustive for a successful Lon application configuration. They are only intended to impart a basic understanding. The training to become a certified Lon system integrator is offered on a country-specific basis by LonMark® organisations.

Each country has its Lon organisation (LonMark®) for system integrator training courses and certificates.

LonMark International : http://www.lonmark.org
Country-specific e.g. : http://www.lonmark.de

0.3 Brands and trademarks

Saia PCD® and Saia PG5®
are registered trademarks of Saia-Burgess Controls AG.

Technical modifications are based on the current state-of-the-art technology.

Published in Switzerland

1. Saia PCD overview of solutions

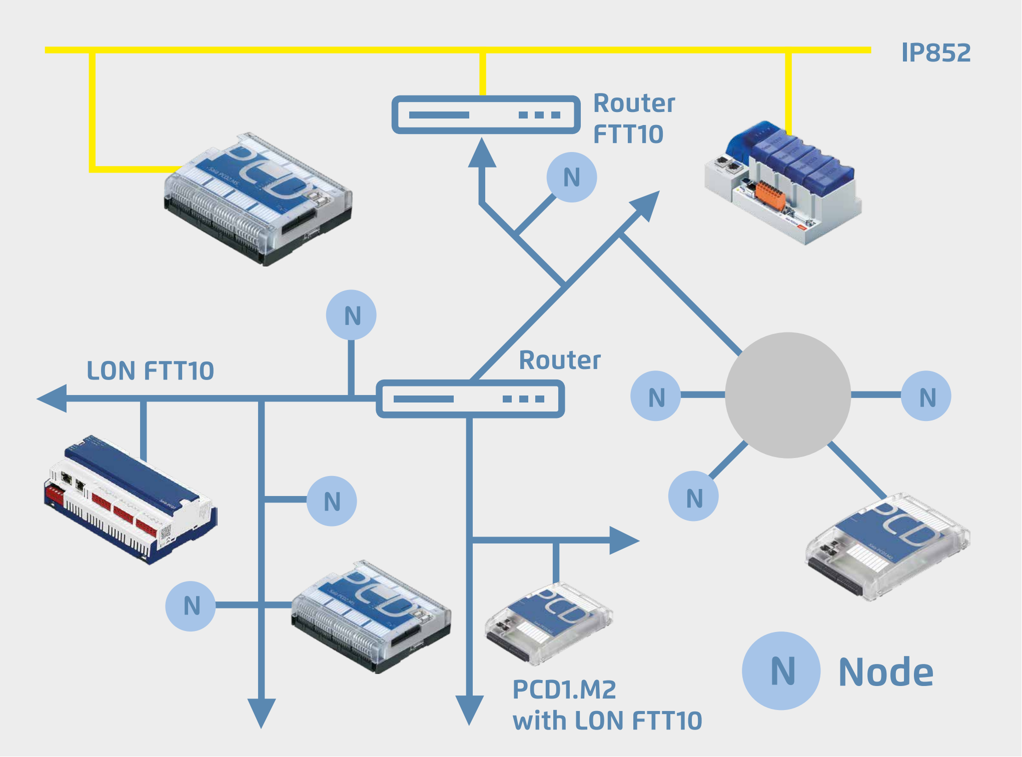


Fig. 1-1 | LonWorks® network

1.1 Concept

The LonWorks® technology is a universal communications protocol that has been established in building and factory automation for years. The various advantages of LonWorks® such as decentralized intelligence, modular structure, interfaces that match requirements and options for adapting to existing infrastructures, all make it an interesting technology for data transfer both in the field and for backbone systems. The individual network users, the so-called nodes, can exchange data among themselves on an event-driven basis. LonWorks® represents the platform for vendor-independent communications within inter-plant building automation.

The Lon over IP host node is based on a modular, freely programmable control and automation system with the state-of-the-art web-IT technology and was developed for the PCD1.M2, PCD3.M3 / .M5, PCD2.M5 series by Saia-Burgess Controls AG. The Lon-Works network driver is a software solution and works as a dedicated task in the SBC COSinus operating system. The driver is supplied on a dedicated flash memory module. The LonWorks® node ID is saved as non-volatile memory on to an EEPROM on the module. The ID is predefined and cannot be modified by the user.

Communication takes place over an Ethernet TCP/IP interface and requires an appropriate router for access to FTT10 or another physical layer. Finally, during engineering one IP852 configuration server is also needed per network.

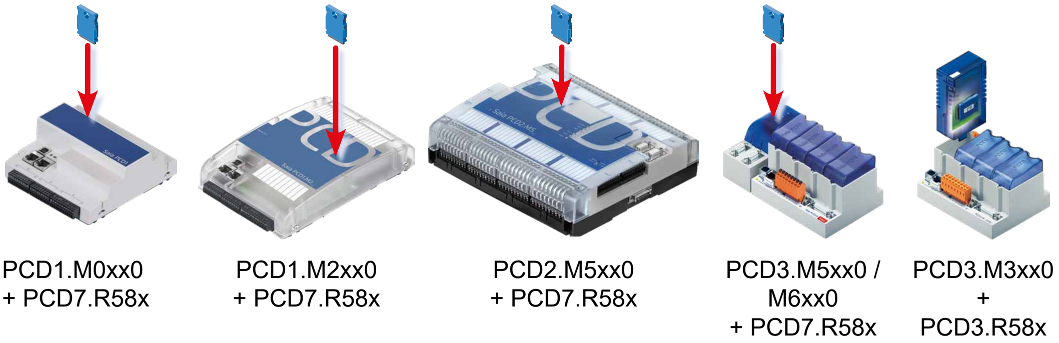
For example

- The PC software “Echelon LonWorks® – IP configuration server” or
- IP 852 for FTT10 routers with integrated "Config servers". (Gesytec or Loytec)

Features:

- Variables supported by LonMark®
- Supported IP-based Lon systems
- Lon over IP configurator integrated into the PG5 for selecting and defining standard network variables (SNVT)

1.1.1 Lon over IP


PCD1.M0xx0
+ PCD7.R58x

PCD1.M2xx0
+ PCD7.R58x

PCD2.M5xx0
+ PCD7.R58x

PCD3.M5xx0 /
M6xx0
+ PCD7.R58x

PCD3.M3xx0
+
PCD3.R58x

Fig. 1-2 | Overview of memory module

Order information

Type	Description
Lon over IP for PCD3.M3xxx | PCD3.M5xxx and PCD3.M6xxx	
PCD3.R580	Flash memory module with Lon over IP firmware for PCD3.M3120 and ..M3330, plugs into I/O slots 0...3
PCD3.R582*	Flash memory module with Lon over IP firmware for PCD3.M3120 and ..M3330, with 128 MByte as backup for user program and 1 MByte with file system, plugs into I/O slots 0...3
Lon over IP for PCD3.M5xxx | PCD3.M6xxx | PCD2.M5xxx | PCD1.M2xxx and PCD1.M0xxx	
PCD7.R580	Flash memory module with Lon over IP firmware for PCD1.M2xxx, PCD1.M0xx, PCD2.M5xxx and PCD3.M5xxx/..M6xxx, plugs into slot M1 or M2
PCD7.R582*	Flash memory module with Lon over IP firmware for PCD1.M2xxx, PCD1.M0xx, PCD2.M5xxx and PCD3.M5xxx/..M6xxx, with 128 MByte as backup for user program and file system, plugs into slot M1 or M2

Tab. 1-1 | Order details

* Lon over IP can only be used on PCD controllers with an Ethernet interface.
It is important to ensure that both IP-UDP ports 1628 and 1629 are reserved for LON. In addition, an IP852 Config Server should be provided for →commissioning and binding.

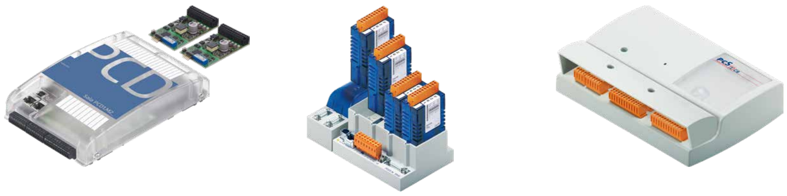
Hardware and firmware versions, which support "Lon over IP" communication:

PCD System	HW from version	FW from version*	PG5 with configurator version from
PCD1.M2120	A	1.16.xx	PG5 2.0.200
PCD2.M5540	A	"	"
PCD3.M3120	E4.8	"	"
PCD3.M3330	E4.8	"	"
PCD3.M5xxx	D	"	"
PCD3.M6xxx	D	"	"

Tab. 1-2 | HW/FW versions

*) The correct firmware is hardware-dependent. Please therefore check the support page «www.sbc-support.com» under the correct SPS-CPU.

1.1.2 Lon FTT10



PCD1.M2xx0
+ PCD2.F2400

PCD3.Mxx60
+ PCD3.F240

PCS1.C88x

Fig. 1-3 | LonFTT10 PCD devices

Type	Description
Lon FTT10 for PCD1.M2xxx | PCD3.M5x6x | PCD3.M6xxx	
PCD2.F2400*	LonWorks® interface module for up to 254 network variables with slot for PCD7.F110S, F121S, F150S, F180S
PCD3.F240*	LonWorks® interface module for up to 254 network variables with slot for PCD7.F110S, F121S, F150S, F180S
Lon FTT10 for PCS1	
PCS1.C88x	Freely programmable compact controllers with integrated LonWorks® interface module

Tab. 1-3 | LonFTT10

1.1.3 Recommendations / System limits

Type	Option	Interface	PG5 configuration, system limits
PCD3.M5560	1× PCD7.R58x* 4× PCD3.F240	IP852 FTT10	Recommended for configurations up to 2000 network variables Suitable for BACnet® and LonWorks® in parallel operation
PCD3.M5540 PCD3.M5340	1× PCD7.R58x*	IP852	Recommended for configurations up to 1500 network variables
PCD3.M3330 PCD3.M3120	1× PCD3.R58x*	IP852	Recommended for configurations up to 1000 network variables
PCD2.M5540	1× PCD7.R58x*	IP852	Recommended for configurations up to 1500 network variables
PCD1.M2xx0	1× PCD7.R58x* 2× PCD2.F2400	IP852 FTT10	Recommended for configurations up to 1000 network variables
PCD1.M0160	1× PCD7.R58x*	IP852	Recommended for configurations up to 1000 network variables
PCD1.M2020 Without Ethernet	2× PCD2.F2400	FTT10	Recommended for configurations up to 500 network variables

Tab. 1-4 | Recommendations, System limits

1.2 LonWorks® XML and XIF files

XIF files

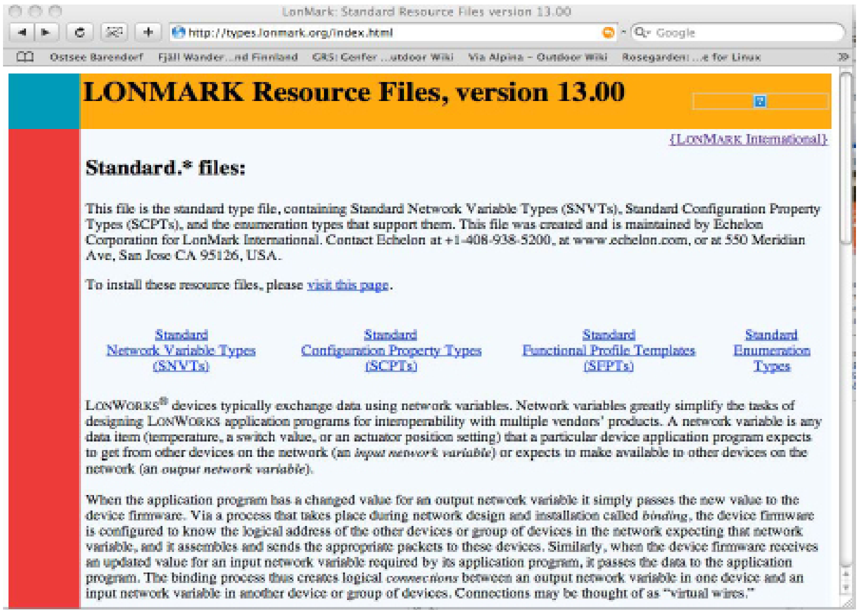
XIF files are external Lon device interface descriptions, which are used by Lon network binding tools (programs).

Resource files describe Lon nodes as a device template in text format. These files contain all standard network variable types (SNVT) and descriptions of the user-defined variable.

XML files

The **new** format for the resource files as from program version 13.00 has the same content in a newly defined **XML-**based text format. That should improve interoperability and use by machines.

You can find all types at http://types.LonMark.org/index.html


LonMark: Standard Resource Files version 13.00
http://types.lonmark.org/index.html
Google
Ostsee Barendorf
Fjäll Wander...nd Finnland
GRS: Genfer ...utdoor Wiki
Via Alpina – Outdoor Wiki
Rosegarden: ...e for Linux

LONMARK Resource Files, version 13.00

{LONMARK International}

Standard.* files:

This file is the standard type file, containing Standard Network Variable Types (SNVTs), Standard Configuration Property Types (SCPTs), and the enumeration types that support them. This file was created and is maintained by Echelon Corporation for LonMark International. Contact Echelon at +1-408-938-5200, at www.echelon.com, or at 550 Meridian Ave, San Jose CA 95126, USA.

To install these resource files, please visit this page.

Standard Network Variable Types (SNVTs)

Standard Configuration Property Types (SCPTs)

Standard Functional Profile Templates (SFPTs)

Standard Enumeration Types

LONWORKS® devices typically exchange data using network variables. Network variables greatly simplify the tasks of designing LONWORKS application programs for interoperability with multiple vendors' products. A network variable is any data item (temperature, a switch value, or an actuator position setting) that a particular device application program expects to get from other devices on the network (an *input network variable*) or expects to make available to other devices on the network (an *output network variable*).

When the application program has a changed value for an output network variable it simply passes the new value to the device firmware. Via a process that takes place during network design and installation called *binding*, the device firmware is configured to know the logical address of the other devices or group of devices in the network expecting that network variable, and it assembles and sends the appropriate packets to these devices. Similarly, when the device firmware receives an updated value for an input network variable required by its application program, it passes the data to the application program. The binding process thus creates logical *connections* between an output network variable in one device and an input network variable in another device or group of devices. Connections may be thought of as "virtual wires."

Fig. 1-4 | LonMark® Resources Files / Internet

The new SBC Lon configurator is needed to generate the Lon node for the Saia PCD3 and PCD2.M5. The configurator is based completely on this new file standard.

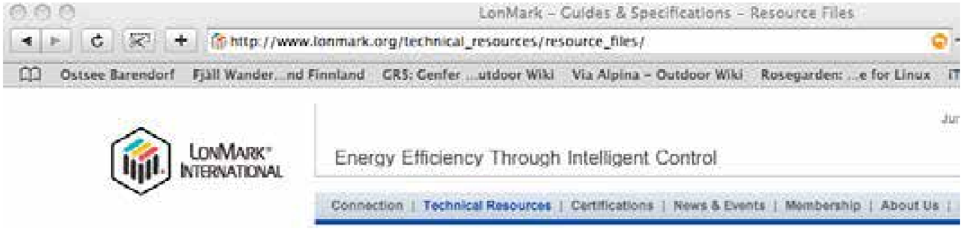

LonMark – Guides & Specifications – Resource Files
http://www.lonmark.org/technical_resources/resource_files/
Ostsee Barendorf
Fjäll Wander...nd Finnland
GRS: Genfer ...utdoor Wiki
Via Alpina – Outdoor Wiki
Rosegarden: ...e for Linux
LONMARK INTERNATIONAL
Energy Efficiency Through Intelligent Control
Connection | Technical Resources | Certifications | News & Events | Membership | About Us |

Fig. 1-5 | LonMark® Resources Files / Internet BBB

The conversion of a "Lon over IP" configuration to XIF is described in chapter 8 "Generating XIF files".

2 Philosophy and components of LON

2.1 The idea behind LON (philosophy)

LON, the Local Operating Network, puts the computer network onto the chip, which is the vision of Echelon, its founder. The aim of the technology is that networks can be built up from a large number of cost-effective so-called nodes. These nodes can be manufactured by different manufacturers and can communicate with one another using the LonTalk protocol.

The nodes all have their own intelligence and are able to exchange data with one another on an event-driven basis. They measure, control, regulate and communicate. This creates an extremely flexible network of functions with virtually any degree of networking and complexity.

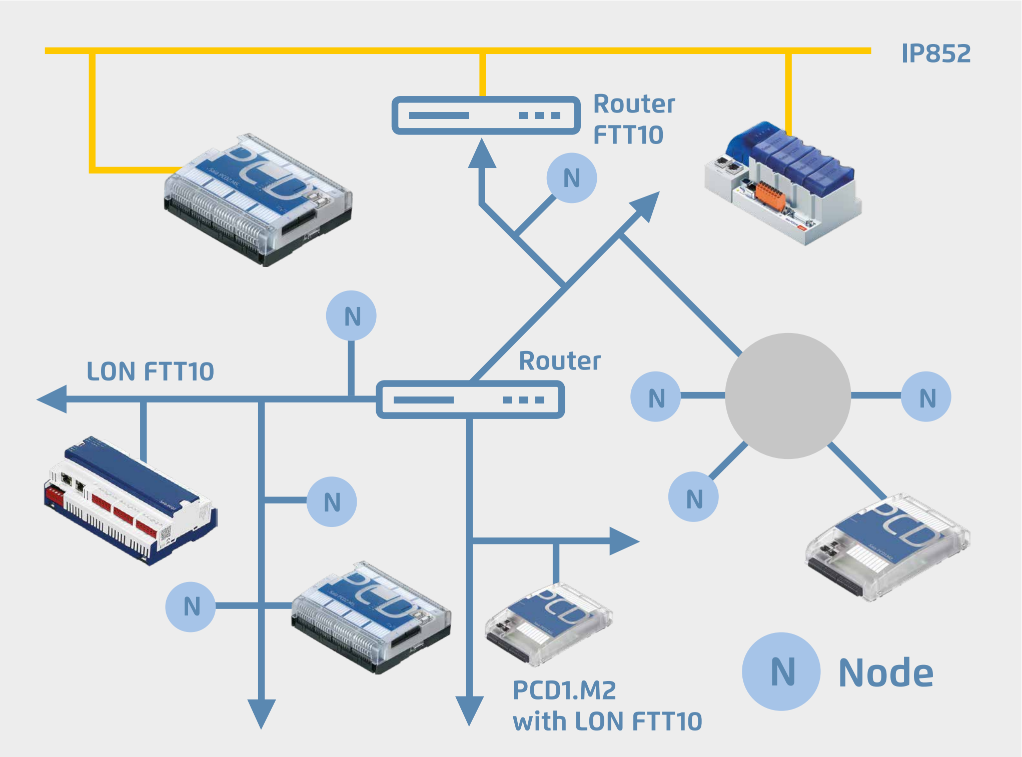


Fig. 2-1: Decentralised nodes

At its core in the form of an open communication language, is the LonTalk protocol, which has been standardised under IEC/ISO, CEN, GBZ and ANSI and can be implemented on any microprocessor. LonWorks® has been accepted in numerous other standardisations. Thus, for example, in BACnet (ASHRAE American Society of Heating and Air-Conditioning Engineers), ISFS (International Forecourt Standard Forum, i.e. all large-scale oil companies), CEN TC-247, SEMI (mass flow rate meters), CELECT (UK for heating systems) and others.

LonMark represents the most important standard, an organisation led by users of Lon components.

LonTalk can be seamlessly transmitted over two-wire lines, 230 V power grids, fibre optic, radio and Ethernet networks.

Its open technology led to it being possible in 1994 to implement the first inter-plant system integrations in buildings with over 3000 devices. Then like today convenience and energy efficiency were of paramount importance.

LonMark has always been the standard, which supports the highest level of modularity and flexibility for complex energy efficiency systems. LonWorks® is the only technology, which has over 30 million powerline devices in use in the field. In terms of new generations of chip LonWorks® is usually that one critical step ahead ensuring that today there are cost-effective solutions, which stand out by comparison.

What, however, distinguishes LonTalk in particular is its sustainability: even today the installations from 1994 are supported with its current tools. The hardware can be loaded with new firmware while the system can be upgraded with the most recent components and with the newest chips.

LonWorks® is across the board today's best choice where sustainability needs need to be reconciled with state-of-the-art technology. And will also be able to offer the same up-to-the-minute benefits in 20 years time.

2.2 The four components of LON

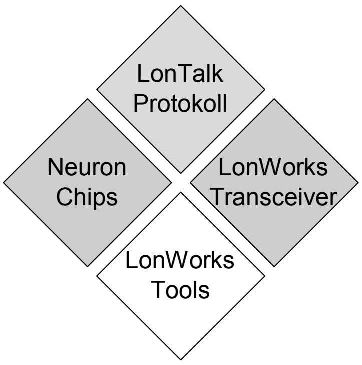


Fig. 2-2 | The four components of LON

In principle, LONWORKS® technology is based on four elements:

The LonTalk protocol defines the language, which is spoken across the medium.

- Microprocessors on switching devices are able to interpret this language and create so-called nodes, which are able to execute networked functions using the LonTalk language.

- LonWorks® transceivers are able to map LonTalk to different physical media so that the language can be transmitted over the most diverse communication channels.

Ultimately, the tools represent the backbone for the development of products, the planning and implementation of installations Accordingly, a distinction is made between development tools (NodeBuilder, Microprocessor Workbench) and installation tools (LonMaker, NL220, NL-Facilities).

2.3 The LonTalk protocol

A Lon-Chip "speaks" LonTalk, that is it sends and receives short telegrams in which the actual usable data is embedded (variable from 0 to 228 byte). So that this takes place efficiently and reliably even when the transmission medium is subject to extreme disturbances, such as for example the 230 V mains grid, reference has been made to best practices from the world of computing and the LonTalk protocol has been provided with a rich array of services based on the 7-layer ISO/ OSI reference model.

2.3.1 Basic structure

2.3.1.1 Transmission modes

Transmission takes place in packets. The compiling and sending of these packets is the responsibility of the firmware; the user does not therefore need to engage with "low-level" functions.

4 different transmission modes are provided in the LON protocol
(so-called services)

unacknowledged	The packet is only sent once. A confirmation is not expected from the recipient.
acknowledged	After sending the packet, a confirmation is expected from the recipient. If this is unsuccessful or turns out to be negative, the packet is sent again. The maximum number of such repeated attempts can be freely specified.
unacknowledged / repeated	The packet is sent on multiple occasions one after another. A confirmation is not expected from the recipient. The number of repeated attempts and the waiting times between these can be freely specified.
request / response	Similar to acknowledged. Other additional data may, however, be available in the confirmation rather than a straightforward acknowledge.

The user can freely determine which mode is to be used.

2.3.1.2 Lon-FTT free topology for 2-wire networks

The data packets are transmitted using a differential Manchester code, i.e. the data information corresponds to a frequency. A period with high frequency corresponds to a 0, and 1 represents a slow period. In this way at least one change of state in the signal is registered per data content. The Manchester decoding makes it possible to run lines without needing to worry about polarity.

2.3.1.3 Lon-IP over Ethernet

The data packets are transmitted from IP networks using the connectionless UDP protocol. Any 2 free UDP ports are required for LonTalk, in which case 1628 and 1629 are recommended as the standard ones.

The Lon files are packaged into the IP telegram as usable data. In this a Lon telegram always consists of the Lon addressing followed by the actual Lon data.

2.3.2 The OSI layer

The definition of OSI (Open System Interconnection) is the basis on which the Internet / Intranet technology has been built. LonWorks® has not re-invented the wheel in terms of structure and has also used the OSI model.

In practice, the larger "overhead" associated with this hardly leads to any noticeable reduction in the transaction or response time behaviour, but makes implementation, commissioning and maintenance of networks a great deal easier. Amongst the aforementioned services, the following should be highlighted:

- efficient access to the transmission medium with priority control (quasi-deterministic behaviour)
- transparent, bidirectional passing and/or filtering of telegrams via integral physical-logical intermediate links (router)
- multiple addressing modes: single node, group, to all (broadcast)
- sending and receiving telegrams with/without acknowledgement, repetition and authentication check
- strategic requesting of data from one or more nodes (request-response, polling)
- event-controlled, prioritised and automated sending and receiving of data via so-called network variables

Use of international standardised values

OSI layer	Meaning	LonTalk Service
7 Application	Compatibility with application level	Object definition Actuator, sensor, controller; standard-type network variables, network management, installation, real-time kernel
6 Presentation	Interpretation	Transport of any telegram frame
5 Session	Action	Request-Response mechanism (polling)
4 Transport	Reliability	Transmission with / without acknowledgement Individual and group addressing Authenticated messages (key, PIN code) duplicate recognition, monitoring sequence
3 Network	Target addressing	Broadcast messages, transparent, configured and self-learning routers, 32385 nodes per domain, 248 domains, 48-bit code in each chip.
2 Link	Media access and frame test	Frame test, data encoding, CRC-16 data security. IP communication or Predictive CSMA, collision avoidance with adaptive allocation of access time slots, optionally with priority time slots and hardware. Collision detection
1 Physical	Electrical connection	Support of various media: RS-485, transformer-coupled 2-wire conductor, radio, IR, LWL, Ethernet, 230 V power grid etc., 610 bit/s □ 1.25 Mbit/s

Table 2-1: the OSI layer model

2.3.3 Address allocation

The LonTalk protocol supports segmentation of a Lon system and the use of different transmission media. The network topology uses the following terms:

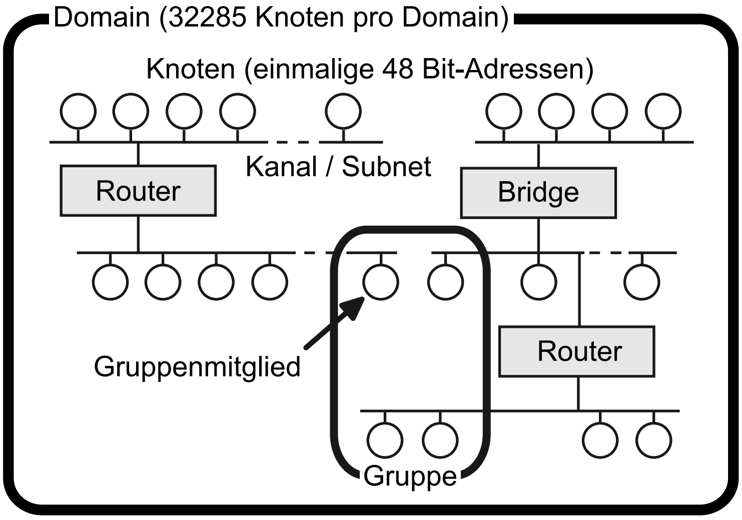

Domain (32285 Knoten pro Domain)
Knoten (einmalige 48 Bit-Adressen)
Kanal / Subnet
Router
Bridge
Gruppenmitglied
Router
Gruppe

Fig. 2-3 | Addressing a Lon system

2.3.3.1 Domain

The domain represents a logical number of nodes on one or more channels. For this data exchange can only take place between nodes within a domain. A domain thus represents a virtual limit of a Lon system.

Different domains can exist alongside one another on a channel. For this they can be used to prevent reciprocal influencing of nodes in different Lon systems on the same channel. If, for example, the nodes are communicating on the network line in a multiple-family dwelling, then the LON systems for two households should use different domain addresses, so that the radio alarm does not also switch on the neighbour's coffee machine as well as its own in the morning. Furthermore, the domain address for the service staff can also be used as a system serial number. A domain can contain 32,512 nodes. A node can, however, be the maximum subscribers in two domains.

A domain can be defined using 0, 1, 3 or 6 bytes. The domain with a length of 0 is used to send the service message, the domain with a length of 1 the the ID 0 is used for development tools and LNS messages. The domain is part of the address in the telegram, i.e. a long domain identification generates more network overhead.

2.3.3.2 Channel

A channel is the physical transmission medium on which serial data is transmitted. It may, for instance, be a cable, a radio frequency or a part of the 230V AC voltage mains supply for power line communication.

A channel is always separated from a second channel by a router or a gateway.

Channels can be freely defined, and so company-specific channels can also be set up.

2.3.3.3 Subnet

A subnet is a logical union of a maximum of 127 nodes within a domain. Within a domain in turn 255 subnets can exist. All nodes in a subnet must be in the same domain.

A channel can in turn control multiple subnets, i.e. subnets are logical addressing groups, which can be used over a variety of physical media. A subnet cannot, however, cross an intelligent router, i.e. channel-crossing subnets must be connected using a bridge or repeater.

2.3.3.4 Node

Each of the 127 LON nodes within a subnet can be addressed via a seven bit long node number. In this way the maximum addressable number per domain of LON nodes comes to 32,385 (127 nodes × 255 subnets).

2.3.3.5 Group

Different LON nodes within a domain can be merged into one group in which case the individual nodes are also allowed to be in different subnets. Based on the 1-byte long group addresses up to 256 groups can be defined within a domain. A Neuron chip can belong to up to 15 groups. In the case of data transfer with confirmation (acknowledged), a group is allowed to incorporated up to 64 nodes. With a telegram without confirmation (unacknowledged), all nodes within a domain can be addressed simultaneously.

Group addressing represents a tried and tested way of reducing the number of telegrams required for broadcast communication (one-to-many). For instance, in a conference hall several lights can be controlled simultaneously with a telegram in this way. As a result, there is no sequenced light effect and the bus is not overloaded with unnecessary data traffic. Thus a group can, for example, contain all light nodes in a factory, although they are controlled via the 220V mains power or via two-wire bus.

With the appropriate installation tools (LNS-based), a group can be divided into multiple subgroups using the so-called "group overloading". These subgroups are automatically created for unacknowledged bindings.

2.3.4 Addressing modes

According to the possible address allocations, different addressing modes can be used. The LonTalk address field in each case describes the sender and the destination address of a LonTalk telegram. Hierarchical addressing is defined in the LonTalk protocol with domain, subnet and node addresses. Domain and group addressing otherwise exists for simultaneous addressing of multiple Lon nodes. A Lon node can therefore be addressed amongst a variety of addresses.

In total there are five addressing modes: The full address field consists of the domain address (0, 1, 3 or 6 bytes), the destination address and the sender address. Depending on the addressing mode, for this the destination address contains the neuron ID (6 bytes), the group address (1 byte) or the subnet and node address (2 bytes together). The sender address always consists of the subnet and node address of the sending node.

A Lon node can always be specifically addressed by means of its neuron ID. Unlike this, the address issued during the installation phase can change during the course of the existence of a node. Due to the length of the neuron ID (6 bytes), it should only be used during installation and configuration of a LON network.
If a node needs to be replaced, then the new node being used is simply given the same address information as the old one. Its communication partners in the network, however, remain unchanged.

A domain is identified by the domain ID (0, 1, 3 or 6 bytes). If the neuron ID for a 6-byte long domain ID of a LON node belonging to the domain is used, the uniqueness of the domain ID is guaranteed.

In a Lon system in which there cannot be any possible overlaps between different areas, it is best to do without the domain ID in favour of a short telegram length.

Depending on the addressing mode, the length of a LonTalk address varies between 3 bytes and 9 bytes. Added to this is the length of the domain ID (0..6 bytes). The address information contained in a LonTalk telegram therefore varies between three bytes for group addressing and fifteen bytes for addressing via the neuron ID with a 6-byte domain address.

2.3.5 Explicit Messages

All Lon telegrams are "explicit messages". They can be compared with a data train, which seeks out its path through the network to the correct destination node. As a guide the locomotive contains the address, which automatically triggers the setting of points in the network. Similar to the internet, data can therefore be transmitted in any form (layer 6). Explicit messages are used by many manufacturers to control their proprietary systems. The recipient's address can either be specified by the programmer or configured in the EPROM.

Advantage:

- more efficient than network variables

Disadvantages:

- without an exact knowledge of the message structure, a connection is not possible (i.e. connecting to nodes of third-party manufacturers is only possible with some difficulty);

- requires larger programming overhead, thus more code.

Lon, however, offers a special "explicit message" on layer 7, which supports direct binding of program variables with the network. The following chapter examines this type of message.

2.3.6 Network variables

Network variables constitute the foundation of an important and in this form unique characteristic of LonWorks, so-called interoperability. This is understood to be the seamless interaction of LonWorks®-based products from different manufacturers operating on the basis of simple rules of play. Because of the different forms of interaction between production and installation engineering by manufacturers, system planners and installation firms, interoperability is an important prerequisite for the distribution of LonWorks® within the industry and in buildings automation. It could also be expressed as follows: LonWorks® allows you to build complex systems as if they were from a single source.

Communication principle:

Network variables (NV)
Variables, which establish bindings between two or more nodes. The binding of variables is optional when programming the application, in the case of the final test on the device, on site during installation or while operating the network.

SNVT / SCPTS
To create bindings between nodes of different manufacturers, so-called standard network variables (SWT) and standard configuration data is used (SCPTS).

SNVTs can be "bound", i.e. based on an entry in the local memory an SNVT thus knows which nodes are expecting data from it. This data is always transmitted in sequence, when its value changes.

To supplement this there are so-called UNVTS / UCPTS (user-defined data types), which defines the format files also provided by the manufacturer.

2.3.7 Configuration and network management

Logically using network variables, a wide range of communication connections can be established (so-called bindings) between the individual nodes. This is generally implemented using an installation tool (hand-held device, PC running DOS and Windows) in the field in which case corresponding entries are made in the EEPROM of the individual nodes. There are also instances, however, such as for instance in a machine controller in which all nodes have already been predefined with all communication relationships.

Multiple scenarios are available for commissioning an Lon system. Depending on the state of the LON nodes being installed, the communication relationships and the application program have to be transmitted to the nodes.

Simplest variant

The plug-and-play installation of nodes preconfigured by the user represents the simplest variant for small-scale systems.

Auxiliary devices

Larger systems are operated with the help of a network management node (NMK, hand-held device or PC). An NMK is able to search a Lon system for newly added nodes and configure them. It can load an application program on to the node, and start, stop and reset it.

Otherwise it is able to read out the communication statistics from the managed node, configure the router and define the structure of a running LON system. During installation an assignment must be made between the physical position of each LON node. To do this using the WINK command the installer is able to invite a node to execute a special function (e.g. light 1 flashes once) to identify or find it. As a result together with the NMK it creates the logical bindings with other nodes.

LNS, the LonWorks® Network Server represents the most popular method offering indisputably the best interoperability support.

Creating a list

Another scenario involves creating a list of the neuron IDs and of the physical positions (and therefore functions) of the LON nodes. The NMK then allocates to the nodes the desired communication relationships and, where applicable, provides them with any application program that may still be missing. To simplify installation, LonTalk provides a node identification string eight bytes in length.

2.4 Node

2.4.1 The Neuron chip family from Echelon

2.4.1.1 Single Chip Processor 3120

The Single Chip 3120 is used for LowCost modules with limited functionality as its data memory is very limited. Programs can be loaded into the EEPROM via the bus.

CPUs	3
EEPROM bytes	512
RAM bytes	1024
ROM bytes (firmware)	10240
External Memory Interface	no
16-bit Timer / Counter	2
Watchdog Timer	yes
Package	SOIC
Pins	32
Network variables	62
Address tables	15

2.4.1.2 Multiple Chip Processor 3150

The 3150 supports controlling an external databus and is therefore suitable for complicated tasks. The 3150 is comparable with a 68HC11 or 80C535 in terms of its processing capacity available for the application.

CPUs	3
EEPROM bytes	512
RAM bytes	2048
ROM bytes (firmware)	0
External Memory Interface	yes
16-bit Timer / Counter	2
Watchdog Timer	yes
Package	PQFP
Pins	64
Network variables	62
Address tables	15

2.4.1.3 Smart Transceiver Chip

The smart transceiver chips are Neuron chips with an embedded signal processor for the FTT or Powerline Transceiver. Smart transceivers exists for all Neuron derivatives.

2.4.1.4 Neuron 5000 Chip

CPUs	4 (Internet)
EEPROM/Flash	external
RAM	64 kB
ROM	16 kB
Watch Dog	yes
Package	7x7 mm QFN
Pins	48
Network variables	254
Supply voltage	3.3 V

2.4.2 MIP (Micro Processor Interface Program)

So that LonTalk can be reproduced on powerful processors, a parallel interface to other processor systems has also been implemented on the NEURON chip. The interface is controlled by means of a link layer and an application message layer protocol and supports full access to the LonTalk protocol by the coupled microprocessor. MIP nodes are no longer limited in terms of processing capacity. A MIP is able to process 4096 selector entries, but the limitation continues to be maintained in terms of the 15 address tables and 2 domain tables (legacy mode).

A MIP-based node does not essentially behave any differently for the system integrator. It only offers more variables and better performance.

2.4.3 Shortstack

Similar to the MIP, the shortstack is special firmware for Neuron chips, which is however connected to a microprocessor via the SCI or SPI interface.

A shortstack node is able to process up to 255 variables / selectors. The limitation to "legacy mode" remains in place.

2.4.4 Open protocol implementations

Stacks from multiple manufacturers are available for LonTalk implementations on powerful processors. Generally these stacks must be built on real-time operating systems, which support timers to millisecond resolutions. The best know suppliers are Loytec, Adept and Echelon.

Such stacks also support connecting field units, directly via Ethernet (e.g. L-Vis from Loytech, PCD from Saia-Burgess Controls AG, InfraDALI from Infranet Partners, i.Lon from Echelon).

2.4.5 Layer 2-MIP by Echelon

The layer 2 MIP is a special firmware, which enables all chips to implement LonTalk on any microprocessors using the Echelon stack. The connection with the respective physical layer is established via SPI.

2.4.6 Terms for all Lon chips

2.4.6.1 Firmware, EEPROM, PROM, Flash PROM, RAM

Firmware	Firmware means the program running in the chip.
EEPROM	Electronically deletable memory space, which can also contain firmware to a limited extent. Generally EEPROM is used for saving configuration data. An EEPROM can be loaded over the network.
FLASH EPROM	A FLASH EPROM can be deleted by means of a UV flash light built into the chip and can be re-programmed several thousands of times. A flash can be loaded over the network and supports functional modifications in devices that have already been installed.
RAM	RAM is volatile memory, which can either be temporarily stored by means of battery or loses its contents after being switched off.

2.4.6.2 Service

The so-called service pin is a special connection of the Neuron chip. It serves as being a mechanical tool for configuration, commissioning and maintenance of the network node, to which the Neuron chip belongs. If a button is connected and thus the service pin is set to earth, it (or rather the Neuron firmware) sends a special network management telegram in which it communicates its unique 48-bit serial number (Neuron chip ID), amongst other things, to all nodes in the network. This information can be used by a network administrator for issuing the logical network address of the node during installation and for the ensuing configuration.

If the service pin is connected with a light emitting diode (LED), it can signal the current operating state of the network node by means of various flashing sequences.

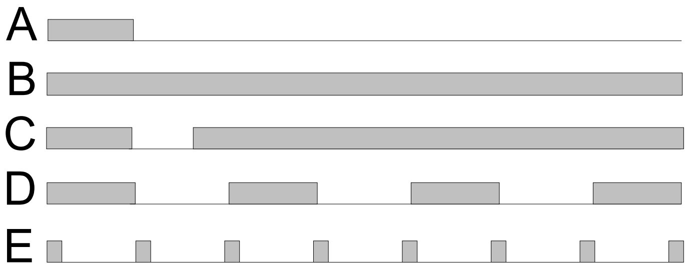


Diagram 2-1 | Flashing sequence of the service LED

The meaning of the LED indicator is as follows:

A) NORMAL OPERATION When starting, the diode briefly lights up (<1 sec) and then goes out for ever. The NEURON chip is configured and is working correctly.

B) FATAL ERROR The NEURON chip could not start (clock, CPU bus, reset or firmware problem). Generally the printed circuit board or its components have been damaged.

C) APPLICATIONLESS In the "applicationless" state the NEURON chip was able to start, but has not found an application matching the hardware. Where this is the case, new firmware needs to be loaded. Upon starting, the LED first exhibits "normal operation" to then continuously switch on the LED after 3 seconds.

D) UNCONFIGURED In the case of an unconfigured node, the LED flashes at a frequency of 1 Hz. The hardware works correctly, although the user program has not yet started. The node now needs to be configured (assignment of a logical address) to be transferred into "normal operation mode".

E) WATCHDOGING The internal watchdog of the NEURON chip restarts the chip every 750 ms, which is displayed by a brief flashing of the LED. The node would actually like to start normally, but is encountering a runtime error. Causes of the error can be non-functioning parallel port or unsynchronised bit-serial interfaces.

The firmware of the chip is in each case started upon activating the service pin irrespective of whether the node is already supporting a user program or the network configuration has already been completed.

The service pin is subject to control by the software (firmware) if it is connected with an I/O pin. The main program of the network processor regularly polls the service pin for each telegram sent or received. It is also possible to access the service pin from the user program. Certain differences should, however, be observed by the programmer when writing the application program in terms of the logical classification of the service pin, which depend on the processor type and the firmware version.

2.4.6.3 Configurability with Legacy nodes

Legacy nodes have a data structure, which supports binding to their network partners. This data structure is generally managed by an installation tool, which assumes control over system functions. Two domain tables are used for saving domain affiliation. Furthermore, 64 selectors can be registered for network variables, which support registering bindings. So that the node knows where it can send outgoing data to, 14 address tables are available to it.

If an output variable contains a new value, the program looks in the **nv_tab** table to see which selector has been registered and with which address table it needs to work. The address table in turn contains the information on which domain is to be used. The address of the telegram is composed in this way. A Legacy chip can therefore address up to a maximum of 15 other nodes directly.

If group addresses are used, up to a maximum of 15 groups can be serviced in which case incoming group messages need to be registered in the address table. The group table can, however, use multiple selectors so that a node can be bound to more than 15 recipients.

2.4.6.4 ECS nodes

- The ECS (Enhanced Command Set) nodes support a larger number of address tables and additional more flexible configurations.
- ECS nodes use additional network commands.
- ECS nodes can be integrated into networks using LNS tools without any compatibility problems.

2.4.6.5 Alias tables

Alias tables support a more flexible integration of devices in complex network structures. These so-called "alias bindings" are managed by LNS tools automatically.

2.5 LonWorks® Transceivers

Transceivers represent the great advantage of LonWorks technology. Using these components it is possible for manufacturers to be able to efficiently access a wide variety of different media. Using the different transceiver technologies, corresponding bus topologies can be created.

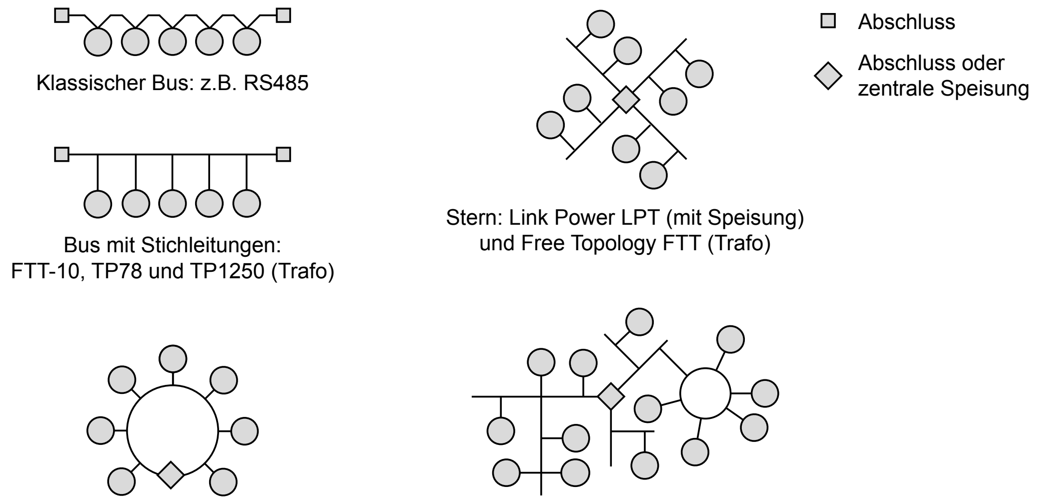

Klassischer Bus: z.B. RS485
Bus mit Stichleitungen:
FTT-10, TP78 und TP1250 (Trafo)
Stern: Link Power LPT (mit Speisung)
und Free Topology FTT (Trafo)
□ Abschluss
◇ Abschluss oder zentrale Speisung

Fig. 2-4 | Possible bus topologies

2.5.1 Twisted Pair TP 78

For conventional bus topology it is possible to work with the twisted pair transceiver for 78.1 kbit/s or 1.25 Mbit/s. The bus separated by means of a transformer guarantees a high level of interference immunity.

TP-78	
Path	1400 m, terminated at both ends
nodes per channel	64
Stub	Spur up to a maximum of 3 m
Specifically	in the case of minus temperatures only 44 nodes per channel
Zero voltage range	+220 V…220 V rms

2.5.2 Free Topology FTT-10

The FTT-10 is undoubtedly the most popular transceiver, which has proven itself to be the standard. Managing a field bus in wild topology is currently a technological peak achievement, as it has always been. Particularly outstanding is the simple integration of these components in products for which the guidelines relating to design practically guarantee successful CE certification.

FTT-10	
Path	2700 m, terminated at both ends and in bus topology 400 m in free topology and terminated at one end.
nodes per channel	64
Zero voltage range	+220 V…220 V rms

2.5.3 RS-485

The RS-485 is still the cheapest solution, although (depending on the type of specification) it only offers a zero voltage range from –7…+12 V. Is suitable in particular for smaller installations.

Type	Medium	kbit/s	Length / Topology / Note	No. node
TP- RS-485	Twisted pair line.	39 to 625	1200 m at 39 kbit/s, bus, with or without electrical isolation	32 per bus segment
TPT/XF 78 transformer	Twisted pair line.	78	1400 m, bus with 3 m spurs, isolation 277 V RMS	64 per bus segment
TPT/XF1250 transformer	Twisted pair line.	1250	130 m, bus with 0.3 m spurs, isolation 277 V RMS	64 per bus segment
FTT10 transformer	Twisted pair line.	78	2700 m as bus, 500 m for free topology, isolation 277 V RMS	64 per bus segment
LPT10 Link Power	Twisted pair line.	78	500 m, free topology, 42 V DC, 5 V / 100 mA per node	32**…**128 per bus segment
PLT20 Power Line	230 VAC or DC	4.8	50 m**…**5 km, BPSK Modulation Cenelec Band C, 132.5 kHz	depending on mains power
PLT30 Power Line	230 VAC or DC	2	50 m**…**5 km, Spread Spectrum Cenelec Band A, 9…95 kHz	depending on mains power
IP-852	Tunnelling via IP		All IP channels	

2.5.4 Link Power

When using Link-Power transceivers, data and power supply energy (48 V) flow together and protected against polarity reversal over a twisted pair line. A switched-mode power supply unit integrated in the transceiver is able to supply the Lon node including application circuitry with up to 100 mA at +5 V. To do this a central power supply unit feeds a bus segment up to 320 m in length. The bus length can be extended by binding multiple link-power segments. When laying the bus line, the installer does not have to pay attention to any maximum lengths of bus junctions or other topological limitations, as the LPT-10 transceiver supports a free selection of topology (star, ring, multi-drop). The same concept was also the triggering factor for the development of the FTT-10. Unlike the LPT-10, each Lon node has its own power supply. Both variants can also be mixed.

2.5.5 Power Line

Generations of development engineers have engaged with the subject of "data transfer over power lines". The power line medium has an enormous advantage: It is already present in residential buildings as in purpose-built buildings thus doing away with the need to rip open walls to lay bus lines.

At the same time, the power line intended for transferring power has an equally significant disadvantage as a medium for data transfer: The line characteristic is different from one place to another and can also change, depending on the type and number of connected consumers, from one moment to the next.

Switched-mode power supplies, electric motors or dimmers are widely used sources of interference in this, which partly corrupt the data signals modulated to the power line until they are unrecognisable. Thanks to full utilisation of the available transmission bandwidth, based on the selection of suitable modulation modes and with appropriate signal filtering the power line can still be made usable for transmitting information. LonWorks® offers three power line transceiver modules for this purpose.

The frequency bands approved by the respective authorities for data transmission on power lines are different in North America, Japan and Europe. In America and Japan the frequency range from 0…500 kHz is released for this purpose. This large bandwidth supports the use of spread spectrum modulation. With it information is transmitted broadband in a large frequency range. Interference, which is limited in many different ways in its bandwidth, cannot therefore affect data transmission throughout the entire frequency band. The power line transceiver PLT-10 only authorised for use in the USA operates in this mode within the range from 100 kHz…450 kHz and in so doing achieves a net data rate of 10 kbit/s.

In Europe the CENELEC (Comité Européen de Normalisation Electrotechnique; European Committee for Electrotechnical Standardisation) only has the frequency range up to 150 kHz (start of long-wave radio) has been released for communicating on the power line. This range is also subdivided into different bands. The CENELEC-A band (9 kHz…95 kHz) is reserved for data exchange of grid operators (electricity companies and distributors). CENELEC-B band (95 kHz…125 kHz) is used for communication without access protocol for end customer applications. In the CENELEC-C band (125 kHz…140 kHz) protocol-controlled data communication takes place for customer applications. The A-Band transceiver PLT-30 also uses spread-spectrum mode and thus achieves a data rate of 2 kbit/s in this frequency band. The narrow C-band requires a different modulation mode. In the case of the PLT-20, BPSK (Binary Phase Shift Keying) is used. This transceiver thus achieves a data rate of 4 kbit/s.

Echelon provides the Power Line Communications Analyzer (PCLA) for analysing available low voltage networks (230 V) for their suitability for use as a data communication medium. This device supports a range of tests, which in addition to telegram error rate also provide information about the analogue transmission parameters (attenuation, interference and signal distortion) of the power line. in addition, there is a PC-based test kit (PLE-30), which can be used to establish a communication connection between two or more PCs so that the sending and receiving of telegrams can be tested using variable transmission parameters.

2.5.6 Other transceivers

In addition, the following transceivers are available on the market:
- fail-safe transceiver 78 kbit/s
- radio 432 MHz
- optical fibre
- Infra-red
- coax
- Tf-conductor
- microwave

2.6 LonWorks® Tools

The fourth element, LonWorks® Tools, include development and installation tools. They are used to develop nodes or plan and carry out installations.

Within the framework of this introduction only a list of the most popular tools has been included, as tools will be handled as part of a developer course or system integration course. Other tools, which are particularly important for developers, are development tools for Neuron-C and ones for host applications. It is possible to create systems in such a way that using field compilers each one supports nodes with the associated source code software and can be extended over the network with new programs. This capability is unique for field bus systems, but is generally only made available on request (disclosure of the firmware source code). At "Runtime-Library" level transparent software maintenance is completely standard on all nodes.

Installation tools:
- LonMaker
- NL-220
- NL-Facilities

All popular tools build on the standards for Windows workstations and support an object-oriented structure (Active-X OXC components) of control software and their node-specific functions. In selecting an installation tool it is important to remember that so-called "device plug-ins" are available for the selected hardware. Such a plug-in provides the system integrator with a graphical user interface for simple configuration of the node, which is incorporated within the installation tool. By double-clicking on the node image, the corresponding plug-in window is opened. Tools are generally marketed so that a fee applies per installed node. In this way the tools are available for smaller installations within a contractually agreed pricing framework. The expense of a system configuration in terms of planning and time is widely underestimated. Whereas in the case of conventional installations individual data points had to be connected by means of cables, the binding in the case of LonWorks® is established using the tool. The expense in processing the information remains the same. At first sight, however, it is not evident in the same way how this is the case for folders filled with electric circuit diagrams.

3 The building blocks of the network

3.1 Nodes

The nodes have been examined in chapter 2.4. Reference is made in this chapter to the information needed by the system integrator to document the system integrator's point of view. The system integrator needs at least the following information for his nodes:

- a good and complete functional specification
- a so-called XIF file, which describes the network interface and/or resource file version 13.
- the description of the electric interface
- where applicable, possible configuration specifications
- where applicable, possible program adaptation and firmware versions

3.2 Building blocks for organising the network

Different channels are logically bound with one another via the router in which case the two bus interfaces of the router can be different or identical in their physical nature. A radio channel with a two-wire line is connected in this way, for example.

Routers consist of two coupled NEURON chips, which exchange telegrams on layer 6 and map them to their respective counterpart. The router algorithms are specified by Echelon and are equivalent for all products.

Facilities for interfacing with a variety of different routing methods (router algorithms) fall under the generic term, router:

3.2.1 Repeaters

A repeat represents the simplest router. It forwards all telegrams from one channel to another. In addition to converting between different transmission media, a repeater can also be used for analogue signal regeneration (amplification) and thus to extend the bus.

3.2.2 Bridges

The next layer in the router hierarchy is the bridge. A bridge is a router with local intelligence. The bridge only routes telegrams within the same domain in which two domains can be transferred.

3.2.3 Learning Router

Learning routers observe the data traffic on the two connected parts of the network and from this make the structure of the network accessible at domain and subnet level. The learning router then uses this intelligence to select the telegrams, which it forwards from one channel to the other. As a learning router is not able to access existing group topologies, all telegrams are always forwarded with group addresses.

3.2.4 Configured Routers

Configured routers, on the other hand, only convert selected telegrams registered in a routing table between channels. The routing table is created using a network management tool. As this tool also manages by issuing group addresses, a configured router can also be programmed for the selective routing of group telegrams.

3.2.5 Why use a router?

Configured routers and learning routers belong to the class of intelligent routers. These are not only a way of connecting physically different transmission media. Thanks to their programming, they can also be used as a telegram filter between physically equivalent channels by only forwarding selected telegrams on to other areas, thereby limiting the telegram acceptance traffic in the local area. The rest of the Lon system continues to be spared from the data traffic that is not relevant for it.

3.3 System limits and tips for overcoming them

3.3.1 Domain limits

The addressing space on the LON bus is split into different hierarchies.

Hierarchies	
top level	The so-called domains form the top level. The different domains are distinguished from one another by means of a 0-, 1-, 3- or 6-byte long identifier, depending on the number of them.
second-highest level	The subnets form the second-highest level. Up to a maximum of 255 subnets can be defined per domain.
third level	The third level is finally formed by the individual nodes. Up to a maximum of 127 nodes can be defined per subnet. On this basis a maximum number of 32,385 nodes per domain is possible.

If the number of domain nodes is exceeded, a second domain can be created and integrated by means of a gateway.

The maximum number of nodes in a domain, however, is not generally the system-limiting factors.

3.3.2 Limited number of groups

Working from this basic setting a large number of grouping possibilities is opened up. Thus, for example, a node can simultaneously belong to two different domains. What is more, different nodes can be defined as groups. Groups have the advantage that the addressing overhead is significantly smaller when sending messages. Such groups can extend over different subnets. Up to a maximum of 256 groups can be defined per domain. In acknowledge mode, an individual group can incorporate up to a maximum of 64 nodes, in unacknowledged mode the number of nodes per group is unlimited. An individual node can belong to up to 15 groups.

The number of groups of 256 is, however, an all-critical limitation meaning 32,385 possible nodes. This limitation is overcome by working with subnet broadcasts at zone level. In addition all non-acknowledged bindings can use addresses on multiple occasions as they are distinguished by the selector (number of the binding). This feature is also referred to as ″overloading″ of a group or subnet address and is automatically used by LNS Tools.
Using overloading a group (or a subnet) is divided into multiple subgroups, which work with the same multicast address but have registered different selectors. In this way the disadvantage of address tables and the group limitation can be obviated whilst maintaining full transparency of the system.

3.3.3 Limited number of channel subscribers

The number of channel subscribers is transceiver-dependent. If the number of permitted nodes (in most cases 64) is reached, another channel can be limited with a router. Subsequent integration of routers into an existing network does not pose a problem if the system is built using so-called “configured routers”.

It is, however, advisable not to fully utilise channels to ensure that a system can be subsequently upgraded according to requirements.

3.3.4 Limited number of address tables

The limitation to 15 address tables, which can only be exceeded for the ECS (Enhanced Command Set in accordance with ISO/IEC 14908-1) nodes, can present a problem in the case of centralised nodes. The 15 address tables mean that a node can only be a member of 15 groups or target addresses.

The LonIP solution, which should be used for centralised nodes, does not impose a limitation in this respect as ECS nodes.

4 The LonMark® standard

4.1 Definition of LonMark® resources

The Lon configurator generates the definition for the mapping of profiles, network variables and configuration parameters in PG5 registers and flags.

The LonMark resource files and their XML report files can be used as raw data. These XML files (fps.xml, nvs.xml and cps.xml) are available for all standard formats in the directory

C:\Documents and Settings\All Users\Saia Burgess\PG5_20\Projects\my Project\Device1\LON\0000000000000000-0.

The configurator creates a ".SY5" and a ".LIP" or ".LFT" file with the mapping of all defined LonMark interface data.

4.1.1 Generating data for LonIP

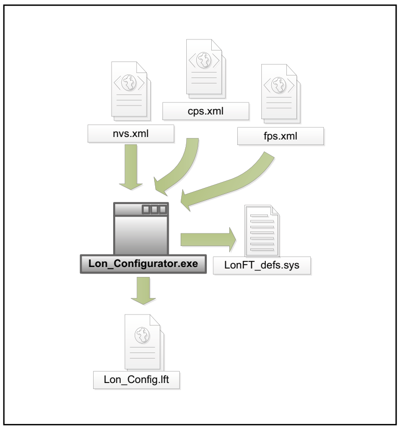


Fig. 4-1 | System overview of the configurator for LonIP operating mode

The configurator is used by the PG5 programmer to define the interface for his/her target hardware (the so-called "LonMark Network Image").
An ".SY5" mapping is created in register and flags as the output as well as the ".LIP" file, which defines the interface in XML. The Lon compiler then generates the binary file ".5lp" for the program download.

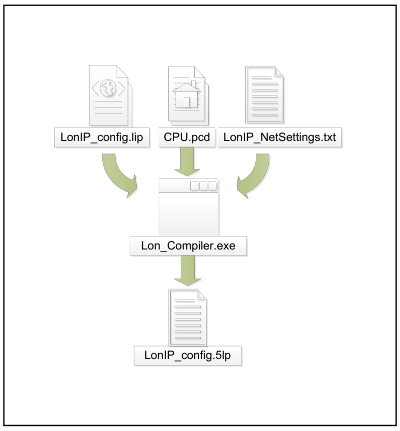


Fig. 4-2 | System overview of the compiler for Lon 3.1.2 operating mode

4.1.2 Generating data for LonFT

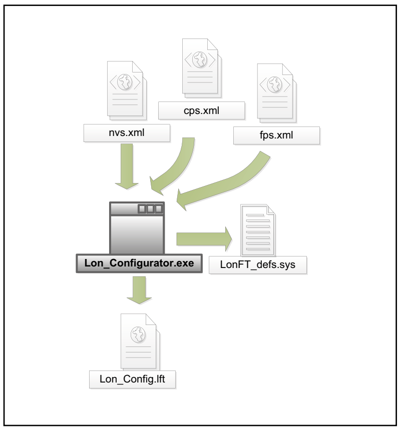


Fig. 4-3 | System overview of the configurator for LonFT operating mode

The configurator generates an ".Sy5" mapping as the output as well as the ".LFT" file, which defines the interface in XML. The Lon compiler then generates the binary file ".5lf" for the program download and a ".XIF" interface definition file for the Lon integration tool.

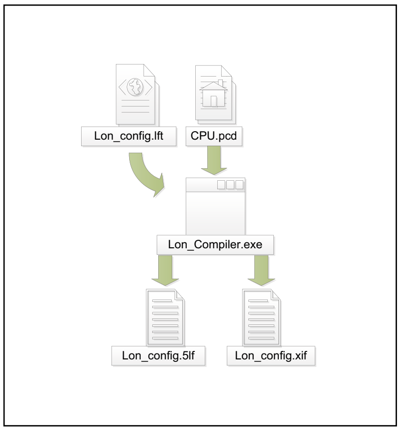


Fig. 4-4 | System overview of the compiler for LonFT operating mode

4.1.3 Predefined profiles in PG5

4.1.3.1 The node object

The node object can be edited in PG5 as it is linked with operating system functions. When opening the nodes, the configurator generates a default XML definition in the definition window.
The variables *nviTimeSet* and *nvoAlarm2* are supported by a csf function.
The other variables are system variables.

4.1.3.2 Type definition for manufacturer's data

LonMark defines standard program IDs, which are assigned to a manufacturer. This standard program ID (SPID) is an 8byte number, and the manufacturer ID is assigned by LonMark. This ID is used to give a unique number to the LonMark network interface.

The 16 hex digits of the SPID are structured as follows in 6 fields: Format (F), Manufacturer (H), Device Class (K), Type of Application (A), Channel Type (T) and Model Number (N) of the device. These 6 fields are organised as follows:

FH:HH:HH:KK:KK:AA:TT:NN

The detailed meaning of the fields is available at www.lonmark.org/spid.

The format definitions can be used with different range of validity ("scope"). These ranges are as follows:

Scope 0:	*FH:HH:HH:KK:KK:AA:TT:NN*	Generally valid, part of the PG5 set-up
Scope 3:	*FH:HH:HH:KK:KK:AA:TT:NN*	Valid for a manufacturer's range
Scope 4:	*FH:HH:HH:KK:KK:AA:TT:NN*	Valid for a manufacturer's application class
Scope 5:	*FH:HH:HH:KK:KK:AA:TT:NN*	Valid for a manufacturer's application class and type of application
Scope 6:	*FH:HH:HH:KK:KK:AA:TT:NN*	Specific for an individual device

If the programmer would like to support manufacturer formats, he can convert them from the format files into XML definitions.

LonMark describes the application layer data formats in resource files and the external interface files (XIF). The manufacturer, which generates these files, generally makes them available. These resource files can be converted using the tools set out below into an XML format for the NV, CP profiles and listings.

The standard XML files (fpt.xml, nvt.xml, cpt.xml Scope 0) are part of the SAIA set-up.

The manufacturer-specific definitions are generated from the format files with the **NodeBuilder Resource Report Generator**
(can be loaded for LonMark members from
http://www.lonmark.org/technical_resources/resource_files/) or the
NodeBuilder Resource Editor

The LONMARK-type definitions can be accessed via http://types.lonmark.org/.

4.2 Restrictions

ISI profiles

"Interoperable Self Installation" is not supported.

Profile inheritance

Inheriting profile formats is not supported. You can create a copy of the profile and use it as an array.

Changeable NV

Variable types that can be modified in real time are not supported.

SCPTnvType

As a result of the aforementioned limitation, the SCPTnvType mechanisms are not supported either.

Unsupported explicit messages

Explicit messages are not supported.

Unsupported self-installation

Self-installation (issuing the LonTalk address from PG5) is not supported.

5 Practical tips

LonWorks® supports a wide range of transmission media such as twisted pair, RS-485, Powerline etc. Depending on this different topologies are possible, such as start, bus or free topologies, however not closed loops. The standardised transmission speed ranges from 300 bit/s to 1.25 Mbit/s. In buildings automation FTT10 at 78 kbit/s is most widely used.
It is possible to combine different topologies and transmission speeds. To do so Lon repeaters, bridges or self-learning routers are used.

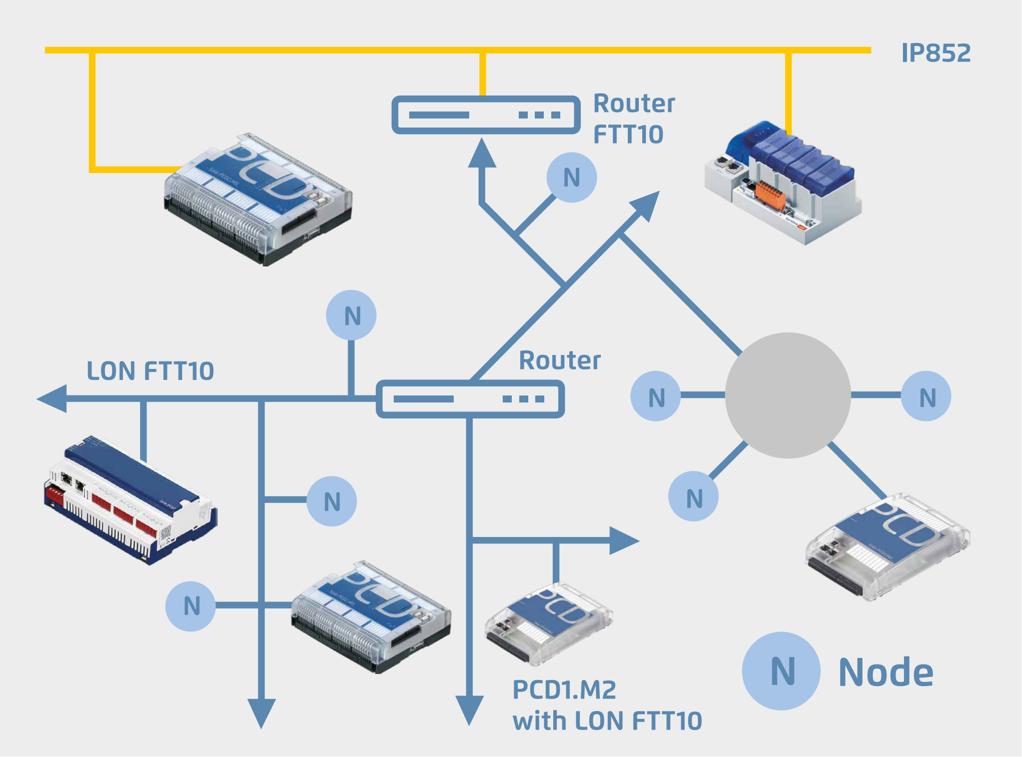


Fig. 5-1 | LonWorks® Network

Wiring	Transceiver	Transmission speed		Topology	max. length	Voltage supply
Twisted Pair	TP/XF-78	78 kbit/s		Line	1400	Separate
	TP/XF-1250	1250 kbit/s		Line	130	
	FTT10-A	78 kbit/s	Can be combined	Line	2700	Separate
				Free	500	
	LPT-10	78 kbit/s		Line	2700	Link Power
				Free	500	

Tab 5-1 | Twisted Pair

The power supply should be supplied according to the type of transceiver.
Two different types of incoming supply are available.

FTT (Free Topology Transceive) Lon nodes wit this transceiver receive their voltage separately, generally 24 V or 230 V.

LPT (Link-Power-Transceiver) This type of Lon nodes obtain their power supply via the bus cabling. In this case a voltage of 42 VDC overlies the bus signal.

5.1 Topology

5.1.1 Free topology

Depending on the conditions in a building it may be necessary to build a free topology. Limitations are imposed as a result of doing so in terms of cabling. According to the different types of cable, different distances between individual Lon nodes are possible. See the table below

Cable type	max. distance between 2 nodes	max. length of the segment
Cat 5	250	450
JY(st)Y 2×2×0.8	320	500
Ul level IV, 22 AwG	400	500
Belden 8471	400	500
Belden 85102	500	500

Tab 5-2 | Cable types

To avoid reflections at the line ends, a 55 Ω terminating resistor (terminator) needs to be fitted. If you use a Link Power power supply, the terminator is usually integrated. Exact details can be taken from the component manufacturer's details.

5.1.2 Line (bus) topology

Line (bus) topologies should be built preferably. The conditions vary somewhat from a free topology. Spurs with a length of up to 3 m are permissible. The following table provides information about the maximum lengths for different cable types.

To avoid reflections at the line ends, a 105 Ω terminating resistor (terminator) needs to be provided. If you use a Link Power power supply, the terminator is usually integrated.

5.1.3 Number of nodes

Irrespective of the topology the number of nodes being used in a segment is limited to 64 FFT nodes or up to 128 LPT nodes. For LPT nodes the power consumption must also be taken into account. Normally only one transceiver of the same type should be used in a segment.

Cable type	average power consumption	max. number of nodes	
		nominal	min.
320 m, uniform configuration, bus topology	125 mw 250 mw 500 mw	128 64 32	96 48 24
320 m, cumulative configuration, line or free topology	125 mw 250 mw 500 mw	64 32 16	48 24 12
150 m, cumulative configuration, line or free topology	125 mw 250 mw 500 mw	128 64 32	96 48 24

Tab 5-3 | Number of nodes

5.2 Infrastructure - components

Network interfaces are needed for connecting the PC to the Lon network.

Link Power Supply devices are needed for the power supply of LPT transceivers.

Repeaters or routers are able to override segment limits. Thus, for example, upon reaching the maximum line length or when exceeding the maximum number of nodes.

Repeaters do not have a filter function, they are used to connect segments of the same transmission medium.

Routers also connect segments with different transmission media. They have a filter function, which then only forwards telegrams to the other segment when the corresponding receiver is also located there.

Routers can be operated in 3 ways:

Configured	The router configuration must be created and loaded
Learning	The router "learns" which transmitter / receiver is in each segment from the telegram communication
Repeater	Acts as a signal booster or preprocessor

5.3 Workflow

Preparation

- Checking all information and creating the device and network plan
- Defining the topology with lines and routers/repeaters based on the device and network plan
- Defining the communication applications (bindings) and functions
- Creating the Lon project with the Lon tool and creating the bindings offline.

Commissioning

- Installing the devices
- Commissioning the devices
  (in so doing the bindings are transferred to the Lon nodes)

5.4 Network variables / Binding

LonWorks® communication basically consists of sending and receiving network variables. In principle, Lon distinguishes between standard and user-defined types:

➔ **snvt** Standard network variables

➔ **unvt** User-defined network variables

Each network variable is able to have just one communication direction, where the direction is always defined as being from the Lon device to the network:

➔ **nvi** read (input variable is received from the network)

➔ **nvo** write (output variable is sent to the network)

Furthermore, configuration parameters are defined, and these are normally only read or written by the Lon Engineering software (e.g. NL220) for device configuration. A direct exchange between Lon nodes is not foreseen.

➔ **nci**
configuration variable

Lon **S**tandard **N**etwork **V**ariable **T**ypes (**SNVT** for short) are listed in the "LonMark® NVT master list". Included in it are all the important details for creating a Lon configuration.

A Lon device, which is based on a Neuron chip FT 3120 or FT 3150, can contain up to 63 network variables (NVs) limited by the Neuron chip. The FT 5000 processor launched in 2010 supports up to 254 NVs for which the standard node object permanently occupies the first 7 network variables.

Each binding to another Lon device creates an entry in the address table of the Neuron chip. 15 entries are possible in this table irrespective of the chip. That means a Lon device is only able to communicate with 15 other Lon devices directly, regardless of how many **NV**s are connected. In addition, another 15 "Group bindings" are possible, i.e. the telegrams are sent as a broadcast to the receiver group.

These limitations do not depend on the transceiver, e.g. LPT, the limitations come via the Neuron chip.

SBC Lon over IP does not recognise these limitations !!!

Functional profile

Network variables are usually combined into functional profiles, thus for example the sccFanCoil profile has been defined for fan coil regulators. A Lon standard functional profile on the one hand describes the network interface with its input , outputs and configuration parameters. Beyond this, regulating and control functionality is also often defined, which can facilitate exchanging devices with the same functional profile.

An electronic device specification, the so-called "XIF" file is required for offline engineering. Included in it are all functional profiles with the associated network interfaces of a Lon device. When creating a Lon node in the LNS database based on the XIF file the bindings can be prepared offline in the office. During commissioning the data are sent to the Neuron chips and stored there.

If manufacturer-specific types should be used in the network variables, the associated resource files are also required from the device manufacturer in order to display the content as plain text in the Lon engineering software.

5.5 Communication / Service types

Data exchange of network variables takes place based upon protocol services, the properties of which can be individually defined during binding.

Acknowledged

- Secure connection by means of the "Acknowledged" response from the recipient of a telegram
- Telegrams are repeated until "Acknowledged"
- It is important to assume a slightly higher bus load as each data transmission consists of two telegrams, a transmission/response

Unacknowledged

- Insecure connection as "Acknowledged" from the recipient is not expected
- Communication errors are not detected

Unacknowledged, repeated

- Insecure connection as "Acknowledged" from the recipient is not expected
- The telegram is sent on multiple occasions (programmable)
- Short communication errors do not have any impact
- Persistent communication errors are not detected

6 The SBC Lon configurator

This chapter describes the Lon configurator from installation to details of its functions and how it can be used. It is similar in structure to a Quick Start which means that the individual steps can also be practically retraced in private study.

- Installation
- PG5 Project "Quickstart"
- Configuring Lon nodes

6.1 Installation

6.1.1 Checking an existing installation

Lon IP is available from version PG5 2.0. As from PG5 2.1 the Lon configurator has been completely revised and supports Lon IP as well as FTT10. It is recommended only to use PG5 2.1 for new projects. Existing projects can continue to be supported with PG5 2.0. If, however, adaptations should become necessary to the Lon configuration, it is recommended porting the project over to PG5 2.1.

If PG5 has already been installed, it is possible to check using the PG5 add-on tool dialog whether the Lon option has been installed.

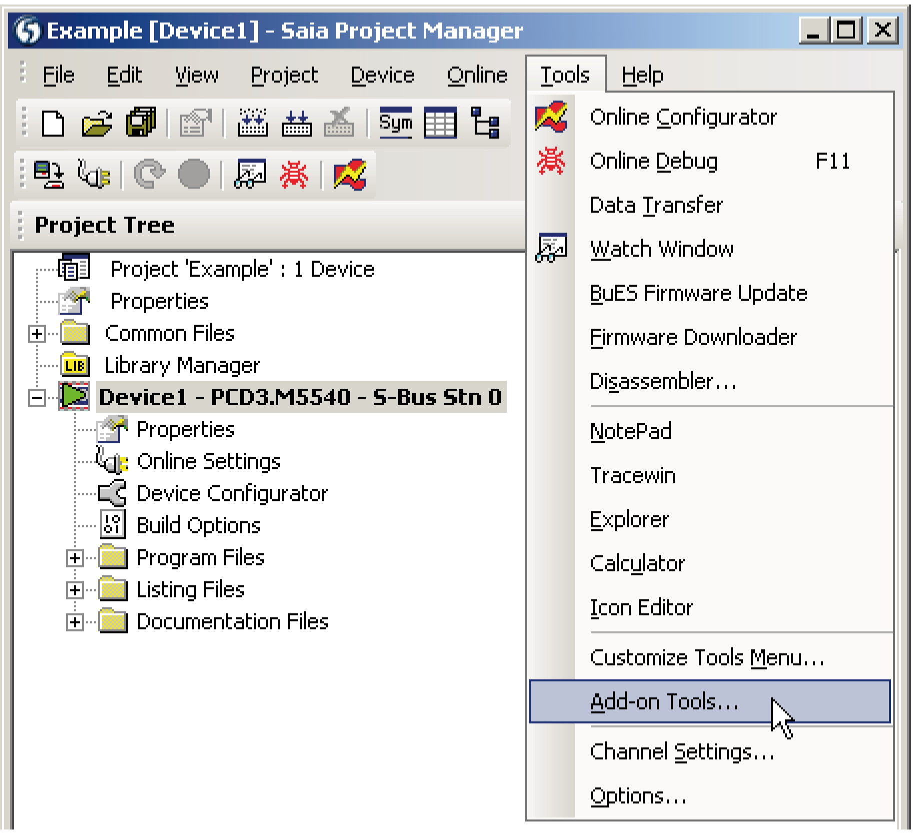
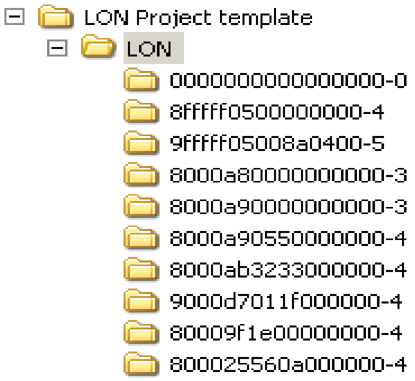


Fig. 6-1 | SBCProjectManager

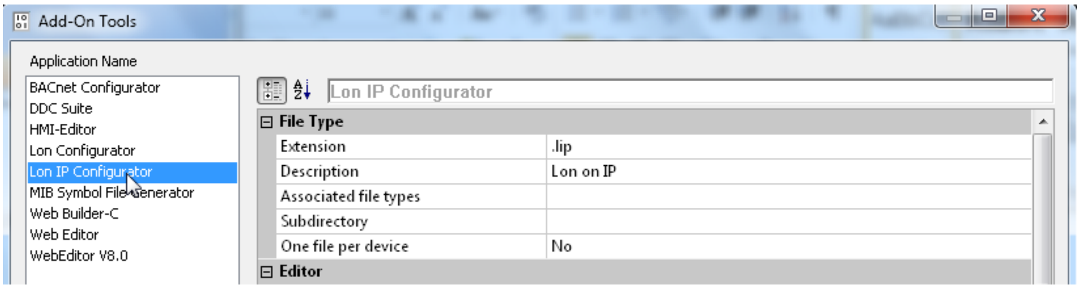
It is important to note that support for LonIP ("Lon IP Configurator") has been available for longer than support for LonFT ("Lon Configurator").

A distinction is made between the settings for the add-on tools for LonIP and LonFT in the rows "Extension", "Description", "Downloadable file extension" and "Downloadable file ID".



- LON Project template
  - LON
    - 0000000000000000-0
    - 8fffff0500000000-4
    - 9fffff05008a0400-5
    - 8000a80000000000-3
    - 8000a90000000000-3
    - 8000a90550000000-4
    - 8000ab3233000000-4
    - 9000d7011f000000-4
    - 80009f1e00000000-4
    - 800025560a000000-4

Fig. 6-2 | "Lon Configurator" add-on tools



Add-On Tools

Application Name

- BACnet Configurator
- DDC Suite
- HMI-Editor
- Lon Configurator
- Lon IP Configurator
- MIB Symbol File Generator
- Web Builder-C
- Web Editor
- WebEditor V8.0

Lon IP Configurator

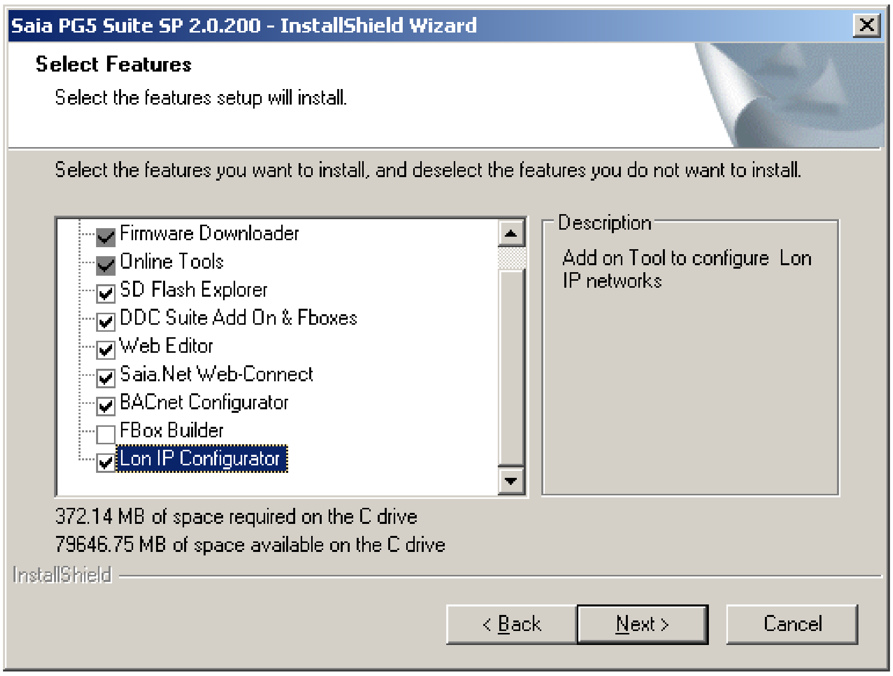
File Type	
Extension	.lip
Description	Lon on IP
Associated file types	
Subdirectory	
One file per device	No
Editor	

Fig. 6-3 | "Lon IP Configurator" add-on tools

If PG5 has been installed without Lon support, PG5 must be uninstalled and reinstalled as set out below. Projects that have already been created are retained. It is, however, recommended first creating a data backup.

6.1.2 Completing an installation

Fig. 6-4 | InstallOptions



Saia PG5 Suite SP 2.0.200 - InstallShield Wizard

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

- ☑ Firmware Downloader
- ☑ Online Tools
- ☑ SD Flash Explorer
- ☑ DDC Suite Add On & Fboxes
- ☑ Web Editor
- ☑ Saia.Net Web-Connect
- ☑ BACnet Configurator
- □ FBox Builder
- ☑ Lon IP Configurator

Description

Add on Tool to configure Lon IP networks

372.14 MB of space required on the C drive

79646.75 MB of space available on the C drive

Fig. 6-5 | InstallOptions

It is important to note during the installation of PG5 that the "Lon IP Configurator" option is selected. This will ensure that the required upgrades for Lon IP and LonFT are installed.

The following prompt windows should be answered accordingly (usually proceed by clicking <Next>).

After successful installation, the configurator, compiler and the Lon Project template are installed in the PG5 program directory.
For example: *C:\Program Files\SBC\PG5 V2.1.100*.

6.2 Lon project template

All newly created Lon configurations are derived from the "Lon Project Template" in which case the contents are copied to the PG5 project as soon as the Lon configurator starts creating a Lon project.

Changes to the project template are thus valid for all new projects, existing projects, however, remain unchanged.

In addition to a predefined director structure, the "Lon Project Template" also contains the LonMark standard resources (XML version 13) in the "0000000000000000-0" directory.

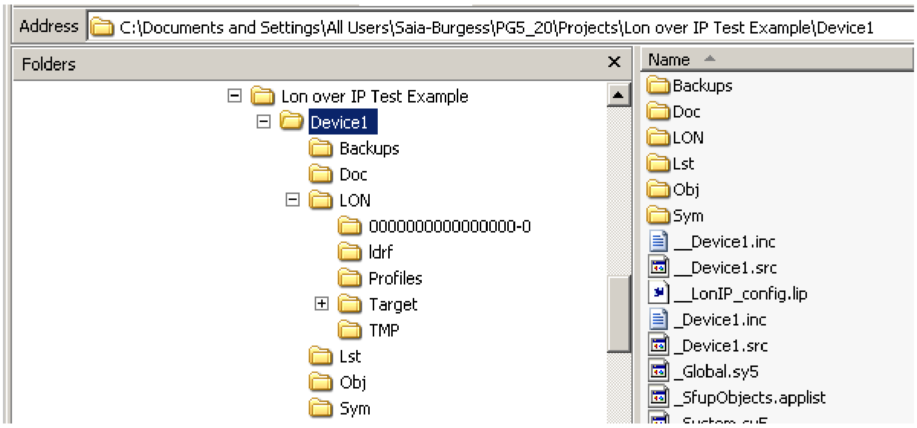


Fig. 6-6 | Lon project template

Fig. 6-6 shows the directory structure for a PG5 project with Lon configuration. The Lon-specific data are always stored in the "LON" subfolder. The name is permanently defined and cannot be modified. Because the Lon configurator manages the content itself, manual manipulation is not necessary.

6.3 Installation of other Lon templates

Other XML templates, e.g. of field components, can be saved in the same way as the LonMark standard resources. They are then available for general use for all new projects as resources.

For example: C:\Program Files\SBC\PG5_20\Lon Project Template

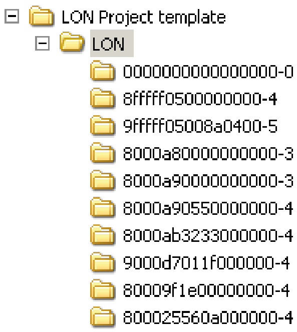
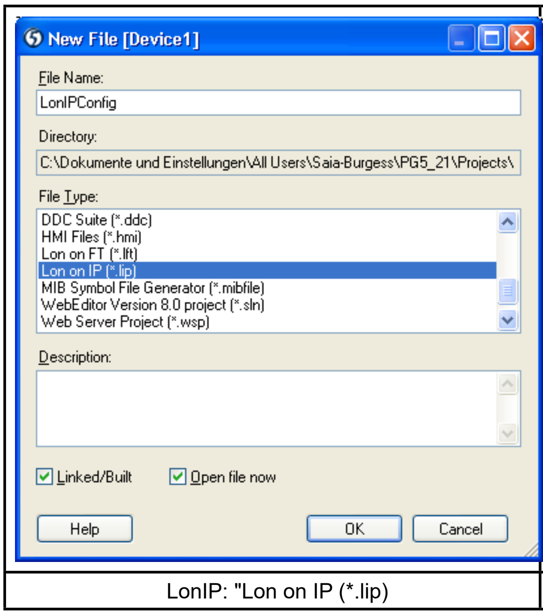
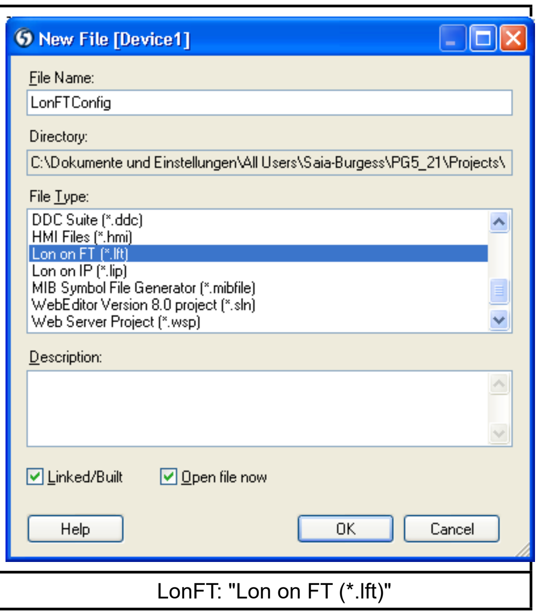


Fig. 6-7 | Lon project templates

6.4 Creating Lon nodes

As soon as the installation is complete, new projects can be created. First the device configuration should be completed using a PCD at one's discretion. A table with the minimum requirements in respect of PCD hardware and firmware is listed in **Chapter 1 "SBC overview of solutions"**.
To create a new PCD program it is recommended starting with the configuration of the Lon node. For this a distinction must be made between the LonIP and LonFT types.

	
LonIP: "Lon on IP (*.lip)	LonFT: "Lon on FT (*.lft)"

New File [Device1]

File Name: LonIPConfig

Directory: C:\Dokumente und Einstellungen\All Users\Saia-Burgess\PG5_21\Projects\

File Type:
- DDC Suite (*.ddc)
- HMI Files (*.hmi)
- Lon on FT (*.lft)
- Lon on IP (*.lip)
- MIB Symbol File Generator (*.mibfile)
- WebEditor Version 8.0 project (*.sln)
- Web Server Project (*.wsp)

Description:

☑ Linked/Built ☑ Open file now

New File [Device1]

File Name: LonFTConfig

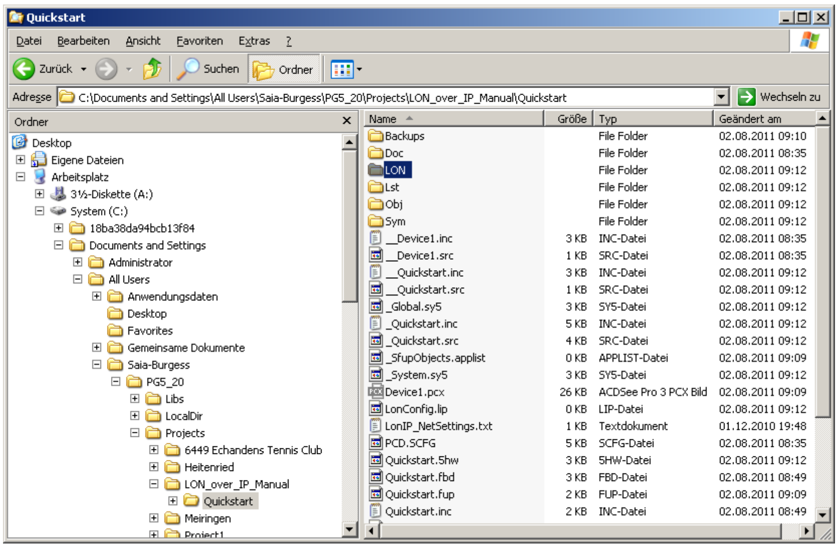
Directory: C:\Dokumente und Einstellungen\All Users\Saia-Burgess\PG5_21\Projects\

File Type:
- DDC Suite (*.ddc)
- HMI Files (*.hmi)
- Lon on FT (*.lft)
- Lon on IP (*.lip)
- MIB Symbol File Generator (*.mibfile)
- WebEditor Version 8.0 project (*.sln)
- Web Server Project (*.wsp)

Description:

☑ Linked/Built ☑ Open file now

PG5 now copies the Lon templates to the current device; in this example "Quickstart".



Adresse C:\Documents and Settings\All Users\Saia-Burgess\PG5_20\Projects\LON_over_IP_Manual\Quickstart

```
Ordner
Desktop
⊞ Eigene Dateien
⊟ Arbeitsplatz
  ⊞ 3½-Diskette (A:)
  ⊟ System (C:)
    ⊞ 18ba38da94bcb13f84
    ⊟ Documents and Settings
      ⊞ Administrator
      ⊟ All Users
        ⊞ Anwendungsdaten
          Desktop
          Favorites
        ⊞ Gemeinsame Dokumente
        ⊟ Saia-Burgess
          ⊟ PG5_20
            ⊞ Libs
            ⊞ LocalDir
            ⊟ Projects
              ⊞ 6449 Echandens Tennis Club
              ⊞ Heitenried
              ⊟ LON_over_IP_Manual
                ⊞ Quickstart
              ⊞ Meiringen
              ⊞ Project1
```

Name	Größe	Typ	Geändert am
Backups		File Folder	02.08.2011 09:10
Doc		File Folder	02.08.2011 08:35
LON		File Folder	02.08.2011 09:12
Lst		File Folder	02.08.2011 09:12
Obj		File Folder	02.08.2011 09:12
Sym		File Folder	02.08.2011 09:12
__Device1.inc	3 KB	INC-Datei	02.08.2011 08:35
__Device1.src	1 KB	SRC-Datei	02.08.2011 08:35
__Quickstart.inc	3 KB	INC-Datei	02.08.2011 09:12
__Quickstart.src	1 KB	SRC-Datei	02.08.2011 09:12
_Global.sy5	3 KB	SY5-Datei	02.08.2011 09:12
_Quickstart.inc	5 KB	INC-Datei	02.08.2011 09:12
_Quickstart.src	4 KB	SRC-Datei	02.08.2011 09:12
_SfupObjects.applist	0 KB	APPLIST-Datei	02.08.2011 09:09
_System.sy5	3 KB	SY5-Datei	02.08.2011 09:12
Device1.pcx	26 KB	ACDSee Pro 3 PCX Bild	02.08.2011 09:09
LonConfig.lip	0 KB	LIP-Datei	02.08.2011 09:12
LonIP_NetSettings.txt	1 KB	Textdokument	01.12.2010 19:48
PCD.SCFG	5 KB	SCFG-Datei	02.08.2011 08:35
Quickstart.5hw	3 KB	5HW-Datei	02.08.2011 09:12
Quickstart.fbd	3 KB	FBD-Datei	02.08.2011 08:49
Quickstart.fup	2 KB	FUP-Datei	02.08.2011 09:09
Quickstart.inc	2 KB	INC-Datei	02.08.2011 08:49

LonDirInDIvice

The Lon subdirectory contains all files, which are needed to create a Lon node. The "0000000000000000-0" directory contains the LonMark standard resources.

Proprietary Lon resources (these directories can be identified by the long numeric string as its name) must always be copied to the Lon subdirectory for use in the

SBC Lon node.

For more information about the LonMark standard resources, the documentation available online can be viewed at http://types.lonmark.org/.

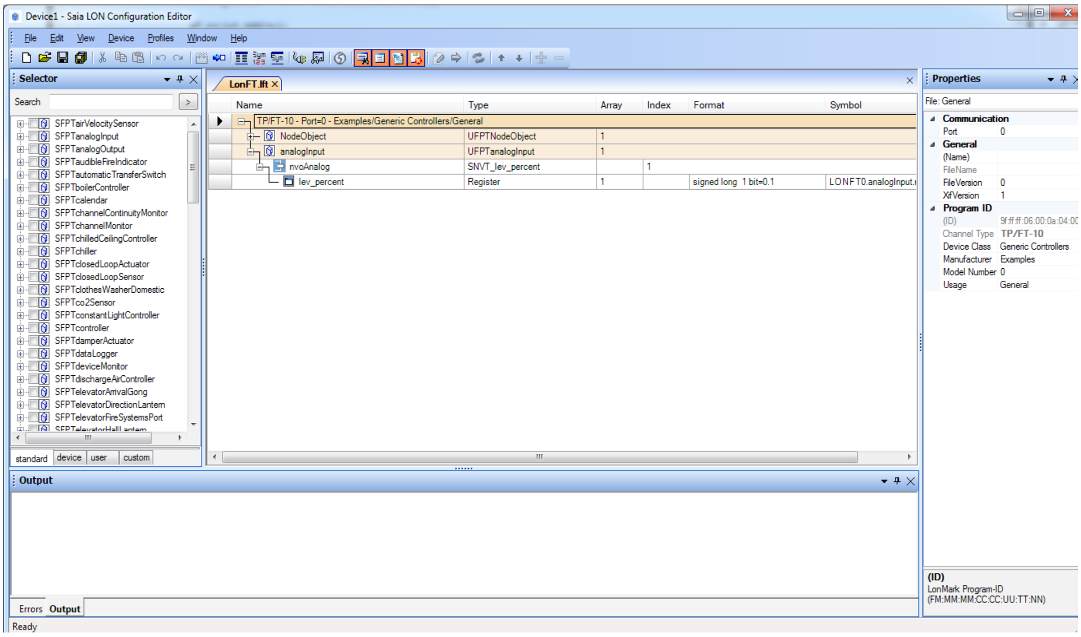
If proprietary Lon resources (these directories can be identified by the long numeric string as its name) are to be used as a template, they must be copied to the SBC Lon subdirectory. See chapter 6.3 'Installation of other Lon templates'

For more information about the LonMark standard resources, the documentation available online can be viewed at http://types.lonmark.org/.

At the time of printing this manual it was detected and indicated that only Microsoft Internet Explorer from version 7 is able to correctly display the content of the specified web site.

6.5 The Lon configurator

After creating a Lon node, the "SBC Lon Configuration Editor" window is automatically opened.



Name	Type	Array	Index	Format	Symbol
TP/FT-10 - Port=0 - Examples/Generic Controllers/General					
NodeObject	UFPTNodeObject	1			
analogInput	UFPTanalogInput	1			
nvoAnalog	SNVT_lev_percent		1		
lev_percent	Register	1		signed long 1 bit=0.1	LONFT0.analogInput.r

File: General

Communication	
Port	0
General	
(Name)	
FileName	
FileVersion	0
XifVersion	1
Program ID	
(ID)	9f:ff:ff:06:00:0a:04:00
Channel Type	TP/FT-10
Device Class	Generic Controllers
Manufacturer	Examples
Model Number	0
Usage	General

(ID)
LonMark Program-ID
(FM:MM:MM:CC:CC:UU:TT:NN)

Overview

This window consists of the following regions:

Menu	Description
Toolbar	Toolbar containing small icons for quick access to frequently used functions
Selector	Resources for use in profiles or Lon nodes
Configuration field	Display area for the Lon nodes and/or profiles. Multiple files can be opened at the same time, between which the programmer can switch using "tabs".
Properties	Settings, relate to the selected object in the configuration field or selector in each case
Errors	Errors, warnings and notifications
Output	Information

There are two different, although similar operational steps that should be distinguished:

Creating Lon profiles

Lon profiles are combined from network variables and configuration parameters.
This working step is required when further user-defined profiles (UFPT) need to be created for a Lon node.
See chapter **Creating and modifying profiles**

Creating Lon nodes

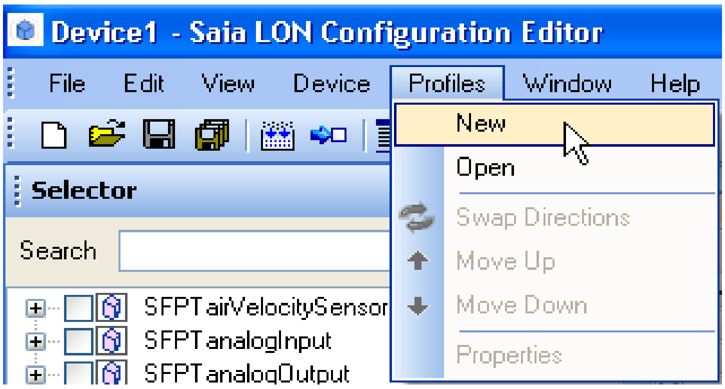
Lon nodes are created from Lon profiles.
See the chapter Creating and modifying Lon nodes.

6.5.1 Creating and modifying profiles

Create new	The profile is newly created completely from scratch
Converting existing profiles	An existing profile is modified but saved with a different name

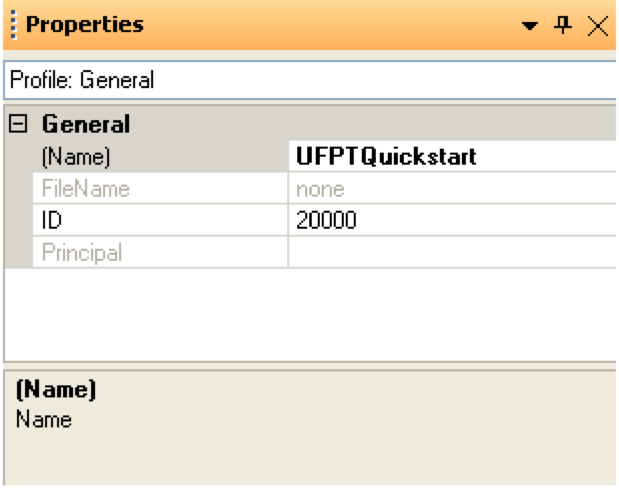
To create a profile, the "New"
entry should be clicked in the "Profiles" menu.

To edit an existing profile, the profile is opened "Open" by selecting the "Profiles" entry. The profile can then replace the existing content using the same name or be saved using "Save as" to a new profile.



NewProfileMenu

As a first step the profi8le should be give a logical name. If the properties of the profile are not yet displayed in the "Properties" section, the first row should be selected in the configuration field. Then the default name "UFPTdefaultName" can be modified. It is important to note that the name must always start with "UFPT".



(Name)	UFPTQuickstart
FileName	none
ID	20000
Principal	

(Name)
Name

Fig. 6-8 | PropertiesProfile

A number in the range between 20000 and 24999 must be selected for the ID. This ID is a numeric identifier for the profile. Within a Lon node this number must be one-to-one for each profile being used. The same number may only be repeated in different Lon nodes with a different profile.

In order to generate a new profile from an existing profile, the desired profile should be selected in the "Selector" area by clicking on the related checkbox. After this the profile can be opened by clicking on the "Edit Profile" icon in the configuration field. Alternatively, the context menu can be opened by right-clicking on the profile entry and then clicking on the "Edit Profile" menu item.

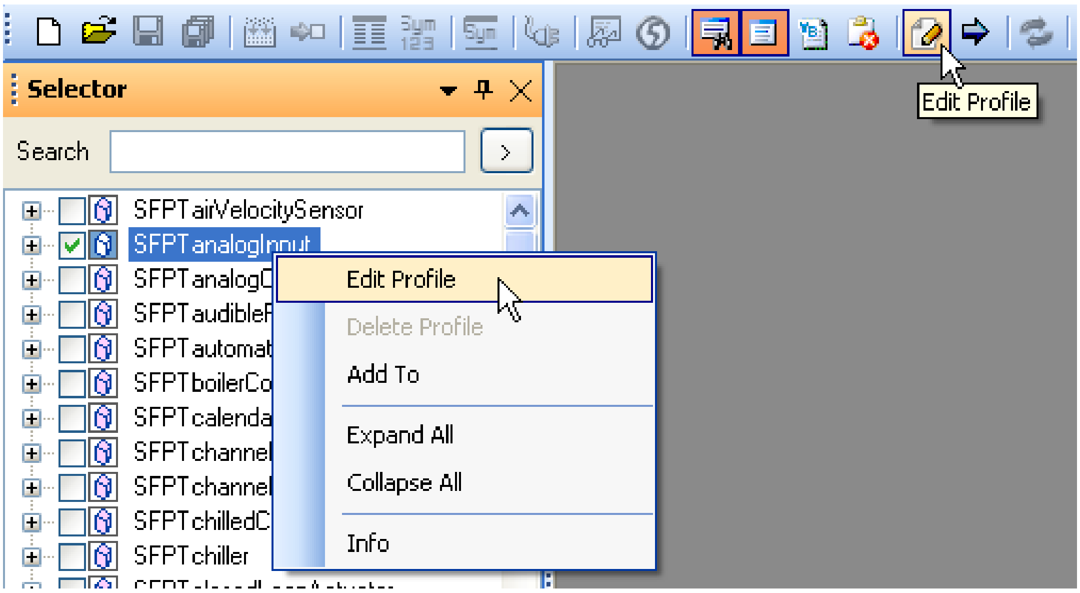


Fig. 6-9 | SelectorProfileMenu

If the existing profile is to be changed, you can immediately start with the actual configuration of the profile. If, on the other hand, a new profile is to be created from an existing profile, the name should be modified in the "Properties" area. If need be, this step can be omitted if a profile is being created based on a LonMark standard profile (SFPT) as the start of the name is already automatically modified by SFPT to UFPT.

Network variables and configuration parameters are added for the actual configuration of the profile. These components of the profile can be found in the "Selector" area under the following tabs:

nvs	LonMark Standard Network Variable Types (SNVT)
cps	LonMark Standard Configuration Parameter Types (SCPT)
custom	Network variable types or configuration parameter types, which are explicitly loaded beforehand. In order to load network variable types or configuration parameter types, the context menu can be opened by right-clicking and then selecting the "Load Profile" menu item.

A network variable can be added to a profile by drag & drop. To do this the network variable must be selected using the left-hand mouse button, dragged over the profile name and the mouse button is then released there. Alternatively, a network variable can be selected by click in the related checkbox and then added to the profile by clicking on the "Add to Target" icon.

In addition to this, the function can also be executed from the context menu by right-clicking on the network variable type and then selecting the "Add To" menu item.

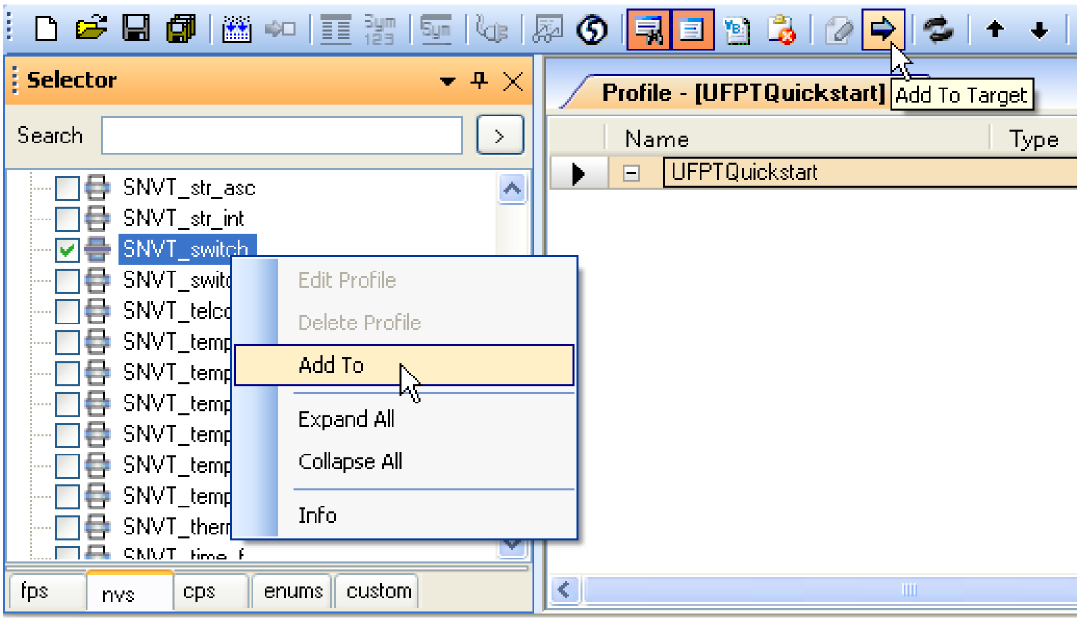
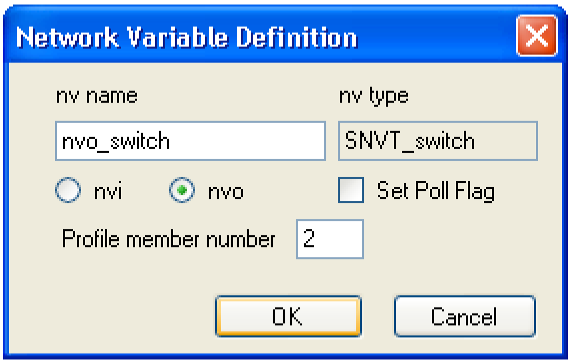


Fig. 6-10 | ADNVMenu

When adding, the "Network Variable Definition" dialog opens in which additional information can be specified. The name of the network variable can be specified in the field under the designation "nv name". It is important to note that the name must start with "nvi" or "nvo".

The direction (input or output) can also be chosen by selecting the "nvi" or "nvo" options set out below this. "Set Poll Flag" should only be selected for network variables input (nvi), if a polled binding is to be used, for instance it is recommended copying a "Profile member number" unchanged. The procedure can be completed by clicking on "OK" or can be cancelled by clicking on "Cancel".



Network Variable Definition

nv name: nvo_switch
nv type: SNVT_switch
○ nvi ● nvo
□ Set Poll Flag
Profile member number: 2

Fig. 6-11 | NVDef.

Configuration parameters are added to the profile in the same way as network variables.

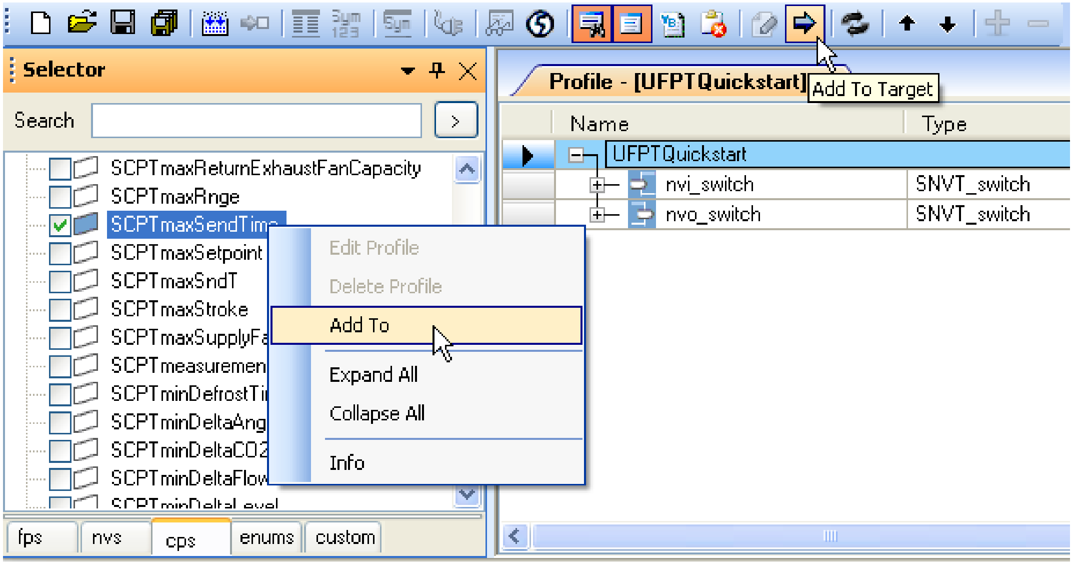
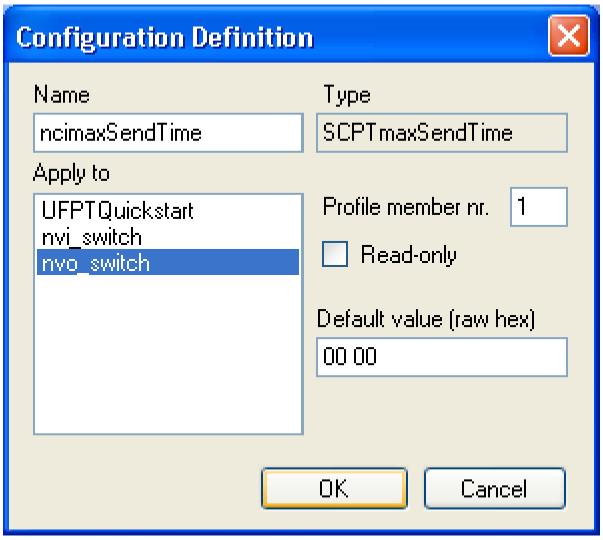


Fig. 6-12 | CPMenu

When adding, the "Configuration Definition" dialog opens in which additional information can be specified. The name of the network variable can be specified in the field under the designation "Name". It is recommended leaving the first three letters as "nci". In the field under the designation "Apply to" it is possible to select whether the configuration parameter should refer to a particular network variable (identifiable by the letter "nv" at the start) or to the profile (identifiable by the letters "UFPT" from the start).

Selecting "Read-only" prevents editing of the value during runtime. A default value can be specified for the value in the field under the designation "Default value (raw hex)". The values should be specified in hexadecimal notation. If the field is too small to display all values, it is possible to navigate within the field using the cursor keys (← or →). It is recommended copying the "Profile member No." unamended. The procedure can be completed by clicking on "OK" or can be cancelled by clicking on "Cancel".



Configuration Definition

Name: ncimaxSendTime

Type: SCPTmaxSendTime

Apply to:
UFPTQuickstart
nvi_switch
nvo_switch

Profile member nr.: 1

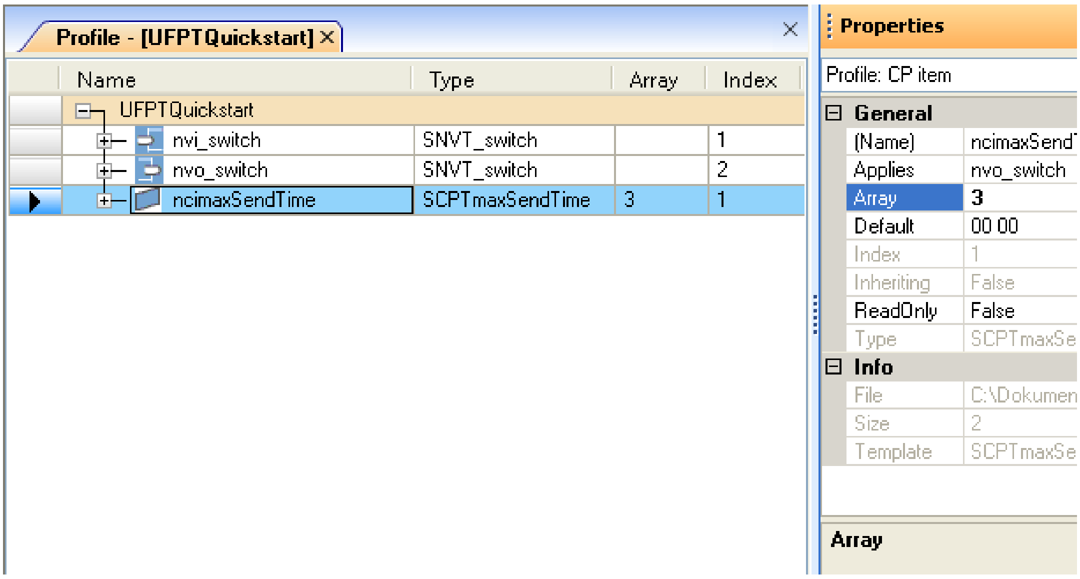
□ Read-only

Default value (raw hex): 00 00

Fig. 6-13 | CPDef.

Note for experienced users

Arrays of configuration parameters are needed for certain purposes. To set the number of elements, the corresponding configuration parameter must first be selected in the configuration field. After this the number of required elements can be set in the "Properties" area on the "Array" row.



Profile - [UFPTQuickstart]

Name	Type	Array	Index
UFPTQuickstart			
nvi_switch	SNVT_switch		1
nvo_switch	SNVT_switch		2
ncimaxSendTime	SCPTmaxSendTime	3	1

Properties

Profile: CP item

General	
(Name)	ncimaxSend`
Applies	nvo_switch
Array	3
Default	00 00
Index	1
Inheriting	False
ReadOnly	False
Type	SCPTmaxSe
Info	
File	C:\Dokumen
Size	2
Template	SCPTmaxSe

Array

Fig. 6-14 | CPArraySize

Tip

If the direction (input ⇄ output) of all network variables is to be reversed, this is possible by clicking on the "Swap Directions" icon or by selecting the "Swap Directions" menu item in the "Profiles" menu. This function is particularly useful when the counterpart to an existing profiles is to be created.

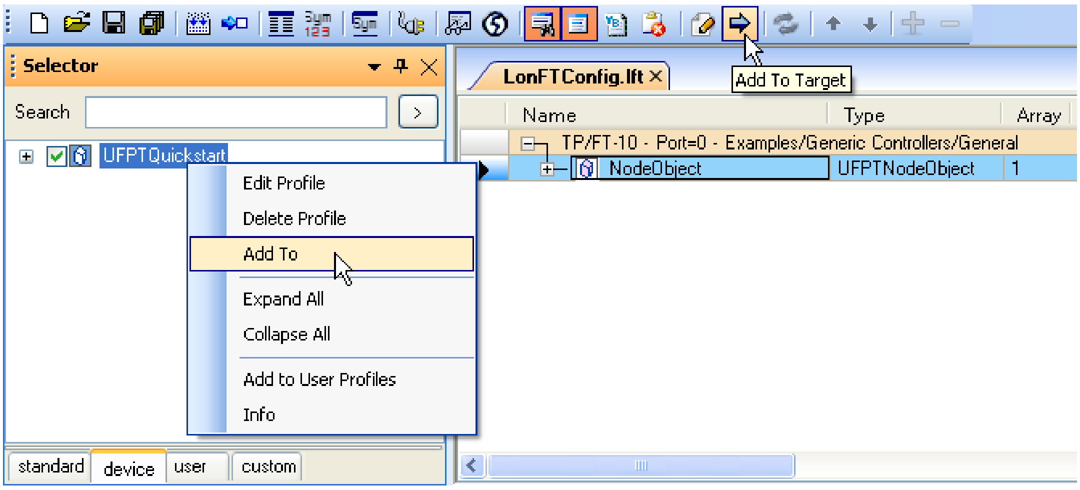
The profile can be save by clicking on the "Save" icon 💾 or by clicking on the "Save" menu item in the "File" menu.

6.5.2 Creating and modifying LON nodes.

First you should carefully note that the intended Lon node is displayed in the configuration field. The name, in particular, should be checked.
Profiles can then be added to the Lon node.

standard	LonMark Standard Profile (SFPT)
device	Device-specific profiles (directory PG5_xy\ Projects\<Project>\<Device>\LON\Profiles)
user	The profiles, which were previously created (as set out in the previous chapter), are save here
custom	User-specific profiles (directory PG5_xy\LON\Profiles) Profiles, which where previously explicitly loaded To load profiles the context menu needs to be opened by right-clicking and selecting the "Load Profile" menu item.

Profiles are added to a node in the same way as network variables are to a profile, see chapter 6.6.2 "Creating and modifying profiles".

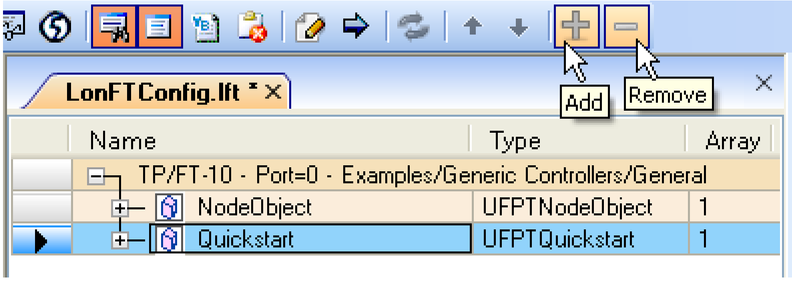


Name	Type	Array
TP/FT-10 - Port=0 - Examples/Generic Controllers/General		
NodeObject	UFPTNodeObject	1

Fig. 6-15 | AddProfileMenu

The success of the action can be identified insofar as the corresponding profile for the Lon node is updated in the configuration field.

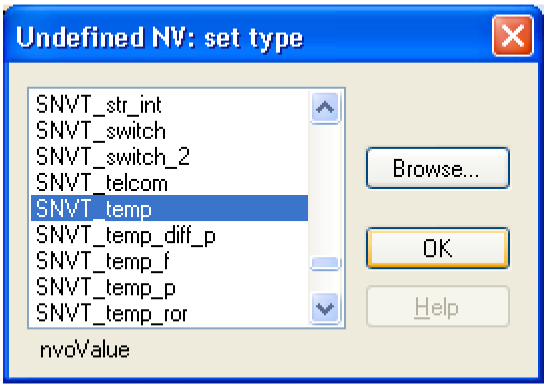
If multiple instances of a profile are to be added, the procedure can be repeated several times. The number of instances of a profile can be seen in the "Array" column. Alternatively, by clicking on the "Add" icon the number of instances can be incremented by 1. The number of instances can be reduced by 1 by clicking on the "Remove" icon or pressing the "Delete" or "Del" key.



Name	Type	Array
TP/FT-10 - Port=0 - Examples/Generic Controllers/General		
NodeObject	UFPTNodeObject	1
Quickstart	UFPTQuickstart	1

Fig. 6-16 | AddRemoveProfile

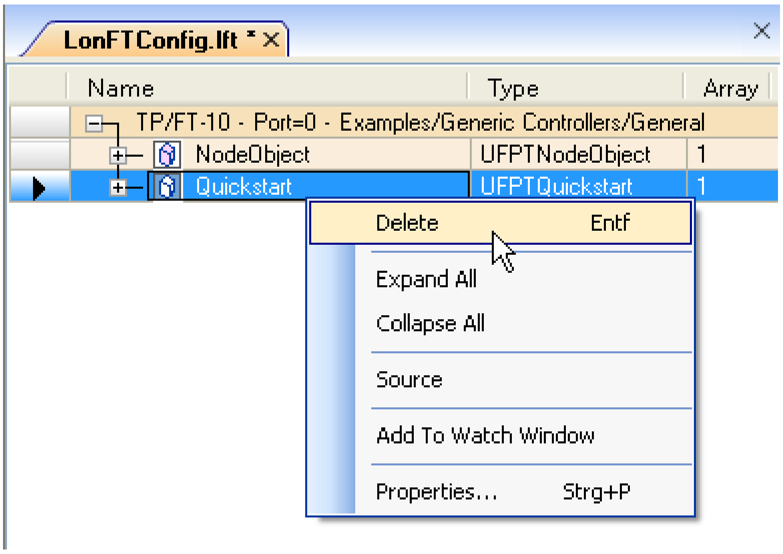
When adding certain profiles, the "Undefined NV: set type" dialog appears. For these kinds of profiles the types have not yet been defined for some network variables for reasons of universal usability. This can now be updated in the dialog that is displayed. In each case it is displayed under the list for which network variables the type is to be defined. The desired network variable type can be selected from the list. Only LonMark standard network variables are listed there for each default setting. If user-specific network variable types are needed, they can be loaded by clicking on the "Browse..." button.
The procedure can be completed or continued by clicking on "OK" or can be cancelled by clicking on "Cancel".


Undefined NV: set type

nvoValue

Fig. 6-17 | DefNVType

To delete a profile from a Lon node, the context menu is opened by right-clicking on the profile entry. There the "Delete" menu item should be clicked in which case the procedure needs to be confirmed by clicking "Yes" when the "Delete Node" dialog appears. It is important to note that all instances of a profile are removed during the deletion process.

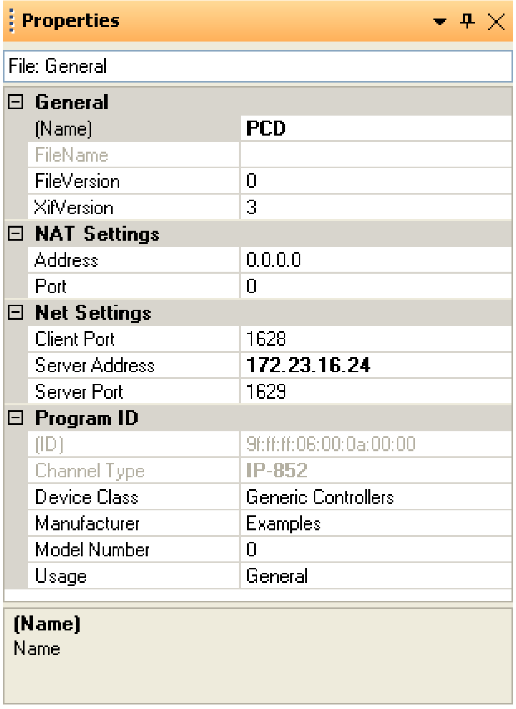


Name	Type	Array
TP/FT-10 - Port=0 - Examples/Generic Controllers/General		
NodeObject	UFPTNodeObject	1
Quickstart	UFPTQuickstart	1

Fig. 6-18 | DeleteProfil

Until the Lon node is generated, the profiles being included can be modified without problem. Once the Lon node has been installed in an application, any change to the profiles and thus to the Lon node is only possible under extremely difficult conditions, e.g. loss of bindings. It is not possible to provide a general statement at this point as the modification method is critical as to whether the bindings can be retained or not. A template needs to be replaced in any case in the Lon database.

The properties of the Lon node are displayed in the "Properties" area provided the top entry is selected in the configuration field. These properties are different for LonIP nodes and LonFT nodes:

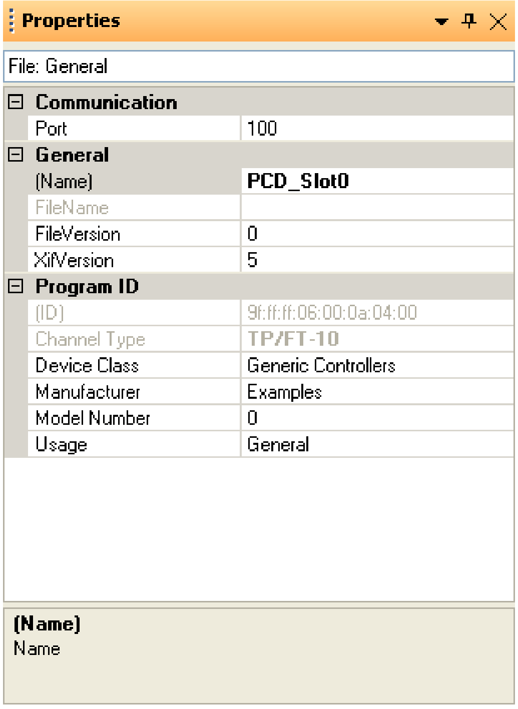


File: General

General	
(Name)	**PCD**
FileName	
FileVersion	0
XifVersion	3
NAT Settings	
Address	0.0.0.0
Port	0
Net Settings	
Client Port	1628
Server Address	**172.23.16.24**
Server Port	1629
Program ID	
(ID)	9f:ff:ff:06:00:0a:00:00
Channel Type	**IP-852**
Device Class	Generic Controllers
Manufacturer	Examples
Model Number	0
Usage	General

(Name)
Name

Fig. 6-19 | Settings for LonIP



File: General

Communication	
Port	100
General	
(Name)	**PCD_Slot0**
FileName	
FileVersion	0
XifVersion	5
Program ID	
(ID)	9f:ff:ff:06:00:0a:04:00
Channel Type	**TP/FT-10**
Device Class	Generic Controllers
Manufacturer	Examples
Model Number	0
Usage	General

(Name)
Name

Fig. 6-20 | Settings for LonFT

Communication	General settings (only for LonFT)
Port	100, 100, 120 or 130 for LonFT module in slot 0, 1, 2 or 3
General	General settings (for LonIP and LonFT)
(Name)	Optional name is displayed in the commissioning software
FileName	
FileVersion	
XifVersion	
NAT Settings	Settings for network address translation (only for LonIP)
Address	
Port	
Net Settings	Network settings (only for LonIP)
Client Port	
Server Address	
Server Port	
Program ID	Program identification (for LonIP and LonFT)
(ID)	Is calculated from the values below
Channel Type	Cannot be modified: IP-852 for LonIP or TP/FT-10 for LonFT
Device Class	Device type
Manufacturer	Manufacturer
Model Number	Model Number
Usage	Scope of application, purpose of application

Note
In each case a brief help tip is displayed on the selected entry at the bottom of the "Properties" area..

If the configuration of the Lon node is complete and the properties have been set accordingly, one concluding step is required to subsequently be able to use the configuration in the project.

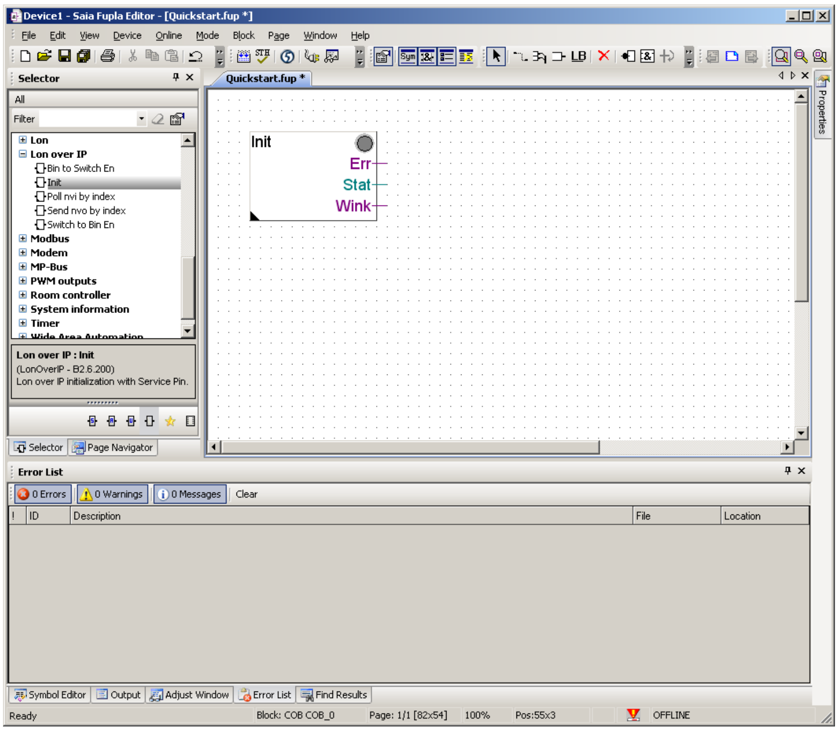
By clicking on the "Create Target" icon of by clicking on the "Create Target" menu item in the "Device" menu, the required data are generated from the configuration of the Lon node.

Any warnings, errors and messages are displayed in the "Output" area. Provided no error messages highlighted in red appear, the procedure has been successful. The Lon configurator can now be closed.

6.6 PCD user program

After the "Create Target" function has been completed in the Lon configurator, all registers and flags relating to the network variables are available to the programmer as public symbols from the Symbol Editor. The symbols correspond to the name of the profiles and network variables from the Lon node in this.

A "Lon over IP" FBox library provides modules for simple initialisation, data conversion, data exchange and some test functions. A description of the FBoxes is available as online help.



Lon over IP : Init
(LonOverIP - B2.6.200)
Lon over IP initialization with Service Pin.

Fig. 6-21 | FuplaEditor

7 LON commissioning software

7.1 Network interface

A "Lon Binding" software uses the so-called network interface to connect with the LonWorks® network. The link between the hardware interface and the software is established by means of appropriate device driver software. The link can be established via Ethernet or via an external adapter. The driver software for Ethernet links is usually installed together with the "binding software". For external devices the associated driver is usually obtained as part of the supply or is made available for downloading from the Internet using the provider's web page.

7.1.1 LONFT

To connect the "Lon binding" software with FTT10, an external device (gateway) is required. Usually, an FTT10 / USB adapter is used as a network interface. Corresponding adapters are provided by the companies Echelon or LOYTEC, amongst others.

7.1.2 LONIP with IP852 configuration server

The network interface can be connected directly with the Lon network via Ethernet (IP). No external hardware is necessary. However, a Lon <> IP infrastructure does need an IP-852 configuration server in the network. This task can be completed by a program on the Lon Engineering Tool PC. Otherwise, some embedded devices support this service.

The Config Server contains a list of IP addresses into which all Lon IP devices must be entered. Lon IP communication is not possible without the correctly completed list.

The config server should be activated for any adaptation to the Lon network using the "binding tool". For larger-scale Lon installations, it is strongly recommended viewing the Lon Engineering PC with the "binding tool" and the config server as being part of the system and to permanently install them on site.

Procedure:

- Start the config server
- Register and/or activate the network interface of the PC using the test function
- Check channel list, are all Lon IP devices accessible on the network?

The software solution with the "Echelon LonWorks IP Configuration Server" is illustrated below. This config server is part of the installation of the LONMAKER or NL220 Lon Engineering Software.

Starting the configuration server

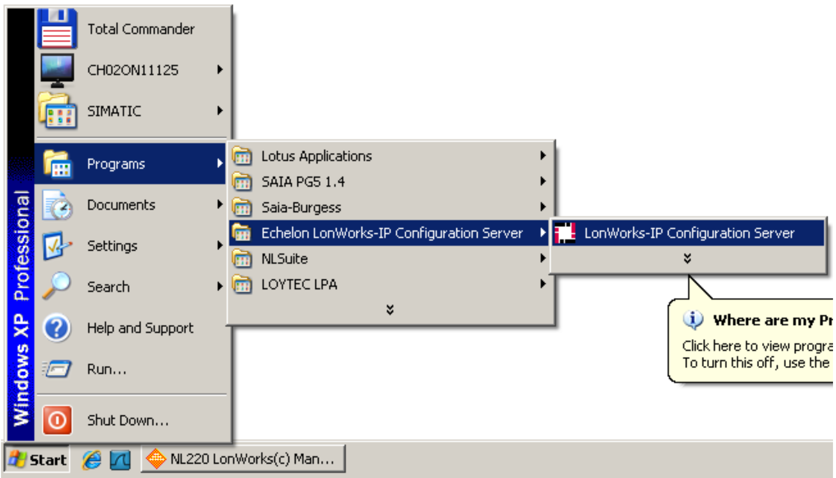
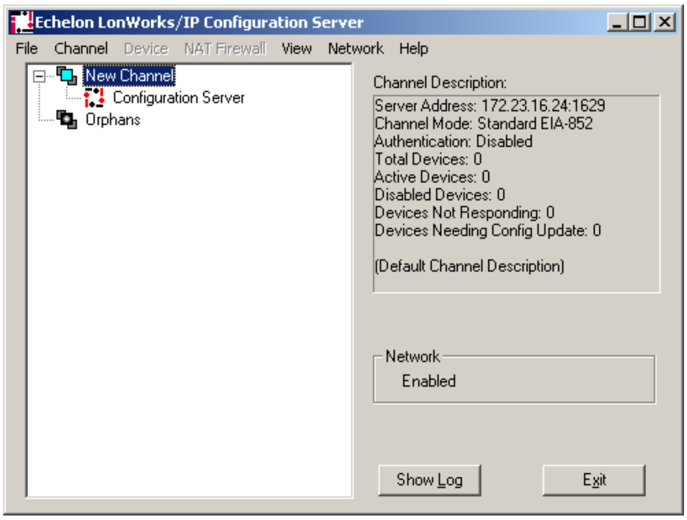


Fig. 7-1 | Starting the LonWorks-IP configurator


Echelon LonWorks/IP Configuration Server
File Channel Device NAT Firewall View Network Help
New Channel
Configuration Server
Orphans
Show Log
Exit

Channel Description:

Server Address: 172.23.16.24:1629
Channel Mode: Standard EIA-852
Authentication: Disabled
Total Devices: 0
Active Devices: 0
Disabled Devices: 0
Devices Not Responding: 0
Devices Needing Config Update: 0

(Default Channel Description)

Network: Enabled

Fig. 7-2 | Echelon IP Configuration Server as a PC program

Registering the PC network interface

The LonWorks® interface requires the connection to the IP-852 configuration server. For this purpose, as illustrated in Fig. 7-2, the checkbox is enabled and the IP address is registered with the config server port. If, as displayed in this example, the config server and the Lon interface are installed on the same PC, the IP address of the network card connected with the Lon network should be selected.

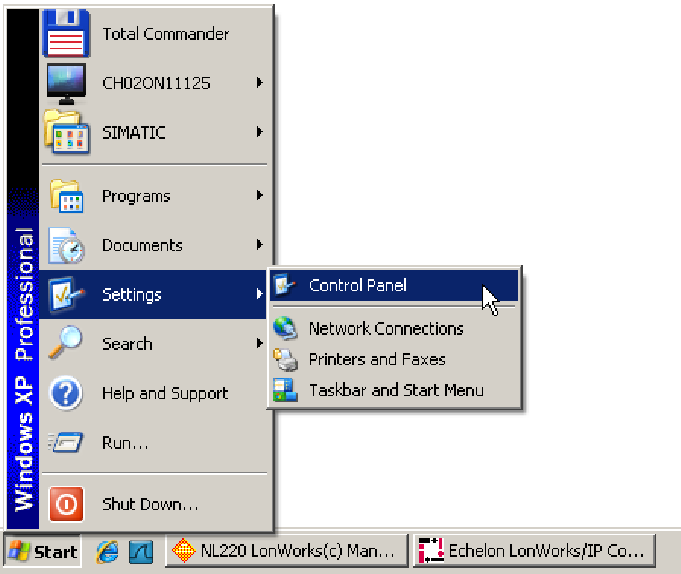


Fig. 7-3 | Open Windows system settings

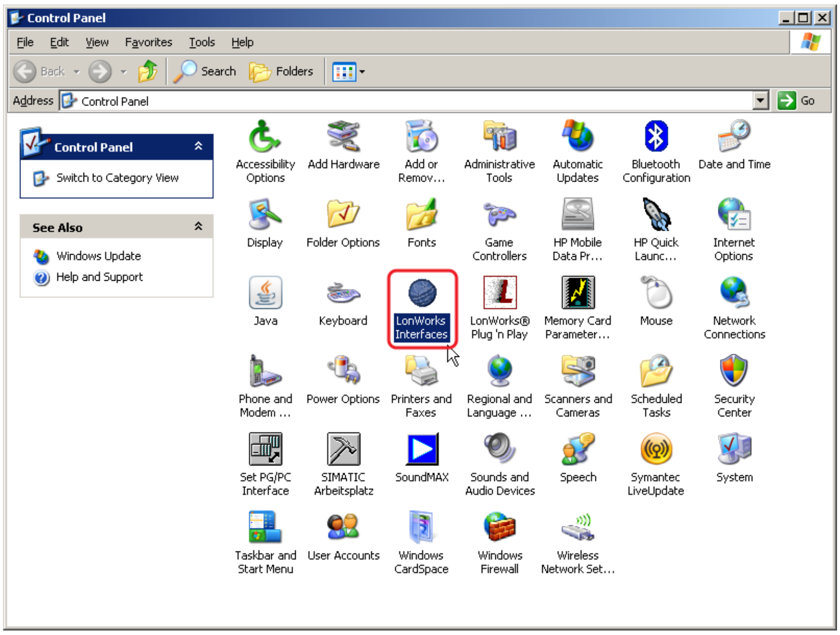


Fig. 7-4 | Start Lon interface

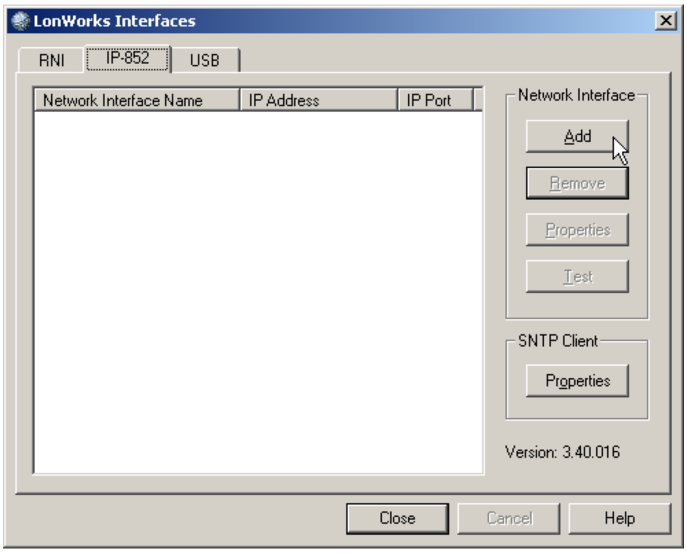
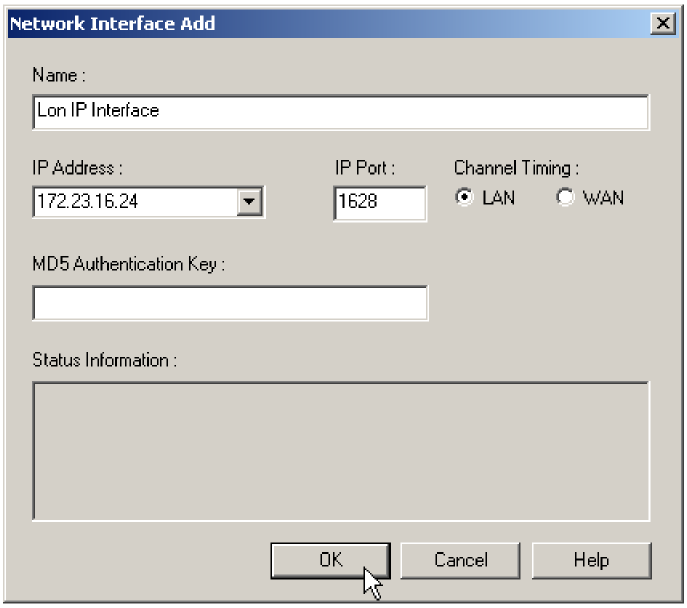


Fig. 7-5 | Add new interface



Name : Lon IP Interface

IP Address : 172.23.16.24

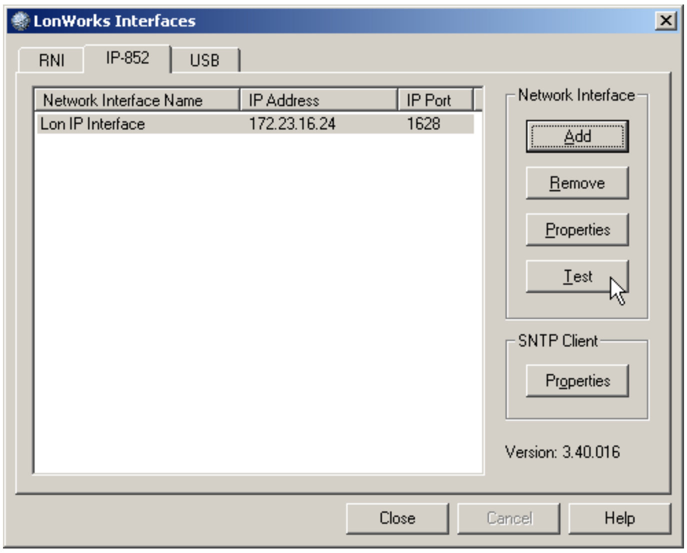
IP Port : 1628

Channel Timing : ● LAN ○ WAN

MD5 Authentication Key :

Status Information :

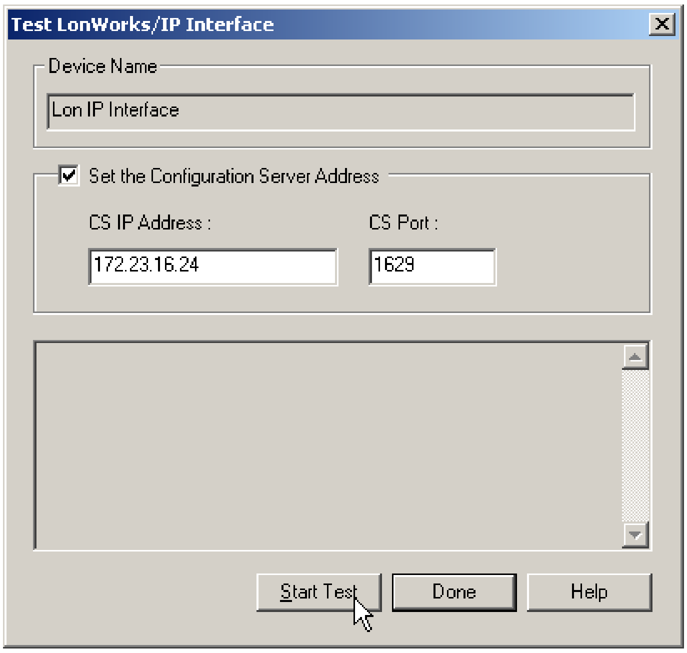
Fig. 7-6 | Configure interface



Network Interface Name	IP Address	IP Port
Lon IP Interface	172.23.16.24	1628

Version: 3.40.016

Fig. 7-7 | Test interface



Device Name: Lon IP Interface

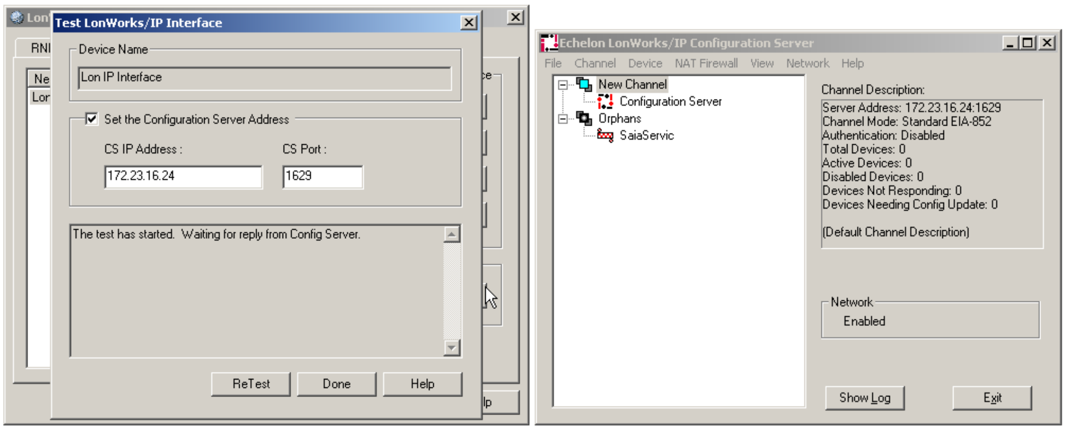
☑ Set the Configuration Server Address

CS IP Address : 172.23.16.24

CS Port : 1629

Fig. 7-8 | Run test function

As soon as the interface test is run, the Lon interface attempts to establish a connection with the config server. A new device will appear in the list of detected by as yet unassigned devices (orphans). Using the mouse pointer and the "drag-and-drop" method this entry is added to the channel list of the configuration server.


Test LonWorks/IP Interface
Echelon LonWorks/IP Configuration Server
File Channel Device NAT Firewall View Network Help
New Channel
Configuration Server
Orphans
SaiaServic
ReTest Done Help
Show Log Exit

Device Name: Lon IP Interface

☑ Set the Configuration Server Address

CS IP Address : 172.23.16.24
CS Port : 1629

The test has started. Waiting for reply from Config Server.

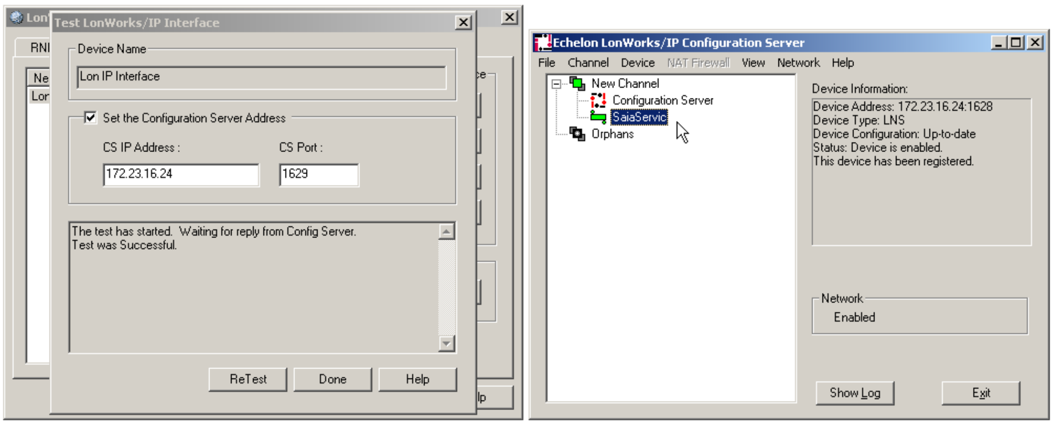
Channel Description:
Server Address: 172.23.16.24:1629
Channel Mode: Standard EIA-852
Authentication: Disabled
Total Devices: 0
Active Devices: 0
Disabled Devices: 0
Devices Not Responding: 0
Devices Needing Config Update: 0

(Default Channel Description)

Network: Enabled

Fig. 7-9 | Interaction of the Lon Interface Test and Config Server

As soon as the new device, in this case the PC itself, is included in the channel list, the function test on the interface stops with an appropriate message.


Test LonWorks/IP Interface
Echelon LonWorks/IP Configuration Server
File Channel Device NAT Firewall View Network Help
New Channel
Configuration Server
SaiaServic
Orphans
ReTest Done Help
Show Log Exit

Device Name: Lon IP Interface

☑ Set the Configuration Server Address

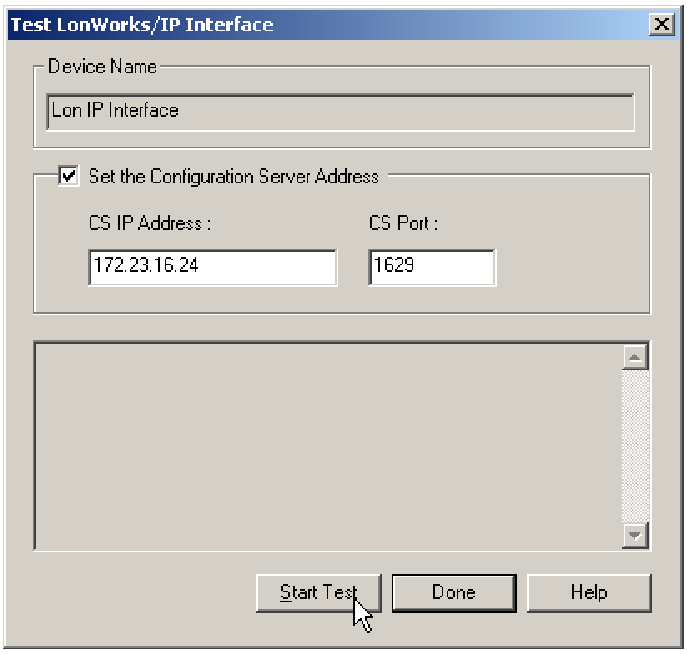
CS IP Address : 172.23.16.24
CS Port : 1629

The test has started. Waiting for reply from Config Server.
Test was Successful.

Device Information:
Device Address: 172.23.16.24:1628
Device Type: LNS
Device Configuration: Up-to-date
Status: Device is enabled.
This device has been registered.

Network: Enabled

Fig. 7-10 | PC registration on to the config server successfully completed



Test LonWorks/IP Interface

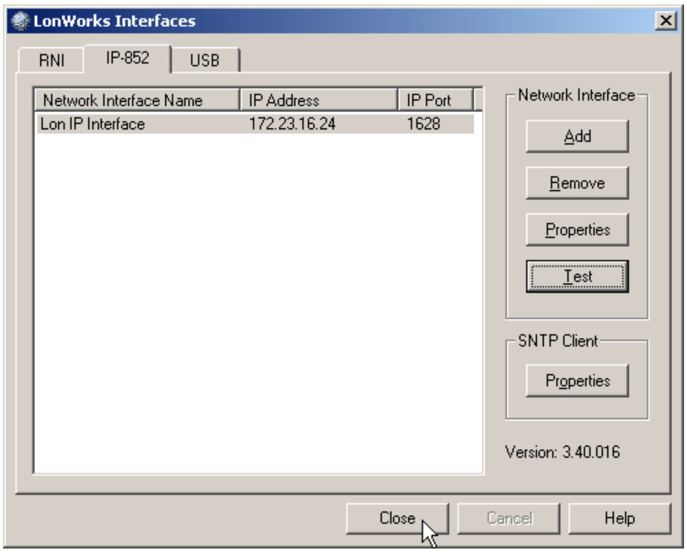
Device Name: Lon IP Interface

☑ Set the Configuration Server Address

CS IP Address : 172.23.16.24

CS Port : 1629

Fig. 7-11 | End Lon interface test function



LonWorks Interfaces

Network Interface Name	IP Address	IP Port
Lon IP Interface	172.23.16.24	1628

Version: 3.40.016

Fig. 7-12 | The dialog for the Lon interface can be closed

7.1.3 Registering the PCD (Lon node)

As soon as the user program is loaded into the PCD, the PCD automatically logs on after starting the Lon firmware on the config server. In the same way as the PC it is added to the channel list via the orphans list using the "drag-and-drop" method.

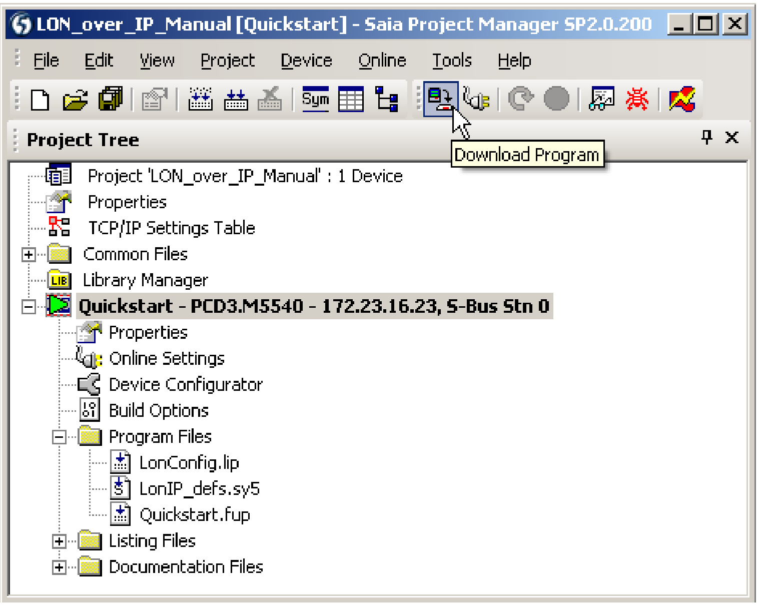
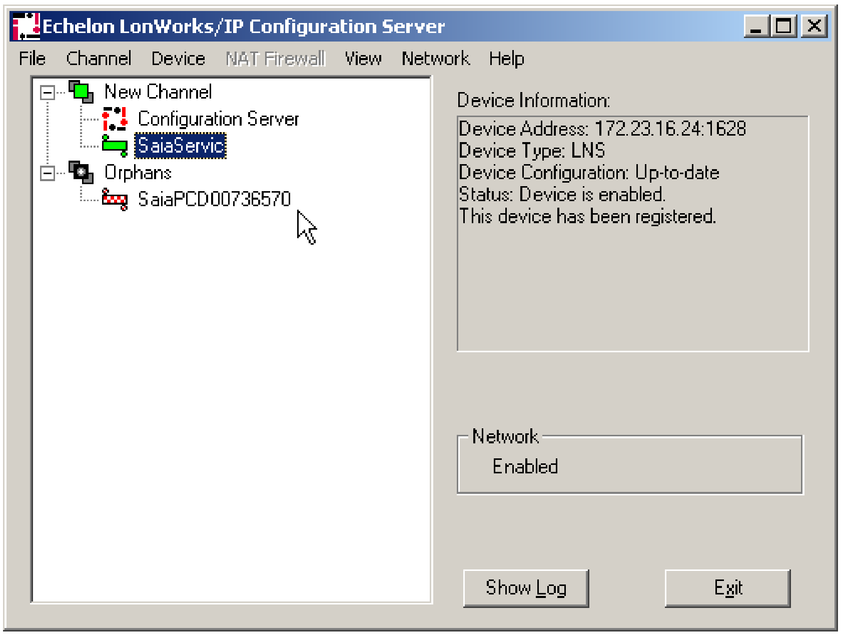


Fig. 7-13 | User program download

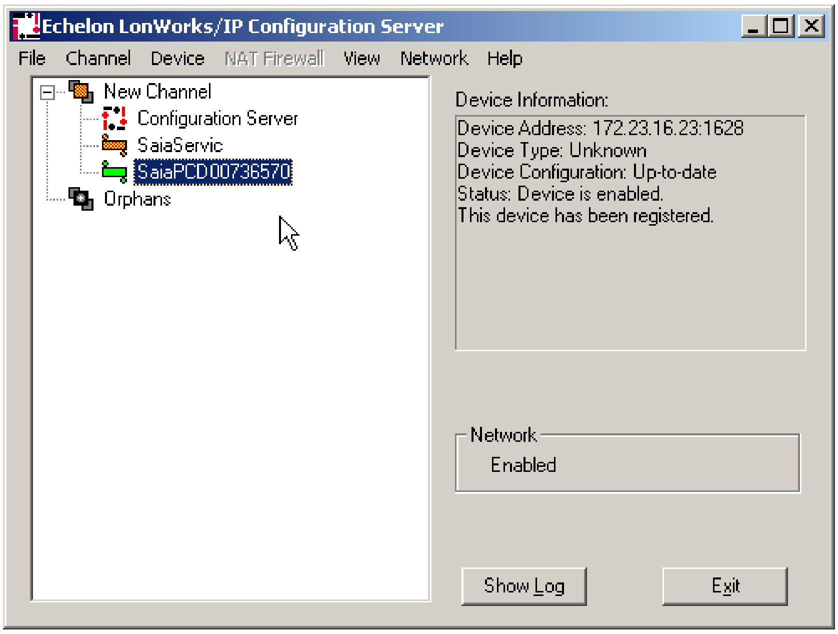

Echelon LonWorks/IP Configuration Server
File Channel Device NAT Firewall View Network Help
New Channel
Configuration Server
SaiaServic
Orphans
SaiaPCD00736570
Show Log
Exit

Device Information:

Device Address: 172.23.16.24:1628
Device Type: LNS
Device Configuration: Up-to-date
Status: Device is enabled.
This device has been registered.

Network: Enabled

Fig. 7-14 | PCD logs on to the config server



Device Information:
Device Address: 172.23.16.23:1628
Device Type: Unknown
Device Configuration: Up-to-date
Status: Device is enabled.
This device has been registered.

Fig. 7-15 | PCD and PC are ready for commissioning

Only when all Lon over IP devices could be successfully added to the channel list, is it possible to continue with Lon commissioning.

7.2 Commissioning LON nodes

The configuration server must be activated without interruption throughout the entire commissioning process.

After this the NL220 software from the NLSuite of Newronsystems is used for commissioning the Lon nodes. This software, like many other "binding tools" (LonMaker, Alex...), is also based on Echelon's LNS database technology.

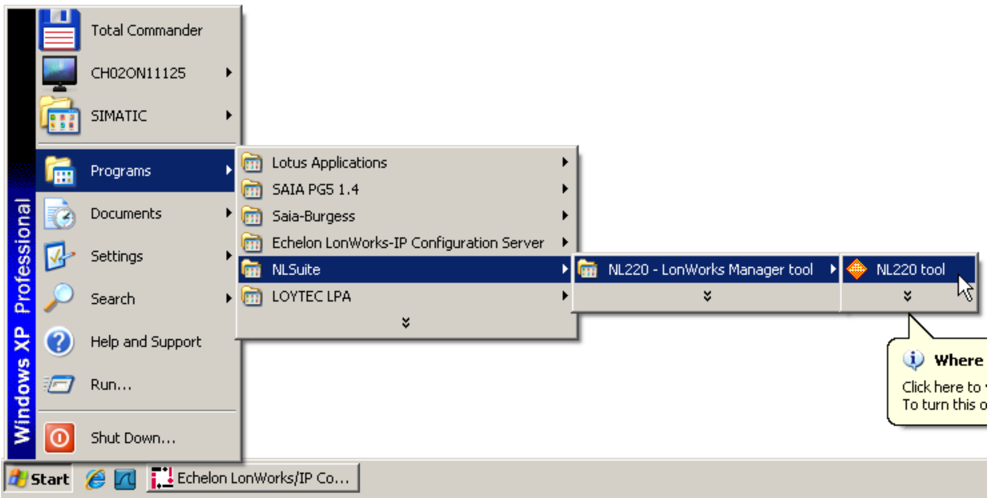


Fig. 7-16 | Starting the NL220 software

After starting NL220 the project selection automatically opens. In the Demo version only the DEMO project is available.

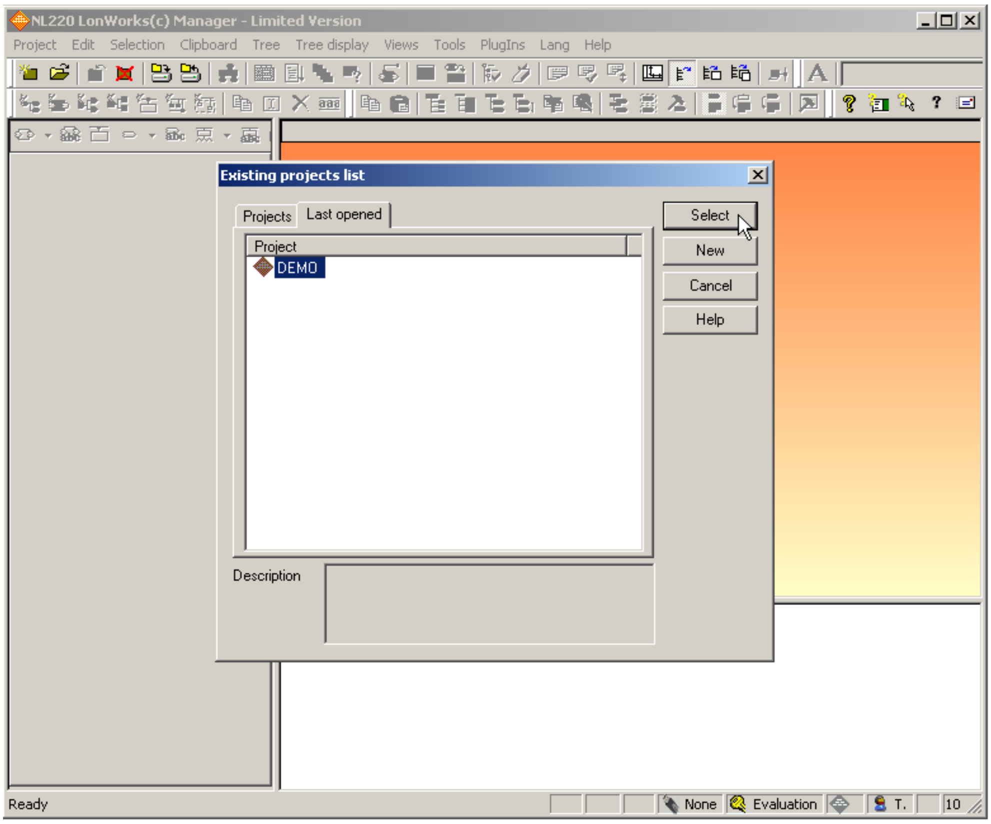
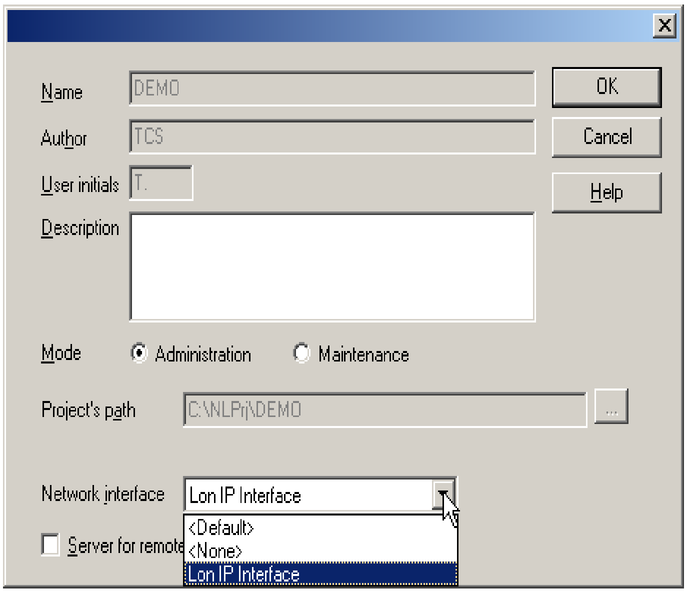

NL220 LonWorks(c) Manager - Limited Version
Project Edit Selection Clipboard Tree Tree display Views Tools PlugIns Lang Help
Existing projects list
Projects
Last opened
Project
DEMO
Select
New
Cancel
Help
Description
Ready
None
Evaluation
T.
10

Fig. 7-17 | Opening the Demo project

The network interface can also be used in demo mode.

You should always select the interface, which has been registered on the config server in the channel list.



Name: DEMO

Author: TCS

User initials: T.

Description:

Mode: ● Administration ○ Maintenance

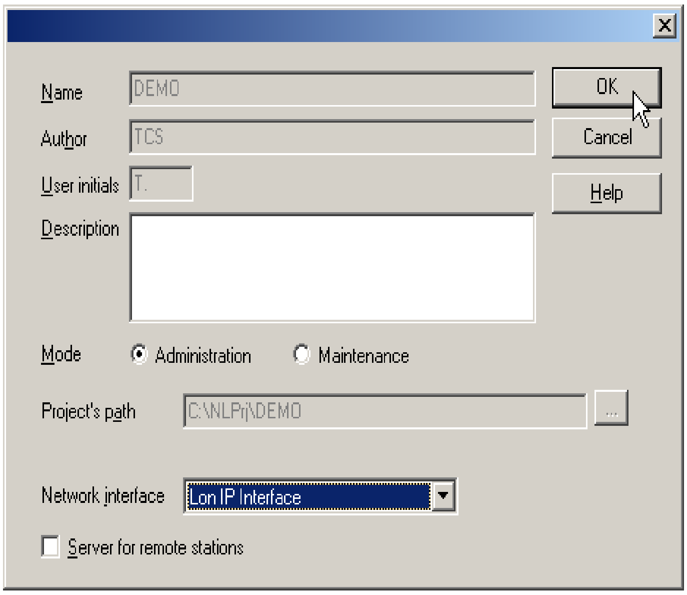
Project's path: C:\NLPrj\DEMO

Network interface: Lon IP Interface

- <Default>
- <None>
- Lon IP Interface

□ Server for remote

Fig. 7-18 | Selecting the network interface



Name: DEMO

Author: TCS

User initials: T.

Description:

Mode: ● Administration ○ Maintenance

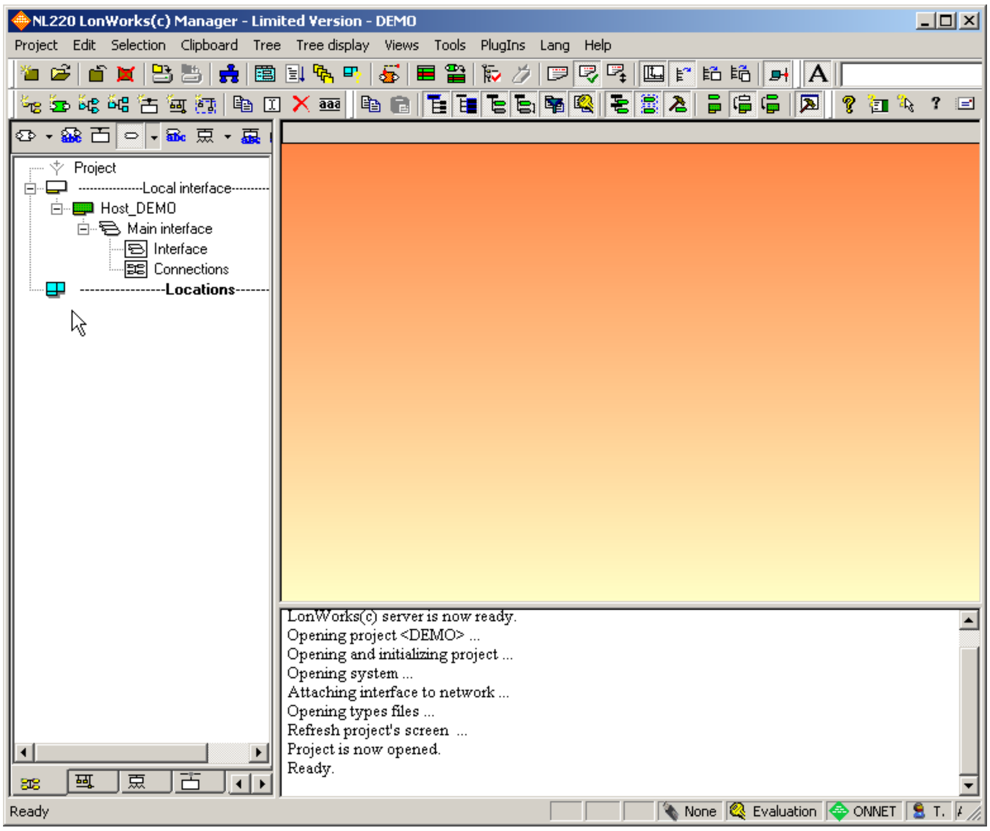
Project's path: C:\NLPrj\DEMO

Network interface: Lon IP Interface

□ Server for remote stations

Fig. 7-19 | Starting the DEMO project

The window of the NL220 is approximately divided into 4 regions. In the upper section below the menu bar are icon buttons for various functions. In the left-hand column is the tree view of the network, which does not contain currently any devices other than the local interface for the NL220. The right-hand window (currently blank) is used by various functions as a display area and the window for displaying status messages is located in the bottom region.



```
LonWorks(c) server is now ready.
Opening project <DEMO> ...
Opening and initializing project ...
Opening system ...
Attaching interface to network ...
Opening types files ...
Refresh project's screen ...
Project is now opened.
Ready.
```

Fig. 7-20 | Program window of the NL220

The next step in our example is to insert the PCD into the NL database. This process is also referred to as commissioning. This process is started by pressing the "Create a new node" button.

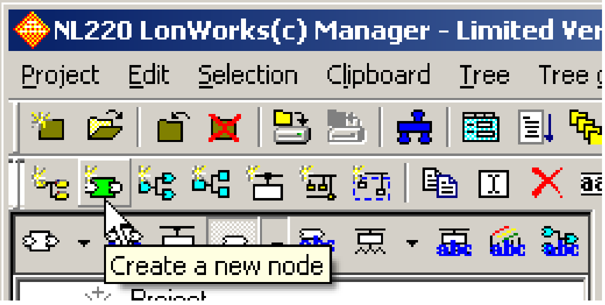
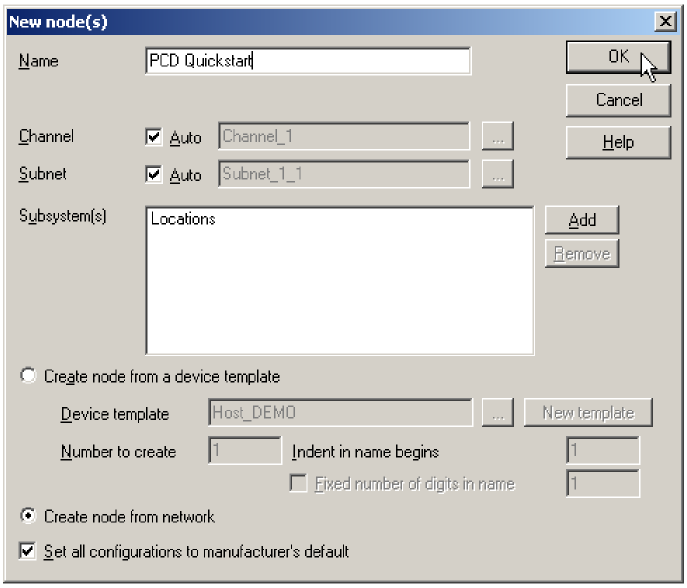


Fig. 7-21 | Creating a new Lon node

In the next window the new node is given a freely definable name, we are calling it "PCD Quickstart". The remaining settings relate to the Lon network and for our example should remain set as can be seen in the following image.



Name: PCD Quickstart

Channel: ☑ Auto Channel_1

Subnet: ☑ Auto Subnet_1_1

Subsystem(s): Locations

○ Create node from a device template

Device template: Host_DEMO

Number to create: 1

Indent in name begins: 1

□ Fixed number of digits in name: 1

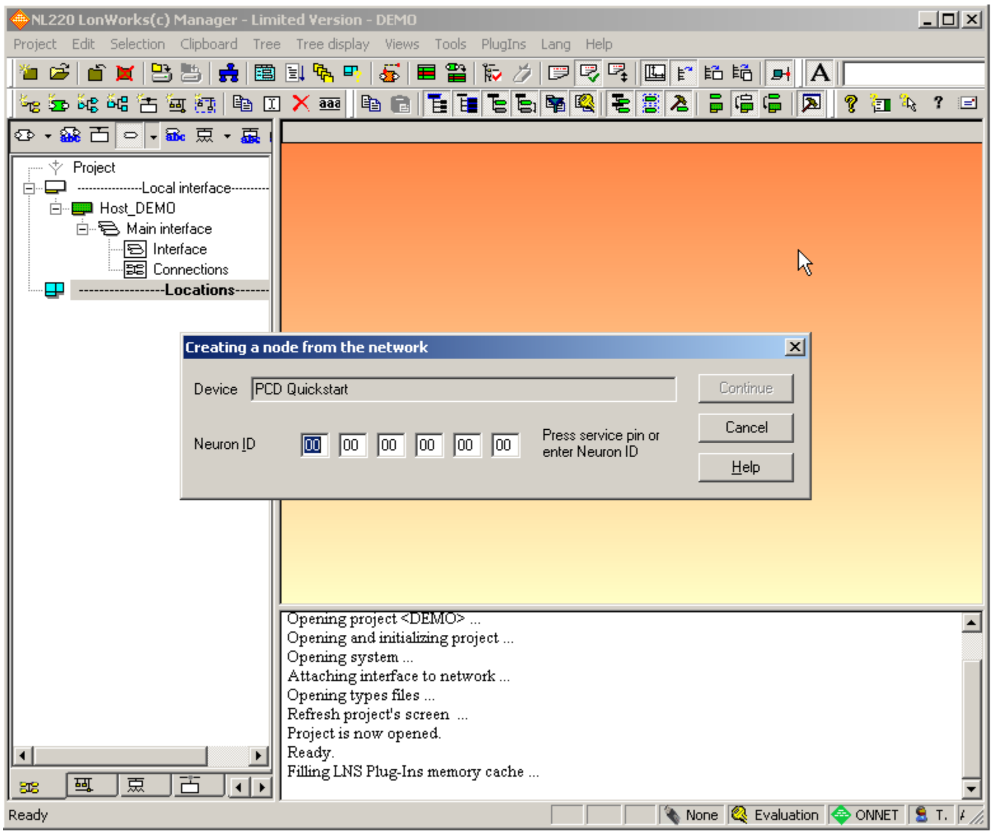
● Create node from network

☑ Set all configurations to manufacturer's default

Fig. 7-22 | Commissioning a Lon node

7

From now on the commissioning process is waiting for receipt of a Neuron ID. This ID is universally unique and characterises a Lon device with a kind of serial number.



Device: PCD Quickstart

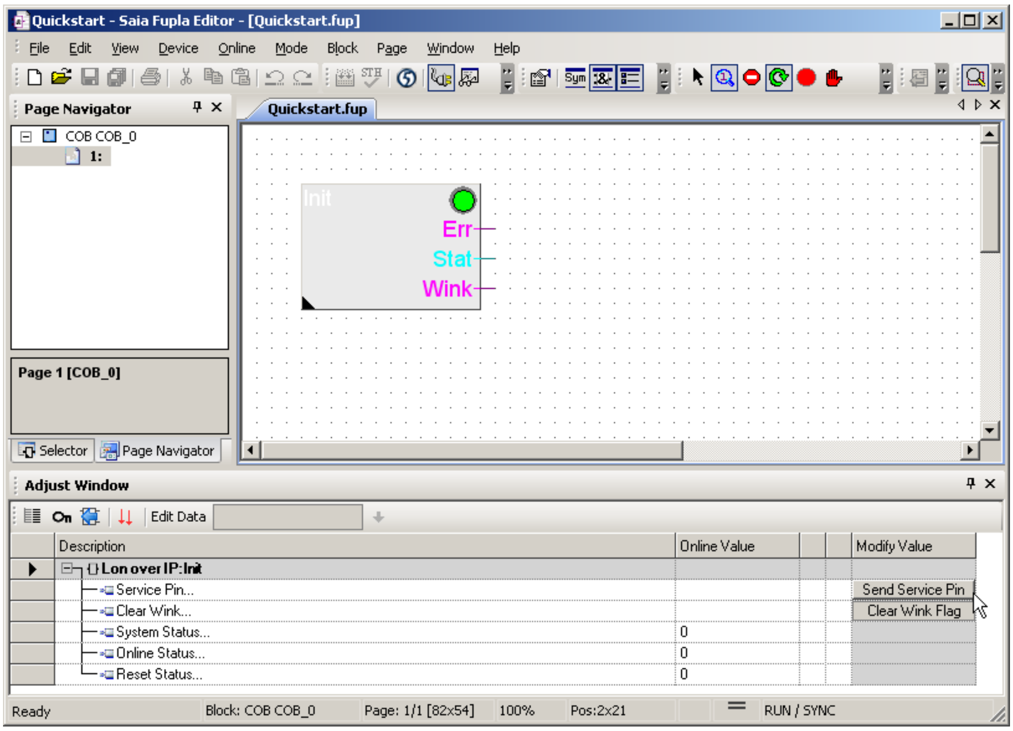
Neuron ID: 00 00 00 00 00 00

Press service pin or enter Neuron ID

```
Opening project <DEMO> ...
Opening and initializing project ...
Opening system ...
Attaching interface to network ...
Opening types files ...
Refresh project's screen ...
Project is now opened.
Ready.
Filling LNS Plug-Ins memory cache ...
```

Fig. 7-23 | NL220 is ready to receive a Neuron ID

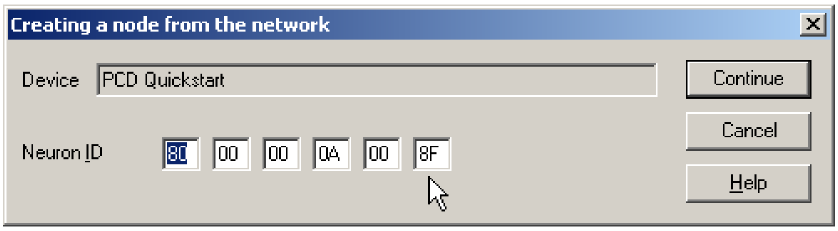
Most Lon field units such as individual room controllers, buttons, etc. have a small button, which upon being pressed triggers the sending out of its own Neuron ID. As the "Lon over IP" solution of the PCD is purely a software implementation, the "Lon over IP Init" FBox prepares the service pin as a software button in the Adjust window. As can be seen in the image below, the Fulpa page with the "Init" FBox can be opened, switches to online mode and opens the Adjust window of the "Init" FBox. Now the "service pin" function can be triggered.



Description	Online Value	Modify Value
Lon over IP:Init		
Service Pin...		Send Service Pin
Clear Wink...		Clear Wink Flag
System Status...	0	
Online Status...	0	
Reset Status...	0	

Fig. 7-24 | Sending the Neuron ID of the PCD using the Service Pin

After activating the service pin function, the NL220 software receives the Neuron ID and show it in the "Create a new node from network" pop-up dialog In real applications at this point it would be important to ensure that the Neuron ID that has been received actually comes from the expected device. Then you confirm the process by pressing "Continue"

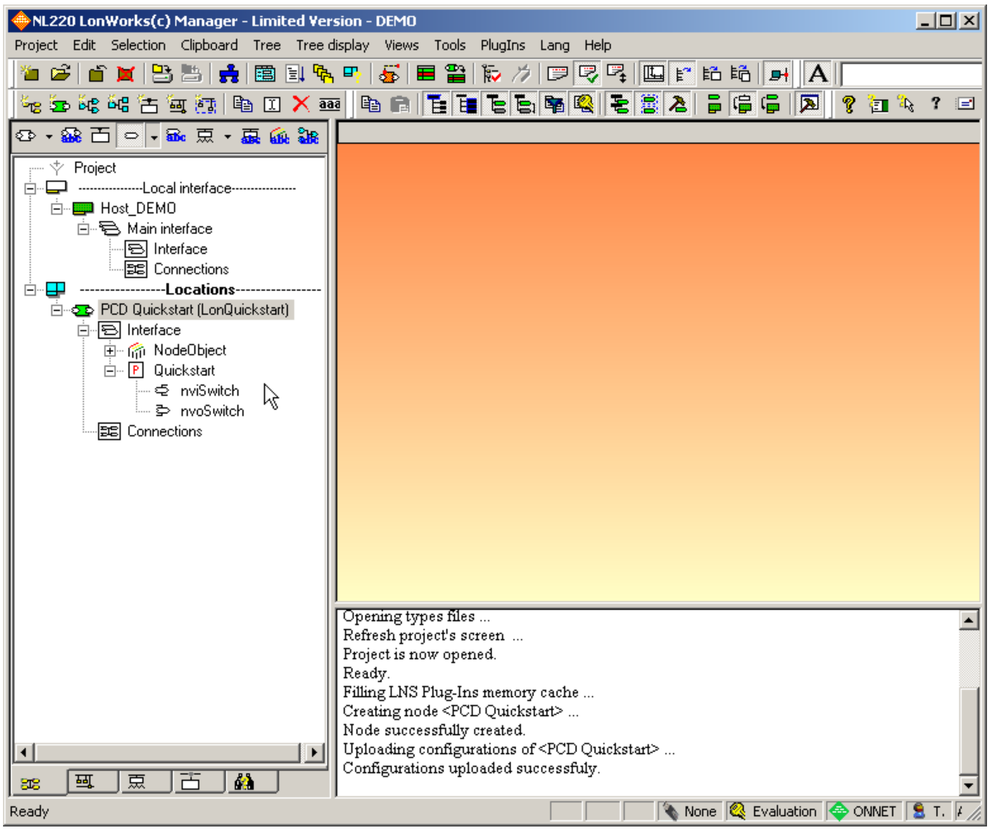


Device: PCD Quickstart

Neuron ID: 80 00 00 0A 00 8F

Fig. 7-25 | Receiving and confirming the Neuron ID

NL220 now loads the Lon node online from the PCD into the LNS database and then displays the PCD as a device in the tree view under "Locations".


NL220 LonWorks(c) Manager - Limited Version - DEMO
Project Edit Selection Clipboard Tree Tree display Views Tools PlugIns Lang Help
Project
Local interface
Host_DEMO
Main interface
Interface
Connections
Locations
PCD Quickstart (LonQuickstart)
Interface
NodeObject
Quickstart
nviSwitch
nvoSwitch
Connections
Ready
None
Evaluation
ONNET

```
Opening types files ...
Refresh project's screen ...
Project is now opened.
Ready.
Filling LNS Plug-Ins memory cache ...
Creating node <PCD Quickstart> ...
Node successfully created.
Uploading configurations of <PCD Quickstart> ...
Configurations uploaded successfuly.
```

Fig. 7-26 | Lon node when commissioning is complete

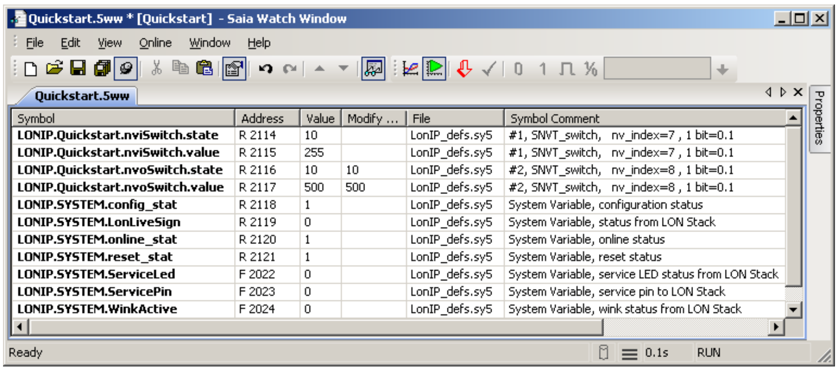
7.3 Online testing Lon nodes

The Lon node of the PCD is not operational and can be tested online. A simple option is available to the PG5 dedicated "WatchWindow" and using the "Variables Browser" in NL220.

It is important to note the direction in which a Lon network variable is always viewed from the respective device interface in the direction of the Lon network. A variable within the PCD Lon node with the prefix "nvo" is sent from the PCD into the Lon network and accordingly variables are received from the network with the identifier "nvi".

LonWorks® does not recognise any bidirectional variables, i.e. a change in value compared to the test must always be carried out on the sending side.

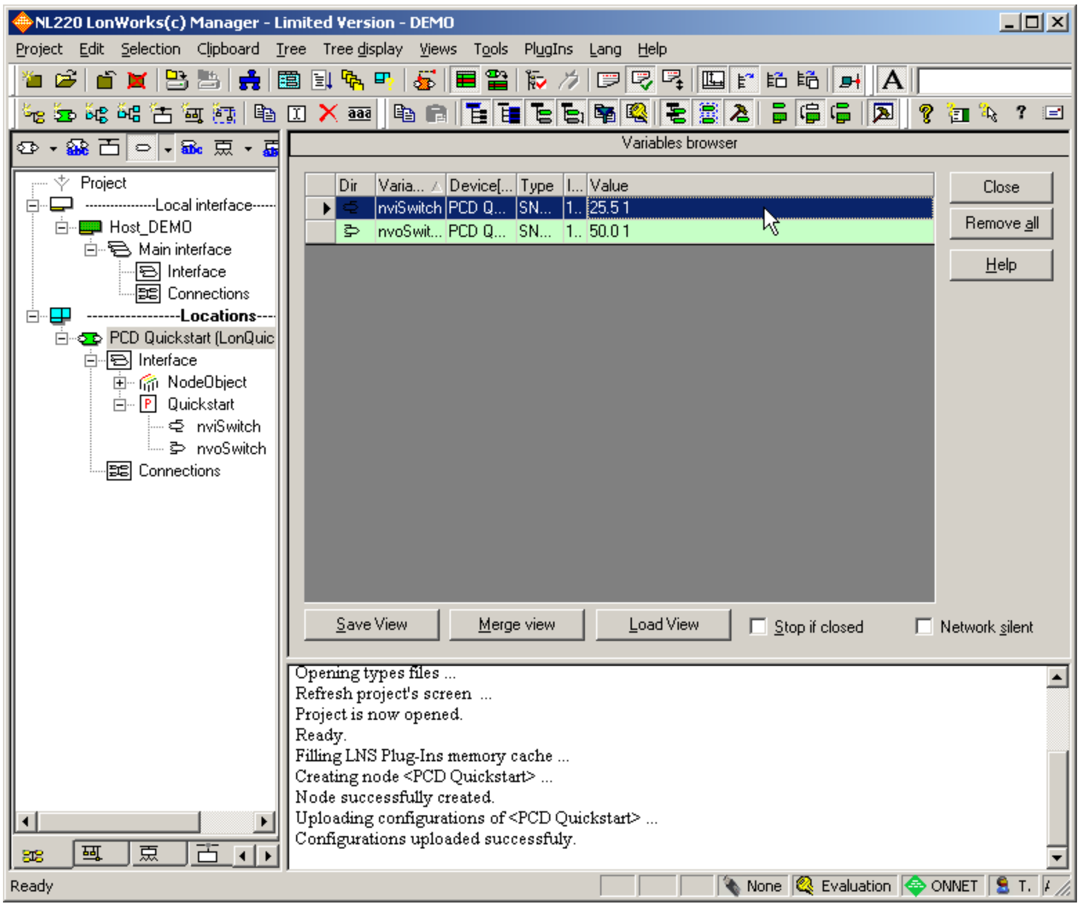
To insert the Lon variables in the PG5 WatchWindow, the complete "LONIP" group can be dragged from the Symbol Browser into the WatchWindow using "drag-and-drop".



Symbol	Address	Value	Modify ...	File	Symbol Comment
LONIP.Quickstart.nviSwitch.state	R 2114	10		LonIP_defs.sy5	#1, SNVT_switch, nv_index=7 , 1 bit=0.1
LONIP.Quickstart.nviSwitch.value	R 2115	255		LonIP_defs.sy5	#1, SNVT_switch, nv_index=7 , 1 bit=0.1
LONIP.Quickstart.nvoSwitch.state	R 2116	10	10	LonIP_defs.sy5	#2, SNVT_switch, nv_index=8 , 1 bit=0.1
LONIP.Quickstart.nvoSwitch.value	R 2117	500	500	LonIP_defs.sy5	#2, SNVT_switch, nv_index=8 , 1 bit=0.1
LONIP.SYSTEM.config_stat	R 2118	1		LonIP_defs.sy5	System Variable, configuration status
LONIP.SYSTEM.LonLiveSign	R 2119	0		LonIP_defs.sy5	System Variable, status from LON Stack
LONIP.SYSTEM.online_stat	R 2120	1		LonIP_defs.sy5	System Variable, online status
LONIP.SYSTEM.reset_stat	R 2121	1		LonIP_defs.sy5	System Variable, reset status
LONIP.SYSTEM.ServiceLed	F 2022	0		LonIP_defs.sy5	System Variable, service LED status from LON Stack
LONIP.SYSTEM.ServicePin	F 2023	0		LonIP_defs.sy5	System Variable, service pin to LON Stack
LONIP.SYSTEM.WinkActive	F 2024	0		LonIP_defs.sy5	System Variable, wink status from LON Stack

Fig. 7-27 | PG5 Watchwindow with LONIP symbols

The Lon variables are inserted into the "Variables Browser" of the NL220 and in a similar way into the profile by "drag-and-drop" into the blank, large window. The "Variables Browser" opens automatically.



Dir	Varia...	Device[...	Type	I...	Value
⇐	nviSwitch	PCD Q...	SN...	1..	25.5 1
⇒	nvoSwit...	PCD Q...	SN...	1..	50.0 1

☐ Stop if closed ☐ Network silent

```
Opening types files ...
Refresh project's screen ...
Project is now opened.
Ready.
Filling LNS Plug-Ins memory cache ...
Creating node <PCD Quickstart> ...
Node successfully created.
Uploading configurations of <PCD Quickstart> ...
Configurations uploaded successfuly.
```

Fig. 7-28 | NL220 with PCD Lon nodes in the Symbol Browser online

As soon as the rows are highlighted light green, a data connection has been established.

Changes in value on the sending side automatically lead to the updating of the value in the other location.

7.4 Generating XIF files

This chapter describes how an XIF file can be generated by a Saia PCD "Lon over IP" node using a plug-in for NL220 or LonMaker.

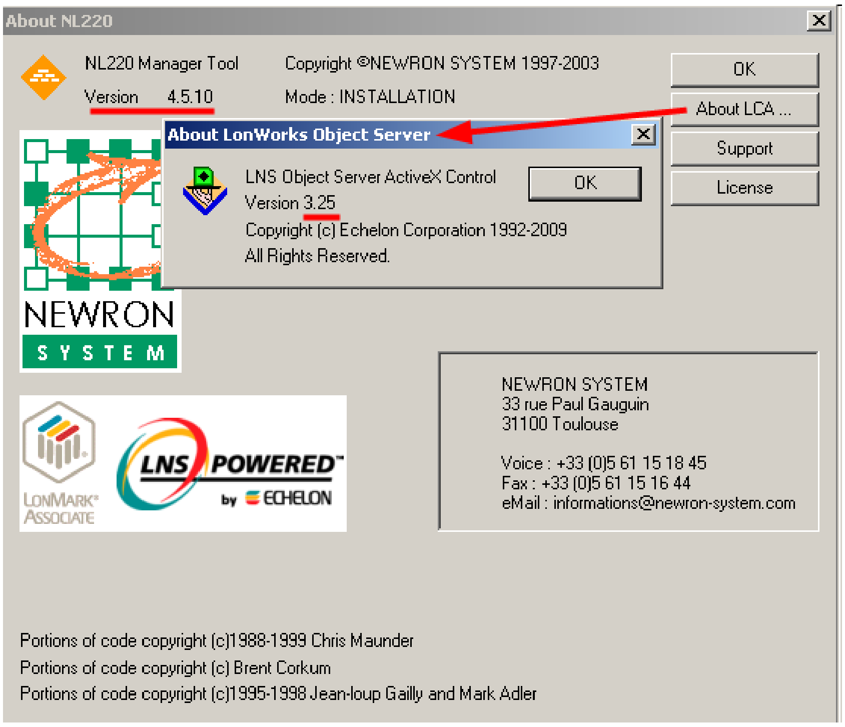
Note
This step is omitted for LonFT nodes. The XIF file is automatically generated and saved in the same directory as the file with the file extension ".lft".

7.4.1 Requirement

7.4.1.1 LonWorks® commissioning software

LonWorks® commissioning software e.g. NL220? or LonMaker
(all further explanations are based on the NL220 software)

NL220 min. Version number 4.5.10 with LCA 3.25 ?



About NL220

NL220 Manager Tool
Version 4.5.10

Copyright ©NEWRON SYSTEM 1997-2003
Mode : INSTALLATION

About LonWorks Object Server

LNS Object Server ActiveX Control
Version 3.25
Copyright (c) Echelon Corporation 1992-2009
All Rights Reserved.

NEWRON SYSTEM
33 rue Paul Gauguin
31100 Toulouse

Voice : +33 (0)5 61 15 18 45
Fax : +33 (0)5 61 15 16 44
eMail : informations@newron-system.com

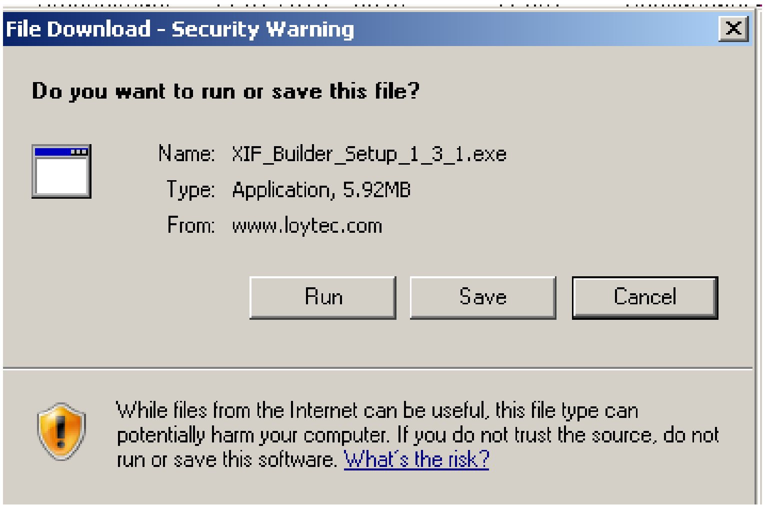
Portions of code copyright (c)1988-1999 Chris Maunder
Portions of code copyright (c) Brent Corkum
Portions of code copyright (c)1995-1998 Jean-loup Gailly and Mark Adler

Fig. 7-29

7.4.1.2 Plug-in installation

XIF-Creator plug-in by the company Loytec

Available as an internet download from the Loytec web site:
http://www.loytec.com



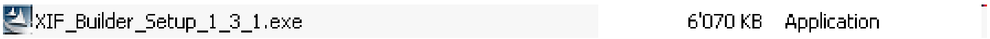
File Download - Security Warning

Do you want to run or save this file?

Name: XIF_Builder_Setup_1_3_1.exe
Type: Application, 5.92MB
From: www.loytec.com

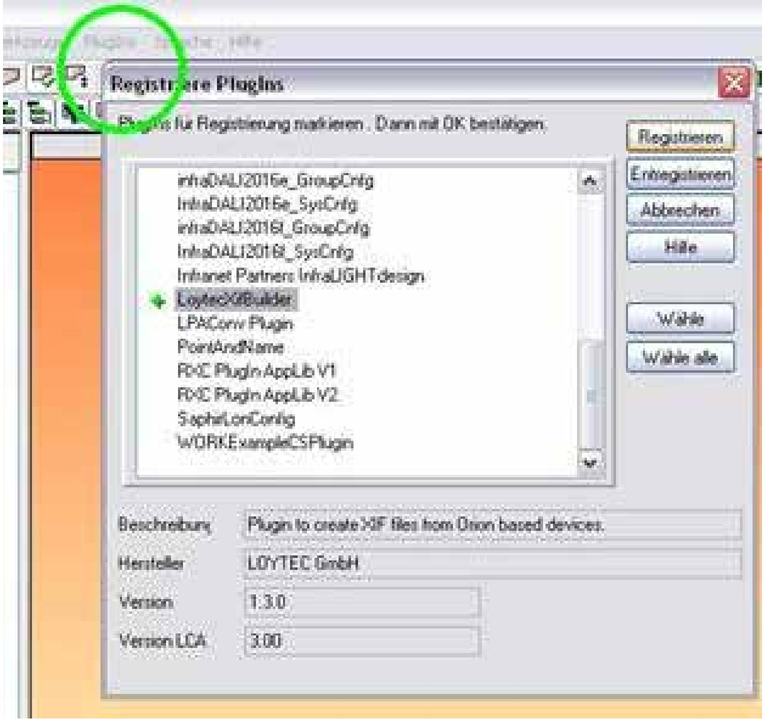
While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

Fig. 7-30



XIF_Builder_Setup_1_3_1.exe 6'070 KB Application

Fig. 7-31



Registriere PlugIns

Plugins für Registrierung markieren . Dann mit OK bestätigen.

infraDALI2016e_GroupCnfg
InfraDALI2016e_SysCnfg
infraDALI2016l_GroupCnfg
InfraDALI2016l_SysCnfg
Infranet Partners InfraLIGHTdesign
LoytecXifBuilder
LPAConv Plugin
PointAndName
RXC PlugIn AppLib V1
RXC PlugIn AppLib V2
SaphirLonConfig
WORKExampleCSPlugin

Beschreibung: Plugin to create XIF files from Orion based devices.
Hersteller: LOYTEC GmbH
Version: 1.3.0
Version LCA: 3.00

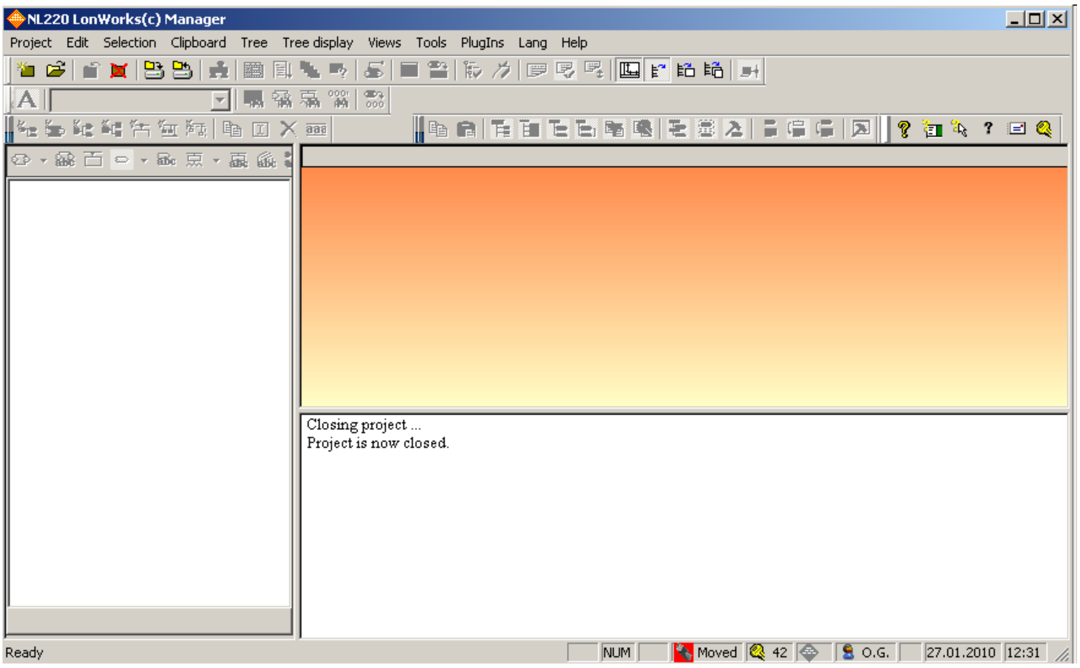
Fig. 7-32

Installation of the plug-in using the XIF_Builder_Setup program

7.4.2 Generating an XIF file

The XIF builder plug-in is launched from the context menu of a Lon node in the NL220 software. To do this the PCD must be commissioned as a Lon node. As bindings are not required, the XIF file can also be generated from an NL220 demo application.

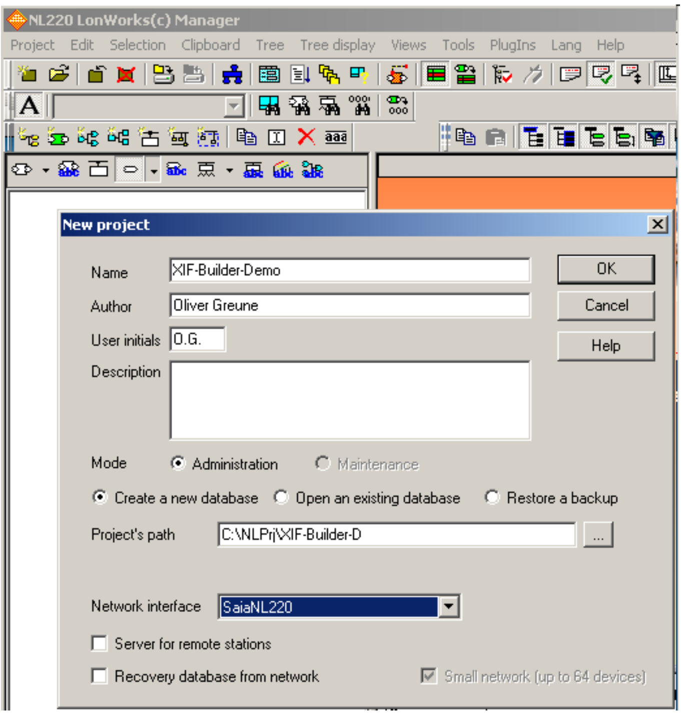
- Downloading the PG5 project to the PCD
- Setting up the IP-852 config server, see chapter 7 for details
- Starting the Lon commissioning software


NL220 LonWorks(c) Manager
Project Edit Selection Clipboard Tree Tree display Views Tools PlugIns Lang Help
Ready
NUM
Moved
42
O.G.
27.01.2010
12:31

```
Closing project ...
Project is now closed.
```

Fig. 7-33

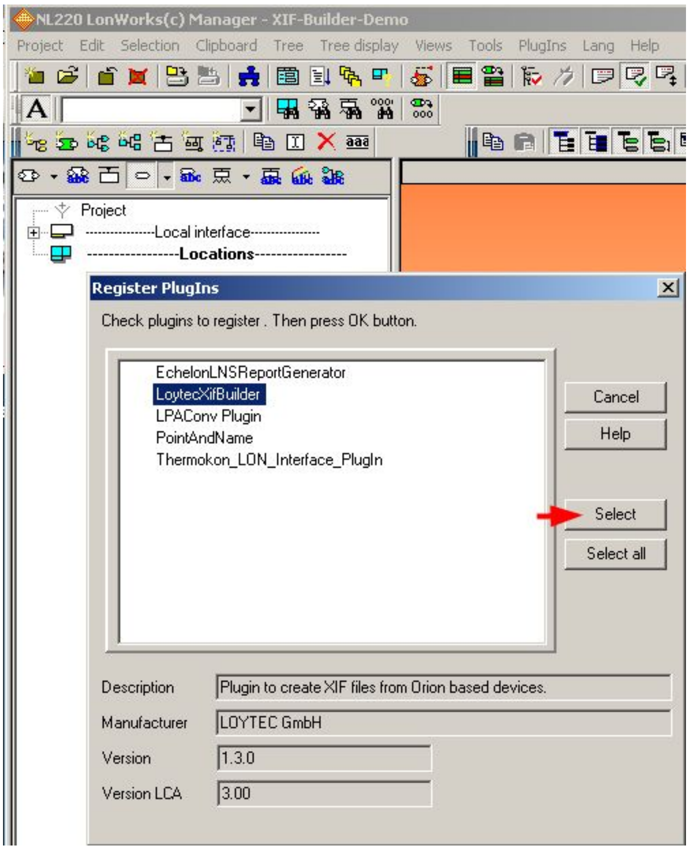
Starting a new project


NL220 LonWorks(c) Manager
Project Edit Selection Clipboard Tree Tree display Views Tools PlugIns Lang Help
New project
OK
Cancel
Help

Name: XIF-Builder-Demo
Author: Oliver Greune
User initials: O.G.
Description:
Mode: ● Administration ○ Maintenance
● Create a new database ○ Open an existing database ○ Restore a backup
Project's path: C:\NLPrj\XIF-Builder-D
Network interface: SaiaNL220
□ Server for remote stations
□ Recovery database from network ☑ Small network (up to 64 devices)

Fig. 7-34

Registering the XIF builder plug-in


NL220 LonWorks(c) Manager - XIF-Builder-Demo
Project Edit Selection Clipboard Tree Tree display Views Tools PlugIns Lang Help
Project
Local interface
Locations
Register PlugIns
Cancel
Help
Select
Select all

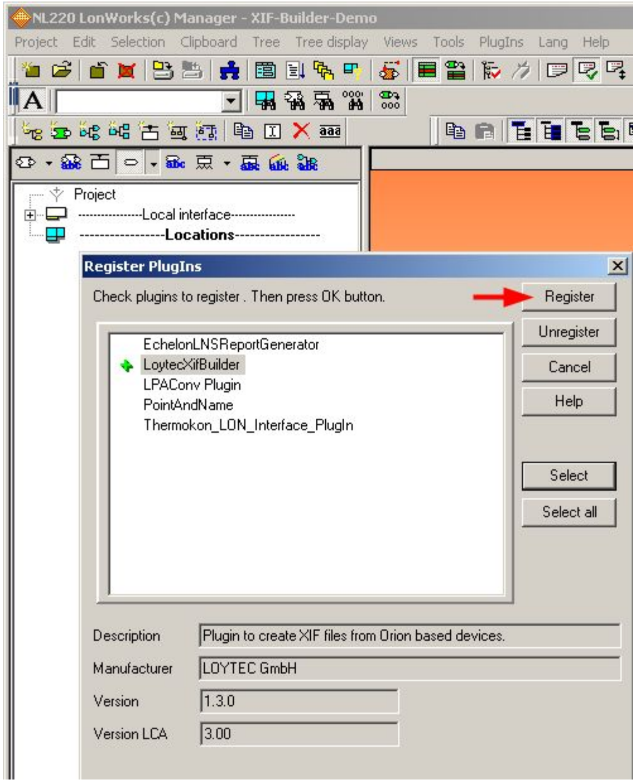
Check plugins to register . Then press OK button.

EchelonLNSReportGenerator
LoytecXifBuilder
LPAConv Plugin
PointAndName
Thermokon_LON_Interface_PlugIn

Description: Plugin to create XIF files from Orion based devices.
Manufacturer: LOYTEC GmbH
Version: 1.3.0
Version LCA: 3.00

Fig. 7-35

Registering the XIF builder plug-in



NL220 LonWorks(c) Manager - XIF-Builder-Demo

Register PlugIns

Check plugins to register . Then press OK button.

- EchelonLNSReportGenerator
- LoytecXifBuilder
- LPAConv Plugin
- PointAndName
- Thermokon_LON_Interface_PlugIn

Description: Plugin to create XIF files from Orion based devices.

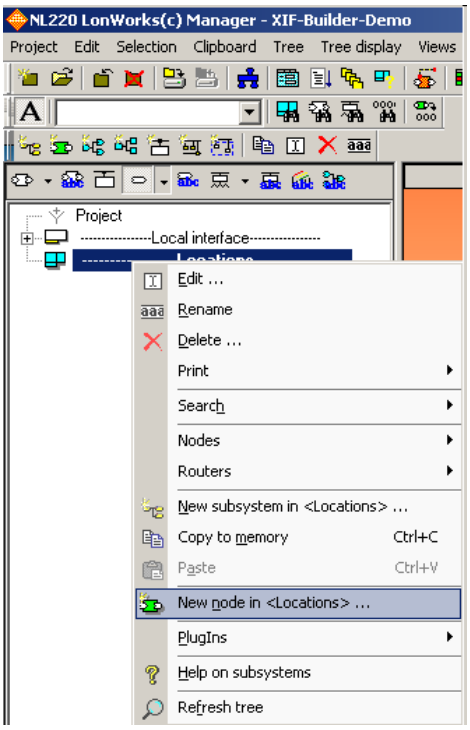
Manufacturer: LOYTEC GmbH

Version: 1.3.0

Version LCA: 3.00

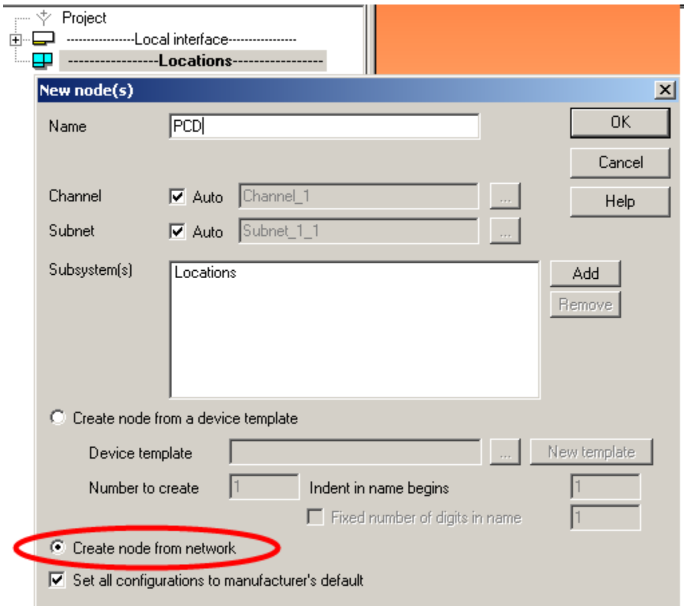
Fig. 7-36

Generating a new node from the network



NL220 LonWorks(c) Manager - XIF-Builder-Demo

Fig. 7-37



Project
---------------Local interface---------------
----------------Locations----------------

New node(s)

Name: PCD

Channel: ☑ Auto Channel_1

Subnet: ☑ Auto Subnet_1_1

Subsystem(s): Locations

○ Create node from a device template

Device template:

Number to create: 1

Indent in name begins: 1

□ Fixed number of digits in name: 1

● Create node from network

☑ Set all configurations to manufacturer's default

OK
Cancel
Help
Add
Remove
New template

Fig. 7-38

7.4.3 Generating XIF files

Open the context menu of the new node

Start the LOYTEC XIF Builder plug-in
(WARNING: switch the connection option to Network

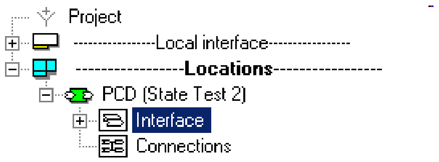

Project
----------------Local interface----------------
-----------------Locations-----------------
PCD (State Test 2)
Interface
Connections

Fig. 7-39

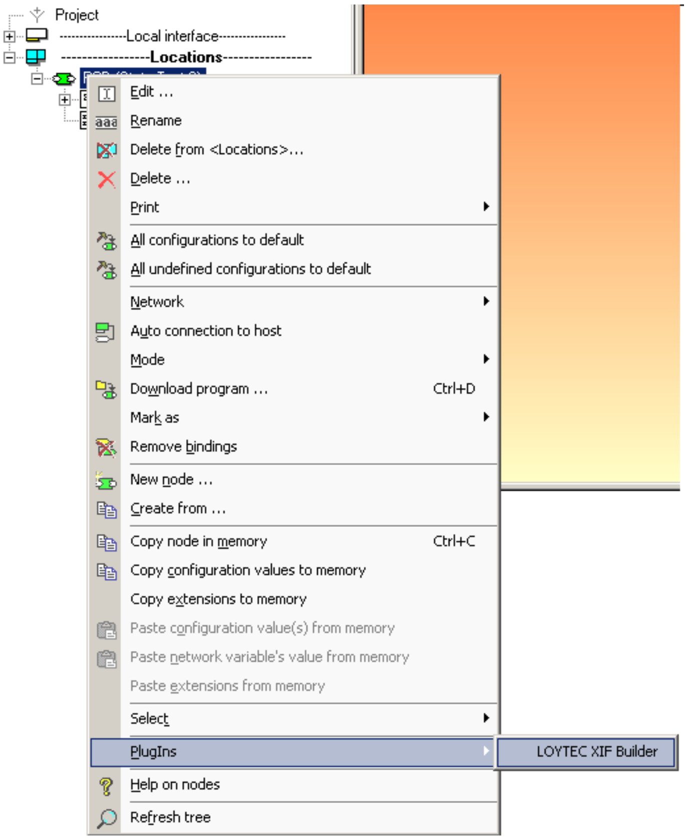
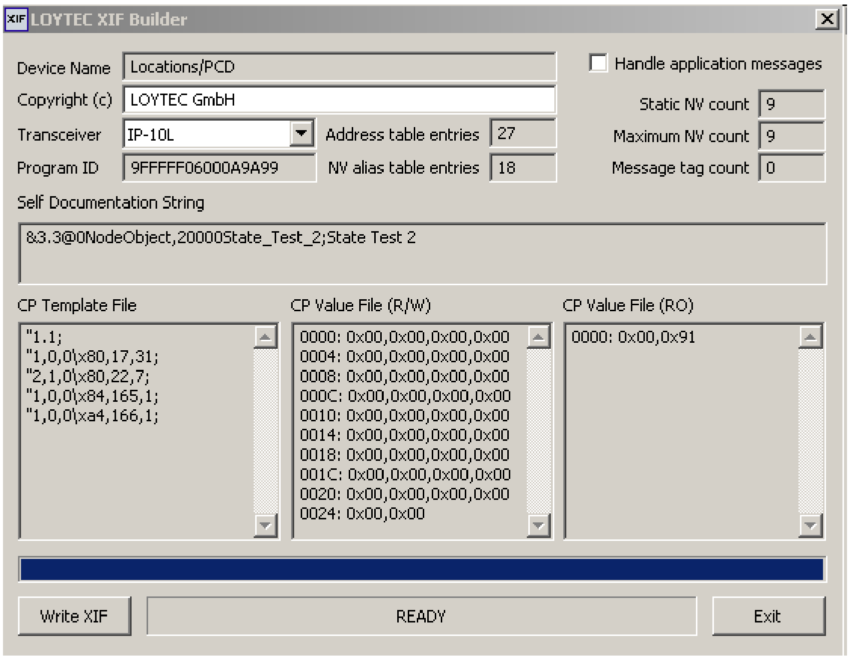

Project
----------------Local interface----------------
-----------------Locations-----------------
Edit ...
Rename
Delete from <Locations>...
Delete ...
Print
All configurations to default
All undefined configurations to default
Network
Auto connection to host
Mode
Download program ... Ctrl+D
Mark as
Remove bindings
New node ...
Create from ...
Copy node in memory Ctrl+C
Copy configuration values to memory
Copy extensions to memory
Paste configuration value(s) from memory
Paste network variable's value from memory
Paste extensions from memory
Select
PlugIns
LOYTEC XIF Builder
Help on nodes
Refresh tree

Fig. 7-40


LOYTEC XIF Builder

Device Name: Locations/PCD

Copyright (c): LOYTEC GmbH

Transceiver: IP-10L

Program ID: 9FFFFF06000A9A99

Address table entries: 27

NV alias table entries: 18

□ Handle application messages

Static NV count: 9

Maximum NV count: 9

Message tag count: 0

Self Documentation String

```
&3.3@0NodeObject,20000State_Test_2;State Test 2
```

CP Template File

```
"1.1;
"1,0,0\x80,17,31;
"2,1,0\x80,22,7;
"1,0,0\x84,165,1;
"1,0,0\xa4,166,1;
```

CP Value File (R/W)

```
0000: 0x00,0x00,0x00,0x00
0004: 0x00,0x00,0x00,0x00
0008: 0x00,0x00,0x00,0x00
000C: 0x00,0x00,0x00,0x00
0010: 0x00,0x00,0x00,0x00
0014: 0x00,0x00,0x00,0x00
0018: 0x00,0x00,0x00,0x00
001C: 0x00,0x00,0x00,0x00
0020: 0x00,0x00,0x00,0x00
0024: 0x00,0x00
```

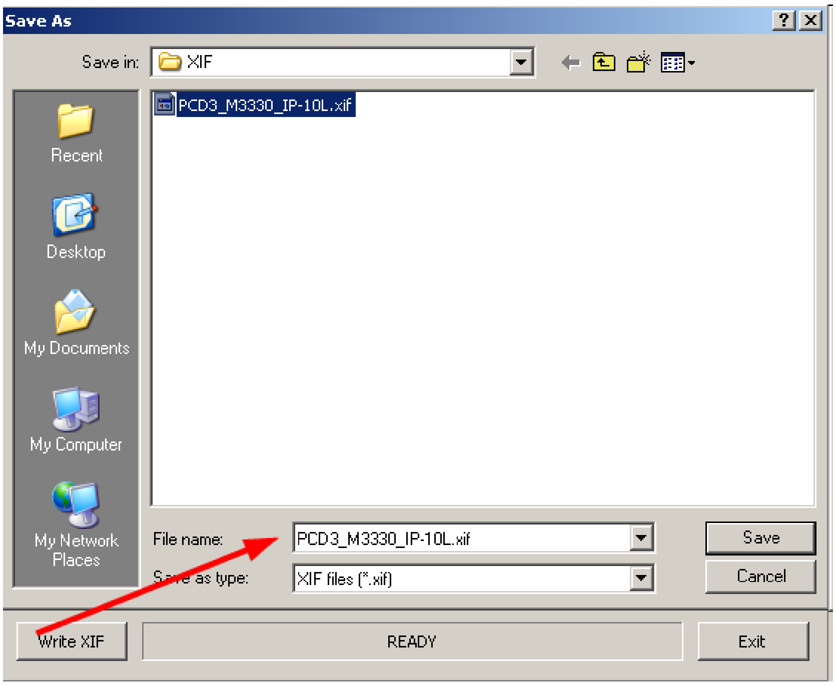
CP Value File (RO)

```
0000: 0x00,0x91
```

READY

Fig. 7-41

Save the XIF file by clicking on the "Write" button


Save As

Save in: XIF

PCD3_M3330_IP-10L.xif

File name: PCD3_M3330_IP-10L.xif

Save as type: XIF files (*.xif)

READY

Fig. 7-42

8 Error Handling

8.1 Lon Life Sign

The LONIP.SYSTEM.LonLiveSign and/or LonFTxx.SYSTEM.LonLiveSign symbol are PCD registers for monitoring Lon Firmware. While the Lon node is initialised by the PCD , the register has a positive value of between 10 and 1. The value 0 displays that the Lon node is operational. Errors, which lead to disabling Lon functionality, are identified with a unique negative value.

8.1.1 System Start

The LonLiveSign register reflects the status when starting the Lon system. The individual steps in initialisation are numbered from 10 to 0.

LonLiveSign	Description
10	The configuration was successfully loaded.
9	The "descriptors" of the Lon firmware have been initialised.
8	CNIP is being initialised
7	Network layer is being initialised
6	SelfDoc string has been created
5	All network variables have been generated
4	The descriptors have been verified
3	The network layer has been started
2	The Lon timers have started, the template and CP files have been initialised. End of the “lon_init()”
1	The connection of network variables to PCD registers/flags has started.
0	Lon function is operational

Table 8-1 | LonLiveSign

8.1.2 "In Run"

As soon as the system has started, the LonLiveSign can be used to determine whether Lon functionality is active. The “LiveSign” register is monitored together with all network variables for changes. In addition any value, for instance 500, can be written by the user program into the register. If Lon functionality is working correctly, the value is automatically reset to 0 by the Lon firmware after a few seconds

8.2 History of error numbers

Most of the following errors represent a problem during system start-up. The errors are written to the PCD history. Depending on the error the affected module is not started. If an error occurs after start-up, the error code is also written as a negative number into the LONFT1x0.SYSTEM.LonLiveSign register.

Format Definition	Example	Description
LonFT E%4d	LonFT E1003	A LONFT_FAIL_CFG_READ error has occurred. This error cannot be assigned to a port.
LonFT P%i E%4d	LonFT P120 E2268	A LONFT_FAIL_LON_PORT_NOT_READY error for port 120 (slot 2) has occurred.
LonFT P%i msg repeated	LonFT P130 msg repeated	The previous error for port 130 was repeated twice or more frequently.
LonFT last msg repeated	LonFT last msg repeated	The previous error was repeated twice or more frequently. This error cannot be assigned to a port.
LonFT P%i 2 msgs repeated	LonFT P120 2 msgs repeated	The last two errors for port 120 were repeated twice or more frequently.
LonFT last 2 msgs repeated	LonFT last 2 msgs repeated	The last two errors were repeated twice or more frequently. This error cannot be assigned to a port.
LonFT rc=%i	LonFT rc=…	Return value (error code) of a function, which encountered the last LonFT... error.
LonFT i=%i	LonFT i=…	Index (of a network variable or of a configuration parameter, ...) to which the last LonFT... error relates.
1009	CFG_LMTYPE	ConfigFile section verification fails.
1010	CFG_LMUNION	
1011	CFG_LMELEMENT	
1012	CFG_NVTABLEMAP	
1013	CFG_NVMAPPING	
1014	CFG_CPMAPPING	
1015	CFG_NVDEFAULTS	
1016	CFG_CPVALDEF	
1017	CFG_CPRODEF	
1018	CFG_NODEDESC	
1019	MISSING_NODEID	The NodeId in the ConfigFile and the NodeId in the EEPROM are both invalid (0 or -1)
1020	LON_INIT	Unspecified error during the lon_init() function. Most likely it will appear together with a more specific history entry.

Tab. 8-2

8.2.1 Error when loading the configuration file

This section contains errors, which can occur when checking the configuration. Otherwise, other system errors are included.

#	Enumerator "LONFT_FAIL_" for LonFT or "LONIP_FAIL_" for LonIP	Description
1000	MALLOC	Error when reserving memory (system error)
1001	NO_CONFIG	No configuration file present
1002	CFG_MALLOC	Memory space for configuration file could not be reserved.
1003	CFG_READ	Configuration file could not be read.
1004	CFG_SIZE	The size of the configuration file does not match the declaration in the NodeDescriptor.
1005	CFG_FPDESC	Checking of the corresponding section of the configuration file failed
1007	CFG_NVDESC	
1008	CFG_CPDESC	
1009	CFG_LMTYPE	
1010	CFG_LMUNION	
1011	CFG_LMELEMENT	
1012	CFG_NVTABLEMAP	
1013	CFG_NVMAPPING	
1014	CFG_CPMAPPING	
1015	CFG_NVDEFAULTS	
1016	CFG_CPVALDEF	
1017	CFG_CPRODEF	
1018	CFG_NODEDESC	
1019	MISSING_NODEID	The node identification (nodeID) in the configuration file and in EEPROM are both invalid (0 or -1)
1020	LON_INIT	Unknown error during execution of the function lon_init(). This error usually occurs together with a more specific error.
1021	PCDMAPPER_INIT	The initialisation of the PCD mapper failed.
1022	CFG_NETSET	Checking of a section in the configuration file failed.
1024	GET_IP	The IP address of the PCD could not be determined (system error)
1025	PCDMAPPER	The PCD mapper was interrupted by an error.
1030	WRONG_COMPILER_ VERSION	In each case this error occurs in conjunction with another error. The compiler version does not match the firmware version.

1031	CFG_NODEDESC_CMP	Error when reading or writing to AdminFile.
1032	CFG_SI_DATA	Checking of a section in the configuration file failed.
1033	CFG_AI_DATA	Checking of a section in the configuration file failed.
1036	CONFIG_NUM_NV	Corrupt configuration file. The number of network variables is conflicting.
1040	CFG_SPECIAL_MAP	Special mappings (SYSTEM symbols) could not be initialised.
1050	PORT_ALREADY_USED	Two configuration files are using the same port.
1051	PORT_NOT_SUPPORTED	Invalid port number. Only 100, 110, 120 or 130 are valid.

Tab. 8-3

8.2.2 System errors

#	Enumerator "LONFT_FAIL_" for LonFT or "LONIP_FAIL_" for LonIP	Description
1200	LOG_OPEN	Not used as no logging system is available.
1301	FLASH	Error in the Saia PCD COSinus module initialisation
1302	HOOKER	
1303	KRNMALLOC	
1304	MAINTASK	
1305	GETNEURON	
1400	IPREGISTER	

Tab. 8-4

8.2.3 Error in the Lon initialisation

These errors were determined when creating the internal data structure. They may relate to the following areas: Configuration file, CNIP initialisation, Orion initialisation and LonMark timer.

#	**Enumerator "LONFT_FAIL_" for LonFT or "LONIP_FAIL_" for LonIP**	**Description**
2000	LON_INIT_DESCRIPTORS	Unknown error in the descriptors in the configuration file. This error usually occurs together with a more specific error.
2001	LON_INIT_D_NV_COUNT	Checking the number of network variables in the configuration returned a conflict.
2002	LON_INIT_D_NV_DEF	The network variables could not be pre-initialised (defaults) because there is a conflict in the size of the required data.
2009	LON_INIT_D_CP_MISMATCH	The number of defined "CpDesc" conflicts with the number of "CpDesc" according to the declaration in the "FpDesc".
2010	LON_INIT_D_CP_COUNT	No configuration parameters have been defined in the "FpDesc".
2012	LON_INIT_D_CP_VALDEF	The variable configuration parameters could not be pre-initialised (defaults) because there is a conflict in the size of the required data. Instead they defaulted to 0.
2013	LON_INIT_D_CP_RODEF	The constant configuration parameters could not be pre-initialised (defaults) because there is a conflict in the size of the required data. Instead they defaulted to 0.
2100	LON_INIT_CNIP	Error when starting the CNIP interface
2110	LON_INIT_CNIP_CCRTR	
2120	LON_INIT_CNIP_CBREG	
2130	LON_INIT_CNIP_GET	
2140	LON_INIT_CNIP_SET	
2150	LON_INIT_CNIP_WAITING	The CNIP configuration server could not be reached within 30 seconds. Further attempts at communication will be continued. The network settings of the LonIP node need to be checked and the L-IP device should be restarted, if required.
2200	LON_INIT_O_MUTEX	Error when creating an Orion Mutex.
2201	LON_INIT_O_INTASK_C	Error when creating TaskInput OssiThread.
2202	LON_INIT_O_INTASK_S	Error when starting TaskInput OssiThread.

2210	LON_INIT_ORION	Error when initialising the Orion Stack.
2220	LON_INIT_ORION_INIT	The OrionInit() function reported an error.
2230	LON_INIT_ORION_ GETXCVR	The OrionGetXcvr() function reported an error.
2240	LON_INIT_ORION_ OPENLC	The OrionOpenLc() function reported an error.
2250	LON_INIT_ORION_CF-GIF	The OrionInit() function reported an error.
2251	LON_INIT_ORION_CF-GIF_TIMEOUT	The OrionConfigInterface() function reported a time-out. The loaded configuration is too large for the currently set time monitoring.
2255	LON_INIT_ORION_ SETSD	The OrionSetSdString() function reported an error.
2260	LON_INIT_CREATE_ NV	It is assumed that the configuration of the network variables has changed from the previously loaded configuration. The program identification (program ID) and the XIF version have, however, remained the same. Please change the XIF version.
2265	LON_INIT_CHK_ DESCS	Checking the descriptor in the lon_init() function failed.
2267	LON_INIT_USPI	The LonFT_InitUSPI() function reported an error. The exact error code can be found the in next history entry. Communication between PCD and Lon module could not be established. Possible reasons are: Incorrect slot, incorrect module, incorrect module software. The slot is ignored.
2268	LON_PORT_NOT_ READY	Time-out (1 second) when initialising the Lon module. Communication between PCD and Lon module could not be established. Possible reasons are: Incorrect slot, incorrect module, incorrect module software. The slot is ignored.
2270	LON_INIT_ORION_ START	The Orion Stack could not be started.
2275	LON_INIT_ORION_ SETNM	The display name for the application could not be set.
2280	LON_INIT_LTIMERS	The LTimer could not be started.
2285	LON_INIT_C_TMPL_ FILE	The Template file could not be started.
2290	LON_INIT_LM_FS	The Lon file system could not be initialised.
2295	LON_INIT_LMFTP	The LonWorks® file transfer could not be started.

Tab. 8–5

8.2.4 Communication error

Error messages relating to communication with the ARM microcontroller or the FT5000 chip.

#	Enumerator "LONFT_FAIL_" for LonFT or "LONIP_FAIL_" for LonIP	Description
3000	USPI_TX_ACCESS	Illegal access to the send buffer
3001	USPI_CONNECTION_LOST	USPI communication test failed. A test request was not answered by the ARM microcontroller. No action to resolve the situation has been taken.
3002	STATUS_FT5000_TIMEOUT	FT5000 status request failed. The status request was not answered. No action to resolve the situation has been taken.
3003	INVALID_MESSAGE_TYPE	Unknown message received. The error code (pMsgIn->cmd) is outlined in the next history entry.
3004	USPI_ERROR_CALL-BACK	The callback function for USPI errors was invoked. The error code is outlined in the next history entry.
3100	SEND_NV	Function call send_nv() failed. The error code (negative number) and the index of the network variables can be found in the next history entries. Currently switched off.

Tab. 8-6

8.2.5 Additional information about communication errors

Additional information relating to communication with the ARM microcontroller or the FT5000 chip.

#	Enumerator LONFT_DEBUG_.	Description
5000	ERRORCNTR_tooManyMsgFromPCD	ARM microcontroller is receiving too many messages from the PCD → Messages are being rejected
5001	ERRORCNTR_noBuffersToPCD	ARM microcontroller is not able to send messages to the PCD → Messages are being rejected
5002	ERRORCNTR_noBuffersToFT5000	ARM microcontroller is not able to send messages to the FT5000 chip → Messages are being rejected
5003	ERRORCNTR_shortStackError	Other errors in the ShortStack software
5004	ERRORCNTR_otherError	Other problem, e.g. failed access to EEPROM
5100	FT5000_WATCHDOG_RESET	FT5000 chip was restarted by a watchdog reset

Tab. 8-7

8.3 LON compiler errors and warnings

The compiler error messages and warnings have the following format:
"LonCompiler: Error/Warning <Number>: <Filename> <Position>: <Message>"

LonCompiler	This part of the message is always the same and identifies the LonCompiler as source of the message
Error/Warning <Number>	Either "Error" for error messages or "Warning" for warnings followed by a number. The number uniquely identifies the message.
<Filename>	Normally either in the "MyConfig.lft" or "MyOtherConfig.lip" type.
<Position>:	Position is any useful information to describe the source of the problem. Examples **<LmElement>[17]/<bitfield>/<array>** for XML node **refrigDisplayC.nvoActuatorValue** for symbol names **0** or (blank) if the problem is not specifically for a particular entry. This may point to a missing or incorrect general setting (e.g. program identification).
<Message>	Actual plain text message

Tab. 8-8

The Lon compiler distinguishes between the following sources and causes of errors and warnings:

Source of the problem	Number range	Notes, additional information
Internal	5000..5019	
Incorrect data	5020..5119	
LDRF (LonMark Resource File API)	5120..5169	http://www.lonmark.org/technical_ resources/ guidelines/docs/ LmRfApi04.pdf
ShortStack NCC (Neuron C Compiler Errors)	5170..5299	http://www.Echelon.com/support/ documentation/ manuals/ devtools/078-0402-01B_Neuron_ Tools_Errors_Guide.pdf
ShortStack LID (LonTalk Interface Developer Errors)	5300..5999	http://www.Echelon.com/support/ documentation/ manuals/ devtools/078-0402-01B_Neuron_ Tools_Errors_Guide.pdf

Tab. 8-9

8.3.1 General internal errors and warnings

Usually other messages are present, which point to the actual problem.

Error/Warning	Number	Message
Error	5000	Unknown error *Unknown error*
Warning	5001	Unknown warning *Unknown warning*
Error	5002	Unknown internal error *Unknown internal error*
Error	5003	Failed to create file *Failed to create file*
Error	5004	LonCompiler not properly installed *LonCompiler not properly installed*
Error	5005	Exception while executing ShortStack Compiler *Exception while executing ShortStack Compiler*
Warning	5006	Access to ResourceFiles failed. Close other Lon tools and try again *Access to ResourceFiles failed. Close other Lon tools and try again.*

Tab. 8-10

8.3.2 Errors and warnings caused by incorrect data

The symbols {0}, {1} and {2} are just placeholders, which are replaced in the case of real error messages with values or names.

Error/ Warning	Number	Message
Error	5020	File not found *File not found*
Error	5021	Failed to open file *Failed to open file*
Error	5022	Validation of XML file with XML Schema failed *Validation of XML file with XML Schema failed*
Error	5023	XML file contains <NodeDescriptor> and <NodeDescriptorLonFT> *XML file contains <NodeDescriptor> and <NodeDescriptorLonFT>*
Error	5024	XML file does not contain <NodeDescriptor> nor <NodeDescriptorLonFT> *XML file does not contain <NodeDescriptor> nor <NodeDescriptorLonFT>*
Error	5025	Wrong xmlVersion for this LonCompiler *Wrong xmlVersion for this LonCompiler*
Error	5026	Invalid format *Invalid format*
Error	5027	Failed to parse hexadecimal value "{0}" *Failed to parse hexadecimal value "{0}"*
Error	5028	Failed to parse decimal value "{0}" *Failed to parse decimal value "{0}"*
Error	5029	Invalid LonIP NetSettings *Invalid LonIP NetSettings*
Error	5030	Number of Network Variables is limited to 254, but {0} Network Variables are defined *Number of Network Variables is limited to 254, but {0} Network Variables are defined*
Error	5031	Only "input" or "output" allowed *Only "input" or "output" allowed*
Error	5032	Only "yes" or "no" allowed *Only "yes" or "no" allowed*
Error	5033	Bitfield width is not within 1..8 *Bitfield width is not within 1..8*
Error	5034	Invalid array size *Invalid array size*
Error	5035	Type not supported *Type not supported*
Error	5036	Unknown LmElement *Unknown LmElement*

Error/ Warning	Number	Message
Error	5037	NV with changeable type is not supported *NV with changeable type is not supported*
Error	5038	Element is missing or empty *Element is missing or empty*
Error	5039	Attribute is missing or empty *Attribute is missing or empty*
Error	5040	{0} is not supported *{0} is not supported*
Error	5041	Array mapping not supported for this type *Array mapping not supported for this type*
Error	5042	Does not contain an element with index {0} *Does not contain an element with index {0}*
Error	5043	Insufficient PCD information *Insufficient PCD information*
Error	5044	Element index is out of range *Element index is out of range*
Error	5045	Invalid LmUnion union reference *Invalid LmUnion union reference*
Error	5046	All {0} entries with the same <Index> in {1} must be declared contiguously! *All {0} entries with the same <Index> in {1} must be declared contiguously!*
Error	5047	<ElementIndex> must be in strictly ascending order for a given <Index>! *<ElementIndex> must be in strictly ascending order for a given <Index>!*
Error	5048	Array mapping is only supported for elements which are declared as arrays. Only one element is mapped. *Array mapping is only supported for elements which are declared as arrays. Only one element is mapped.*
Error	5049	Array mapping out of range. Only {0} element(s) mapped *Array mapping out of range. Only {0} element(s) mapped.*
Error	5050	Array size is out of range (255) *Array size is out of range (255)*
Error	5051	Invalid value ({0}) *Invalid value ({0})*

Error	5052	Cannot find iCpDesc for iCp={0}
		Cannot find iCpDesc for iCp={0}

Error	5053	Cannot find iNvDesc for iNv={0}
		Cannot find iNvDesc for iNv={0}

Error	5054	Cannot find referenced NV {0}
		Cannot find referenced NV {0}

Error/ Warning	Number	Message
Error	5055	Cannot handle precedence in following expression {0} *Cannot handle precedence in following expression {0}*
Error	5056	Unknown operator in following expression {0} *Unknown operator in following expression {0}*

Error	5057	Usage of invalid LonMark NV Type with index {0}
		Use of invalid LonMark NXType with index {0}

Error	5058	Usage of invalid LonMark CP Type with index {0}
		Usage of invalid LonMark CP Type with index {0}

Error	5059	Not enough LmType entries *Not enough LmType entries*

Error	5060	Invalid value in LmUnion[{0}], element[{1}]: {2}
		Invalid value in LmUnion[{0}], element[{1}]: {2}

Error	5061	Applies to '{0}' from '{1}' is invalid *Applies to '{0}' from '{1}' is invalid*
Error	5062	No Support for self-installation *No Support for self-installation*
Error	5063	union-id has no array size *union-id has no array size*
Error	5064	Size of ValueFile is {0} bytes, but {1} bytes expected *Size of ValueFile is {0} bytes, but {1} bytes expected*
Error	5065	Size of ReadOnlyFile is {0} bytes, but {1} bytes expected *Size of ReadOnlyFile is {0} bytes, but {1} bytes expected*
Error	5066	Reference to non existent {0} *Reference to non existent {0}*
Error	5067	Wrong type for type-inheriting CP {0} applying to {1} *Wrong type for type-inheriting CP {0} applying to {1}*
Error	5068	There is no principal NV defined *There is no principal NV defined*
Warning	5069	More than one input file of the same type *More than one input file of the same type*
Warning	5070	Unknown command line argument "{0}" *Unknown command line argument "{0}"*
Warning	5071	LonIP XML file with extension "lft" *LonIP XML file with extension "lft"*
Warning	5072	LonFT XML file with extension "lip" *LonFT XML file with extension "lip"*
Warning	5073	Invalid index number for element *Invalid index number for element*
Warning	5074	Failed to parse decimal value *Failed to parse decimal value*

Error/ Warning	Number	Message
Warning	5075	Name {0} was truncated to 16 characters *Name {0} was truncated to 16 characters*
Warning	5076	Invalid characters *Invalid characters*
Warning	5077	{0} does not exist in PCD file *{0} does not exist in PCD file*
Warning	5078	Define either <symbol> or <PcdDataType> and <PcdMediaPointer> *Define either <symbol> or <PcdDataType> and <PcdMediaPointer>*
Warning	5079	Unknown <PcdDataType> *Unknown <PcdDataType>*
Warning	5080	Name string is missing. Replaced by {0} *Name string is missing. Replaced by {0}*
Warning	5081	Inconsistent type definition for NV with the name {0}: {1} vs. {2} *Inconsistent type definition for NV with the name {0}: {1} vs. {2}*
Warning	5082	Inconsistent type definition for CP with the name {0}: {1} vs. {2} *Inconsistent type definition for CP with the name {0}: {1} vs. {2}*
Warning	5083	No definition found for the SNVT "{0}" (Number: {1}) in the installed STANDARD Resource Files *No definition found for the SNVT "{0}" (Number: {1}) in the LonMark STANDARD Resource files*
Warning	5084	No definition found for the SCPT "{0}" (Number: {1}) in the installed STANDARD Resource Files *No definition found for the SCPT "{0}" (Number: {1}) in the LonMark STANDARD Resource files*
Warning	5085	<scope> must be in the range of 0..6 *<scope> must be in the range of 0..6*
Warning	5086	Arrays of Bitfields are not supported *Arrays of Bitfields are not supported*
Error	5087	The ProgramId {0} is already used by {1} *The ProgramId {0} is already used by {1}*
Warning	5088	The ProgramId {0} is already used by {1}. This is only allowed if the Lon configurations are equal *The ProgramId {0} is already used by {1}. This is only allowed if the Lon configurations are equal.*

Tab. 8-11

A Appendix

A.1 Icons

i	This symbol refers to additional information, which is available in this or another manual or in technical documentation on this subject. There are not direct references to such documents.
!	This symbol designates instructions, which need to be strictly followed.
⚡	This symbol warns the reader that components may be damaged as a result of electrostatic discharge when touched. Recommendation: as a minimum touch the negative terminal of the system (PGU connector housing) before coming into contact with the electronic components. Better still is to wear an earthed strap on your wrist, which is connected with the negative terminal of the system.
Classic	Explanations next to this symbol are only valid for the Saia PCD Classic series.
xx7	Explanations next to this symbol are only valid for the Saia PCD xx7 series.

A.2 Terms

3120
NEURON-Chip 3120. MOTOROLA / TOSHIBA chip with internal EEPROM, RAM and integrated LON interface for network communication on OSI layer 7.

3150
NEURON-Chip 3150. MOTOROLA / TOSHIBA chip with internal EPROM, external EEPROM and integrated LON interface for network communication on OSI layer 7.

Address table
A table in a Neuron chip, which defines the group membership of a node and the send address of a linked network variable. 15 different address tables can be defined on one Neuron chip.

Network variable alias
A secondary location in a network variable table, which references a "primary netvar". A network variable alias is addressed in parallel to the primary NV and supports multiple binding of data (e.g. reset Kdo via group address, normal Kdos via subnet/node address).

Application Image
The application program, which is able to run on a Neuron chip.

Application Layer
Transport layer, which ensures application level compatibility. See also under OSI layer 1-7.

Application message
An explicit message with a message code between 0x00 and 0x3e (62d). Interpreting the code is left to the application.

Binder
A software tool, which is able to bind network variables or msg_tags.

Binding
The process, which defines the binding between nodes.

Bridge
Router with two NEURON chips, which displays the messages from a max. of 2 domains on both sides.

Broadcast
Method of addressing, which poles all nodes within a subnet or a domain simultaneously.

Channel
Physical Lon-bus component, e.g. between 2 routers

cloned_domain
The domain of multiple nodes whose "must_be_one" bit has been set to 0. A cloned_domain is only used in exceptional circumstances and does not comply with the "interoperability guidelines" as set out by LONMARK. Subnet/node addressing is no longer used in a cloned_domain. Broadcast and NeuronID addressing is used in such a domain.

cloned_node
A node whose must_be_one bit is set to 0. Is able to receive messages from nodes, which work with the same subnet/node address. Is set when exporting the MIP on the LON Builder or by the update_clone_domain function.

Configuration network variable
A special network variable class, which supports saving application configuration data. Configuration data are always input variables, which are saved in the EEPROM. In the case of host-based nodes the host must ensure that the data are saved in a non-volatile memory area.

Configured Router
Router with two NEURON chips, which based on configuration data knows which telegrams are to be transmitted.

Connection
The implicit addressing, which is installed by the binding. A connection exists between two or more participating nodes.

Declared msg_tag
msg_tags defined in the application node. Declared msg_tags are always bidirectional.

Differential Lon Interface
LON interface electrically isolated with an isolating transformer on a 2-wire line. The transmission rate in the majority of applications is 78.1 kbps.

Domain
A logical binding of multiple nodes on one or more channels. Communication can only take place between nodes in the same DomainID unless a router binds two domains.

DomainID
The top level of the Lon bus address hierarchy. The ID can have a length of 0, 1, 3 or 6 byte. The 0-length is reserved for NSS-10 nodes to coordinate installation tasks and should not be used by application nodes.

Downlink
Data transfer from a host into a Neuron chip, generally via the parallel interface.

Embedded
Embedded is frequently used as a term for non-PC-based devices in which their functionality and the hardware is sold together as a device. For instance, a Saia PCD could be described as being an "embedded controller".

Explicit address
The address contained in the message, created and administered by the application (e.g. MIP).

Explicit message
Message explicitly triggered by a NEURON or host application, for which the contents and the time of transmission are defined by the application code.

FBox name
Name of the graphical PG5 Function Box

Flush
The flush status of an MIP interface ensures that messages transmitted to the LON bus are not recorded. Following a reset the MIP is set to its flush status so that the host application has sufficient boot time.

Flush cancel
So that the MIP interface records the LON messages, following a reset the "Flush Cancel" command needs to be sent via the parallel interface. If the Neuron chip reports "Flush complete", the host application is bound with the LON bus.

Free Topology Transceiver
Active transceiver at 78.1 kbps, which supports unrestricted bus topology. A Lon bus with FTT technology can be operated over a maximum distance of 400m. After each 400m segment a Physical Layer Repeater (2- or 4-path, one FTT per path) needs to be installed. In this way a practically unlimited overall network length can be achieved.

Gateway
Data bridges, which exchange data on the application layer. Can be used between two domains or different network protocols.

Group
Facility to create logical groups beyond the subnet limit. Up to a maximum of 256 different groups are possible.

Group address
Facility to address logical groups or individual group members beyond the subnet limit.

Group ID
A number for identifying a group. Each group is defined with a (unique) group number between 0 and 255. The number 0 applies in respect of "huge groups", i.e. a group with an unlimited number of members.

Group member
Member of a group. Up to a maximum of 64 individual addressable group members are supported or an unlimited number of group members, which are non-addressable by means of the member identification.

Host
A microprocessor, which has integrated layer 7 of the LON protocol. It can be a microprocessor coupled to the Neuron chip or it can be a Neuron chip.

Host application
The application program integrated in a host.

Host based node
A node, in which layer 7 of the LonTalk protocol is able to run in a non-Neuron chip microprocessor.

Hub
The binding centre. The hub either has an input and multiple outputs or multiple outputs and just one input.

Implicit address
An address implicitly contained within the NEURON EEPROM, which is used when accessing a network variable or a msg_tag. The application references the address via the network variable selector or the msg_tag.

Implicit message
A message triggered by the NEURON core when the application is assigning data to a network variable. Is transmitted during the first pass of the NEURON scheduler following data assignment.

Interoperability guidelines
Binding guidelines on which basis certification can be obtained. A product certified in accordance with these regulations is entitled to bear the LONMARK logo.

Interoperability, interoperable node
A product classification, which guarantees that different nodes from different manufacturers can be integrated in a network. For this installation to be completed, it does not need any customer-specific tools or special developments Interoperability is guaranteed by the LonMark certification.

Intersecting connections
A set of bindings, which share more than one global binding (multiple binding of variables).

Node
Is a node, as defined in LON bus technology: An application with a Lon interface.

Commissioning
Designates the insertion of a Lon node into the node list database for "Lon Binding" software.

Learning Router
Router with two NEURON chips, which learns from the incoming network traffic which messages need to be transmitted.

Link Layer
Transport layer, which defines access to the transmission medium and the transmission format. See also under OSI layer 1-7.

Lon-Bus
Field bus defined by the company Echelon, which can be controlled by means of the NEURON chips. The LON bus is a standard bus, which can transmit a standardised protocol over a wide variety of media such as 2-wire line, fibre optic, microwave routes, radio routes, network transmission etc..

LonBuilder
Development tool with emulators and routers, which support the development of individual nodes and entire networks.

LonIP
SBC Lon solution for IP852 (ISO/IEC 14908-4) channel.

LonFT
SBC Lon solution for Free Topology (ISO/IEC 14908-2) channel.

Lon-Manager
A set of hardware and software tools, which support the installation, configuration, maintenance, monitoring and control of a LonWorks® network.

LonMark
A certification program, which guarantees the compatibility of products of different manufacturers.

LonTalk®
The protocol used on LONWORKS networks, which standardises communication. It defines the standard under which individual nodes exchange information.

LonTalk file transfer protocol
A defined way of exchanging data files between nodes. File types 0 and 1 are defined by LonMark as configuration data files.

LONWORKS
A set of tools and components for creating a neural network of sensors, actuators and control devices.

Mapper
Node, which maps data based on explicit messages in SNVT according to the LonMark standard.

Message code
A field in an explicit message, which defines the type of message.

Microprocessor interface program
Firmware, which maps the telegrams received on the bus in an application buffer. In this way the LonTalk layer 4-7 can be implemented in a powerful microcomputer.

msg_in
A msg_tag, which exists by default on all nodes to receive incoming messages. Msg_in cannot be used for outgoing messages.

msg_tag
Variable in EEPROM, which supports integrating explicit messages into the EEPROM address information. Is used for implicit addressing of explicit messages and in principle acts as a "network variable" for messages. Is always bidirectional for input and output.

Network
A sub-system

Network address
The logical address of a node (domain/subnet/node).

Network driver
Software, which runs on a (non-Neuron chip) host, to operate the network interface (link to Neuron chip).

Network image
A network address of a node and its binding information. It consists of the domain, address and network variable configuration table. It is incorporated in the EEPROM of the Neuron chip or with host applications (network variable configuration table) on the host.

Network interface
A piece of equipment, which couples network layer 6 to a host (e.g. PCLTA PC LonTalk adapter)

Network interface API
A software library (C source), which supports basic communication functions. Is included in the NSS-10 developers kit.

Network Layer
Layer for transmission, which the destination address is responsible for. See also under OSI layer 1-7.

Network management
The process of logically defining, installing and maintaining a network.

Network services API
A software library (C source), which supports basic service functions. Is included in the NSS-10 developers kit.

Network variable
High-level objects, which are used for communicating between application nodes. The types, function and number of network variables are defined by the application code of the node. Network variables support a single type of communication particularly if Neuron chip-hosted applications are being used.

Network variable configuration table
A table, which assigns a selector to a network variable index. For downlink variables an address table is also assigned and additionally bound. In the case of a Neuron chip hosted node, the table is in the Neuron chip EEPROM. In the case of host applications, the table is saved in the host if the MIP has been created with the netvar_processing_off pragma.

Network variable index
A number, which is used to identify the network variable. The index numbers are assigned by the Neuron-C compiler based on the position of the variable in the section of the declaration. The first variable corresponds to the index 0. Neuron chip-hosted nodes can process up to the maximum of index 61, host applications can be extended up to index 4095.

Network variable selector
A 14 bit number to identify the binding between network variables. The selector numbers are assigned by the node responsible for the application.

Neuron Chip-hosted node
A node, in which layer 7 of the Lon Talk protocol is implemented in a Neuron chip.

NEURON Chip
Name derived from Neuron (the cell) for an integrated circuit, which contains a Lon interface and allows implementation of an application.

NeuronID
48-bit long identification number burned in during manufacture for each NEURON chip. Each number is a guaranteed unique identifier.

Node
Node. A piece of equipment, which contains layer 1 to 6 of the LonTalk protocol and a Neuron chip, Lon Transceiver, memory and carrier hardware.

NodeID
The lowest level of the LonTalk address hierarchy consisting of domain/subnet and node. During installation each node is assigned a subnet / node combination that only occurs once. Exception: cloned_node. 127 different NodeIDs can be defined (1..127). The NodeID 0 is used for a node that has not yet been installed.

Orphans List
In LonWorks IP networks the word orphans designates devices, which have actually been found by an IP-852 configuration server, but have not yet been assigned a channel. See chapter 7.1.3

OSI-Layer 1-7
Layer 7: Application Layer. Application level compatibility: Standard Network Variable Types
Layer 6: Presentation Layer. Data Interpretation: Network variables, foreign frame transmissions.
Layer 5: Session Layer. Remote Actions: Request-response, authentication, network management, network interface.
Layer 4: Transport Layer. Point-to-point reliability: Ackd / Unackd Service, unicast/ multicast authentication, address allocation and double entry control.
Layer 3: Network Layer. Destination addressing: Addressing router
Layer 2: Link Layer. Access to the transmission medium and transmission format: Framing, data encoding, CRC error checking, CSMA, collision avoidance, priority and collision identification (optional)
Layer 1: Physical Layer. Electrical connection: twisted pair, power line, radio frequency, coaxial cable, infra-red, fibre optic, RS-485 etc.

Physical Layer
Layer for transmission, which defines the electrical connection. See also under OSI layer 1-7.

Poll
An explicit request to a node to send the value of a variable with the corresponding selector.

Polled network variable
An output network variable, which only sends its contents based on polling requests. Network variables normally automatically send their contents if it has changed (i.e. if the variable has been described by the application).

Polling network variable
An input network variable, which only updates its contents based on polling requests to an output variable.

Presentation Layer
Transport layer, which defines data presentation. See also under OSI layer 1-7.

Priority
A mechanism supported by the LonTalk protocol to transmit prioritised messages. Priority messages are transmitted within a reserved slot before the normal messages. Particularly suitable for transmitting deterministic information (timestamp, time-critical data).

Processed netvar
Addressing the network variable by means of the implicit address, i.e. with address information contained in the NEURON chip EEPROM.

Program ID
An identification string, which is stored in the EEPROM of the Neuron chip. The string is used to identify the application program, all nodes with the same program ID must have the same external interface as otherwise problems will occur with installation tools. Interoperable nodes, which are certified in accordance with LonMark, contain a standard program ID.

Property
An attribute of an object, e.g. the location of the node.

Repeater
Router with two NEURON chips or physical repeaters, which maps all messages for one channel to the next channel.

Self-documentation
A mechanism, which enables the application node to accommodate defining information in the EPROM.

Self-identification
A mechanism, which supports documenting SNVT variables in the PROM of the application node (SNVT ID). This information can be requested during installation using a software tool suitable for this purpose.

Serial LonTalk Adapter
A network interface based on an EIA-232 interface. This information can be requested during installation using a software tool suitable for this purpose.

Session Layer
Transport layer, which defines external access (remote actions). See also under OSI layer 1-7.

SMX-compatible transceiver
Each transceiver, which uses the standard modular transceiver identification code.

Standard network object
A collection of network variables with associated behaviour according to the requirements of the LonMark Interoperability Guidelines.

Standard Network Variable Type
Standard network variable types are variables standardised by LONMARK, which make it possible to simply exchange data from nodes of different manufacturers.

Standard Network Variable Type ID
A standardised code, which is assigned to a corresponding variable type. In Echelon documents is occasionally also designated as the SNVT index. An SNVT ID is always a number that does not equal 0, in which 0 means that in the case of variables it is not an SNVT variable.

Standard program ID
A program ID of a node certified in accordance with LonMark Interoperability Guidelines, which supports references to manufacturer, application and software version.

Subsystem
Two or more nodes, which fulfil a common function. The configuration of all nodes in a subsystem is implemented by an individual installation tool.

Subnet
Logical subnet within a domain. It can contain up to a maximum of 127 nodes, a domain can contain 255 subnets.

subnet / node address
Standard address of a Lon node. In total 32385 combinations are possible.

Subnet ID
The second level in a subnet/node addressing hierarchy. Valid subnet numbers are 1..255. The subnet number 0 is used for a node that has not been installed.

System
One or more independently administered subsystem(s). A system can use one or more domain(s).

Transceiver
A piece of equipment, which physically connects the Neuron chip to the transmission medium.

Transceiver ID
A 5-bit number, which supports hardware decoding of the transceiver type.

Transport Layer
Transmission layer, which is responsible for point to point transmission. See also under OSI layer 1-7.

Turnaround network variable connection
A network variable binding in which case the input and output are on the same node.

Typeless network variable
A network variable for which neither the type nor the data length are known. The host application is responsible for transmitting such variables.

UDP-Protocol
UDP (User Datagram Protocol) is an IP based connectionless transmission method for exchanging data between machines and PCs. It is frequently used for protocols such as LonWorks IP-852, KNX-IP or BACnet-IP. The data packets are assigned by a so-called port number to a particular protocol. In this way different UDP-based protocols can be used in parallel without mixing up the data. LonWorks IP-852 devices usually use ports 1628 and 1629.

Unprocessed netvar
Addressing the network variable by means of the explicit address, i.e. with address information delegated to the host application code.

Uplink
Data transfer from a Neuron chip into a host microcomputer, generally via the parallel interface.

Variable Fetch
A request to a node to send the content of the variables with a corresponding index.

A.3 Abbreviations

CRC	Transmission control and error correction
CSMA	Collision-enabled network protocol, i.e. each subscriber is permitted to actively send given an unrestricted medium
ECS	Enhanced Command Set
FTT	Free Topology Transceiver
IP	Internet Protocol
IP-852	IP tunnelling standard for field buses (including LonTalk)
ISO	International Standard Organisation
kbps	kilobytes per second 1 kbps = 1000 bytes/sec = 1kHz
LNS	Lon Network Services
LON	Local Operating Network
LPA	Lon Protocol Analyser
MIP	Microprocessor Interface Program
NIC	Network Interface Card
OSI	Open Systems Interconnection
SCPT	Standard Configuration Parameter Type
SLTA	Serial LonTalk Adapter
SFPT	Standard function profile type
SNVT	Standard Network Variable Type
TP	Twisted Pair

A.4 Books, links, references

A.4.1 Books

LonWorks® Installation Manual VDE Verlag ISBN 3800725754

LonWorks® Planner Manual VDE Verlag ISBN 3800725991

LonWorks®- Technik in der Gebäudeautomation
Huss- Medien GmbH
Verlag Technik
ISBN 3341013466

A.4.2 Links

LonMark® home page
http://www.LonMark.org

LonMark® NVT Master List available via
http://www.Echelon.com

LonMark® Resource Files V13.10
http://types.LonMark.org

A.4.3 References

Book Title	Edition	Type of Book
LONTALK PROTOCOL	April 1993	LonWorks® Engineering Bulletin
NEURON Chipbased Installation of LonWorks® Networks	1991	Echelon Engineering Bulletin
Installation Overview	January 1995	LonWorks® Engineering Bulletin
Enhanced Media Access Control with LONTALK Protocol	January 1995	LonWorks® Engineering Bulletin
FTT-10 Free Topology Transceiver	1994 Version 1.2, Document Echelon 078-0114-01B	LonWorks® Users Guide
LonWorks® Host Application Programmers Guide	Revision 2 078-0016-01B	
Neuron Chip Data Book	January 1995	Echelon Data Book
Neuron Chip Distributed Communications and Control Processors	1994 Rev 3	MOTOROLA Data Book
Application Layer Interoperability Guidelines	1995 V 2.0	LonMark
Layers 1-6 Interoperability Guidelines	1994 V 1.3	LonMark
Local Operating Network	ELRAD Book 12/1994,1/1995	Ludwig Brackmann
Offene Kommunikation mit LON und BACNET ['Open Communication with LON and BACNET']	LNO Info 1996	Nils Meinert
BACNET specification 1995	ANSI / ASHRAE 135-1995	ISSN 1041-2336
Grundlagenpräsentation zur LonWorks Technologie ['Presentation of the Fundamentals of LonWorks Technology']	Jan 1997	Fritz Kurt, EBV Elektronik
Lon-Technologie ['Lon Technology'] Dietrich Loy Schweinzer	1998	Hüthig Verlag, ISBN 3-7785-2581-61998
LonWorks® technology	1998	Tiersch F. LonTech® Thüringen e. V.ISBN 3-932875-03-6
GNI-Handbuch der Raumautomation ['GNI Manual of Room Automation']	1. Edition	AZ Verlag ISBN 3-905214-33-4
LonWorks Installation Manual, Lon Nutzer Organisation e.V.		ISBN 3-8007-2575-4
LonWorks® Gewerkübergreifende Systeme ['LonWorks inter-plant systems'], Lon Nutzer Organisation e.V.		ISBN 3-8007-2669-6

A.5 Company address of Saia-Burgess Controls AG

Contact

Saia-Burgess Controls AG
Bahnhofstrasse 18
3280 Murten, Switzerland

Telephone switchboard	+41 26 580 30 00
Telephone SBC Support	+41 26 580 31 00
Fax	+41 26 580 34 99

Support

E-mail Support:	support@saia-pcd.com
Support site:	www.sbc-support.com
SBC site:	www.saia-pcd.com

International representations & SBC sales companies:	www.saia-pcd.com/contact

Repair

Postal address for customers to return products in Switzerland:

Saia-Burgess Controls AG
After sales service
Bahnhofstrasse 18
3280 Murten, Switzerland