Manual

SAIA BURGESS CONTROLS

LONMARK®

LONWORKS® Networks with Saia PCD®

Document 26-767 | Edition ENG05 | 2018-10-08

Content

0

0 Content

Content

Content

Content

0

0.1 Document History

Date	Version	Changes	Remarks
1999-01-01	EN01	-	First edition
2000-06-01	EN02	-	UpDate
2011-04-26	EN03	-	New Outfit, xx7 Step. Doc. on FAQ
2013-10-23	EN04	-	New Logo and Brand
2018-10-08	ENG05	Ch. A	New phone number (2015)
2019-07-24	ENG05	Ch. A	Fixed wrong phone number

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.

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 Philosophy and components of LON

1.1 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. The nodes measure, control, regulate and communicate. This creates an extremely flexible network of functions with virtually any degree of networking and complexity.

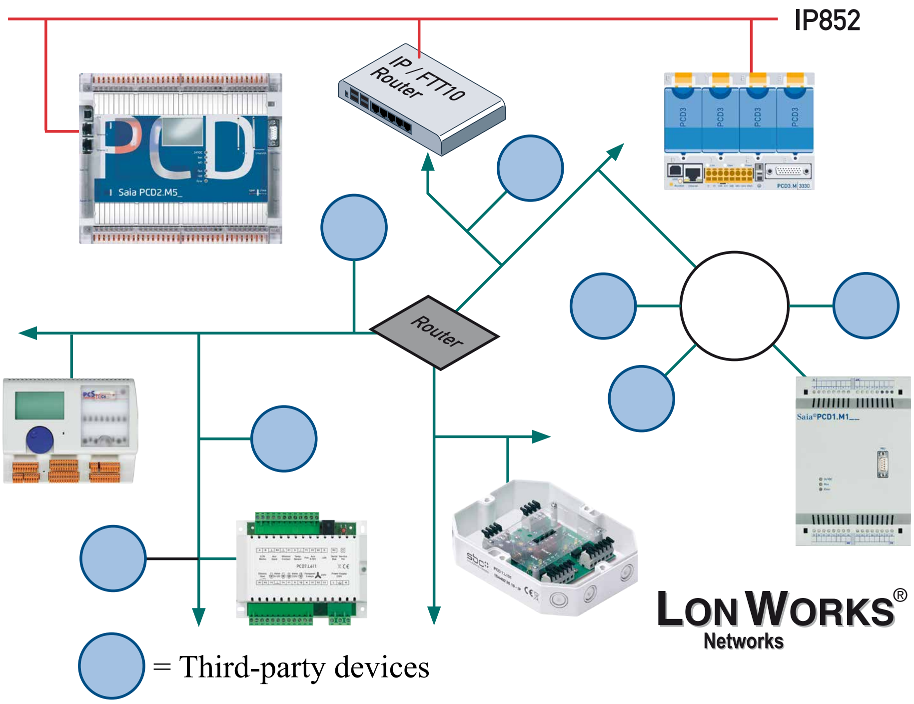


Figure 1-1: Decentralised nodes

From the outset implementation rather than standardisation was the motto of the founder of this technology around A.C. Markkula, which was able to establish a name for itself even before this was the case for Intel and Apple as a manager of high tech companies in their pioneering stage.

ECHELON succeeded by making a chip available with an integrated communication system by bringing about the rapid dissemination of a quasi standard. At its core, the LonTalk® protocol, was only made available using these specific chips until the standard had become established.

As of today the protocol is now standardised and released for implementation on other chips. LonWorks has been used in numerous standardisations, thus for example iin BACnet (ASHRAE American Society of Heating and

Air-Conditioning Engineers), ISFS (International International Forecourt Standard Forum, i.e. all large oil companies), CEN TC-247, SEMI (mass flow rate meters), CELECT (UK for heating systems) and IEC 708.1..708.3.

LonMark® represents the most important standard, an organisation founded by ECHELON consisting of suppliers of LON components, which have taken on this standard themselves.

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

1.2 The four components of LON

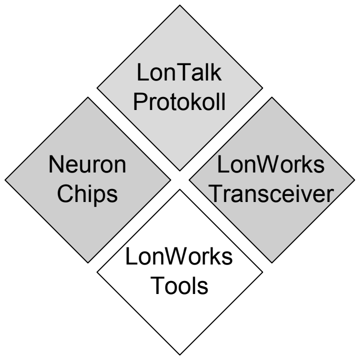


Figure 1-2: The four elements

In principle, LONWORKS technology is based on four elements:

- The LonTalk protocol defines the language, which is spoken on the LON medium.
- The Neuron chips are able to interpret this language and create 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 (LonBuilder, Node-Builder) and installation tools (LonMaker, ICELAN-G, Helios).

1

1.3 The LonTalk® protocol

The NEURON chip "talks" 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.

1.3.1 Basic structure

1.3.1.1 Transmission modes

Transmission takes place in packets. The compiling and sending of 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:

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 several times in succession, 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.

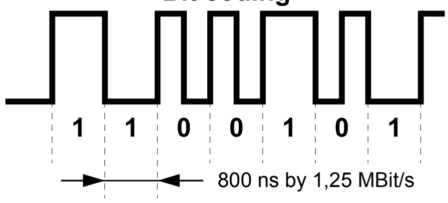
The user can freely determine which mode is to be used.

The data packets in the case of an FTT-10 interface 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 of the signal is registered per data content. The Manchester decoding makes it possible to run lines without needing to worry about polarity. The baud rate (number of bits transmitted per second) corresponds to the frequency in this mode, that is data transmission of 78.1 kHz is able to deliver 78.1 kBit/sec of information.

This data rate is not, however, achieved by the LON bus, as the telegram length is limited.

- **Differential Manchester code without DC portion for chosen media**
- **Bit synchronisation adjustable to Transmission medium**
- **Variable useful data: 1- 228 Bytes**

Bit coding



Telegram structure

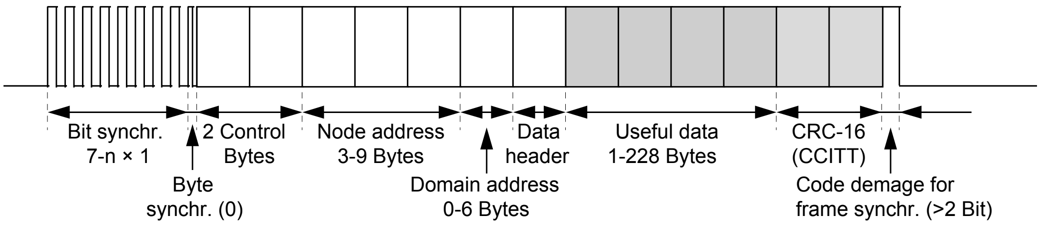


Figure 1-3: Data format

A telegram always consists of the synchronisation bits that can be adapted to the respective transceiver (consequence of "1"). The synchronisation bits are used for switching the transceiver so that it can settle to the receiving frequency. The first 0 indicates the start of the address data, which show the receiving node whether it actually needs to take note of the incoming telegram. The usable data or the ACK/ NACK bytes follow the address to show whether a message was successfully received.

1.3.1.2 Data integrity

With open bus systems additional data integrity can optionally be granted. In a special transmission mode the recipient can verify the authenticity of the sender. To do this when installing the network a 48 bit code number is agreed on between the sender and the recipient. This code does not depend on the chip-specific identification number. The code number is send using encryption processes that change upon each transmission, which guarantees a high level of security.

If a node receives an authenticated message, then it requests the sender to prove its authorisation. To do this it send it an encrypted random number (64 bit). The sender encodes this number using his password and sends the result back. The recipient compares the answer with his own encryption result. If matching, the recipient's network CPU accepts the original message and forwards it on to the application program. Otherwise, the receiving node ignores the original telegram and increments an error counter. The authentication can be defined for each individual network variable and for network management commands.

1

1.3.1.3 Priorities

The different nodes can be given different priorities. For high-priority messages special time bins are reserved at the end of each packet during which the transmission of one of these packets can actually start. Low-priority nodes can only start the transmission at a later time provide the transmission channel is not then already occupied by a high-priority node. In this way shorter access time can be provided for time-critical applications for certain nodes.

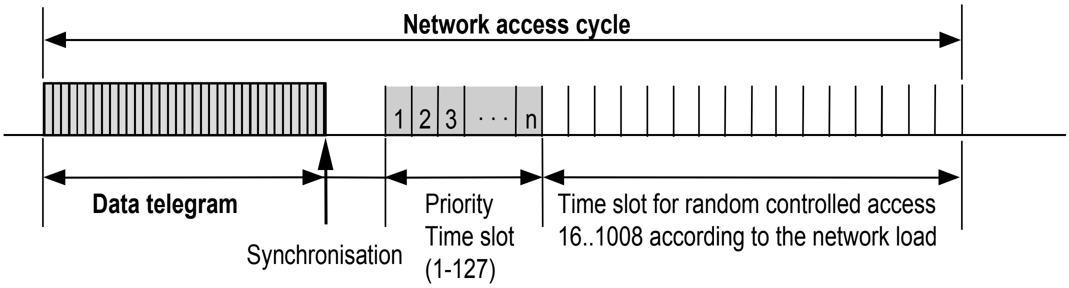


Figure 1-4: Priority time slot

Figure 4 shows the telegram sequence with the time slot reserved for prioritised messages. As a result the protocol supports prioritising a limited number of messages to be sent very quickly. The delay time within the priority slot and the normal time slot is allocated at random using the CSMA process.

1.3.2 What is CSMA ?

CSMA stands for "Carrier Sense Multiple Access". Different subscribers to a system may access the communication medium for which the most intelligent possible algorithms are used for identification and to avoid collisions.

To minimise the probability of collisions, a refined mechanism has been developed. A node, which wants to issue a packet, first of all "listens" to the bus to determine whether or not it is already occupied. If it finally detects the end of a foreign packet, it does not immediately start to send but waits for a specific number of units of time, so-called 'time bins' (which are only a few bits long) The node will finally start on the transmission of its packet during one of these time bins.

The first few time bins are intended for high-priority nodes (see above). If the node has lower priority, it continues to wait for a specified number of time bins, until it finally starts to send. This number is determined by the random number generator. While waiting, the node continues to monitor the events on the bus. If another node arrives before this to send, the procedure starts over.

The probability that two nodes start to send during exactly the same time bin is relatively small thanks to control using the random number generator. Thus the number of collisions can be kept to a relatively small value even with a high bus traffic load.

The LonTalk® protocol is characterised by the "predictive p-persistent CSMA" algorithm developed at the Stanford University. This algorithm makes it possible to transmit a guaranteed data rate in the event of a network overload. LonTalk is therefore superior to the other field bus systems in terms of overload behaviour. Not even the Internet is able to lay claim to such capabilities.

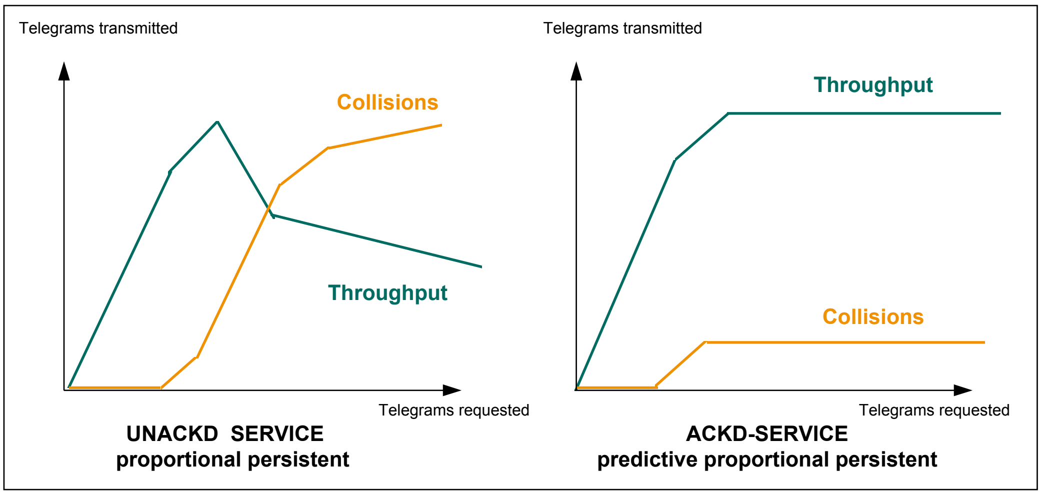


Figure 1-5: predictive p-persistent CSMA

1.3.3 The OSI layer

The definition of OSI (Open System Interconnection) is the basis on which the Internet / Intranet Technology has been built. Lon-Works 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. 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 two-wire line, radio, IR, LWL, coax, Tf-conductor, 230 V mains voltage etc. 610bit/s - 1.25Mbit/s

Table 1-1: the OSI layer model

1.3.4 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:

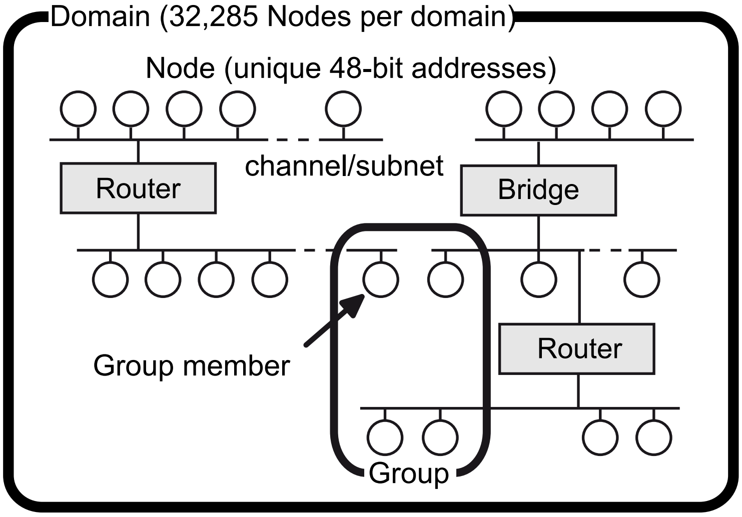


Figure 1-6: Addressing a LON system

1.3.4.1 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 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 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.

1.3.4.2 Channel

A channel is the physical transmission medium on which serial data is transmitted. The channel can, for instance, be a cable, a radio frequency or a part of the 230 VAC 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.

1.3.4.3 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. Thus a subnet can, for example, contain all light nodes in a factory, although they are controlled via radio, the 230 V mains power or via two-wire bus.

1.3.4.4 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).

1.3.4.5 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.

With the appropriate installation tools, a group can be divided into multiple subgroups using the so-called "group overloading". See chapter 5.

1.3.5 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 3 bytes for group addressing and 15 bytes for addressing via the neuron ID with a 6-byte domain address.

1.3.6 Explicit Messages

All LON telegrams are "explicit messages", i.e. a "data train" which finds its path through the network to the correct destination node. The locomotive contains as a guide the address, which automatically triggers the setting of points in the network. Similarly in the internet, data can therefore be transmitted in any form (layer 6). Explicit messges 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.

Advantages:

- more efficient as a network variable

Disadvantages:

- without an exact knowledge of the message structure, a binding is not possible (i.e. connection to node of third-party manufacturers is only possible with some difficulty)
- require larger programming overhead, thus more code

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

1.3.7 Network variables

Network variables constitute the foundation of an important and in this form unique characteristic of LonWorks, so-called interoperability. It is understood to mean seamless interaction operating based on simple "rules of play" of LonWorks-based products from different manufacturers, e.g. a gas burner, a temperature sensor in the boiler, a circulation pump, a room controller with multiple room temperature sensors and radiator valves. Because of the different forms of interaction between production and installation engineering, interoperability is an important prerequisite for the distribution of Lon-Works within the industry and in buildings automation. This could be expressed differently: With LonWorks complex systems of this kind can be built as though they were from the same source. So LonWorks is constantly developing, although inexorably towards becoming a de facto standard.

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.
- 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 their data. This data is always transmitted in sequence, when its value changes.

1.3.8 Configuration and network management

Logically using network variables a wide range of communication connections can be established (so-called bindings) between the individual Neuron nodes. Generally this is completed using an installation tool in the field (hand-held device, PC with Windows and software such as, e.g. "NL220" from the company "Neuron Systems" or "LonMaker" from Echelon", see appendix). 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 systems.

Auxiliary devices

Larger systems are operated with the help of a network management node (NMK for short, hand-held equipment or PC). An NMK can interrogate and configure a LON system for newly added nodes, and then load, start, stop and reset an application program onto the node. 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 allocation must be made between the physical position of each LON node. To do this using the ... command the installer can 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.

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, the neuron chips provide a node identification string each eight bytes in length.

1.4 LonWorks® nodes

1.4.1 NEURON®-based nodes

1

At the core of LonWorks technology is the NEURON® chip. It exists in two variants, as a single chip (type 3120) for simple applications and as a chip with up to 64 kBytes of external memory (type 3150) for complex applications.

The same applies to both, that each has 3 CPUs. Two of them are exclusively engaged with processing message telegrams via the communication port, whereas the third CPU processes the user program. Data exchange between the CPUs takes place via the RAM data buffer. The firmware for the event-controlled operating system, the LonTalk® protocol and a library of currently 34 I/O models are in the integral ROM memory, using which any complex digital inputs and outputs can be processed to the pins of the I/O block application. The 3120 chip also has enough EPROM in which the application program and the network configuration parameters are saved. As a result it is possible to integrate what might still be a "virgin" node into a network and to load (download) "its" specific application program over the network. Other important properties are two fast clock & timer circuits as a basis for the timing of the I/O functions and a globally unique 48-bit serial number. This is not responsible for installation purposes, but may also be useful for issuing ID numbers in the in-house product database.

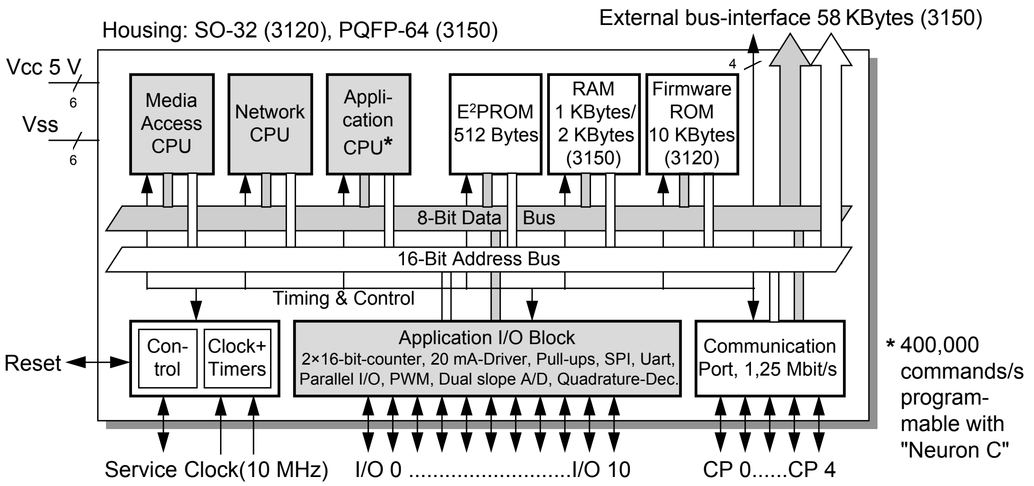


Figure 1-7: the NEURON chip

1

1.4.1.1 I/O chip connections

The Neuron chip provides 11 I/O pins that can be configured in a whole range of ways for the application interface. Together with the integrated 16-bit timer/counter blocks and 29 operating routines, they open up to the application programmer an interesting and extensive range of functions for controlling different types of sensor and actuator. The I/O routines available in the firmware (also referred to as I/O objects) save the programmer the laborious task of setting bit shift algorithms at assembler level. Accordingly, cost-effective functional modifications are possible on Lon-Works® products.

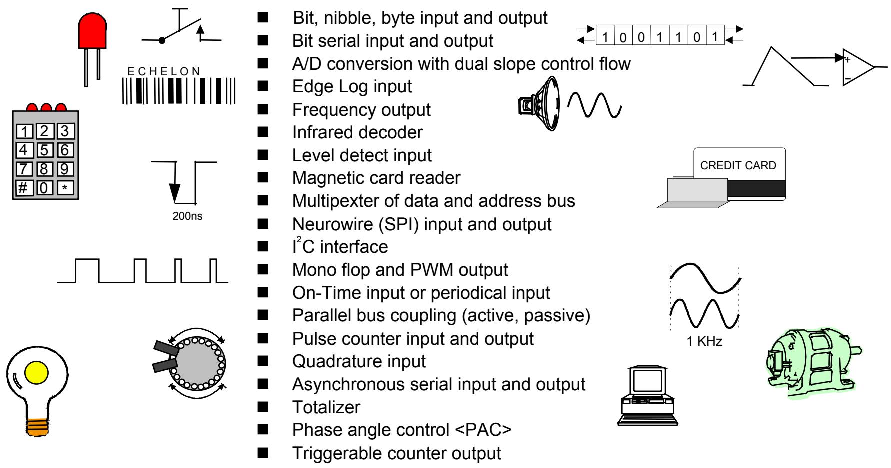


Table 1-2: IO capabilities

1.4.1.2 Firmware, EEPROM, PROM, Flash PROM, RAM

Important terms for handling nodes are the memory types of NEURON microprocessors:

Firmware Firmware is understood to be the program running in the Neuron® chip.

EEPROM The Neuron® contains 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.

PROM A PROM contains firmware and can no longer be externally modified after being programmed.

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 (Random Access Memory) RAM is volatile memory, which can either be stored by means of battery or loses its contents after being switched off.

1.4.1.3 Service pin

The so-called service pin is a special connection of the Neuron chip. It represents a natural 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 ground, 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.

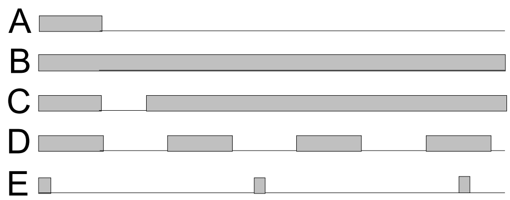


Diagram 1-1: Flashing sequence of the service LED

Meaning of the LED displays		
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 with a frequency of 1 Hz. The hardware works correctly, although the user program has not yet started. The node now needs to be configured (allocation 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 Neuron chip is in each case started upon activating the service pin irrespective of whether the node is already supporting a user program and whether 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 (processor 2 on the Neuron chip) 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 user program in terms of the logical classification of the service pin, which depend on the processor type and the firmware version.

1

1.4.1.4 Neuron-C programming

Neuron chips are programmed in "Neuron-C". Generally the nodes can be reloaded over the network; the entire network is thus turned into a freely programmable application.

The functionality of an entire LON system can thus be described in the form of a C program whose individual procedures communicate with one another via network variables. The fact that the individual subroutines run on different microcontrollers physically interconnected only by means of one bus, is secondary to the programmer.

Companies are found amongst the LonTEch service providers, which can efficiently implement special applications.

1.4.1.1 Configurability

NEURON nodes have a data structure, which supports binding to their network partners. This data structure is generally managed by an installation tool, which takes on 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.

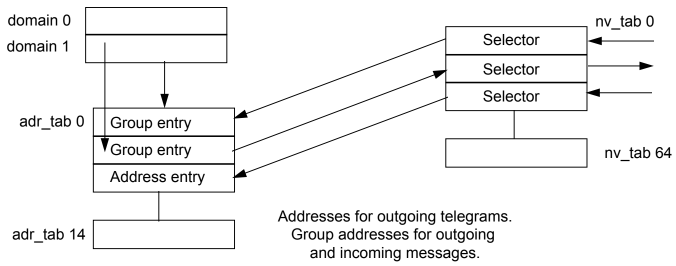


Figure 1-8: The configuration data of a NEURON

If an output variable contains a new value, the program looks in the "nv tab" 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 NEURON can therefore address up to a maximum of 14 other nodes directly. If group addresses are used, up to a maximum of 14 groups can be serviced in which case incoming group messages in the address table also need to be registered. The group table can, however, use multiple selectors so that a node can be bound to more than 14 recipients.

1.4.2 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.

Chip type	3120
CPUs	3
EEPROM bytes	512
RAM bytes	1,024
ROM bytes (firmware)	10,240
External Memory Interface	no
16-bit Timer/Counter	2
Watchdog Timer	yes
Package	SOIC
Pins	32

Table 1-3: NEURON 3120 chip data

1.4.3 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.

Chip type	3150
CPUs	3
EEPROM bytes	512
RAM bytes	2048
ROM bytes (firmware)	-
External Memory Interface	yes
16-bit Timer/Counter	2
Watchdog Timer	yes
Package	PQFP
Pins	64

Table 1-4: NEURON 3150 chip data

1.4.4 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.

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

1.4.5 HOST nodes (NMK Network Management Nodes)

Host nodes are nodes, which are also able to take on network management functions. HOST nodes manage and connect other nodes.

Host nodes do not have volatile memory (EEPROM, hard disk) and are able to manage 4096 selectors and any number of addresses as the host node issues the entries in the nodes itself. The installation tool integrated on the Host node decides on issuing groups and subnet/node addresses and can therefore adapt the entries to requirements.

The installation tool must, however, manage with 15 groups for messages being received, although it can issue multiple selectors into the same group per address table.

In the conventional architecture of a LON bus system, it is only possible to work with a single host per installation, which makes integration of larger systems more difficult.

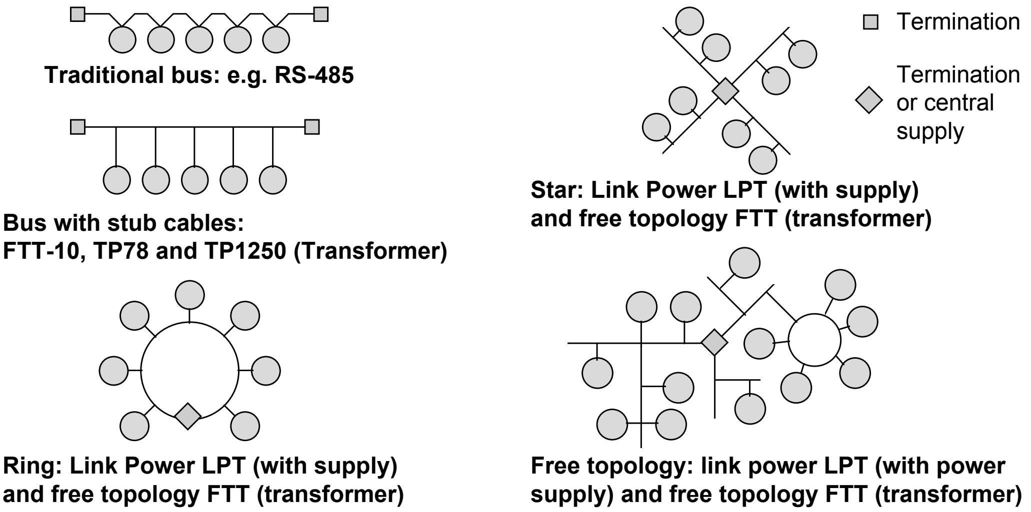
The Lon Network Service architecture (LNS) supports multiple host nodes, which coordinate with one another the entries in the configuration data using the client-server principle.

1

1.5 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. Drawing 1-9 illustrates possible topologies:



Drawing 1-9: LonWorks bus topologies

1.5.1 Twisted Pair TP 78

For conventional bus topology it is possible to work with 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
Spur:	maximum 3 m
Special:	in the case of minus temperatures only 44 nodes per channel
Zero voltage range:	+230 V...230 Vrms

Table 1-5: Data Twisted Pair TP 78

1.5.2 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 400m in free topology and terminated at one end.
Nodes per channel:	64
Zero voltage range:	+230 V...230 Vrms

Table 1-6: Free Topology FTT-10

1.5.3 RS-485

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

Type	Medium	kBit/s	Length / Topology / Note	No. Nodes
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	Twisted pair line	78	1200 m at 39 kbit/s, bus, with or without electrical isolation	32 per bus segment
TPT/ XF1250	Twisted pair line	1250	130 m, bus with 0.3 m spurs, isolation 277Vrms	32 per bus segment
FTT10 transformer	Twisted pair line	78	2700m as bus, 500 m for free topology, isolation 277 Vrms	32 per bus segment
LPT10 Link	Twisted pair line	78	500 m, free topology, 42 V DC, 5 V / 100 mA per node	32 -128 per bus segment
PLT20 Power	230 VAC or DC	4.8	50 m – 5 km, BPSK Modulation Cenelec Band C, 132.5 kHz	depending on mains power
PLT30 Power	230 VAC or DC	2	50 m – 5 km, BPSK Modulation Cenelec Band C, 132.5 kHz	depending on mains power
IP-852	Tunnelling via IP		All IP channels	

Table 1-7: Overview of the LonWorks transceivers

1.5.4 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 free selection of topology (star, ring, multi-drop). The same concept was also the triggering factor for the development of the FTT-10, the free-topology transceiver. Unlike the LPT-10, each LON node has its own power supply. Both variants can also be mixed.

1.5.5 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 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 useable 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 to 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 works in this mode within the range from 100 kHz to 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 to 95 kHz) is reserved for data exchange of grid operators (electricity companies and distributors). CENELEC-B band (95 kHz to 125 kHz) is used for communication without access protocol for end customer applications. In the CENELEC-C band (125 kHz to 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.

1.5.6 Other transceivers

In addition, the following transceivers are available on the market:

- Fail-safe transceiver 78 kbit/s
- Radio 432 MHz
- Fibre optic
- Infra-red
- Coax
- Tf-line
- Microwave

1.6 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 quite standard on all nodes.

1.6.1 Installation tools

Two possible tools in Widnows for installing a LON network are, for example,:

- "LonMaker" by Echelon
- "NL220" by Neuron Systems

With installation tools there are two generations, namely the Helios, Ice-lan-G, Alto and Metravision built on the first Windows Application Interface, and the LNS/LCA (Lon Network Server/Lon Component Architecture) Tools LonMaker for Windows, Unilon, Response and Pathfinder.

With all installation tools commonly used today it is possible to stipulate that the inclusion of building plans and the graphical representation of the assembly location of the nodes should be supported.

The more recent LNS/LCA tools build on modern standards for Windows 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 in the installation tool. By double-clicking on the node image, the corresponding plug-in window is opened.

Tools are generally marketed so that no fees apply 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.

2

2 The LonMark® standard

2.1 The physical layer (layer 1)

LonMark's physical layer adopts the transceiver specification and has been defined for the following transceivers:

- TP-RS 485-39
- TP/XF-78
- TP/XF-1250
- TP/FT-10
- PL-10 (L-E)
- PL-20 (L-N)
- PL-20 (L-E)
- PL-30 (L-N)
- RF100
- IP-852 (Lon over IP)
-

2.2 Layer 2 - 6

LonMark® only defines minimal additional conditions in layer 2 and 4:

- Layer 2: minimal quartz frequencies in relation to transceivers
- Layer 4: setting the minimum size of the transaction buffer to 66 bytes

2.3 The Application Layer (Layer 7)

To guarantee interoperability, network variables for objects are collated which, logically viewed, represent sensor, actuator and controller functions. In this respect, LonMark® is applying the lever and has already defined over a hundred SNVTs (Standard Network Variable Types) and SCPTs (Standard Configuration Parameter Types), which guarantee interoperability of variables in terms of meaning, importance and range.

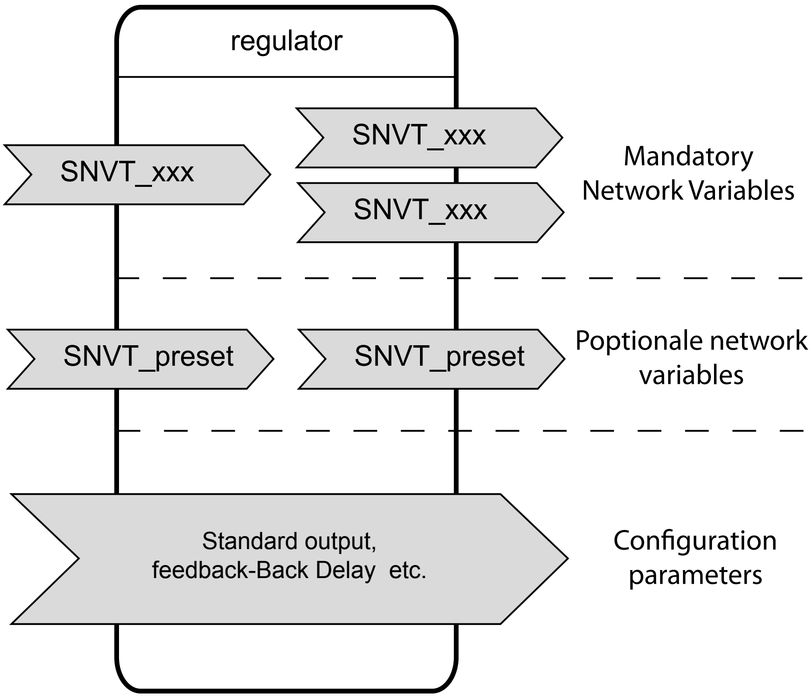


Figure 2-1: Documentation for a node with LonMark® objects

An SNVT is given a number, which defines its type. Furthermore, information is saved in the node relating to the SNVT, which can be read from the node using the installation tool. This text information generally contains the variable names so that the function can be understood from it.

The following table displays an extract from the SNVT definition of LonMark®:

Measurement	Name	Range	No.
Speed	SNVT_speed	0..6553.5 m/sec in 0.1 m/s	34
	SNVT_speed_f	-1E38..+1E38 m/s	39
Sound level	SNVT_sound	-327.68...327.67 dB (0.01 dB)	33

Table 2-1: Extract of SNVT definition of LonMark®

SNVTs are able to contain complete structures. Thus, for example, the "SNVT_ time_ stamp" contains full time information in terms of year, month, day, hour, minute and second.

An up-to-date SNVT list can be found on LonMark's home page (see Appendix).

2.4 LonMark® objects

2.4.1 The structure of a LonMark® object

For use on an interoperable network node, the Lon-Talk protocol exists and thus access to the network only in the form of LonMark® objects. These are characterised by their type (by means of a number issued by the LonMark® organisation), a set of network input and output variables and a set of configuration parameters. The LonMark® objects represent standards relating to form and meaning from the point of view of the desired interoperability in the network. The generally valid structure is illustrated in Figure 4-5.

The object number characterises the type, the name is used only to make it comprehensible. The object always has one or more indispensable (mandatory) network variables and it may support optional NVs. The input variables are shown on the left, and the output variables on the right. Both types are numbered in series from 1 to n, for which a distinction needs to be made between input and output variables. Only SNVTs are used.

The configuration, on the other hand, which can also be entered using NVs, bears the number of the associated configuration parameter (SCPT) from the SCPT list [8]. The names of the NVs should not be longer than 11 characters, should not contain an underscore and are written in lowercase apart from the first letter of a word. Examples can be found in the following sections.

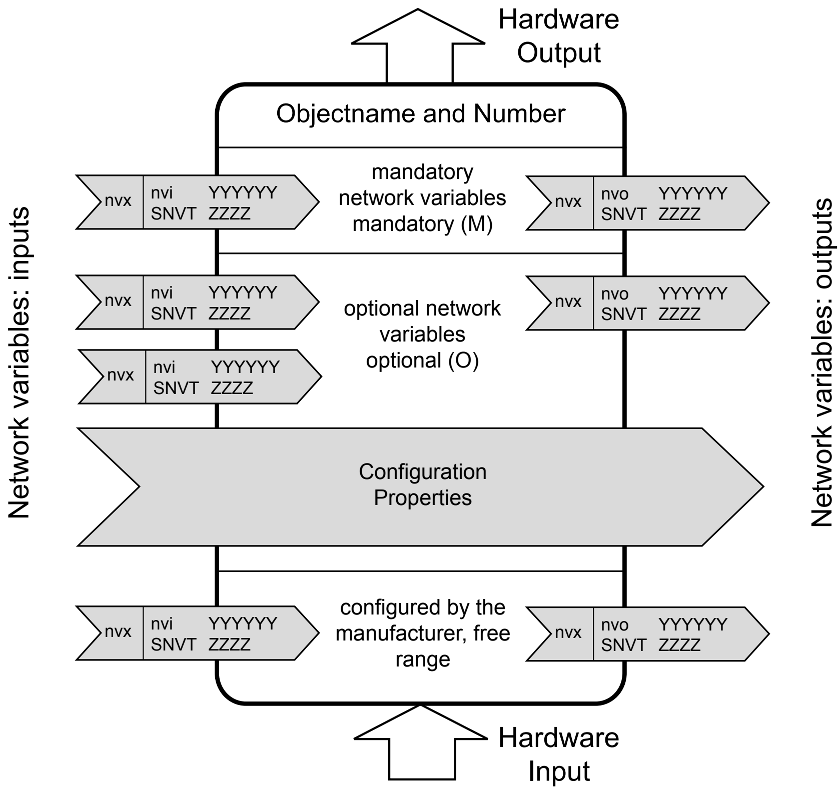


Figure 2-2: General structure of a LonMark® object

The names are given a prefix. The prefix describes its memory class and the direction of transmission:

variable	saved in
nvi ~ input variable	RAM
nvo ~ output variable	RAM
nci ~ configuration variable	EEPROM
nro ~ (read only) output variable	ROM

Table 2-2: Prefix for variables

The relationship to the user process is represented by an arrow, as a hardware output above and as a hardware input below. By specifying this general information, the different object types are created, such as node object, sensor object, HVAC object, etc.

2.4.2 The node object

The node object, to which the object type number "0" is permanently assigned, is used for monitoring and influencing the functions of all objects in the network node. This is achieved using the two indispensable NVs ‹nviRequest› of the type ‹SNVT_obj_request›, and ‹nvoStatus› of the type ‹SNVT_obj_status›.

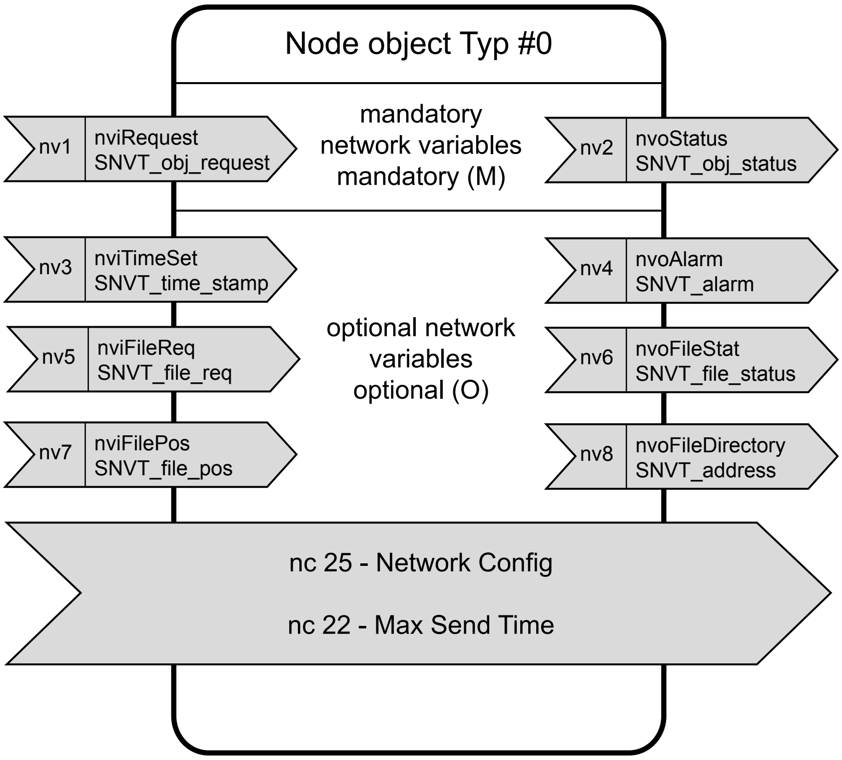


Figure 2-3: Structure of the node object (object type #0)

The network variable ‹nvi_Request› contains a 2-byte field for the number of the object on the node and a 1-byte field for the command encoded as a number, e.g.

0 ~ Rq_Normal
2 ~ RQ_Update_Status
3 ~ RQ_Self_Test

Using the command "0" the command being addressed, e.g. inactive state is reset to its normal mode of operation. If the command "2" is given to a particular object, this object sends its current status using the output variable ‹SNVT_obj_status› for the node object. With the command "3" an object on the node can be triggered to perform a self-test. If, however, the command "0" is sent to the node object itself, all objects are set to their normal state on the network node.

The following listing (not exhaustive) shows the state that an object can be in: ‹disabled›, ‹out_of_limits›, ‹mechnical fault›, ‹electrical fault›, ‹unable_to_measure›, ‹comm_failure›, ‹in_alarm› and others. Electrical and mechanical faults can only then, however, be reported with certainty the hardware requirements were created for developing the nodes for this.

A full description of this background information is provided in the "LonMark® Application Layer Interoperability Guidelines". The same also applies for the optional NVs and the configuration parameters.

The configuration parameter ‹Max Send Time› defines the maximum waiting time after the expiry of which the object reports its status by itself using the network variable ‹NVT_obj_status› without an NV updating preceding this. This function is referred to as a "heartbeat" and as such shows that the object is "still alive". The configuration parameter ‹Max Send Time› bears the number 22 from the master list of SCPTs.

2.4.3 Sensor objects

The sensor objects are general LonMark® objects for use with any sensor for any physical measurements such as temperature, pressure, humidity and even for digital values from monitoring devices and switches.

The data can be transmitted directly to an actuator node or a controller node using the output network variable ‹nvoValue› of the type ‹SNVT_xxx›. There are 2 types of sensor objects: the ‹Open Loop Sensor Object› of object type #1 and the ‹Closed Loop Sensor Object› of object type #2. The two are distinguished by the non-availability or availability of feedback NVs in the object. Figure 2-4 shows the structure of the sensor object without feedback.

The measured value sent out using the NV ‹nvoValue› can be converted from the data logging component of the application program into the correct physical value. Where applicable, the required linearisation can also be completed there. Similarly, the conversion and linearisation of the raw measurement value can be completed using the configuration parameters ‹Translation Table X› and ‹Translation Table Y›.

Indeed the value range of the SNVT used displays a minimum and maximum value, but where required the value range can also be limited using the configuration parameters ‹Min Range› and ‹Max Range›. It is recommended to make an appropriate choice of the configuration parameter ‹Send on Delta›, which determines the size of the change in the sensor value. Only upon reaching this is an NV update sent out. The "heartbeat" function is used by the parameter specification for ‹Max Send Time›. On the other hand, the update rate should also be limited by specifying a value for ‹Min Send Time›. The default values are determined depending on the bit rate of the transmission medium: This is 1 s at 1.25 mbit/s and 60 s at a bit rate of 2 kbit/s.

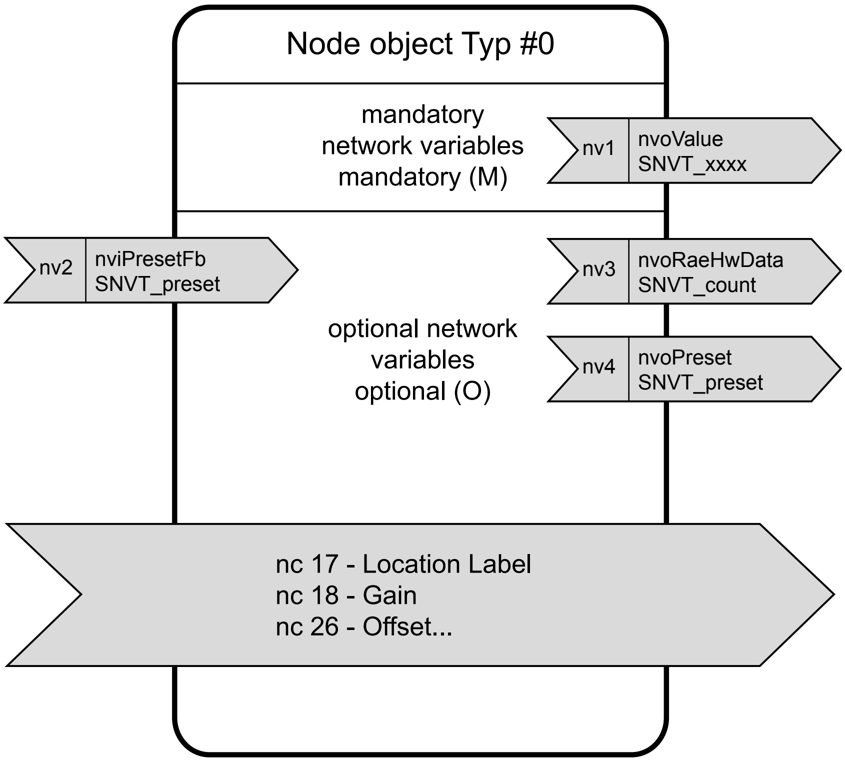


Figure 2-4: Structure of the Open Loop Sensor Object (Object Type #1)

The sensor object with feedback in Figure 2-5 is suitable for applications in which multiple sensors have to work together in any combination with multiple actuators,multiple sensors with one actuator or one sensor with multiple actuators. The same information must be present at all end points. This is the case, for example, if a lighting system can be switched on and off from different positions and there is no visual contact. In Figure 2-5 the essential difference can be identified with the no-feedback sensor object: the additional NV ‹nviValueFb› (Fb ~ Feedback).

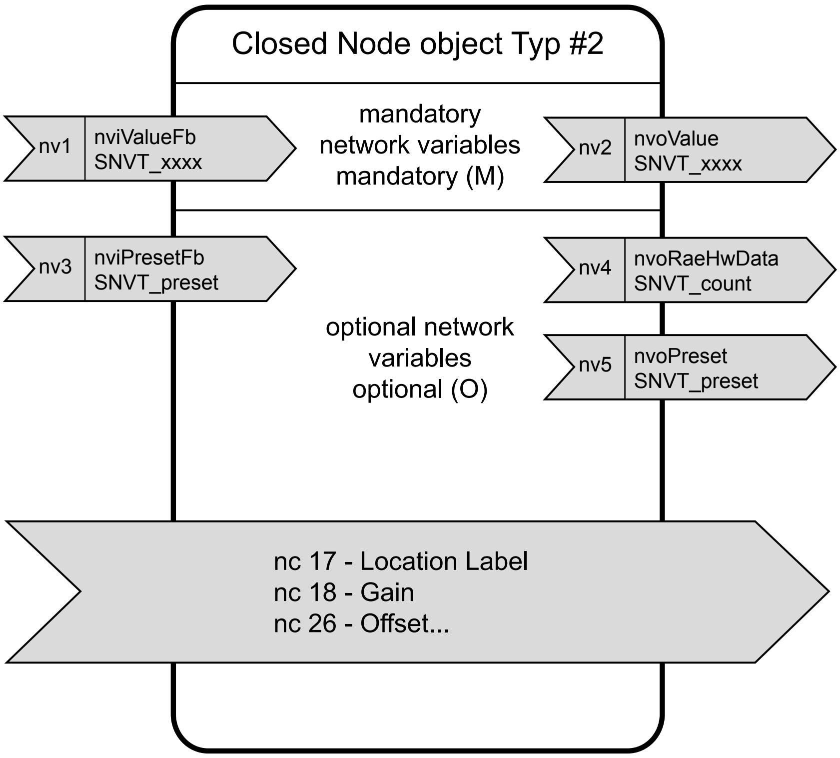


Figure 2-5: Structure of the Closed Loop Sensor Object extract (Object Type #2)

There are two typical methods for linking between feedback sensor objects and corresponding actuator objects (Figure 2-6). Based on method 1 the output variable of the actuator object, ‹nvo-ValueFb›, is returned to the ‹nviValueFb› inputs of the sensor objects. The current actuator status is not, however, reported back, instead the value specified in relation to ‹nviValue›. In the case of the second method, the variables ‹nvoValue› of the sensor objects are returned to the ‹nviValueFb› inputs for the sensor objects. This method involves a lower network load and works with smaller delay times.

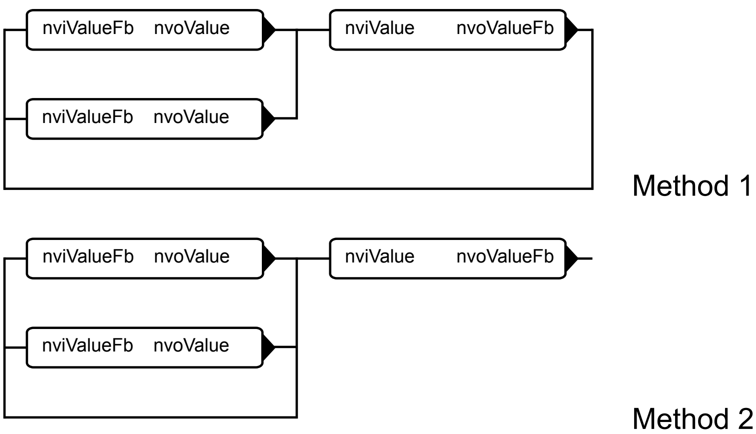


Figure 2-6: Coupling variants between sensor and actuator objects for feedback [6]

2.4.4 Actuator objects

The actuator objects (Open and Closed Loop Actuator Objects) bear object type numbers 3 and 4. These are also defined as general objects and can therefore also be used in motor control units and for the fan drive system or completely different types of final controlling elements. Figure 2-7 shows the structure of the actuator object with feedback. This differs from the actuator object without feedback only by the additional use of the NV ‹nvoValueFb› and of configuration parameter No. 15 - Input ‹Value Feedback Delay›. The feedback is used to synchronise between the actual and desired value.

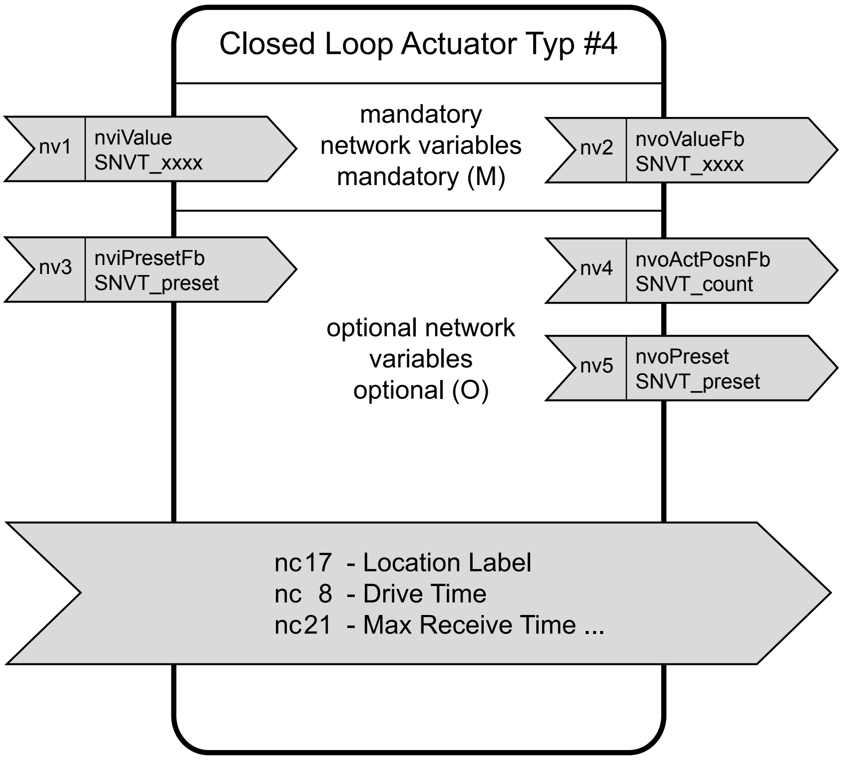


Figure 2-7: Structure of the actuator object with feedback (object type #4)

2.4.5 The controller object

An application will only manage by itself in exceptional cases with sensor and actuator objects. In practice, the processing algorithms for sensor data are more complex than is the direct conversion of a new sensor value to an actuator response. Comparing a temperature actual value with a temperature nominal value already requires a complex processing algorithm. If the actuator has to minimise the difference by influencing a heater, this is a classic case of a controller. The controller object is defined for this and other application scenarios (Figure 2-8).

The controller object has (as a generic object) any number of network input and network output variables, which should be grouped into send and receive variables to gain a better understanding. Thus (thankfully) a sender and a receiver section is created on the controller object. The NVs of the sender section are connected with corresponding NVs for an actuator object and the NVs for the receiver section are bound with corresponding NVs for a sensor object (Figure 2-8). The NV ‹nviValue-Fb› is used together with actuator objects with feedback. The NV ‹nvoValueFb› is only used together with sensor objects with feedback and are therefore immediately updated if a new value is received by means of the NV ‹nviValue›.

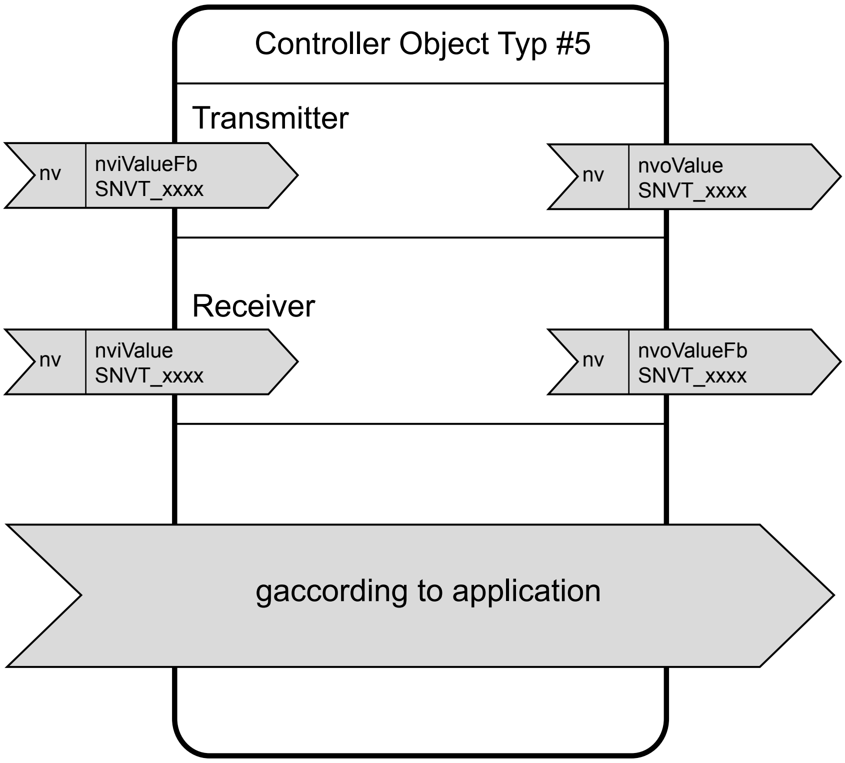


Figure 2-8: Structure of the controller object (object type #5)

An application-specific object is created from a generic LonMark® object by allocating a function profile. Should it be of interest for a large number of applications, this can be certified as a LonMark® object. The as yet unspecified configuration parameters in Figure 2-8 are selected first of all with reference to the sensor and actuator objects used, but secondly need to be controller-specific. In this way the controller object, if it is to be used as a PID controller, requests the control parameters proportional range, reset time and derivative time as well as a value for the sampling rate.

2.4.6 Function profiles

Function profiles have been derived from the basic objects, which are customised to specific applications. The function profiles are objects derived (inherited) from the basic classes, in which case the LonMark® objects correspond to the basic classes and the function profiles represent the derived subclasses. This architecture allows us to build networks in an object-oriented manner and to map them according to control computers to create configurable control systems.

The following function profiles have been defined (as of 06/06/1999):

LonMark® designation	German designation
Generic	General
Analogue Input	Analoger Eingang
Analogue Output	Analoger Ausgang
Sensors	**Sensors**
Light Sensor	Lichtsensor
Pressure Sensor	Drucksensor
Temperature Sensor	Temperatursensor
Relative Humidity Sensor	Feuchtigkeitssensor
Occupancy Sensor	Bewegungsmelder
CO2 Sensor	CO2-Sensor
Air Velocity Sensor	Durchflussmesser
Light Control	**Lichtsteuerung**
Lamp Actuator	Lighting
Constant Light	Controller Lichtregler
Occupancy Controller	Präsenzsteuerung
Room Control	**Raumsteuerung**
Switch	Schalter
Scene Panel	Raumpanel
Scene Controller	Raumkontroller / Regler
Partition Wall Controller	Zwischenwandkontroller
Time control	**Zeitbausteine**
Real Time Keeper	Echtzeituhr
Real Time Based Scheduler	Echtzeitschaltuhr
Motor Control	**Motorsteuerungen**
Variable Speed Motor Drive	Motorenantrieb

Continuation of the table ➔

2

HVAC	HLK
VAV controller (VAV)	Ventilationsregler
Fan Coil Unit (FCU)	Ventilationssteuereinheit
Roof Top Unit (RTU)	Dachabschlusseinheit
Chiller	Kühlung
Heat Pump with Temperature Control	Heizpumpe mit Temperaturregler
Thermostat	Thermostat
Chilled Ceiling Controller	Kühldeckenregler
Unit Ventilator Controller	Regler für Ventilationseinheit
Space comfort Control	
Command Module	Raumkomfortsregler Eingabemodul
Space Comfort controller	Raumkomfortsregler
Damper	Brandklappen
Damper Actuators (general purpose; fire and smoke airflow control)	Brandklappenantrieb
Refrigerating	Kältetechnik
Refrigerated Display Case	
Controller: Defrost Object	Defroster
Refrigerated Display Case Controller Evaporator Control Object	Verdampfer
Refrigerated Display Case Controller Thermostat Object	Thermostat
Fire Alarming	**Sicherheitstechnik**
Universal Fire Initiator	Fire alarm
Smoke (Intelligent) Fire Initiator	Smoke detector
Thermal Fire Initiator	Heat detector
Audible Fire Initiator	Alarmglocke
Visible Fire Indicator	Alarmanzeige
Universal Fire Indicator	Universelle Brandalarmanzeige
Power	**Energieerzeugung**
Generator Set	Generator
Utility	Energieverteilung
Utility Data Logger Register	Datalogger (Verbrauchsmessung)

Table 2-1: List of function profiles

3 The building blocks of the network

3.1 Nodes

The nodes have been examined in chapter 2.4.2. 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

• a 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 with 32,385 possible nodes, which is practically always reached. Consequently, some installation tools are very generous in how they handle the issuing of groups.

The group limitation is circumvented by forming a global domain, which contains all system-wide bindings. Local bindings are implemented in some domains so that the group area does not have to be exceeded.

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 in an existing network is not supported by all installation tools. It is therefore advisable not to fully utilise channels to ensure that a system can be upgraded according to requirement.

3.3.4 Limited number of address tables

The limitation to 15 address tables, which can only be exceeded for network management nodes, can lead to problems with first-generation tools. When selecting installation tools, where necessary, it is important to pay attention to whether "group overloading" is supported. All LNS tools support overloading and apply this automatically.

Where this is the case a group is divided into multiple subgroups, which work with the group 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.

4 Saia PCD® devices for LON networks

4.1 LON host module PCD7.F80x

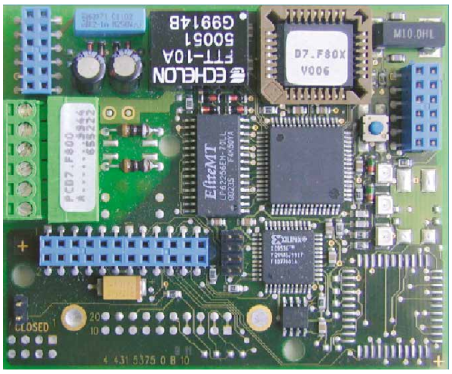


Figure 4-1: View PCD7.F804

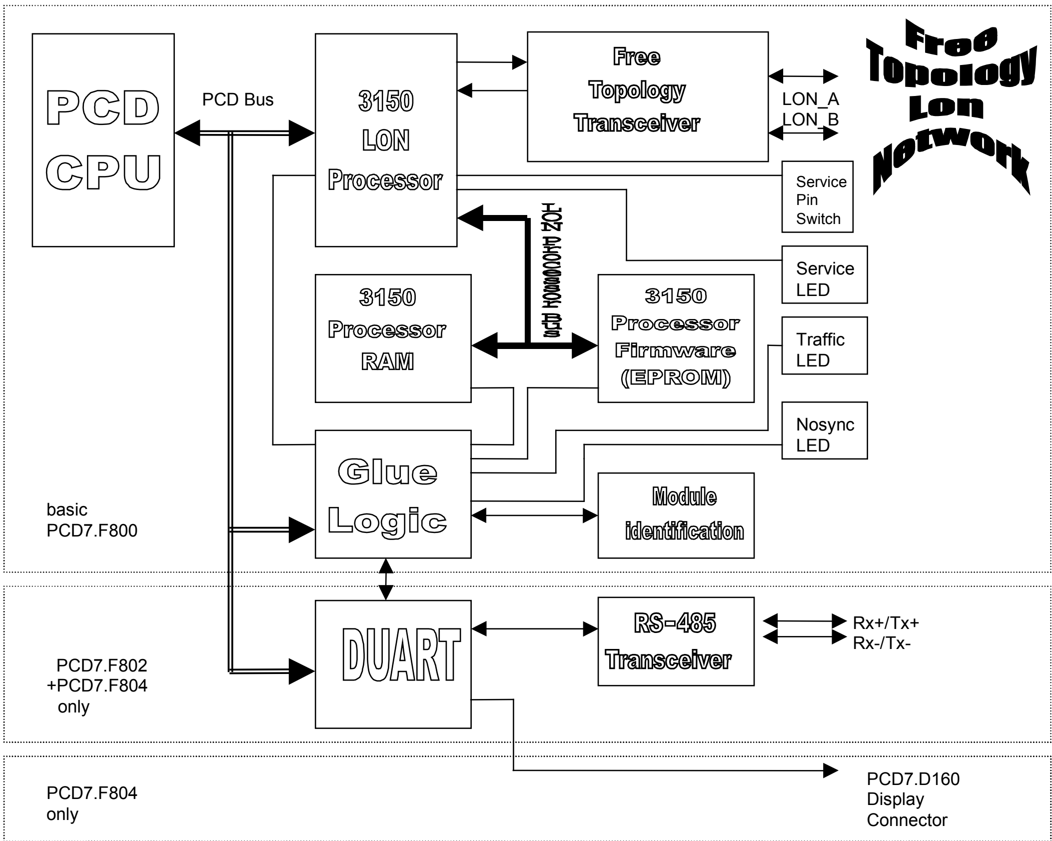


Figure 4-2: Block diagram PCD7.F80x

4.1.1 Available LON interface module

Module	Function	for
PCD7.F800	LON interface module for	PCD1.M120 / M130 PCD2.M120 PCD6.M300
PCD7.F802	LON Interface Module with interface 3, type RS-485, for	PCD2.M120
PCD7.F804 *)	LON Interface Module with interface 3, type RS-485 and connection for a PCD8. D160 terminal for	PCD1.M120 / M130 PCD2.M120

Table 4-1: Available LON interface module

*) Only available as terminal set PCD7.D165.
This set includes a plug-in terminal D160 with the additional RS-485 communication interfaces on Port 3 (not electrically isolated) and LON FTT10a interface.

For PCD1, Port 3 is not supported and consequently for the terminal the housing cover with groove, order number 4 104 7338 0, needs to be used.

4.1.2 Hardware and firmware versions

The LON interface module PCD7.F80x is supported by:

Saia PCD® System	HW		FW PDC 1/2/6	FW PCD7. F80x	Saia PG5®
	as from version	modification	as from version	as from version	as from version
PCS1.C88x	A	---	090	Internal	2.0.10
PCD1.M120/130	D	---	$63	LN0	2.0.10
PCD2.M120	J	---	$73	LN0	2.0.10
PCD2.M150	A	---	0A0	LN0	2.0.10
PCD2.M170	B	---	010	LN0	2.0.10
PCD2.M250	J *)	---	$73	LN0	2.0.10
PCD2.M480	---	---	---	---	---

Table 4-2: Hardware and firmware versions

*) Version of the PCD2.M15x board

Saia PCD® NT systems (PCD1.M2xxx, PCD2.M480, PCD2.M5xxx and PCD3) do not support the module PCD7.F80x !

Supported variables

Variables	Number	Size of the usable data
SNVT	max. 4095 per PCD *)	variable
Explicit message	max. 4095 per PCD *)	up to 50 bytes (LON MarkTM)

*) depending on the PCD memory

4.1.3 LON Controller

The LON controller MC143150 from ECHELON is used on the PCD7.F80x card. The firmware of the Neuron is saved in a socketed 32k EPROM. Working memory consists of one 256 kbit SRAM.

4.1.4 LON bus interface

The LON interface is equipped with a FTT_10A transceiver. The LON transceiver uses Manchester code for data transfer and can therefore also be used for AC coupling. (Equipment variant for the PCD7.F80x card) By default the PCD7.F80x card is supplied as an AC coupler.

4.1.5 AC/DC mode

The module is designed so that LPT mode is possible (see also section 1.5.4). AC coupling is the default. The LON module is supplied with voltage directly via the PCD. In AC mode the 48 VDC used for the LTP application are decoupled using two capacitors.

4.1.6 Transceiver specification

The installation regulations of the FTT_10A transceiver must be carefully observed. (see FTT_10A transceiver manual, available on the Internet from ECHELON "www.echelon.com"). To protect the transceiver, "spark gaps" have been used on the printed circuit board. (Specification: VDE (EN132 400, IEC384-12) rel.2 UL1414 and CSA C22.2 No.0;1).

4.1.7 Connecting the LON interface to the PCD1/2

In the case of the PCD1 and PCD2 the LON interface connection is made via the 6-pin connector on the PCD7.F80x module.

In the case of the PCD6.M3 the connection is made via the 9-pin D-Sub connector for port No. 3.

LON connection PCD1 / PCD2 with the modules PCD7.F80x

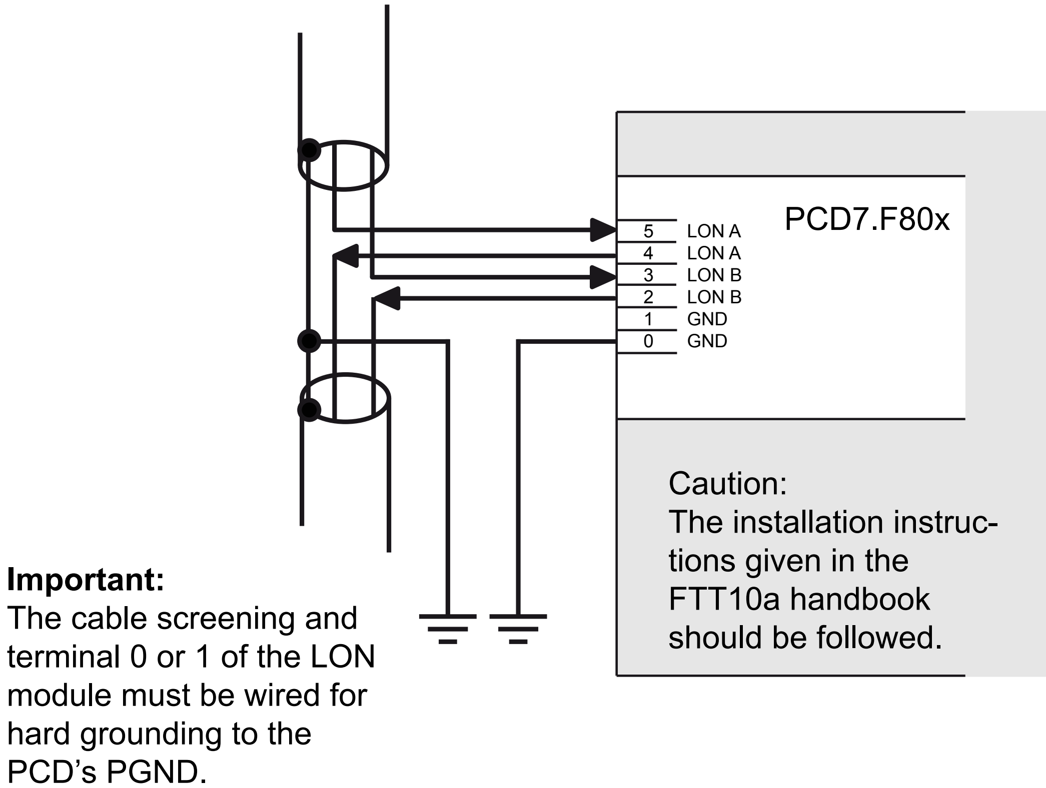


Figure 4-3: LON connection PCD1 / PCD2 with PCD7.F80x module

For a gentle earth, according to FTT10a HB, a 470 kΩ resistor to earth should be used or terminating components should be used.
(Can be found at: http://www.lontech.com).

4.1.8 Connecting the RS-485 interface to the PCD2

On a PCD2.M120 a RS-485 interface is also available in addition to the LON interface. This interface does not have electrical isolation. The terminating resistors can be activated using the jumper on the bottom left of the LON card.

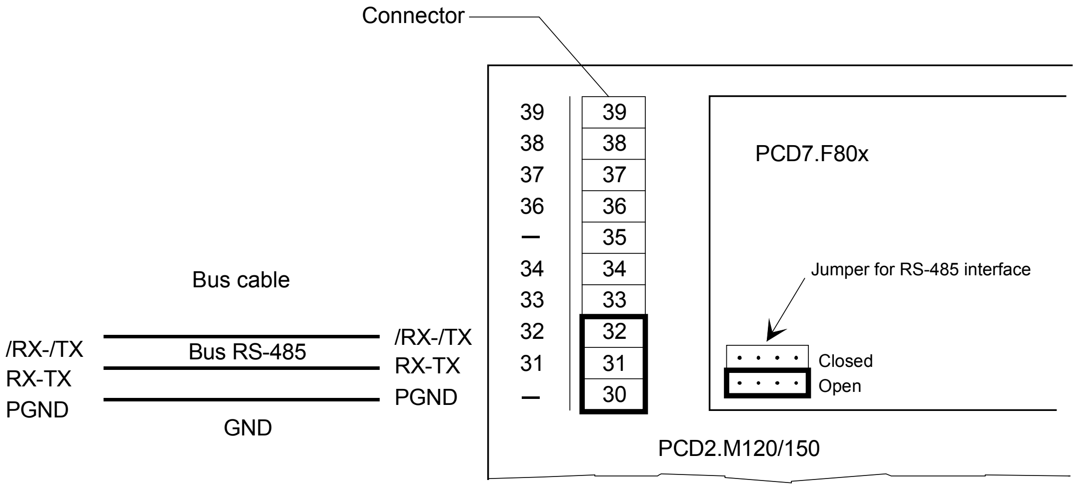


Figure 4-4: PCD2 Connecting interface 3, type RS-485

4.1.9 Terminating resistors

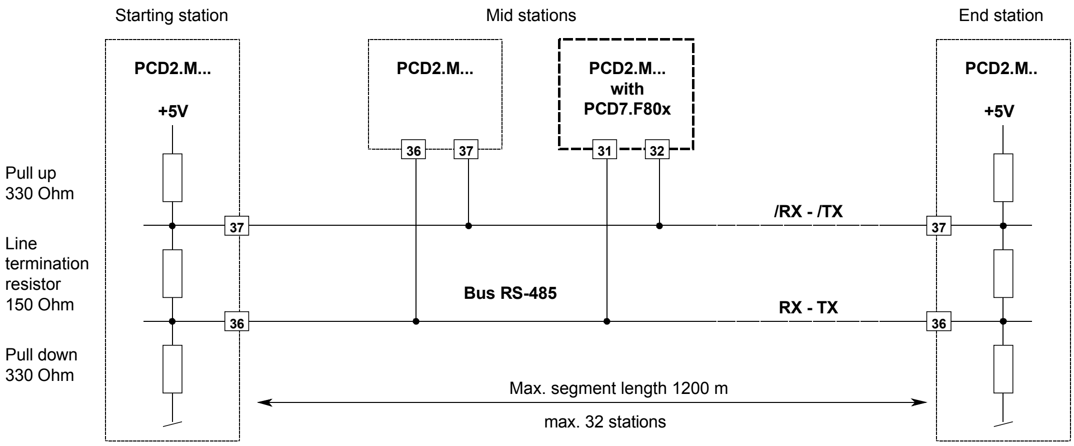


Figure 4-5: Terminating resistors

See manual 26/740 "Installation components for RS-485 networks"

4.1.10 Connections for LON

Signal	Clamp-type terminal PCD7. F80x	Clamp-type terminal CD2. M120
LON A	4 + 5	---
LON B	2 + 3	---
GND	0 + 1	--
\RX / \TX	--	32
RX / TX	--	31
GND	--	30

Table 4-3: Connections for LON

4.2 Operating modes

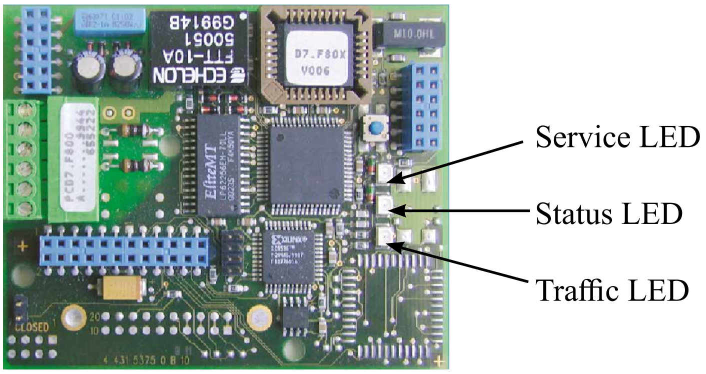


Figure 4-6: Layout of the LEDs on the module

The hardware for the interface consists of an individual circuit board, which is plugged into the main circuit board of the PCD. The module contains an FTT-10 transceiver, the bus connector and the Neuron® chip.

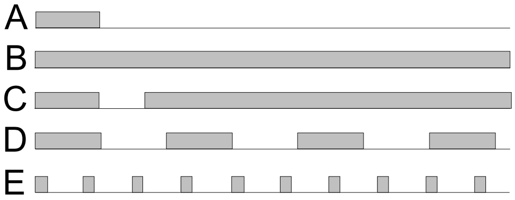
4.2.1 Meaning of the LEDs

The system status is displayed using 3 LEDs:

Service LED	displays the status of the Neuron® chip
Status LED	displays information about the status of the PCD driver
Traffic LED	displays information about data traffic

4.2.2 Behaviour of the Service LED

The Service LED is the Neuron® chip Service LED. The LED is illuminated when the module is in its reset status. The diagram set out below shows the behaviour for the possible statuses of the Service LED.



Service LED display		
A)	NORMAL OPERATI-ON	In this status the Neuron® chip is configured and works in synchronous mode with the PCD micro-processor. When switching on, the LED lights up for a few milliseconds. A second brief flashing may be observed when synchronising the module with the driver during the switch-on period.
B)	FATAL ERROR	For this status the LED is continuously red after being switched on. Where this is the case, the module will probably have to be replaced.
C)	APPLICATIONLESS	In this application status, the firmware of the Neu-ron® chip is faulty. The LED is illuminated for a period of 1 second after switching on, then goes out for 2 seconds and after this is continuously illuminated.
D)	UNCONFIGURED	In this status the installation tool has not yet configured the node or the configuration has been modified. This status is displayed by the LED flashing in a 2-second cycle. (1 second on, 1 second off).
E)	WATCHDOGING	In case of a software error (interface synchro-nisation error) the LED briefly flashes on every 750 ms. To rectify the cause of the error, a cold start needs to be completed on the PCD.

4.2.3 Behaviour of the Status LED

The Status LED displays the status of the PCD driver. The LED is illuminated when the module is in its reset status. The diagram set out below shows the behaviour for the possible statuses of the Status LED.

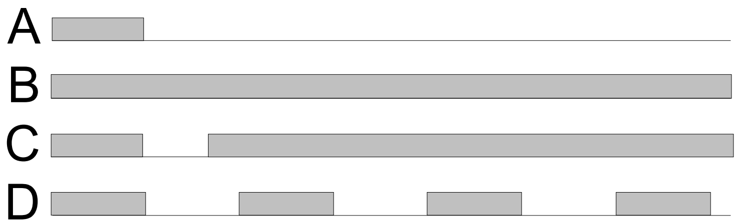


Diagram 4-2: Behaviour of the Status LED

Status LED display		
A)	Normal Operation	The LED lights up following a reset for a few 100 ms. This status shows that the interface is working normally and that the user program has correctly processed the data with the current configuration.
B)	Hardware error	The LED is continuously red if the PCD microprocessor is not able to correctly initialise the LON module. The cause of this behaviour can originate from a reset problem or a general module addressing error.
C)	Application	This status occurs when a user program is running in the PCD. After loading an application, the LED will adopt status A or D.
D)	Backup not up to date	After downloading all variable information from the PG4 into the PCD, the normal status of the Status LED is status "A". Status "D" indicates to the user that binding information has been written from the installation tool to the PCD node, i.e. new binding information has been written for one or more variables to the address table of the host node.

4.2.4 Behaviour of the Traffic LED

When transferring data via the interface of the Neuron® chip, the LED lights up for a min of 100 ms per telegram. A continuously illuminated LED means that at least 10 messages are being transferred per second. The interface can transfer up to 160 telegrams per second.

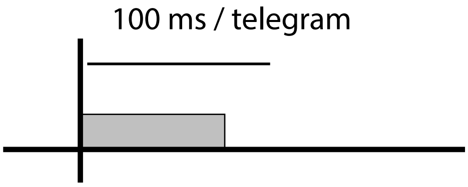


Diagram 4-3: Behaviour of the Traffic LED

5 Planning and installation of a LON network

All information for this chapter should be taken from the homepages listed below:

http://www.lontech.com/

http://www.echelon.com/

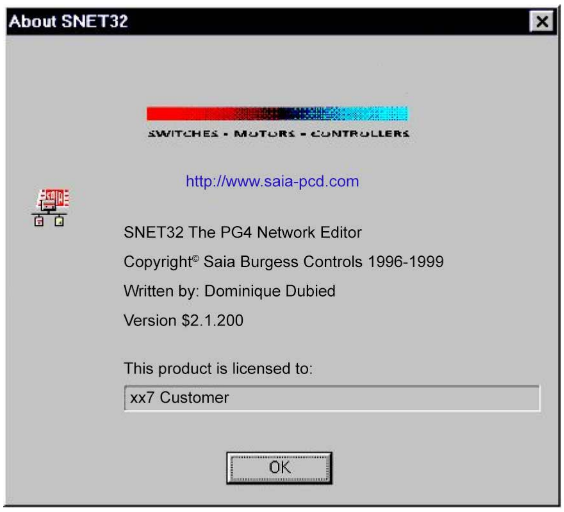
https://www.LonMark.org/

6 The LON configurator

The definition and configuration (bus parameters, network stations and definition of variables) of a LON host node can, depending on the size of the project, be quite extensive. This task is made significantly easier for the user by using the Saia PG5® LON configurator.

6.1 General

The SBC LON configurator is an add-on tool component in the Saia PG5® Engineering Suite, which runs under MS-Windows. The operating system can access data on a 32 bit or 64 bit system. No special hardware is required.



6.2 Procedure for configuration of the LON

The procedure can be broken down into the following steps:

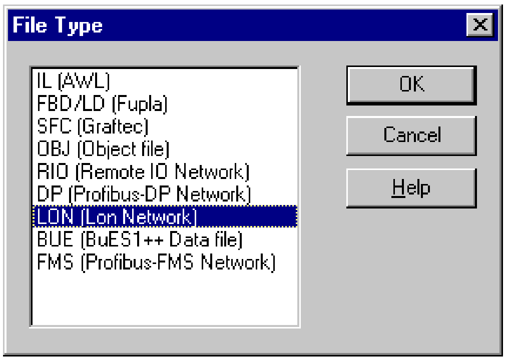
1. Start Saia PG5®
2. Defining a new project
3. Defining and retrieving a LON project in the Project Manager
4. Selecting the LON host node in the network configurator
5. Defining variables
6. Defining station parameters
7. Saving the configurator
8. Generating the documentation

6.3 Calling and description of the LON configurator

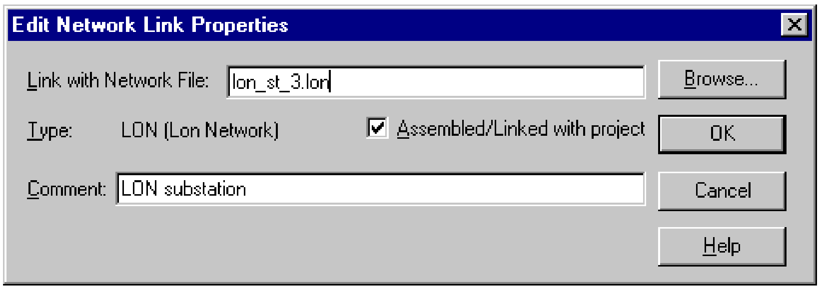
6.3.1 Opening a new project

After starting the Saia PG5® a new project is defined in the Project Library, or an existing project is opened, e.g. "LON_Demo".

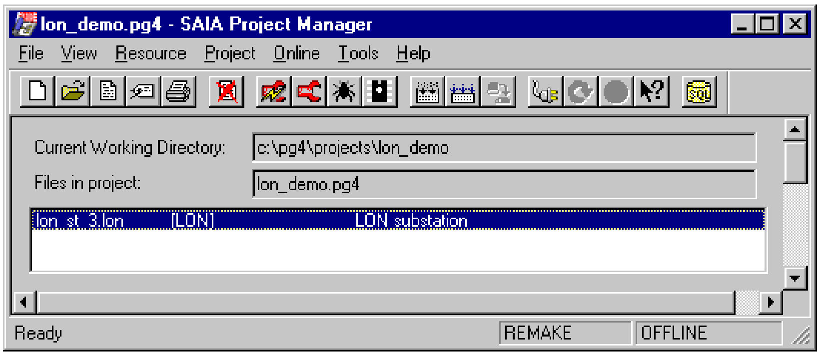
After double-clicking on the project, the Project Manager appears. The LON configuration file is opened in it. After selecting ‹File› - ‹New...›, selection of file types appears.



"LON (Lon Network)" should be selected. After pressing ‹OK› the following window appears, which should be filled in accordingly.

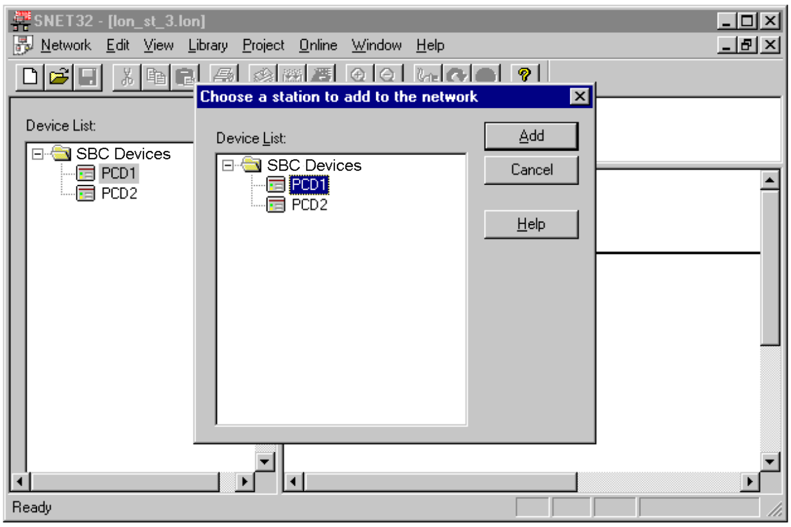


The Project Manager window is now displayed as follows:

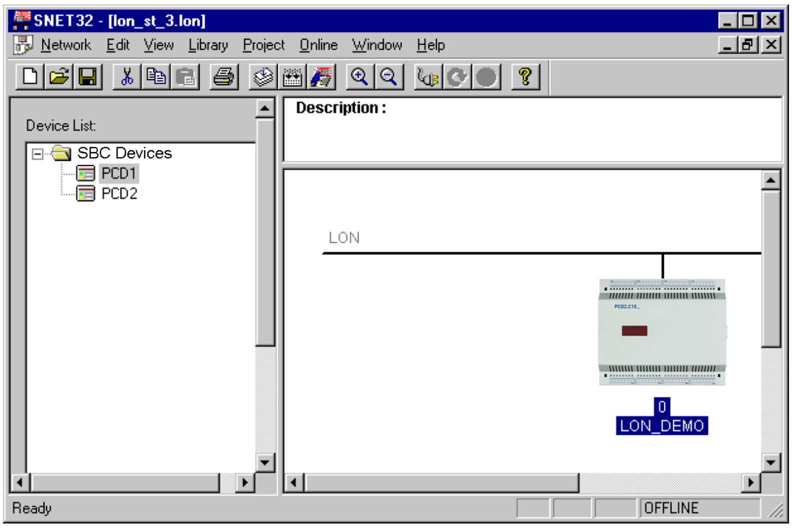


6.3.2 Structure of the main screen

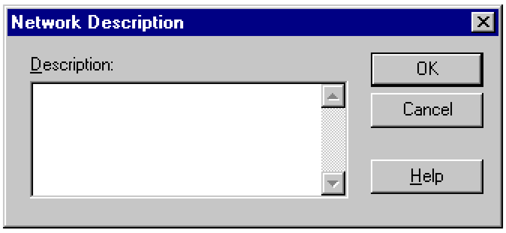
After double-clicking on the LON configuration file (lon_st_3.lon), the following windows appear:



In the active, smaller window the PCD being used as the LON host node should be selected from the "Device List" by double-clicking on it, at which point it is entered in the main window.

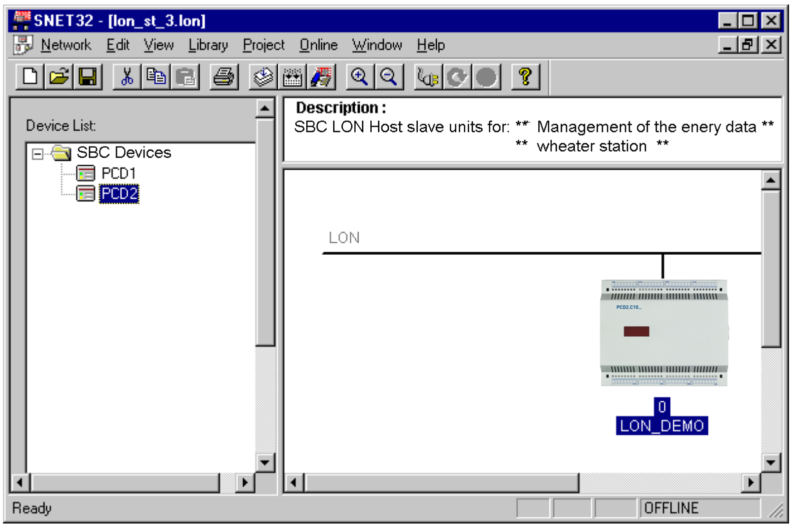


After double-clicking on the "Description:" field, the following window appears,



which can be filled in accordingly. After pressing ‹OK› the edited description text is entered in the "Description" field..

The main screen is now displayed as follows:

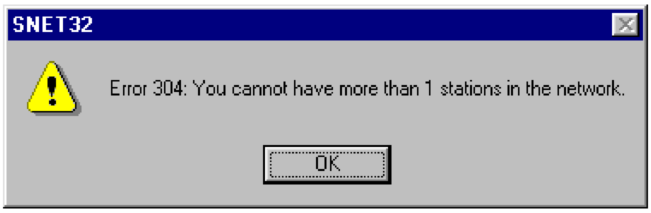


6

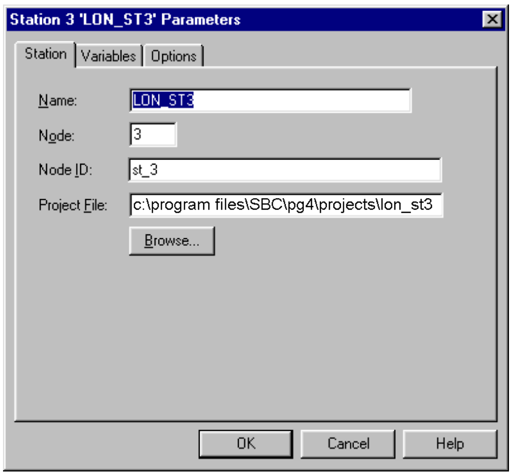
Action of the mouse buttons in the individual windows:	
‹Device List› window:	
Left mouse button	Double-clicking on the device adds it to the network.
Right mouse button	From the menu, open Insert Station / Add / Remove Device
‹Description› window:	
Left mouse button	Double-clicking on the description opens the input window for the network description.
‹Network› window:	
Left mouse button	Double-clicking on a device opens the parameter input for the device.
Right mouse button	From menu, open Parameter / Edit Project / Cut / Copy / Duplicate / Delete / Print.

! The functions that have just been listed can only ever be used for one LON node configuration as a maximum of one LON station can be configured in each project..

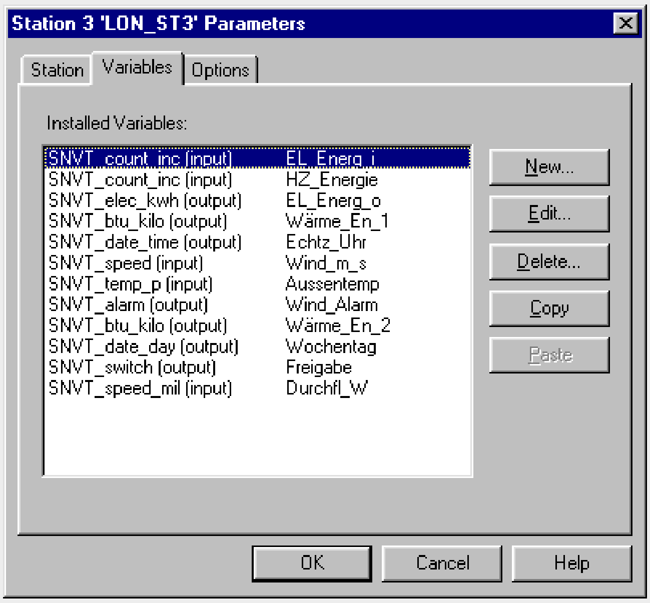
The following error message appears if the user would like to configure more than one station.



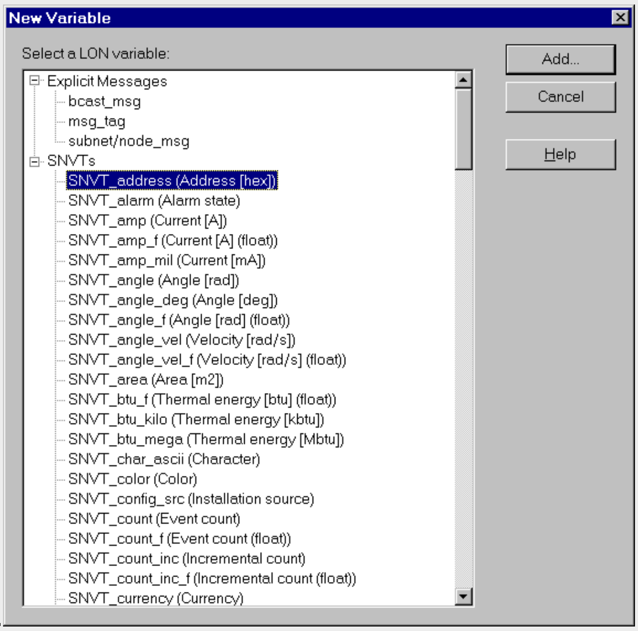
Station Parameters...		
Station:	Name:	The station name is entered here.
	Node:	This value remains set to "0" by default as the node address is assigned using the installation tool.
	Node ID:	Assign a name of a max. of 8 characters to identify the node in the installation tool.
	Project File:	Path assignment for saving the project file.



Variables: Definition of the SNVTs (standard network variables), which are to be used in the LON host node. All pre-defined LON host variables are listed in this window.



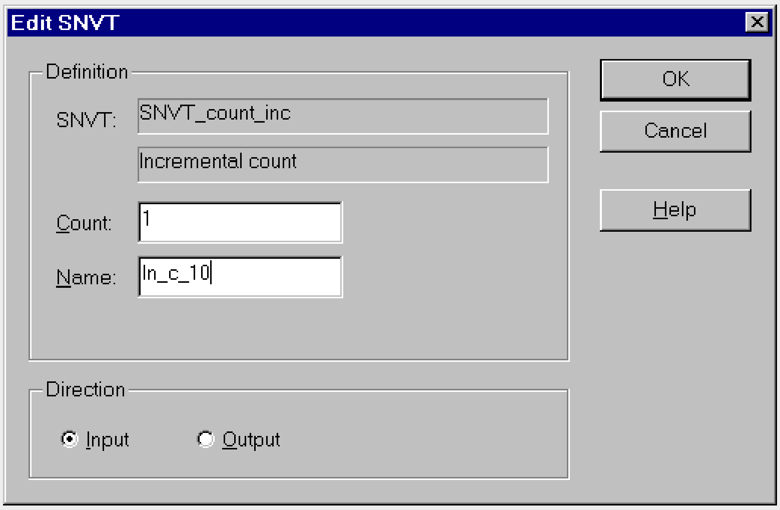
Using the "NEW" function it is possible to select a variable type from a list of all SNVTs specified in the LON Mark®



.

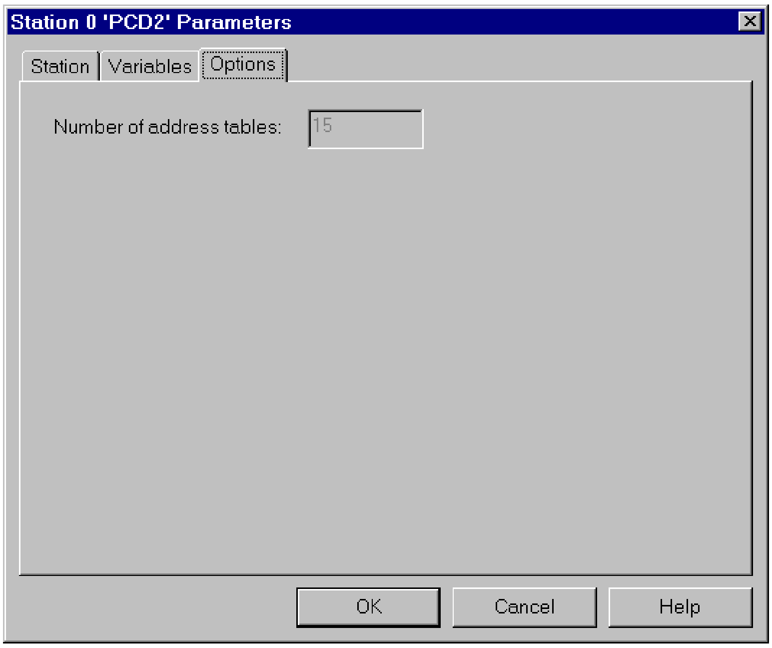
6

After selecting the SNVT type, a symbolic name of up to a max. of 10 characters can be assigned to the variable. It also has to be determined whether it is an input or output variable.



The "Count" option combined with the extensible FBoxes support configuration of multiple SNVTs of the same type in a single operation.

Options: This is a submenu, which enables the user to expand the current number of 15 address tables. As there is currently a limit of 15 address tables in LON Talk, the parameter remains unchanged set to zero.
(This upgrade was requested by ECHELON)

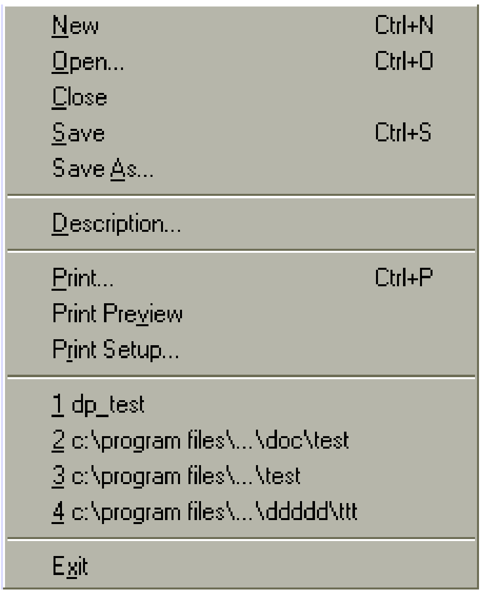


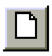
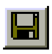
6

6.4 The LON configurator menus

The following submenus can be selected from the window menu:

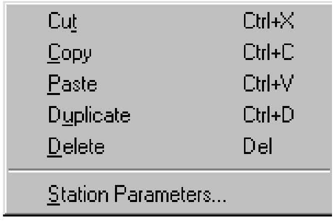
6.4.1 Structure of the ‹Network› submenu



'New'		A new project is opened up using this menu. In doing so there is an option to choose between a PROFIBUSDP, an SRIO and a LON network.
'Open'		Open an existing project.
'Close'		Close the active project.
'Save'		Save the active project under its current name.
'Save as...'		Save the active project under a new name.
'Description...'		Description of the project. This description can be seen on the main screen on the top right-hand side of the window.
'Print...'		Prints the configuration parameters of the project. The parameters can also be printed to an ASCII file.
'Print Preview'		Generates a print preview on the screen. All devices used, their settings and associated media are displayed in it.
'Print Setup...'		Settings for the printer type and paper format.
'1 .. 4'		Display the last 4 projects that have been edited.
'Exit'		Exit SNET.

6

6.4.2 Structure of the ‹Edit› submenu



'Cut'  Cut and save a selected LON node to the clipboard. In so doing the entire device configuration is copied, i.e. the installed variables are also copied.

'Copy'  Copy a selected LON node to the clipboard. In so doing the entire device configuration is copied, i.e. the installed variables are also copied.

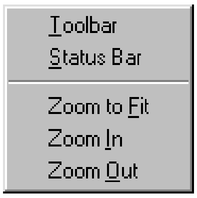
'Paste'  Insert a LON node, which is available on the clipboard, to the active project. In so doing the entire device configuration is copied, i.e. the installed variables are also copied.

'Duplicate' Create a copy of a selected LON node. Corresponds to the command sequence COPY and PASTE. In so doing the entire device configuration is copied, i.e. the installed variables are also copied.

'Delete' Delete a selected LON node.

6

6.4.3 Structure of the ‹View› submenu



'Toolbar' Display or hide the toolbar at the top edge of the screen.

'Status Bar' Display or hide the statusbar at the bottom edge of the screen.

'Zoom to Fit' With this option all devices available in the network are displayed on the screen.

'Zoom In'  Make the network screen content bigger.

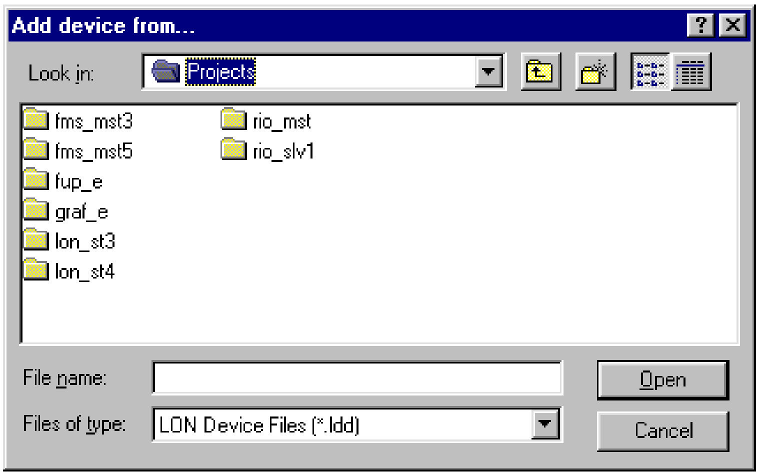
'Zoom Out'  Make the network screen content smaller.

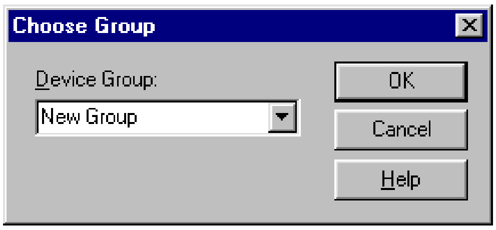
6.4.4 Structure of the ‹Library› submenu

'Add Device' Insert new LON nodes.

These nodes must have a file with the '.idd' extension. After selecting the '.idd' file, the device can be assigned to a device group.

In so doing, the device can either be saved to an existing group or to a new group.





To define a group, the new group name is simply entered in the input field. This new group is then automatically added to the 'Device' list.

'Remove Device' Delete a LON device node from the 'Device' list.

'Rename Group' Rename a 'Device' group. (is not active here).

6.4.5 Structure of the ‹Project› submenu

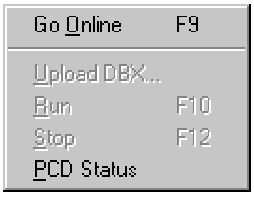
'Compile File' The specified project is compiled, i.e. the '.def' and '.src' files are created for the LON node.

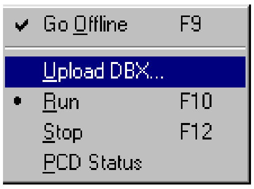
'Build' The selected project is assembled and linked together with the compiled configurator files.

'Edit Project' The Saia PG5® project manager for the selected project is retrieved.

6

6.4.6 Structure of the ‹Online› submenu





'Go Online' Makes the window go online.

'UploadDBX...' This function is used to save the binding information. All information, which has been saved by the binding tool in the host node, can be backed up over it and saved to the PC project-specifically.

Important: Before downloading new program information to the PCD, the binding information must be backed up via 'UploadDBX' as otherwise all bindings in the host node would be lost!

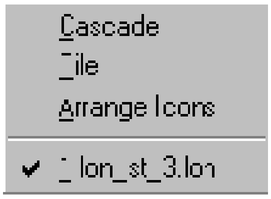
Important: After the variables (SNVT) have been associated for a LON project, it is important to complete a cold start so that the binding information is transferred from the memory of the LON module to the PCD memory.

'Run' if online, the CPU switches to Run.

'Stop' if online, the CPU switches to Stop.

'PCD Status' displays the CPU status.

6.4.7 Structure of the ‹Window› submenu



'Cascade'	All open projects are shown on the screen. They are displayed in an overlapping cascade so that every project title can be seen.
'Tile'	All open projects are shown on the screen. They are displayed in windows without projects overlapping
'Arrange Icons'	Rearrange the way all minimised projects are displayed.
'1 .. 10'	Selection list of all open projects.

6

6.4.8 Structure of the ‹Help› submenu

'Go Online'		Overview of the Help topics.
'Tile Using Help'		Description of how Help should be used.
'About Snet32'		Displays the version number and the name of the licensee.

7 Programming in the user program

Using LON variables (SNVT) in the PCD user program (FUPLA LON library)

7.1 Overview of the LON library

When working with this FBox library for the first time, it is recommended that you first read the following important topics:

Saia PG5® LON FBoxes and Saia PG5® LON Configurator Chapter 7.2.1

When searching for a particular SNVT FBox, the SNVT list helps:

SNVT List Chapter 7.2.2

SNVT Master List

Additional information about SNVT definitions, formats and structures can be found at:

http://types.LonMark.org

or as a PDF file with the name «SNVT Master List V11Rev2.pdf» at the PCD support page at:

http://www.sbc-support.com

The Auto Send mechanism Chapter 7.2.3

Use the ‹Help file› button to find the FBox description for this library.

If installed, the English and French Help files can also be retrieved.

7.2 Topics

7.2.1 Saia PG5® LON FBoxes and Saia PG5® LON Configurator

To connect the PCD variables to the LON network,

1. the LON variables need to be configured by the configurator
2. the correct FBox needs to be used in the Fupla and the PCD variables need to be excluded.

The LON configurator can be called from the Project Manager. The LON variables (SNVT) must be defined and the correct options selected. The SNVT name is being administered as a reference in Fupla.

Additional information can be found in the Help file of the LON configurator.

In Fupla the corresponding FBox has to be selected for each SNVT. The format and the send direction must be adapted in the configurator, according to the declaration.

- The binary SNVT require binary FBoxes.
- The numeric SNVT (1 to 4 bytes) require integer FBoxes.
- The SNVT with floating point require FBoxes with floating point format
- The PCD inputs require SND FBoxes.
- The PCD outputs require RCV FBoxes.

Click on the 'ref:' label of the FBox to insert the reference. In the reference field, specify the name declared in the LON configurator. In this way, the FBox will reach the corresponding SNVT.

Should you declare an SNVT table (array), a single name will be specified for all SNVTs for the table. An extendible FBox has to be used for an SNVT table. The SNVT name is automatically supplemented with an index. (Name00, Name01, Name02…).

If the FBox does not correspond with the referenced SNVT, assembler error messages appear.

7.2.2 SNVT List

The following list will help you to find the FBox, which supports a particular SNVT. Not all SNVTs are supported in this version. New SNVTs can be developed by SBC on request. To do this, a description on their use in the PCD needs to be added.

The SNVTs are assembled in FBoxes in accordance with value formats.

Examples: All SNVT with floating point format are supported by the FBox SEND floating point.

If the desired FBox has been selected, the corresponding help file can be used.

Group / SNVT Name	RCV FBoxes	SEND FBoxes
Binary group		
SNVt_switch	RCV Binary	SEND Binary
	RCV Binary+Rcv	SEND Binary Auto
	RCV Binary+Rcv_Wert	SEND Binary Snd
	RCV Binary Code	SEND Binary+Rcv-Wert
		SEND Binary Code Auto
Time group		
SNVT_time_stamp	RCV Date and time	SEND Date and time
Floating point group		
SNVT_amp_f	RCV floating point	SEND floating point
SNVT_count_f	RCV floating point	SEND floating point Snd
SNVT_count_inc_f		SEND floating point Auto
SNVT_volt_f		
Integer group		
SNVT_char_ascii	RCV integer	SEND integer
SNVT_count	RCV integer Rcv	SEND integer Snd
SNVT_count_inc		SEND integer Auto
SNVT_flow		
SNVT_flow_mil		
SNVT_freq_hz		
SNVT_freq_kilohz		
SNVT_freq_milhz		
SNVT_hvac_emerg		
SNVT_hvac_mode		
SNVT_lev_count		
SNVT_lev_disc		
SNVT_lev_percent		
SNVT_occupancy		

SNVT_temp		
SNVT_temp_p		
SNVT_time_sec		
Object group		
SNVT_obj_status	RCV Object Status	
SNVT_obj_request		SEND Object request
State group		
SNVT_state	RCV state	SEND state
Alarm group		
SNVT_alarm	Alarm RCV	Alarm SEND
Magnetic card group		
SNVT_magcrd	RCV magnetic card	SEND magnetic card
Group Settings		
SNVT_Setting	RCV Settings	SEND Settings

Table 7-1: SNVT list

7.2.3 The Auto Send mechanism

All FBoxes with auto-transmission mechanisms have similar initialisation options. The "Min" and "Max" parameters and the "Snd" and "En" inputs are described below.

Parameters	
Initialisation	Initialisation option Yes All values are transmitted for the PCD initialisation. No No transmission during initialisation.
Minimum change in value	The value is only transmitted if the change since the last transmission is greater than the parameter value. If the parameter is 0, the value is always transmitted. In the case of the simple binary FBoxes, this parameter is not implemented.
Minimum interval	A new value is only transmitted after the minimum interval. If the parameter becomes 0, the minimum change or the maximum interval is critical.
Maximum interval	The value is transmitted at least once after the interval even if the minimum value change has not been reached. If the parameter is 0, the function is disabled.

Initialisation

During PCD initialisation, data transmission is blocked for 2 seconds. This allows analogue values to stabilise before being transmitted. After this time:

- the values for inputs S0 ..S9 and V0..V9 are copied as the first reference values.
- the min. and max. interval is started.
- all values are copied at once, if the initialisation option is active.

Minimum / maximum interval and minimum value change

These 3 parameters make it possible to automate and optimise the transmission of the value.

Each function can be individually disabled by means of a zero setting. If the 3 parameters are 9, the value is never automatically transmitted. Only by enabling the input 'Snd' can a transmission be triggered.

Minimum interval
The minimum interval makes it possible to limit the number of telegrams if the value is changing too quickly (as a result network overloading is limited).

Maximum interval
The maximum interval alls for regular, forced transmission even if the value has not changed. This ensures that the recipient gets a value after switching off. Any lost telegrams are repeated.

Minimum change
The minimum change prevents the continuous transmission of values, which are hardly changing. For calibrated values, this parameter must be greater than the resolution, otherwise the same values can be transmitted as the change is monitored prior to calibration.

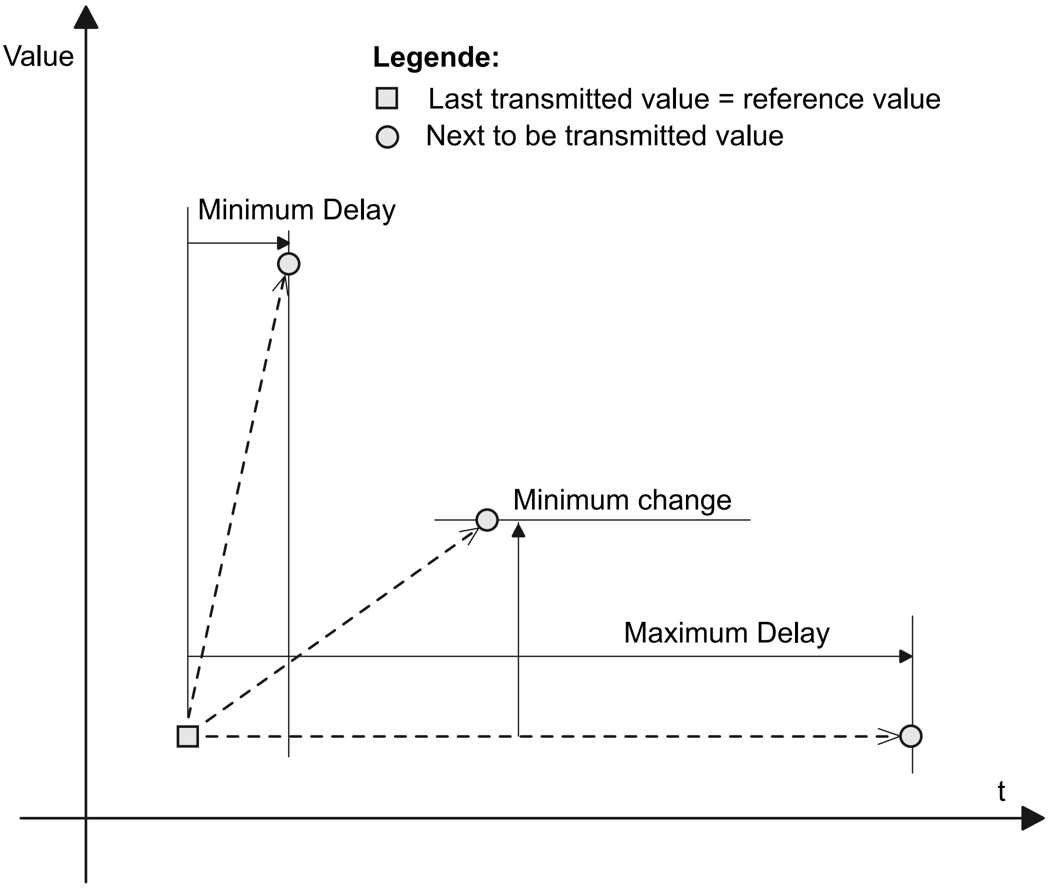


Figure 7-1: Minimum / maximum interval and minimum value change

Each function can be individually disabled by means of a zero setting. If the 3 parameters are set to 0, the value is never automatically transmitted. Only by enabling the input 'Snd' can a transmission be triggered.

<u>‹Snd› and ‹En› Inputs</u>

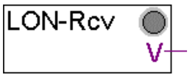
The ‹Snd› input supports a transmission even if the interval and the minimum change have not been reached.

If the ‹En› input is set to 0, a transmission cannot take place. This is advantageous for avoiding the transmission of incorrect values, during commissioning, and in the event of breakdowns or repairs.

7.3 SND and RCV Saia PG5® FBoxes

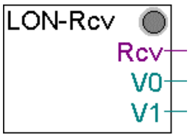
7.3.1 Binary

7.3.1.1 RCV Binary



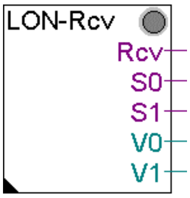
Supported SNVT	
SNVt_switch	

7.3.1.2 RCV Binary Rcv



Supported SNVT	
SNVt_switch	

7.3.1.3 RCV Binary + value Rcv

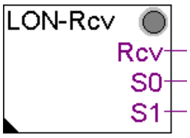


Supported SNVT	
SNVt_switch	

Outputs / LED		
Rcv	Reception	Switches to 1 for a cycle as soon as new data is received.
S0..S9	State	Binary state.
V0..V9	Unit	Numeric value. The starting value is converted depending on the selected range.
LED	LED	In case of a receive error, the LED lights up red.

Parameters	
Range of output signals	Range of output signals, corresponding 100%. The received LON value has a resolution of 0.5 %. Range 1000 means a resolution of 5 units. The LON value is in a range from 0 to 200 and corresponds to 0 to 100%. A range of 200 transmits the value to the LON network without conversion.

7.3.1.4 RCV Binary Code



Supported SNVT	
SNVT_temp_switch	

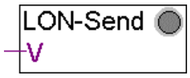
Outputs / LED		
Rcv	Reception	Switches to 1 for a cycle as soon as new data is received.
S0..S9	State	Binary state.
LED	LED	In case of a receive error, the LED lights up red.

Parameters	
Code for the OFF state	Code to be obtained for the OFF state, hex-coded
Code for the ON state	Code to be obtained for the ON state, hex-coded

Description

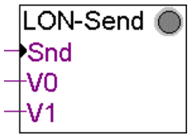
With this special FBox you can specify the received code for the ON and OFF state. It is therefore possible to adjust an SNVT_switch to a device, which does not have a standard function. The SNVT_switch has a code with 2 bytes. The highest byte is evaluated as a percentage (range 0 to 200). The lowest byte represents the binary state. If only the binary state is required, the standard code for the ON state is C801 Hex (means 100% ON) and the code for the OFF state is 0000 Hex. However the code 0001 Hex is used for the ON state on some devices.

7.3.1.5 SEND Binary



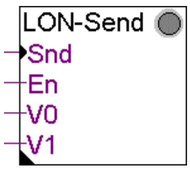
Supported SNVTs	
SNVt_switch	

7.3.1.6 SEND Binary Snd



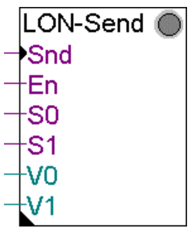
Supported SNVTs	
SNVt_switch	

7.3.1.7 SEND Binary Auto



Supported SNVTs	
SNVt_switch	

7.3.1.8 SEND Binary + Value Auto



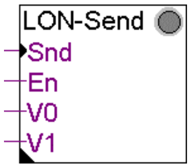
Supported SNVTs	
SNVt_switch	

Inputs / LED		
Rcv	Transmission	A pulse set to 'Snd' activates the transmission.
And	Activation	Activation of the transmission.
S0..S9	State	Binary state.

V0..V9	Value	Numeric value. The starting value is scaled depending on the selected range.
LED	LED	In case of a receive error, the LED lights up red.

Parameters	
Range of input signals	Range of input signals, corresponding 100%. The received LON value has a resolution of 0.5 %. Range 1000 means a resolution of 5 units. The LON value is in a range from 0 to 200 and corresponds to 0 to 100%. A range of 200 transmits the value to the LON network without conversion.
Other parameters	Further information can be found on the topic here: Automatic transmission

7.3.1.9 SEND Binary Code Auto



Supported SNVTs	
SNVT_temp_switch	

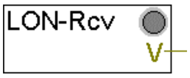
Inputs / LED		
Snd	Transmission	A pulse set to 'Snd' activates the transmission.
And	Activation	Activation of the transmission
S0..S9	State	Binary state.
LED	LED	In case of a receive error, the LED lights up red.

Parameters	
Code for the OFF state	Range of input signals, corresponding 100%. The received LON value has a resolution of 0.5 %. Range 1000 means a resolution of 5 units. The LON value is in a range from 0 to 200 and corresponds to 0 to 100%. A range of 200 transmits the value to the LON network without conversion
Code for the ON state	Code to be obtained for the ON state, hex-coded

Description
With this particular FBox you can specify the received code for the ON and OFF state. It is therefore possible to adjust an SNVT_switch to a device, which does not have a standard function. The SNVT_switch has a code with 2 bytes. The highest byte is evaluated as a percentage (range 0 to 200). The lowest byte represents the binary state. If only the binary state is required, the standard code for the ON state is C801 Hex (means 100% ON) and the code for the OFF state is 0000 Hex. However the code 0001 Hex is used for the ON state on some devices.

7.3.2 Integer

7.3.2.1 RCV integer

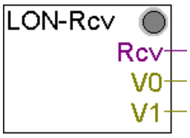


Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_amp_mil		SNVT_mass_kilo
SNVT_angle		SNVT_mass_meg
SNVT_angle_vel		SNVT_mass_mil
SNVT_btu_kilo		SNVT_occupancy
SNVT_btu_mega		SNVT_override
SNVT_char_ascii		SNVT_power
SNVT_config_src		SNVT_power_kilo
SNVT_count		SNVT_ppm
SNVT_count_inc		SNVT_press
SNVT_data_day		SNVT_press_p
SNVT_elec_kwh		SNVT_res
SNVT_elec_whr		SNVT_res_kilo
SNVT_flow		SNVT_rpm
SNVT_flow_mil		SNVT_sound_db
SNVT_freq_h		SNVT_speed
SNVT_freq_kilohz		SNVT_speed_mil
SNVT_freq_milhz		SNVT_telcom
SNVT_grammage		SNVT_temp
SNVT_hvac_emerg		SNVT_temp_p
SNVT_hvac_mode		SNVT_time_sec
SNVT_length		SNVT_vol
SNVT_length_kilo		SNVT_vol_kilo
SNVT_length_mic		SNVT_vol_mil
SNVT_length_mil		SNVT_volt
SNVT_lev_count *		SNVT_volt_dbmv
SNVT_lev_disc		SNVT_volt_kilo
SNVT_lev_percent *		SNVT_volt_mil
SNVT_lux		

* These SNVTs do not have a decimal resolution such as 1, 0.1, 0.01 or 0.001. Conversion of the values being transferred outside the FBox may be necessary.

SNVT	Resolution
SNVT_lev_cont	0.5
SNVT_lev_percent	0,005

7.3.2.2 RCV integer Rcv



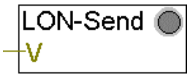
Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_amp_mil		SNVT_mass_kilo
SNVT_angle		SNVT_mass_meg
SNVT_angle_vel		SNVT_mass_mil
SNVT_btu_kilo		SNVT_occupancy
SNVT_btu_mega		SNVT_override
SNVT_char_ascii		SNVT_power
SNVT_config_src		SNVT_power_kilo
SNVT_count		SNVT_ppm
SNVT_count_inc		SNVT_press
SNVT_data_day		SNVT_press_p
SNVT_elec_kwh		SNVT_res
SNVT_elec_whr		SNVT_res_kilo
SNVT_flow		SNVT_rpm
SNVT_flow_mil		SNVT_sound_db
SNVT_freq_h		SNVT_speed
SNVT_freq_kilohz		SNVT_speed_mil
SNVT_freq_milhz		SNVT_telcom
SNVT_grammage		SNVT_temp
SNVT_hvac_emerg		SNVT_temp_p
SNVT_hvac_mode		SNVT_time_sec
SNVT_length		SNVT_vol
SNVT_length_kilo		SNVT_vol_kilo
SNVT_length_mic		SNVT_vol_mil
SNVT_length_mil		SNVT_volt
SNVT_lev_count *		SNVT_volt_dbmv
SNVT_lev_disc		SNVT_volt_kilo
SNVT_lev_percent *		SNVT_volt_mil
SNVT_lux		

* These SNVTs do not have a decimal resolution such as 1, 0.1, 0.01 or 0.001.
Conversion of the values being transferred outside the FBox may be necessary.

SNVT	Resolution
SNVT_lev_cont	0.5
SNVT_lev_percent	0.005

7.3.2.3 SEND integer

{button Übersicht,JumpID(`LON_Library_Overview›) {button Verwandte FBox-en,AL(«Integer»,0,`›,`›)}



Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_amp_mil		SNVT_mass_kilo
SNVT_angle		SNVT_mass_meg
SNVT_angle_vel		SNVT_mass_mil
SNVT_btu_kilo		SNVT_occupancy
SNVT_btu_mega		SNVT_override
SNVT_char_ascii		SNVT_power
SNVT_config_src		SNVT_power_kilo
SNVT_count		SNVT_ppm
SNVT_count_inc		SNVT_press
SNVT_data_day		SNVT_press_p
SNVT_elec_kwh		SNVT_res
SNVT_elec_whr		SNVT_res_kilo
SNVT_flow		SNVT_rpm
SNVT_flow_mil		SNVT_sound_db
SNVT_freq_h		SNVT_speed
SNVT_freq_kilohz		SNVT_speed_mil
SNVT_freq_milhz		SNVT_telcom
SNVT_grammage		SNVT_temp
SNVT_hvac_emerg		SNVT_temp_p
SNVT_hvac_mode		SNVT_time_sec
SNVT_length		SNVT_vol
SNVT_length_kilo		SNVT_vol_kilo
SNVT_length_mic		SNVT_vol_mil
SNVT_length_mil		SNVT_volt
SNVT_lev_count *		SNVT_volt_dbmv
SNVT_lev_disc		SNVT_volt_kilo
SNVT_lev_percent *		SNVT_volt_mil
SNVT_lux		

* These SNVTs do not have a decimal resolution such as 1, 0.1, 0.01 or 0.001. Conversion of the values being transferred outside the FBox may be necessary.

SNVT	**Resolution**
SNVT_lev_cont	0.5
SNVT_lev_percent	0.005

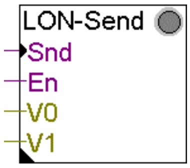
7.3.2.4 SEND integer Snd

Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_amp_mil		SNVT_mass_kilo
SNVT_angle		SNVT_mass_meg
SNVT_angle_vel		SNVT_mass_mil
SNVT_btu_kilo		SNVT_occupancy
SNVT_btu_mega		SNVT_override
SNVT_char_ascii		SNVT_power
SNVT_config_src		SNVT_power_kilo
SNVT_count		SNVT_ppm
SNVT_count_inc		SNVT_press
SNVT_data_day		SNVT_press_p
SNVT_elec_kwh		SNVT_res
SNVT_elec_whr		SNVT_res_kilo
SNVT_flow		SNVT_rpm
SNVT_flow_mil		SNVT_sound_db
SNVT_freq_h		SNVT_speed
SNVT_freq_kilohz		SNVT_speed_mil
SNVT_freq_milhz		SNVT_telcom
SNVT_grammage		SNVT_temp
SNVT_hvac_emerg		SNVT_temp_p
SNVT_hvac_mode		SNVT_time_sec
SNVT_length		SNVT_vol
SNVT_length_kilo		SNVT_vol_kilo
SNVT_length_mic		SNVT_vol_mil
SNVT_length_mil		SNVT_volt
SNVT_lev_count *		SNVT_volt_dbmv
SNVT_lev_disc		SNVT_volt_kilo
SNVT_lev_percent *		SNVT_volt_mil
SNVT_lux		

* These SNVTs do not have a decimal resolution such as 1, 0.1, 0.01 or 0.001.
Conversion of the values being transferred outside the FBox may be necessary.

SNVT	Resolution
SNVT_lev_cont	0.5
SNVT_lev_percent	0.005

7.3.2.5 SEND integer Auto



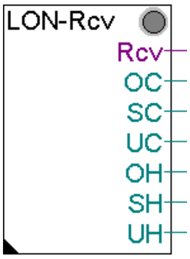
Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_amp_mil		SNVT_mass_kilo
SNVT_angle		SNVT_mass_meg
SNVT_angle_vel		SNVT_mass_mil
SNVT_btu_kilo		SNVT_occupancy
SNVT_btu_mega		SNVT_override
SNVT_char_ascii		SNVT_power
SNVT_config_src		SNVT_power_kilo
SNVT_count		SNVT_ppm
SNVT_count_inc		SNVT_press
SNVT_data_day		SNVT_press_p
SNVT_elec_kwh		SNVT_res
SNVT_elec_whr		SNVT_res_kilo
SNVT_flow		SNVT_rpm
SNVT_flow_mil		SNVT_sound_db
SNVT_freq_h		SNVT_speed
SNVT_freq_kilohz		SNVT_speed_mil
SNVT_freq_milhz		SNVT_telcom
SNVT_grammage		SNVT_temp
SNVT_hvac_emerg		SNVT_temp_p
SNVT_hvac_mode		SNVT_time_sec
SNVT_length		SNVT_vol
SNVT_length_kilo		SNVT_vol_kilo
SNVT_length_mic		SNVT_vol_mil
SNVT_length_mil		SNVT_volt
SNVT_lev_count *		SNVT_volt_dbmv
SNVT_lev_disc		SNVT_volt_kilo
SNVT_lev_percent *		SNVT_volt_mil
SNVT_lux		

* These SNVTs do not have a decimal resolution such as 1, 0.1, 0.01 or 0.001.
Conversion of the values being transferred outside the FBox may be necessary.

SNVT	Resolution
SNVT_lev_cont	0.5
SNVT_lev_percent	0.005

7.3.3 Temperature nominal values

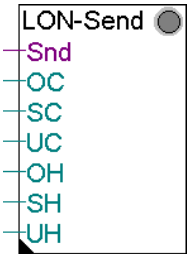
7.3.3.1 RCV Temp nominal values Rcv



Supported SNVTs	
SNVT_temp_setpt	

Outputs / LED		
Rcv	Reception	Switches to 1 for a cycle as soon as new data has been received.
OC	Occupied cold	---
SC	Standby cold	---
UC	Unoccupied cold	---
OH	Occupied hot	---
SH	Standby hot	---
UH	Unoccupied hot	---
LED	LED	In case of a receive error, the LED lights up red.

7.3.3.2 SEND Temp nominal values Snd



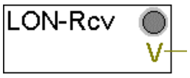
Supported SNVTs	
SNVT_temp_setpt	

Inputs / LED		
Snd	Transmission	A pulse set to 'Snd' activates the transmission.
OC	Occupied cold	---
SC	Standby cold	---

UC	Unoccupied cold	---
OH	Occupied hot	---
SH	Ready for use hot	---
UH	Unoccupied hot	---
LED	LED	In case of a receive error, the LED lights up red.

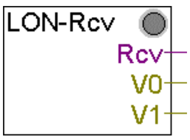
7.3.4 Floating point

7.3.4.1 RCV floating point



Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_amp_mil		SNVT_power
SNVT_sngle_vel_f		SNVT_ppm_f
SNVT_btu_f		SNVT_press_f
SNVT_count_f		SNVT_pwr_fact_f
SNVT_count_inc_f		SNVT_res_f
SNVT_density_f		SNVT_sound_db_f
SNVT_elec_whr_f		SNVT_speed_f
SNVT_flow_f		SNVT_temp_f
SNVT_freq_f		SNVT_time_f
SNVT_grammage_f		SNVT_vol_f
SNVT_lenght_f		SNVT_volt_f
SNVT_lev_cont_f		

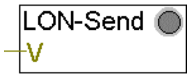
7.3.4.2 RCV floating point Rcv



Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_angle_f		SNVT_power
SNVT_sngle_vel_f		SNVT_ppm_f
SNVT_btu_f		SNVT_press_f

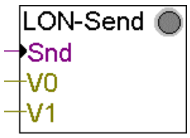
SNVT_count_f		SNVT_pwr_fact_f
SNVT_count_inc_f		SNVT_res_f
SNVT_density_f		SNVT_sound_db_f
SNVT_elec_whr_f		SNVT_speed_f
SNVT_flow_f		SNVT_temp_f
SNVT_freq_f		SNVT_time_f
SNVT_grammage_f		SNVT_vol_f
SNVT_lenght_f		SNVT_volt_f
SNVT_lev_cont_f		

7.3.4.3 SEND floating point



Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_angle_f		SNVT_power
SNVT_sngle_vel_f		SNVT_ppm_f
SNVT_btu_f		SNVT_press_f
SNVT_count_f		SNVT_pwr_fact_f
SNVT_count_inc_f		SNVT_res_f
SNVT_density_f		SNVT_sound_db_f
SNVT_elec_whr_f		SNVT_speed_f
SNVT_flow_f		SNVT_temp_f
SNVT_freq_f		SNVT_time_f
SNVT_grammage_f		SNVT_vol_f
SNVT_lenght_f		SNVT_volt_f
SNVT_lev_cont_f		

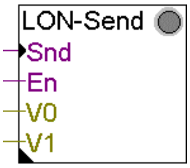
7.3.4.4 SEND floating point Snd



Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_amp_mil		SNVT_power
SNVT_sngle_vel_f		SNVT_ppm_f
SNVT_btu_f		SNVT_press_f

SNVT_count_f		SNVT_pwr_fact_f
SNVT_count_inc_f		SNVT_res_f
SNVT_density_f		SNVT_sound_db_f
SNVT_elec_whr_f		SNVT_speed_f
SNVT_flow_f		SNVT_temp_f
SNVT_freq_f		SNVT_time_f
SNVT_grammage_f		SNVT_vol_f
SNVT_lenght_f		SNVT_volt_f
SNVT_lev_cont_f		

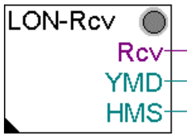
7.3.4.5 SEND floating point Auto



Supported SNVTs		
SNVT_amp		SNVT_mass
SNVT_angle_f		SNVT_power
SNVT_sngle_vel_f		SNVT_ppm_f
SNVT_btu_f		SNVT_press_f
SNVT_count_f		SNVT_pwr_fact_f
SNVT_count_inc_f		SNVT_res_f
SNVT_density_f		SNVT_sound_db_f
SNVT_elec_whr_f		SNVT_speed_f
SNVT_flow_f		SNVT_temp_f
SNVT_freq_f		SNVT_time_f
SNVT_grammage_f		SNVT_vol_f
SNVT_lenght_f		SNVT_volt_f
SNVT_lev_cont_f		

7.3.5 Date and time

7.3.5.1 RCV date and time



Supported SNVTs	
SNVT_timp_stamp	

Inputs / LED		
Rcv	Reception	Switches to 1 for a cycle as soon as new data has been received.
YMD	Year, Month, Day	PCD format. Depending on the selected option, the PCD clock is also set on receipt.
HMS	Hour, minutes, seconds	PCD format. Depending on the selected option, the PCD clock is also set upon receipt.
LED	LED	In case of a receive error, the LED lights up red.

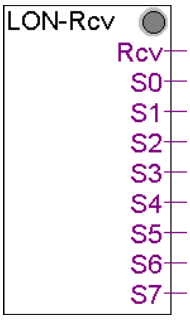
7.3.5.2 SEND date and time

Supported SNVTs	
SNVT_timp_stamp	

Inputs / LED		
Snd	Send	A pulse set to 'Snd' activates the trans-mission.
YMD	Year, Month, Day	PCD format. Depending on the selected option, the data is either read from this input or directly from the PCD clock.
HMS	Hour, minutes, seconds	PCD format. Depending on the selected option, the data is either read from this input or directly from the PCD clock.
LED	LED	In case of a send error, the LED lights up red.

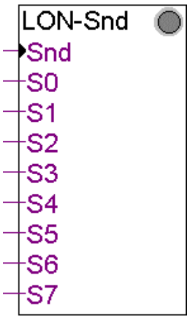
7.3.6 State

7.3.6.1 RCV state



Supported SNVTs	
SNVT_state	

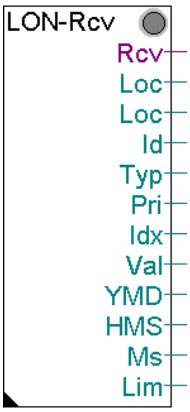
7.3.6.2 SEND state



Supported SNVTs	
SNVT_state	

7.3.7 Alarm

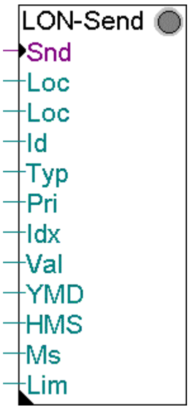
7.3.7.1 RCV Alarm



Supported SNVTs	
SNVT_alarm	

Inputs / LED		
Rcv	Reception	Switches to 1 for a cycle as soon as new data has been received.
Loc	Ort	4 bytes on the first output, 2 bytes on the second.
Id	Object ID	
Type	Alarm type	
Pri	Priority level	
Idx	SNCT index	
Val	Value	
YMD	Year, Month, Day	PCD- or LON-format, depending on the selected option.
HMS	Hour, minutes, seconds	PCD- or LON-format, depending on the selected option.
Ms	Milliseconds	
Lim	Alarm limit	
LED	LED	In case of a send error, the LED lights up red.

7.3.7.2 SEND Alarm



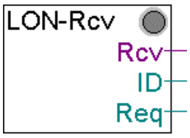
Supported SNVTs	
SNVT_temp_alarm	

Inputs / LED		
Snd	Transmission	A pulse set to 'Snd' activates the trans-mission.
Loc	Ort	4 bytes on the first output, 2 bytes on the second.

Id	Object ID	
Type	Alarm type	
Pri	Priority level	
Idx	SNCT index	
Val	Value	
YMD	Year, Month, Day	PCD- or LON-format, depending on the selected option.
HMS	Hour, minutes, seconds	PCD- or LON-format, depending on the selected option.
Ms	Milliseconds	The milliseconds are available in the PCD. The input can be set to K0.
Lim	Alarm limit	
LED	LED	In case of a send error, the LED lights up red.

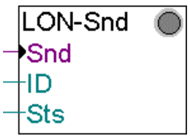
7.3.8 Object

7.3.8.1 RCV object status



Supported SNVTs	
SNVT_obj_status	

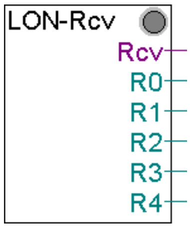
7.3.8.2 SEND object request



Supported SNVTs	
SNVT_obj_request	

7.3.9 Magnetic card

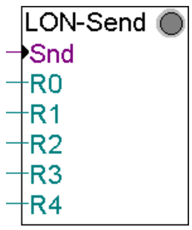
7.3.9.1 RCV magnetic card



Supported SNVTs	
SNVT_magcrd	

Inputs / LED		
Rcv	Reception	Switches to 1 for a cycle as soon as new data has been received.
R0..R4	Register 0 to 4	Register each for 4 send bytes. Total 20 bytes.
LED	LED	In case of a receive error, the LED lights up red.

7.3.9.2 SEND magnetic card

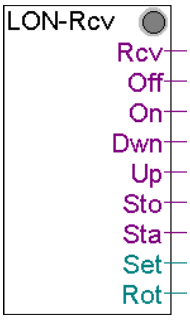


Supported SNVTs	
SNVT_magcrd	

Inputs / LED		
Rcv	Transmission	A pulse set to Snd activates the trans-mission.
R0..R4	Register 0 to 4	Register each for 4 send bytes. Total 20 bytes.
LED	LED	In case of a send error, the LED lights up red.

7.3.10 Settings

7.3.10.1 RCV Settings



Supported SNVTs	
SNVT_setting	

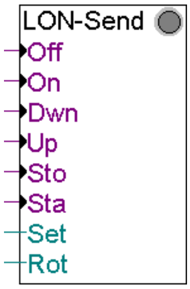
Inputs / LED		
Rcv	Received	Is set to 1 for a cycle when new data has been received. The received values are transmitted to the Set and Red outputs. If the received function is valid, the corresponding output is activated and all other outputs are set to zero. If the function code is invalid, all outputs are set to zero.
Off	Off	Receive Off function
On	On	Receive On function
Dwn	Down	Receive Down function
Up	Up	Receive Up function
Sto	Stop	Receive Stop function
Sta	State	Receive State function
Set	Setting	Receive Setting
Rot	Rotation	Receive Rotation
LED	LED	In case of a receive error, the LED lights up red.

As soon as new values have been received, the received setting and rotation values are transmitted to the corresponding outputs. If the receiving function is valid, the corresponding binary output is activated and all other outputs are set to zero. If the function code is invalid, all other outputs are set to zero.

The setting represents a value from 0 to 200 and corresponds to a level in ½ %.

The rotation represents a value of 1/100 degrees and extends from -359.98 to 360.00.

7.3.10.2 SEND Settings



Supported SNVTs	
SNVT_setting	

Inputs / LED		
Off	Off	Receive Off function
On	On	Receive On function
Dwn	Down	Receive Down function
Up	Up	Receive Up function
Sto	Stop	Receive Stop function
Sta	State	Receive State function
Set	Setting	Receive Setting
Rot	Rotation	Receive Rotation
LED	LED	In case of a send error, the LED lights up red.

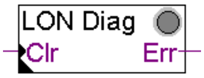
After receiving a positive edge, the corresponding function is transmitted with the setting and rotation value.

The setting represents a value from 0 to 200 and corresponds to a level in ½ %.

The rotation represents a value of 1/100 degrees and extends from -359.98 to 360.00.

7.3.11 Other Saia PG5® FBoxes

7.3.11.1 LON Diagnostics



Optional FBox to display the diagnostics of the LON network.

Error indication	Description
SASI LON	Fatal error, which may appear when assigning the LON channel. This is normally a firmware or hardware problem: - Incorrect PCD firmware - Incorrect firmware for the LON module - No LON module - Faulty LON module

Bit	Diagnostics flag	Description
0	Wink (flasher)	The message 'is flashing' is displayed when the Wink function is enabled.
1	Synchronisation	The flag is enabled when a synchronisation error appears when starting or during program execution.
2	Reception diagnostics	Is enabled when there is a receive error. Other information about possible errors are displayed in the diagnostics register (bits 0...15). As soon as the diagnostics register is deleted by the button displayed below, the flag is automatically set to zero.
3	Resetting the LON interface	The flag is enabled when the LON node is reset.
4	New binding	New binding (binding information) is loaded into the PCD. A restart is, however, required before loading the binding in the LON configurator.
5	Send diagnostics	Is enabled when a transmission error occurs. Other information about possible errors are displayed in the diagnostics register (bits 16...31). As soon as the diagnostics register is deleted by the button displayed below
6	Interface occupied	Is enabled during synchronisation with the LON network. Is automatically set to zero when synchronisation has terminated

7	Node Online	If the information about the LON module is wrong or the node has been switched to offline by the binding tool, ‹Not Online› is displayed. If the flag has been enabled, receiving or sending a polling on the SNVT network is not possible. If the message ‹Not Online› is displayed and the Service LED is regularly flashing, the LON node status must be switched to ‹Configured› by the binding tool.

Diagnostic Register	**Description**
Delete register	Diagnostics delete button.

Bit	**Diagnostics flag**	**Description**
0	LON interface faulty	Too many SNVT receives. Receipt of up to 80 SNVT (receive and transmit) possible simultaneously.
1	Data already on network	Is enabled when the PCD receives an SNVT, which has already been sent by the PCD without the confirmation of the LON network.
2	Obtain NAK when polling (SRXM)	Attempt of an SNVT polling, receipt of a NAK. Possible binding problem (no binding, incorrect binding), or no binding of the module to the LON network.
3	Not used	
4	Undefined SNVT	Is enabled when the PCD attempts to send an undefined SNVT. The LON configuration loaded in the PCD has nothing to do with the current project or is faulty. The project needs to be recompiled (by pressing the Rebuild all button) and reloaded into the PCD. *When this happens, please contact Saia Burgess Controls.*
5	Undefined SNVT type	Receipt of an SNVT of undefined type. The LON configurator needs to be updated.
6	SNVT DB is undefined	A diagnostic memory has been corrupted, deleted or is too small. The project needs to be recompiled and reloaded into the PCD. *When this happens, please contact Saia Burgess Controls.*

7	Diagnostics error	A diagnostic memory has been corrupted, deleted or is too small. The project needs to be recompiled and reloaded into the PCD. *When this happens, you must contact Saia Burgess Controls.*
8	SNVT value from the DB	Is enabled when the PCD receives an SNVT from the network, which is greater than the space reserved for it in the PCD memory. *When this happens, you must contact Saia Burgess Controls*
9	SNVT value from the network	Is enabled when the PCD receives an SNVT from the network, the value of which is different to that of the value reserved by the PCD memory. *When this happens, you must contact Saia Burgess Controls.*
10 .. 15	Not used	

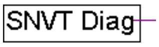
Transmission

Bit	Diagnostics flag	Description
16	LON interface faulty	Too many transmissions from SNVT. Receipt of up to 80 SNVT (receive and transmit) possible simultaneously.
17	Data already on network	Attempt at renewed variable transmission by the PCD. The previous transmission was not confirmed by the network. *When this happens, please contact Saia Burgess Controls.*
18	Obtain NAK during update (STXM)	NAK obtained during the update (STXM).
19	Not used	
20	Undefined SNVT	Is enabled when the PCD attempts to send an undefined SNVT. The LON configuration loaded in the PCD has nothing to do with the current project or is faulty. The project needs to be recompiled (by pressing the Rebuild all button) and reloaded into the PCD. *When this happens, please contact Saia Burgess Controls.*

21	SNVT of undefined types	Sending of an SNVT of undefined types. The LON configurator needs to be updated.
22	SNVT DB is undefined	Sending an SNVT to the network although there is no configuration for the affected SNVT. The project needs to be recompiled and reloaded into the PCD. *When this happens, please contact Saia Burgess Controls.*
23	Diagnostics error	A diagnostic memory has been corrupted, modified, replaced, deleted or is too small. The project needs to be recompiled and reloaded into the PCD. *When this happens, please contact Saia Burgess Controls.*
24	SNVT value from the DB	Is enabled when the PCD sends an SNVT over the network, which is greater than the memory capacity reserved in the PCD memory. *When this happens, please contact Saia Burgess Controls.*
25	SNVT value from the network	Is enabled when the PCD sends an SNVT over the network, the value of which does not match the memory capacity reserved for it in the PCD. *When this happens, please contact Saia Burgess Controls.*
26 .. 31	Not used	

For further information, please consult the LON manual.

7.3.11.2 SNVT Diagnostics



Optional FBox to be able to access the diagnostics of each FBox. The destination FBox is specified by means of the reference. The LED state is copied to the FBox output. Green=0, Red=1.

If an error is displayed, this is usually a problem in transmitting the value.

It should be noted that serious errors relating to the LON module, the configuration or the network connection are displayed in the LON diagnostics FBox.

8 Commissioning and Debugging

When setting up a LON project, a variety of difficulties might be encountered. Some of these points are listed in this section and might help the FUPLA programmer to find a solution.

The minimum requirements for setting up a LON project:

• LON module F80x

• PCD2 as from hardware index ‹J› with at least 4 MB internal RAM

• Firmware with LON capabilities for PCD1 or PCD2

• PG4 with LON functions (as from 2.0 with spec. SNET)

• PG4 and firmware must therefore be LON-compatible

8.1 History messages

Should it not be possible to switch a LON project to Run after being downloaded, checking the ‹History› might help to work out what the cause of the problem might be.

History / Message	**Cause**	**Remedy**
LON FAIL 000	Possible a syntax error in the SASI text «MODE:LON;CONF:DBXooO; DIAG:Faaa, Fbbb,Fccc,Rddd;» The SASI text can be displayed in the debugger: - Display program (SASI 9) - The Text number is in the 2nd row - The text xxx can now be displayed	Syntax should be checked for: MODE LON CONFIG DBX DIAG F and R
	DBX number too big	Requests for DBx limit (currently 4)
	DBX does not exist in the PCD memory	The configurator generated a DBx whose number does not match that of the SASI text. The configurator did not generate the DBx
	Flag/register addresses are outside the range	The range of the flags extends from 0 to 8191 and the register from 0 to 4095
LON FAIL 001	The user is attempting a SASI LON without a corresponding module on the PCD2	A LON module (F800/ F802/ F804) should be used in the PCD2.
LON FAIL 002	A data block provided in the LON firmware for the user does not exist or has not been correctly defined.	An SBC specialist needs to be engaged to resolve the problem.

LON FAIL 003	An SASI was executed during an internal LON configuration (firmware-specific).	This is a note. If this note appears more than once, the SBC office should be informed.
LON FAIL 004	The data structure in which the LON configuration has been saved (DBX) exists but has an incorrect identifier.	An SBC specialist needs to be engaged to resolve the problem.
LON FAIL 005	The configurator has generated a data structure (DBX), which is not compatible with the firmware.	The firmware and the software need to be updated to the most recent versions.
LON FAIL 006	■ The LON driver in the firmware generated an error. ■ MIP is sending an error code, when the PCD_CONFIG is called.	An SBC specialist needs to be engaged to resolve the problem.
LON FAIL 100	LON only runs with the new hardware from index 'J' and at least 4 MB RAM. If you attempt to run a program on an older PCD2 or on a PCD2 with less than 4 MB RAM, this error message appears.	The program has to be run with hardware that supports LON.

Table 9-1: History messages

These errors mainly relate to programming in IL (AWL). When using Saia PG5® LON FBoxes in FUPLA, error-free operation is guaranteed when used correctly.

8.2 Additional information on LON with Saia PCD®

Tips on binding network variables (binding):

Only one address table with a maximum of 15 entries is available to each LON node (Neuron chip). This also applies in respect of the Neuron chip on the PCD6.F8xx card, i.e. a node can directly address up to a maximum of 15 other nodes. This limitation can, however, be lifted using the binding option in the installation tool.

Depending on the type of binding network variables, there are different types of binding entries in this table. If it is necessary to communicate with as many other nodes of the PCD as possible, we recommend the ‹Broadcast› binding method.

If a binding is set up with a binding tool between two nodes using network variables, the following options should be set for this:

Broadcast, unacknowledged or
Broadcast, unacknowledged, repeated (higher bus load)

With the normal binding option, the entries in the address tables of the nodes are generally made by the binding tool with the ‹Sub-Net/Node› or ‹Group› option.

Connection option	Purpose of connection
SubNet / Node	Node A sends to node B
Group	Node A sends to node B, C and ... X
Broadcast	Node A sends to all nodes in the subnet

Table 9-2: Connection options

The ‹Group› binding option is automatically used by many binding tools, if a network variable is to be sent from one node to several other nodes.

NODE A:

Address table

Index	Type	Domain	Mbr/ Nod	Rpt tmr	Retries	Rcv tmr	Tx tmr	Grp / Sbnt
0	Sb/Nd	0	7	32	3	128	96	1
1	Gspz 3	0	0	32	3	768	96	0
2	Unused	0	0	16	0	128	16	0
3..13	Unused	0	0	16	0	128	16	0
14	Unused	0	0	16	0	128	16	0

Table 9-3: E.g. Knoten_A [Node_A] address table

Domain Table

Index	Size	Subnet	Node	Auth Key	Domn ID
0	1	1	3	FF FF FF FF FF FF	01

Table 9-4: E.g. Knoten_A [Node_A] Domain Table

Variables table

Index	Selctr	Dir	Prio	Auth	Addridx	Service	TrnArnd	Grp / Sbnt
0	0002	out	no	no	0	Ackd	no	1
1	0003	out	no	no	0	Ackd	no	0
2	0004	out	no	no	1	Ackd	16	0

Table 9-5: E.g. Knoten_A [Node_A] variables table

Saia PCD®:

Address table

Index	Type	Domain	Mbr/ Nod	Rpt tmr	Retries	Rcv tmr	Tx tmr	Grp / Sbnt
0	Sb/Nd	0	3	32	3	128	96	1
1	Sb/Nd	0	2	32	3	128	96	1
2	Gspz 3	0	1	32	3	768	96	0
3..13	Unused	0	0	16	0	128	16	0
14	Unused	0	0	16	0	128	16	0

Table 9-3: E.g. Knoten_A [Node_A] address table

Domain Table

Index	Size	Subnet	Node	Auth Key	Domn ID
0	1	1	7	FF FF FF FF FF FF	01

Table 9-4: E.g. Knoten_A [Node_A] Domain Table

Variables table

Index	Selctr	Dir	Prio	Auth	Addridx	Service	TrnArnd
0	0000	out	no	no	0	Ackd	no
3	0001	out	no	no	0	Ackd	no
7	0005	out	no	no	1	Ackd	no
8	0006	out	no	no	1	Ackd	no
9	000E	out	no	no	1	Ackd	no
16	000C	out	no	no	1	Ackd	no
17	000B	out	no	no	1	Ackd	no

Table 9-5: E.g. Knoten_A [Node_A] variables table

NODE B:

Address table

Index	Type	Domain	Mbr/ Nod	Rpt tmr	Retries	Rcv tmr	Tx tmr	Grp / Sbnt
0	Sb/Nd	0	7	32	3	128	96	1
1	Gspz 3	0	2	32	3	768	96	0
2	Unused	0	0	16	0	128	16	0
3..13	Unused	0	0	16	0	128	16	0
14	Unused	0	0	16	0	128	16	0

Table 9-3: E.g. Knoten_A [Node_A] address table

Domain Table

Index	Size	Subnet	Node	Auth Key	Domn ID
0	1	1	2	FF FF FF FF FF FF	01

Table 9-4: E.g. Knoten_A [Node_A] Domain Table

Variables table

Index	Selctr	Dir	Prio	Auth	Addridx	Service	TrnArnd
28	000F	out	no	no	0	Ackd	no
77	0010	out	no	no	0	Ackd	no
110	000D	out	no	no	0	Ackd	no

Table 9-5: E.g. Knoten_A [Node_A] variables table

The following binding structure can be read from the tables:

Red arrow	Green arrow	Blue arrow
Node A sends variables 1 and 2 to the PCD and variable 5 to node B.	The PCD sends variables 0 and 3 to node A and variables 7, 8, 9, 16 and 17 to node B.	Node B sends variables 28, 77 and 110 to the PCD.

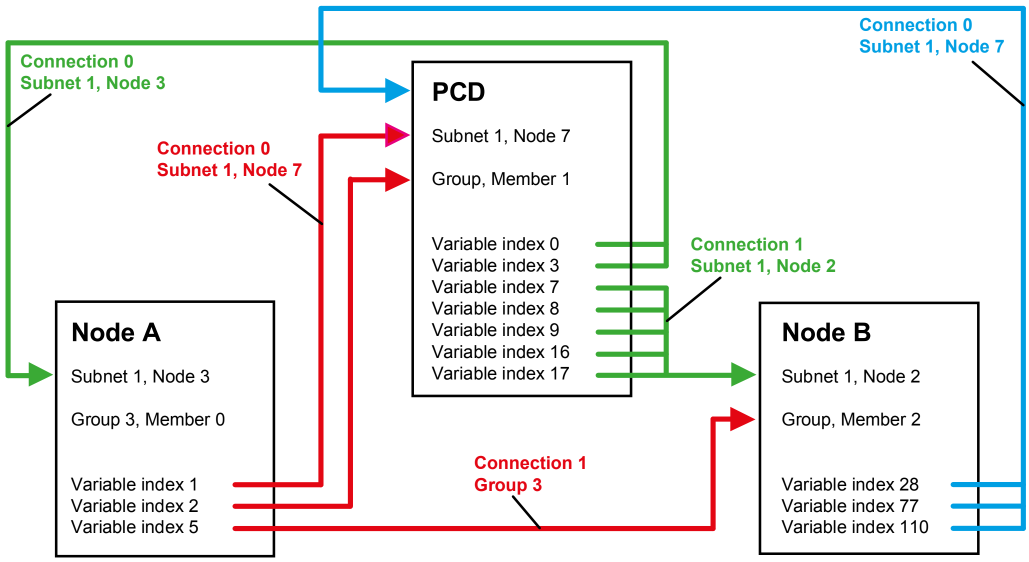


Figure 9-1: Binding structure

An illustration of the input variables has not been provided as only output variables generally need an address entry.

For all three nodes the common group address 3 has been entered in the address table although only node A sends anything via this binding. For group addresses an entry must be made in the address table in each node so that it can receive the sent data can.

In the case of the PCD a total of 3 rows have been used in the address table for this example. If you were now to add other Type-A nodes and in the same way were to bind with the PCD and node B, the address table in the LON node would quickly be populated.

Significance of the tables in the LON nodes:

Table	Meaning	Lines
Address table	List of usable connections	15
Domain Table	Some addresses in the node as subnet / node	1
Variables Table	List of all its own network variables and via which binding they are to be sent.	normal: 63 4096*) for PCD

Table 9-12: Significance of the tables in the LON nodes

*) The variables table is extended to 4096 by the MIP host node.

Effects of the ‹Broadcast› binding option on the binding:

NODE A:

Address table

Index	Type	Domain	Mbr/ Nod	Rpt tmr	Retries	Rcv tmr	Tx tmr	Grp / Sbnt
0	Unused	0	0	16	0	128	16	0
1	Unused	0	0	16	0	128	16	0
2		0	0	32	3	128	96	1
3..13	Unused	0	0	16	0	128	16	0
14	Unused	0	0	16	0	128	16	0

Table 9-13: E.g. Broadcast Knoten_A [Node_A] address table

Domain Table

Index	Size	Subnet	Node	Auth Key	Domn ID
0	1	1	3	FF FF FF FF FF FF	01

Table 9-14: E.g. Broadcast Knoten_A [Node_A] Domain Table

Variables table

Index	Selctr	Dir	Prio	Auth	Addridx	Service	TrnArnd	Grp / Sbnt
0	0013	out	no	no	2	Rep›td	no	1
1	0014	out	no	no	2	Rep›td	no	0
5	0015	out	no	no	2	Rep›td	no	0

Table 9-15: E.g. Broadcast Knoten_A [Node_A] variables table

Saia PCD®:

Address table

Index	Type	Domain	Mbr/ Nod	Rpt tmr	Retries	Rcv tmr	Tx tmr	Grp / Sbnt
0	Unused	0	0	16	0	128	16	0
1	Unused	0	0	16	0	128	16	0
2	Unused	0	0	16	0	128	16	0
3	Bcast	0	0	32	3	128	96	1
4..14	Unused	0	0	16	0	128	16	0

Table 9-16: E.g. Broadcast Knoten_A [Node_A] address table

Domain Table

Index	Size	Subnet	Node	Auth Key	Domn ID
0	1	1	7	FF FF FF FF FF FF	01

Table 9-17: E.g. Broadcast Knoten_A [Node_A] Domain Table

Variables table

Index	Selctr	Dir	Prio	Auth	Addridx	Service	TrnArnd
0	0011	out	no	no	3	Rep›td	no
3	0012	out	no	no	3	Rep›td	no
7	0016	out	no	no	3	Rep›td	no
8	0017	out	no	no	3	Rep›td	no
9	001B	out	no	no	3	Rep›td	no
16	0018	out	no	no	3	Rep›td	no
17	0019	out	no	no	3	Rep›td	no

Table 9-18: E.g. Broadcast Knoten_A [Node_A] variables table

8

NODE B:

Address table

Index	Type	Domain	Mbr/ Nod	Rpt tmr	Retries	Rcv tmr	Tx tmr	Grp / Sbnt
0	Unused	0	0	16	0	128	16	0
1	Bcast	0	0	32	0	128	96	1
2	Unused	0	0	16	0	128	16	0
3	Unused	0	0	16	0	128	16	0
4..15	Unused	0	0	16	0	128	16	0

Table 9-19: E.g. Broadcast Knoten_A [Node_A] address table

Domain Table

Index	Size	Subnet	Node	Auth Key	Domn ID
0	1	1	2	FF FF FF FF FF FF	01

Table 9-20: E.g. Broadcast Knoten_A [Node_A] Domain Table

Variables table

Index	Selctr	Dir	Prio	Auth	Addridx	Service	TrnArnd
28	001C	out	no	no	1	Rep›td	no
77	001D	out	no	no	1	Rep›td	no
110	001A	out	no	no	1	Rep›td	no

Table 9-21: E.g. Broadcast Knoten_A [Node_A] variables table

As can be seen, in each case only one entry is used in the address tables of the nodes. Using this broadcast addressing the information is sent to all nodes in the network. Both the send and recipient receive the entry of the broadcast address in their address tables. The previously used address entries are deleted for the binding and the next free entry is used. If a node simultaneously sends a network variable to multiple nodes, then an additional group entry is no longer required.

Distinct from the normal binding, in the case of the ‹Broadcast› binding option in the PCD now only one address entry is used, previously there were three. If a binding is made using the ‹Broadcast› binding option, then any number of nodes can communicate with a PCD.

Setting up the ‹Broadcast› binding option in the binding tool

Setting in the Pathfinder tool (version 1.5) from the company T-LON:

Figure 9-2
normal connection options
(Tool «Pathfinder»)

Figure 9-3
Broadcast connection options
(Tool «Pathfinder»)

Setting in the Alex tool (version 1.0) from the company Mentzel & Krutmann:

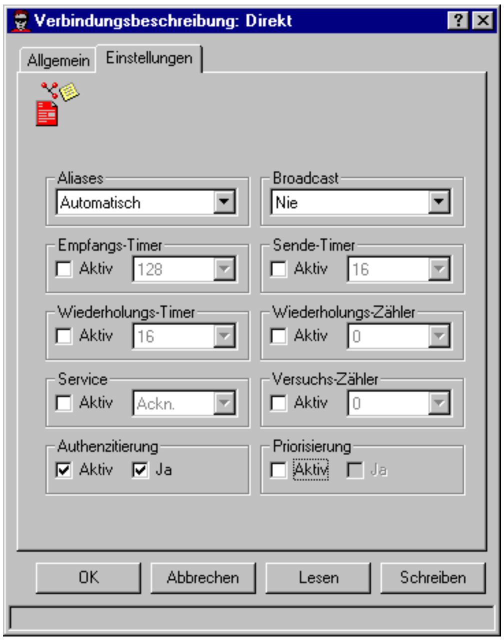


Figure 9-4
normal connection options
(Tool «Alex»)

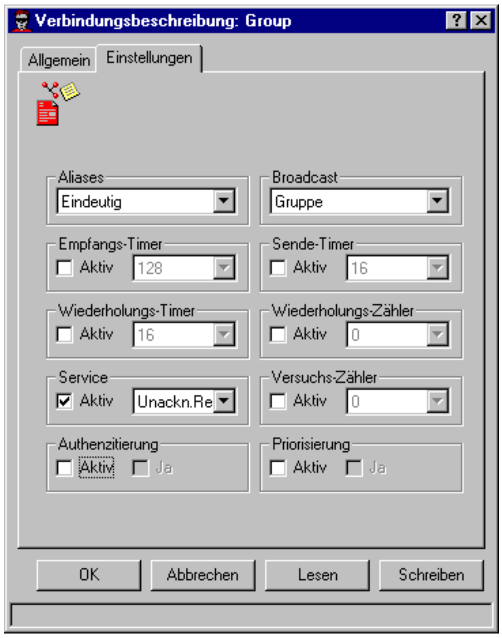


Figure 9-5
Broadcast connection options
(Tool «Alex»)

9 Terms, Abbreviations, List of References

9.1 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-RAM 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 variables 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›. ‹Broad-cast›- 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. Config-uration 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 2 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. De-clared ‹msg_tags› are always bidirectional.
Differential LON Interface	LON interface electrically isolated with an isolat-ing transformer on a 2-wire line. The transmis-sion 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 coordi-nate 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.
Explicit address	The address contained in the message, created and administered by the application (e.g. MIP).

9

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.
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 exchanges 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	Members 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 Lon Talk 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 access 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.
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 normal protocol over different 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.

LonManager	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 path for 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 subsystem.
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 the 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	Transport layer, which ensures destination addressing. 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 network-variable indexes to a selector. 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 uniquely occurring subnet/node combination. 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.

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 Transport layer, 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. This can contain up to a maximum of 127 nodes, a domain can contain 255 subnets.
subnet / node address	Standard address for 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	Transport 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.
Unprocessed netvar	Addressing the network variable by means of the ‹explicit address›, i.e. with address information delegated to the host application code.

9

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.

9.2 Abbreviations

AWL	Instruction List ➔ English IL
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
IL	Instruction List
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
LTM-10	Lon Talk Module. Hardware module by Echelon, which can be used as a development interface.
MIP	Microprocessor Interface Program
NIC	Network Interface Card
NSS-10	Hardware/firmware from Echelon. Module, which is suitable as a host interface with integrated network management.
OSI	Open Systems Interconnection
SCPT	Standard Configuration Parameter Type
SLTA	Serial LonTalk Adapter
SNVT	Standard Network Variable Type
TP	Twisted Pair

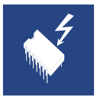
9.3 List of References

Book Title	Edition	Type of Book
LONTALK PROTOCOL	April 1993	LonWorks Engineering Bulletin
NEURON Chip-based 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
Die LonWorks®-Technologie ['The LonWorks® Technology']	1998	Tiersch F. LonTech® Thüringen e. V.ISBN 3-932875-03-6

9

A Appendix

A.1 Icons

	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 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.
	This symbol designates instructions, which need to be strictly followed.
	Explanations next to this symbol are only valid for the Saia PCD® Classic series.
	Explanations next to this symbol are only valid for the Saia PCD® xx7 series.

A.2 Books and home pages

- LonWorks® Installation Manual VDE Verlag ISBN 3800725754
- LonWorks® Planer Manual VDE Verlag ISBN 3800725991
- LonWorks- Technik in der Gebäudeautomation
  Huss-Medien GmbH
  Verlag Technik
  ISBN 3341013466
- Homepage von LonMark® www.lonmark.ch
- LonMark NVT Master List available from www.echelon.com

A

A.3 Contact

Saia-Burgess Controls AG
Bahnhofstrasse 18
3280 Murten
Switzerland

Phone .... +41 (0)26 580 30 00
Fax... +41 (0)26 580 34 99

Email support: .... support@saia-pcd.com
Supportsite: .... www.sbc-support.com
SBC site: .... www.saia-pcd.com
International Represetatives &
SBC Sales Companies: .... www.saia-pcd.com/contact

Postal address for returns from customers of the Swiss Sales office

Saia-Burgess Controls AG
Service Après-Vente
Bahnhofstrasse 18
3280 Murten
Switzerland