An Introduction to control networks based on

LONWORKS® Technology

Presented by

EBV Elektronik GmbH & Co KG
Im Technologiepark 2-8, D-85586 Poing

Version 5.11 / July 2004
Updated by Philippe Bénédet
From an initial idea by Franz Wiedemann

Content: page

Echelon Corporation

A leader in control networks used by over 4000 OEMs with 30 million nodes installed worldwide that deliver efficiency, safety and productivity to business, home, and manufacturing environments.

The developer of the LONWORKS® control network family of software, hardware, and technology.

A company whose value proposition to end-users and integrators is lower cost of ownership and increased functionality and flexibility through open, inter-operable, multi-vendor control solutions.

A global leader in the building controls industry supported by leading suppliers in every facet of building automation – including access control, HVAC, security, elevators, CCTV, and lighting.

An emerging force in the transportation industry, with successful applications in aviation, rail transportation, luxury coaches, and emergency vehicle systems.

A partner with electric utilities worldwide in offering demand side management, automatic meter-reading, appliance control, and other value-added services.

A global leader in industrial sensors and field bus applications as diverse as brewing, textile dyeing, printing, catering, and semiconductor manufacturing.

Some definitions

LON stands for **L**ocal **O**perating **N**etwork.

LonTalk is the embedded communication protocol in the NEURON Chip.

LonWorks is the name of the overall technology including NEURON chip, transceivers, protocol and software concepts.

LNS (**L**onWorks **N**etwork **S**ervices) is a standard software engine used to design custom applications able to install, manage and maintain LonWorks Networks.

LonMaker Integration Tool is a standard Network Management tool used to design, install and commission LonWorks networks.

***i*.LON** is a generic name for the Echelon internet servers or routers – product allowing to route messages or to access remotely to LonWorks devices through the Internet protocol.

Echelon's major investors/share holders

Mike Markkula (co-founder of Intel and Apple)

- Ken Oshman (CEO of Echelon)
- Detroit Edison
- Motorola
- ENEL
- Rock Arthur

Since 1992, EBV is the official distributor of Echelon products within all Europe.

EBV supports their customers by a staff of highly experienced field application engineers.

Intellectual Properties of Echelon

Echelon invented the Neuron chip technology and license it to CYPRESS and TOSHIBA for production.

Echelon has developed and released generic SW packages to design, install, service and maintain LonWorks control networks.

Echelon holds over 67 patents in network technology.

LonWorks : an accepted standard

- Accepted as an ANSI/EIA standard: ANSI/EIA 709.1-A-1999.
- Proposed as an European Standard of open system in building automation (prEN-16484).
- Accepted as a standard in the railway industry (IEEE-1473-1999)
- Echelon transceivers (FTT ou PL) are compliant to worldwide standards in safety and EMI.
- And many others (IFSF, EIA-852,….)

User Groups

They manage marketing and product validation activities, conferences as well as exhibitions and trade shows for their members.

- **LNO** – German User Group (160 members)
- **UKOSA** – British User Group (30 members)
- **LUF** – French User Group (50 members)
- LonTech Thun – Swiss User Group (30 members)
- **Swedish LonUsers** (142 members)
- **Danish LonUsers** (100 members)
- **Polish LonUsers** (14 members)
- **LonUsers Italia** – Italian User Group (30 members)
- **LonMark** Association (300 members)

Main Technology Benefits

Interoperability of HW products enable smart installation, service and maintenance while lowering cost of ownership.

Applications can be reconfigured and expanded without the need to reengineer an installation or even shut down the network.

Applications are independent from the physical transport media e.g. powered/unpowered power line, fiber optic twisted pair cable, wireless RF, COAX etc. **Moreover, LonWorks « travels » over Internet in a standard way (EIA-852).**

LonWorks supports multi-vendor installations without the need to create/manage application specific gateways and/or protocol converters. Only one single generic installation/service tool: e.g. LonMaker Integration Tool is required.

LonWorks is considered to be a front runner in Open System Technology in the international marketplace.

LonWorks supports LAN/WAN infranets. Using the i.LONs, networked control systems can be monitored and controlled via Internet connections based on TCP/IP, RTC or DSL lines.

Echelon SW packages are based on widely accepted programming standards like OOP, COM, OLE, ActiveX or XML.

Echelon provides the complete set of products and tools needed to build interoperable networks. This approach further reduces expenses allocated to warehouse logistics and parts procurement.

Echelon's worldwide business results (as of end of 2003)

Echelon was founded in 1990 by Dr. Ken Oshman and Mike Markkula. In July 1998, Echelon became a public company (NASDAQ:ELON). Echelon employs 300 people worldwide.

- 1999 Result 37 millions $
- 2001 Result 76 millions $
- 2003 Result 118 millions $

Echelon has a base of over 4000 OEM customers worldwide.

Echelon's product portfolio

Transceivers (FTT10A, LPT11, FT31xx, PL31XX) interconnect Neuron chips with the communication media.

Control modules are generic customizable LonWorks platforms.

PC interface cards to connect networks with Windows OS.

Routers modules (LonPoint router, i.LON1000) to segment large networks and combine different communication media.

LonPoint modules ready-to-go HW for analog IO, digital IO, network controller that perform remote or local input/output functions.

Generic Software and Firmware that provide the middleware to deploy a LonWorks system. (LNS, LonMaker, ShortStack, gateway software, driver SW for Windows 95/98/2000/NT/XP)

Test equipment for verification of data transmission (power line communication analyzer, protocol analyzer for LonWorks networks)

Politique de licences de produits Echelon

All software products (including LonTalk firmware) are made available to customers under the terms and conditions of software licenses that give customer the right to generate and use application code based on Echelon SW.

OEM customers agree to pay royalties to Echelon when redistributing parts of SW licensed from Echelon.

Echelon does not license the technology of their transceivers, routers and HW development systems to 3rd parties.

The LonTalk protocol is a public specification (ANSI/EIA 709.1-A-1999) and can be embedded into any processor, from 8-bit microcontrollers to 32-bit microprocessors.

Assistance et formation des clients

Echelon maintains customer support & training centers in London, Londres, Malmö, Erlangen and Venise.

Echelon offers technical support and replacement/repair services for all of their hardware and software products.

Sales Channels

- Distribution (via EBV) in Europe
- Distribution in APAC region
- Direct Sales in North America
- Distribution (via Engenuity Systems) in USA
- Network Integrators
- Direct Sales to key accounts

Major technology partners

- TOSHIBA (Neuron chip)
- CYPRESS (Neuron chip)

Echelon's important customers and early adopters

- EBV (distribution)
- ALSTOM (Railway)
- Bombardier (Railway)
- Honeywell (automation)
- Johnson controls (automation)
- Merloni (white goods)
- SAMSUNG (white goods)
- SAMPO (home appliances)
- Invensys Building Systems (automation)
- Legrand (lighting)
- ENEL (utility)
- EBM (clean room air conditioning)
- TAC / Schneider (automation)
- Edwards High Vacuum
- Philips Lighting
- Enermet (metering)
- Siemens Building Technology (automation)
- Trend
- Schindler

1 Introduction

Networks are changing our lives. Everywhere we turn, they are used to collect and move data, connected to computers and run business.

In the past, network performed these functions locally. Today, the Internet allows that networks operate globally.

Now, another networking revolution will expand the impact of Internet to the world of control.

Control networks greatly expand the possibilities of communication. When combined with data networks, control networks provide immediate and vital information on the enterprise, along with the means to act on it instantly.

Control networks link devices – replacing the central controllers and wiring harnesses of yesterday.

Although open systems are commonplace in data networks, control systems have traditionally been closed and proprietary architectures.

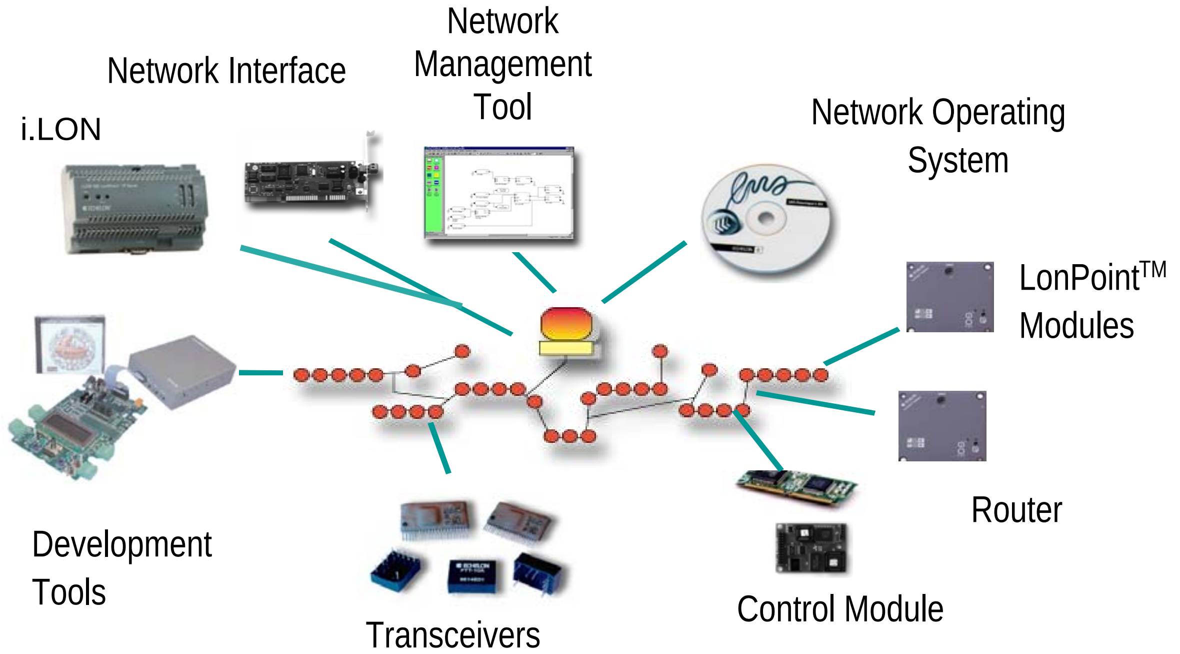
Echelon is the first and only networking company that provides a multi-vendor, open network architecture for control applications. This brings the benefit of reduced costs and enhanced flexibility to the world of controls.

Each device in the control network is smart and intelligent enough to function independent from a central super-visor system. Networked together, these smart devices communicate with each other to provide distributed monitoring and control.

Applications range from small networks embedded in a single machine to large networks with thousands of devices that monitor and control all functions in an entire building, transportation system or manufacturing operation.

This flexible approach eliminates the limits of traditional control technology and opens the way to a multitude of new applications and services.

Echelon's LONWORKS Technology provides a great variety of powerful HW and SW components that are used to build intelligent distributed LONWORKS networks.



The benefits for the user of this technology are obvious:

- Lower installation cost
- Improved reliability
- Increased flexibility
- Improved diagnostics
- Lower maintenance costs
- Lower system costs
- Unlimited remote data access

The most important standards to which LONWORKS complies:

- IFSF (Petrol)
- IEEE P1473.1 (Rail Transit)
- EIA 709.1 (LonTalk protocol)
- EIA 709.2 (FTT10 transceiver)
- EIA 709.2-A-2000 (power line communication)
- EIA- 852 (LonWorks over IP)
- Pr-EN16484 (European Norm about Open Systems)

2 Basics of LONWORKS

Most of LONWORKS products, called *nodes,* are based on a special microcontroller, the *Neuron* chip. The functional model of a Neuron chip as well as the firmware supplied with the Neuron, the *LonTalk©* protocol*,* were specified by Echelon in 1990.

The Neuron chip matches the following requirements:

- It provides powerful I/O capabilities and communication functions needed in a distributed system.
- It uses an unique identifier, the *Neuron ID*, to be addressed within a network.
- It can be easily programmed in *Neuron C*, a structured language based on the ANSI C standard.
- It use a *media-independent communication* model: Network data may be transmitted using simple twisted pair wires, RF links, fibre optic links, power line, COAX cable etc.
- The LonTalk protocol firmware provides services to efficiently transport and route data from peer to peer.

Once a node is commissioned with a network management tool, powered-up and executing the LonTalk firmware, the user can control the operation mode of the node through network management messages. He might e.g:

- Download program in the node
- Replace program in the node
- Change node configuration parameters
- Put the node on-line or off-line
- Execute a reset sequence
- Get data from the node
- Send data to the node
- Bind the node into a network (automatic data update)
- Move or remove the node from a network

Echelon and some other 3rd party suppliers provide powerful network installation, service and maintenance software in order to fulfill the above tasks.

All basic functions and services required to manage a node are supplied by the firmware. This firmware provides an event-driven operating system to schedule and execute the application program, handle the data structures needed to communicate with other Neuron chips and operate the local 11-pin I/O block.

The unique Neuron ID stored in the Neuron chip identifies each node in a LONWORKS network. This write-protected identifier is 48 Bit wide and is stored in the on-chip EEPROM when the chip is manufactured.

Neuron chips communicate with each other by sending data telegrams over the network. Each telegram is formatted by the Neuron‘s firmware and contains address-, routing- & control information as well as the application data along with a checksum.

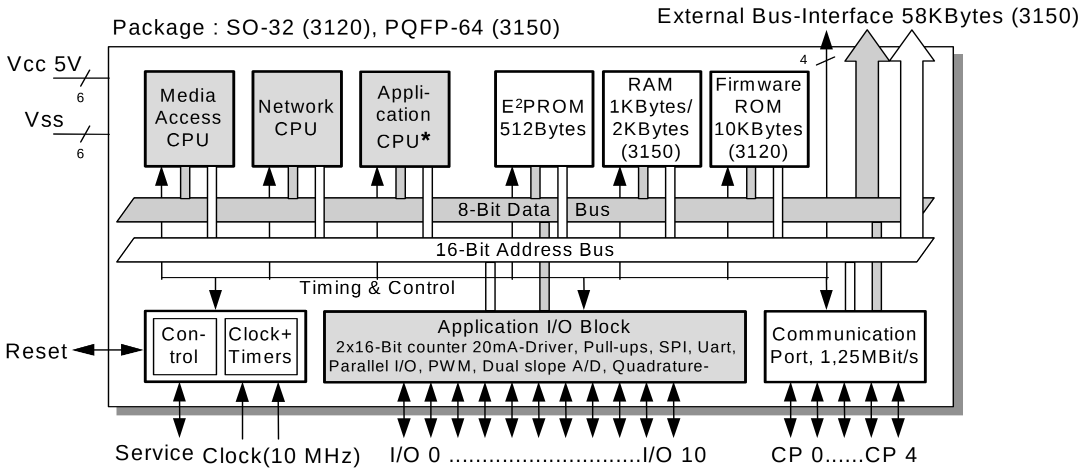
The data transmission is initiated und supervised by the Neuron‘s firmware. Each telegram may contain up to 229 bytes of data.

User/application data may either have the format of an *explicit message* or of a *network variable*. Network variables provide the node with a structured data communication model and fully automated data send/receive handling operated by the LonTalk protocol.

Explicit messages provide a simple data send/receive handling operated under control of the node‘s application program.

3 Neuron Chip

The following picture shows the functional blocks of the NEURON IC.



TOSHIBA offer various Neuron IC models in different package types and memory sizes: The model 3120 is a single chip Neuron – no external memory is required; the 3150 is an equivalent to a 3120 with the exception of its external memory interface used to store the firmware, program and application data.

The 3120‘s internal non-volatile EEPROM contains both the user application program and communication parameters. The on-chip static RAM holds local program data and communication data buffers needed by the application program.

The 3150‘s external bus interface is used to interface the IC with SRAM, NVRAM, EEPROM and/or memory mapped I/O.

The 3150 stores the application program and firmware on external memory whereas the 3120 uses on-chip memory resources to store the firmware and the application program.

Both models of the Neuron allows for program download and application parameterization via its communication port by means of the embedded LonTalk firmware.

Each of the three CPU's of the Neuron execute a dedicated job:

- the *media access CPU* handles all serial I/O on the communications port.
- the *network CPU* supplies protocol data handling services, timing services used in various states of the data processing within the Neuron IC and provides subroutines to drive the local application IO block.
- the *application CPU* runs the application program. The application program is edited, compiled and linked by means of a Neuron-C program development system. The program itself might be downloaded via the communication port or supplied as external memory.

The user/programmer has access to only the application CPU. He may, if necessary, control the operating modes of the other CPUs by supplying them appropriate parameters.

The communication port is software-configurable to run at bus clock speeds from 600 baud up to 1.25 Mbps. The hardware designer of a node connects this port to an external transceiver in order to decouple and electrically isolate the Neuron IC from the bus wiring. The node may use whatever transceiver is appropriate to match the application requirement: RS485, transformer, fibre optic, infrared, power line, COAX cable, radio frequency etc. Adapting the Neuron chip communications port to these transceivers is just a matter of software parameterization of the node: the application program is independent from the type of transceiver used to connect the node to the network.

Unlike other microprocessors, the Neuron chip does not provide an external interrupt pin. All synchronization between the Neuron chip and the external H/W is done by S/W. Signals applied to the IO application block must be stable for at least 200ns(for a 10MHz Neuron chip) in order to be correctly sampled by the internal firmware.

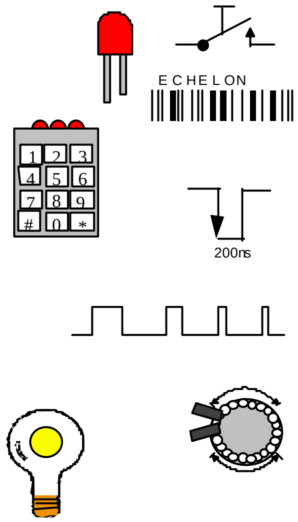
The bidirectional service-pin is provided to:

- Monitor the internal firmware state
- Make the Neuron chip sending its Neuron ID.

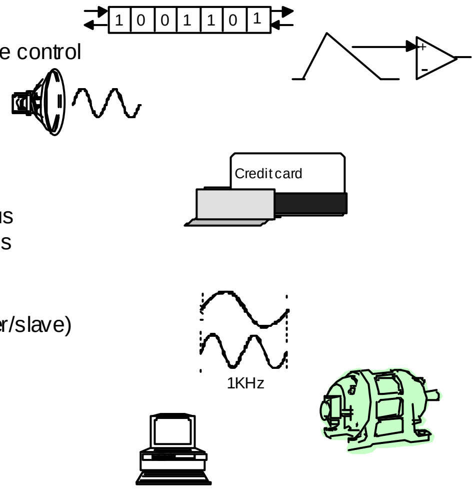
The following table shows the currently available and future versions of Neuron chips manufactured by TOSHIBA.

4 Neuron Chip I/O Models

The Neuron may be connected to one or more physical I/O devices. Examples of simple I/O devices include temperature and position sensors, valves, switches and LED displays. Neuron chips can also be connected to other microprocessors. The Neuron chip firmware implements I/O objects that manage the physical interface of external devices of a Neuron C application. The next picture shows some of the available I/O objects:



- Bit, Nibble and Byte IO
- Serial bit IO
- ADC with external dual-slope control
- Edge log input
- Frequency output
- Infrared decoder
- Level detect input
- Mag card input
- Multiplexed data/address bus
- Neurowire (SPI) serial IO bus
- Monoflop and PWM output
- Pulse width measurement
- parallel bus coupling (master/slave)
- Pulse count IO
- Quadrature decoder
- Asynchronous serial IO
- pulse count
- Triac control
- Triggerable counter



The programming model of the Neuron C language allows the programmer to declare one or more pins as I/O objects. These declarations can be thought of as written firmware routines in ROM which are accessed by the application program. The application program refers to these objects in *io_in* and *io_out* system calls to perform the actual input/output function during execution of the program.

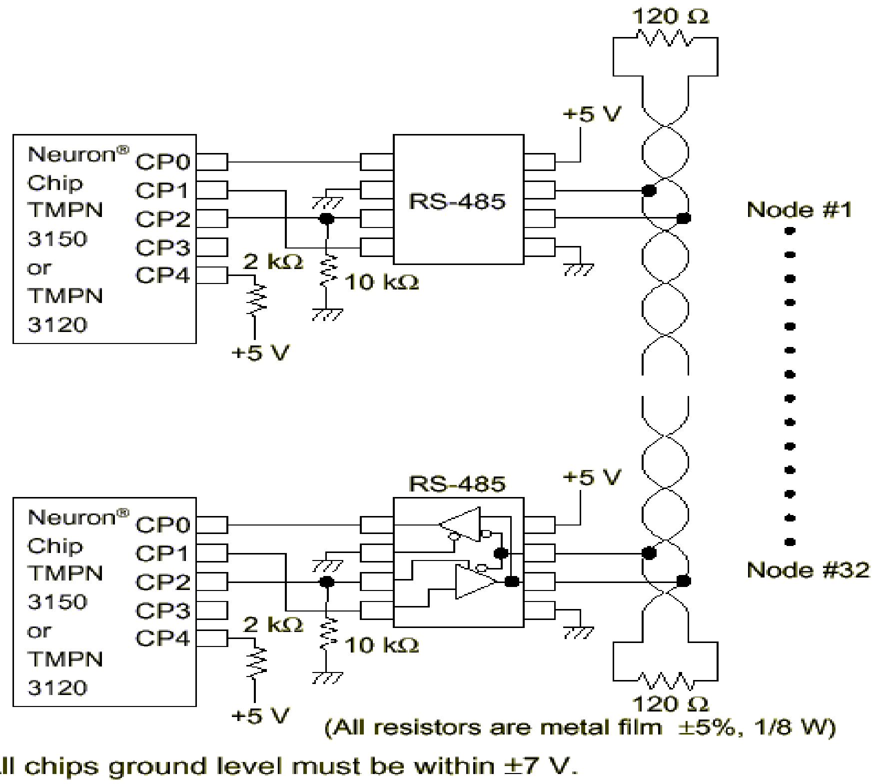
There are 34 different I/O objects available in the 3150 Neuron firmware. Most are available in the 3120 Neuron internal ROM. The additional objects can be loaded into the 3120 EEPROM if needed.

The Neuron has two on-chip 16Bit timer/counters. The input to timer/counter 1, also called the *multiplexed timer/counter,* is selectable among pins IO4 through IO7 via a programmable multiplexer and its output may be connected to pin IO0. The input to timer/counter 2, also called the *dedicated timer/counter,* may be connected to pin IO4 and its output to pin IO1.

5 Transceivers

The Neuron chip provides a very versatile communications port. It consists of five pins that can be configured to interface to a variety of network transceivers and operate over a wide range communication rates. The communications port can be configured to operate in one of three modes: single-ended, differential, or special purpose mode.

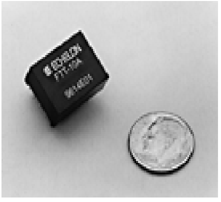
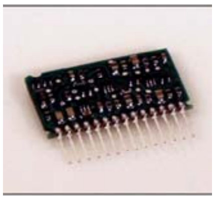
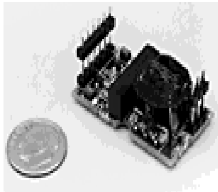
The following schematic shows a simple RS485-based network interface between the Neuron chip and the LON using the single ended mode.



Opto isolators can be added to increase the common mode voltage range. EIA-485 transceivers allow for cost performance/size advantages compared to other solutions. Data rates of up to 1.25 Mbaud are supported.

Transformer- coupled networks work well for applications that require higher performance, high common mode rejection and noise immunity between nodes. Transformer coupled transceiver designs can operate up to 1.25 Mbps and achieve common mode ranges of +/- 277Vrms. The most widely used transformers are the LPT11 and FTT10A manufactured by Echelon.

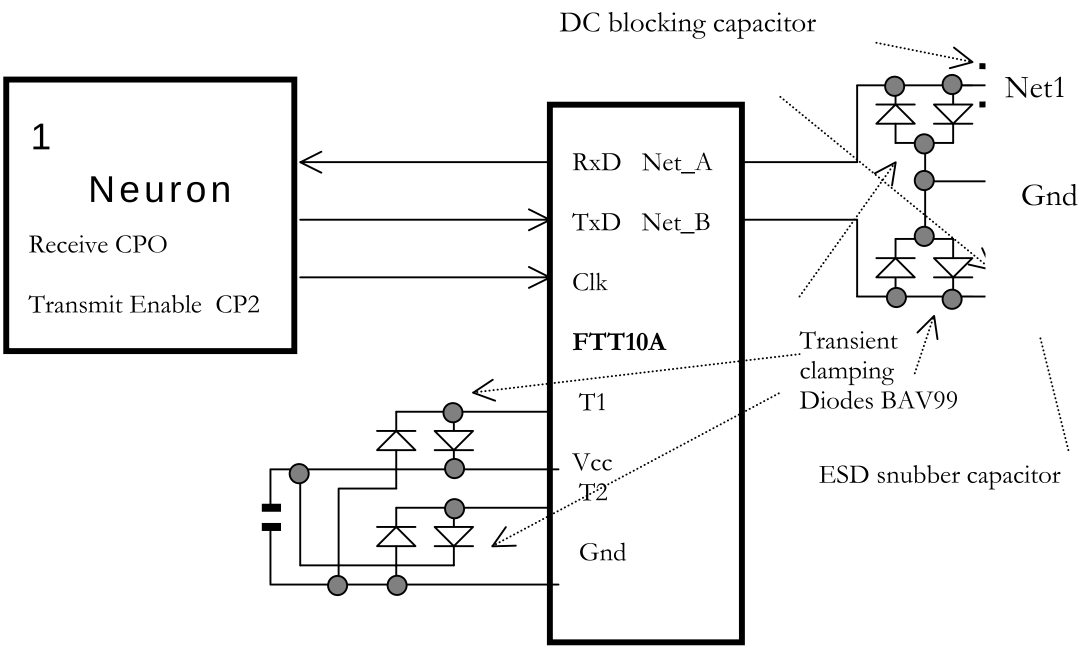
Echelon's transceivers are designed to directly interface to the communications port of a Neuron chip and cannot be used with other microprocessors.

				
FTT10A	FT3120/50	LPT11	TP/XF1250	PL3120/50
Transceiver 78kbit 5VDC Power Supply required	**Transceiver 78kbit** 5VDC Power Supply required	**Transceiver 78kbit** 48VDC “Linkpower” Supply required	**Transceiver 1.25Mbit** 5VDC Power Supply required	**Transceiver 4.8kbit** 5V/9V DC Power Supply and coupling circuit required
bus, loop and star topologies	bus, loop and star topologies	bus, loop and star topologies	bus topology	All topologies
	Interoperable with FTT10A	Interoperable with FTT10A		

5.1 FTT10A Free Topology Transceiver

Echelons FTT10A transceiver provide many advantages over the RS485 transceiver. Since the FTT10A is basically a transformer-coupling unit, it electrically isolates the Neuron chip from the network, thus preventing the Neuron chip or the custom device being damaged due to ESD, cable short circuits or by overvoltage imported from the cabling.

Part of the preferred interconnection between the FTT10A transceiver and a Neuron chip is shown below.



The network connection (NET1 and NET2) is polarity insensitive and therefore either of the two twisted pair wires can be connected to either of these two network connections.

The FTT10A receives its clock input from the Neuron chip. The input clock may be 5, 10 or 20 MHz. It automatically detects the clock rate and configures internal circuitry appropriately.

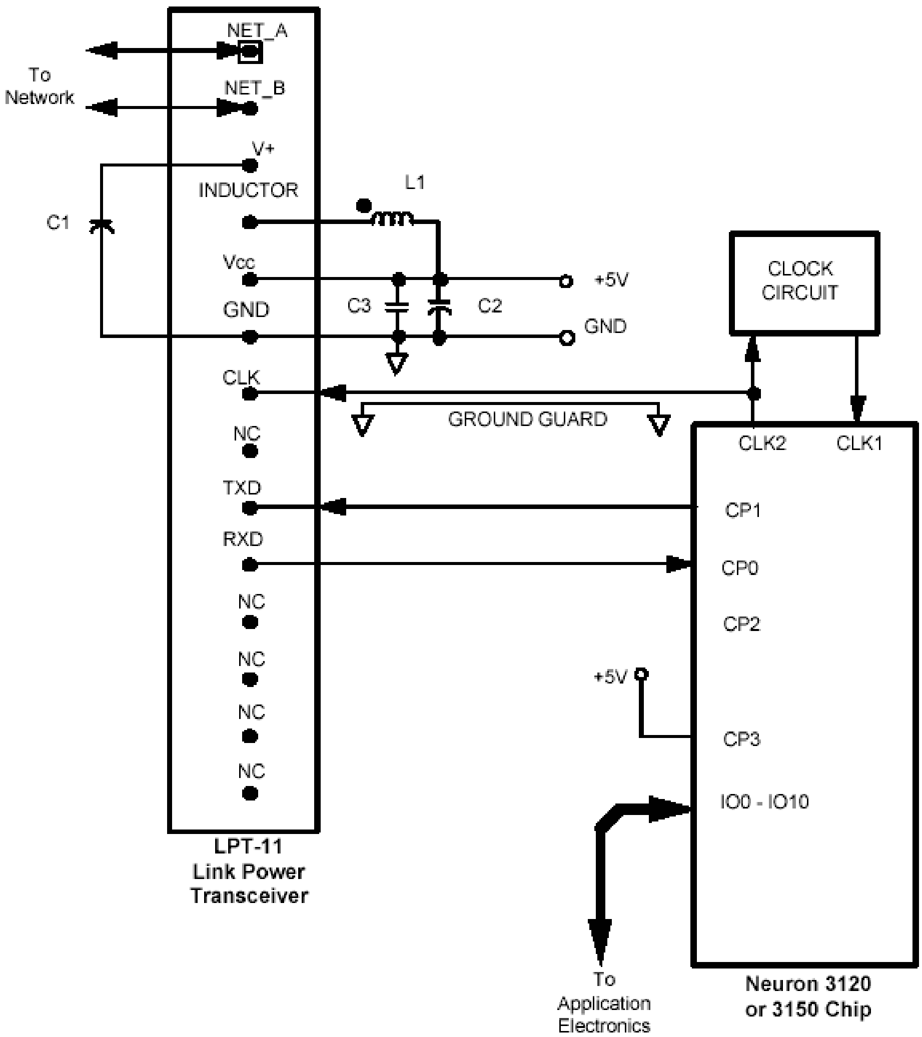
The FTT10A transceiver has a fixed bit transmission rate of 78Kbps. This transceiver allows free topology wiring, which reduces the time and expense of system installation by allowing the wiring to be installed in the most expeditious manner. It also simplifies network expansion by eliminating restrictions on wire routing, splicing, and node replacement.

5.2 LPT-11 Link Power Transceiver

The LPT-11 Link Power Twisted Pair Transceiver combines power and data on a common twisted wire pair, and allows the user to wire LPT-10 transceivers with virtually no topology restrictions. The LPT-10 provides a regulated +5VDC@100mA for the node from the power sent via the twisted pair, eliminating the need to use a local power supply.

The LPT11 is compatible with the FTT10A and these transceivers can communicate with each other on a single twisted pair wire.

The following drawing shows the typical LPT-11 to Neuron interface



5.3 Free topology Smart Transceivers FT31X0

The FT 3120 combines together a Neuron chip with 4 Kb of Flash memory for applications, 2 Kb of RAM and 12 Kb of ROM, and a free topology transceiver.

The FT 3150 together a Neuron chip with external memory interface and a free topology transceiver.

The FT3120 and FT3150 are supplied with an external commun ication transformer (FT-X1) which enables operation in presence of high frequency common mode noise on unshielded twisted-pair networks. Properly designed nodes can meet the rigourous Level 3 requirements of EN 61000-4-6 without the need for a network isolation choke.

Ideally suited for use in sensors, actuators, switches, lamps and motors, the FT 3120 can operate at a clock speed up to 40MHz. It is available in two compact packages : 32 pins SOIC or 44 pins TQFP.

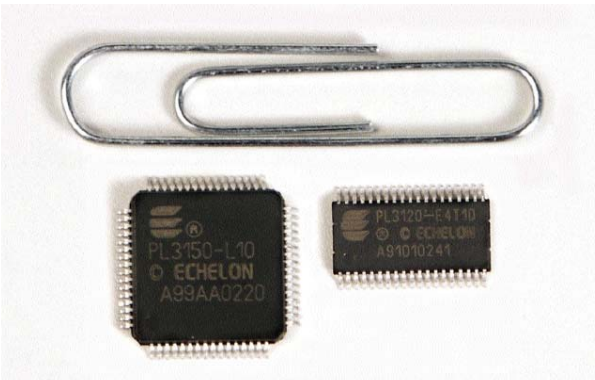
The FT 3120 can be programmed by many universal programmers from most well-knows suppliers.

I Ideally suited for use in sophisticated devices requiring large application software space and extended I/O, the FT 3150 can operate at a clock speed up to 20MHz. It comes within the a 64 pins TQFP package.

Then, the FT3120 is pin compatible with the 3120 Neuron while the FT3150 is pin compatible with the 3150 Neuron from Cypress and Toshiba.

Product Code	Max. Clock (MHz)	Internal Flash (Bytes)	Internal RAM (Kb)	Internal ROM (Kb)	External Memory	Package
FT3120-F4S40	40	4094	2	12	No	SOIC32
FT3120-F4P40	40	4096	2	12	No	TQFP44
FT3150-P20	20	512	2	N/A	Yes	TQFP64

5.4 PowerLine Smart Transceivers PL31X0

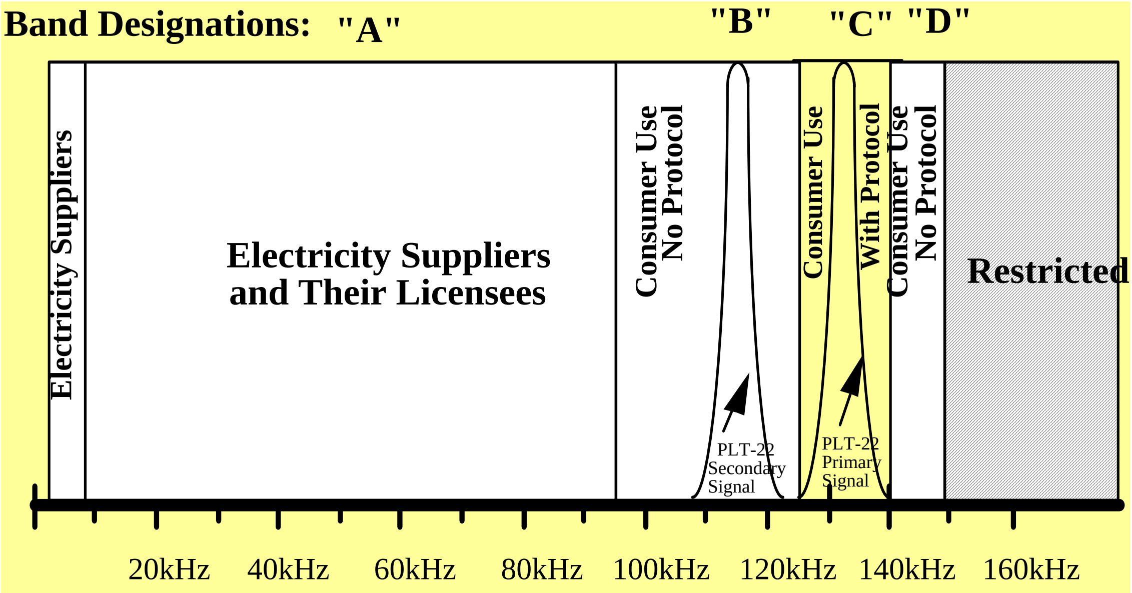


The PL3120 / PL3150 combine an EIA709.2 compliant powerline transceiver with a Neuron chip with extended I/O capabilities.

Available in extremely compact packages (For ex. the PL3120 is smaller than 10 x 6mm), they allow designers to create powerline communicating product at very low cost.

The PowerLine transceiver complies with CENELEC EN50065-1 regulations and transmits primarily in the 125kHz to 140kHz frequency band.

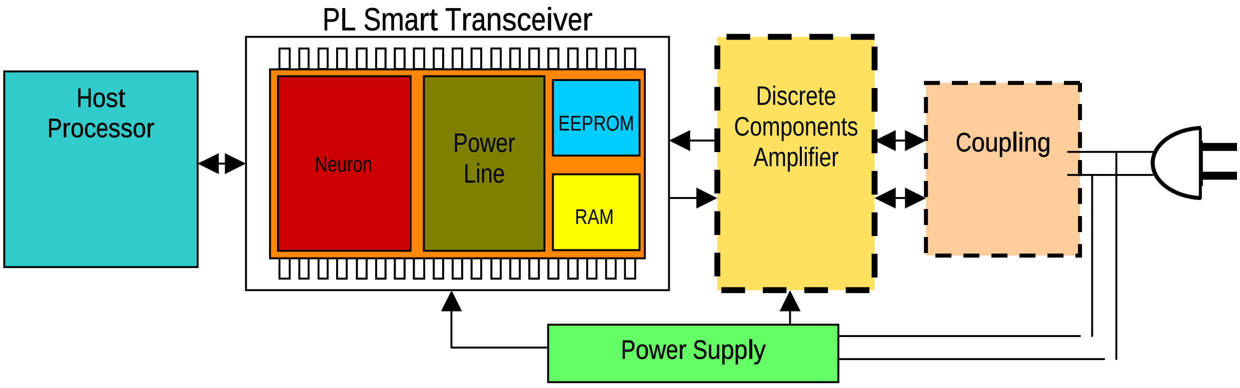
The powerline transceiver can automatically take care of the CENELEC access protocol which enables systems (sets of devices) from various vendors to share the same power cable and still communicate without interfering.



CENELEC frequency Allocation

The PL3120 and PL3150 are also different from standard neuron chips because of their 12 pins I/O port which provides 38 I/O models. Both feature a full-duplex hardware UART able to operate up to 115 kbps in SCI mode and up to 625 kbps in SPI mode.

The PL3150 features a full address/data bus allowing to address an external memory space of 58 Kbytes.



The power amplifier is an around-50 discrete components design (mostly resistors and condensators along with a small number of transistors).

The power management feature of the PL31X0 which constantly monittors the supply voltage allows to power simple nodes with a very unexpensive energy storage power supply.

Similarly to any product using RadioFrequencies, placing and routing of components is neverhteless a critical stage of the design process. To ease this task and insure compliance to regulations, Echelon provides a « design kit » for any PL chip, including a design review by their engineering group.

6 The LonTalk Protocol (EIA 709.1 Standard)

The Neuron chip implements a complete networking protocol using the media access CPU and the network CPU. This networking protocol is designed according to the ISO OSI reference model for network protocols. It allows programs running on the application CPU to communicate with applications running on other Neuron chip nodes elsewhere on the same network. The protocol services are invoked by application level objects called network variables and explicit messages.

The main features of the LonTalk protocol are:

multiple media support:

- Twisted pair, transformer coupled, PowerLine, RF, COAX, infrared, fibre optic and others.

support for multiple communications channels:

- a channel is a physical transport medium for data telegrams (packets) and can contain up to 32.385 nodes. A network may consist of one or more channels. Packets are transferred from one channel to another by using routers.

The next table resumes the services and functions provided by the 7 layers of the LonTalk protocol.

OSI Layer	Purpose	Services
Application	Application Program	Standard objects & types, configuration properties, file transfer, network services
Presentation	Data Interpretation	Network variables, application messages, foreign frames
Session	Remote Actions	Dialog, remote procedure calls, connection recovery
Transport	End-to-End Reliability	End-to-End acks, service type, packet sequencing, duplicate detection
Network	Destination Addressing	Unicast & multicast, destination addressing, packet routing
Data Link	Media Access & Framing	Framing, data encoding, CRC, media access, collision detect
Physical	Electrical Interconnect	Media specific details, transceiver, type, physical connect

Nodes talk to each other by sending messages. The message concept is used to even transport data which the application program refers to as network variables.

The LonTalk protocol offers four basic types of message service:

- ***Acknowledged*** (or end-to-end acknowledged) service: a message is sent to a node or a group of nodes and individual. Acknowledgements are expected from each re-ceiver. If an acknowledgement is not received from all destinations, the sender times out and re-tries the transaction. The number of re-tries and the time-out are both selectable. The network CPU automatically generates the acknowledgements. Transactions Ids are used to keep track of messages and acknowledgements so that the application does not receive duplicate messages

- Request/response service is used to send a message to a node or a group of nodes from which individual responses are expected. The incoming message is processed by the application on the receiving side before a response is generated. The same retry and time-out options are available as with acknowledged service. Response data may include data, so that this service is particularly suitable for remote procedure call, or client/server applications.

- *Repeated* (or unacknowledged repeated) service sends a message to a node or group of nodes multiple times, and does not expect a response from the receiving nodes. This service is typically used when broadcasting to large groups of nodes, in which the traffic generated by all the responses would overload the network.

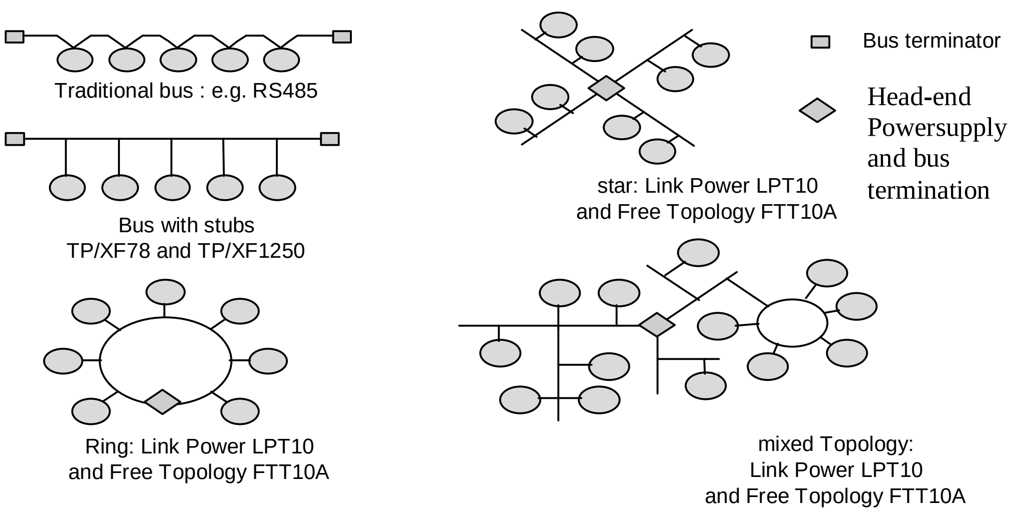
- *Unacknowledged* service sends a message to a node or a group of nodes once and does not expect a response. This is typically used when the highest network performance is required, network bandwidth is limited or when the application is not sensitive to the loss of a message.

The LonTalk protocol also supports authenticated messages; the receiver of an authenticated message checks if the sender is authorized to send that message. This method prevents unauthorized access to nodes or their applications. The use of authentication is configured individually for each network variable. Network management transactions may also be optionally authenticated.

7 Network Topologies

LONWORKS provides support for multiple communications channels. A channel is a physical transport medium for data telegrams (packets) and can contain up to 32.385 nodes. A network may consist of one or more channels. Packets are transferred from on channel to another by using routers.

LONWORKS supports bus, loop and star topologies. The design of the transceiver determines the number of nodes in one channel as well as the maximum transmission distance between nodes in one channel.



The following table lists the various characteristics of some frequently used transceiver:

Transceiver Type	Bit Rate	Network topology	Nodes per channel	Maximum Length	Type of isolation of Neuron	Typical Application
TP/XF1250	1.25 Mbps	Bus	64	130 m	Transformer	Ind. & Bldg
FTT10A	78 kbps	Bus	64	2700 m	Transformer	Ind. & Bldg
FTT10A	78 kbps	Free	64	500m	Transformer	Ind. & Bldg
LPT10	78 kbps	Bus	128	2200 m	Transformer	Ind. & Bldg
PowerLine	4.8 kbps	Free	Various	Up to 5 km	Custom	Various

8 Network Addressing

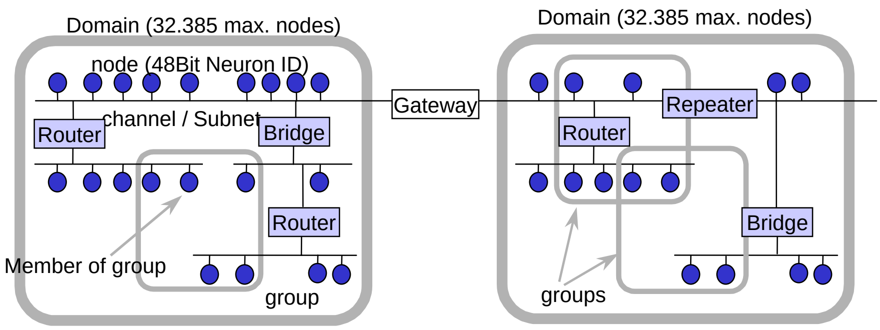
LONWORKS uses an efficient 3-level addressing hierarchy to access nodes in the network.

The 1st level addressing of the hierarchy is the domain. The domain identifier has a selectable length of 0,1,3 or 6 bytes. A single node can be a member of up to two domains.

The 2nd level of addressing is the subnet. There may be up to 255 subnets per domain. A subnet is a logical grouping of nodes from one or more channels. Routers operate at the subnet level, determines which subnet lie on which side of it and forward packets accordingly.

The 3rd level of addressing is the node. There may be up to 127 nodes per subnet. Thus a maximum of 255 x 127 = 32.385 nodes may reside in a single domain. Any node may be a member of one or two domains, allowing a node to serve as an inter-domain gateway. This also allows, e.g., a single sensor node to transmit its data into two different domains.

Nodes may also be grouped. Groups of nodes may span several subnets within a domain. The channel does not affect the way a node is addressed. Domains can contain several channels. Subnets and groups of nodes may also span several channels.



Router: selectively forwards packets between two channels of same or different physical media. (e.g. RS485 to FTT10A).

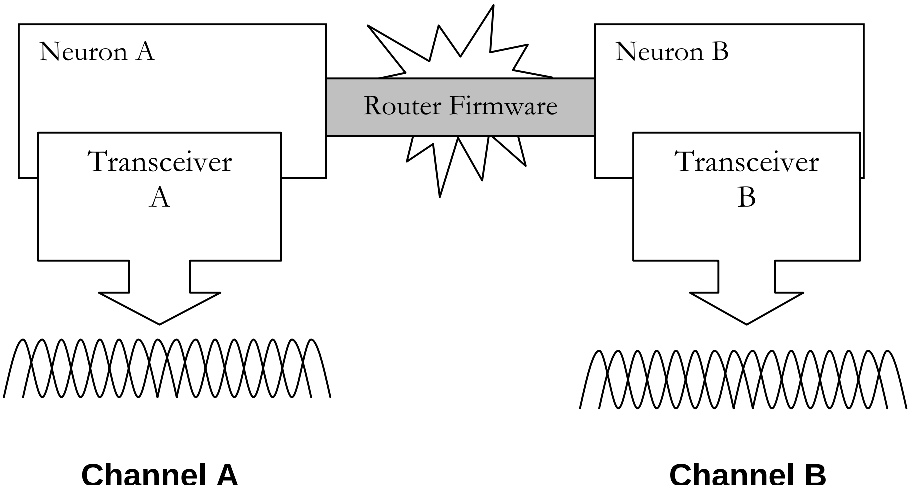
Repeater: forwards all packets between two channel of same phy. Media. (e.g. RS485 to RS485, FTT10A to FTT10A).

Gateway: connects one channel in the 1st domain to another channel in the 2nd domain.

Bridge: forwards all packets between two channels of same or different physical media. (e.g. FTT10A to FTT10A). Both channels must reside in the same domain.

9 Router

A LONWORKS router basically consists of two Neuron chips and two transceivers. Both Neuron chips communicate with each other via their application port. The router firmware executes in both Neuron chips and forward packets from one channel to another steered by internal routing tables. These tables are either pre-programmed during network configuration/installation or configured to be self-learning.

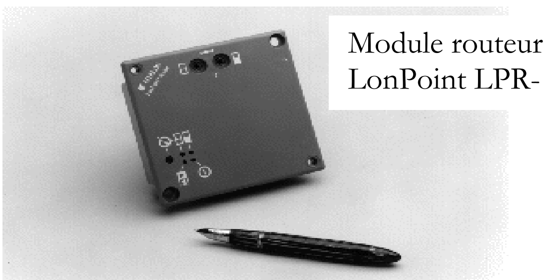


Routers are used to optimize packet traffic on channels. Packets generated on one channel are selectively forwarded to another channel if they contain address information referencing that channel.

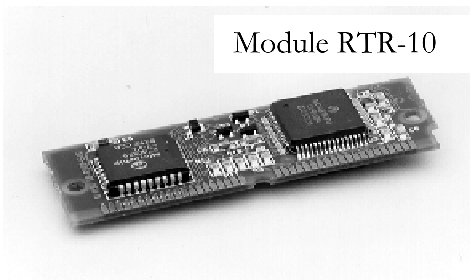
Routers can also be configured to operate as bus repeaters. This application mode is especially useful where the distance between nodes exceeds the constraints specified by the transceiver designer.

Echelon offers ready-to-use router (model LPR-10) and a so-called router core module (model RTR-10). The LPR-10 is available with FTT10A, TP/XF78 and TP/XF1250 trans-ceivers. The RTR-10 SIMM module only needs two external transceivers to build a custom router.

Routers are configured at the network installation or commissioning.



Module routeur LonPoint LPR-

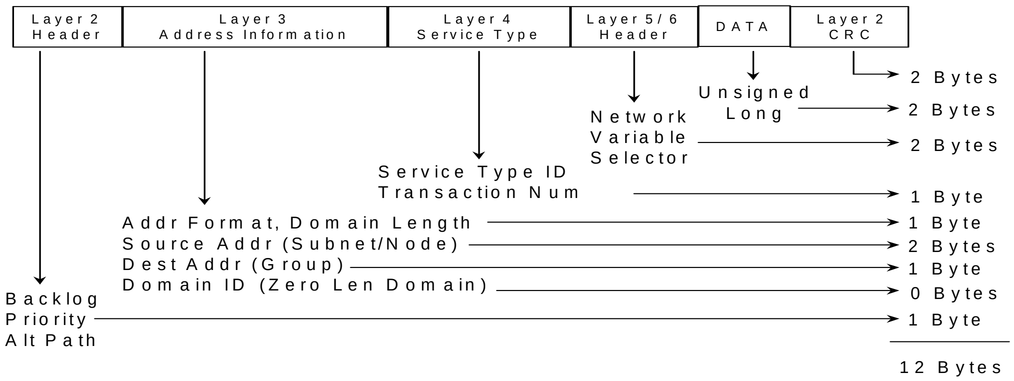


Module RTR-10

10 LonTalk Packets

LonTalk nodes communicate with each other by sending datagrams which carry many types of information needed to operate the control network in a reliable and consistent way.

The following table details the individual components of a LonTalk protocol datagram:



The data field may be up to 228 bytes long. The domain ID field may be 0, 1, 3 or 6 bytes long. In general, the protocol overhead is of constant length of 10 bytes.

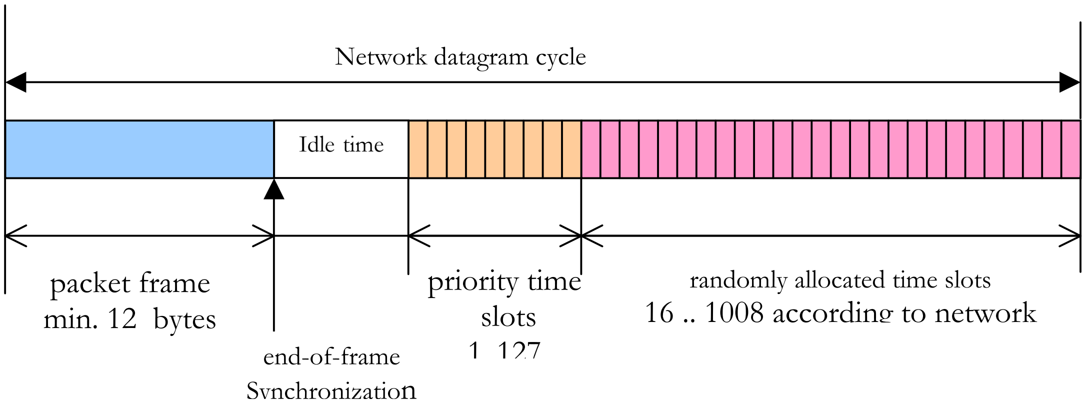
The frame handling is done by the media access CPU and the network CPU. The application program just has to provide the content of the DATA field. This content is referenced within the application program either as a *network variable* or a *message field.*

The following table summarizes the channel throughput capability measured for a 10MHz 3150 Neuron using a direct connect transceiver sending 12 Bytes long datagrams:

Network Bit Rate (Kbps)	**Channel capacity (in packets per second)**
9,8	100
19,5	192
39,1	337
78,1	410
156,3	508
312,5	615
625	696
1250	1021

11 LonTalk Media Access

The LonTalk firmware provides a CSMA/CA algorithm to manage the transmission of datagrams. To do so, the media access CPU enters into a bus access phase after the network is in idle state, which is the case when the media access CPU of a sending node has sent the *end-of-frame* synchronization byte.



Each node on a network channel is limited to a single outgoing transaction at a time. The LonTalk protocol implements a mechanism to avoid that access to the network is denied indefinitely. To do so, the time when the transmission of a telegram frame should start is calculated using a randomly generated time slot. This start time calculation can be overridden by assigning a priority time slot to each node on a channel. In this case, all nodes waiting to send a frame would then start sending in the order they have been prioritized. The priority feature uses separate buffers within each node allowing outgoing priority packets to get in front of non-priority packets, which have already been queued up for transmission. Additionally, the priority feature uses dedicated bandwidth at the end of each packet which eliminates contention for the communications medium after the transmission of a packet.

For even higher performance applications, transceivers may be implemented which perform collision resolution by hardware. These transceivers are most useful when the channel bandwidth is limited and/or there is a need to run the network at its maximum capacity for a sustained time.

12 Network Variables

A network variable is an object on one node that can be connected to one or more network variables on one or more additional nodes.

A node‘s network variables define its inputs and outputs from a network point of view and allow sharing of data in a distributed application. Whenever a program writes into one of its **output network variables**, the new value of the network variable is propagated across the network to all nodes with **input network variables** connected to that output network variable. Although the propagation of network variables occurs through LonTalk messages, these messages are sent transparently. The application program does not need explicit instructions for sending and receiving network variable updates.

Network variables greatly simplify the process of developing and installing distributed systems because nodes can be defined individually, then connected and reconnected easily into new LONWORKS applications.

Network variables promote interoperability between nodes by providing a well-defined interface that nodes use to communicate. Interoperability simplifies installation of nodes into different types of networks by keeping the network configuration independent of the node's application. A node may be installed in a network and logically connected to other nodes in the network as long as the data types (e.g. *int* or *long*) match.

To further promote interoperability, the LonTalk protocol provides **Standard Network Variable Types** (SNVTs). SNVTs are a set of predefined types with associated units, such as degree C, volts, meters, seconds. The following table shows some of currently defined SNVTs:

Name	Unit	Bits	Range of value
SNTV_lev_cont	Continuous value	8	0 à 100
SNTV_lev_disc	discrete value	8	On, off, high, low, med
SNTV_temp	Temperature	16	-273,2 à +6279,0
SNTV_power	power	16	0 à 65535
SNTV_date_time	Time HH:MM:SS	24	00:00:00 à 23:59:59
SNTV_str_asc	ASCII string	248	30 caractères

A network management tool can use LonTalk network management messages to determine the type of every network variable declared as a SNVT.

Network variables also provide a *Self-Documentation* (SD), a feature that the application programmer can use to create a text string including a network variable name, special installation instructions, etc. This information is stored with the application program on the node.

Network variables are first declared within the program that runs on an individual Neuron chip. The complete syntax for declaring a network variable object is one of the following: /see section 26 for additional literature/

```
network input|output [netvar-modifier][class] type
     [connection info] identifier [=initial value];
network input|output [netvar-modifier][class] type
     [connection info] array-bound [=initializer list];
```

Up to 62 network variables (including array elements) may be declared on a node in a Neuron C program. Up to 4096 network variables can be declared when using a LONWORKS network interface connected to a host processor.

The maximum size of a network variable is 31 bytes. In the case of a network variable array, each element is limited to a size of 31 bytes.

Examples of network variable declarations are:

```
network input SNVT_temp temp_set_point;
network output SNVT_lev_disc heater_command;
network output int current_temp;
```

Examples of priority network variable declarations are:

```
network output boolean bind_info(priority)fire_alarm;
network output boolean bind_info(priority(nonconfig))
            fire_alarm;
```

Example of declaring a network variable using unacknowledged service:

```
network input SNVT_lev_cont bind_info(unackd)  control_dial;
```

13 Explicit Messages and Foreign Frames

Applications requiring a different data interpretation model than the network variable model, can send and receive explicit messages. Explicit messages use the messaging services of the LonTalk protocol with minimal data interpretation. Each explicit message contains a message code that the application can use to determine the type of interpretation to be used on the content of the message.

For Neuron hosted nodes, explicit messages are transmitted by assigning the message code and message contents to a special output object used for transmitting explicit messages. Explicit messages are received in another special input object that contains the message code and contents.

A special range of message codes is reserved for foreign frame transmission. Up to 228 bytes of data may be embedded in a message packet and transmitted like any other message. The LonTalk protocol applies no special processing to foreign frames – they are treated as a simple array of bytes. The application program may interpret the data in the way it wishes.

Foreign frame messages are sent and received by both Neuron chip-hosted nodes and host-based nodes using the same techniques as explicit messages, except that a different range of message codes is used.

An outgoing message is defined in the Neuron C program as follows:

```
typedef enum{FALSE,TRUE} boolean;
typedef enum{ackd,unackd_RPT,UNACKD,REQUEST} service_type;

struct {boolean         priority_on;   // TRUE if priority message
        msg_tag         tag;           // Message Identifier
        int             code;          // Message code
        int             data[MAXDATA]; // 0 to 228 bytes of data
        boolean         authenticated; // TRUE if authentication
        service_type    service;       // ACKD by default
        msg_out_addr    dest_addr;     // destination address (option)
        } msg_out;
```

14 Network Management Messages

In addition to application message services, the LonTalk protocol provides network management services for installation and configuration of nodes, downloading of application programs, and diagnosis of the network.

A system may be configured so that critical network management messages are subject to authentication protection. This means that only authorized network manager nodes may execute these functions. Some of the available network management messages are listed below:

- The ***request-to-query*** message is designed for use during installation when unconfigured nodes are asked to identify themselves.
- The ***security*** message is used to transmit incremental changes in the authentication key over the network. The key itself is never transmitted, thus protecting the authentication scheme against misuse.
- The ***modify-address-table*** message and *modify-net-variable* messages may be used to dynamically bind network variables and message tags. This is used during installation and reconfiguration to estab-lish the addressing information needed to route message and network variable updates between nodes.
- The ***write-memory*** message may be used to download a new application program into a node over the network.
- The ***wink*** message is used during installation to instruct nodes to identify themselves physically (e.g. by flashing a LED indicator)
- The ***service-pin*** message is an unsolicited message used during installation so that a node may identify itself on the network by sending its unique 48Bit Neuron ID.

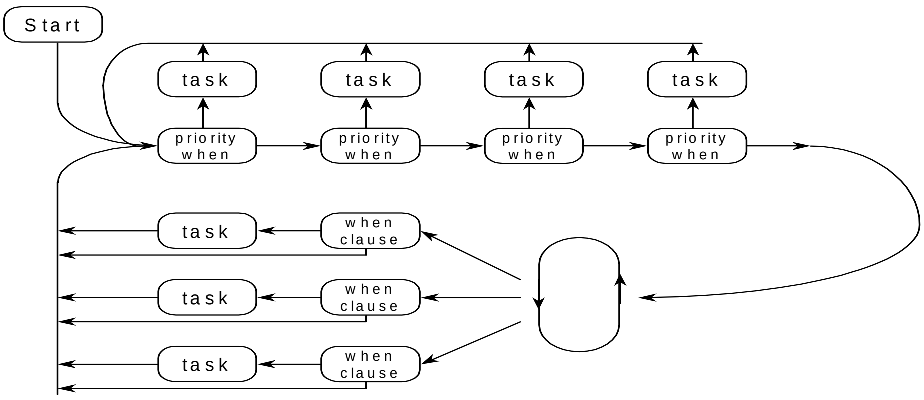
15 Neuron C Programming

Neuron chips are programmed using the Neuron C language. The Neuron chip can not be programmed in assembler.

A Neuron C program is composed of a collection of tasks. A task consists of the special syntax *when* followed by the body of that task.

```
[priority] when (condition)
{
        < task body>
}
```

Tasks may run on either of two priority levels: *normal* or *priority*. The task scheduler provided with the LonTalk firmware executes the task if the *condition* is met.



task scheduler for when statements

The Neuron C language syntax supports different classes of events, which are expressed by the *condition* clause.

IO event	A transition at IOx pin occurred A HW-timer has terminated
Communication event	A network variable or message update occurred.
Software timer event	A network communication time-out occurred. One of the 15 S/W-16Bit timers has expired.
Special event	The Neuron chip has changed its state. (e.g. RESET, ONLINE, OFFLINE etc.) A network management message was received.
User defined events	Every programmed C expression which evaluates > 0.

The use of any of the 11 pins of the Neuron chip must be declared as an IO object in the program before that IO pin can be controlled within the application program.

Par Ex.

```
IO_4 input bit switch;     // IO object declaration
when (io_update_occurs(switch))
{
     status = io_in(switch);
}
```

The primary communication mechanism of LonTalk nodes is network variables. These variables transport values which a program can either read or write. The name of a network variable must be declared as an communication object in the program before it can be used.

The following code segments illustrates some program syntax available with Neuron C:

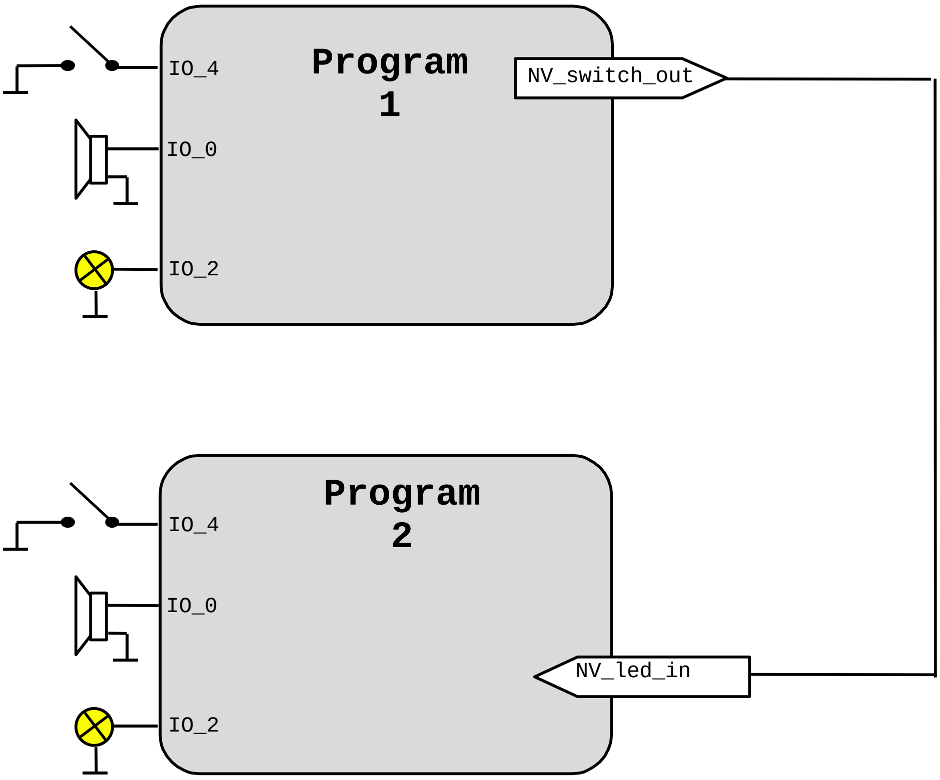
```
// program 1
// outputs the status of switch in a network variable
// to turn on the LED controlled by program 2

IO_4 input bit switch;     // IO object declaration
Int status;                // local variable
network output int led_go_on; // network variable declaration
when (io_update_occurs(switch))
{
     status = io_in(switch); // read data from IO_4
     led_go_on  = status;  // update network variable
}

// program 2
IO_2 output bit led; // IO object declaration
network input int switch_on; // network variable declaration
when (nv_update_occurs(switch_on)) // NV was updated
{
     io_out(led,switch_on); // use NV value
}
```

Note that the application program does not establish the communication links between individual network variables.

LONWORKS sharply distinguish between the programming of a node and its installation in the network. The node only provides a S/W interface to the network.

The communication links between individual variables of different nodes are created on-site, at network installation time. The process of creating the communication links between nodes is called *binding*.

The following picture gives the graphical representation of the above program:



In order for both nodes to communicate, the network variables must be connected together using a network management tool /see section 22/. The network management tool is also needed to identify both nodes by sending network management messages to the nodes which would themselves reply by sending their Neuron ID'.

Using this ID and a device installation table which maps the Neuron ID to a user-selected 3-level address scheme /see section 8/, the network management tool can than set up the Neuron's internal configuration.

The network variable selector entry stored in the EEProm-table in the Neuron chip is used to reference the name of an individual network variable. It is this selector, which is actually transmitted in the packet /see section 10/ when a network variable is updated.

16 LonMark Compliancy

The LonMark association is an independent member organization of producers, consumers and consultants of LONWORKS-based products and services. The LonMark association specifies and publishes recommendations and S/W implementation guidelines describing the operating modes of devices and services by means of objects and functional profiles.

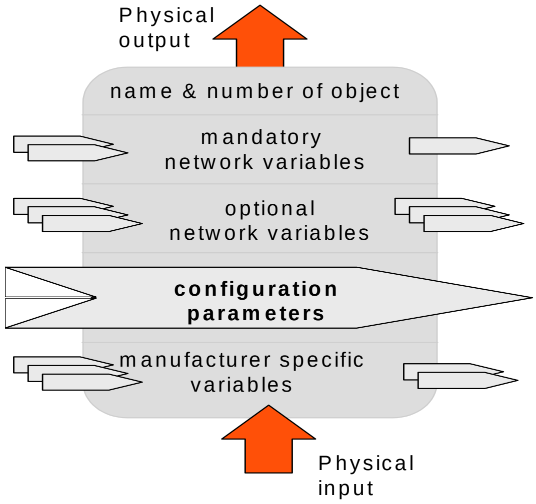
Functional profiles are prototype representations of devices which functionality can be described by generic objects such as actor, sensor and controller.

LonMark objects form the base of interoperability at the OSI application layer level. The LonMark objects describe standard formats for how information is input and/or output from a node and shared with other nodes on the network.

LonMark objects are defined as a set of one or more standard network variables /see section 12/ with semantic definitions relating the behaviour of the object to the network variable values, in addition to a set of configuration properties.

To provide for future expansion and to enable manufacturer differentiation, the LonMark object definitions comprise both mandatory and optional network variables as well as configuration sections.

The following picture shows the graphical representation of a prototype model for a node's application layer interface.



17 NodeBuilder® 3 Development Tool

The NodeBuilder development tool allows to create applications for the neuron chips and Smart Transceivers from Echelon. It includes a software suite for the Windows operating system and an hardware platform for prototyping I:

Software

- NodeBuilder Software Pack
- LonMaker 3.1 Network management tool
- LNS DDE Server OEM Edition

Hardware

- LTM-10A LonWorks Platform
- Gizmo 4 I/O board
- LonWorks Application Interface Module

The LNS-based LonMaker 3.1 allows designing, installing and maintaining LonWorks networks. It ease the test of prototypes nodes designed with NodeBuilder along with other devices. It is based on Microsoft's VISIO®2002 and then offers an intuitive graphical user interface.

The LNS DDE Server offers a simple but high-performance way of communication between the LNS engine servicing a LonWorks network and SCADA software's or an user interface. using the DDE mechanism of Windows.

The NodeBuilder Software Pack includes:

- **NodeBuilder Resource Editor:** which makes possible to define or display the types of variables and objects used by the application. These types may be standard or custom (user-defined).

- **NodeBuilder Code Wizard:** generates automatically the Neuron C source code of the application from a node definition built with a « drag and drop » user interface
- **NodeBuilder Project Manager:** edit, compile and load the Neuron C-written applications in the LonWorks nodes (LTM10A or others). It allows source-code debugging for any node in the project.
- **LNS Device Plug-in Wizard:** allows to generate (in Visual Basic language) and test automatically standard LNS Plug-in..

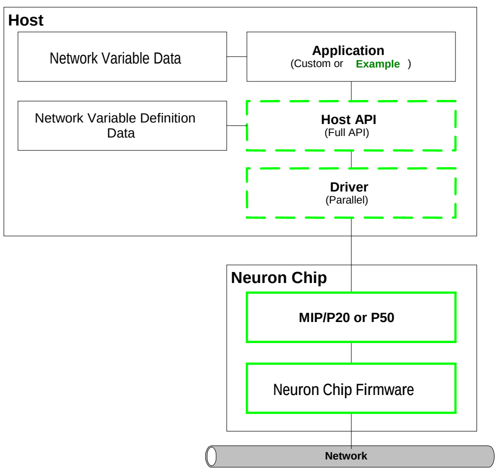
The NodeBuilder hardware platform includes:

- **LTM10A Module:** Generic LonWorks Node with Flash and Ram memory used as a debug platform by the NodeBuilder
- **GIZMO 4:** multi-I/O board for LTM10A or a TP/FT10A control module.
- **LonWorks MAI:** The MAI replaces the control module in a custom device so that the LTM10A platform can be used to debug the custom hardware.

18 MIP - Microprocessor Interface Program

The Microprocessor Interface Program (MIP) is a firmware library that transforms the Neuron chip into a communi-cations coprocessor for an attached host processor. MIP is recommended to implement complex application software that would be too large to run on a Neuron chip.

The MIP moves the upper layers of the LonTalk protocol from the Neuron chip to the attached host, extending the reach of LONWORKS technology to a variety of hosts including PCs, workstations, embedded controllers and µControllers.



The MIP is delivered as a Neuron C library that allows the NodeBuilder software to include system calls for the MIP. The NodeBuilder is needed to create a ROM image that executes on the Neuron chip.

The MIP also includes a sample host application, which illustrates how a host can send and receive network variables and explicit messages using the supplied network driver.

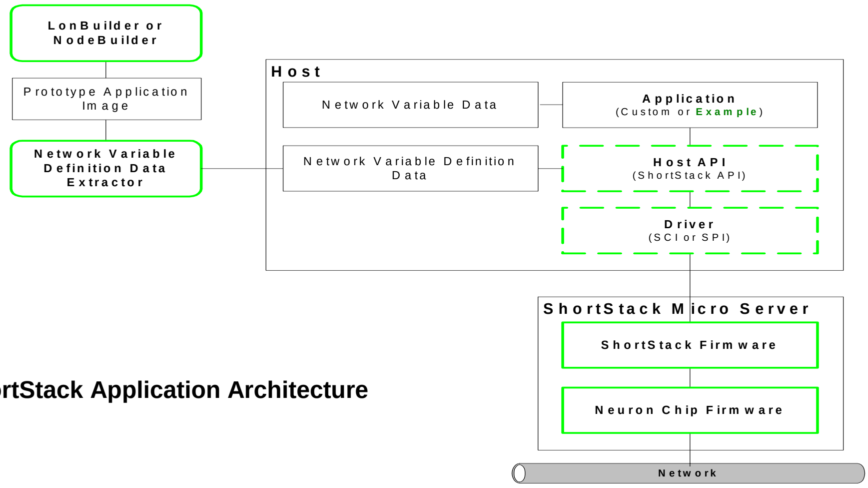
With MIP, the host processor can manage up to 4096 network variables. The algorithm which maps network variables into host addressable variables is included in the library package.

MIP provides a full network management and explicit message addressing support.

19 ShortStack Micro Server

Like MIP (see above), the ShortStack micro server is a software package, which converts the Neuron chip as a communication coprocessor for an attached host processor. ShortStack allows manufacturers to add sophisticated communication functions to their products while preserving their previous software investments. Their products can then, communicate with other smart products within a LonWorks® network and be remotely accessible through the Internet.

Only limited by imagination, such device could be a white good, a thermostat, a security device, an industrial or building automation controller.



ShortStack Application Architecture

Hardware-wise, a ShortStack™ micro server consists of a Neuron Chip and a FTT/PL transceiver attached to the host processor (which can be a 8, 16 or 32 bits microcontroller, microprocessor or DSP) through a serial port. The ShortStack Server firmware fits only Neuron Chip with at least 4Kb of application memory. The C-written API, which provides various LonWorks network access functions, typically needs around 4Kb of program memory space and 200 bytes of RAM when compiled for the host processor. No additional non-volatile memory. is required.

ShortStack can manage up to 62 network variables, supports explicit messages and provides functions for limited network management.

The ShortStack development Kit is free of charge, downloadable from the web site www.echelon.com/shortstack. It includes the ANSI C source code of the API to port on the host, the ShortStack object code to implement in the Neuron Chip/Smart Transceiver, an application example and the relative documentation.

Furthermore, ShortStack™ is royalty-free when used with standard Echelon transceivers (free topology, PowerLine or Link power).

20 LonTalk Interfaces

Host processor interfaces are frequently called LonTalk adapters. Such an interface provides, together with the processing power of the host, additional features and functionality to the host. Echelon offers both hardware and software products to built custom LonTalk adapters.

A NSI interface can be used to address all nodes in one domain (max. 32385 nodes). The NSI also provides firmware to access the services provided by an LNS server /see next section /, which is used to configure all devices on a network and maintain a directory of available devices and services on the network. NSIs are typically provided in operator interface panels, system monitors, data loggers, gateways, installation tools, maintenance tools and diagnostic tools.

Additionally, some interfaces allow to run in a specific NSI sub-mode called Virtual Network Interface (VNI). In this mode, a large part of the protocol stack is executed by the Host processor, allowing for highest performances and unlimited number of network resources.

The most commonly used LonTalk interfaces are:

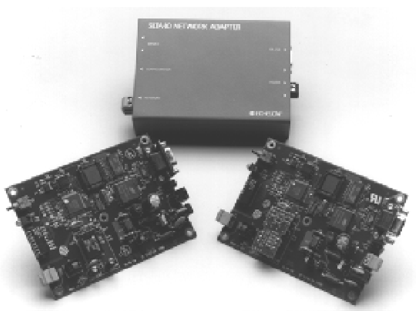
Product	Transceiver	Connection	Driver	Features
PCLTA-21	FTT10A,TP78, TP1250 or RS485	Bus PCI	WIN Emulated DOS	NSI Firmware support VNI mode support
Ilon-10	FTT or PL	Ethernet	WIN	IP Addressing Configuration Web Page NSI Firmware support
SLTA-10	FTT10A,TP78, TP1250 or RS485	RS232	WIN Emulated DOS	NSI Firmware support
PCC-10	FTT10A	PC-Card	WIN Emulated DOS	NSI Firmware support VNI mode support
PSG-20	User-selected	RS232	User-defined	Custom firmware NSI not supported
PSG-3	FTT10A,TP78, TP1250 or RS485	RS232	User-defined	Custom firmware NSI not supported

Besides the product choice offered by Echelon, a wide variety of third party products is available, allowing LONWORKS networks to be accessed from hosts with interface such as VMEbus, USB, PC/104, etc.

The PCLTA-21 interface for use with PCI-bus PCs is available with FTT10A or TP/XF1250 (1.25Mbps) transceivers.

The PCLTA-21 is NSI compatible

The PCC-10 is an NSI-compatible LonTalk interface with integrated FTT10A transceiver. The PCC-10 plugs into an PC-Card slot. Optionally available connection pods allow the user to interface other transceivers like TP/XF1250, PLT-xx, infrared etc.



The SLTA-10 is a NSI-compatible LonTalk adapter with a RS232 interface to the host computer.

The SLTA-10 is available with either FTT10A or TP/XF1250 transceiver.

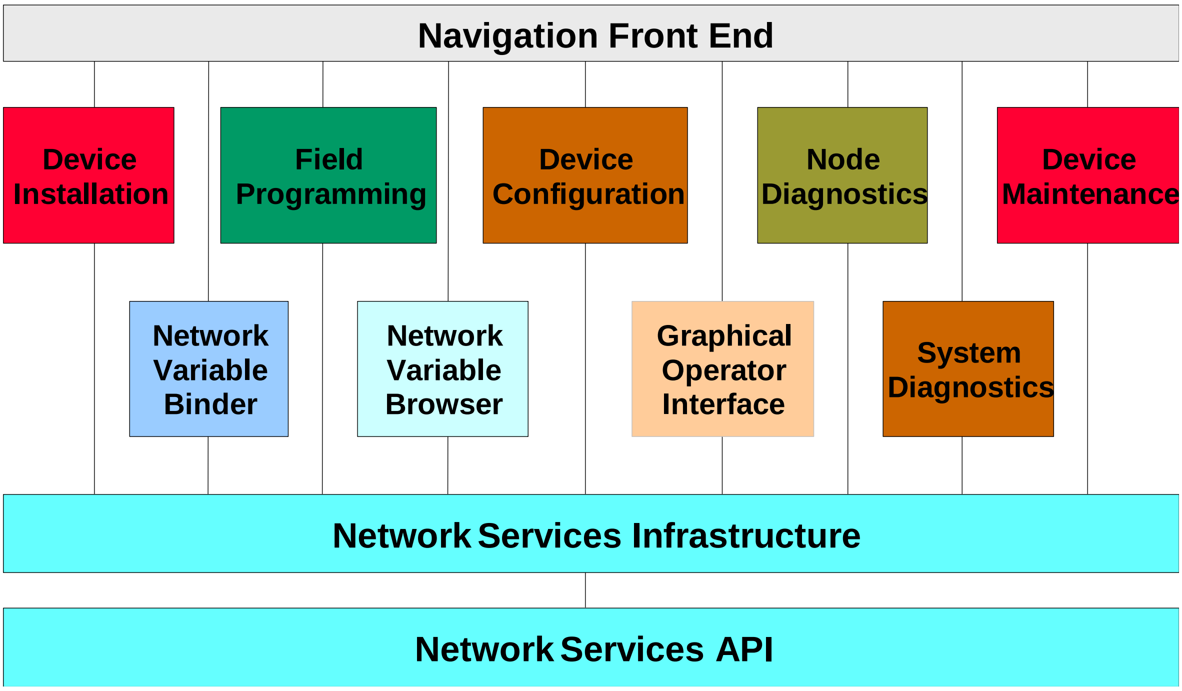
The RS232 interface supports baud rates of up to 115kbps.

The ilon-10 is an NSI compatible Remote Network Interface with an Ethernet 10 Mbps connection to the Host PC.

The ilon-10 is available with built–in FTT or PowerLine transceiver.

21 LNS - LonWorks Network Service

LNS is a network operating system for LONWORKS networks. It provides the essential directory, installation, management, monitoring and control services needed to efficiently master complex LONWORKS networks.



LNS provides a compact, object-oriented programming model that reduces development time, host code space, and host processing requirements. LNS represents the network as a hierarchy of objects that provide a set of services, contain a number of properties, and that report changes with events. To simplify development wherever possible, LNS automates common system tasks. E.g., it automatically discovers the presence of new unconfigured nodes on the network – without a tool's host having to do anything. LNS manages the network, freeing developers to focus on their application.

The LNS consists of two major components:

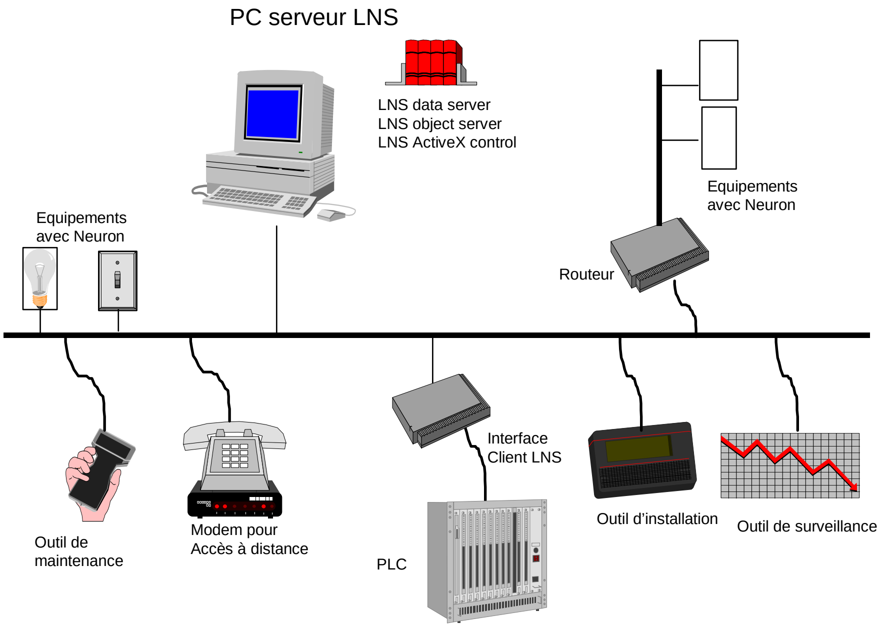
- the Network Service Server (NSS)
- the Network Service Interface (NSI).

The NSS hosts and processes network services. It maintains a network database. It enables and coordinates multiple points of access to its services and data. It also maintains a directory of all network and application service providers and event sources.

Clients speak to servers via a hardware component called Network Service Interface (NSI). NSIs provide the physical connection to the network and the messaging connection to the NSS. They automatically consult the NSS as needed to determine which server provide a given service and route the request transparently.

LNS is a client/server operating system with a single LNS server that supports many interoperating client applications. The LNS server S/W can run as a stand-alone application on a PC attached to the LONWORKS network. Clients on other PCs (called remote clients) can log into a LNS server to access the shared LNS database.

All nodes in a LONWORKS network are classified together as devices. Every device has local processing and input-output hardware. Each device can communicate with other devices using the LonTalk protocol.



LNS based network tools are available in various forms:

- LonMaker Integration Tool Version 3.1
- LNS Version 2.1 DDE server for Windows
- LNS Version 3 application development kit

22 LNS DDE Server

The LNS DDE server is a software package that allows any DDE-compatible Microsoft Windows application to monitor and control LONWORKS control networks – without any additional programming. Typical applications for the LNS DDE server include interfaces for HMI applications, data logging and trending applications, and graphical process displays.

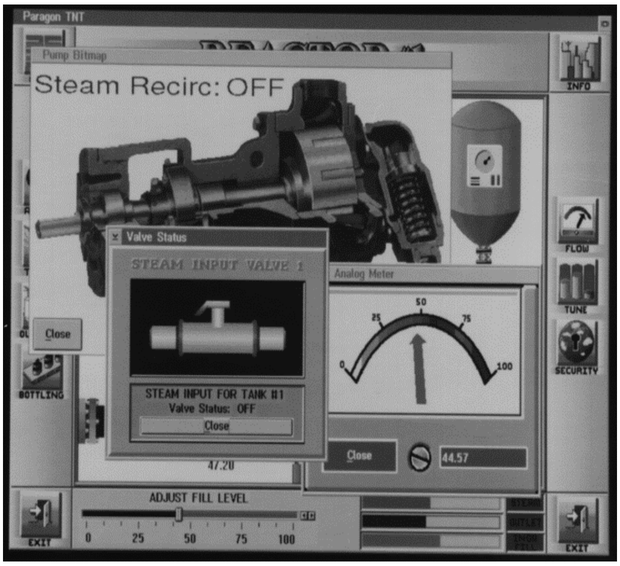
Using the LNS DDE server S/W and a NSI-compatible LonTalk adapter connected to a LONWORKS network, a PC may:

- read, monitor, and modify the value of any network variable.
- Supervise and change the configuration of nodes.
- Send and receive application messages.
- Test, enable, disable and override LonMark objects.
- Test, wink, and control devices.

The S/W is compatible with Wonderware's *InTouch*, Microsoft's *Excel* and Microsoft's *Visual Basic* and other widely used programming environments. It also supports Wonderware's *FastDDE* protocol for improved performance with InTouch.

No separate configuration step is required to use the LNS DDE server – LNS ensures that all of the required information is retrieved from the LNS data base.

Multiple PCs running the LNS DDE server may simultaneously access the same network, allowing several HMIs and maintenance tools to run at the same time.



Screenshot of a LONWORKS network monitoring Software based on the LNS DDE server

23 LNS Application Developer's Kit

The LNS Developer's Kit for Windows is a S/W development tool for designing and deploying open, interoperable LNS network tools for LONWORKS control networks. This product also provides the ability to redistribute the LNS network operating system as part of the developer's LNS network tool product.

The LNS network operating system can be used to design networks offsite (known as engineered systems), with device and router commissioning occurring later when the LNS Server is brought onsite. In addition, the LNS Server can be removed from the network after commissioning the network. This feature is especially desirable for smaller networks where an onsite management server is not equired.

To optimize performance and minimize network traffic, remote Windows-based client applications can cache directory information received from the LNS Server. These applications can then perform monitoring and control functions directly without interaction with the LNS Server. Client applications can read network variables using polled or event-driven updates, and can optionally filter redundant updates to minimize application overhead. Client applications can request that the LNS Server notify them if any of the cached information changes, ensuring consistency between the database and the caches. In addition, monitoring and control applications can continue to function if the LNS Server is not available.

Network variable and configuration property values on devices can be automatically converted to and from formatted strings to simplify user interaction. Formatting can be based on standard resource files for standard network variable types and standard configuration property types, or manufacturer-specific resource files for user-defined network variable and configuration property types.

LNS includes comprehensive support for the latest version of the LONMARK Interoperability Guidelines. LNS is able to manage certified and prototype LONMARK devices as well as other LONWORKS devices. The LONMARK objects on LONMARK devices can be easily controlled, allowing LNS applications to override, enable, test, or disable individual objects on a device.

Network variables can be accessed either by their device name or by their member name within a LONMARK object.

LONMARK configuration properties can be accessed as easily as network variables, even if the configuration properties are stored in the device's memory and not exposed as network variables.

Plug-in Component Standard

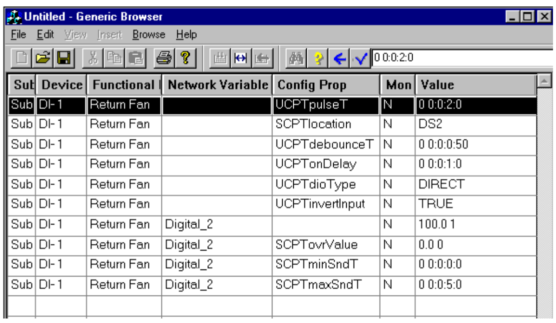
To provide interoperability between LNS applications from different vendors, LNS defines and supports a standard plug-in architecture where an LNS application can invoke the services of any other LNS application on the same PC.

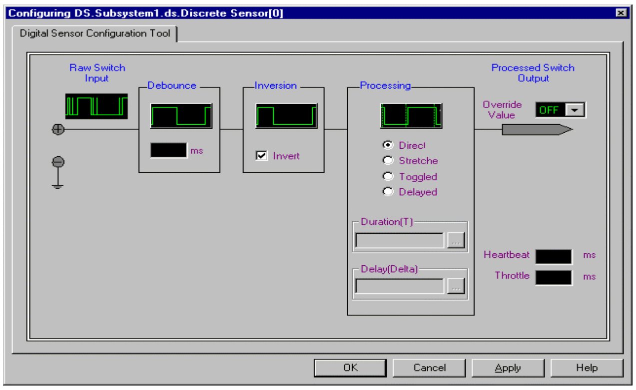
The LNS plug-in standard allows a single user interface or installation tool application to navigate or manage all the devices in a network, and then invoke device-specific applications (known as LNS Device Plug-ins) for any device on the network. LNS Device specific applications will typically be developed by device manufacturers to simplify the installation, configuration, or operation of their devices.

The LNS Device plug-in provide a configuration tool to the installer/integrator which does not need to be an expert of the product. Integrators do not need to be trained specificly for every product they have to deal with. This is particularly important for open systems where many different suppliers can be involved. A Device Plug-in of a given product can also be used for calibration, on-line help or programming tool.

Many applications support the LNS plug-ins : one of the most famous is the network management tool **Lonmaker Integration Tool**. Another significative advantage of the LNS Plug-ins is that, once created, they are usable by the integrators in any LNS based software including maintenance or SCADA software.

A listing of currently available LNS plug-ins is published at www.echelon.com/plugins.





The LNS plug-in standard supports a component application to be invoked on any type of object in the LNS Object Hierarchy, allowing such plug-ins as system plug-ins, subsystem plug-ins, channel plug-ins, Plug-in applications can also be developed for general purpose applications such as device drivers for HMI or SCADA applications.

Example Applications

Example applications are included which demonstrate how to use the LNS Object Server ActiveX Control. The examples offer a range of complexity, starting with simple tutorial examples written in Microsoft Visual C++, Microsoft Visual Basic, and Borland Delphi. Additional example LNS example applications are available on the LNS home page at http://www.echelon.com/lns.

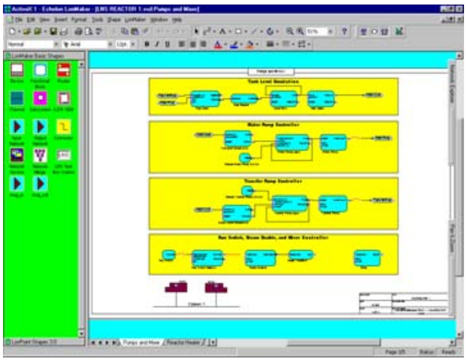
Licensing

A license to distribute copies of the LNS Servers and LNS Remote Client set up packages is included. A flexible royalty structure allows royalty payments based on the number of LNS Device Credits that are distributed with each LNS Server. One LNS Device Credit is required for each device managed by an LNS Server.

24 LonMaker Integration Tool

The LonMaker Integration Tool is a software package for designing, installing and maintaining multi-vendor, open, interoperable LONWORKS control networks.

Based on Echelon's LONWORKS network services (LNS) operating system, the LonMaker tool combines a powerful client/server architecture with a Visio® user interface. The result is a tool that is sophisticated enough to design, commission, and maintain a distributed control network, yet economical enough to be left on-site as a maintenance tool.



The LNS network operating system provides a standard platform for supporting interoperable applications on LONWORKS networks. LNS permit multiple applications and users to manage and interact simultaneously with a network. This feature allows multiple installers equipped with a LonMaker tool to commission devices on the network at the same time.

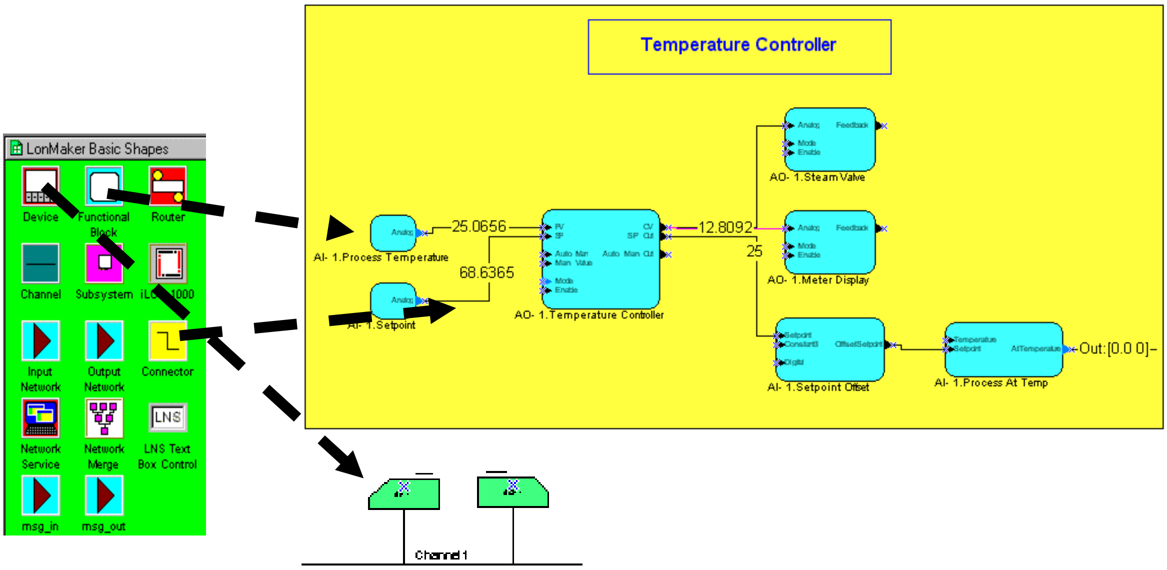
The LonMaker tool provides comprehensive support for LonMark® certified devices as well as other LONWORKS devices. The tool takes full advantage of LonMark features such as functional profiles, configuration properties, resource files, and network variable aliases.

LonMark functional profiles are exposed as graphical function blocks within a LonMaker drawing, making it easy to visualize and document the logic of a control system.

For engineered systems, network design is usually done off-site, without the LonMaker tool attached to the network. However, network design may also take place on-site, with the tool connected to a commissioned network. This feature is especially desirable for smaller net-works where adds, moves, and changes are a regular occurrence.

Users are provided with a familiar, CAD-like environment in which to design a control network. Visio's smart shape drawing feature provides an intuitive, simple means for creating devices. The LonMaker tool includes a number of smart shapes for LONWORKS networks, and users can create new custom shapes. Custom shapes may be as simple as a single device or a functional block, or as complex as a complete subsystem with predefined devices, function blocks, and connections between them.

Using custom subsystem shapes, additional subsystems can be created by simply dragging the shape to a new page of the drawing, a timesaving feature when designing complex systems.

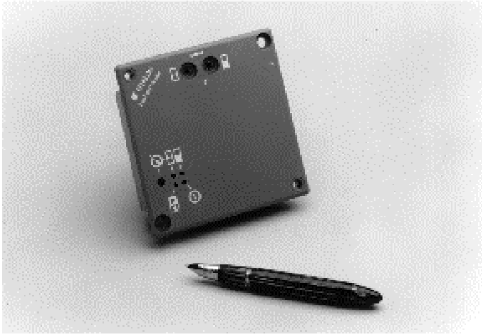


Network installation time is minimized by the ability of the installer to commission multiple devices at the same time. Devices can be identified by service pin, bar code scanning Neuron chip IDs, or manually entering the IDs. Testing and device configuration is simplified by an integrated application for browsing network variables and configuration properties. A management window is provided to test, enable/disable, or override individual function blocks within a device — or to test, wink, or set online and offline states for devices.

The LonMaker tool is the first installation tool to conform to the LNS plug-in standard. This standard allows LONWORKS device manufactures to provide customized applications for their products. These applications are automatically integrated into the LonMaker tool, making it easy for system engineers and technicians to define, commission, maintain, and test the associated devices.

In addition, the LonMaker tool can both import and export AutoCAD® files and generate as-built documentation. An integrated generator can also be used to generate a detailed report of the network configuration.

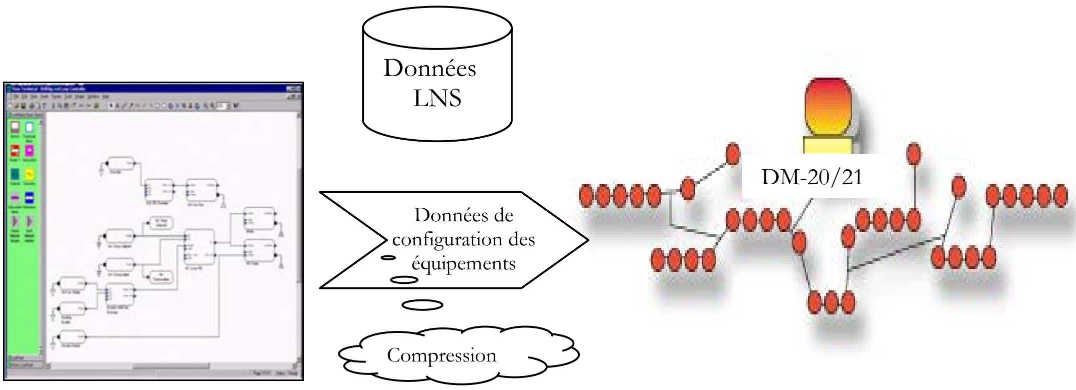
25 DM-21: Embedded Network Manager



The DM-20 and DM-21 Device Manager modules are embedded network management devices. The DM-20 is shipped as a printed circuit board suitable for mounting on a motherboard. The DM-21 is completely packaged with plastic faceplate and mounting hardware.

The DM-20/21 Device manager provide the following features:

- Automatic installation, fault detection, and device replacement of a LONWORKS network with up to 128 devices and one router.
- FTT10A free topology transceiver.
- Autonomous operation (without a local PC) after a database created with the LonMaker Integration Tool is loaded by the DM20/21 compression utility.
- Recording of system activity using an internal FLASH-based event log..
- Provides network access from the front-panel connector
- Quick installing and low cost design.
- Support for LonMark 3.0 objects and configuration properties.
- Support for both network variables and message tags.



The above diagram illustrates the three steps necessary to successfully use the DM-21/21 to manage your network:

1) Create your network using the LonMaker Integration Tool.
2) Export the managed device list(s) to the DM-20/21 using the DM-20/21 compression utility.
3) Attach the DM-20/21 module to your network.

Once attached, the DM-20/21 will automatically find and commission all the devices defined in the compressed LNS database.

26 *i*.LON™ Internet Servers

The *i*.LON™ Internet servers build a relable connectivity between the LonWorks control networks and the most common data networks using the Internet (IP) protocol. This capability allows creating virtual private control networks for many types of application :

- Connect together the LonWorks networks on every floor of a large building through an high speed Ethernet backbone.
- Connect the machines of a manufacturing plant to the enterprise's LAN.
- Send data from remote sites to a central service center through an Internet Wan
- Monitor and control an automated system through Internet with a web browser.
- Check remote equipment status through a Wireless LAN.

26.1 iLon-10 less expensive than moving the truck

The iLon-10 is the smallest and simplest model in the iLon product range.. It is essentially a Remote Network Interface. (RNI). It allows connecting to a LonWorks network through the Internet.

Designed to be installed in various building or residential remote control applications, it is available either with an FTT or a PowerLine transceiver.

Used with an LNS-based software application and linked to the enterprise management software, it allows the reliable and secure remote monitoring, control and configuration of devices dedicated to automation, measurement or metering..

Several thousands of remotes sites distributed over an area can be fully managed remotely:. Technical Alarms, storage levels, etc. are immediately sent to the management center through an uplink connection.

Its very low cost is paid back by the first time it saves you to move the truck to the remote site.

26.2 iLon-100 Internet Remote control Gateway

The iLon-100 features a complete set of remote control functions that can be used in an autonomous way while remaining accessible though the Internet.

The embedded Web server of the iLon-100 allows accessing easily any data (as network variables or explicit messages) from the control networks. This function, password protected, allows monitor and control data from a LonWorks network from anywhere – without the need of a specific software – through a LAN, WAN or the Internet.

The iLon-100 includes many schedulers, data loggers – either periodic or event-driven and sophisticated alarm managers. All of them are accessible through standard by-default Web pages that can nevertheless be totally redesigned by the user as per application-related Web pages.

Using the latest Internet technologies and providing a large number of WebServices for every of its functions, the ilon-100 exports the recorded data in XML format through SOAP messages. It is, as such, very easy to interface with any Microsoft .net based software, for example.

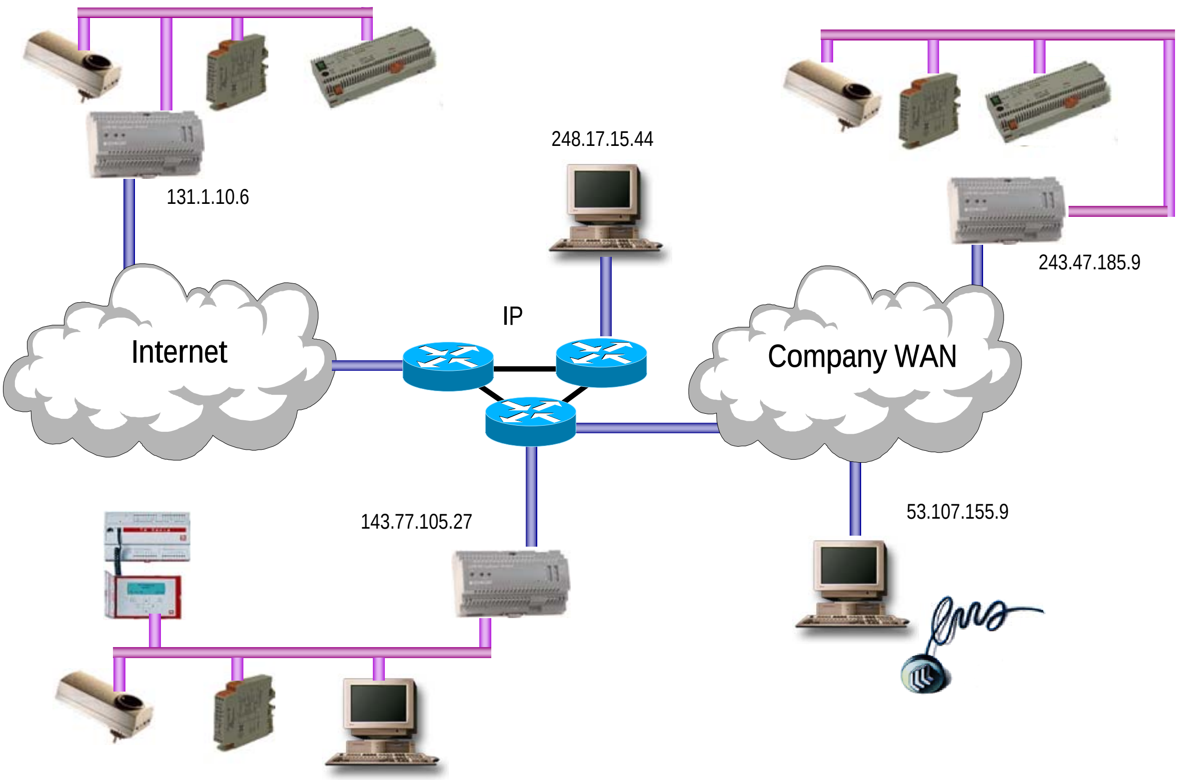
The ilon-100 identifies alarm conditions, records them and route them to email recipients. Additionally, a remote management center can manage the alarms through the SOAP interface.

Available either with a FTT twisted pair transceiver or a PowerLine transceiver, it features also its own digital inputs, relay outputs and metering inputs. Coming in standard with an Ethernet 10/100Mbps port, it may optionally be equipped with an internal V90 modem for an easy connection to any Internet provider.

Like the iLon-10, the Ilon-100 is also a Remote Network Interface that can be used with any LNS based software.

26.3 i.Lon-600 for high-speed routing

The iLon-600 is a high performances, highly reliable router coming in a rugged DIN-Rail mounting package protected against electromagnetic interferences .The iLon-600 is the result of the Echelon's know-how in layer-3 routing and conforms to the EIA-852 norm. It allows high-bit rate in control applications for the process control industry, the telecom industry, the utilities, and the transport…



LonWorks Packets are encapsulated within the TCP/IP packets.
Every iLon-600 has its own IP address.

The iLon-600 is unique in its ability to support as well peer-to-peer communications as master-slave. Il allows remote devices to communicate to each-other through the IP networks as they were on the same LonWorks channel.. For example, devices in various floors of a building, or various buildings spread in a campus can be bound together in a fully transparent way and without communication losses.

The impressive performances of the iLon-600 come from the successful combination of a powerful RISC 32 bits processor and the Echelon's *Virtual Network Interface* (VNI) software architecture. Both allow for a high throughput in control networks with large number of nodes, providing high refresh rate of data handling or SCADA displays.

The iLon-600 can be installed with standard LonWorks installation tool like the *LonMaker Integration Tool*™.

Moreover, the iLon-600 can be pre-configured through its setup web page..

From the IT world's perspective, the iLon-600 is a good citizen. As many IP devices, the iLon-600 supports several standards communication protocols: TCP/IP, DHCP, ICMP, SNMP, MD5. Moreover, packet aggregation, addressing, bandwidth use and security features can be fine tuned by the user.

Functions	ILon-10	ILon-100	Ilon-600
Remote Network Interface	√	√	
Web Server		√	
WebServices		√	
Router (EIA-852)			√
Applications	**ILon-10**	**ILon-100**	**Ilon-600**
Alarm Manager		√	
Data logger		√	
Scheduler		√	
Interfaces	**ILon-10**	**ILon-100**	**Ilon-600**
Ethernet 10MBps	√	√	√
Ethernet 100MBps		√	√
Internal V90 Modem		√ (optional)	
LonWorks Interface	FTT10 or PL	FTT10 or PL	FTT10 or TP1250
Digital I/O		√	
Metering inputs		√	
Internet Protocols	**ILon-10**	**ILon-100**	**Ilon-600**
PPP	√	√	
DHCP	√	√	√
FTP		√	√
SNTP		√	√
SMTP		√	
SOAP		√	
MD5	√	√	√
NAT Compatible	√	√	√

27 Additional Informations

1. Echelon LonWorks Products Catalogue
2. Echelon Neuron C Reference Guide
3. Echelon Neuron C Programmer's Guide
4. Echelon LonMaker Integration Tool User's Guide
5. Echelon LNS DDE Server User' Guide
6. Echelon Documentation CD 2004 (part # CDROM-ECH2)
7. Echelon The LonWorks Network Services (LNS) Architecture Technical Overview
8. LonMark Application Layer Interoperability Guidelines Rev. 3.3
9. F.Tiersch LonWorks Technology – An Introduction DESOTRON Verlagsgesellschaft, Erfurt 2000 - ISBN 3-932875-11-7
10. www.echelon.com
11. www.ebv.com
12. www.toshiba.com
13. www.lonmark.org
14. www.lontech.ch Swiss LonUsers Web Site
15. www.lonuser.asso.fr French LonUsers Web Site
16. www.lno.de German LonUsers Web Site
17. www.ansi.org to get the ANSI/EIA 709.1 protocol specifications