User Manual

Gesytec GmbH
Pascalstr. 6
52076 Aachen, Germany

Tel. + (49) 24 08 / 9 44-0
Fax + (49) 24 08 / 94 4-100
email: info@gesytec.de
http://www.gesytec.com

Doc. ID: LPP/UserDoc/LPP3/LPP_Manual-E-V3.5.docx,
Version v3.5, Date: 24.6.2014

Gesytec

This manual …

… provides you with all the information which you will require to use the Easylon PCI Bus Interface card.

However, this manual will neither explain aspects of Echelon's® LONWORKS® technology, nor Echelon's Microprocessor Interface Program (MIP), nor Echelon's Network Service Interface (NSI) used on this network interface card nor details concerning the Easylon PCI Bus Interface network drivers, which has been designed in accordance with the driver specifications of the Echelon Corporation. For further information on the LONWORKS technology please refer to the extensive documentation provided by Echelon. Especially Echelon's "LONWORKS Host Application Programmer's Guide" will be required if applications are to be developed using Gesytec's Easylon PCI Bus Interface as a network interface.

After a general presentation of the Easylon PCI Bus Interface card in Chapter 1, Chapter 2 describes the necessary steps to install the card.

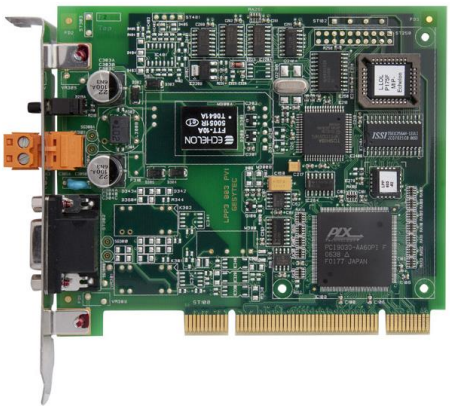
Chapter 3 contains a general technical description.

Chapter 4, “Programming Instructions”, contains the information which will be of importance, should you wish to develop your own network driver software for the Easylon PCI Bus Interface.

Contents

1 Product Information

This manual describes the Easylon Interface card:



Easylon PCI Bus Interface,
card for 32-Bit PCI Bus Slots.

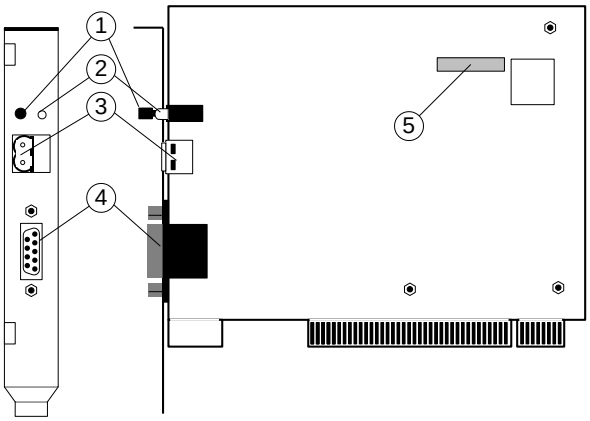


Figure 1-1 Easylon PCI Bus Interface

(1) Service button
(2) Service LED
(3) Screw-plug terminal (not with EIA-485 variants)
(4) 9 pin D-type connector
(5) Type identifier on rear side (see table 1.1)

1.1 Variants

The following variants of the Easylon PCI Bus Interface are currently available and are described in this documentation. Each variant is identifiable by a type code sticker on the rear of the card.

Type Code	Order Code	Transceiver	Variants
LPP.AA	P.P10204-3	FTT	MIP/P50
LPP.AC	P.P10214-3	FTT	NSI
LPP.BA	P.P10201-3	TP/XF-78	MIP/P50
LPP.BC	P.P10211-3	TP/XF-78	NSI
LPP.CA	P.P10202-3	TP/XF-1250	MIP/P50
LPP.CC	P.P10212-3	TP/XF-1250	NSI
LPP.DA	P.P10203-3	RS-485	MIP/P50
LPP.DC	P.P10213-3	RS-485	NSI
LPP.EA	P.P10205-3	Direct Connect	MIP/P50
LPP.EC	P.P10215-3	Direct Connect	NSI
LPP.FA	P.P10206-3	FTX	MIP/P50
LPP.FC	P.P10216-3	FTX	NSI

Table 1-1 Variants, order-codes and type identifiers for Easylon PCI Bus Interface cards

1.2 Scope of Delivery

- PC plug-in card with Echelon‘s MIP/P50 or NSI firmware
- Mounting instruction
- Drivers & Documentation CD with
  - network drivers for 32 bit and 64 bit versions of Windows XP, Vista, 7, 8, Server 2003, Server 2008, Server 2008 R2, Windows CE, Linux and MS-DOS
  - Easylon RNI Software for remote LONWORKS access
  - EasyCheck utility for Easylon Interfaces
  - WLDV32.DLL
  - Documentation in Adobe Acrobat .PDF format

1.3 Overview

The Easylon PCI Bus Interface is a cost-effective link between a PCI bus computer and the LONWORKS control network. The PC plug-in card for a PCI bus

slot provides one interface to the network. Variants are available for connection via transformer coupled twisted pair (TP/XF), free topology (FTT) and Direct Connect transceivers and for EIA-485 serial connection.

The Easylon PCI Bus Interface card is provided with a service button and LED. In the TPT/XF and FTT variants, the card is equipped with two connectors, either of which can be used:

- 9 pin D-type connector
- 2 pin screw-plug terminal

In OEM versions the D-type connector can optionally be replaced by an RJ45 connector and Phoenix connectors can be used instead of the usual Weidmüller.

As an alternative solution to the standard LonTalk adapters described in this documentation there is a modern solution available with the Easylon Interfaces+. Especially in more demanding applications these ISO/IEC-14908 based LonWorks compatible network interfaces offer many advantages. Find more information at www.gesytec.com.

2 Installation

Installation of the Easylon PCI Bus Interface Card is carried out in two steps:

1. Insertion of the card into the PC
2. Installation of the network driver

For specially developed applications which shall use the Easylon Interface cards the status setting has to be taken care of. Chapter 4.2 provides further information on this subject.

The external interface files (.xif) for the different card variants can be found on the accompanying CD-ROM. Which ".xif"-file belongs to which variant can be found in Table 4-1.

2.1 Insertion of the Card

When inserting the Easylon PCI Bus Interface card in your computer, please be sure to observe all the computer manufacturer's instructions regarding the insertion of additional interface cards.

The Easylon PCI Bus Interface Card is to be inserted into a vacant 32-Bit PCI slot while the computer is switched off.

The address and the interrupt are automatically assigned by the computer’s PCI BIOS.

Please refer to section 3.5, for information regarding the connector pin assignments for connection to the LONWORKS network.

2.2 Driver Installation

Drivers for different operating systems are available for the Easylon PCI Interface. Currently these are Windows 2000, XP, Vista and 7 and the Windows Server OS 2003, 2008 und 2008 R2. The drivers support both, the 32 and the 64 bit version of these operating systems. Furthermore there are drivers for Windows CCE, Linux and MS-Dos. Latest driver versions you can downloadvia the Easylon Support pages of our web site: www.gesytec.com
Installation is described in the following sections:

16-Bit driver under 32-bit Windows	chapter 0
DOS Driver	chapter 2.2.4

This section also describes in short the diagnosis utility “EasyCheck” which can be installed separately from CD.

A Linux driver is available in source code from the “Linux” directory on the CD. This also contains hint s and comments.

The “Driver and Documentation” CD will lead you to the installation of drivers for different operating systems (OS). However, all setups can as well be started manually for each OS and the respective interface card directly from the CD.

2.2.1 Driver for Windows Operating System (WDM Drivers)

This section describes installation and setup of the Easylon Interface card drivers for the Windows operating system from Windows XP onwards.

The setup program is using the same WDM driver (Windows Driver Model) for all operating systems.

Note: For installation you can either use the Windows assistant or the program FastUpd.exe for manual installation. The latter is much more directly and especially helpful if you have to install several instances of the driver.

Finally de-installation of the driver is explained

2.2.1.1 Installation using the Windows Assistant

After the Easylon Interface has been mounted Windows will recognize the new device and start the hardware assistant.

If Windows should not find the driver on the Drivers and Documentation CD or the driver should be elsewhere, please select the appropriate drive and select the setup file „LppWdm.inf“ and the driver „Gesytec LPP WDM Driver PCI“ from the LPCLPP directory.

2.2.1.2 Manual Installation and Update

The easiest way to install the driver is to ignore the hardware assistant and run

FastUpd.exe or **FastUpd64.exe**[1]

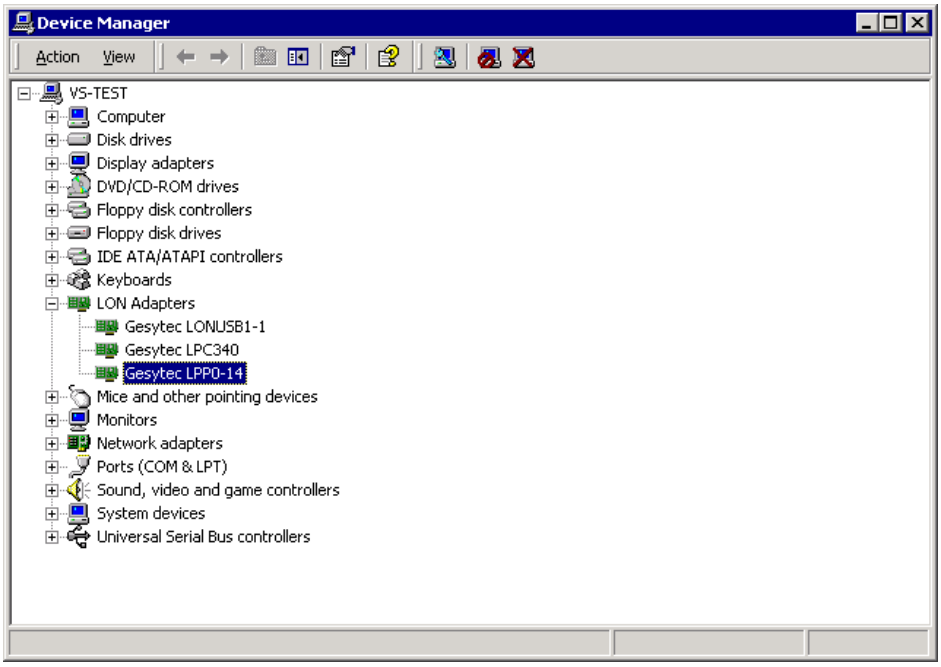
from the „ LpcLpp “ folder of the CD-ROM.

[1] For 64-bit systems

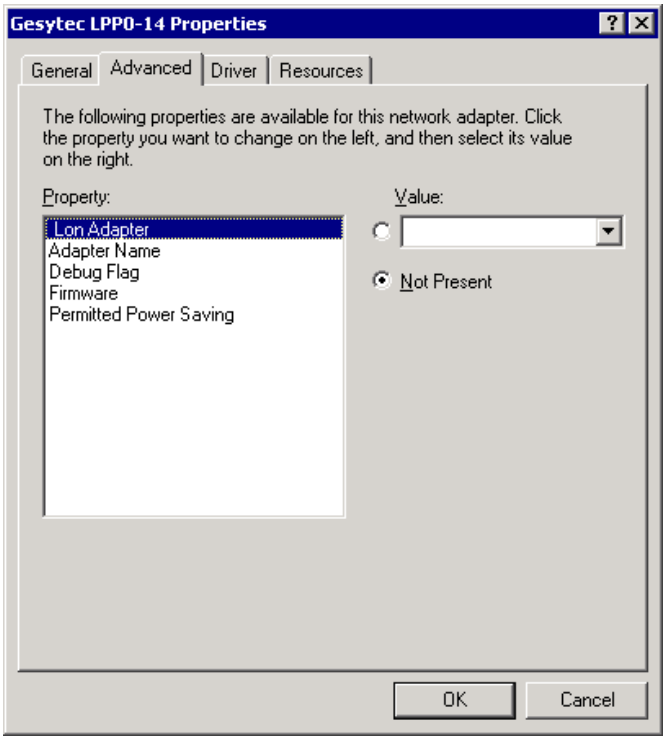
The same program you will use to update an existing driver.

2.2.1.3 Parameter Setting

Certain operating conditions may require special settings for the Easylon Interfaces. In the Device Manager select the respective interface board under LON Adapters and modify the settings.



The „Advanced“ TAB allows setting of individual properties.



Lon Adapter

You can assign a name „LON1“ ... „LON9“ to the interface board, as some applications may require.

ATTENTION The name chosen must not be in use by any other driver. The device will not start if a name is used twice. (Code 10).

Adapter Name

Alternatively a name can be chosen freely (e.g. building 7).

NOTE If names are given in both sections „Lon Adapter“ and „Adapter Name“ only the name defined under „Lon Adapter“ will be used.

Debug Flag

This field contains a DWORD in hexadecimal notation of different flags for debug purposes. Usually it is set to 0 (= not existing). By setting the single bits certain debug features can be turned on. At the moment the bits 0, 1 and 2 are used:

Bit 0: LON telegrams at the interface from and to the application are displayed in the debug output.

Bit 1: LON telegrams at the interface from and to the Neuron Chip are displayed in the debug output.

Bit 2: LON telegrams at the interface from and to the Easylon Watcher[2] are displayed in the debug output.

Bit 3: CREATE and CLOSE) of the driver are displayed in the debug output.

NOTE: The Debug Output can be displayed using, for example, the program DebugView, available at www.sysinternals.com.

Firmware

The options MIP/NSI or EEBLANK are displayed.

Permitted Power Saving

Usually the Easylon Interfaces allow with applications running the standby mode as well as the hibernate mode. However, in certain operating conditions problems may arise if the PC, with a LON application running, changes to hibernate or standby mode. This can be turned off by selecting "None".

2.2.1.4 De-Installation

WDM drivers are de-installed using the "Device Manager". Among "LON Adapters" select the "Gesytec LPPx-yy" driver and click "de-install".

2.2.2 EasyCheck – Test Utility for Windows Drivers

In addition to the drivers, the test utility "EasyCheck" can be installed in the respective program directory (default: : \Easylon\Lpx). The program checks interface and software environment and displays information, from which can be concluded on the reasons for problems in connection with the interface.

EasyCheck runs an analysis of the system's software. It will open the selected interface, check the driver version and display it. By sending a "query status" command the communication with the hardware is tested. Using the "read memory" command the utility will show if the device is running MIP or NSI firmware. Properly installed Easylon Interfaces will send a corresponding answer.

2.2.3 Windows and 16 Bit Applications

The Windows driver for the 32 bit Windows versions also provides a 16 bit interface. (Unfortunately Microsoft does not support this in the 64 bit versions.) To use it, the following entry has to be made in the file „config.nt", usually found in the windows\system32 directory:

[2] The Easylon Watcher cannot be used with this interface card version. Information is only provided for the sake of completeness.

Device=%SystemRoot%\system32\ lpxdos.exe –Llppwdm0-14

The 32 bit LON device used is specified by the optional –L or /L parameter:

/Lname

name =

lpp0-14	for device LPP with PCI-Bus-number 0 and PCI device number 14

Note: Two subsequent “l” characters have to be entered, one indicating the parameter -L, the second as first character of the name: –Llxxxx

The 16 bit LON device used is specified by the following optional parameter:

/Dn

with n = 1...9 for LON1 to LON9

Without this parameter, the interface will be assigned the first unused name starting with “LON1”.

2.2.4 DOS Driver

The network driver for MS-DOS supplied with the Easylon Interface card has been designed in accordance with the specifications by Echelon Corporation. For information on the network driver interface which is required to develop applications, please refer to the "LONWORKS Host Application Programmer's Guide" from Echelon.

The driver can be taken form the CD-ROM’s „DOS“ directory.

„lppdrv.exe“

The driver file „lppdrv.exe“ has to be copied onto the hard disk of your computer, e.g. into a directory named C:\easylon.

The network driver for the Easylon Interface cards requires 1.6 Kbytes of resident program code, 2 Kbytes of output buffer and 2, 4 or 8 Kbytes of input buffer.

2.2.4.1 Installation

The network driver will be installed in the system as device with the first free name starting with „LON1:“ by adding in the "config.sys" file the line

device =*{path}*\lppdrv.exe */I /D*

or

devicehigh =*{path}*\lppdrv.exe */I /D*

{path} is describing the location of the lppdrv.exe file in your system.

Options

The following options can be used:

/D Setting of device number
The device number may be in the range from 1 to 9 (LON1: – LON9:). If this option is not specified, the network driver will be defined as LON1: (default). If another network driver has already been installed with the same device number, this will result in the error message:

Invalid or duplicate device name

If the parameter /D is specified without entering a subsequent numerical value, the device number will be assigned automatically. If all possible device numbers have already been assigned to other network drivers, this will result in the error message:

LON1: ... LON9: already defined

/I Increasing the input buffer
The input and output buffers of the Easylon Interface network driver are configured as byte-level FIFOs, i.e. the space requirement of a message is dependent on its length. Consequently, a buffer capacity of 2 Kbytes (default, approx. 50–100 messages per buffer) should be quite adequate in most cases. However, should it be necessary to store an even larger number of incoming messages, the input buffer can be increased. Valid values for parameter /I are 2, 4 or 8 (Kbytes).

Instead of the slash, '/', it is also possible to enter a dash '-' to identify the options. No distinction is made between upper case and lower case characters.

Multiple Easylon Interface cards installed

The network driver for the Easylon Interface cards only supports one interface card. If several of these cards are installed in the computer, the network driver must be installed an appropriate number of times in the 'config.sys' file. The Easylon PCI Bus Interfaces are installed according to the sequence of their PCI bus slots.

If it is established during loading that another network driver has already been installed for the Easylon Interface, the copyright message will be suppressed.

2.2.4.2 Display of the Network Drivers Installed in the Computer

The 'lppdrv.exe' file can be called from the DOS command line in the same manner as any program to show all the network drivers installed in the system, the appurtenant device names and their storage requirements.

/R The option /R additionally enables modification of the device number.

Example: lppdrv -r13
changes the name LON1: to LON3:

If the first device number does not exist, or if the second number has already been assigned to another device driver, the message

Invalid or duplicate device name

will appear.
Renaming device names is not restricted to network interface devices defined by this Easylon Interface network driver and can also be applied from a Windows DOS box as a global function for the entire system, including 16-bit Windows applications.

3 Technical Description

3.1 Network Interface

The Easylon PCI Bus Interface is based on the NEURON 3150® Chip. Under MIP/P50 firmware the NEURON Chip is operated with up to 32 Kbytes ROM as program memory and 24 Kbytes SRAM as data memory. For the NSI firmware versions the memory is 48 Kbytes ROM and 9 Kbytes SRAM. It is connected to the PCI bus in slave_A mode.

For monitoring purposes, a reset flip-flop is additionally implemented on the NEURON Chip; a reset of the NEURON Chip can be identified by the PC via a status byte. The same mechanism is implemented for the interrupt flip-flop.

In order to visualize the status and to initiate the service function of the node, the service pin of the NEURON Chip is available in the front panel with service LED and service button to activate the function (cf. fig. 1.1).

3.2 Bus Interface

The PCI Bus interface has been developed as 32-Bit I/O interface according to the “PCI Local Bus Specification, Version 2.2”. It can be used in PCI slots providing a 3.3 V signal as well as in PCI-X slots. In PCI-X machines it may be necessary to enable the Easylon Interface in the BIOS. Please refer to the respective computer manual.

Note: Please do not confound PCI-X and PCI express. While the first uses a parallel communication the latter uses a serial one.

The Easylon PCI Bus Interface card will be assigned the I/O addresses from the PCI BIOS. The necessary addresses consist of one addresses for data transfer between the PC and the network interface card, one address for status queries and control of the NEURON Chip.

3.3 Reset Procedure, System Control

Reset of the NEURON Chip of the interface card can be initiated by the PC, via a program-controlled function. After a system reset the NEURON Chip starts up automatically.

3.4 Block Diagram

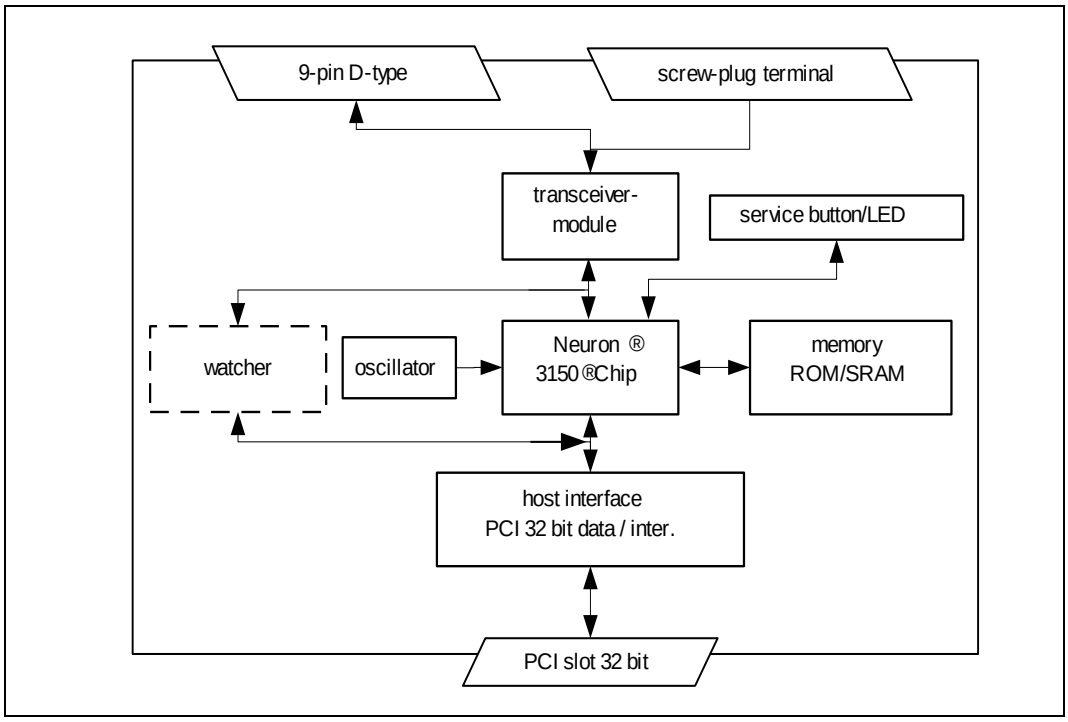


Figure 3-1 Block diagram Easylon PCI Bus Interface

3.5 Connector Pin Assignments

The Easylon PCI Bus Interface provides a 9-pin D-type connector (Figure 1-1 Ã) for network connection. In the TPT/XF and FTT variants, the card is additionally equipped with a 2 pin screw-plug terminal (Figure 1-1 ✹), which can be used alternatively.

Connector type	Pin	Signal	Remark
9-pin D-type	1	data	Data
	2	data	Data
	3	—	Reserved
	4	—	Reserved
	5	—	Reserved
	6	—	Reserved
	7	—	Reserved
	8	—	Reserved
	9	—	Reserved
2-pin screw-plug Terminal	1	data	Data
	2	data	Data

Table 3-1 Connector pin assignments for TPT/XF and FTT network connection

Connector type	Pin	Signal	Remark
9 pin D-type	1	—	Reserved
	2	—	Reserved
	3	DA-	Data
	4	SA-	RTS –
	5	—	Reserved
	6	+5 V	U+ supply voltage, electrically isolated
	7	0 V	U– supply voltage, electrically isolated
	8	DA+	Data
	9	SA+	RTS +

Table 3-2 Connector pin assignments for EIA-485 network connection

3.6 Service LED

The service LED (Figure 1-1 ✸) signals the card status. The following signals are defined the service LED:

Service LED	Status	Remarks
Flash (1 Hz)	No driver installed or driver conflict.	Check the driver settings, IRQ- or address conflict to other cards?
Blink (1/2 Hz)	Driver installed, node is “unconfigured”.	Configure the node.
Permanently ON	Node is „applicationless“ and „unconfigured“.	
Permanently OFF	Installation ok	Normal operation

Table 3-3 Service LED

3.7 Technical Specification

Bus Interface	32 bit Data (I/O), according to „PCI Local Bus Specification, Version 2.2“ compatible with 3.3 V and PCI-X	
I/O addresses	3	
Control register	8 Bit	
Status register	8 Bit	
Compatibility	LonTalk, ISO/IEC-14908.1	
CPU	Neuron 3150, 10 MHz	
Coupling	parallel, slave_A mode	
Memory		
MIP/P50:	ROM	32 Kbytes
	RAM	24 Kbytes
NSI :	ROM	48.75 Kbytes
	RAM	9 Kbytes

Network interface

Order Code*	Network Interface	Transmission Rate	Network Connector	Protection
P.P10201 P.P10211 P.P11201 P.P11211	TPT	78 kbps	9-pin D-Type + 2 pin screw-plug terminal	Sparc-gaps
P.P10202 P.P10212 P.P11202 P.P11212	TPT	1.25 Mbps	9-pin D-Type + 2 pin screw-plug terminal	Sparc-gaps
P.P10203 P.P10213 P.P11203 P.P11213	EIA-485, electr. Isol.	39 kbps	9-Pin D-Type	Zener Diode
P.P10204 P.P10214 P.P11204 P.P11214	FTT	78 kbps	9-pin D-Type + 2 pin screw-plug terminal	Sparc-gaps
P.P10205 P.P10215 P.P11205 P.P11215	Direct Connect	line length dependant	9-pin D-Type + 2 pin screw-plug terminal	Sparc-gaps
P.P10206 P.P10216 P.P11206 P.P11216	FTX	78 kbps	9-pin D-Type + 2 pin screw-plug terminal	Sparc-gaps

* cf. Table 1-1 for variant identifiers on the card.

Voltage Supply	5 V, from PC
Power consumption	typically 2 W
Temperature	
operational	0 °C - 50 °C
non-operational	-20 °C - +70 °C
EMC	EN 610 00-6-2 EN 550 22 A/B
Humidity	according to DIN 40040, Class F
Dimensions	130 mm x 105 mm, incl. connectors; for 32-Bit PCI Slot

4 Programming Instructions

4.1 LONWORKS Network Node

The Easylon Interface card is a network node in the LONWORKS network. It is operated under Echelon's Microprocessor Interface Program MIP/P50 or with NSI firmware using the NEURON 3150 Chip as communication processor. The appropriate external interface files (.xif) are on the installation CD-ROM. Which .xif-file is describing which interface card variant is shown in the following table:

Network Interface	Transmission Rate	XIF –File
TP	78 kbps	lolp072f.xif
TP	1.25 Mbps	lolp073f.xif
EIA-485, electr. isol.	39 kbps	lolp074f.xif
FTT	78 kbps	lolp075f.xif

Table 4-1 Card variants and .xif files

4.1.1 Network Interface

The various network interface variants are each operated directly via the communications port (CP0...CP4) of the NEURON Chip.

Configuration is done automatically by the MIP/P50 firmware. The NSI variants are delivered with the correctly configured network interface. In case of problems they have can be reconfigured with the "EasyCheck" utility.

4.1.2 Node CPU

The interface node is designed on the basis of the NEURON 3150 Chip. The standard clock pulse for the processor is 10 MHz. Under MIP/P50 firmware the processor is equipped with a 32 Kbytes ROM as program memory. A 24 Kbytes SRAM serves as the data memory. The memory for NSI firmware is 48.75 Kbytes ROM and 9 Kbytes SRAM.

The 11 I/O ports of the NEURON Chip are all used for parallel coupling with the PCI bus interface.

The status of the service pin of the NEURON chip is indicated by a LED. The service function can be activated via the service button (cf. fig. 1.1)

4.1.2.1 Coupling Neuron Chip ↔ PCI Bus Interface

The NEURON Chip is coupled in slave_A parallel mode. The handshake bit defined by the NEURON Chip (NHS) to control the data flow can be checked via the status byte of the Easylon PCI Bus Interface. Please refer to the NEURON 3150 Chip data book with regard to the data communication mechanism in slave_A mode.

4.1.2.2 Interrupt Function Neuron Chip ↔ PCI Bus

An interrupt flip-flop is set via write access to a defined memory address. This interrupt flip-flop is reset (acknowledged) by the host via access to an I/O address.

Reading back the interrupt status by the NEURON Chip is not possible.

4.1.2.3 Neuron Chip Address Map

Address Range			Module
$0000 … $7FFF	&	Read	ROM 32 Kbytes, program memory
$8000 … $DFFF	&	Read/Write	SRAM 24 Kbytes, data memory
$E000 … $E7FF	&	Write	Set Interrupt Flip-flop
$E800 … $FFFF			NEURON Chip internal

Table 4-2 NEURON Chip address map of MIP/P50 version

Address Range			Module
$0000 … $C2FF	&	Read	ROM 48,75 Kbytes, program memory
$C300 … $D6FF	&	Read/Write	SRAM 9 Kbytes, data memory
$E700 … $E7FF	&	Write	Set Interrupt Flip-flop
$E800 … $FFFF			NEURON Chip internal

Table 4-3 NEURON Chip address map of NSI version

Note: Setting of the interrupt flip-flop is data-independent.

4.2 Device Status

Custom applications which work with the Easylon PCI Bus Interface have to properly control the device status. Below we show a code fragment illustrating this. The structures used are taken from the so-called HOSTAPPLICATION by the Echelon Company and adapted to this example. The general basis HOST APPLICATION is available from Echelon's homepage (www.echelon.com) .

```
#pragma pack(1)
#define NM_update_domain  0x63
#define NM_set_node_mode  0x6C
#define SVC_request       0x60
#define niRESPONSE    0x16
#define niLOCAL           0x22
#define niRESET           0x50
#define LDV_OK        0

typedef struct {
    BYTE cmq;         // cmd[7..4]                    queue[3..0]
    BYTE len;
    BYTE svc_tag;     // 0[7] Service[6..5] auth[4]   tag[3..0]
    BYTE flags;       // prio path cplcode[5..4] expl altp pool resp
    BYTE data_len;
    BYTE format;      // rcv: domain[7] flex[6]
    union {
        struct {
            BYTE dom_node;      // domain[7] node/memb[6..0]
            BYTE rpt_retry;     // rpt_timer[7..4]      retry[3..0]
            BYTE tx_timer;      //                      tx_timer[3..0]
            BYTE dnet_grp;      // destination subnet or group
            BYTE nid[6];        // NEURON ID
        } send;
            struct {
            BYTE snet;          // source subnet
            BYTE snode;         // source node
            BYTE dnet_grp;      // destination subnet or group
            BYTE dnode_nid[7];  // destination node or NEURON ID
        } rcv;
        struct {
            BYTE snet;          // source subnet
            BYTE snode;         // source node
            BYTE dnet;          // destination subnet
            BYTE dnode;         // destination node
            BYTE group;
            BYTE member;
            BYTE reserved[4];
        } resp;
    } adr;
    BYTE code;                   // message code or selector MSB
    BYTE data[239];
} ExpAppBuf;
```

```
ExpAppBuf msg_out;    // Explicit message buffer for outgoing messages
ExpAppBuf msg_in;     // Explicit message buffer for incoming messages
ExpAppBuf msg_rsp;    // Explicit message buffer for response messages
int ni_handle;
BYTE my_domain[15] =
    {0,0,0,0,0,0, 0x01, 0xC0, 0, 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};

int send_local( int len ) {
    int ldv_err;
    msg_out.cmq = niLOCAL;
    msg_out.svc_tag = SVC_request;
    msg_out.flags = 8;
    msg_out.len = len + 15;
    msg_out.data_len = len + 1;
    if( ldv_write( &msg_out, len + 17 ) ) return(0);
    while( 1 ) {
        ldv_err = ldv_read( &msg_in, 256 );
        if( ldv_err == LDV_OK ) {
            if(msg_in.cmq == niRESET) return(0);        // Local reset
            if(msg_in.cmq == niRESPONSE) {
            memcpy(&msg_rsp, &msg_in, msg_in.len + 2);
            return(1);                                  // Ok
            }
        }
    }
    return(0);
}

int set_config_online() {
    msg_out.code = NM_update_domain;
    msg_out.data[0] = 0;                          // Domain index 0
    memcpy( &msg_out.data[1], &my_domain, 15 ); // Subnet 1, Node 64
    if( !send_local(16)) return(0);

    msg_out.code = NM_set_node_mode;
    msg_out.data[0] = 3;                          // Change state
    msg_out.data[1] = 4;                          // Configured online
    if( !send_local(2)) return(0);
    return(1);                                    // Success
}
```

4.3 PCI Bus Interface

The PCI bus Interface has been implemented as 32-bit I/O Interface according to the specification " PCI Local Bus Specification, Version 2.2".
This card is occupying an address range of three I/O addresses. The PC uses one 8-bit control and an 8-bit status register on the PCI bus interface for control and status check of the NEURON.

4.3.1 I/O Address Map

PLX chipselect	Device
CS 0 & Write	Write control byte
CS 0 & Read	Read status byte
CS 1 & Read	Read data, NEURON Chip
CS 1 & Write	Write data, NEURON Chip
CS 3 & Read/Write	Data port watcher[3]

Table 4-4 I/O-Address map, PCL bus

4.3.1.1 Signal Assignments Control Byte

Data Bit	Signal	Description
D4	IL0	Clear interrupt flip-flop from NEURON Chip
D3	NSERV	NEURON Chip service pin, high active
D2	/NCF	NEURON Chip clear reset flip-flop, low active
D1	WTCRES	Watcher reset, high active
D0	NRES	NEURON Chip reset, high active

Table 4-5 Signal assignments control byte

The control byte of the interface card is reset on every power-on; the reset signal to the NEURON Chip and the service pin are not activated.

[3] The Easylon Watcher cannot be used with this interface card version. Information is only provided for the sake of completeness.

4.3.1.2 Signal Assignments Status Byte

Data bit	Signal	Description
D3	/NINT	Status of NEURON Chip interrupt flip-flop, low active
D2	/NRF	Status of NEURON Chip reset flip-flop, low active
D1	/WTCHS	Watcher[4] handshake, low active
D0	/NHS	NEURON Chip handshake, low active

Table 4-6 Signal assignments status byte

4.3.2 Example

```
/*
  Code fragment for ascertaining  the I/O adresses and interrupt
  numbers of the Easylon PCI Interface assigned by the PCI-BIOS
  Prerequisites:
     PCI-Bus-Number = 0
     Configuration mechanism = 1
*/

// Vendor ID = 0x1555 (Gesytec GmbH)
// Device ID = 0x0002 (Easylon PCI Interface)

#define GESYTEC_LPP     0x00021555

#define BYTE    unsigned char
#define WORD    unsigned short
#define DWORD   unsigned long

// 32 bit I/O access functions, must not be splitted into two 16 bit
accesses !
extern void _outpd(WORD adr, DWORD data);
extern DWORD _inpd(WORD adr);

// Globals
WORD cs0;       // I/O-Adresse of Control/Status-Byte
WORD cs1;       // I/O-Adresse of Neuron-Data port
WORD cs3;       // I/O-Adresse of Watcher-Data port
WORD inta;      // IRQ-Nummer
WORD plxirq;    // I/O-Adresse of PLX9050-Interrupt-Registers
DWORD type1adr;

void setreg(BYTE reg, DWORD value) {
                _outpd(0xCF8, type1adr | reg);
                _outpd(0xCFC, value);
```

[4] The Easylon Watcher cannot be used with this interface card version. Information is only provided for the sake of completeness.

```
}

DWORD getreg(BYTE reg) {
                _outpd(0xCF8, type1adr | reg);
                return( _inpd(0xCFC) );
}

int pci_getcfg() {
                int slot;
                for(slot=0; slot<32; slot++) {
                        type1adr = (slot << 11) | 0x80000000;
                        _outpd(0xCF8, type1adr);
                        if(_inpd(0xCFC) == GESYTEC_LPP) {
                          cs0    = (WORD)(getreg(0x18) & 0xFFF0);
                          cs1    = (WORD)(getreg(0x1C) & 0xFFF0);
                          cs3    = (WORD)(getreg(0x24) & 0xFFF0);
                          inta   = (WORD)(getreg(0x3C) & 0x0F);
                          plxirq = (WORD)((getreg(0x14) & 0xFFF0) +
0x4C);
                          return(1);  // Success
                        }
                }
                return(0);  // Interface card not found
}
-------------------------------------------------------------------
Switching on the Neuron-Interrupt in PLX-Chip:
     _outpw(plxirq, 0x41);  // Enable IRQ on PLX
Switching off the Neuron-Interrupt in PLX-Chip:
     _outpw(plxirq, 0x00);  // Disable IRQ on PLX
```

4.3.3 Reset Procedure

The NEURON Chip starts up automatically when the power of the PC is switched on.

During operation, a hardware reset of the NEURON Chip can be initiated via a control bit (NRES).

The NEURON Chip is able to initiate a reset independently during operation. An additional reset flip flop is implemented on the NEURON Chip, to enable the PC to identify such a reset. The status of this flip-flop (/NRF) can be checked via the status byte of the interface card. The flip-flop is reset and deactivated via the control bit (/NCF). When /NCF is held ‘low’, the reset flip-flop is deactivated (/NCF = ‘0’).

5 List of Figures

6 List of Tables

7 Index