



## Handbuch

Gesytec GmbH  
Pascalstr. 6  
D-52076 Aachen

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

Dok. ID: LPP/UserDoc/LPP3/LPP\_Manual-D-V3.5.docx,  
Version v3.5 vom 24.6.14

## Dieses Handbuch ...

... gibt Ihnen alle nötigen Informationen, um die Easylon® PCI-Bus Interfacekarte effizient zu nutzen.

Dieses Handbuch behandelt ausschließlich die Handhabung der Interfacekarte. Es wird weder auf die Echelon® LONWORKS Technologie (LON) eingegangen, noch werden das Echelon Microprocessor Interface Program (MIP) oder Echelons Network Service Interface (NSI) erklärt, die auf der Karte als Firmware eingesetzt werden. Die Treiber der Interfacekarte wurden gemäß der Spezifikation der Firma Echelon entwickelt. Auch sie werden hier nicht im Detail behandelt. Ausführliche Informationen zur LON Technologie finden Sie in den Dokumentationen der Firma Echelon. Insbesondere hilfreich für die Entwicklung von Applikationen, die eine Easylon Interfacekarte verwenden, dürfte Echelons „LONWORKS Host Application Programmer's Guide“ für Sie sein.

Nach einer kurzen Vorstellung des Easylon PCI-Bus Interface in Kapitel 1 beschreibt Kapitel 2 die nötigen Schritte zur Installation der Karte.

Kapitel 3 gibt eine ausführliche technische Beschreibung.

Die „Programmierhinweise“ in Kapitel 4 geben Ihnen die nötigen Information zum Erstellen eines eigenen Netzwerktreibers für die Easylon Interfacekarte.

Diese Dokumentation kann jederzeit ohne Ankündigung geändert werden. Gesytec übernimmt keinerlei Verantwortung für Fehler oder Ungenauigkeiten in dieser Dokumentation und etwaige sich daraus ergebende Folgen.

Gesytec sowie deren Repräsentanten und Mitarbeiter haften in keinem Fall für etwaige Defekte, indirekt verursachte oder aus dem Gebrauch folgenden Schäden, die aufgrund der Verwendung oder der Nichtanwendbarkeit der Software oder der begleitenden Dokumentation entstehen.

Easylon ist ein registriertes Warenzeichen der Gesytec GmbH.

Echelon, LON, LONWORKS und NEURON sind registrierte Warenzeichen der Echelon Corporation.

Windows und Windows CE sind eingetragene Warenzeichen der Firma Microsoft. Andere Namen können eingetragene Warenzeichen der entsprechenden Firmen sein.

Die Easylon PCI-Bus Interfacekarten werden entweder mit dem MIP/P50 oder mit dem NSI Programm der Firma Echelon betrieben. Die Rechte an dieser Software liegen bei der Echelon Corporation.

# Inhalt

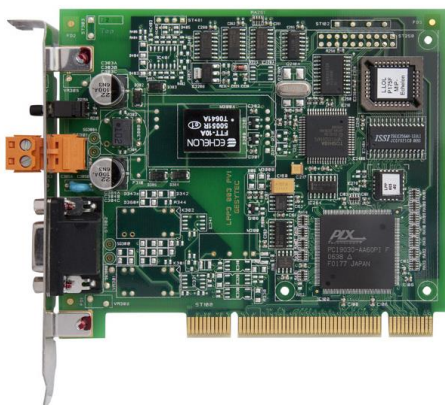
<b>1</b>	<b>Produktinformation</b>	<b>5</b>
1.1	Varianten	5
1.2	Lieferumfang	6
1.3	Überblick	6
<b>2</b>	<b>Installation</b>	<b>8</b>
2.1	Einbau der Karte	8
2.2	Treiber-Installation	8
2.2.1	Treiber für Windows Betriebssysteme (WDM-Treiber)	9
2.2.1.1	Installation mit Windows Assistent	9
2.2.1.2	Manuelle Installation und Update	9
2.2.1.3	Parametrierung	10
2.2.1.4	Deinstallation	12
2.2.2	EasyCheck – Test Utility für die Windows Treiber	12
2.2.3	Windows und 16 Bit Applikationen	13
2.2.4	DOS Treiber	13
2.2.4.1	Installation	14
2.2.4.2	Anzeigen der installierten Treiber	15
<b>3</b>	<b>Technische Beschreibung</b>	<b>16</b>
3.1	Netzwerk Interface	16
3.2	Bus Interface	16
3.3	Reset, Systemsteuerung	16
3.4	Blockdiagramm	17
3.5	Steckerbelegung	18
3.6	Service LED	19
3.7	Technische Daten	19
<b>4</b>	<b>Programmierhinweise</b>	<b>21</b>
4.1	LON Netzwerk Knoten	21
4.1.1	Netzwerk-Interface	21
4.1.2	CPU	21
4.1.2.1	Koppelung NEURON Chip ↔ Bus Interface	22
4.1.2.2	Interrupt Funktion NEURON Chip → PCI-Bus	22
4.1.2.3	NEURON Chip Adressbelegung	22
4.2	Baugruppenstatus	23
4.3	PCI-Bus Interface	25
4.3.1	I/O- Adressbelegung	25
4.3.1.1	Signalaufbau Steuerbyte	25
4.3.1.2	Signalaufbau Statusbyte	26

4.3.2	Beispiel .....	26
4.3.3	Reset-Verhalten.....	27
<b>5</b>	<b>Liste der Abbildungen .....</b>	<b>28</b>
<b>6</b>	<b>Liste der Tabellen .....</b>	<b>28</b>
<b>7</b>	<b>Index.....</b>	<b>29</b>

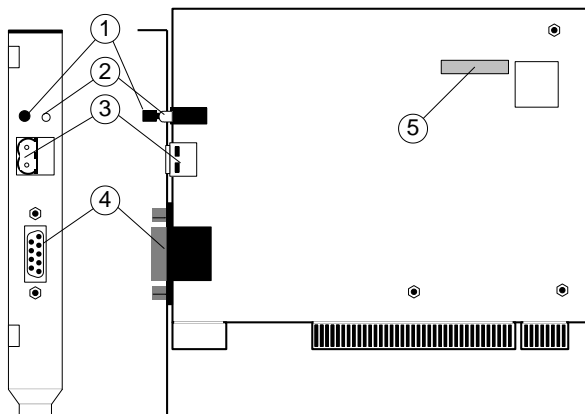
# 1

# Produktinformation

Dieses Handbuch beschreibt die Easylon Interfacekarte:



Easylon PCI-Bus Interface,  
Einsteckkarte für 32-Bit PCI-Bus  
Slots.



**Bild 1-1 Easylon PCI-Bus In-  
terface**

- (1) Service Taster
- (2) Service LED
- (3) Steck-Schraubklemme  
(nicht bei RS485 Varianten)
- (4) 9poliger Sub-D Stecker
- (5) Typenschild auf Rückseite  
(vgl. Tabelle 1.1)

## 1.1 Varianten

Folgende Varianten der Easylon PCI-Bus Interfacekarte sind in dieser Dokumentation beschrieben. Die einzelnen Varianten der Interfacekarte sind durch ihr Typenschild identifizierbar.

Typ Kennung	Bestellcode	Transceiver	Variante
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

**Tabelle 1.1 Varianten, Bestellnummern und Typenbezeichnung der Easylon PCI-Bus Interfacekarte**

## 1.2 Lieferumfang

- PC-Einsteckkarte mit Echelons MIP/P50 oder NSI
- Montageanleitung
- Installations- und Dokumentations-CD mit
  - Netzwerktreiber für 32- und 64-Bit Versionen von Windows XP, Vista, 7, 8, Server 2003, Server 2008, Server 2008 R2, Windows CE, Linux und MS-DOS
  - WLDV32.DLL für Windows
  - Easylon RNI Software für Fernzugriff auf LON
  - EasyCheck Diagnosesoftware für Easylon Interfaces
  - Dokumentation

## 1.3 Überblick

Die Easylon PCI-Bus Interfacekarte ist eine Schnittstellenkarte zur Verbindung zwischen einem PCI-Bus PC und dem LON Netzwerk. Es gibt dabei Varianten für Anschluss über Transformer Coupled Twisted Pair (TPT/XF), Free Topology (FTT), Direct Connect oder RS 485.

Das Easylon PCI-Bus Interface verfügt über Service Pin und Service LED. Bei allen Varianten außer RS485 ist die LON-Schnittstelle parallel auf zwei Anschlussbuchsen gelegt, die alternativ benutzt werden können:

- 9polige Sub-D Buchse
- 2polige Steck- Schraubklemme

In OEM Versionen kann optional die Sub-D Buchse durch eine RJ45 Stecker ersetzt werden. Optional können auch Phoenix an Stelle der üblichen Weidmüller Klemmen eingesetzt werden.

Als moderne Alternative zu den in dieser Dokumentation beschriebenen Standard LonTalk Adaptern stehen bei höheren Anforderungen an die Leistung Easyton Interfaces<sup>+</sup> gemäß DIN EN 14908 zur Verfügung. Informieren Sie sich unter [www.gesyttec.de](http://www.gesyttec.de).

# 2

## Installation

Die Installation der Easylon PCI-Bus Interfacekarte erfolgt in zwei Schritten:

1. Einbau der Karte in den Rechner
2. Installation des Netzwerktreibers

Bei Verwendung der Easylon PCI-Bus Interfacekarte in eigenen Applikationen muss diese Einstellung im Programm berücksichtigt werden. Im Kapitel 4.2 finden sich dazu nähere Hinweise.

Die zur Interfacekarte gehörigen „eXternal Interface Files“ (.xif) finden Sie auf der mitgelieferten CD-ROM. Die Zuordnung von „.xif“-Datei und Kartentyp entnehmen Sie bitte der Tabelle 4.1.

### 2.1 Einbau der Karte

Bevor Sie die Easylon PCI-Bus Interfacekarte einbauen, lesen Sie bitte die beiliegende Montageanleitung. Beachten Sie bitte auch alle Vorschriften des Rechnerherstellers über den Einbau zusätzlicher Schnittstellenkarten.

Die Easylon PCI-Bus Interfacekarte wird bei ausgeschaltetem Rechner in einen freien 32-Bit PCI-Slot gesteckt.

Die Busadresse und der Interrupt werden vom PCI-BIOS des Rechners automatisch vergeben.

Zum Anschließen des LON Netzwerkes beachten Sie bitte die Pinbelegung der Stecker, die in Kapitel 3.5 beschrieben ist.

### 2.2 Treiber-Installation

Für diese Easylon Interfacekarten stehen Treiber unter verschiedenen Betriebssystemen zur Verfügung. Aktuell sind dies Windows XP, Vista und 7 sowie die Windows Server Betriebssysteme 2003, 2008 und 2008 R2. Die Treiber der Easylon Interfaces sind für die 32-Bit und 64-Bit Versionen der genannten Betriebssysteme verfügbar. Zudem gibt es Treiber für Windows CE, Linux und MS-Dos. Aktualisierte Versionen der Treiber finden Sie im Internet auf den Easylon Support Seiten der Gesytec: [www.gesytec.de](http://www.gesytec.de).

Die Installation ist in den folgenden Abschnitten beschrieben:



Windows Betriebssysteme                      Kapitel 2.2.1

16-Bit Treiber unter 32-Bit Windows        Kapitel 2.2.3

DOS Treiber                                      Kapitel 2.2.4

In diesem Kapitel finden Sie auch Erläuterungen zur Testprogramm „Easy-Check“, sowie zum Einbinden des 32-Bit Treibers zu einer 16-Bit Applikation.

Ein Linux Treiber ist im Quellcode mit Compilier-Anleitung auf der CD im Verzeichnis „Linux“ verfügbar. Weitere Angaben finden sich dort.

Die Benutzerführung der Treiber CD bietet verschiedenen Möglichkeiten der Auswahl von Treiber und Dokumentation. Der Start des Setups für das jeweilige Betriebssystem und die vorliegende Interfacekarte ist jedoch ebenso direkt von der CD aus möglich.

## **2.2.1        Treiber für Windows Betriebssysteme (WDM-Treiber)**

Dieser Abschnitt beschreibt die Installation und das Setup des Treibers für die Easyton Interface Karten unter den genannten Windows Betriebssystemen ab Windows XP.

Das Setup Programm benutzt für alle Betriebssysteme den gleichen WDM-Treiber (Windows Driver Model).

**Hinweise**        Prinzipiell bestehen für die Installation die Möglichkeiten den Windows Assistenten zu benutzen oder eine manuelle Installation mit dem Programm FastUpd.exe vorzunehmen. Letztere Möglichkeit führt mit wenigen Klicks zum Ziel. Dies ist insbesondere dann sinnvoll, wenn mehrere Treibeinstanzen installiert werden sollen.

Zum Schluss wird die De-Installation des Treibers erklärt.

### **2.2.1.1        Installation mit Windows Assistent**

Nachdem Sie die Karte eingebaut haben, findet Windows beim nächsten Booten automatisch die Karte und startet den Hardwareassistenten.

Sofern Windows den Treiber (auf der „Drivers & Documentation“ CD) nicht automatisch findet oder der Treiber auf einem anderen Laufwerk liegt, wählen Sie dort bitte aus dem Verzeichnis „LPCLPP“ als Setup-Datei „LppWdm.inf“ und als Treiber „Gesytec LPP WDM Driver PCI“ aus.

### **2.2.1.2        Manuelle Installation und Update**

Am einfachsten installieren Sie den Treiber, indem Sie den Hardwareassistenten ignorieren. Starten Sie stattdessen direkt

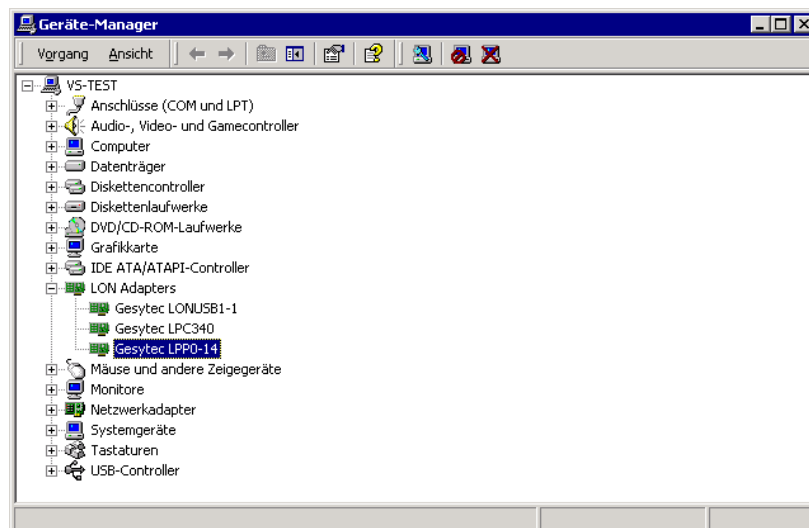
### FastUpd.exe oder FastUpd64.exe<sup>1</sup>

aus dem Verzeichnis „LPCLPP“ auf der CD-ROM.

Dasselbe Programm verwenden Sie, um einen bereits vorhandenen Treiber zu aktualisieren.

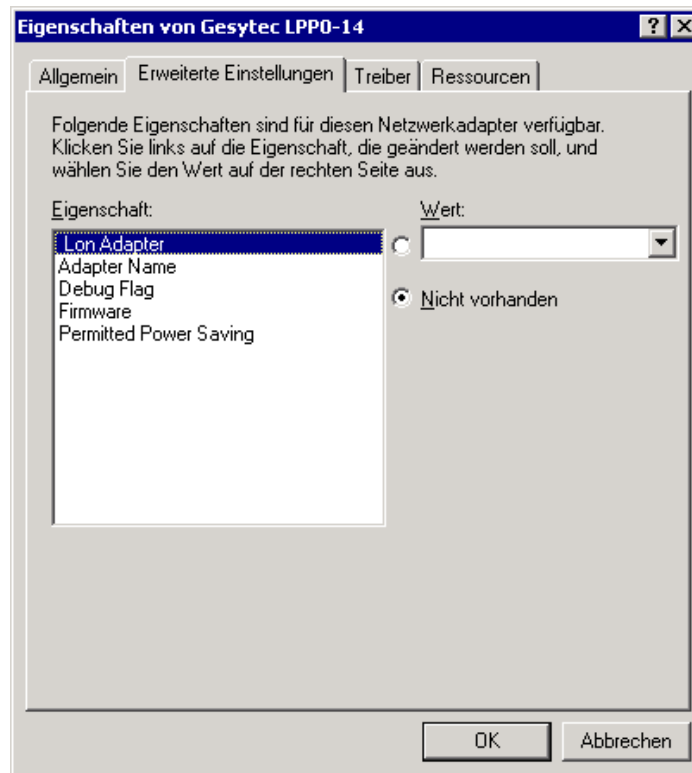
#### 2.2.1.3 Parametrierung

Unter bestimmten Betriebsbedingungen kann es sinnvoll sein, eine Parametrierung der Easylon Interfacekarte vorzunehmen. Dazu wählen Sie im Geräte-Manager unter „LON Adapters“ die entsprechende Easylon Interfacekarte aus und betrachten deren Eigenschaften:



Unter „Erweiterte Einstellungen“ finden Sie dann Möglichkeiten zur Parametrierung der Interfacekarte:

<sup>1</sup> Bei 64-Bit Systemen



### Lon Adapter

Hier können Sie der Interfacekarte einen Namen von „LON1“ ... „LON9“ zuweisen, wie ihn bestimmte Applikationen benötigen.

**ACHTUNG** Der Name darf noch nicht bereits von anderen Treibern belegt sein. Ist der Name schon belegt, so kann das Gerät nicht gestartet werden (Code 10).

### Adapter Name

Alternativ kann auch ein beliebiger, frei wählbarer Adaptername (z.B. „Haus 7“) vergeben werden.

**Anmerkung** Werden sowohl „Lon Adapter“ als auch ein „Adapter Name“ vergeben, so wird nur der Eintrag bei „Lon Adapter“ verwendet.

### Debug Flag

Dieser Wert umfasst ein DWORD in hexadezimaler Notation von verschiedenen Flags zu Debug Zwecken. Es steht normalerweise auf 0 (= „nicht vorhanden“). Durch Setzen der einzelnen Bits können bestimmte Debug Features eingeschaltet werden. Derzeit sind die Bits 0, 1 und 2 benutzt.

**Bit 0:** Es werden LON-Telegramme an der Schnittstelle von und zur Applikation über Debug Output angezeigt.

- Bit 1: Es werden LON-Telegramme an der Schnittstelle vom und zum Neuron-Chip über Debug Output angezeigt.
- Bit 2: Es werden Telegramme an der Schnittstelle vom und zum Watcher<sup>2</sup> über Debug Output angezeigt.
- Bit 3: Es wird Öffnen (CREATE) und Schließen (CLOSE) des Treibers über Debug Output angezeigt.

Hinweis: Der Debug Output kann z.B. mit dem Programm DebugView angezeigt werden, welches unter [www.sysinternals.com](http://www.sysinternals.com) frei erhältlich ist.

### **Firmware**

Es werden die Optionen MIP/NSI bzw. EEBLANK angezeigt. Permitted Power Saving

Normalerweise erlaubt die Easylyon Interfacekarte bei laufenden Applikationen sowohl den Standby-Modus (Standby) als auch den Ruhezustand (Hibernate). Unter bestimmten Bedingungen kann es jedoch unerwünscht sein, wenn der Rechner bei laufenden LON-Applikationen in den Ruhezustand oder Standby-Modus geht. In diesen Fällen können Ruhezustand (Auswahl: Standby) oder Ruhezustand und Standby-Modus (Auswahl: None) unterbunden werden.

#### **2.2.1.4 Deinstallation**

Die WDM-Treiber können über den „Geräte-Manager“ deinstalliert werden. Wählen Sie unter „LON Adapters“ den Treiber „Gesytex LPPx-yy“ aus und klicken Sie mit der rechten Maustaste auf „Deinstallieren“.

#### **2.2.2 EasyCheck – Test Utility für die Windows Treiber**

Zusätzlich zum Treiber kann das Testprogramm „EasyCheck“ in das Installationsverzeichnis (Standard: \Easylyon\Lpx) installiert werden. Diese Utility prüft Interface und Softwareumgebung und erlaubt Rückschlüsse auf die möglichen Ursachen bei Problemen an denen das Interface beteiligt sein kann.

Das Programm „EasyCheck“ führt eine Analyse der Software auf dem System durch. Es öffnet das ausgewählte Interface, überprüft die Version des Treibers und zeigt diese am Bildschirm an. Durch das Senden eines „query status“-Befehls wird die Kommunikation zur Hardware getestet. Außerdem wird durch ein „read memory“ festgestellt, ob die Instanz MIP- oder VNI verwendet. Korrekt installierte Easylyon Interfacekarten schicken eine entsprechende Antwort.

<sup>2</sup> Der Easylyon Watcher ist mit dieser Version der Interfacekarte nicht mehr einsetzbar. Die Information ist nur aus Gründen der Vollständigkeit aufgeführt.

### 2.2.3 Windows und 16 Bit Applikationen

Der Windows Treiber für die 32-Bit Versionen der unterstützten Betriebssysteme unterstützt auch ein 16-Bit Interface. (Leider unterstützt Microsoft dies nicht für die 64-Bit Versionen). Um das 16-Bit Interface zu benutzen, muss die nachstehende Zeile in die Datei „config.nt“ im Verzeichnis „windows\system32“ (das Verzeichnis kann auf Ihrem Rechner auch anders heißen) eingetragen werden:

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

Die nähere Spezifikationen des benutzten 32 Bit LON-Device erfolgt dabei über den optionalen -L bzw. /L Parameter:

/Lname

name =

lpp0-14

für Device LPP mit PCI-Bus-Nummer 0 und PCI Device-Nummer 14

**Anmerkung:** Bei der Eingabe beachten Sie bitte die 2 aufeinander folgenden „l“ vom Parameter -L bzw. /L und dem unmittelbar folgend Namen lxxxx.

Eine nähere Spezifikationen des zur Verfügung gestellten 16 Bit LON-Device erfolgt über folgenden optionalen Parameter:

/Dn

mit n = 1...9 für LON1 bis LON9

Machen Sie keine explizite Angabe, erfolgt die Zuordnung auf den ersten freien Namen, beginnend bei „LON1“.

### 2.2.4 DOS Treiber

Der Netzwerktreiber für den Betrieb der Easylon Interfacekarten unter MS-DOS wurden gemäß Spezifikation der Firma Echelon entwickelt. Ausführliche Informationen zum Erstellen von Applikationen, z.B. eines eigenen Treibers, finden Sie im „LONWORKS Host Application Programmer's Guide“ von Echelon.

Der Treiber ist auf der mitgelieferten Installations-CD-ROM im Verzeichnis „DOS“ zu finden.

„lppdrv.exe“

Die Treiberdatei „lppdrv.exe“ muss von der CD-ROM auf Ihren Computer kopiert werden, z.B. in ein Verzeichnis C:\easylon.

Der Netzwerktreiber für die Easylon Interfacekarten benötigt 1,6 KByte residenten Programmcode, 2 KByte Outputbuffer und 2, 4 oder 8 KByte Inputbuffer.

### 2.2.4.1 Installation

Der Netzwerktreiber wird in Ihrem Computer als Device auf den ersten freien Namen, beginnend bei „LON1:“ installiert, indem Sie ihn in der "Config.sys" Datei mit folgender Zeile eintragen

```
device={pfad}\lppdrv.exe /D
```

oder

```
devicehigh={pfad}\lppdrv.exe /D
```

Unter {pfad} ist das Verzeichnis anzugeben, in das die Treiberdatei kopiert wurde.

#### Optionen

Weiter stehen folgende Optionen zur Verfügung:

##### /D Setzt die Device Nummer

Für die Interfacekarte können Device Nummern zwischen 1 und 9 (LON1: – LON9:) verwendet werden. Benutzt ein bereits installierter Treiber die angegebene Device Nummer, so erfolgt die Fehlermeldung:

```
Invalid or duplicate device name
```

Fehlt hinter der Optionskennung /D die Angabe eines Wertes oder wird die Option /D nicht angegeben, so wird automatisch eine gültige Nummer eingesetzt. Sind alle möglichen Device Nummern vergeben, so erfolgt die Meldung:

```
LON1: ... LON9: already defined
```

##### /I Erhöhen des Eingangspuffers

Ein- und Ausgangspuffer der Easyton Interfacekarte sind als „byte-level FIFOs“ ausgelegt, d.h. der benötigte Speicherplatz hängt von der Nachrichtenlänge ab. Hieraus folgt, dass eine Speichergröße von 2 KByte (Default Wert, entsprechend ca. 50–100 Meldungen) normalerweise ausreichen sollte. Müssen mehr Informationen zwischengespeichert werden, kann die Größe des Eingangspuffers erhöht werden. Zulässige Werte für den Parameter /I sind 2, 4 und 8 (KByte).

Anstatt des Schrägstriches „/“ kann auch das Minuszeichen „-“ zur Kennzeichnung der Optionen verwendet werden. Die Schreibweise der Optionen ist unabhängig von Groß-/Kleinschreibung.

#### Die Installation mehrerer Easyton PCI-Bus Interfacekarten

Der Netzwerktreiber unterstützt jeweils nur eine Interfacekarte. Möchten Sie mehrere Easyton PCI-Bus Interfacekarten betreiben, so müssen Sie für jede Interfacekarte einen Eintrag in der „config.sys“ Datei vornehmen. Die Karten werden

in der Reihenfolge der PCI Slots angemeldet. Wird beim Laden festgestellt, dass bereits ein Easyton Netzwerktreiber installiert ist, so wird die Copyright Meldung nicht mehr angezeigt.

#### **2.2.4.2 Anzeigen der installierten Treiber**

Mit der Datei „lppdrv.exe“ können die im System installierten Netzwerktreiber angezeigt werden. Sie wird wie jedes DOS-Programm gestartet. Es werden die Namen und die benötigte Speichergröße eines jeden Treibers angezeigt.

**/R** Die Option /R ermöglicht ein nachträgliches Ändern einer Device Nummer.

Beispiel:        lppdrv -r13  
                  ändert die Device Nummer von LON1: in LON3:

Ist die erste Device Nummer nicht vergeben, oder ist die zweite Nummer bereits von einem anderen Treiber in Gebrauch, so erfolgt die Meldung:

Invalid or duplicate device name.

Das Umbenennen der Device Nummer ist nicht auf die von Easyton Interface Netzwerktreiber definierten Schnittstellengeräte beschränkt. Es kann auch aus einer Windows DOS-Box heraus als globale Funktion auf das gesamte System, inklusive 16-Bit Windows-Applikationen, angewendet werden.

# 3 Technische Beschreibung

## 3.1 Netzwerk Interface

Die Easylon Interfacekarten basieren auf dem NEURON 3150® Chip. Dieser arbeitet unter MIP/P50 mit bis zu 32 KByte ROM als Programmspeicher und 24 KByte SRAM als Datenspeicher. Unter NSI stehen 48,75 KByte ROM und 9 KByte SRAM zur Verfügung. Der NEURON Chip ist im Slave\_A Mode an den PCI-Bus gebunden.

Zur Überwachung des NEURON Chips ist ein Reset Flipflop auf der Karte implementiert. Ein NEURON-Reset kann im PC identifiziert werden, indem Sie das Status-Byte auslesen. Gleiches gilt auch für das Interrupt Flipflop.

Zur Anzeige des Knotenstatus ist eine Service LED und zum Aktivieren der Servicefunktion des NEURON Chips der Service Pin verfügbar (Bild 1.1).

## 3.2 Bus Interface

Das PCI-Bus Interface wurde als 32-Bit I/O Interface entsprechend der Spezifikation "PCI Local Bus Specification, Version 2.2" entwickelt. Sie ist kompatibel zu PCI Slots mit 3,3 Volt Signalspannung und kann in PCI-X Slots verwendet werden. Weiter kann es notwendig sein, im BIOS eines PCI-X PCs die Baugruppe freizugeben. Hierzu sollte das Handbuch des PCs geprüft werden.

Anmerkung: Der Unterschied zwischen PCI-X und PCI express ist zu beachten, PCI-X verwendet eine parallele Kommunikation, PCI express hingegen eine serielle.

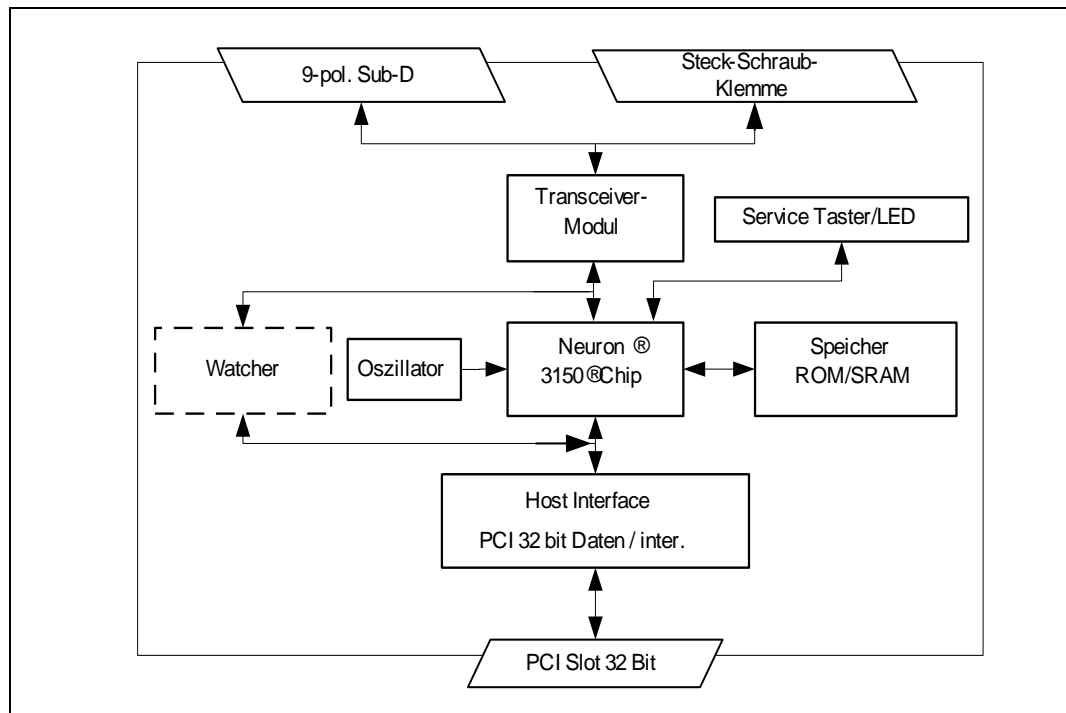
Der Easylon PCI-Bus Interfacekarte werden die I/O-Adressen über das PCI BIOS zugewiesen. Die benötigten Adressen unterteilen sich in eine Adresse für den Datentransfer zwischen dem PC und der Netzwerk-Interfacekarte, eine Adresse für Statusabfrage und Steuerung des NEURON Chip.

## 3.3 Reset, Systemsteuerung

Ein Reset des NEURON Chip auf der Interfacekarte kann vom PC programmgesteuert initiiert werden. Nach einem System-Reset startet der NEURON Chip automatisch.



### 3.4 Blockdiagramm



**Bild 3-1 Blockdiagramm Easylon PCI-Bus Interfacekarte**

## 3.5 Steckerbelegung

Die Easylon PCI-Bus Interfacekarte verfügt über einen 9poligen Sub-D Stecker (Bild 1-1 Ä) für die Verbindung mit dem LON Netzwerk. Die TPT/XF und FTT Variante verfügt zusätzlich über eine 2polige Steck-Schraubklemme (Bild 1-1 ☺), die alternativ zum 9poligen Sub-D Stecker benutzt werden kann.

Steckertyp	Pin	Signal	Bemerkung
9polig Sub-D	1	Data	Datenleitung
	2	Data	Datenleitung
	3	-	Reserviert
	4	-	Reserviert
	5	-	Reserviert
	6	-	Reserviert
	7	-	Reserviert
	8	-	Reserviert
	9	-	Reserviert
2polige Steck-Schraubklemme	1	Data	Datenleitung
	2	Data	Datenleitung

**Tabelle 3.1 Steckerbelegung für die TP/XF und FTT Variante der Easylon PCI-Bus Interfacekarte**

Steckertyp	Pin	Signal	Bemerkung
9polig Sub-D	1	-	Reserviert
	2	-	Reserviert
	3	DA-	Datenleitung
	4	SA-	RTS -
	5	-	Reserviert
	6	+5 V	U+ Versorgungsspannung, galvanisch getrennt
	7	0 V	U- Versorgungsspannung, galvanisch getrennt
	8	DA+	Datenleitung
	9	SA+	RTS +

**Tabelle 3.2 Steckerbelegung für die RS-485 Variante der Easylon PCI-Bus Interfacekarte**

## 3.6 Service LED

Die Service LED (Bild 1-1) signalisiert den Knotenstatus der Easylon Interfacekarte. Es sind folgende der Service LED Signale definiert:

Service LED	Status	Bemerkung
Blitz (1Hz)	Kein Treiber geladen, oder Treiberkonflikt	Prüfen Sie die Treibereinstellungen. IRQ-, Adressenkonflikt mit anderen Karten?
Blinkt (1/2Hz)	Treiber ist installiert, Knoten ist „unconfigured“.	Knoten auf „configured“ setzen
Konstant AN	Karte ist „applicationless“ und „unconfigured“.	
Konstant AUS	Installation ok	normaler Betriebszustand

**Tabelle 3.3 Service LED Bedeutung**

## 3.7 Technische Daten

<b>Bus Interface</b>	32 Bit Daten (I/O) gemäß „PCI Local Bus Specification, Version 2.2“ kompatibel zu 3,3 Volt und PCI-X		
I/O Adressen	3		
Kontrollregister	8 Bit		
Statusregister	8 Bit		
Kompatibilität	LonTalk, EN 14908		
<b>CPU</b>	NEURON 3150, 10 MHz		
Kopplung	parallel, Slave_A Mode		
Speicher			
MIP/P50:	ROM	32 KByte	
	RAM	24 KByte	
NSI :	ROM	48.75 KByte	
	RAM	9 KByte	

**Netzwerkinterface**

<b>Bestellcode*</b>	<b>Netzwerk Interface</b>	<b>Übertra- gungsrate</b>	<b>Stecker</b>	<b>Überspannungs- schutz</b>
P.P10201 P.P10211 P.P11201 P.P11211	TP/XF-78	78 kbps	9polig Sub-D + 2polige Steck- Schraubklemme	Sparc-gaps
P.P10202 P.P10212 P.P11202 P.P11212	TP/XF- 1250	1.25 Mbps	9polig Sub-D + 2polige Steck- Schraubklemme	Sparc-gaps
P.P10203 P.P10213 P.P11203 P.P11213	RS485, galvan. Isol.	39 kbps	9polig Sub-D	Zehner Diode
P.P10204 P.P10214 P.P11204 P.P11214	FTT-10	78 kbps	9polig Sub-D + 2polige Steck- Schraubklemme	Sparc-gaps
P.P10205 P.P10215 P.P11205 P.P11215	Direct Connect	abhängig von Leitungslänge	9polig Sub-D + 2polige Steck- Schraubklemme	Sparc-gaps
P.P10206 P.P10216 P.P11206 P.P11216	FTX	78 kbps	9-pin D-Type + 2 pin screw- plug terminal	Sparc-gaps

\* zur Identifikation des Kartentyps siehe Tabelle 1.1.

**Spannungsversorgung;** 5 V, vom PC

**Leistungsaufnahme** typisch 2 W

**Temperatur**

Betrieb 0 °C - 50 °C

Lagerung -20 °C - +70 °C

**EMV**

EN 610 00-6-2

EN 550 22 A/B

**Feuchtigkeit**

nach DIN 40040, Klasse F

**Abmessungen**

130 mm x 105 mm, incl. Anschlüsse,  
für 32-Bit PCI Slot

# 4

## Programmierhinweise

### 4.1 LON Netzwerk Knoten

Die Easylon PCI-Bus Interfacekarte ist ein Netzwerkknoten im LON Netz. Sie arbeitet mit Echelons Microprocessor Interface Programm MIP/P50 oder mit NSI auf einem NEURON 3150 Chip als Kommunikations-CPU. Die zugehörigen „eXternal Interface Files“ (.xif) finden Sie auf der mitgelieferten CD-ROM. Die Zuordnung von „.xif“-Datei und Kartentyp entnehmen Sie bitte der folgenden Tabelle:

Netzwerk Interface	Übertragungsrate	XIF –Datei
TPT/XF	78 kB/s	lolp072f.xif
TPT/XF	1,25 MB/s	lolp073f.xif
RS485, galvanisch isol.	39 kB/s	lolp074f.xif
FTT	78 kB/s	lolp075f.xif

**Tabelle 4.1 Zuordnung der .xif Dateien**

#### 4.1.1 Netzwerk-Interface

Die verschiedenen Netzwerk-Interface-Varianten arbeiten direkt über den Kommunikationsport (CP0...CP4) des NEURON Chip.

Die Konfiguration erfolgt automatisch unter MIP/P50 Firmware. Interfaces mit NSI Firmware werden mit konfiguriertem Netzwerk Interface ausgeliefert. Sollten Probleme auftreten, können das Netzwerk Interface mit der „EasyCheck“ Utility neu eingestellt werden.

#### 4.1.2 CPU

Der Netzwerk-Interfaceknoten ist auf der Basis des NEURON 3150 Chip entwickelt. Die Taktrate beträgt 10 MHz. Der Prozessor ist bei Betrieb unter MIP/P50 mit 32 KByte ROM als Programmspeicher und mit 24 KByte SRAM als Datenspeicher ausgelegt. Unter NSI stehen 48,75 KByte ROM und 9 KByte SRAM zur Verfügung.

Alle 11 I/O Ports des NEURON Chip werden für die Koppelung an das PCI-Businterface verwendet.

Der Status des „Service Pin“ des NEURON Chip wird über die Service LED angezeigt; seine Servicefunktion kann über den Service Taster aktiviert werden. (s. Bild 1-1)

#### 4.1.2.1 Koppelung NEURON Chip ↔ Bus Interface

Der NEURON Chip ist im Slave\_A parallel Mode gekoppelt. Das Handshake Bit (NHS), das durch den NEURON Chip zur Kontrolle des Datenflusses definiert ist, kann mittels Statusbyte der Easylon PCI-Bus Interfacekarte überwacht werden. Weiterführende Informationen zum Datenkommunikations-Mechanismus im Slave\_A Mode finden Sie im NEURON 3150 Chip Datenbuch

#### 4.1.2.2 Interrupt Funktion NEURON Chip → PCI-Bus

Ein Interrupt Flipflop wird über einen Schreibzugriff auf eine vordefinierte Adresse gesetzt, die durch den Host über I/O Adress-Zugriff zurückgesetzt (acknowledged) werden kann.

Das Zurücklesen des Interruptstatus durch den NEURON Chip ist nicht möglich.

#### 4.1.2.3 NEURON Chip Adressbelegung

Adressenbereich	Module
\$0000 ... \$7FFF & Read	ROM 32 KByte, Programmspeicher
\$8000 ... \$DFFF & Read/Write	SRAM 24 KByte, Datenspeicher
\$E000 ... \$E7FF & Write	Setzen des Interrupt Flipflop
\$E800 ... \$FFFF	NEURON Chip intern

**Tabelle 4.2 NEURON Chip Adressbelegung für MIP/P50**

Adressenbereich	Module
\$0000 ... \$C2FF & Read	ROM 48,75 KByte, Programmspeicher
\$C300 ... \$D6FF & Read/Write	SRAM 9 KByte, Datenspeicher
\$E700 ... \$E7FF & Write	Setzen des Interrupt Flipflop
\$E800 ... \$FFFF	NEURON Chip intern

**Tabelle 4.3 NEURON Chip Adressbelegung für NSI**

Anmerkung: Das Setzen des Interrupt Flipflop erfolgt datenunabhängig.

## 4.2 Baugruppenstatus

Eigene Programme die das Easydon PCI-Bus Interface nutzen müssen den Zustand der Baugruppe korrekt verwalten. Als Beispiel dafür wird nachfolgend ein Codefragment gezeigt. Die darin verwendeten Strukturen sind der so genannten HOST APPLICATION der Firma Echelon entnommen und auf dieses Beispiel angepasst. Die allgemeine Grundlage HOST APPLICATION ist auf der Webseite der Firma Echelon ([www.echelon.com](http://www.echelon.com)) verfügbar.

```
#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

Das PCI-Bus Interface wurde als 32-Bit I/O Interface gemäß der Spezifikation „PCI Local Bus Specification, Version 2.2“ implementiert.

Diese Karte belegt einen Adressbereich von drei I/O Adressen. Der PC verwendet je ein 8-Bit Steuer- und Statusregister im PCI-Bus Interface für die Steuerung und Zustandsüberprüfung des NEURON.

### 4.3.1 I/O- Adressbelegung

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 <sup>3</sup>

**Tabelle 4.4 I/O-Adressbelegung, PCI Bus**

#### 4.3.1.1 Signalaufbau Steuerbyte

Datenbit	Signal	Beschreibung
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 <sup>3</sup> reset, high active
D0	NRES	NEURON Chip reset, high active

**Tabelle 4.5 Signalaufbau Steuerbyte**

Das Steuerbyte der Interfacekarte wird mit jedem Einschalten zurückgesetzt; das Resetsignal zum NEURON Chip und der Service Pin sind nicht aktiv.

<sup>3</sup> Der Easylyon Watcher ist mit dieser Version der Interfacekarte nicht mehr einsetzbar. Die Information ist nur aus Gründen der Vollständigkeit aufgeführt.

### 4.3.1.2 Signalaufbau Statusbyte

Datenbit	Signal	Beschreibung
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 <sup>4</sup> handshake, low active
D0	/NHS	NEURON Chip handshake, low active

**Tabelle 4.6 Signalaufbau Statusbyte**

### 4.3.2 Beispiel

```

/*
  Programm-Fragment zur Ermittlung der vom PCI-BIOS vergebenen
  I/O-Adressen und Interrupt-Nummer des Easylon PCI-Bus Interface
  Voraussetzungen:
    PCI-Bus-Nummer = 0
    Configuration mechanism = 1
*/

// Vendor ID = 0x1555 (Gesyttec GmbH)
// Device ID = 0x0002 (Easylon PCI-Bus Interface)

#define GESYTEC_LPP    0x00021555

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

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

// Globals
WORD cs0;    // I/O-Adresse von Control/Status-Byte
WORD cs1;    // I/O-Adresse vom Neuron-Datenport
WORD cs3;    // I/O-Adresse vom Watcher-Datenport
WORD inta;   // IRQ-Nummer
WORD plxirq; // I/O-Adresse des PLX9050-Interrupt-Registers
DWORD typeladr;

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

```

<sup>4</sup> Der Easylon Watcher ist mit dieser Version der Interfacekarte nicht mehr einsetzbar. Die Information ist nur aus Gründen der Vollständigkeit aufgeführt.

```

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

int pci_getcfg() {
    int slot;
    for(slot=0; slot<32; slot++) {
        typeladr = (slot << 11) | 0x80000000;
        _outpd(0xCF8, typeladr);
        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
}

```

```

-----
Einschalten des Neuron-Interrupt im PLX-Chip:
    _outpw(plxirq, 0x41);    // Enable IRQ on PLX
Ausschalten des Neuron-Interrupt im PLX-Chip:
    _outpw(plxirq, 0x00);    // Disable IRQ on PLX

```

### 4.3.3 Reset-Verhalten

Der NEURON Chip startet automatisch beim Einschalten des PCs.

Während des Betriebs kann ein Hardware-Reset des NEURON Chips mittels des Steuerbits (NRES) initiiert werden.

Der NEURON Chip kann einen Reset während des Betriebs durchführen. Ein zusätzliches Reset Flipflop ist auf der Easylon Interfacekarte implementiert, um dem PC einen NEURON-Reset mitzuteilen. Der Status des Flipflops (/NRF) kann mit dem Statusbyte der Easylon Interfacekarte überprüft werden (Statusbit: /NRF). Dieses Flipflop wird über das Steuerbit (/NCF) zurückgesetzt und deaktiviert. Wenn /NCF auf ‚low‘ gehalten wird, ist das Flipflop deaktiviert (/NCF = '0').

## 5 Liste der Abbildungen

Bild 1-1	Easylon PCI-Bus Interface .....	5
Bild 3-1	Blockdiagramm Easylon PCI-Bus Interfacekarte .....	17

## 6 Liste der Tabellen

Tabelle 1.1	Varianten, Bestellnummern und Typenbezeichnung der Easylon PCI-Bus Interfacekarte.....	6
Tabelle 3.1	Steckerbelegung für die TP/XF und FTT Variante der Easylon PCI-Bus Interfacekarte.....	18
Tabelle 3.2	Steckerbelegung für die RS-485 Variante der Easylon PCI-Bus Interfacekarte .....	18
Tabelle 3.3	Service LED Bedeutung.....	19
Tabelle 4.1	Zuordnung der .xif Dateien .....	21
Tabelle 4.2	NEURON Chip Adressbelegung für MIP/P50 .....	22
Tabelle 4.3	NEURON Chip Adressbelegung für NSI .....	22
Tabelle 4.4	I/O-Adressbelegung, PCI Bus .....	25
Tabelle 4.5	Signalaufbau Steuerbyte.....	25
Tabelle 4.6	Signalaufbau Statusbyte .....	26

# 7 Index

- .xif Dateien 21
- 16 Bit Applikationen 13
- 3,3 Volt 16
- Abmessungen 20
- Adapter Name 11
- Adressbelegung
  - PCI-Bus 25
- Blockdiagramm 17
- CPU 19
- Debug Flag 11
- Deinstallation 12
- Device Nummer 14
- DOS 13
- EasyCheck 12
- EasyCheck 21
- Eingangspuffer 14
- EMV 20
- Feuchtigkeit 20
- Firmware 12
- Hibernate 12
- I/O Adressen 19
- I/O Port 21
- Installation 8
- Interrupt 22
- IRQ 19
- Leistungsaufnahme 20
- Lon Adapter 11
- Netzwerk Interface 16
- Netzwerk-Interface 21
- Netzwerktreiber 15
- Neuron Chip Adressbelegung
  - MIP/P50 22
  - NSI 22
- PCI-Bus Interface 16, 25
- PCI-X 16
- Permitted Power Saving 12
- Programmierhinweise 21
- Reset 16, 27
- Service LED 5, 16, 19, 22
- Service Pin 22
- Service Taster 5
- Speicher 19
- Standby 12
- Status-Byte 26
- Steck-Schraubklemme 5, 18
- Steuer-Byte 25
- Sub-D Stecker 5, 18
- Technische Daten 19
- Temperatur 20
- Treiber 6, 8
- Varianten 5
- Windows Treiber 8