



Dieses Handbuch betrifft die Baugruppenversion LPS2.

Gesytec GmbH  
Pascalstr. 6  
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: LPS2/UserDoc/LPS2\_Manual-DE-2.1.docx,  
Version 2.1, 4.7.2014



## Dieses Handbuch ...

... gibt Ihnen alle nötigen Informationen, um das EasyLON® Serial Socket Interface einsetzen zu können. Dabei geht es jedoch weder auf die Echelon® LONWORKS Technologie ein, noch wird das Echelon Microprocessor Interface Program (MIP) erklärt, das auf der Baugruppe als Firmware eingesetzt wird. Die Treiberschnittstelle wurde gemäß der Spezifikation der Firma Echelon entwickelt. Auch sie werden hier nicht im Detail behandelt.

Nach einer kurzen Vorstellung des EasyLON Interface in Kapitel 1 beschreibt Kapitel 2 die zur Installation nötigen Schritte.

Kapitel 3 gibt eine technische Beschreibung.

Die „Programmieranleitung“ in Kapitel 4 geben Ihnen die nötigen Informationen zum Erstellen eines eigenen Netzwerktreibers für das EasyLON Interface.

### Literaturhinweis

- “WLDV32 – Programming API for LonWorks Access”, Gesytec
- “LPS Protocol Description”, Gesytec
- 078-0016-01B Host Application Programmers Guide; Echelon Corp.

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.

Das EasyLON Interface verwendet das MIP Programm der Firma Echelon. Die Rechte an dieser Software liegen bei der Echelon Corporation.

# Inhalt

<b>1</b>	<b>Produktinformation</b>	<b>7</b>
1.1	Lieferumfang	7
1.2	Überblick	7
<b>2</b>	<b>Installation</b>	<b>9</b>
2.1	Einbau des Moduls	9
2.2	Netzwerk Interface API	9
2.2.1	EasyCheck	10
<b>3</b>	<b>Technische Beschreibung</b>	<b>11</b>
3.1	Netzwerk Interface	11
3.2	Service LED	11
3.3	DIP Schalter	11
3.4	Serielles Interface	12
3.5	Spannungsversorgung	12
3.6	Steckerbelegung	13
3.7	Technische Daten	14
<b>4</b>	<b>Programmierhinweise</b>	<b>15</b>
4.1	LONWORKS Knoten	15
4.2	Baugruppen Status	15
4.3	Linux Treiber	17
4.3.1	Implementierung	17
4.3.1.1	Serial Line Discipline	17
4.3.1.2	Das Reliable Transport Protokoll	18
4.3.2	Installation	18
4.3.2.1	Übersetzen des Quell-Codes für Treiber und Hilfs-Programme	18
4.3.2.2	Eintragen des Modul-Alias	18
4.3.2.3	Init-Script	18
4.3.2.4	Verwendung des SLTA Treibers	19
4.3.2.5	Gerädateien und Zugriffs-Rechte	19
4.4	Windows CE – Application Interface	21
<b>5</b>	<b>5-Volt-Version</b>	<b>22</b>
5.1	Abweichende Technische Daten	22

5.2	Steckerbelegung .....	23
<b>6</b>	<b>Verzeichnis der Abbildungen.....</b>	<b>24</b>
<b>7</b>	<b>Verzeichnis der Tabellen.....</b>	<b>24</b>
<b>8</b>	<b>Index.....</b>	<b>24</b>

# 1

## Produktinformation

Dieses Handbuch beschreibt das Easylon Serial Socket Interface mit 3,3 V Versorgung. Angaben zum Board mit 5 V Versorgung finden sich in Kapitel 5.



**Bild 1-1 Easylon Serial Socket Interface**

### 1.1 Lieferumfang

- Steck-Baugruppe mit Echelon's MIP Firmware
- „Drivers & Documentation“ CD mit u. a.
  - WLDV32.DLL für Windows und Windows CE (auf Anfrage ist ein Linux Treiber erhältlich)
  - Easylon RNI Software für Fernzugriff auf LonWorks
  - EasyCheck Diagnosesoftware für Easylon Interfaces
  - Beispieldesign für ein Trägerboard (Gerber und Step Daten)

### 1.2 Überblick

Das Easylon Serial Socket Interface stellt als Aufsteckmodul für eine serielle Schnittstelle eine Verbindung zum LonWorks Netzwerk her. Es ist für die Integration in OEM Geräte vorgesehen. Die serielle Verbindung zum CPU Board ist als Conexant Socket Modem kompatibler Steckverbinder ausgeführt über den auch die Spannungsversorgung erfolgt.

Die Baugruppe wird mit 3,3 Volt betrieben. Ein NEURON Chip vom Typ FT5000 zusammen mit einem FT-X2 Transceiver stellt die Schnittstelle zum LonWorks Netz dar. Eine Host CPU mit einem 8051 Kern dient als Protokoll- bzw. Applikationsprozessor. Die Leitungen für Service Taster und -LED sind über den Steckverbinder verfügbar und müssen bei Bedarf auf dem Trägerboard vorgesehen werden. Ebenfalls stehen Signale zum Ansteuern von Receive- und Transmit Leuchtdioden für LonWorks vor. Zwei weitere Leuchtdioden werden von der 8051 CPU angesteuert. Der Entwurf für ein Trägerboard ist wird auf der Drivers

& Documentation CD angeboten, die Sie mit der Baugruppe erhalten. Über die serielle Verbindung kann die Firmware in die Baugruppe geladen werden.

Zum Easylon Serial Socket Interface ist ein Evaluation Kit erhältlich, das Anschlüsse und Signale der Baugruppe einfach zugänglich macht.

Als OEM Modul unterliegt diese Baugruppe einer gewissen Flexibilität in Hinsicht auf Kundenanforderungen. Dies betrifft z. B. die Art und Lage der Steckverbinder oder auch die Bauform. Deshalb kann das tatsächlich vorliegende Modul von den in dieser Dokumentation gemachten Angaben abweichen.

### **Firmwarevarianten**

Je nach verwendeter Firmware kann das Modul in zwei unterschiedlichen Anwendungen eingesetzt werden.

- Das Easylon Serial Socket Interface kann zum einen als serieller LonTalk Adapter verwendet werden. Für den Zugriff auf das Interface wird die WLDV32.DLL unter Windows oder Windows CE verwendet. Für Linux steht ein Treiber zur Verfügung.
- Zum anderen kann das Modul als serielles Gateway verwendet werden. Eine auf dem Modul implementierte Host Application ermöglicht die Verwendung von Netzwerkvariablen (NV), sogar mehr als 62. So können mehr Datenpunkte als in Lösungen nur mit Neuron verwendet werden. Der eingesetzte Prozessor mit großem Speicher gestattet auch den Einsatz komplexerer Protokolle.

Außerdem können auch OEM Programme auf dem Modul implementiert werden, die der Kunde selbst oder Gesytec realisiert.

Anmerkung: Diese Dokumentation beschreibt den Standard Funktionsumfang eines LonWorks Interfaces.

# 2

## Installation

Die Interfacekarte wird im Zustand „unconfigured“ ausgeliefert. Es ist eine Domain mit der Länge 1 und dem Wert 0x0 eingetragen Subnet und NodeID sind 123. Vor Verwendung als Schnittstelle zum LONWORKS Netz muss die Interfacekarte in den Zustand "configured“ gesetzt werden, ebenfalls eine passende Adresse. Netzwerkmanagement-Tools nehmen diese Einstellung automatisch vor oder bieten dem Anwender ein Kommando für diesen Zweck an.

Bei Verwendung diese Easylon Interface 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.

### 2.1 Einbau des Moduls

Bevor Sie das Easylon Interface einbauen, lesen Sie bitte die zugehörige Montageanleitung. Beachten Sie bitte auch alle Vorschriften des Rechnerherstellers über den Einbau zusätzlicher Schnittstellenkarten.

- Stecken Sie das Easylon Serial Socket Interface auf einem freien Conexant Steckplatz auf. Die Übertragungsrate für die serielle Kommunikation ist auf 115200 Baud einzustellen. Andere Baudraten können auf Anfrage geliefert werden.
- Das Modul ist mit einem geeigneten Kabel an das LONWORKS Netz anzuschließen.

### 2.2 Netzwerk Interface API

Es gibt für Windows und Windows CE keinen speziellen Treiber für das Easylon Serial Socket Interface. Gesytec's WLDV32.DLL wird hierfür als Schnittstelle eingesetzt. Einzelheiten dazu finden Sie im Handbuch zur WLDV32.DLL.

Um die WLDV32.DLL zu installieren starten Sie setup.exe im entsprechenden Verzeichnis der "Drivers & Documentation" CD.

Optional ist auch ein Linux Treiber in Quellcode für diese Baugruppe verfügbar.

### 2.2.1 EasyCheck

Auf der Treiber CD gibt es auch einen Setup für die Utility "EasyCheck". Mit ihr sind grundlegende Test und Einstellungen für LonWorks Interfaces von Gesytec möglich.

Für die Verwendung des Easylon Serial Socket Interface sind einige Registry Einstellungen erforderlich. Auf der Drivers & Documentation CD (Drivers/WLDV32) gibt es eine Datei namens "LPS-1-115200.reg", die in die Registry importiert werden muss. Anschließend steht das Serial Socket Interface als COM1-115200 EasyCheck zur Verfügung. Die REG Datei enthält die folgende Einstellung:

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\LonWorks\DeviceDrivers\COM1-115200]
"device name"="LPS/1:115200"
```

EasyCheck öffnet das Interface und zeigt die Treiberversion an. Durch das Senden eines „query status“-Befehls wird die Kommunikation zur Hardware getestet.



# 3

## Technische Beschreibung

### 3.1 Netzwerk Interface

Das Easylon Serial Socket Interface verwendet den FT 5000 NEURON® Chip. Unter der MIP Firmware läuft der FT 5000 mit bis zu 29070 Byte RAM für Netzwerk und Programm Buffers. Der Neuron ist im Slave\_b Modus an eine Embedded Host CPU gekoppelt. Die Host CPU implementiert das serielle Protokoll zur Kommunikation mit einem externen Host.

Auf der Baugruppe ist eine Service LED des Neuron Chip. Ein Service Taster kann durch Verwendung der entsprechenden Pins angeschlossen werden. Dazu kommen Signale für LonWorks Receive und Transmit. Weiterhin können 2 Status LEDs an das Modul angeschlossen werden.

### 3.2 Service LED

Das Service LED Signal ist sowohl auf der Baugruppe wie über den Steckverbinder verfügbar. Es signalisiert den Baugruppen Status. Zusätzlich zu den von Echelon definierten Signalen werden die folgenden Kennungen verwendet.

Service LED	Status	Bemerkung
Blitzt (1 Hz)	Firmware nicht geladen oder ausgefallen.	
Blinkt (1/2 Hz)	Treiber geladen, Knoten ist im Status „unconfigured“ <sup>1</sup> .	Knoten auf „configured“ setzen
Konstant AN	Knoten ist „applicationless“ und „unconfigured“.	
Konstant AUS	Installation ist OK	normaler Betriebszustand

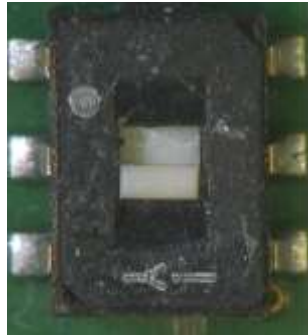
**Tabelle 3.1 Service LED**

### 3.3 DIP Schalter

Der auf der Baugruppe befindliche Schalter dient der Umschaltung zwischen Programmier- und Betriebsmodus. Im Programmiermodus ist eine Aktualisierung der Firmware möglich.

<sup>1</sup> Die Baugruppe wird im Zustand „unconfigured“ ausgeliefert.

**Achtung** Es ist wichtig, dass sich in der Anwendung der DIP Schalter im Betriebsmodus befindet, da die Baugruppe sonst über Hardwarehandshake Signale neu gestartet werden könnte.



DIP Schalter im  
Programmiermodus



DIP Schalter im  
Betriebsmodus

### 3.4 Serielles Interface

Die serielle Schnittstelle verwendet nur die TXD und RXD Signale. Die ebenfalls aufgeführten RTS und DTR Leitungen (vgl. Bild 3-1) werden nur zur Programmierung der Host CPU verwendet.

Die Übertragungsrate ist auf 115 200 Baud eingestellt. Ein Autobaud ist nicht implementiert. Um die Baugruppe mit anderen Übertragungsraten zu betreiben, wenden Sie sich wegen einer anderen Firmware an Gesytec.

### 3.5 Spannungsversorgung

Die serielle Kommunikation zwischen Socket Interface und dem es nutzenden Gerät verwendet TTL Pegel. Deshalb sind folgende Anforderungen an die Spannungsversorgung zu berücksichtigen:

- Das Socket Interface und das serielle Gerät müssen aus der gleichen Spannungsversorgung gespeist werden.
- Beide Geräte müssen gleichzeitig mit Spannung versorgt werden, sonst kann es im nutzenden Gerät oder dem Socket Interface zu einem Latch-Up kommen, der die korrekte Funktion verhindert.

## 3.6 Steckerbelegung

Das Easylon Serial Socket Interface übernimmt die Steckerbelegung vom Conexant Socket Modem, von dem es einen Teil der 64 Pins des Steckers verwendet.

PIN 1 Position s. Bild 3-2  
in Grau: Pin existiert nicht.

LON	o 1	64 o	Serv# key (TTL, input/output)
LON	o 2	63 o	LED1
Earth	o 3	62 o	LED2
	o 4	61 o	3,3 V
	5	60	
	6	59	
	7	58	
	8	57	
	9	56	
	10	55	
	11	54	
	12	53	
	13	52	
	14	51	
	15	50	
	16	49	
	17	48	
	18	47	
	19	46	
	20	45	
	21	44	
	22	43	
	23	42	
Reset# (TTL, input)	o 24	41 o	GND
	o 25	30 o	DTR (TTL, input)
GND	o 26	39 o	
LON RX LED	o 27	38 o	CTS (TTL, output)
LON TX LED	o 28	37 o	
	o 29	36 o	
	o 30	35 o	TxD (TTL, input)
	o 31	34 o	RxD (TTL, output)
	o 32	33 o	RTS (TTL, input)

**Bild 3-1 Steckerbelegung, 3,3-V-Version**

## 3.7 Technische Daten

### Netzwerk Interface

CPU	Neuron FT 5000, 80 MHz
Transceiver	FT-X2
Stecker	Pfostenstecker
Service LED	auf Modul oder über Conexant Stecker
Service Taster	über Conexant Stecker
RX, TX LEDs	über Conexant Stecker
Isolation	1000 V <sub>eff</sub> 60 s
Kompatibilität	LonTalk, DIN EN 14908

### Serielles Interface

CPU	AT89C51RE2, 18,432 MHz
Speicher	Flash 64 KByte, RAM 8 KByte
Typ	TTL, Conexant Standard, RxD, TxD, RTS, CTS
Stecker	Pfostenstecker
Übertragung	115,2 kBd (andere auf Anfrage, max. 230,4 kBd)

### Spannungsversorgung

Spannung	3,3 V DC +- 5%, extern über Conexant Anschluss
Stromverbrauch	70 mA typ., 95 mA max.

### Abmessungen & Umgebungsbedingungen

Temperatur	Betrieb -40 – +85 °C Lagerung -40 – +85 °C
Feuchtigkeit	Klasse F nach DIN 40040, keine Betauung
Abmessungen	64,5 x 26,5 [mm]
Bauhöhe über Board	8,4 mm

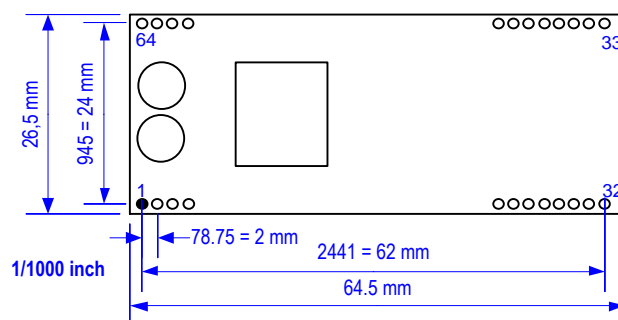


Bild 3-2 Abmessungen und Steckerpins

# 4

## Programmierhinweise

Dieser Abschnitt gibt einige Programmierhinweise zum Easylon Serial Socket Interface.

### 4.1 LONWORKS Knoten

Das Easylon Serial Socket Interface ist ein Knoten im LONWORKS Netzwerk. Es wird unter Echelon's Microprocessor Interface Program MIP Firmware betrieben und verwendet den NEURON FT5000 Chip als Kommunikationsprozessor. Die zugehörige XIF-Datei (eXternal Interface File, .xif) findet sich auf der „Drivers & Documentation“ CD. Die Zuordnung von „.xif“-Datei und Kartentyp entnehmen Sie bitte der folgenden Tabelle:

Netzwerk Interface	Übertragungsrate	XIF-Datei
TP/FT	78,125 kbps	P.P109x6.xif

**Tabelle 4.1 Zuordnung der .xif Dateien**

### 4.2 Baugruppen Status

Eigene Programme die das Easylon Serial Socket 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 Web-Seite 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 Linux Treiber

Optional wird für die Interface Baugruppe ein Linux Treiber im Quellcode zur Verfügung gestellt. Dieser implementiert ein Device namens „slta“. Die Treiberschnittstelle entspricht der der anderen EasyLyon Interfaces.

### 4.3.1 Implementierung

#### 4.3.1.1 Serial Line Discipline

Der Linux Kernel bietet mit dem Konzept der „Serial Line Discipline“ eine flexible Methode, um vielfältige Protokolle auf serielle Übertragungstechniken abzubilden. Für den „SLTA“ Treiber ergeben sich dadurch folgende Vorteile:

- größtmögliche Kompatibilität zu verschiedenen Kernel-Versionen, da nur eine „Zwischenschicht“ der seriellen Ports implementiert werden muss, damit entstehen nur geringe Abhängigkeiten von kernel-spezifischen Funktionen
- Unabhängigkeit von der verwendeten seriellen Hardware, der Treiber kann über jede TTY-Schnittstelle betrieben werden

Damit der Treiber auf allen Systemen funktionsfähig ist, wurde die ID einer dem Linux Kernel bereits bekannten Line Discipline verwendet. Die Verwendung einer neuen eindeutigen ID würde die Änderung des Kernels notwendig machen. Um dies zu umgehen, wird die reservierte aber nicht verwendete ID N\_MASC eines „Mobitex Moduls“ verwendet.

**Hinweis:** Die ID der Line Discipline sind in /usr/include/asm//termios.h definiert.

### 4.3.1.2 Das Reliable Transport Protokoll

Der Treiber basiert auf dem Solaris Treiber von Echelon. Dieser Treiber verwendet eine Receive- und Transmit-Statusmaschine, die das Reliable Transport Protokoll unterstützt.

## 4.3.2 Installation

### 4.3.2.1 Übersetzen des Quell-Codes für Treiber und Hilfs-Programme

Zum Übersetzen des Treibers ist die Installation des Kernel Quell Codes notwendig. Je nach verwendeter Distribution ist dazu das Kernel Entwicklerpaket zu installieren. Zur Installation des Treibers sind Administrator Rechte notwendig.

Mit den folgenden Kommandos

```
make
make tools
```

werden der Treiber und das Programm slta-attach übersetzt. Mit den Befehlen

```
make install
make setup
```

werden der Treiber und das Programm slta-attach in das System kopiert.

### 4.3.2.2 Eintragen des Modul-Alias

Damit die verwendete Line Discipline mit dem richtigen Treiber verknüpft wird, sollte in den Konfigurationsdateien /etc/modprobe.conf bzw. /etc/modprobe.conf.local folgende Zeile eingetragen werden:

```
alias tty-ldisc-8 slta
```

Damit wird der Index 8 der seriellen Line Discipline mit dem Treiber-Modul slta verbunden. Mit der zusätzlichen Zeile

```
options slta slta_debug=1
```

wird dem Treiber die angegebene Option beim Laden übergeben.

### 4.3.2.3 Init-Script

Um den SLTA Treiber zu aktivieren, muss der Treiber geladen und das Programm slta-attach beim Systemstart aufgerufen werden. Dies wird bei den meisten Linux Distributionen mit einem SysV-Initscript durchgeführt, exemplarisch für SuSE-Linux ist ein Beispiel initscript.suse beigefügt. Kopieren Sie dieses Script in das SysV-Init Verzeichnis:

```
cp initscript.suse /etc/init.d/slta
```



Aktivieren Sie dann mittels des Yast Runlevels-Editors den SLTA Treiber, so dass das Init-Script bei jedem Systemstart automatisch aufgerufen wird. Für andere Distributionen existieren entsprechende Setup-Tools, konsultieren Sie die System Dokumentation. Alternativ kann das Script manuell aufgerufen und der Treiber gestartet werden:

```
/etc/init.d/slta start
```

#### 4.3.2.4 Verwendung des SLTA Treibers

In dem Init-Script müssen die Namen der Gerätedateien der seriellen Ports angegeben werden, an denen SLTAs angeschlossen sind. Dazu muss mit einem Texteditor folgende Zeile des Init-Scripts entsprechend angepasst werden.

```
DAEMON_PORTS="/dev/ttyS0 /dev/ttyS1"
```

Weiterhin legt das Init-Script die Gerätedateien mit folgenden Namen an:

```
/dev/lon1, /dev/lon2, usw.
```

Sie erhalten einen aufsteigendem Index entsprechend der Reihenfolge in der die seriellen Ports in der Variable DAEMON\_PORTS genannt sind.

#### 4.3.2.5 Gerätedateien und Zugriffs-Rechte

Die seriellen Geräte werden über Einträge im /dev Verzeichnis identifiziert. Leider ist die Namensgebung nicht einheitlich, und auch zwischen den Distributionen existieren unterschiedliche Benennungen. Meistens bezeichnet /dev/ttyS0 den ersten seriellen Port, entsprechend COM1: unter DOS. Mit Einführung des Device-Filesystems devfs wurde der Name /dev/tts/0 eingeführt. Die aktuelle Entwicklung ermöglicht eine flexible Benennung mittels udev, aber auch hier setzen nicht alle Distributionen auf eine einheitliche Namensgebung. Dennoch wird meist – auch bei SuSE 10.x und Fedora Core 3 und größer - wieder /dev/ttyS0 für COM1: verwendet. Ein serieller USB Port wird meist /dev/ttyUSBn genannt, bzw. /dev/usb/tts/0 bei Verwendung des devfs.

Um den SLTA Treiber verwenden zu können, müssen dem Benutzer-Account die entsprechenden Zugriffs Rechte auf den seriellen Port eingeräumt werden. Je nach Linux bzw. Kernel Version können unterschiedliche Ansätze befolgt werden.

##### Abhängig von der Distribution

- a) SuSE 10.x, SuSE 9.x, Fedora Core

Der serielle Port kann von allen Mitgliedern der Gruppe uucp verwendet werden, fügen Sie den Benutzer-Account, unter dem das Anwendungsprogramm ausgeführt wird, der Gruppe uucp hinzu.

Hinweis: Zur Verwaltung von Benutzer-Accounts siehe auch hier:  
<http://www.oreilly.de/german/freebooks/rlinux3ger/ch058.html#7724>

## Abhängig von der System- und Kernel-Konfiguration

### a) Verwendung von statischen Gerädateien

Bei einem statisch angelegten /dev Verzeichnis können über die Rechte des Dateisystems mit dem Kommando

```
chmod 0666 /dev/ttyS0
```

allen Benutzern des Systems Lese- und Schreibrechte auf den seriellen Port eingeräumt werden.

### b) Verwendung von devfs

Werden die Gerädateien bei jedem Systemstart dynamisch mittels devfs angelegt, muss die Rechtevergabe in der Konfigurationsdatei /etc/devfsd.conf vorgenommen werden.

Hinweis: ob devfs aktiv ist, können Sie anhand der Existenz der Datei /dev/.devfsd feststellen

Mit der Zeile

```
REGISTER ^tts/. * PERMISSIONS root.dip 0660
```

wird festgelegt, dass alle Gerädateien für serielle Ports dem Benutzer root sowie der Gruppe dip angehören, der Benutzer root und alle Mitglieder der Gruppe dip habe Lese- und Schreibrechte, alle anderen Systembenutzer dürfen nicht auf den Port zugreifen. Um allen Benutzern die Verwendung zu gestatten, müssen die Zugriffsrechte in

```
0666
```

geändert werden. Alternativ kann auch allen Mitgliedern der Gruppe users der Zugriff erlaubt werden, die Gruppenzugehörigkeit muss also auf

```
root.users
```

geändert werden.

### c) Verwendung von udev

Seit der Kernelversion 2.6.13 ist devfs nicht mehr verfügbar, statt dessen wird das Userspace Programm udev verwendet. Die zugehörige Konfigurationsdatei steht in dem Verzeichnis /etc/udev/rules.d, meistens mit dem Namen 50-udev.rules.

Hinweis: ob udev aktiv ist, können Sie anhand der Existenz des Verzeichnisses /etc/.udev feststellen

Mit der Zeile

```
KERNEL="ttyS*", GROUP="uucp", MODE="0660"
```

wird festgelegt, dass alle Gerädateien für serielle Ports dem Benutzer root sowie der Gruppe uucp angehören, der Benutzer root und alle Mitglieder der Gruppe uucp habe Lese- und Schreibrechte, alle anderen Systembenutzer dürfen nicht auf den Port zugreifen. Um allen Benutzern die Verwendung zu gestatten, müssen die Zugriffsrechte geändert werden in:

```
MODE="0666"
```

## 4.4 Windows CE – Application Interface

Auf der „Drivers & Documentation“ CD gibt es eine spezielle Windows CE Version der WLDV32.DLL zur Verwendung dieser Baugruppe unter Windows CE.

# 5 5-Volt-Version

Um Anwendern der Vorgänger Version des Easylon Serial Socket Interface einen nahtlosen Umstieg zu ermöglichen, gibt es eine Version mit 5 V Spannungsversorgung und dem „alten“ Pin-out. Diese Kapitel beschreibt die Unterschiede der 5-Volt-Version zu der zuvor beschriebenen 3,3-Volt-Version.

## 5.1 Abweichende Technische Daten

Hier sind nur solche Informationen aufgeführt, die von der 3,3-V-Version abweichen.

### Spannungsversorgung

Spannung	5 V DC +- 5%, extern über Conexant Anschluss
Stromverbrauch	85 mA typ., 110 mA max.

## 5.2 Steckerbelegung

Das Easylyon Serial Socket Interface orientiert sich an der Steckerbelegung des Conexant Socket Modem, von dem es einen Teil der 64 Pins des Steckers verwendet.

PIN 1 Position s. Bild 3-2  
in Grau: Pin existiert nicht.

LON	o 1	64 o	Serv# key (TTL, input, output)
LON	o 2	63 o	LED1
Earth	o 3	62 o	LED2
	o 4	61 o	<b>5 V</b>
	5	60	
	6	59	
	7	58	
	8	57	
	9	56	
	10	55	
	11	54	
	12	53	
	13	52	
	14	51	
	15	50	
	16	49	
	17	48	
	18	47	
	19	46	
	20	45	
	21	44	
	22	43	
	23	42	
Reset# (TTL, input)	o 24	41 o	GND
	o 25	30 o	DTR (TTL, input)
GND	o 26	39 o	
LON RX LED	o 27	38 o	CTS (TTL, output)
LON TX LED	o 28	37 o	
	o 29	36 o	
	o 30	35 o	<b>TxD (TTL, output)</b>
	o 31	34 o	<b>RxD (TTL, input)</b>
	o 32	33 o	RTS (TTL, input)

Bild 5-1 Steckerbelegung, 5-V-Version

## 6 Verzeichnis der Abbildungen

Bild 1-1	Easylon Serial Socket Interface .....	7
Bild 3-1	Steckerbelegung, 3,3-V-Version.....	13
Bild 3-2	Abmessungen und Steckerpins .....	14
Bild 5-1	Steckerbelegung, 5-V-Version.....	23

## 7 Verzeichnis der Tabellen

Tabelle 3.1	Service LED.....	11
Tabelle 4.1	Zuordnung der .xif Dateien.....	15

## 8 Index

5-Volt-Version	22	serielles Interface	12
Abmessungen	14	Service LED	11
Baudrate	12	Service Taster	11
configured	9	Steckerbelegung	13, 23
DIP Schalter	11	Steckverbinder	7, 13, 23
EasyCheck	10	Technische Daten	14, 22
Linux	17	Übertragungsrate	12
Netzwerk Interface	11	unconfigured	9
Netzwerk Interface API	9	Windows CE	21
Programmierhinweise	15	WLDV32.DLL	9, 21
serielles Gateway	8	xif Dateien	9, 15