FCC NOTICE (for USA only)

Federal Communications Commission Radio Frequency Interference Statement

WARNING

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the use is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Changes or modifications not expressly approved by Echelon Corporation could void the user’s authority to operate the equipment.

Safety

<table>
<thead>
<tr>
<th>UL</th>
<th>Listed, per UL-60950</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSA (c-UL)</td>
<td>Certified, per C22.2 no 60950-00</td>
</tr>
<tr>
<td>TUV</td>
<td>Certified, per EN 60950</td>
</tr>
</tbody>
</table>

Canadian Doc Notice

This digital apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n’émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe B prescrites dans le règlement sur la brouillage radioélectrique édicté par le Ministère des Communications du Canada.
This document describes how to use the Power Line Serial LonTalk® Adapter (PL-SLTA) to connect a host processor with an EIA-232 (formerly RS-232) serial interface to a LONWORKS® network. The connection to the host processor can be made either directly or remotely through a pair of modems.
Contents

This manual provides detailed information about the hardware and software for the PL-SLTA.

- Chapter 1 introduces the PL-SLTA and provides a quick overview.
- Chapter 2 describes the PL-SLTA hardware.
- Chapter 3 describes how to attach a PL-SLTA.
- Chapter 4 describes the configuration switches of the PL-SLTA.
- Chapter 5 describes the software for the PL-SLTA NSI mode.
- Chapter 6 discusses using the Windows® 98/NT/2000/XP network driver and the SLTA Link Manager software with the PL-SLTA NSI mode.
- Chapter 7 discusses using a PL-SLTA with a modem and provides specific instructions for setting up the PL-SLTA for remote dial-in access for network monitoring and service operations.
- Chapter 8 explains how to use a programmable serial gateway.
- Chapter 9 is a troubleshooting section.

Related Manuals

The following Echelon documents are suggested reading:

- The LonBuilder® User’s Guide describes how to develop LONWORKS applications with the LonBuilder Developer's Workbench.
- The NodeBuilder® User's Guide describes how to develop LONWORKS applications with the NodeBuilder Development Tool.
- Neuron® Chip data books provided by Toshiba and Cypress contain specifications and describe the architecture of the Neuron Chip.

Web Access

Engineering bulletins and data sheets supporting this product are available on the Echelon Web site. General information regarding Echelon, its business, and its products is also located on the site at http://www.echelon.com.
Contents

Preface .......................................................... 4
  Content .......................................................................................... 5
  Related Manuals ............................................................................. 5
  Web and FTP Access ....................................................................... 5

1 PL-SLTA Overview .................................................. 8
  Introduction ......................................................................................... 9
  Configuration ....................................................................................... 10
  Transceivers ..................................................................................... 10
  Power supply .................................................................................... 10
  Configuration ....................................................................................... 11
  Configuration ....................................................................................... 11

2 PL-SLTA Hardware .............................................. 12
  Mechanical Description ................................................................. 13
  Switches, Indicators, and Connectors............................................. 13
  Connecting Power ............................................................................ 14

3 Cabling and Connections .......................................... 15
  LonWorks Network Connection ....................................................... 16
  Attaching the PL-SLTA ..................................................................... 16
  Attaching the PL-SLTA to a PC ......................................................... 16
  Attaching the PL-SLTA to a Modem ................................................ 17

4 Hardware Configuration ....................................... 19
  Configuring the PL-SLTA Hardware ............................................. 20
  Autobaud (AB) .............................................................................. 21

5 The PL-SLTA NSI Mode Software ......................... 24
  PL-SLTA (NSI Mode) c2.Software Overview ................................ 25
  Software Installation Procedure .................................................... 25

6 Using the Windows Driver and SLTA Link Manager with the PL-SLTA 26
  Software Overview ......................................................................... 27
  Establishing a Communications Line for Dialing into a Network .... 28
  Establishing a Communications Line for Calls Dialed Out to the PC .. 30
  Establishing Remote and Local Network Sites ............................. 31
  SLTA Link Configuration Script Formats ....................................... 31
  The SLTA Link Manager Programmatic Interface ....................... 36
  Characteristics of a Well Designed System .................................. 37
  Call Initiation ................................................................................. 37
  Call Termination ............................................................................... 39
  Monitoring Application Termination Strategy ............................ 39
  Monitoring Missing Messages After a Dial-Out ........................... 40
  Monitoring LNS Application Design Issues ................................. 40
  Good Practices/Schemes that Work ............................................... 41

7 Using the PL-SLTA with a Modem ................................ 45
  Overview .......................................................................................... 46
Power Line SLTA Overview

The Power Line Serial LonTalk Adapter, called PL-SLTA, (Models 76000-1, 76000-2, 76000-3 and 76000-4) is a network interface that enables any host processor with an EIA-232 serial interface to connect to a LONWORKS network. The PL-SLTA extends the reach of LONWORKS technology to a variety of hosts, including desktop, laptop, and palmtop PCs, workstations, embedded microprocessors, and microcontrollers.

The PL-SLTA is compatible with LNS-based applications.
Introduction

The PL-SLTA is an EIA-232 (formerly RS-232) compatible serial device that allows any host with an EIA-232 interface and proper software to communicate with a LONWORKS network.

When used with a PC host and the LNS Developer’s Kit for Windows software or Echelon’s OSGi bundles, the PL-SLTA can be used to build sophisticated network management, monitoring, and control tools for LONWORKS networks. A PL-SLTA can be connected to the host through a pair of modems and the telephone network, allowing for remote monitoring, controlling, or network management. The PL-SLTA can be set to answer incoming calls from a remote host. In addition, any node on the local network can initiate a telephone call to a remote host computer. The PL-SLTA itself may be configured to initiate a phone call to a remote host computer whenever it detects that a network variable on the local network should be sent to a remote host.

Figure 1 illustrates a typical node based on a PL-SLTA. Chapter 7, Using the PL-SLTA with a Modem, shows a PL-SLTA connected to a host through the telephone network.

The PL-SLTA provides network services interface (NSI mode) functionality for use with LNS-compliant applications.
Table 1. PL-SLTA NSI Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>PL-SLTA (NSI Mode)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports LNS applications</td>
<td>YES</td>
</tr>
<tr>
<td>Available drivers</td>
<td>Windows 98/NT/2000/XP</td>
</tr>
<tr>
<td>Software used to establish connections via modems</td>
<td>SLTA Link Manager (TAPI application)</td>
</tr>
<tr>
<td>Who initiates calls from a remote network</td>
<td>PL-SLTA itself or “Helper / Dialer” node</td>
</tr>
<tr>
<td>Input power options</td>
<td>100-240VAC</td>
</tr>
<tr>
<td>Configuration Switches</td>
<td>Externally accessible</td>
</tr>
<tr>
<td>Processor Input Clock</td>
<td>10 MHz</td>
</tr>
<tr>
<td>Transceiver versions available</td>
<td>PLT-22</td>
</tr>
<tr>
<td>To attach to a modem</td>
<td>Special null modem cable (include port number)</td>
</tr>
<tr>
<td>Default Buffer Configuration</td>
<td>See Chapter 4</td>
</tr>
</tbody>
</table>

**The PL-SLTA Configurations**

The PL-SLTA is available with the following transceiver and power supply options:

- **Transceivers**: The PL-SLTA is available with the PLT-22 transceiver. Both CENELEC and non-CENELEC modes of operation are supported.

- **Power supply**: 100 - 240 VAC 50/60Hz. Power cords are available in U.S./Canada (Model 76000-1), U.K. (Model 76000-4), continental European (Model 76000-3), and Japanese (Model 76000-4) configurations. The power line coupling circuit is built into the PL-SLTA.

**Software Availability**

The PL-SLTA is not shipped with software. Software for the PL-SLTA is available with the LonMaker Integration Tool (Model 37000), on the LNS Application Developer’s Kit for Windows CD (Model 34309) or from the Developer’s Toolbox at the Echelon Web site (http://www.echelon.com).
**LNS Compatibility**

When the PL-SLTA is connected directly to the PC host (i.e., no modems), the PL-SLTA behaves like any other NSI, such as the PCLTA-XX or the PCC-10 PC Card. The PL-SLTA is also designed so that a PC host can be connected to the network through a pair of modems and the telephone network. When the PC initiates the phone call to the PL-SLTA, the remote host is said to “dial-in to the network”. When the PL-SLTA initiates a call to the PC, the PL-SLTA is said to “dial-out to the remote host”. Once the phone connection is established, the application running on the remote host can perform network management, monitoring, or control activities.

*Note:* A remote LNS Server (via modems) requires a dedicated PL-SLTA. Thus, some networks may require multiple PL-SLTAs—one for the remote LNS Server and others to allow access for other PCs.

---

**TAPI Compatibility**

The SLTA Link Manager software uses TAPI release 1.3 or higher. This is supported in Windows NT 4.0 and higher.
This chapter provides a physical description of the PL-SLTA.
Mechanical Description

Figure 2 shows the PL-SLTA in its enclosure.

Figure 2. PL-SLTA Enclosure

Switches, Indicators, and Connectors

ESD WARNING

THIS PRODUCT CONTAINS COMPONENTS WHICH ARE SENSITIVE TO STATIC ELECTRICITY. BEFORE INSTALLING OR REMOVING THE NETWORK OR SERIAL CABLES, TOUCH EARTH GROUND WITH YOUR HAND TO DISCHARGE ANY STATIC ELECTRICITY WHICH MAY HAVE ACCUMULATED.

Table 2 describes the external connections, switches and LEDs on the PL-SLTA.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Button</td>
<td>Pressing this button grounds the service request pin on the Neuron Chip within the PL-SLTA. While this switch is pressed, the service LED should light to maximum intensity. If Switch 3 (the Network Disable switch) on the configuration DIP switches panel is in the ON/up position the service LED will light, but no service message will be sent, even if the PL-SLTA has been configured after the last power up, until the host attaches to it and commands it to leave this flush state.</td>
</tr>
<tr>
<td>Service LED (Yellow)</td>
<td>Indicates that either the Service Button is being pressed or, if not:</td>
</tr>
<tr>
<td></td>
<td>- on The PL-SLTA firmware has detected an unrecoverable error and/or the PL-SLTA is Applicationless. Reboot the PL-SLTA from another network interface on the channel.</td>
</tr>
<tr>
<td></td>
<td>- blinking PL-SLTA is unconfigured.</td>
</tr>
<tr>
<td></td>
<td>- off PL-SLTA is configured or there is no power. Check LED.</td>
</tr>
<tr>
<td>Power Indicator LED (Green)</td>
<td>Indicates presence of input power to the PL-SLTA. This LED is driven by IO_8 and indicates not only power but also the fact that the device is executing its firmware.</td>
</tr>
<tr>
<td>PKT Detect LED (Yellow)</td>
<td>Indicates when the PL-SLTA has detected a valid LonTalk packet.</td>
</tr>
<tr>
<td>Configuration DIP Switch</td>
<td>See Chapter 4 for details.</td>
</tr>
</tbody>
</table>

**Connecting Power**

The PL-STA is designed to plug directly to the power mains with an input voltage of 100 – 240VAC @ 50-60Hz. When power is connected, the yellow service LED will briefly flash and the green power indicator LED will turn on. Once the PL-SLTA is powered and configured, the service LED will remain off unless the service request switch is pressed.
This chapter demonstrates how to attach the PL-SLTA to a 
LONWORKS network, a PC, and a modem.

To connect a PL-SLTA to a modem, use a SPECIAL null modem 
cable. The PL-SLTA Null Modem Cable (Model 73380) is available 
for purchase from Echelon. The cable specifications also have been 
included in this chapter. Note that most off-the-shelf null modem 
cables will not work in this configuration.
LONWORKS Network Connection

The PL-SLTA communicates by superimposing data communications over the existing power mains. The power cord serves both as the power mains cord and the network wire.

Attaching the PL-SLTA

EIA-232 devices are configured as either Data Circuit-terminating Equipment (DCE) or as Data Terminal Equipment (DTE). A DCE device connects to a DTE device, unless a null modem cable is used. Using a null modem cable, a DCE device connects to a DCE device and a DTE device connects to a DTE device. The PL-SLTA is a DCE device.

The standard configuration for a PC/AT EIA-232 serial I/O port is a DTE device. PCs usually take the ‘terminal’ role in communications. Modems should always be DCE devices. To connect a PL-SLTA to a PC, simply connect one end of the serial cable to the PL-SLTA, and the other end of the cable to the PC’s serial port. To connect a PL-SLTA to a modem, a SPECIAL null modem cable must be used. Note that most standard off-the-shelf null modem cables will not work in this configuration.

Echelon offers the Model 73380 PL-SLTA Null Modem Cable, which is a DB-9 male to DB-25 male null modem cable. This and other cables used with the PL-SLTA are described in detail in this chapter.

Attaching the PL-SLTA to a PC

Most PC serial I/O ports have a 9-pin male DB-9 connector, although some have a 25-pin male DB-25 connector. When connecting to a PC, use a straight-through cable.

Table 3. PC DB-9 to DB-9 Connection

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>PC Connector DB-9 Male</th>
<th>Cable DB-9 Female</th>
<th>Cable DB-9 Male</th>
<th>PL-SLTA (DCE) DB9 Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD</td>
<td>Pin 2</td>
<td>Pin 2</td>
<td>Pin 2</td>
<td>Pin 2</td>
</tr>
<tr>
<td>TxD</td>
<td>Pin 3</td>
<td>Pin 3</td>
<td>Pin 3</td>
<td>Pin 3</td>
</tr>
<tr>
<td>Signal Ground</td>
<td>Pin 5</td>
<td>Pin 5</td>
<td>Pin 5</td>
<td>Pin 5</td>
</tr>
</tbody>
</table>

Figure 3. DB-9 to DB-9 Connection
### Table 4. PC 25-Pin to DB-9 Connection

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>PC Connector</th>
<th>Cable DB-25 Female</th>
<th>Cable DB-9 Male</th>
<th>PL-SLTA (DCE) DB-9 Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD</td>
<td>Pin 3</td>
<td>Pin 3</td>
<td>Pin 2</td>
<td>Pin 2</td>
</tr>
<tr>
<td>TxD</td>
<td>Pin 2</td>
<td>Pin 2</td>
<td>Pin 3</td>
<td>Pin 3</td>
</tr>
<tr>
<td>Signal Ground</td>
<td>Pin 7</td>
<td>Pin 7</td>
<td>Pin 5</td>
<td>Pin 5</td>
</tr>
</tbody>
</table>

![Diagram of PC 25-Pin to DB-9 Connection](image)

**Figure 4. PC 25-Pin to DB-9 Connection**

---

### Attaching the PL-SLTA to a Modem

You must use a “specific null modem cable” to attach the PL-SLTA to a modem. See the following tables and figures for the correct cable. When using the SLTA-10 with a modem, power up the modem first and then power up the SLTA-10 so the SLTA-10 can send an initialization string to the modem.

### Table 5. DCE Modem to PL-SLTA Connection (DB-9 to DB-9)

<table>
<thead>
<tr>
<th>Modem Signal Name</th>
<th>Cable DB-9 Male</th>
<th>Null Modem</th>
<th>Cable DB-9 Male</th>
<th>PL-SLTA (DCE) DB-9 Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD—Pin 2</td>
<td>Pin 2</td>
<td>Pin 2-3</td>
<td>Pin 3</td>
<td>TxD—Pin 3</td>
</tr>
<tr>
<td>TxD—Pin 3</td>
<td>Pin 3</td>
<td>Pin 3-2</td>
<td>Pin 2</td>
<td>RxD—Pin 2</td>
</tr>
<tr>
<td>DCD—Pin 1</td>
<td>Pin 1</td>
<td>Pin 1-4</td>
<td>Pin 4</td>
<td>DTR—Pin 4</td>
</tr>
<tr>
<td>DTR—Pin 4</td>
<td>Pins 4 &amp; 7</td>
<td>Pins 4, 7 - 6</td>
<td>Pin 6</td>
<td>DSR—Pin 6</td>
</tr>
<tr>
<td>RTS—Pin 7</td>
<td>GND—Pin 5</td>
<td>Pin 5-5</td>
<td>Pin 5</td>
<td>GND—Pin 5</td>
</tr>
</tbody>
</table>

![Diagram of DCE Modem to PL-SLTA Connection (DB-9 to DB-9)](image)

**Figure 5. DCE Modem to PL-SLTA Connection (DB-9 to DB-9)**
Table 6. DCE Modem to PL-SLTA Connection (DB-25 to DB-9)

<table>
<thead>
<tr>
<th>Modem Signal Name</th>
<th>Cable DB-25 Male</th>
<th>Cable DB9 Male</th>
<th>PL-SLTA (DCE) DB-9 Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD—Pin 3</td>
<td>Pin 3</td>
<td>Pin 3</td>
<td>TxD—Pin 3</td>
</tr>
<tr>
<td>TxD—Pin 2</td>
<td>Pin 2</td>
<td>Pin 2</td>
<td>RxD—Pin 2</td>
</tr>
<tr>
<td>DCD—Pin 8</td>
<td>Pin 8</td>
<td>Pin 4</td>
<td>DTR—Pin 4</td>
</tr>
<tr>
<td>DTR—Pin 20</td>
<td>Pins 20 &amp; 4</td>
<td>Pin 6</td>
<td>DSR—Pin 6</td>
</tr>
<tr>
<td>RTS—Pin 4</td>
<td>Pins 20 &amp; 4</td>
<td>Pin 6</td>
<td>DSR—Pin 6</td>
</tr>
<tr>
<td>GND—Pin 7</td>
<td>Pin 7</td>
<td>Pin 5</td>
<td>GND—Pin 5</td>
</tr>
</tbody>
</table>

Figure 6. PL-SLTA Null Modem Cable (DB-25 to DB-9)
This chapter describes how to install and configure a PL-SLTA.
Configuring the PL-SLTA Hardware

Eight configuration switches are located on the PL-SLTA’s configuration DIP-switch block. These inputs are read by the PL-SLTA firmware to configure or enable features. Figure 7 shows the factory default settings. Changes to the switch configurations will not occur until the power is cycled on the PL-SLTA. The switches are read immediately after a power reset.

Figure 7. PL-SLTA Default Switch Settings

Reserved (Switch1 / CFG3)

Modem Support (Switch 2 / CFG2)

Switch2 / CFG2 controls the use of the PL-SLTA with a modem. If the PL-SLTA is connected directly to a PC, then CFG2 should be set to the Local Host state (down position). This is the PL-SLTA default. If the PL-SLTA is connected to a modem, then CFG2 must be set to the Remote Host state.

Figure 8. PL-SLTA Host Switch2 / CFG2
Switch 3 / CFG1 enables (down position) or disables (up position) network communications after reset. If disabled, the PL-SLTA will not be able to communicate on the network after a reset until it receives an \texttt{niFLUSH\_CANCEL} command from the host.

The PL-SLTA prevents network communications by entering a \texttt{FLUSH} state. This state causes the PL-SLTA to ignore all incoming network messages and prevents all outgoing network messages, even service pin messages. In the disabled state, if the service pin is pressed the service LED will light but no service message will be sent. This \texttt{FLUSH} state is provided to prevent any other network management tools from performing network management functions on the PL-SLTA before the PL-SLTA Adapter’s host has a chance to perform any of its own network management functions. This state is canceled with the \texttt{niFLUSH\_CANCEL} command from the host.

Reserved (Switch4)

CENELEC (Switch5)

Switch 5 (labeled CENELEC) on the PL-SLTA is used to select the CENELEC access protocol. When the switch is up/on, the PL-SLTA is using CENELEC. When the switch is off/down (default position), CENELEC is disabled.

![Figure 9. PL-SLTA Network Switch 5 / CENELEC/Non-CENELEC](image)

To allow multiple power line signaling devices from different manufacturers to operate on a common AC mains circuit, the CENELEC standard EN 50065-1 specifies an access protocol for the C-band (125kHz to 140kHz). The frequency 132.5kHz is designated as the primary band-in-use frequency that indicates when a transmission is in progress.

Every power line signaling device must both monitor the 132.5kHz band-in-use frequency and be able to detect the presence of a signal of at least 80dB\(\mu\)Vrms anywhere in the range from 131.5kHz to 133.5kHz which has a duration greater than or equal to four milliseconds. A power line signaling device is permitted to transmit if the band-in-use detector shows the band to have been free for at least 85 milliseconds. Each device must randomly choose an interval for transmission, and at least seven evenly distributed intervals must be available for selection. A group of power line signaling devices is allowed to transmit continually for a period less than or equal to one second, after which it must cease transmitting for at least 125 milliseconds.
When the CENELEC access protocol is enabled, overall network throughput is reduced by 11%.

**Serial Bit Rate (Switches[6..8] / BAUD[2..0])**

Switches 6, 7 and 8, named BAUD[2..0], are used to set the PL-SLTA serial bit rate. Eight bit rates ranging from 300 bps to 115,200 bps are available. The PL-SLTA factory default is 38,400 bps. All data is transmitted using 1 start bit, 8 data bits, no parity bits, and 1 stop bit. The PC serial bit rate is configured using the SLTA Link Manager as explained in Chapter 6.

![Serial Baud Rate Switches Diagram](image-url)
<table>
<thead>
<tr>
<th>NSI-mode Buffer Parameter</th>
<th>Default Count</th>
<th>Default Size in Bytes</th>
<th>Default Total Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive transaction buffers</td>
<td>16</td>
<td>13</td>
<td>208</td>
</tr>
<tr>
<td>Transmit transaction buffers</td>
<td>2</td>
<td>28</td>
<td>56</td>
</tr>
<tr>
<td>Application input buffers</td>
<td>3</td>
<td>255</td>
<td>765</td>
</tr>
<tr>
<td>Application output buffers</td>
<td>3</td>
<td>255</td>
<td>765</td>
</tr>
<tr>
<td>Network input buffers</td>
<td>2</td>
<td>66</td>
<td>132</td>
</tr>
<tr>
<td>Network output buffers</td>
<td>2</td>
<td>66</td>
<td>132</td>
</tr>
<tr>
<td>Priority application output buffers</td>
<td>3</td>
<td>255</td>
<td>765</td>
</tr>
<tr>
<td>Priority network output buffers</td>
<td>2</td>
<td>66</td>
<td>132</td>
</tr>
<tr>
<td><strong>Total bytes used for buffers</strong></td>
<td></td>
<td></td>
<td><strong>2955</strong></td>
</tr>
</tbody>
</table>
This chapter describes the Windows Windows 98/NT/2000/XP software used with the PL-SLTA. This software is available with the LonMaker Integration Tool (Model 37000), as part of the LNS Application Developer’s Kit (ADK) (Model 34303) and on the Echelon Web site (www.echelon.com).

*Note:* Echelon no longer supports Windows 98 and Windows NT. The information presented in this user’s guide is provided for completeness and should be used for reference only.
PL-SLTA (NSI Mode) Software Overview

The PL-SLTA is not shipped with software. The Windows 98/NT/2000/XP driver and SLTA Link Manager software are available with the LonMaker Integration Tool (Model 37000), the LNS Application Developer’s Kit for Windows (Model 34303), and on the Echelon Web site (www.echelon.com).

The PL-SLTA set-up installs the PL-SLTA NSI mode Windows 98/NT/2000/XP Driver and the SLTA Link Manager software.

The PL-SLTA includes firmware that moves the upper layers of the LonTalk Protocol from the Neuron Chip within the PL-SLTA to a host processor. This firmware allows the PL-SLTA to be used by an LNS based application to send and receive LonTalk messages. The firmware in the PL-SLTA is loaded in ROM and cannot be reprogrammed.

Software Installation Procedure

Prior to installation, ensure that the computer is running the Windows Operating System and log in as Administrator. You must have a compression utility that opens and extracts ZIP files.

1. Open Internet Explorer and go to www.echelon.com and enter the Downloads Web page.
2. Complete the registration process and proceed to the download list.
3. Click LonWorks SLTA-10 Serial LonTalk Adapter Software for Windows 98/2000/XP.
4. Click the Proceed to Download button and accept the User Agreement.
5. Specify a folder to save the software and click Open after the download is complete. This will launch your ZIP utility program.
6. Double-click the slta10_Win.exe file and click Setup when prompted. Follow the instructions in the Setup Wizard. This will install the program onto your hard disk.
This chapter describes the SLTA Link Manager software, which establishes and configures local and remote links from the host PC to the PL-SLTA. A local link requires a direct cable connection from the host PC to the PL-SLTA. A remote link requires a pair of modems: one attached to the PL-SLTA and the other attached to the host PC. The SLTA Link Manager software controls a remote PL-SLTA via a pair of modems through Windows’ Telephony Application Programming Interface (TAPI) services under Windows 98/NT/2000/XP or later.

**Note:** Echelon no longer supports Windows 98 and Windows NT. The information presented in this user’s guide is provided for completeness and should be used for reference only.

The SLTA Link Manager determines when a standard driver open call in a host application requires dialing and handles these cases. Thus, the host application does not need to know if the network services interface is a local PL-SLTA or a remote PL-SLTA.

**Note:** Remote PL-SLTAs cannot be used with Windows NT 3.51 because Windows NT 3.51 does not include the 32-bit TAPI services used by the SLTA Link Manager software.
Software Overview

The SLTA Link Manager can monitor a modem line, answer an incoming phone call, associate the incoming call’s PL-SLTA (and therefore its network) with a LON device, and launch a pre-determined application for that particular network or PL-SLTA. Combined with an LNS host application, the SLTA Link Manager lets a LONWORKS network establish a connection to a remote PC through a pair of modems based on an event that occurs locally on the network.

The SLTA Link Manager provides a graphical user interface for creating, editing, and diagnosing “links”. Each link represents a particular PL-SLTA and its network. A link identifies several important aspects of the set-up, including the type of connection (a remote connection via modems or a direct local connection), the COM port, the PL-SLTA Remote Identifier (see page 34), the baud rate of the serial port on the PL-SLTA, and the dial-in password, if any. In addition, the link indicates if a security callback is required and may be associated with a host application. The link information is stored in a .slta file, located by default in the c:\lonworks\bin\slta folder.

The SLTA Link Manager can associate a link with a LON device name and then interface with the PL-SLTA driver. The SLTA Link Manager dials into the network from the PC host, providing the ability for applications with no knowledge of modems or phone numbers to run remotely through a pair of modems. The SLTA Link Manager application can be used to connect or disconnect from a remote PL-SLTA. In addition, the SLTA Link Manager has a simple interface that allows an application to run the PL-SLTA and perform a number of functions, such as dial a phone number or hang up. The SLTA Link Manager also includes many diagnostic functions.

Figure 11 shows the main startup screen of the SLTA Link Manager.
To dial into a network, you must first establish a communications line. Perform the following steps:

1. Choose **Dialing Preferences** (Line menu). Figure 12 displays the message that appears. This message only appears when telephony information has not been provided. This usually occurs if the computer has never been configured to use a modem.
2. Click on the **Dialing Properties** button to configure your settings. The **Location Information** dialog box appears. Enter your dialing information and click **Close**.

3. Enter your location information in the **Dialing Properties** dialog box (see Figure 14) and click **OK**.
Establishing a Communications Line for Calls Dialed Out to the PC

The next step is to select a line/modem to monitor for incoming calls. Figure 16 shows the Monitor Line window that is displayed when “Monitor for SLTA dial-in” is chosen from the Line menu. The SLTA Link Manager can monitor multiple phone lines or modems (for receiving incoming calls).
The list box displays the modems which have been set-up for use on this computer. The list can be created/modified by using the Modem Control Panel. Select the line/modem to be used for incoming calls, then click **OK**.

---

**Establishing Remote and Local Network Sites**

Choosing **Select/Action** from the **Link** menu displays a screen similar to the screen shown in **Figure 17**. Select “Local PL-SLTA” from the **Link Name** list and click **Edit**. The Link Description window appears and allows you to customize the connection—including changing it from Local to Remote, or modifying the name. **Figure 18** shows the default local setup.

![Figure 17. Completed SLTA Link Selection Window](image)

![Figure 18. Default SLTA Link Selection](image)

**SLTA Link Configuration Script Formats**

The SLTA Link Configuration dialog box can accept a script file for importing values to the dialog’s user interface. Configuration values can then be applied to the PL-SLTA by clicking **Apply**. See **Figure 19**.
Configuration items are processed on a line-by-line basis. All values are not required to be in the script file, and any duplicated assignments are overwritten.

Any line may start with a semi-colon, which is treated as a comment line. Line length is limited to 80 characters. Assignments are structured as follows:

```
keyword=value (note that there are no blank spaces).
```

Argument strings do not need to be quoted. Keywords are case-insensitive. Edit the script file as a simple text file with carriage returns and line feeds terminating each line.

### Table 8. Acceptable Assignments

<table>
<thead>
<tr>
<th>Keyword Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password=password string</td>
<td>Password: Up to 8 characters.</td>
</tr>
<tr>
<td>Callback=switch value</td>
<td>Callback enable: Enter a ‘1’ or a ‘0’ for enabled or disabled</td>
</tr>
<tr>
<td>HangupTimer=minutes value</td>
<td>Hangup timer: Enter a number between 0 and 255 (in minutes).</td>
</tr>
<tr>
<td>ModemInit=string</td>
<td>Modern initialization string: String length is limited by the available</td>
</tr>
<tr>
<td></td>
<td>EEPROM pool space in the SLTA.</td>
</tr>
<tr>
<td>ModemDialPrefix=string</td>
<td>Modern dial prefix: String length is limited by the available EEPROM pool</td>
</tr>
<tr>
<td></td>
<td>space in the SLTA.</td>
</tr>
<tr>
<td>DialDir1=string to</td>
<td>Dial Directories 1 through 5: String length is limited by the available</td>
</tr>
<tr>
<td>DialDir5=string</td>
<td>EEPROM pool space in the SLTA.</td>
</tr>
<tr>
<td>NVConnect=two digits</td>
<td>NV Auto-connect: Two digits to enable, or no digits to disable. The first</td>
</tr>
<tr>
<td></td>
<td>digit represents the starting dial directory number and the second digit</td>
</tr>
<tr>
<td></td>
<td>represents the last dial directory.</td>
</tr>
<tr>
<td>NSIConnect=two digits</td>
<td>NSI Auto-connect: Two digits to enable, or no digits to disable. The first</td>
</tr>
<tr>
<td></td>
<td>digit represents the starting dial directory number and the second digit</td>
</tr>
<tr>
<td></td>
<td>represents the last dial directory number.</td>
</tr>
<tr>
<td>ClearEEPPool=switch value</td>
<td>Either a ‘1’ or a ‘0’ to enable or disable clearing of the EE pool before</td>
</tr>
<tr>
<td></td>
<td>applying.</td>
</tr>
</tbody>
</table>
Example

; A PL-SLTA Configuration Script.
Password=Echelon
Callback=1
HangupTimer=5
GuardTimer=20
ModemInit=ATE0V0&C1&D2S0=1M0
ModemDialPrefix=ATDT
Dia1Dir1=14155557001
Dia1Dir2=17145557002
Dia1Dir3=18585557003
Dia1dir4=
Dia1dir5=
NVConnect=
NVConnect=1 2

Name of Link

The name of the link should be descriptive enough to clearly define the connection and remote location.

Remote Identifier

The Remote Identifier is used to identify a specific link when a dial-in to the computer occurs. It represents the remote PL-SLTA in a 12-byte string of characters or hexadecimal numbers. This value should match the value stored in the remote PL-SLTA. It can be entered as a string in single quotes, or as a series of hexadecimal numbers separated by dashes.

If this field is blank or all zeroes (00-00-00-00-00-00-00-00-00-00-00-00), then the Remote Identifier will be captured and stored here the next time this connection is made.

If this field is all FFs (FF-FF-FF-FF-FF-FF-FF-FF-FF-FF-FF-FF), then any Remote Identifier will be accepted. The identifier will not be stored on the PC. This is known as the wildcard Remote Identifier. Question marks (?) can be used in place of “Fs” as a wildcard Remote Identifiers. The SLTA Link Manager software translates “?” to all FFs.

The Update Identifier checkbox indicates that the SLTA Link Manager will read the remote identifier from PL-SLTA and the new value will be stored in the .s10 file the next time this link is used.

The SLTA Link Manager software lets you create two links with different names using the same Remote Identifier. However, when a network dials out to a PC with multiple links (each with the Remote Identifier) the link is automatically selected.

Link Type

The type of link specifies whether the PL-SLTA connects directly to the PC (Local), or if the PL-SLTA must be accessed remotely via a set of modems.

Configuring the Modem Line

Clicking Configure Line causes the selected modem’s property window to appear. The property window shows the available options of the modem driver such as volume control and dial tone detection.

TAPI services will handle the structuring of the call based on the Location Information (see Figure 19).
SLTA Password

The Password box lets you enter an eight-character password for a remote PL-SLTA. If entered, the password will be sent to the remote PL-SLTA when a connection is made.

**The password is not encrypted when stored on the host computer.**

Starting an Application

The SLTA Link Manager provides a space to enter the startup application for this link. This may be a full executable path name, or the name of an application that can be found in the system’s search path. You can also add command line arguments including special macros for link connection variables:

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%LINKNAME%</td>
<td>Expands to the name of the link, in quotes.</td>
</tr>
<tr>
<td>%DEVNAME%</td>
<td>Expands to the device name used by LONWORKS 32-bit applications to access the logical device. This serves the same purpose as %DOSNAME% does for DOS.</td>
</tr>
<tr>
<td>%NSSNAME%</td>
<td>Expands to the device name used by LONWORKS LNS application, for PL-SLTAs this will be “SLTALONr”.</td>
</tr>
<tr>
<td>%ID%</td>
<td>Expands to the remote identifier. This is expressed as either a quoted ASCII string, or as a series of hexadecimal numbers if the identifier contains non-ASCII data.</td>
</tr>
<tr>
<td>%RESULT%</td>
<td>Expands to an unquoted word that represents the success or fail reason of the connection.</td>
</tr>
</tbody>
</table>

The startup application will be launched when a dial-in occurs for this link, or optionally, when a manual connection is made to the link. It will **not** start up if the link is connected to due to an “auto-connect” case.

Enabling a Callback

If a remote PL-SLTA has callback enabled, then it will expect a callback command whenever someone dials into it. Check the Enable box if you need to enable the callback feature.

The callback command includes a directory index that points to a phone number stored in the remote PL-SLTA. If callback is enabled then one of the remote directory numbers (“Address 1” through “Address 5”) can be selected to call the host back.

See the *Characteristics of a Well-Designed System* section in this chapter for more information on how to implement callback.
Configuration

The SLTA-10 Configuration screen lets you set the security, timers, modem settings, dial directories, and auto-dial out configuration modes for your PL-SLTA. Table 9 lists these features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Password</td>
<td>The SLTA may be configured to accept incoming calls and connect the network to the host. Incoming callers may be required to provide a password before the SLTA will connect them to the network.</td>
</tr>
<tr>
<td>Enable Callback</td>
<td>Check this box if you need to enable callback. If a remote SLTA has callback enabled, then it will expect a callback command whenever someone dials into it. The callback command includes a directory index that points to a phone number stored in the remote SLTA.</td>
</tr>
<tr>
<td>Timers</td>
<td></td>
</tr>
<tr>
<td>Hangup Timer, minutes</td>
<td>Lets you enter a hangup timer value for the SLTA. This must be within the range of 0-255, for 0 (disabled) to 255 minutes. The hangup timer controls how many minutes of inactivity must pass before the SLTA hangs up (disconnects the modem).</td>
</tr>
<tr>
<td>Guard Time, seconds</td>
<td>Lets you enter a guard time value for the SLTA, in seconds. This must be within the range of 0-255. The guard timer controls how long to wait before attempting to dial the next number for the auto-connect case.</td>
</tr>
<tr>
<td>Modern Settings</td>
<td></td>
</tr>
<tr>
<td>Initialization String</td>
<td>Lets you enter a modem initialization/configuration string for the SLTA. This string is sent to the modem whenever the SLTA is reset and it is not currently connected. Special characters may be embedded in the string: ! Causes a carriage return to be sent. ~ Causes a 500ms pause. The carriage return is not required at the end of the string.</td>
</tr>
<tr>
<td>Dial Prefix</td>
<td>Lets you enter a modem dial prefix for the SLTA. This controls the characters sent to the modem before the actual phone number is sent. The default is “ATDT” for tone dialing. ! Causes a carriage return to be sent. ~ Causes a 500ms pause. The carriage return is not required at the end of the string.</td>
</tr>
<tr>
<td>Clear EEPROM Pool on Apply</td>
<td>Check this box to clear the SLTA’s EEPROM pool before applying the configuration. This will force the clearing and re-programming of the strings that use the EEPROM pool: The Modern Initialization and Dial Prefix strings, and the Dial Directories. If not checked, then only the SLTA’s strings that have been changed will be updated. Use this feature if you are increasing the size of one string and decreasing the size of another in one step.</td>
</tr>
<tr>
<td>Dial Directories</td>
<td>Lets you enter the dialout directory numbers, as strings. To choose which directory entry to edit, simply select one of the buttons above. ! Causes a carriage return to be sent. ~ Causes a 500ms pause. The carriage return is not required at the end of the string.</td>
</tr>
<tr>
<td>Auto-dialout Configuration</td>
<td>Check this box to enable the Network Variable update auto-connect feature. When enabled, the SLTA will attempt to connect to a remote host whenever the SLTA is not connected and a non-broadcast host-bound NV update message is pending.</td>
</tr>
</tbody>
</table>

Using the Windows Driver and SLTALink Manager with PL-SLTA 35
The SLTA starts with the directory number first specified on the right. If this attempt fails, then the next directory number is used, until the last directory number has been used. To use a single number simply specify the same directory number in both fields.

**NSI Connect**
Check this box to enable the Network Services (NSI) auto-connect feature. When enabled, the SLTA will attempt to connect to a remote host whenever the SLTA is not connected and a host-bound NSI message is pending.

The SLTA starts with the directory number first specified on the right. If this attempt fails then the next directory number is used, until the last directory number has been used. To use a single number simply specify the same directory number in both fields.

### Diagnostics

A number of diagnostic and testing services are provided via the **Diagnostic** screen, accessed through the **Devices** menu (see Figure 20). The **Test** button retrieves status and error counts from the PL-SLTA. The **Service** button causes the PL-SLTA to broadcast a service pin message on the network. The **Reset** button causes a reset of the Neuron Chip in the PL-SLTA, but does not clear the Neuron Chip’s system image. Some buttons are left for future releases of additional features.

![Figure 20. Diagnostic Screen](image)

### The SLTA Link Manager Programmatic Interface

**SLTALINK.EXE** executes as a single process. If you try and run another copy of it, it will defer itself to the original process. However, command line arguments may be passed in this manner which will direct the existing process to perform certain types of tasks. These command line options are:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Linkname&quot;</td>
<td>A link name is required for all actions. If the link name alone is passed then that link will be connected. It must be enclosed in double quotes, since the link name can have embedded spaces.</td>
</tr>
<tr>
<td>C:\lonworks\bin\sltalink.exe &quot;Remote&quot;</td>
<td></td>
</tr>
<tr>
<td>/D</td>
<td>This causes the specified link to be disconnected.</td>
</tr>
<tr>
<td>C:\lonworks\bin\sltalink.exe &quot;Remote&quot; /D</td>
<td></td>
</tr>
<tr>
<td>/# &quot;number&quot;</td>
<td>This overrides the phone number for the link. If you have checked the “Use Country Code and Area Code” option for this link, then the number must be in the ‘canonical’ format without a ‘+’ sign and enclosed in double quotes. A canonical number is defined as a country code followed by a space, followed by an area code or city code enclosed in parenthesis, followed by a space and the rest of the phone number. TAPI will take this and decide how to translate it before performing the dial-out, based on your dialing preferences. Even if the call is local you should include the area code / city code. This</td>
</tr>
<tr>
<td>C:\lonworks\bin\sltalink.exe &quot;Remote&quot; /# &quot;1234567890&quot;</td>
<td></td>
</tr>
</tbody>
</table>
is the same format as the number that appears in the link selection dialog under the Port/Number column. Example: "1 (800) 555-1213"

If you have not checked the "Use Country Code and Area Code" option then the number will be used un-translated for dial-out. In this case you probably won’t specify a canonical number.

C:\lonworks\bin\sltalink.exe "Remote" /# "1 (650) 555 1213"

/P “passwd”

This overrides the password for the link. It must be enclosed in double quotes.

C:\lonworks\bin\sltalink.exe "Remote" /P "passwd"

[Note: Always include a space between each element of the command line arguments.]

Characteristics of a Well-Designed System

To build a reliable system design using the SLTA Link Manager, you must effectively use Call Initiation, Call Termination and Monitoring.

Call Initiation

The four scenarios for call initiation are: dial-in to the network only, dial-out to the remote PC only, dial-in / dial-out, and callback.

Dial-In to the Network Only

In the most straightforward case, a user launches an application that opens the driver, which is associated with a particular link. The SLTA Link Manager dials the phone number in the link (or the phone number the application passes down to the SLTA Link Manager perhaps to a generic link) and establishes the connection. Similarly, the user could select the link from within the SLTA Link Manager and connect manually. At this point, either the SLTA Link Manager would launch the pre-determined application from the information stored in the link file or the user could manually launch the application. In all of these cases, the user is assumed to initiate the call. The user could be a human operator or another application that initiates a dial-in based on a clock, for example.

For the dial-in only scenario, the system strategy issues primarily have to do with associating the link or phone number with the application. Where there are only one or two links, this is very easy. When one PC host can be connected to many different networks, we offer two standard solutions. The first is to have the user navigate the SLTA Link Manager’s interface. Under the Link menu, the Select item lists approximately 40 links (of the 1000 possible). The second solution is a monitoring application that programmatically interacts with the SLTA Link Manager to send down the appropriate phone number, perhaps to a generic outgoing link.

Dial-Out to the Remote PC Only

The three common approaches for initiating a dial-out are: sending a network variable update to the PL-SLTA, sending an AddMyNSI message to the PL-SLTA, or sending an explicit message from a “Helper/Dialer” node on the network.

In the first case, the remote host application needs to have an explicitly-bound input network variable and the PL-SLTA’s NSI mode EEPROM must be configured correctly. See the configuration section in this chapter. See also Call Termination below.
In the second case, the remote host PC is assumed to have the NSS engine and a second NSI on the network (perhaps a service technician with a laptop running a PCC-10 card) is required to send the AddMyNSI message. Also, PL-SLTA’s NSI mode EEPROM must be configured correctly.

In the third case, a custom Neuron Chip application must be written.

All three cases could be used with the same PL-SLTA.

In the dial-out only case, besides the call initiation, the SLTA Link Manager must be able to launch the appropriate application – with the correct database and device driver name. One system solution is to create a separate link for each PL-SLTA. Each link then stores the Remote Identifier of its PL-SLTA after the first connection. Upon connection, the appropriate application and command line arguments stored in the link are launched. A second approach is to create a generic link that uses the wild card as a Remote Identifier to launch a generic application using command line arguments to specify the appropriate network or database and device driver name. These arguments are available and described above under Invoking an Application.

Note: If the device driver information used in the application does not match the device driver name being used by the link, the newly launched application can open a second device driver – which may result in an attempt to dial-in to the network. Since the modem is still presumably in use with the original dial-out call to the host PC, the second call will fail. The result is a system-level failure.

Dial-In / Dial-Out

These scenarios are variations of the above examples. However, not all dial-in strategies can co-exist with all dial-out strategies. For example, if the dial-out strategy involves having the SLTA Link Manager match the incoming call to the wild card Remote Identifier and if the dial-in strategy requires a separate link for each PL-SLTA Remote Identifier, then it is possible a call initiated from the network will be received by the SLTA Link Manager and will be matched with the link created for the dial-in case. The correct application may not be launched and a system-level failure may occur.

Callback

The PL-SLTA callback functionality works as follows: a call is initiated from a remote PC to a PL-SLTA on a network, which must have its NSI mode EEPROM configured to require callback. The PL-SLTA answers; the remote PC identifies to the PL-SLTA one of the PL-SLTAs directory entries to use for the callback. A PL-SLTA configured to require a callback will not accept any other direction from the host at this time. The original call is terminated, and the PL-SLTA calls the phone number indicated in its directory. Note: this does NOT need to be the phone number of the original remote PC that initiated the call. Typically, the SLTA Link Manager on the remote host dialed answers the call and launches the appropriate application.

Several possible system-level failures include:

- The original remote host expects to receive the callback, but the directory index reflects the phone number of another remote host.

- The original remote host application opens an LNS database and initiates the first call. The callback is directed back to the original remote host PC. The SLTA Link Manager on this PC receives the callback just like any other normal dial-out call and launches the application contained in the link. At this point there may be two copies of the application open. Depending on sharing configuration, the second application may fail because the appropriate LNS database is already opened.

To prevent these failures, Echelon recommends that the initial call should either be a manual connection from within the SLTA Link Manager or the initial call should originate from a “dummy” application that terminates itself without opening the LNS databases.
Call Termination

The four scenarios for call termination include: termination of the host application, application controlled hang-up, a manual disconnect in the SLTA Link Manager, and time-out. In all of these cases, the important system-level issues involve making sure that the termination strategy is compatible with the call initiation strategy.

The behavior of an application at termination is not always known. By default, applications based on the Object Server of LNS 1.5 and higher should exhibit two types of behavior. First, if the interface adapter is a PL-SLTA in NSI modem using modems and there are explicitly bound network variables to the host application, then at the LNS application’s termination, the host network variables and their connections should not be removed from the LNS database. This facilitates using the first case described under Dial-Out to the Remote PC Only where a network variable update addressed to the PL-SLTA`s NSI mode results in a call being initiated. Second, if the interface adapter is not a PL-SLTA in NSI modem using modems or there are no explicitly bound network variables to the host application, then at the LNS application’s termination, any host network variables and their connections are removed from the LNS database and the other nodes in the network. In addition, if the interface does not host the LNS database, the interface is deconfigured after the LNS application closes.

The second behavior described would be desirable if multiple remote host PCs need to dial-in to the same network. As long as no explicitly-bound network variables were left when the host applications terminated, then several remote PCs could share one PL-SLTA. Note that this assumes that the LNS Server exists somewhere on the network and is not located on one of the remote PCs sharing the single PL-SLTA.

Both an application controlled hang-up (using the SLTALINK.EXE programmatic interface) and a manual disconnect in the SLTA Link Manager will terminate the phone call; however, neither results in the termination of the host application. In these scenarios, the host application remains running as the call can be re-established by the host application itself, by a manual connect in the SLTA Link Manager, or a dial-out initiated on the network. The disadvantage of these system solutions is that they do not scale well to monitoring multiple networks on one PC. The result is many applications continually and concurrently running on the same PC. Also, one possible system failure to avoid is that the SLTA Link Manager settings may result in multiple copies of the host applications.

By default, the PL-SLTA NSI mode will terminate a phone call after three minutes with no modem traffic. As with the application-controlled hang-up and the manual disconnect in the SLTA Link Manager, this scenario does not result in the application terminating. This scenario therefore carries the same advantages and disadvantages as those described in the previous paragraph.

Monitoring: Application Termination Strategy

Three strategies are used to terminate the remote LNS monitoring application.

The first strategy requires the user to terminate the application manually. No special software is used for this case. When dialing in, the user can immediately shut down the application. When dialing out, a user can monitor every network call. The user should be aware if the network is sending alarms.

The second strategy requires a node on the network to send a network variable update to the host when the network no longer requires the host application to be running. When the host application receives this network variable, it should enter a shut-down routine. This approach automates the process and places the responsibility on the network to determine when the host application is needed and requires a hook in the monitoring application, but provides a level of automation when dial-out is used.
In the third strategy, the remote host application terminates itself after a given time period or after completing a series of tasks. In general, Echelon does not recommend using this method. However, this may be useful for some applications. For example, a connection may be established through dial-in or dial-out based on a timer. The host application could then take a series of measurements from the network, log them to a file, and then terminate itself.

**Monitoring: Missing Messages After a Dial-Out**

In general, the message that triggers a dial-out from the PL-SLTA on the network to a remote PC host is lost.

When a network variable update is sent to the PL-SLTA and no phone connection is currently up and running, the PL-SLTA is typically configured to dial-out to the host. On the remote PC host, the SLTA Link Manager answers the incoming call. The SLTA Link Manager then reads the Remote Identifier in the PL-SLTA and searches through all the .s10 files for a match. The SLTA Link Manager then typically launches the application listed in the link with optional command line arguments available, for example, to open the correct LNS database with the correct device name. Opening the LNS application results in the buffers of the PL-SLTA being over-written; thus, the original message is lost. Since this message network variable update is important enough to warrant establishing a connection to the host, a system strategy is required so that this data reaches the host.

The two basic system strategies are:

1. Have the node on the network continue to send out the network variable updates until the host application sends a message to the node telling it to stop.
2. Have the host application seek out the information upon being launched.

The first approach places the burden on nodes in the network; the second approach places the burden on the host application. The first is more direct and will likely result in the information getting to the host more quickly. The second approach has the primary advantage that no special Neuron Chip application code is required; also, since the call initiation and host application launch may require extra time, the time it takes the host application to poll several network variables is not significant. In the best scenario, the host application would first check with a datalogging node on the network that records alarms and also the system state, mode, or health information.

**Monitoring: LNS Application Design Issues**

A well-designed LNS monitoring application using the PL-SLTA through modems should use the correct version and layer of LNS, should handle initialization of the application with the correct LNS database and device driver name, and should have a phone call session termination and an application termination strategy.

The remote monitoring application is based on LNS Object Server of LNS 1.5 or higher. Applications based on the LNS 1.0 and LNS 1.01 do not allow for certain dial-out scenarios. Applications written at the NSS layer are not supported for use with the PL-SLTA across a pair of modems.

The remote monitoring application should open the appropriate LNS and device driver. Specifically, the mapping of the Remote ID to the LNS database should be handled by the application. Echelon recommends using the command line arguments with the information available from the SLTA Link Manager. Note that LNS capacity keys may need to be hard-coded in the monitoring application.

Finally, the LNS application needs to implement a termination strategy that meets the needs of the application and the system.
Good Practices/Schemes that Work

Use these guidelines to avoid the system-level failures described above:

- When expecting the PL-SLTA to dial-out from the network to a remote PC, dedicate one PL-SLTA to dial-out and always have that PL-SLTA connect to the same remote PC. Make sure that the NSI EEPROM is configured correctly. See Figure 21.

- If an LNS-based application is connected to the network through modems, dedicate one PL-SLTA in the network to handle dial-out and use the PC that has the LNS server to dial-in. Make sure that the PL-SLTA's EEPROM is configured correctly. See Figure 22.

- Several PCs can share one PL-SLTA as long as the calls are all initiated on the remote PC hosts (i.e., dial-in only) and each remote LNS application removes the bound connections to its host before terminating. See Figure 23.
Figure 21. Dedicated PL-SLTA Using Dial-out
Figure 22. Dedicated PL-SLTA Hosting the NSS
Figure 23. Shared PL-SLTA Using Dial-in
This chapter describes how the PL-SLTA operates when a remote host computer is connected using a pair of modems to the PL-SLTA. In this set-up, any node on the network may request the PL-SLTA to initiate a dial-out operation to connect to the host. In addition, the PL-SLTA itself may be configured to initiate a dial-out operation to connect to the host. The PL-SLTA can store a telephone directory of commonly called numbers, as well as accept commands to dial any other number.

The PL-SLTA may also be configured to accept incoming calls and connect the network to the host. Incoming callers may optionally be required to provide a password before the PL-SLTA will connect them to the network.

See Chapter 6, *Using the Windows Drivers and SLTA Link Manager with PL-SLTA*, for configuring the PL-SLTA through the SLTA Link Manager application.
Overview

The PL-SLTA network interface may be attached to the host processor using modems and the switched telephone network. Figure 24 illustrates this option.

Figure 24. Using the PL-SLTA with Modems

When the PL-SLTA is configured for modem support, it will respond to special network management messages that cause it to communicate with the attached modem. The modem must support the Hayes standard AT command set that allows it to dial-out, auto-answer incoming calls, and set various parameters.

Once a connection is established, the host computer communicates transparently with the network as though the PL-SLTA was attached locally.

The PL-SLTA can be configured to initiate the connection with the host PC when LNS network management messages and/or network variable updates are addressed to the PL-SLTA. Alternatively, another node on the local network can command the PL-SLTA to initiate the connection.
In order to support the modem functions, the PL-SLTA Switch2/CFG2 input must be set to the Remote Host state (UP position). See Chapter 4 for details of the configuration inputs.

**PL-SLTA Connection States**

When the PL-SLTA is operating in the remote host mode, several internal states (or connection states) will control its behavior:

- The **IDLE** state is entered after power-up reset. In this state, any uplink bound messages are ignored since the PL-SLTA is not connected to a host. The IDLE state is also entered whenever the telephone connection is broken and the modem drops the DCD (Data Carrier Detect) line.

- The **CALL_IN_PROCESS** state is entered once a connection is initiated by a node on the network connected to the PL-SLTA. In this state, uplink traffic is still discarded while the PL-SLTA monitors the modem for connection completion or connection failed events.

- The **CONNECTED** state is entered once a connection is established. The normal network interface protocol resumes between the PL-SLTA and the remote host. This state may be entered as a result of a node on the local network initiating a call, or as a result of a remote host calling up this PL-SLTA.

- The **FAILED** state is entered if the connection process fails. This state is operationally the same as IDLE.

The connection state of the PL-SLTA is preserved across software resets, allowing normal network management resets to occur without breaking the connection. The PL-SLTA will not preserve the connection state after it has been through a power reset.

**Command Set Assumptions**

The PL-SLTA uses the following strings received from the modem to interpret the connection state. These strings are consistent with all Hayes AT compatible modems operating in the word response mode (alphabetic responses). [CR] is the hex 0D character.

CONNECT [any text] [CR]
BUSY [CR]
VOICE [CR]
NO [any text][CR]
ERROR [CR]

The “CONNECT” string may be, and typically is, followed by other informative text, such as connection serial bit rate or error correction methods in use. These other four states indicate a failure to make the connection.
Translated Characters

All strings that are sent specifically to the modem (as commands) by the PL-SLTA are scanned for certain characters by the PL-SLTA firmware. These characters are then translated into specific functions or characters unless they are preceded by a backslash ("\"). The characters are:

~ The tilde causes the PL-SLTA to pause 500ms before sending the next character to the modem. The tilde itself is not sent.

! The exclamation point causes the PL-SLTA to send a carriage return (0x0D) to the modem. The exclamation point itself is not sent.

The tilde is provided for users familiar with existing modem packages. It can be used, for example, in the command string, which causes the modem to hang-up. When dialing out, Hayes AT-compatible modems use a comma to insert delays between portions of a dial command; the comma should continue to be used for dialing out.

DTE Connections

In addition to the basic three wire connections—Transmit Data (TXD), Receive Data (RXD), and Signal Ground—two additional signals must be connected: Data Carrier Detect (DCD) and Data Terminal Ready (DTR). The modem must also be configured to use these signals. DCD is used by the PL-SLTA to determine that a connection has been made and DTR is used to terminate a connection by hanging up. Note that many modems default to ignore these two signals and must be configured to enable them. The following AT command enables these two signals on many modems: AT&C1&D2[CR]

Network Management Messages

Network management messages are used to configure the network operation, as opposed to delivering application data. All LONWORKS nodes respond to standard network management messages as described in Appendix B of the Neuron Chip Data Book. For nodes based on the PL-SLTA, additional network management messages are defined to configure and control the PL-SLTA. The message codes for network management messages are reserved, preventing possible conflicts with any applications. The additional PL-SLTA network management messages use the reserved message code 7D hex. The first data byte is used as a sub-code, and is 01 hex to indicate a PL-SLTA function. The second data byte is a specific application command code. Table 10 summarizes the network management messages specific to the PL-SLTA.
Table 10. PL-SLTA Network Management Messages

<table>
<thead>
<tr>
<th>Message Code</th>
<th>Message Sub-code</th>
<th>Application Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x7D</td>
<td>1</td>
<td>1</td>
<td>Product Query</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>2</td>
<td>Send Modem String</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>3</td>
<td>Modern Status Readback</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>4</td>
<td>Modem Response Query</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>5</td>
<td>Connection Status Query</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>6</td>
<td>Install Directory Entry</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>7</td>
<td>Dial From Directory</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>8</td>
<td>Hang-up</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>9</td>
<td>Install Password</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>10</td>
<td>Install Modem Configuration String</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>11</td>
<td>Install Hangup String</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>12</td>
<td>Install Dial Prefix</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>13</td>
<td>Install Hangup Timer</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>14</td>
<td>Configure Modem</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>15</td>
<td>Request/Release Modem</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>16</td>
<td>Clear EEPROM Pool</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>17</td>
<td>Install NV Connect</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>18</td>
<td>Install NSI Connect</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>19</td>
<td>Install Callback Enable</td>
</tr>
<tr>
<td>0x7D</td>
<td>1</td>
<td>20</td>
<td>Report SLTA EEPROM contents</td>
</tr>
</tbody>
</table>

These network management messages may be sent from any node on the network to the PL-SLTA. The application node sends modem-control network management messages to the PL-SLTA using explicit messaging. See Chapter 4 of the *Neuron C Programmer’s Guide* for details on explicit messaging. The message should be delivered using request/response service, and the message code for modem control messages is always 7D hex. The data portion of the message always begins with a sub-code value of 1, indicating that the message is addressed to a PL-SLTA, followed by an application command byte indicating which modem-control command this is. This is followed by parameters specific to the application command. These messages may be sent by a Neuron C-based node, a node based on the Microprocessor Interface Program (MIP), or a node based on another PL-SLTA.

See the supplied Neuron C program DIALOUT.NC for an example of an application that sends a message to a PL-SLTA to cause it to dial-out using an attached modem. The example GIZSETUP.NC sends messages to a PL-SLTA to configure its modem strings using a Gizmo I/O module as the user interface. A node can send these messages using either implicit or explicit addressing. When implicit addressing is used, a network management tool binds the output message tag in the node wishing to send the message to the PL-SLTA as the destination. When explicit addressing is used, the node sending the message must explicitly insert the destination address in the outgoing message.
Note that the PL-SLTA will be unable to receive any messages, including these network management messages, if network communications are disabled, i.e., it is in the FLUSH state. This will normally be the case if it is not connected to a host. Therefore, to dial-out and connect to a host, the PL-SLTA must be configured to initialize with network communications enabled, i.e., the NORMAL state, with the CFG1 input. See Chapter 4 for more details.

The messages should be sent with request/response service, and the response code should be checked to see if the message was executed correctly - it will be 0x3D if there was no error, and 0x1D if there was. Possible failure conditions are noted under each message. Some of these network management messages return data to the sender in the response structure as noted below.

Note that some of these messages cause one or more bytes of EEPROM in the PL-SLTA to be written. Each byte of EEPROM takes 20ms to write, and the response is not sent by the PL-SLTA until after the command is executed. This time should be taken into account when setting the transaction timer (tx_timer) in the message tag connection for implicit addressing, or in the destination address for explicit addressing. If the LonBuilder or LonManager API binder is used to create the connection, the transaction timer may be explicitly specified.

Example:

```c
msg_tag SLTA_tag;
when ( ... ) {
    msg_out.code = 0x7D; // network mgmt msg code
    msg_out.tag = SLTA_tag;
    msg_out.service = REQUEST;
    msg_out.data[0] = 1; // sub-code for PL-SLTA
    msg_out.data[1] = app_command; // specific command
    msg_out.data[2] = ..... // additional parameters
    msg_send();
}
when (msg_fails(SLTA_tag)) { // PL-SLTA did not respond to the message
    ...
}
when (msg_succeeds) { ... }
when (resp_arrives(SLTA_tag) { // PL-SLTA did respond to the message
    if (resp_in.code == 0x3D)
        // command executed successfully
        ...
}
```

Certain PL-SLTA network management messages, which are designed to send a command string directly to the modem, will fail if the connection status is CONNECTED. This applies to the following messages: Send Modem String, Dial From Directory, and Configure Modem. The telephone connection must be broken and the PL-SLTA returned to the IDLE state for these messages to be issued.

Many of these network management messages, for example the “Install...” messages, may be sent to the PL-SLTA from the host computer via the telephone link if the PL-SLTA is in the CONNECTED state. In this case, they should be sent using the niNETMGMT network interface command so that they are addressed to the PL-SLTA itself.

Structures are defined in the file SLTA_ANM.H for each of the application commands, and include the sub_code and app_command fields, as well as any additional parameters for the specific application.
command. If the response contains data other than the response code, a structure for the returned data is also defined.

EEPROM String Pool Management

The PL-SLTA’s EEPROM is used to store a number of varying-length configurable strings, which are used to control the modem. Sending network management messages to the PL-SLTA, either from the modem side or the network side, can set these strings. Since the strings vary in length and EEPROM space is limited, a pool management system is used to optimize EEPROM.

The EEPROM pool consists of the following strings:

- Up to five dial-out directory entries. These may be used by index to initiate a dial-out connection, and may contain any combination of AT commands and numbers.
- One modem initialization string. This string is used to initialize the modem as required.
- One dial-out prefix string. This string is sent as a prefix to any dial-out operation to specify the modem dial command and to indicate whether tone or pulse dialing should be used.
- An 8-byte dial-in password string.
- A 1-byte code that enables an auto dial-out on a network variable message.
- A 1-byte code that enables an auto dial-out on a NSI message.
- A 1-byte code that enables callback.

The EEPROM pool manages EEPROM storage and allocation for these strings. This allows flexible utilization of the PL-SLTA’s EEPROM space. The pool consists of 8 blocks, each with 12 bytes of data storage space. A string occupies one or more blocks.

The EEPROM pool does not require the exclamation point (translated to a carriage return) in the dial directories. In addition, the EEPROM pool does not require a null terminator on exact sector size strings.

The network management functions that install these strings allow incremental EEPROM writes, and include a total_size field. Incremental writes allow long strings to be installed without the requirement of large buffer sizes on the PL-SLTA. If the total_size field is greater than the amount of EEPROM storage space available then the network management message response will indicate a failed status. In all cases, the strings should include a null terminator.

The offset field in the install messages indicates the starting point in the complete string of the current string piece to be installed. The first byte of the complete string is at offset zero. If the offset is such that the string piece would not fit in the allocated total size for the string, the total size for that string is automatically extended, if space is available.

All pieces of a string should specify the same value for total_size. Otherwise it means that you are starting over with a new string.

Once a particular string has been installed in the pool, it may be deleted by re-installing it with a total size of 0. Alternatively, the Clear EEPROM Pool message may be issued to clear the whole EEPROM string pool.
Product Query

This message may be used to check the type of interface product that the node is running. The value returned for the PL-SLTA is a '1'. Application nodes will respond with a failed code of 0x1D returned for this or any message where the request message code is 0x7D.

```c
typedef struct {
    byte sub_code;   // always #1
    byte app_command;  // value = 1
} ANM_product_query_request;
typedef struct {
    byte product;   // for PL-SLTA, = 1
} ANM_product_query_response;
```

Send Modem String

This message is typically used to send AT command character strings to the modem from another node on the local network. This provides full control of the modem and its internal control registers and settings. Normal use of this feature is to dial-out to a number that is not in the telephone directory of the PL-SLTA, or to change or set modem control registers and options. This allows any application node on the network to control the modem. The message definition is:

```c
typedef enum {
    NO_CHANGE  = 0,
    DIAL_OUT   = 1,
    HANGUP_DIAL_OUT  = 2
} STR_mode;
typedef struct {
    byte sub_code;  // always #1
    byte app_command;  // value = 2
    STR_mode mode;
    char  modem_string[];
} ANM_send_str_request;
```

The length of `modem_string` is limited to the application and network buffer sizes within the PL-SLTA node and the node which is communicating with it. The PL-SLTA as shipped has buffers sizes allowing for a maximum of 46 characters in the modem string sent with this message. The string must be null terminated.

If a large string needs to be sent to the PL-SLTA, use a series of these requests with a single carriage return in the last string. Note that many modems have a limited input buffer size, typically 32 to 80 bytes.

The `mode` parameter is used to control the connection state of the PL-SLTA. The values for this parameter are:

0  Make no change to the PL-SLTA's modem connection state. Send only if not CONNECTED. Otherwise, respond with a failed status.

1  Initiate a dial-out connection. If the PL-SLTA is already currently connected, preserve that connection, and ignore the message. The PL-SLTA's connection status changes from IDLE to CALL_IN_PROCESS, unless the connection is already made, in which case the state stays at CONNECTED. The dial-out prefix is sent first.

2  Same as '1', but disconnect (hang-up) if currently connected before initiating the new connection.
Modem Response Query

ASCII strings received from the modem when the PL-SLTA is not connected to a host will be buffered and may be read back by a node via this message. The storage for this string is limited, and so it will contain only the first 16 characters received from the modem in the disconnected state. Executing the Send Modem String message will always clear this buffered string. It will not be cleared when the modem disconnects, aiding in diagnosis of connection problems.

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 4
    byte max_length; // limits the size of the response.
} ANM_modem_response_query_request;

typedef struct {
    char response[]; // null terminated string
} ANM_modem_response_query_response;

Connection Status Query

The PL-SLTA’s connection status may be polled with this message. Most modems may be configured with various time-outs for the different stages of establishing a connection. Consult your modem’s documentation for details.

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 5
} ANM_connect_state_request;

typedef enum {
    IDLE   = 0,
    FAILED = 1,
    CALL_IN_PROCESS = 2,
    CONNECTED = 3
} CONN_state;

typedef struct {
    CONN_state connection_state;
} ANM_connect_state_response;

Install Directory Entry

This message stores a dial-out directory entry in the PL-SLTA’s EEPROM string pool. Up to 5 dial-out entries can be stored in the PL-SLTA EEPROM pool. The entries are numbered 0 to 4 as specified by the dir_num field.

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 6
    byte dir_num;  // value = 0-4
    byte total_size; // of the data string
    byte offset;  // for piecemeal writes
    char dir_string[];
} ANM_install_dir_request;

See the section on EEPROM String Pool Management above for details on the EEPROM string pool.
### Dial From Directory

Using one of the directory entries specified by `dir_num`, dial-out to a remote host and establish a connection. Based on the `mode` argument, the directory string may be sent to the modem and the PL-SLTA enters the `CALL_IN_PROCESS` state. Nodes can periodically check connection progress using the `Connection Status Query` message. The `mode` parameter is used to control the connection state of the PL-SLTA. The values for this parameter are:

1. Initiate a dial-out connection. If the PL-SLTA is currently connected, preserve that connection, and ignore the message. The PL-SLTA’s connection status changes from `IDLE` to `CALL_IN_PROCESS`, unless the connection is already made, in which case the state stays at `CONNECTED`. The dial-out prefix is sent first.

2. Same as “1”, but disconnect (hang-up) if currently connected before initiating the new connection.

```c
typedef enum {
    DIAL_OUT   = 1,
    HANGUP_DIAL_OUT = 2
} STR_mode;
```

```c
typedef struct {
    byte  sub_code;   // always #1
    byte  app_command; // value = 7
    STR_mode mode;
    byte  dir_num;   // value = 0-7 for MIP; 0-4 for NSI
} ANM_dial_dir_request;
```

If the directory entry does not exist, the response to this message will indicate a failure. A successful response to this message indicates that the PL-SLTA has sent the dial-out command to the modem, not that the modem has successfully dialed. The `Connection Status Query` message should be sent to determine whether a successful connection has been established.

### Hang-up

This message causes the PL-SLTA to pulse the EIA-232 DTR signal (Data Terminal Ready) low for 500ms. If the DCD signal is still ON following this step, then the PL-SLTA will send the hang-up string (see below) if it is not a null string. This will terminate the connection, and the PL-SLTA enters the `IDLE` state. The response will not be sent until this process is complete. Therefore, the transaction timer for this message should be set to at least 768ms if it is sent from a node on the network. If this message is sent from a remote host, no response should be expected, since the connection will have been broken before the response can be sent.

```c
typedef struct {
    byte  sub_code;  // always #1
    byte  app_command; // value = 8
    boolean if_config;
} ANM_hangup_request;
```

If the PL-SLTA is forced to send the hang-up string, and this string does not exist in the EEPROM configuration, the response to this message will indicate a failure. If the `if_config` byte of this message is non-zero, the PL-SLTA sends the configuration string to the modem following a hang-up. This provides a remote host with the ability to dial up a remote PL-SLTA, change the configuration string, hang-up, and reconfigure the modem.
Install Password

This message stores a dial-in password in EEPROM. The default setting for this string is a null string, which results in no password requirement by the PL-SLTA. Any node on the network may change the password, but an external host must have already connected and used the existing password in order to change it, unless it was blank. Any installed PL-SLTA should use a password, otherwise an unauthorized user might reset the password.

The password string is limited to 8 characters, and may be any sequence of non-zero eight bit values. The string must be null-terminated.

typedef struct {
    byte sub_code;     // always #1
    byte app_command;  // value = 9
    char passwd_string[8];
} ANM_install_passwd_request;

If a password is installed in the PL-SLTA, the host must issue the \texttt{niPASSWD} network interface command to the PL-SLTA after the connection is established. The code for this command is \texttt{0xE4}, and it should be followed by up to eight password characters, null terminated. For details on sending network interface commands, see the \textit{LONWORKS Host Application Programmer's Guide}.

If modem support is selected, a password is present in the PL-SLTA's configuration, and the connection state changes from \texttt{IDLE} to \texttt{CONNECTED}, the PL-SLTA will not process any other network interface commands until the correct password is sent downlink. If the password is correct, the PL-SLTA will respond with the \texttt{niACK} code. Otherwise the PL-SLTA will respond with the \texttt{niNACK} code.

Install Modem Configuration String

Whenever the PL-SLTA is either powered or reset, the connection state is not \texttt{CONNECTED}, or is commanded to by the \textit{Configure Modem} message, the PL-SLTA will send its modem configuration string to the modem. This string is stored in EEPROM, and may be changed by this message. The default setting for this string is "ATE0V0S0=1&C1&D2". The commands specified in this string are:

- \texttt{E0} is the AT command to disable local echo of characters received by the modem.
- \texttt{VO} is the AT command to enable numeric response codes from the modem.
- \texttt{&C1} is the AT command to enable the Data Carrier Detect (DCD) output of the modem, which is active when the modem detects the carrier signal of another modem on the line.
- \texttt{&D2} is the AT command to enable the Data Terminal Ready (DTR) input of the modem. The modem will hang-up, enter command state, and disable auto-answer when it detects an on-to-off transition of the DTR input.
- \texttt{S0=1} is the AT command to set register S0 to 1, meaning that the modem should auto-answer incoming calls on the first ring. This option should be used if remote hosts will be dialing in to the PL-SLTA. Use "\texttt{S0=0}" if the PL-SLTA will only be used for dialing out to remote hosts.

See \textit{Modem Compatibility} later in this chapter for additional sample modem configuration strings.

typedef struct {
    byte sub_code;     // always #1
    byte app_command;  // value = 10
} ANM_install_config_string;
Install Dial Prefix

The default setting for this string is “ATDT”. This string is sent as a prefix for any dial-out operations. This particular sequence instructs the modem to dial using Touch-Tone (DTMF) signaling. If pulse dialing is required, the prefix should be set to “ATDP”.

```c
typedef struct {
    byte sub_code; // always #1
    byte app_command; // value = 12
    byte total_size; // of the data string
    byte offset; // for incremental writes
    char dpre_string[];
} ANM_install_dpre_request;
```

This message uses the EEPROM pool to store the dial prefix. See the section on EEPROM String Pool Management above for details.

Install Hangup Timer

The hangup timer is controlled by an eight bit value in EEPROM. The default setting for this value is zero, which results in no hangup timer action. If the hangup timer is a non-zero value, the PL-SLTA will hang-up and break a connection (if in the CONNECTED state) when the number of minutes specified by timer_value have elapsed and no uplink or downlink activity has occurred. The default value for this timeout is zero, meaning that the phone connection will remain active indefinitely, even if there is no activity on the link.

```c
typedef struct {
    byte sub_code; // always #1
    byte app_command; // value = 13
    byte timer_value;
} ANM_install_hangt_request;
```

Configure Modem

This message will cause the PL-SLTA to send its Modem Configuration String to the modem provided it is not in the CONNECTED state. See the Install Modem Configuration String message for further details. If the modem is in the CONNECTED state, the failed status will be returned.

```c
typedef struct {
    byte sub_code; // always #1
    byte app_command; // value = 14
} ANM_modem_config_request;
```

Request/Release PL-SLTA

This message may be used to grant ownership access of the PL-SLTA to any node on the local network. In a design where there may be more than one network node that wishes to control the PL-SLTA’s connection...
states, Request/Release provides a method of managing this control. In this case a node will request the PL-SLTA, and the response to this request will be successful (code 0x3D) if access was granted, or failure (code 0x1D) if it was denied. At the end of such a session the node which was granted ownership must release the PL-SLTA, allowing other nodes access to it. This process only works if all nodes employ this mechanism. The request state is not preserved across resets of the PL-SLTA.

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 15
    boolean req_rel; // TRUE if request, FALSE if release
} ANM_request_slta_request;

Clear EEPROM Pool

This message will clear the entire EEPROM pool, and will remove the following configurable strings. The strings will be set to null strings, not their default values.

- Modem configuration string
- Hang-up string
- Dial prefix string
- All dial-out directory entries

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 16
} ANM_clear_eepool_request;

Install NVConnect

This message writes the NVConnect byte. A value of 0xFF (the default) will disable this feature. When this feature is enabled, the PL-SLTA initiates a dial-out when it receives a network variable update.

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 17
    byte NVConnect; // two 4-bit fields
} ANM_install_nvconnect;

Install NSIConnect

This message writes the NSIConnect byte. A value of 0xFF (the default) will disable this feature. When this feature is enabled, the PL-SLTA initiates a dial-out when it receives an AddMyNSI message.

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 18
    byte NSIConnect; // two 4-bit fields
} ANM_install_nsiconnect;
Install CallbackEnable

The CallbackEnable configuration byte is written as a Boolean value using this command. A value of zero (the default) will disable the callback configuration. When this feature is enabled and a remote host dials-in to the PL-SLTA, the PL-SLTA terminates the call and initiates a dial-out using the address entry requested by the initial dial-in.

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 19
    byte NSIConnect; // Non-zero enables callback
} ANM_install_callbackenable;

Report SLTAE

This command will result in a response that includes the address of the SltaEE data, the revision of the data structure, and its length. This information can be used to read back any portion of the structure for analysis and deconstruction.

typedef struct {
    byte sub_code;  // always #1
    byte app_command; // value = 20
} ANM_report_sltaee;

The response is:

typedef struct {
    word abs_address; // 16-bit absolute address of
    // structure
    byte version;  // currently 1
    byte length;
} ANM_report_sltaee_resp;

Modem Compatibility

The PL-SLTA has been tested with the following modems:

<table>
<thead>
<tr>
<th>Modem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Data</td>
<td>Smart One external modem 33,600 bps Data/Fax modem</td>
</tr>
<tr>
<td>Diamond</td>
<td>SupraExpress 336e external faxmodem</td>
</tr>
<tr>
<td>Hayes</td>
<td>Accura 336 external fax modem</td>
</tr>
<tr>
<td>US Robotics</td>
<td>Sportster Voice external 28.8 faxmodem</td>
</tr>
</tbody>
</table>

Synchronous communication should be disabled when synchronous modems are connected at lower serial bit rates (less than 4800 bps). Alternatively, data compression can be disabled at lower bit rates. For example, two V.42 or V.32bis modems connected at 2400 bps will have very low throughput due to the slow serial bit rate.

You should disable XON/XOFF flow control in the modem when using it with the PL-SLTA.

Note: When using the Hayes Accura external Fax Modem with Simultaneous Voice and Data 33.6 modem with the US Robotics 28.8 Faxmodem with Personal Voice Mail, an incompatibility in communication occurs. Testing the two modems via Hyperterminal reveals that the CD signal (Carrier Detect) is incorrect. Due to this communication incompatibility, do not use these two modems together with the PL-SLTA.
V.90 Modems Tested with the PL-SLTA

The PL-SLTA has been tested with the following V.90 modems:

<table>
<thead>
<tr>
<th>Company</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Data</td>
<td>Smart One 56SX V.90</td>
</tr>
<tr>
<td>Zoom Telephonics</td>
<td>Faxmodem 56XX Dualmode</td>
</tr>
<tr>
<td>US Robotics</td>
<td>External Faxmodem V.90 56K</td>
</tr>
</tbody>
</table>

The V.90 modems were tested using default Windows settings on the host side:

<table>
<thead>
<tr>
<th>Windows Setting</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use error control</td>
<td>On</td>
</tr>
<tr>
<td>Compress Data</td>
<td>On</td>
</tr>
<tr>
<td>Hardware (RTS/CTS) Flow Control</td>
<td>On</td>
</tr>
<tr>
<td>Standard Modulation Type</td>
<td>On</td>
</tr>
</tbody>
</table>

In all cases, both the host side and the PL-SLTA side were set to 115200 baud.

Since 56K V.90 or V.FAST modulation is asymmetrical and requires a digital modem and special phone line connections at one end, do not expect a modulation rate of greater than 33.3K when connecting with two of these modems. This is normal.

When using the USRobotics modem on the PL-SLTA side, set the DIP switch SW4 on the modem to the DOWN position. This configures the modem for “No echo offline commands”.

Remote Site Monitoring

This section describes how to configure a PL-SLTA located at a remote site to dial-in to a Windows 98/NT/2000/XP host PC running an LNS application using the equipment shown in Figure 24.

Hardware Setup

If you are using a modem with speeds less than or equal 28.8K band, set the switches as shown in Figure 25.

![DIP Switch Settings for Use with a Diamond SupraExpress 336e Modem](image)

Use an Echelon Model 73380 Null Modem Cable between the PL-SLTA and the modem. A 3-conductor cable, complete with connectors wired as shown in Table 3, also will be required.

Using the PL-SLTA with a Modem
Software Setup

The host PC must operate under Windows 98/NT/2000/XP.

1. Run the setup.exe supplied for the PL-SLTA Driver and the SLTA Link Manager software. After common installation queries are addressed, the setup program will display the dialog box shown in Figure 26.

2. Answer “Yes” if using 16-bit LonManager API for Windows-based applications. Answer “No” if using 32-bit LNS-based applications, i.e., LonMaker Integration Tool.

Figure 26. DOS Virtual Device Driver Configuration Screen

3. Reboot the PC after installing the PL-SLTA driver software.

SLTA Link Manager

The SLTA Link Manager software provides the interface to setup the PL-SLTA for different modes of operation. The remote PC is assumed to have the network database required for the connection application. The LonMaker tool is used in this example.

The SLTA Link Manager (sltalink.exe) launches automatically from the Start/Programs/Startup folder when Windows boots up. The SLTA Link Manager must be running prior to accessing the PL-SLTA from the LonMaker tool. The SLTA Link Manager does not appear as a button on the taskbar, but as an icon in the notification section of the taskbar as shown in Figure 27.

Figure 27. Taskbar with SLTA Link Manager Running

To configure the PL-SLTA and modem for remote dial-in access, follow these steps:

1. Temporarily connect the serial port of the host PC directly to the PL-SLTA.

2. Double-click on the SLTA Link Manager icon in the notification section of the taskbar (Figure 27).
3. Confirm that the left panel of the SLTA Link status bar shows the ‘Local SLTA-10’. If not, then click on the Link.Select> menu to select the Local PL-SLTA.

4. Click on the Manual Connect Link speed button (left-most button on the toolbar) to establish a connection to the local PL-SLTA. A successful connection will immediately be reported in trace frame of the SLTA Link Manager. If the application requires only a dial-in from a remote PL-SLTA to a host PC, then the main configuration concern is the password used to gain access to the PL-SLTA remotely.

5. Use the Device.Configure SLTA… dialog to set the password used to control access to the PL-SLTA. Figure 29 shows a Password, with Clear EE Pool on Apply selected, and Auto-dialout options not selected.

6. Click Apply. The PL-SLTA connected to the PC updates with the new configuration information.
7. The PL-SLTA can now be tested using the LonMaker tool and selecting SLTALON1 to make the current active local PL-SLTA be the network interface. In the LonMaker tool, the LonMaker.Status Summary... command will provide a quick confirmation that the PL-SLTA functions properly and can reach all devices in the target network.

8. Once the operation of the PL-SLTA has been established, shut down the LonMaker tool and manually disconnect the local PL-SLTA using the SLTA Link Manager.

9. Connect the proper null modem cable between the PL-SLTA and the modem and verify that both devices are powered.

10. From SLTA Link Manager, use the Link.New... command to define the remote connection description. See Figures 30 through 32.

11. Provide the dialing information as shown in Figure 31.

12. Provide the password that was configured during the local connection, as shown in Step 5.
13. Attach the remote PL-SLTA modem to the telephone network. The PL-SLTA is now configured to dial-in and perform the network management functions required to diagnose and maintain the network just as if there was a local connection between the host PC and the remote site. To establish the remote connection, connect the modem of the PC operating the network management tool (i.e., LonMaker tool) to the telephone network, select the remote link just defined using the Link.Select> menu function, and click on the Connect button. The network management application can now connect to the SLTALON1 device.
This chapter describes how to develop a gateway for a user-defined EIA-232 serial protocol. The PL-SLTA is shipped with SLTA firmware that allows it to be used as a LonTalk network interface for network management, monitoring, and control. When the SLTA firmware is installed, the data passed between the firmware and the LNS host application via the EIA-232 port is in the format of LonTalk protocol application buffers. Serial gateways are developed using the model76100-n PL-PSG/3 (Programmable Serial Gateway). These products allow you to replace the SLTA firmware with your own firmware, enabling custom protocols to be translated into LonTalk.
Creating a Serial Gateway

For PL-SLTA hardware platform applications that cannot use the LonTalk network interface protocol on the EIA-232 link, the PL-SLTA hardware may be programmed for other data formats using Neuron C. For example, the PL-SLTA hardware may be used to connect ASCII devices such as printers, modems, and terminals to a LONWORKS network. The PL-SLTA hardware may also be programmed to build a gateway between a foreign EIA-232-based protocol, such as the serial port on a Programmable Logic Controller (PLC), and the LonTalk protocol. The PL-SLTA hardware without the firmware (e.g., without the PROM) is called a programmable serial gateway.

The programmable serial gateway is used to create a LONWORKS application node, so all the standard Neuron Chip firmware features described in the Neuron C Programmer’s Guide are available to the application programmer. The firmware images for NSI and MIP mode, as well as capabilities described in the LONWORKS Host Application Programmer’s Guide are not available. For example, there is no support for modem control or for more than 62 bound network variables on such a node. The serial gateway will also not be usable as a network interface for LonMaker, LonManager DDE Server, or LNS-based applications, although it can still be installed and managed by such an application.

SLTA/PSG History

Echelon manufactures (or has manufactured) the following LONWORKS serial devices. Each device ships with firmware:

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
<th>Firmware</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLTA</td>
<td>Original Serial LonTalk Adapter.</td>
<td>Firmware implemented MIP level functionality.</td>
<td>Replaced by SLTA/2</td>
</tr>
<tr>
<td>SLTA/2</td>
<td>Second generation SLTA.</td>
<td>Firmware implemented MIP level functionality.</td>
<td>Replaced by SLTA-10</td>
</tr>
<tr>
<td>SLTA-10</td>
<td>Third generation SLTA.</td>
<td>Firmware implements MIP or NSI level functionality.</td>
<td>Current product</td>
</tr>
<tr>
<td>PL-SLTA (Model 76000-n)</td>
<td>Third generation SLTA with support for PLT-22 power line transceiver.</td>
<td>Firmware NSI level functionality.</td>
<td>Current product</td>
</tr>
</tbody>
</table>
Each of the serial adapters listed above may also be ordered without firmware. The hardware is identical, but no firmware is shipped with the device. The user must provide firmware using the functions described below.

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
<th>Firmware</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSG</td>
<td>Original Programmable Serial Gateway; Hardware is identical to SLTA.</td>
<td>None</td>
<td>Replaced by PSG/2</td>
</tr>
<tr>
<td>PSG/2</td>
<td>Second generation PSG; Hardware identical to SLTA/2.</td>
<td>None</td>
<td>Replaced by PSG/3</td>
</tr>
<tr>
<td>PSG/3</td>
<td>Third generation PSG; Hardware identical to SLTA-10.</td>
<td>None</td>
<td>Current Product</td>
</tr>
<tr>
<td>PL-PSG/3</td>
<td>Third generation PSG with support for PLT-22 power line transceiver. Hardware identical to PL-SLTA.</td>
<td>None</td>
<td>Current Product</td>
</tr>
</tbody>
</table>

Depending on the hardware you intend to use, one of the following `#defines` must be defined in your source (.NC) just before the PSG.H file is included.

- `#define SLTASIP` When targeting PSG-10
- `#define SLTA2` When targeting PSG/2
- `#define PSG20` When targeting PSG/3, PL-PSG/3 or PSG-20

**Programmable Serial Gateway Hardware Resources**

The PL-SLTA hardware includes a UART attached to the I/O pins of a Neuron Chip, memory, oscillator, reset circuitry, and a power line transceiver, as shown in Figure 33.

![Figure 33. PL-SLTA Block Diagram](image-url)
Developing a PSG Application with the NodeBuilder Development Tool

Developing an application for the PL-PSG/3 serial gateway is similar to developing an application for any other custom device. Follow the device creation procedure outlined in Chapter 5 of the NodeBuilder User's Guide.

Warning: The PSG Software relies on the software update provided in NodeBuilder Patch 5. Use the following steps to make certain this patch has been applied to your NodeBuilder development tool before building your PSG images.

1. Select About NodeBuilder… in the NodeBuilder HELP menu.
2. Check to be sure the version listed in the “About” dialog box is Version 1.5 Build 05.

PSG Software Installation

The PL-PSG/3 software contains the following files:

- PLPSG.DTM //Device Template for PSG/3
- Readme.txt //Must read notes
- PSG.LIB //Neuron C library containing PSG functions
- PSG.H //Function prototypes
- PSGREGS.H //UART register and address definitions

To install the PSG software on your development tool:

1. Copy the device template (PLPSG.DTM) to your device template directory. By default, this directory is c:\lonworks\template.
2. The two header files (PSG.H and PSGREGS.H) should be copied to your development tool include directory. By default, this directory is c:\lonworks\neuronc\include.
3. Copy the PSG.LIB file to your development tool’s library directory. By default, this directory is c:\lonworks\images.
Firmware Library Support

To aid in programming the custom serial gateway application, a firmware library (PSG.LIB) provides hardware access functions callable from Neuron C. (PSG.LIB replaces and supercedes the SLTA.LIB library that shipped with the SLTA/2 and PSG-10 products.) The include files also have been renamed to PSG.H and PSGREG.H. The new include files contain functions that are backward compatible with the functions declared in SLTA.H and SLTAREG.H.

In most cases, this library provides all the functions necessary to control the UART hardware and to read and write data from the serial port. Your application program will normally make use of some or all of the firmware functions included in the library PSG.LIB. Prototypes for the following functions, and the enumeration literals used in the following descriptions, are in the include file PSG.H.

This library is available for free download from the licensed software section of the Developer’s Toolbox at www.echelon.com. The library also is provided with the NodeBuilder Development tool.

Usage

A single programmable serial gateway library (PSG.LIB) is included with the PSG software. Depending on the hardware you intend to use, one of the following #defines must be defined in your source (.NC).

```
#define SLTASIP      When targeting PSG-10
#define SLTA2        When targeting PSG/2
#define PSG20        When targeting PSG/3, PL-PSG/3 or PSG-20
```
This will control which low-level I/O access functions are used. Include the `#define` before the `#include <PSG.H>` statement. For example:

```c
#include <netdbg.h>
define PSG20
#include <psg.h>
... when (reset)
{
    ...
}
```

### Code Development and Debugging

The PSG applications can be debugged only with the NodeBuilder Development Tool. The LonBuilder Development Tool may be used to develop/load/export applications, however no debugging facilities are available.

Use the Nodebuilder tool with the PLPSG device templates described above to create the following default [dummy] program:

```c
#include <netdbg.h>
define PSG20
#include <psg.h>
when (reset)
{
}
```

1. Export the program to Motorola S Records or Intel HEX format. See Chapter 5 of the *NodeBuilder User's Guide* for details.

2. Load an AT29C512 or AT29C010 FLASH part with your PROM programmer.

3. Insert the FLASH. into the PSG and power the PSG. Be sure that your NodeBuilder tool is connected to the PSG using the LONWORKS network. (The same way you would connect to any other custom node.)

It is now possible to create a real application and download it over the LONWORKS network to the PSG. The NodeBuilder debugging features are available if you included `<netdbg.h>` in your source.

**Note:** You should remove the reference to `<netdbg.h>` when creating your final distribution build.

### PSG.LIB Functions

**void slta_init(slta_format, slta_baud, slta_intfc);**

This function initializes the UART. It sets up the frame format, the serial interface bit rate, and the modem handshake lines. The frame format parameters are listed in `PSG.H` and may be set to:

<table>
<thead>
<tr>
<th>Format Parameter</th>
<th>Data Bits</th>
<th>Parity</th>
<th>Stop Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>format_8N1</td>
<td>8</td>
<td>no</td>
<td>one</td>
</tr>
</tbody>
</table>
The slta baud parameter may be set to baud_300, baud_600, baud_1200, baud_2400, baud_4800, baud_9600, baud_14400, baud_19200, baud_38400, baud_57600, or baud_115200. The EIA-232 interface parameter may be set to intfc_3wire, or intfc_8wire. Configuration switch inputs are ignored—all configuration information is taken from the parameters supplied to slta_init(). This function should be called once in the reset clause of the application program.

Example:

```c
when (reset) {
    slta_init(format_8N1, baud_38400, intfc_3wire);
}
```

unsigned slta_config(void);

This function reads the configuration inputs. Each input corresponds to a bit in the value returned by slta_config(). All of the configuration inputs are available for application use. The model PL-PSG/3 DIP switches are only valid for the standard SLTA firmware. An application which uses the software in PSG.LIB may allocate these inputs for any purpose. To set the serial bit rate and other parameters, see the function slta_init(). The DIP switches on the PL-PSG/3 return a logic 1 when set to up position, and a logic 0 when set to the down position. The BAUD0 input corresponds to the least significant bit, followed by the BAUD1, BAUD2, AUTOBAUD, and the CFG0 through CFG3 inputs. The CFG3 input corresponds to the most significant bit.

```c
boolean slta_txrdy(void);
```

This function returns TRUE if the UART is ready to accept a character to be transmitted and FALSE otherwise.

```c
void slta_putchar(unsigned data);
```

This function waits until the UART is ready, and then transmits the data character. If the UART is busy, this can take up to one character time. Since the UART is buffered, this function can return before the character is actually transmitted.

```c
void slta_puts(const char *s)
```

This function waits until the UART is ready and then outputs a null-terminated string to the UART. Since the UART is buffered, this function can return before the string is
completely transmitted. The terminating NUL is not transmitted. For other useful string-
manipulation functions, see the Neuron C Reference Guide.

Example:

```
network input SNVT_str_asc text_message;
when(nv_update_occurs(text_message)) {
  slta_puts(text_message.ascii);
  slta_puts("\r\n");
}
```

boolean slta_rxrdy(void);

This function returns TRUE if the UART has one or more characters in its input FIFO
buffer and FALSE otherwise. The UART used in the programmable serial gateways has a
16-character input FIFO buffer.

```
long slta_getchar(void);
```

This function tests to see if a character is waiting in the UART’s input FIFO buffer. If
there is no character waiting, the function returns -1. If there is a character waiting, it is
returned in the least significant byte, and zero is returned in the most significant byte.

Example:

```
when (slta_rxrdy()) {   // keep polling UART
  char c;
  c = (char) slta_getchar(); // get the character
}
```

## Advanced Applications

For most applications, the functions in PSG.LIB are all that are necessary to create custom programs for a
programmable serial gateway using Neuron C. However, for specialized applications, the registers of the
UART and the programmable serial gateway may be accessed directly by the application software. This
section describes these registers and how they are accessed. For complete documentation, obtain a data sheet
for the UART. The UART is an Exar Model 16C550I
(\[http://www.exar.com/products/st16c550.html\]).

The registers are accessed using the `slta_write_uart()` and `slta_read_uart()` functions, which
can be used to access any of the locations in the PL-PSG/3 or PSG-20 I/O space. When using these functions,
include the file `PSGREG.H`, which contains definitions for many of these locations.

```
void slta_write_uart (unsigned addr, unsigned data);
```

This function writes the data byte to the memory mapped I/O location defined by `addr`.

```
extern unsigned slta_read_uart (unsigned addr);
```

This function reads the data byte from the memory mapped I/O location defined by `addr`, and returns that
value.

The map for the extended address space is shown in the following sub-sections. Neuron C declarations for
these addresses and bit assignments are provided in the include file `SLTAREG.H` on the PSG software
diskette.
UART Registers

The UART registers are located at address 0xE780 – 0xE787. See the UART datasheet for register usage information.

PROM/FLASH Specifications

The following specifications apply to PROM and FLASH.

- PROM (90ns)
- FLASH 29C512 or 29C010
This chapter provides solutions to problems that may arise with a modem attached to a PL-SLTA.
Troubleshooting

A Host/Modem - Modem/PL-SLTA configuration has many user-selected options including the choice of modems, configuration of the modems, the operating system of the host, the network interface link protocol, and the serial bit rates of both the host/modem link layer and the PL-SLTA modem link layer.

The most common problem is a failure to use the correct PL-SLTA Null Modem Cable specified in Chapter 3.

PL-SLTA and Modem Do Not Answer or Pick Up

The modem attached to the PL-SLTA must be configured to auto-answer if you want it to pick up and connect when dialed up. Set the modem's S0 register to a non-zero value.

Modems Will Not Connect

This is an unusual case with current modems that utilize modulation scheme fallbacks and error control negotiation fallbacks. One rule for modem modulation speed configuration is that if your modem can be configured to connect at the speed of the last ‘AT’ command, do so. Also, be sure that the modem’s connect/carrier (Register S7) wait time is sufficient. Start with 60 seconds.

PL-SLTA to Host Link Fails Completely

This can be observed as repeated retries at the link layer when the first actual downlink or uplink transfer is attempted. This can be due to the following condition:

- Using XON/XOFF flow control in your modem. Since the PL-SLTA network interface protocol is a binary one, this configuration will interfere with, or lock up, your modem. Be sure that this feature is disabled in your modems.

PL-SLTA to Host Link Fails Partially

This can be observed as retries at the link layer when certain downlink or uplink transfers are attempted. This can be due to any of the following conditions (in order of likelihood):

- Extreme-case delays in either the modem or the connection.

- Modem serial bit rate overrun. For example, if the PL-SLTA serial bit rate is set to 38,400 bps and the modem telephone line speed is set to 2,400 bps, the modem will likely be overrun by sending it data faster than it can transmit it. This can occur since no flow control schemes can be used to restrict the rate that data is sent to the modem. In general, set the modem link rate equal to the telephone line speed. In certain cases it will be acceptable to exceed the telephone line speed - for instance, with a 14,400 bps V.32bis modem with data compression enabled, it may be possible to run the modem link at 19,200 bps.

- Full duplex FIFO overrun. This is caused by excessive full duplex traffic when using the buffered link protocol.
PL-SLTA Sends Modem Configuration String, But It Has No Effect

Most modems will determine the serial bit rate based on the assumption that the first two characters sent to them (while in the command mode) are the characters “AT”. This means that a type of bit rate detection is being performed when these characters are sent to the modem. If the modem has been power cycled it will need to repeat this process. Some modems cannot handle being sent an entire configuration string following these first two characters during this link rate sensing phase. One way to accommodate these modems is to add an extra AT command, followed by a delay, to the front of the configuration string.

For example:

```
“AT!~ATE0&CLD2S0=1!”
```

This will allow the modem time to become fully synchronized with the new link bit rate before sending any actual command strings to the modem.