LONWORKS® Custom Node Development

January 1995 **LONWORKS Engineering Bulletin**

Introduction

This engineering bulletin describes the steps needed to design and test a LONWORKS custom node. By definition, a custom node includes a Neuron® Chip and contains all the components necessary to function as a LONWORKS node. This bulletin describes custom nodes that run their entire application program on the Neuron Chip; see the LONWORKS *Host Application Programmer's Guide* for a description of custom nodes based on other host processors.

A custom node may be intended for shipment to customers or intended for use as a prototype to analyze the node's behavior in its working environment. In both cases, the node is based on custom hardware remote from the LonBuilder® or NodeBuilder™ hardware connected using an appropriate transceiver, or implemented with the NodeBuilder LTM-10 node.

Before moving to a custom node, the Neuron C application program for the node, along with its associated I/O hardware, should be tested using a LonBuilder Developer's Workbench or NodeBuilder.

Although not residing within the LonBuilder Development Station, a custom node's interaction with other nodes (emulators or custom nodes) may be closely monitored using the LonBuilder or LonManager® Protocol Analyzer. In addition, the LonBuilder Network Manager or LonMaker Installation Tool may be used to perform network management functions including application code downloading, binding, node testing, and reconfiguration.

Both LonBuilder and NodeBuilder permit browsing of network variables of a node. This feature allow the user to easily verify a node's input and output network variables' behavior.

Echelon's LONWORKS transceivers and control modules provide a straightforward and quick means to move from the development environment to a custom node using off-the-shelf-parts. Restrictions on form factor, special transceiver requirements, and the need for different memory configurations may require the development of a custom node without use of the control modules. This bulletin describes how to build a custom node whether or not you are using LONWORKS control modules.

In addition, a list of common pitfalls is included at the end of this document. This list includes hints, advice, and assorted bits of wisdom collected from experiences in developing custom nodes.

Related Documentation

The following is a list of documents available from Echelon that provide additional information related to custom nodes:

- *Neuron Chip Data Book.* Describes the memory, I/O, and communications ports of the Neuron Chip, together with the support circuitry required for clock, reset, and service functions. Pin assignment, pad layout, and electrical and environmental specifications are also provided.
- *Neuron 3150 Chip External Memory Interface* engineering bulletin. Describes the interface and timing requirements for the memory bus of the Neuron 3150 Chip. Describes interfaces to PROM, RAM, EEPROM, flash memory, and NVRAM.
- *TPT/XF-78 Twisted Pair Transceiver Module Data Sheet.* Describes a 78kbps twisted-pair transceiver module which can be used as a drop-in transceiver solution for any custom node.
- *TPT/XF-1250 Twisted Pair Transceiver Module Data Sheet.* Describes a 1.25Mbps twisted-pair transceiver module which can be used as a drop-in transceiver solution for any custom node.
- *LPT-10 Link Power Twisted Pair Transceiver Module Data Sheet.* Describes a free-topology 78kbps twisted-pair transceiver module which permits the data and power delivery on the same pair of wires.
- *FTT-10 Free Topology Twisted Pair Transceiver Module Data Sheet.* Describes a free-topology 78kbps twisted-pair transceiver module which can be used as a drop-in transceiver solution for any custom node.
- *PLT-10 Power Line Transceiver Module Data Sheet.* Describes a 10kbps spread spectrum power line transceiver which can be used as the core module for developing a power line communication interface for a custom node.
- *PLT-20 Power Line Transceiver Module Data Sheet.* Describes a 5kbps narrow-band power line transceiver which can be used as the core module for developing a power line communication interface for a custom node.
- *PLT-30 Power Line Transceiver Module Data Sheet.* Describes a 2kbps spread spectrum power line transceiver which can be used as the core module for developing a power line communication interface for a custom node.
- *LonBuilder Hardware Guide.* Describes the process of software and hardware prototyping of custom nodes using LonBuilder. Chapter 2 describes the hardware products that can be used for prototyping and developing custom nodes. These products include the LonBuilder I/O Evaluation Board, the LonBuilder Extender Card, and the LonBuilder Application Interface Kit. The LonBuilder Application Interface Kit includes the LONWORKS Module Application Interface which is used to debug custom nodes based on LONWORKS control modules.

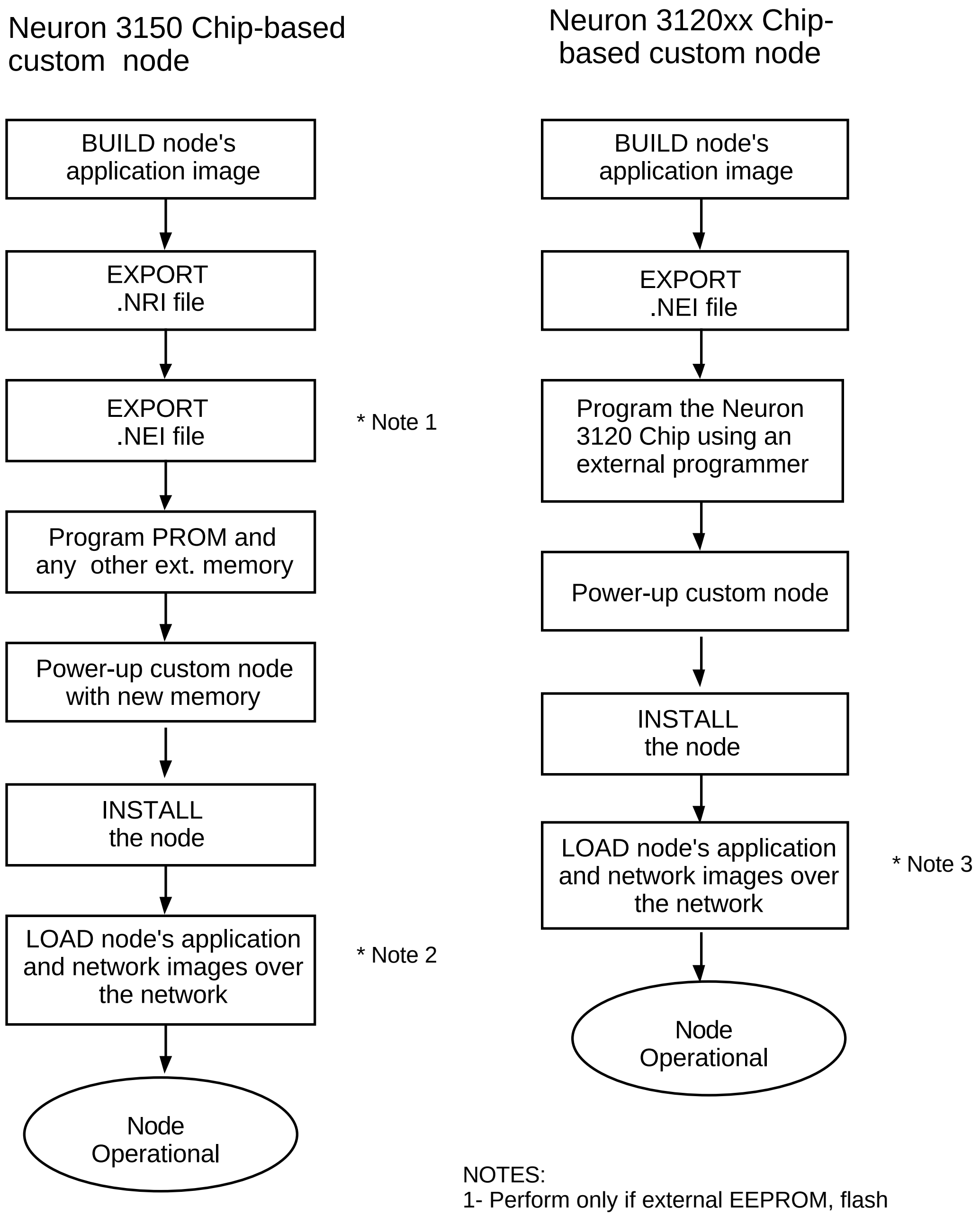
- *LonBuilder User's Guide.* Describes the process of software and hardware prototyping of custom nodes using LonBuilder. Chapter 7 includes a detailed discussion of custom node development.
- *NodeBuilder User's Guide.* Describes the process of software and hardware prototyping of custom nodes using NodeBuilder.
- *LTM-10 User's Guide.* Describes how the NodeBuilder LTM-10 hardware can be used to develop and test custom node hardware.
- *Neuron C Programmer's Guide.* Provides an overview of the Neuron C programming language.
- *Neuron C Reference Guide.* Provides reference information on the functions and features available within Neuron C.
- *LONWORKS Host Application Programmer's Guide.* Describes the process of developing application programs that run on host processors other than the Neuron Chip.
- *LONWORKS Twisted Pair Control Module User's Guide.* Describes the interface requirements for custom nodes using LONWORKS Control Modules.

Bringing Up a Custom Node

This section discusses the steps required to bring up a custom node. The actual procedure applies to both Neuron 3150 and 3120xx Chip-based custom nodes unless otherwise noted. To create a custom node, a developer must:

- Create the node's application program
- Specify the node's hardware
- Program the node's memory
- Install the node on a network
- Load the node
- Test the node

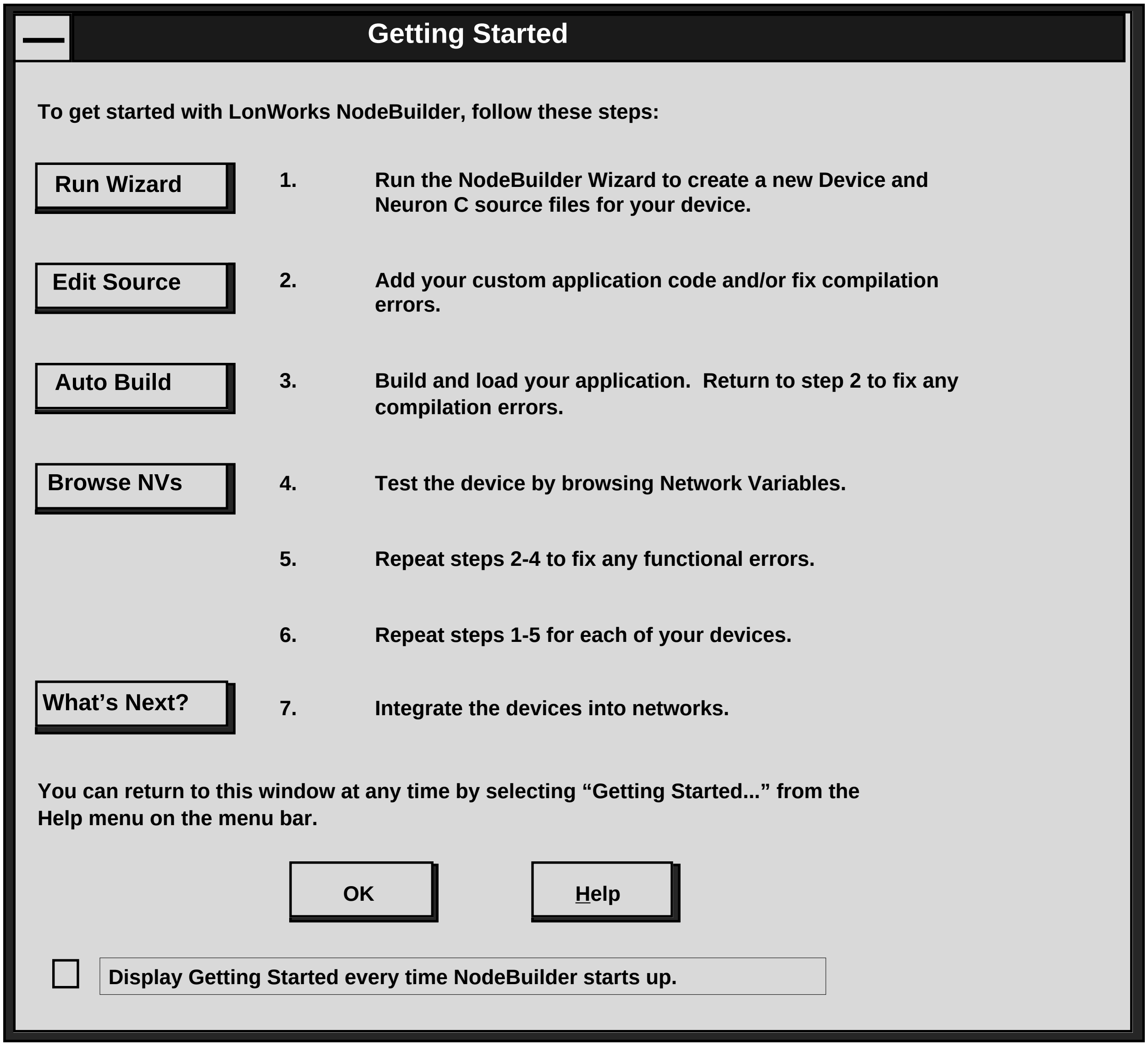
Figure 1 summarizes the steps required to bring up Neuron 3150 and 3120xx Chip-based LONWORKS custom nodes using the LonBuilder Development Station.


Neuron 3150 Chip-based custom node
BUILD node's application image
EXPORT .NRI file
EXPORT .NEI file
* Note 1
Program PROM and any other ext. memory
Power-up custom node with new memory
INSTALL the node
LOAD node's application and network images over the network
* Note 2
Node Operational
Neuron 3120xx Chip-based custom node
BUILD node's application image
EXPORT .NEI file
Program the Neuron 3120 Chip using an external programmer
Power-up custom node
INSTALL the node
LOAD node's application and network images over the network
* Note 3
Node Operational

NOTES:
1- Perform only if external EEPROM, flash memory, or NVRAM is used for user code/data and `applicationless` firmware state was not used during export
2- Not required if .NRI file was Exported with `Configured` option
3- Not required if .NEI file was Exported with `Configured` option

Figure 1 Steps for bringing up a LONWORKS custom node with a LonBuilder Development Station

Figure 2 summarizes the steps requires to do the same with NodeBuilder.



Getting Started

To get started with LonWorks NodeBuilder, follow these steps:

1. Run the NodeBuilder Wizard to create a new Device and Neuron C source files for your device.

2. Add your custom application code and/or fix compilation errors.

3. Build and load your application. Return to step 2 to fix any compilation errors.

4. Test the device by browsing Network Variables.
5. Repeat steps 2-4 to fix any functional errors.
6. Repeat steps 1-5 for each of your devices.

7. Integrate the devices into networks.

You can return to this window at any time by selecting “Getting Started...” from the Help menu on the menu bar.

□ Display Getting Started every time NodeBuilder starts up.

Figure 2 The Getting Started Screen from the NodeBuilder Help Menu

Creating an Application Program

The first step in creating a custom node is to create the application program for the node. The application program should be tested and debugged initially on a LonBuilder Neuron Emulator or NodeBuilder LTM-10 node. This allows the application program to be debugged independently of the hardware design. The *LonBuilder User's Guide* and the *NodeBuilder User's Guide* describe the process of developing and testing an application program.

Specifying the Node's Hardware

A custom node is a LONWORKS node implemented by a LonBuilder or a NodeBuilder user. A custom node can be programmed, implemented, and tested with the LonBuilder Developer's Workbench or NodeBuilder as long as it meets the specifications in the following section. Figure 3 below shows the basic components of a LONWORKS custom node. The components inside the dashed box may be replaced by a LONWORKS Control Module.

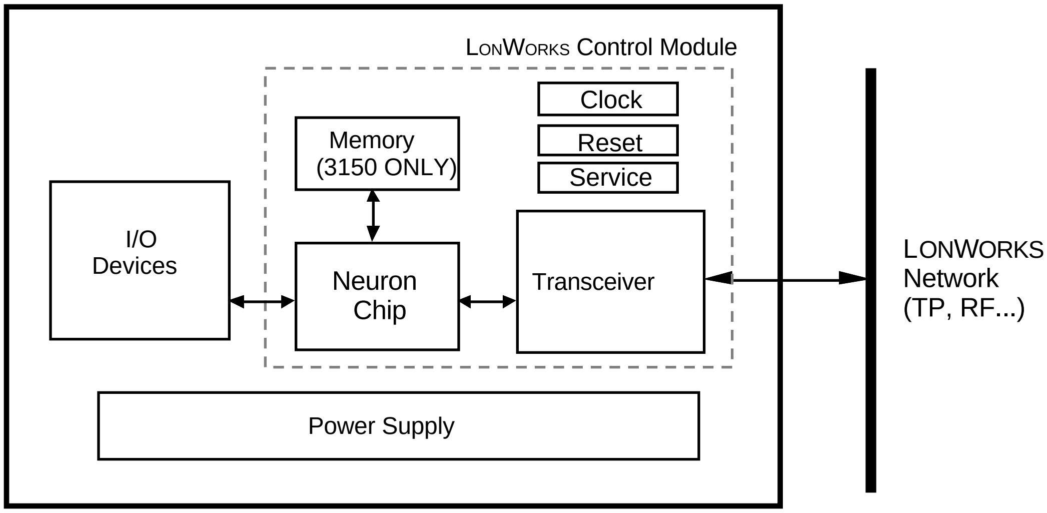


Figure 3 A LONWORKS custom node

Note that external memory is not needed for a node based on a Neuron 3120xx Chip. The following is a description of each of the components needed in a LONWORKS node.

- **Neuron Chip**

  A Neuron 3120, 3120E1, 3120E2, or 3150 Chip. The Neuron Chip provides the control, communications, and I/O for the node. The Neuron Chip may be the only processor on a custom node, or the Neuron Chip may be used as an interface to another microprocessor or microcontroller as described in the *LONWORKS Host Application Programmer's Guide.* The Neuron Chip is programmed as described in the next section.

- **Off-Chip Memory**

  Custom nodes implemented with the Neuron 3120, 3120E1, or 3120E2 Chip use only the memory on the Neuron Chip and do not have off-chip memory. Custom nodes implemented with the Neuron 3150 Chip must have at least

16Kbytes of off-chip ROM or flash memory for the Neuron Chip firmware, and may have additional ROM, EEPROM, flash memory, and RAM as or described below. The off-chip memory is programmed as described in the next section:

- **ROM** 0 or 16 - 68Kbytes (in multiples of 256 bytes) of contiguous memory starting at 0. It may be any type of ROM, PROM, EPROM, or NVRAM. However, the memory must be non-volatile and the custom node developer must program the memory as described in the next section.
- **FLASH** 0 or 21.5 - 42Kbytes (in multiples of 256 bytes) of contiguous flash memory located after the ROM block, if any, but before the RAM block, if any. If there is no ROM on the node, flash memory must start at address 0000 and there must be at least 21.5Kbytes. Flash memory is automatically protected by the Neuron Chip firmware.
- **EEPROM** 0 - 42Kbytes (in multiples of 256 bytes) of contiguous EEPROM located after the ROM block and before the RAM block, if any, but not necessarily contiguous with either. The write time may be configured from 0 to 255 ms. Circuitry should be included for write protecting the EEPROM during power down/power up, if necessary. This can also be other non-volatile memory, such as battery-backed RAM.
- **NVRAM** 0 - 58Kbytes (in multiples of 256 bytes) of contiguous non-volatile RAM located after the ROM block but before the RAM block, if any. NVRAM may be used for the Neuron Chip firmware. Memory used for the Neuron Chip firmware may be write protected, but the custom node developer is responsible for controlling the write protection circuitry.
- **RAM** 0 - 42Kbytes (in multiples of 256 bytes) of contiguous read/write memory.
- **MEMORY MAPPED I/O** 0 - 42Kbytes (in multiples of 256 bytes) of address space dedicated to memory mapped I/O.

For details on interfacing memory to a Neuron Chip 3150, see the *Neuron Chip 3150 External Memory Interface Engineering Bulletin.*

- **Transceiver**

A transceiver is required to interface the custom node with a LONWORKS network. The transceiver must meet the specifications for the communications port interface. Transceiver modules for various media are available from Echelon and other vendors. Echelon provides the following transceivers: TPT/XF-78 and TPT/XF-1250 twisted pair transceivers, LPT-10 link power transceiver, FTT-10 free topology transceiver, PLT-10, PLT-20, and PLT-30 power line transceivers.

- **Clock**

  An input clock for the Neuron Chip. The clock must run at one of the standard input clock rates: 10MHz, 5MHz, 2.5MHz, 1.25MHz, and 625kHz. Refer to the *Clock* section of the *Neuron Chip Data Book* for more information on the clock circuitry.

- **Reset**

  A power-on reset circuit is required to reset a custom node on power up. A reset button is optional, but is useful for resetting a custom node at times other than power up. An LVI (Low Voltage Interrupt) device should be connected to the Neuron Chip's ~Reset pin in order to ensure that the Neuron Chip is not running below a minimum power supply voltage. Refer to the *Reset* section of the *Neuron Chip Data Book* for more information on the reset circuit requirements.

- **Service Button and Service LED**

  A service button is used to install a custom node on a network. When a custom node is installed in a development network, the LonBuilder or NodeBuilder software requests that the user press the service button which causes the Neuron Chip firmware to send a service message. The network manager uses the service message to install the custom node. A service LED is optional, but is highly recommended as it is useful for diagnosing the state of the Neuron Chip firmware: on indicates that the node is applicationless (does not have an application loaded), blinking indicates that the node has an application but has not yet been configured, off indicates that the node has an application and has been configured. Refer to the *Service* section of the *Neuron Chip Data Book* for more information on the service circuitry.

- **I/O Interface**

  An interface between the eleven Neuron Chip I/O pins and external I/O devices. The interface and devices are dependent upon the custom node application.

Programming the Node's Memory

Once the node's application program has been debugged and the custom node hardware has been built, the next step is to program the node's memory. The task of programming memory involves the *export* of the appropriate files. Custom nodes contain memory within the Neuron Chip, and may contain off-chip memory when using the Neuron 3150 Chip. When a node is mass produced, the Neuron Chip, external ROM or flash memory may be programmed as part of the manufacturing process. During development much of the on-chip and off-chip memory may be programmed over the network by the LonBuilder Network Manager or NodeBuilder, with the following two exceptions:

- **Communications Parameters**

The communications parameters are contained in the Neuron Chip's EEPROM and are a portion of a node's application image. The communications parameters define the type of transceiver connected to the Neuron Chip and are used by the Neuron Chip firmware for interfacing with the transceiver. Once a node is installed on a development network, the communications parameters can be changed over the network by LonBuilder or NodeBuilder. However, for a node to be installed on a development network and communicate with the development tool, the node's communications parameters must match the node's transceiver. The initial default communications parameters on a new Neuron Chip are for a 1.25 Mbps twisted-pair or direct-connect transceiver. Therefore, nodes implemented with a 1.25Mbps twisted-pair or direct-connect transceiver do not require their communications parameters to be changed.

For nodes based on the Neuron 3150 Chip, the ROM or flash image created by the LonBuilder or NodeBuilder software defines the initial communications parameters that are copied to internal EPROM when the boot ID in EEPROM does not match the boot ID in the ROM or flash image. If the communications parameters are modified over the network, the original defaults in the ROM or flash image are not used unless the ROM or flash memory is changed. For nodes based on the Neuron 3120xx Chip, the communications parameters must be programmed by using an external Neuron Chip programmer. NodeBuilder does not update communications parameters other than priority.

- **ROM or Flash**

A custom node's ROM cannot be programmed over the network and therefore must be programmed prior to powering-up the node. This is also true of flash memory that takes the place of ROM and contains the system image. The ROM always contains at least the Neuron Chip firmware. On a Neuron 3150 Chip with off-chip memory, the ROM or flash memory can also contain all or part of the application image as described under *Off-Chip Memory* in Chapter 6 of the *Neuron C Programmer's Guide*. The Neuron 3120xx Chip's ROM is pre-programmed and does not require programming by the user. See *Building Custom Nodes* in Chapter 7 of the *LonBuilder User's Guide* for instructions on how to create ROM or flash files that can be used with a PROM programmer to create the ROM or flash chips for a Neuron 3150 chip-based custom node. Refer to *Exporting a LONWORKS Device* and *Building a Custom Node* topics under the NodeBuilder online help for more information.

Installing the Node on a Network

A newly developed custom node will initially be installed on a development network that includes a LonBuilder Development Station or a NodeBuilder. This allows the LonBuilder network management, protocol analysis, and network variable browsing tools and the NodeBuilder's debugging and network variable browsing tools to be used to test the node and its interaction with other nodes.

When developing multiple nodes with LonBuilder, the developer should focus on one node at a time when making the transition from an emulator-based network to a custom node-based network, so that potential problems may be isolated.

Installation of a custom node in a network involves the identification of the node by a network management tool using the node's Neuron ID. LonBuilder and NodeBuilder get this ID from the service pin message issued when the node's service pin is activated. This means that the node must be able to communicate over the network, which implies that it must have a working transceiver and appropriately configured communication parameters. These parameters are set during the programming step. In addition, any routers between the node and the network manager must be installed and loaded. See *Installing Application Node Target Hardware* in Chapter 6, and *Installing Custom Nodes* in Chapter 7, of the *LonBuilder User's Guide*. Refer to the *Building and Loading a LONWORKS Device* topic under the NodeBuilder online Help for more information.

To move an application program from a LonBuilder Neuron Emulator to a custom node, the hardware properties specification must match those of the custom node hardware. Hardware properties are described under *Defining Properties* in Chapter 6 of the *LonBuilder User's Guide*. After this is checked, three modifications may be required from the LonBuilder software for installing the custom node:

- Create a new target hardware object with a hardware type of *Custom Node*. This step is described under *Defining Target Hardware* in Chapter 6 of the *LonBuilder User's Guide*.
- If the media characteristics for the custom node are different from those used when the node was debugged on a LonBuilder Neuron Emulator, creating a new channel object as described under the *Defining Channels* in Chapter 10 of the *LonBuilder User's Guide*. Ensure that a communications path exists from the network manager and protocol analyzer to this channel, either by moving the network manager and protocol analyzer to the new channel, or by installing and loading a router to the new channel. Debugging a transceiver is simplified by first testing it on a LonBuilder Neuron Emulator, in which case the channel definitions will be the same.
- Change the definition of the node specification to select the new target hardware and channel objects. This step is described under *Defining Node Specifications* sections in Chapter 6 of the *LonBuilder User's Guide*.

Loading the Node

Loading a custom node is required if the node has been created using either the `applicationless` or the `unconfigured` state, or if you wish to make changes to the application or configuration without reprogramming the node's memory. A node created in the `configured` state does not need to be loaded for it to be fully functional. See *Loading Built Nodes and Routers* in Chapter 7 of the *LonBuilder User's Guide*. Refer to the *Building and Loading a LONWORKS Device* topic under the NodeBuilder online Help for more information.

For a Neuron 3150 Chip-based node, the loading process involves updating application image parts that reside in external flash memory, EEPROM, and NVRAM, in addition to the network image. This includes the internal EEPROM and any external memory declared as EEPROM that may exist on the node. Application code allocated to RAM by the user will also be updated. For a Neuron 3120xx Chip-based node, the loading process only updates the on-chip EEPROM.

Reloading a Neuron 3150 Chip-based custom node with application changes will require the programming step to be repeated if any application code will reside in ROM. This is because changes to the program linkage may invalidate the code already there. If you are going to change the remainder of the application over the network, you only need to program the ROM. If you want the node to come up in the `configured` or `unconfigured` state, you must also reprogram any portion of the application image in external flash memory, EEPROM, or RAM. You should load the node after you have installed the new ROM.

If you want to be able to download new application programs completely over the network, limit the ROM memory map to 16 Kbytes so that all of the application program is loaded into the on-chip EEPROM. Alternatively, you could use flash memory for the application program which could also contain the Neuron Chip firmware.

Changing the communication parameters may cause the node to stop functioning if the new parameters do not match the transceiver attached to the Neuron Chip.

Testing the Node

The LonBuilder Application Interface Kit (AIK), available from Echelon, is specifically designed to help in the debugging process of custom nodes. Two adapters are included for debugging control module-based devices or custom Neuron-based devices. The control module adapter replaces the LONWORKS control module in the custom node to allow connection to a LonBuilder emulator for both I/O and transceiver testing.

The Neuron Chip adapter is designed specifically for use on custom nodes which are not based on the LONWORKS control module family of products. This is accomplished through a surface mount adapter which essentially replaces the Neuron Chip on the target board with the necessary hardware and interface logic for proper connection to a LonBuilder emulator. I/O and transceiver testing can then be accomplished in much the same way as with the application interface kit.

See the *LonBuilder Hardware Guide* for information on the application interface kit.

Assuming no hardware problems, once the custom node has been properly installed and loaded, the application on that node should start running.

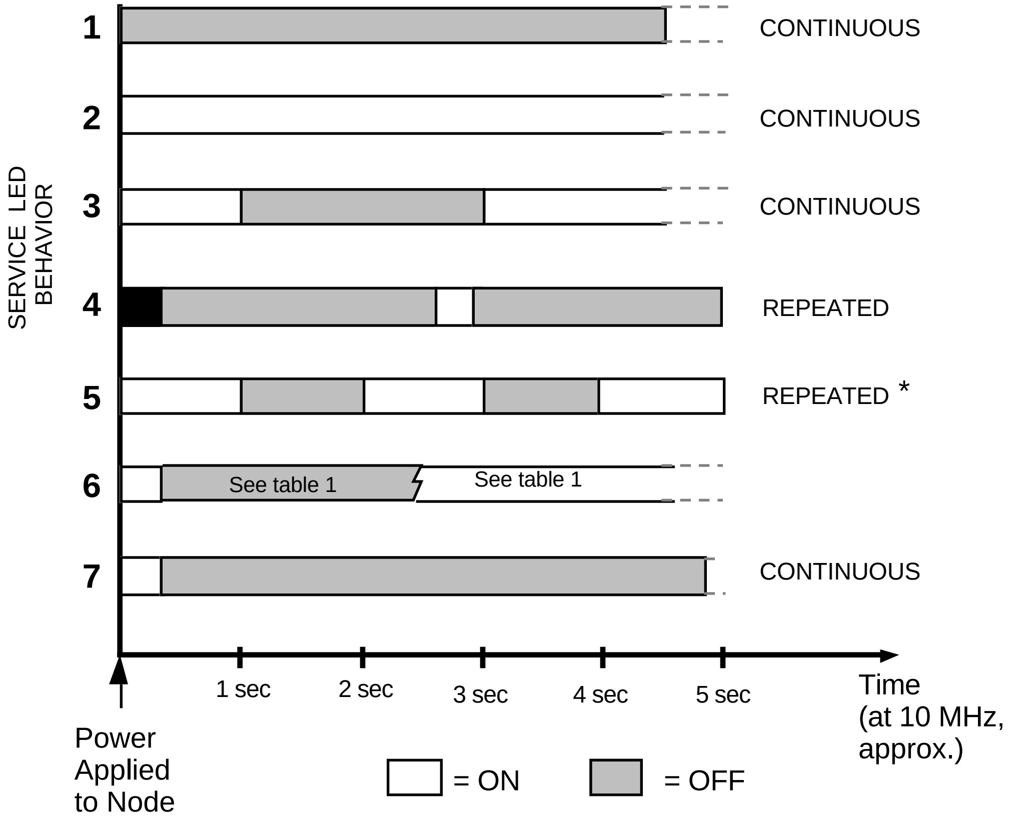
In addition to the application I/O on the node, the condition of the service LED may be observed for clues about the actual state of the node. The behavior of the service

LED, specifically the duration of ON and OFF states, varies depending on the operating state of the node.

Figure 3 illustrates the different service LED behaviors alternatives for any Neuron Chip-hosted node. These are the most common behaviors, others are possible since the state of the service LED is under firmware control and can be affected by both hardware and software anomalies. The state of the service LED can also be affected by the application program using the `active_service_led` built-in variable.

Table 1 describes each of the behaviors shown in Figure 4 under different contexts. Again, this list is not exhaustive and therefore does not provide explanations for every possible service LED behavior.

Behavior 7 represents that of a fully configured node during normal operation.



*Does not scale with the Neuron Chip clock

Figure 4 Possible service LED behaviors showing different duty cycles

Table 1 Explanation of the service LED behaviors shown in Figure 3

BEHAVIOR	CONTEXT	LIKELY EXPLANATION

1	Power-up of a Neuron 3120xx Chip-based node, or a Neuron 3150 Chip-based node with any PROM	Bad node hardware
2	Power-up of a Neuron 3120xx Chip-based node, or a Neuron 3150 Chip-based node with any PROM	Bad node hardware
3	Power-up/Reset	Node is applicationless May be caused by the Neuron Chip firmware when a mismatch occurs on application checksums This behavior is normal if the application was exported to come up `applicationless`
4	Anytime	Watchdog timer resets occurring Possible corrupt EEPROM For a Neuron 3150 Chip-based node, use a newly programmed PROM, or EEBLANK and follow bring-up procedure If you are using an LTM-10 node, you can make the Neuron Chip applicationless by holding down the service button while pressing the rest button. Hold the service button down until the reset LED briefly flashes.
5	Anytime	Node is unconfigured but has an application. Proceed with loading the node.

6	Using EEBLANK on a Neuron 3150 Chip-based custom node	The OFF duration is approximately 10 seconds (10MHz Neuron Chip). This duration scales proportional to the system clock. After this OFF time the service LED should turn ON and stay ON, indicating the completion of the blanking process
6	First power up with a new PROM on a Neuron 3150 Chip-based custom node. Applicationless firmware state exported	The OFF duration is approximately 1 second. Service LED should then turn ON and stay on indicating an applicationless state
6	First power up with a new PROM on a Neuron 3150 Chip-based custom node. Unconfigured firmware state exported	The OFF duration is 1-15 seconds depending on the application size and system clock. Service LED should then begin flashing as in behavior 5 shown in figure 3, indicating an unconfigured state
6	First power up with a new PROM on a Neuron 3150 Chip-based custom node. Configured firmware state exported	The OFF duration is indefinite (1-15 seconds to load internal EEPROM; stays OFF indicating configured state.)
7	Anytime	Node in configured and running normally

As a first step in establishing communication with a newly powered-up (previously blank) custom node, the service pin may be used. Activating the service pin on any node running the Neuron Chip firmware, regardless of the state of the application and network configuration for that node, causes a broadcast message containing the Neuron ID and the program ID for that Neuron Chip to be sent on the network. The LonBuilder Protocol Analyzer or the LonManager Protocol Analyzer can be used to monitor the network for any such messages.

Activating the service pin on any node can serve as a simple test; it should cause the protocol analyzer status LED to blink, indicating the presence of a packet on the network. The protocol analyzer can be used to log the service pin message as described in Chapter 12, *Monitoring Development Networks*, of the *LonBuilder User's Guide.* The logged service pin message indicates that the node hardware and the network connection for that node are operating normally.

It is possible for the protocol analyzer to receive a packet but not have it logged. This may indicate a communication parameter mismatch between the protocol analyzer and the node sending the packet.

Once communication has been established with a node and the node has been installed, the *Test* and *Wink* commands may be issued from the LonBuilder network manager or NodeBuilder to further test the hardware and communication integrity of the custom node. The network manager also includes a network variable browser that can be used to send network variable updates to the custom node and receive network variable updates from the node. The browser can therefore be used to validate that a custom node is functioning correctly. Use of the LonBuilder network manager is described in Chapter 13, *Testing Development Networks*, of the *LonBuilder User's Guide*. Refer to the *Testing a LONWORKS Device* topic of the NodeBuilder online help for more information.

Once all the nodes have been installed on the development network and tested as custom nodes, they may be disconnected from the development network and moved to a production network. Or, the debugged node image may be used to produce the additional custom nodes. See *Exporting and Importing* in Chapter 7 of the *LonBuilder User's Guide*. Refer to *Exporting a LONWORKS Device* topic of the NodeBuilder online help for more information.

Common Pitfalls

This section lists common pitfalls along with hints and suggestions to use during custom node development and bring-up.

• *Installation of custom node fails.* The network manager and the custom node may not have matching transceiver properties and/or hardware. This problem may also be caused if there is no communication path between the network manager and the custom node.

• *Protocol Analyzer not receiving or losing packets.* The protocol analyzer and the custom node do not have matching transceiver properties and/or hardware. This problem may also be caused if there is no path between the protocol analyzer and the custom node. Also make certain that proper termination exists if using a twisted-pair network.

• *Loading errors while trying to load a custom node in a highly active network.* Heavy network traffic on the network could cause a node to miss LonBuilder loading messages. In such a case, the network statistics screen will show a very low network bandwidth utilization figure. In addition, the *test* command should indicate many missed messages for that node. A solution is to take the node(s) causing the heavy traffic offline before trying to load the node.

• *A custom node stops working after a reload.* The communication parameters were changed (LonBuilder only) and then installed into the Neuron Chip of the custom node. If the new parameters have taken effect and if they require the custom node to have a new transceiver, the node will no longer communicate with the network. For a Neuron 3150 Chip-based custom node, use EEBLANK (see Chapter 7, *Programming Custom Nodes*, in the *LonBuilder User's Guide)* and proceed with bring-up as shown in Figure 1 or 2. For a Neuron 3120xx Chip-based custom node, change the node's transceiver so that it corresponds to the newly installed communication parameters.

• *Neuron 3150 Chip-based node hangs during application/configuration update procedure.* The EEPROM data inside the Neuron Chip has been corrupted. Use EEBLANK and proceed with bring-up as shown in Figure 1 or 2. Or else, program a new PROM with the `unconfigured` or `configured` options.

• *Service LED exhibits behavior not shown in Figure 3.* Bad node hardware. Check for possible mis-wirings and PCB faults.

Disclaimer

Echelon Corporation assumes no responsibility for any errors contained herein. No part of this document may be reproduced, transclated, or transmitted in any form without permission from Echelon.

Part Number 005-0024-01 Rev. C

Echelon Corporation
4015 Miranda Avenue
Palo Alto, CA 94304
Telephone (415) 855-7400
Fax (415) 856-6153

Echelon Europe Ltd
Elsinore House
77 Fulham Palace Road
London W6 8JA
England
Telephone +44-81-563-7077
Fax +44-81-563-7055

Echelon Japan K.K.
Kamino Shoji Bldg. 8F
25-13 Higashi-Gotanda 1-chome
Shinagawa-ku, Tokyo 141
Telephone (03) 3440-7781
Fax (03) 3440-7782