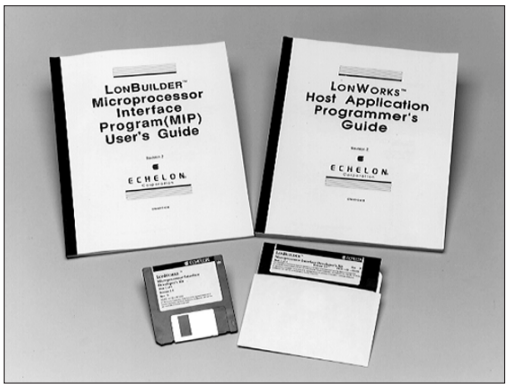
LonWorks® MIP/P20 and MIP/P50 Developer's Kit
Model 23200



- Adds a LonWorks interface to any host processor
- Network interfaces based on the MIP can be used with any host application
- High-speed parallel interface sends and receives hundreds of packets per second
- Optional up-link interrupt for the MIP/P50 reduces latency to incoming network traffic by asynchronously informing the host of the availability of an up-link packet
- Supports host applications with up to 4096 network variables
- ANSI C source code for a network interface library and sample host application included
- ANSI C and PC assembly source code for a sample network driver included

Description

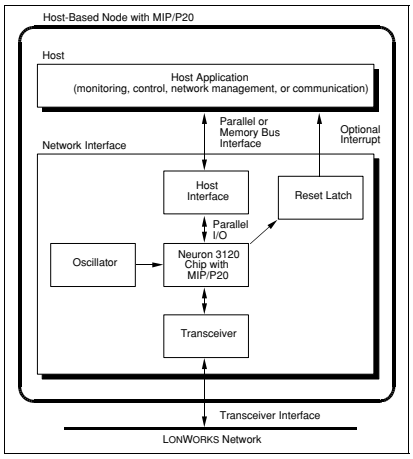
The Microprocessor Interface Program (MIP) is firmware for the Neuron® Chip that transforms the Neuron Chip into a communications coprocessor for an attached host processor. The MIP enables the attached host to implement LonWorks applications and to communicate with other devices using the LonWorks protocol. Applications on the host can send and receive network variable updates and application messages, as well as poll network variables. The MIP opens the LonWorks protocol to a variety of hosts including PCs, workstations, embedded microprocessors, and micro-controllers.

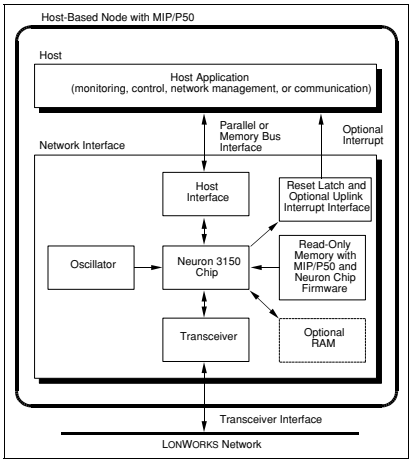
The MIP/P20 and MIP/P50 versions of the MIP use an 11-pin parallel interface to the host, making them ideally suited for micro-controller and microprocessor hosts. The MIP/P20 is optimized for the Neuron 3120® Chip, providing the lowest-cost network interface. The MIP/P50 is optimized for the Neuron 3150® Chip, providing better performance for the cost of an additional PROM. The MIP/P50 may also be used with the Neuron 3120E1 and 3120E2 Chips, but without the up-link interrupt option.

Typical applications for the MIP/P20 include LonWorks devices based on micro-controllers such as the 68HC11, 68HC16, and 8051. Typical applications for the MIP/P50 include LonWorks devices based on high-end micro-controllers such as the 68332 or microprocessors such as the 680x0 or 80x86.

The MIP enables the attached host to communicate on a LonWorks network using the LonWorks protocol. When used with a PC host, network monitoring and control applications for Windows are easily built by using the MIP with the LNS™ DDE Server, or with LNS for Windows. The LonMaker Integration Tool transforms a MIP-equipped PC into a complete network design, installation, and maintenance tool.

The MIP/P20 and MIP/P50 are used to build custom LonWorks network interfaces. A network interface is a device that provides a communications interface between a host processor and a LonWorks network. The network interface may be an integrated device such as the Echelon PCLTA-20 LonTalk Adapter or the SLTA-10 Serial LonTalk Adapter, or may be an embedded device directly attached to the host processor. A custom network interface based on the MIP/P20 or MIP/P50 includes a Neuron Chip running the MIP firmware, transceiver, reset circuitry, and oscillator. The following two figures illustrate the components of LonWorks devices using host processors and network interfaces based on the MIP/P20 and MIP/P50, respectively.





Devices based on the MIP split LONWORKS protocol processing between the host processor and network interface. The network interface with the MIP handles layers 1 through 5 of the LONWORKS protocol. This significantly reduces overhead in the host since the host processor does not have to deal with lower layer network services such as media access control, collision avoidance, acknowledgments, retries, duplicate message detection, message validation, authentication, and priority processing. The host processor is left to run the application program and handle the layer 6 and 7 protocol services: network variable processing and explicit message processing. Using these services, the host can easily send and receive both network variable updates and application messages.

Separating the upper two layers of the LONWORKS protocol from the lower five layers has the added benefit of making the MIP independent of the host application. The host application, including its network variables, can be changed at any time without modifying the MIP code in the network interface. This lowers development and maintenance costs since the MIP code in the network interface does not have to be tailored to an application and never has to be modified.

Hosts using the MIP can contain up to 4096 local network variables, each of which can be connected to an unlimited number of network variables on other devices. This limit is higher than the Neuron Chip-hosted device limit of 62 bound network variables because the network variable configuration is managed by the host instead of the Neuron Chip inside the network interface. The use of bound network variables reduces network loading and increases system capacity by allowing values to be updated over the network only when necessary and eliminating the need for constant polling. As with any device, there is no limitation on the number of network variables that can be explicitly written or polled.

Usage

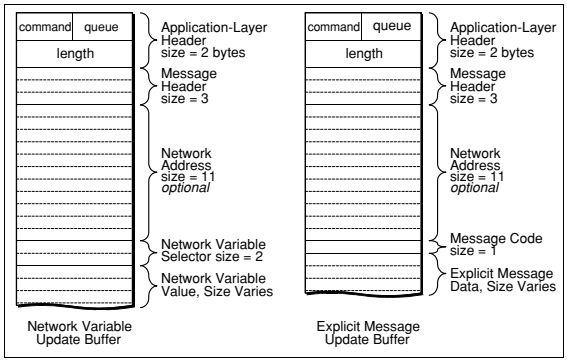
The MIP/P20 and MIP/P50 are delivered in a Neuron C library that extends the LonBuilder® and NodeBuilder® software to include system calls for the MIP. The developer creates a short Neuron C program that defines the initial buffer configuration for the network interface and calls the MIP/P20 or MIP/P50 system function. The LonBuilder or NodeBuilder tool is used to create a ROM or Neuron EEPROM image that is programmed into a PROM or Neuron 3120 Chip using a PROM or Neuron 3120 programmer. The PROM and Neuron 3150 Chip or the Neuron 3120 Chip is installed in a network interface that includes a parallel or memory mapped host interface, reset latch, oscillator, and communications transceiver. The host interface must meet the requirements of the Neuron Chip parallel I/O interface as described in the *Parallel I/O Interface to the Neuron Chip* engineering bulletin. External RAM may be added for the MIP/P50 to increase the communication's buffer space. Additional buffer space may be required for applications using large packet sizes, or applications receiving many messages, responses, or acknowledgments simultaneously.

An address decoder can be added to generate an up-link interrupt for the MIP/P50. A user-replaceable function writes to a memory location whenever a packet is available for the host. By decoding the memory write and generating an interrupt, the host can use the interrupt to reduce latency of response to incoming network variable updates and messages.

A simple network driver is implemented on the host processor to manage the interface with the MIP. The LONWORKS network driver protocol defines a standard interface specification for LONWORKS network interfaces that isolates the host application from implementation dependencies of the network interface. This allows the same host application to be used with multiple network interfaces, preserving investment in host application development. Complete specifications for the network driver are included in the *Microprocessor Interface Program (MIP) User's Guide*. Complete source code for a DOS network driver is provided with the MIP that can be used as the basis for a network driver for any host.

A sample host application is also provided with the MIP/P20 and MIP/P50. This application illustrates how a host application can send and receive network variables and application messages using the network driver. The samples demonstrates how a host application can implement network variables and respond correctly to network management messages for binding those network variables.

The sample host application also includes source code for a network interface library that simplifies the use of any LONWORKS network interface, including a network interface based on the MIP. The library includes function calls to initialize and reset the network interface; send, receive, and respond to LONWORKS messages; and handle errors. LONWORKS messages may include network variable updates and polls, as well as application messages. For example, to send a network variable the host application initializes a buffer that contains the network variable selector and the new value for the network variable and then calls the `NiSendMsgWait( )` function in the network interface library. The following figure illustrates the application buffer format for network variable updates and application messages.



Specifications

MIP/P50 Throughput		
Unacknowledged:		
1-byte message	303 packets/sec	(2,426bps)
8-byte message	289 packets/sec	(18,490bps)
32-byte message	260 packets/sec	(66,560bps)
228-byte message	158 packets/sec	(288,739bps)
Acknowledged:		
1-byte message	106 packets/sec	(851bps)
8-byte message	103 packets/sec	(6,618bps)
32-byte message	94 packets/sec	(24,141bps)
228-byte message	55 packets/sec	(100,685bps)

Note: PC/386 host, 25MHz; Neuron 3150 Chip network interface with interrupts enabled, 10 MHz; protocol overhead of 9 bytes per message.

MIP/P20 Throughput		
Unacknowledged:		
1-byte message	205 packets/sec	(1,636bps)
8-byte message	205 packets/sec	(13,088bps)
32-byte message	170 packets/sec	(43,546bps)
183-byte message	103 packets/sec	(150,499bps)
Acknowledged:		
1-byte message	76 packets/sec	(606bps)
8-byte message	74 packets/sec	(4,717bps)
32-byte message	68 packets/sec	(17,510bps)
183-byte message	47 packets/sec	(68,662bps)

Note: PC/386 host, 25MHz; Neuron 3150 Chip network interface, 10 MHz; protocol overhead of 9 bytes per message.

MIP Network Interface Commands	Buffer Request Send Message Local Network Management Command Reset Flush and Flush Cancel Online and Offline Source Quench and Resume
MIP Network Interface Responses	Buffer Grant Incoming Message Incoming Response Completion Event Reset Flush Complete

Network Driver Commands	Open Network Interface Read and Write Buffer I/O Control Register Callback Function Close Network Interface
Network Interface Library Functions	Initialize Network Interface Reset Network Interface Send LONWORKS Message and Wait for Completion Get Next Response Receive LONWORKS Message Send LONWORKS Response Local Network Management Command Handle Error

Note: Runtime copies of the MIP/P20 and MIP/P50 are subject to a runtime license fee.

Documentation

The following documentation is included with the MIP/P20 and MIP/P50 Developer's Kit. The documentation describes how to build a network interface and how to use the network interface to create a host application.

Document	Echelon Part Number
LONWORKS Host Application Programmer's Guide	078-0016-01
LONWORKS Microprocessor Interface Program (MIP) User's Guide	078-0017-01

Ordering Information

The MIP/P20 and MIP/P50 Developer's Kit is used with the LonBuilder Developer's Kit or NodeBuilder Development Tool to create a network interface.

The LTM-10 LonTalk Module can be used to develop a network interface without using the LONWORKS MIP/P20 and MIP/P50 Developer's Kit.

Product	Echelon Model Number
LONWORKS MIP/P20 and MIP/P50 Developer's Kit	23200
LTM-10 LonTalk Module	65100-100