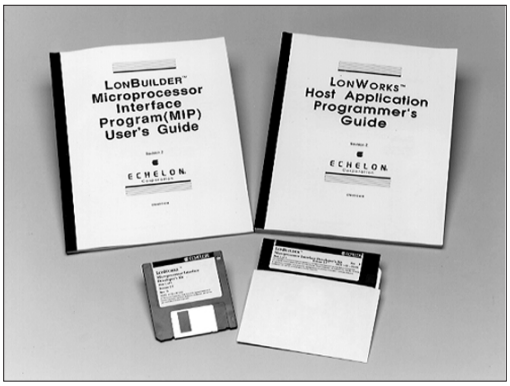
# LonWorks® MIP/DPS Developer's Kit
# Model 23210




- Adds a LonWorks interface to any host processor
- High-speed dual-ported RAM interface sends and receives hundreds of packets per second with minimum host overhead
- Network interfaces based on the MIP can be used with any host application
- Optional up-link interrupt 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 devicess 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.

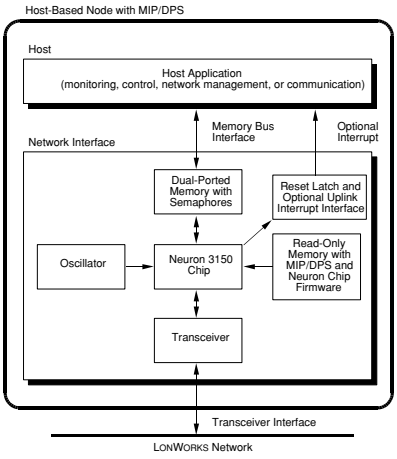
Several versions of the MIP are available. The MIP/DPS is the highest performance version and is designed for applications requiring the host to handle hundreds of packets per second while minimizing host overhead. The MIP/DPS is typically used with high-performance hosts such as 32-bit microprocessors, but may be used with any host. The Neuron Chip running the MIP/DPS communicates with the host processor using a dual-ported RAM with hardware semaphores.

Typical applications for the MIP/DPS include high performance LonWorks devices based on high-end microprocessors such as the PowerPC, 68040, 88000, 80486, or 80960.

The MIP enables the attached host to communicate on a LonWorks network using the LonWorks protocol.

The MIP/DPS is used to build a custom LonWorks network interface. A 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/DPS includes a Neuron Chip running the MIP/DPS firmware, a dual-ported RAM with hardware semaphores, transceiver, reset circuitry, and oscillator.

The following figure illustrates the components of a LonWorks device using a host processor and a network interface based on the MIP/DPS.




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 can be changed at any time, including its network variables, 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

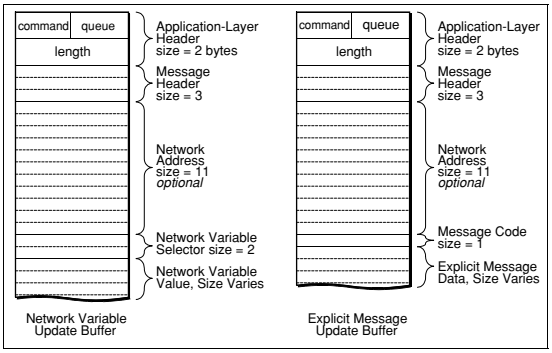
The MIP/DPS is delivered in a Neuron C library that extends the LonBuilder® or NodeBuilder® software to include system calls for the MIP. The developer creates a short Neuron C program that defines the initial buffer configuration for the network interface and calls the MIP/DPS system function. The LonBuilder or NodeBuilder tool is used to create a ROM image which is programmed into a PROM using a PROM programmer. The PROM is installed in a network interface that includes a Neuron 3150® Chip, dual-ported RAM, reset latch, oscillator, and communications transceiver. The dual-ported RAM must meet the memory timing specifications of the Neuron 3150 Chip, and must implement a hardware semaphore as specified in the *LONWORKS Microprocessor Interface Program (MIP) User's Guide*. Two devices that meet these specifications are the IDT 71342 and Cypress 78144 dual-ported RAMs.

The IO0 pin on the Neuron Chip can be used to generate an up-link interrupt. The MIP/DPS pulses IO0 whenever a packet is available for the host. The host can use this 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 specifications 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/DPS. This application illustrates how a host application can send and receive network variables and application messages using the network driver. The sample 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 `ni_send_msg_wait()` function in the network interface library. The following figure illustrates the application buffer format for network variable updates and application messages.



## Specifications

| MIP/DPS Throughput | | |
|---|---|---|
| Unacknowledged: | | |
| 1-byte message | 404 packets/sec | (3,235bps) |
| 8-byte message | 396 packets/sec | (25,325bps) |
| 32-byte message | 364 packets/sec | (93,184bps) |
| 228-byte message | 149 packets/sec | (272,141bps) |
| 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, 10 MHz; protocol overhead of 9 bytes per message.*

| | |
|---|---|
| MIP Network Interface Commands | Send Message<br>Local Network Management Command<br>Reset<br>Flush and Flush Cancel<br>Online and Offline |
| MIP Network Interface Responses | Incoming Message<br>Incoming Response<br>Completion Event<br>Reset<br>Flush Complete |
| Network Driver Commands | Open Network Interface<br>Read and Write Buffer<br>I/O Control<br>Register Callback Function<br>Close Network Interface |
| Network Interface Library Functions | Initialize Network Interface<br>Reset Network Interface<br>Send LONWORKS Message and Wait for Completion<br>Get Next Response<br>Receive LONWORKS Message<br>Send LONWORKS Response<br>Local Network Management Command<br>Handle Error |

*Note: Runtime copies of the MIP/DPS are subject to a runtime license fee.*

## Documentation

The following documentation is included with the MIP/DPS 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/DPS Developer's Kit is used with the LonBuilder Developer's Kit or NodeBuilder Development Tool to create a network interface. LonBuilder 2.2, NodeBuilder 1.5, or a newer version of either is required for the MIP/DPS.

| Product | Echelon Model Number |
|---|---|
| LONWORKS MIP/DPS Developer's Kit | 23210 |