MINI-MAX/51-E
Single Board Computer
IPX SERVER

Quick Start Guide

Document Revision: 1.01

Date: 7 June, 2003

BiPOM Electronics, Inc.

16301 Blue Ridge Road, Missouri City , Texas 77489
Telephone: 1-713-283-9970. Fax: Fax: 1-281-416-2806
E-mail: info@bipom.com
Web: www.bipom.com

Thank you for your purchase of the MINI-MAX/51-E Single Board Computer. This document describes how to establish communications from a PC to MINI-MAX/51-E using the IPX protocol. IPX protocol will give you very fast and reliable Ethernet connection for your embedded applications.

IPX (Internetwork Packet eXchange) basics

IPX network layer protocol was originally developed at XEROX Corporation and was made popular by Novell, Inc as the basic protocol in their Novel NetWare file server operating system. IPX is a datagram protocol used for connectionless communications. Higher-level protocols, such as SPX and NCP, are used for additional error recovery services.

Ethernet boards communicate using the MAC-addressing.

Note. A unique Ethernet MAC-address is factory-programmed into each MINI-MAX/51-E. The MAC-address is written to IC7, AT24C04A (please see MINI-MAX/51-E manual).

Multiple network boards can be installed in a Netware server, which is often done to improve network performance. For EACH network-card with its attached network-cable, a NET-number is assigned on the Netware server (in addition, each Netware server requires an internal NET-number for itself). These NET-numbers must be UNIQUE on the complete network. The complete Network-address of a system using IPX-protocol is the combination of NET-number and MAC-address.

When installing a Network card, Windows installs a default set of Clients and protocols, which includes the IPX/SPX protocol. Figure 1 shows where you can find installed Network protocols.

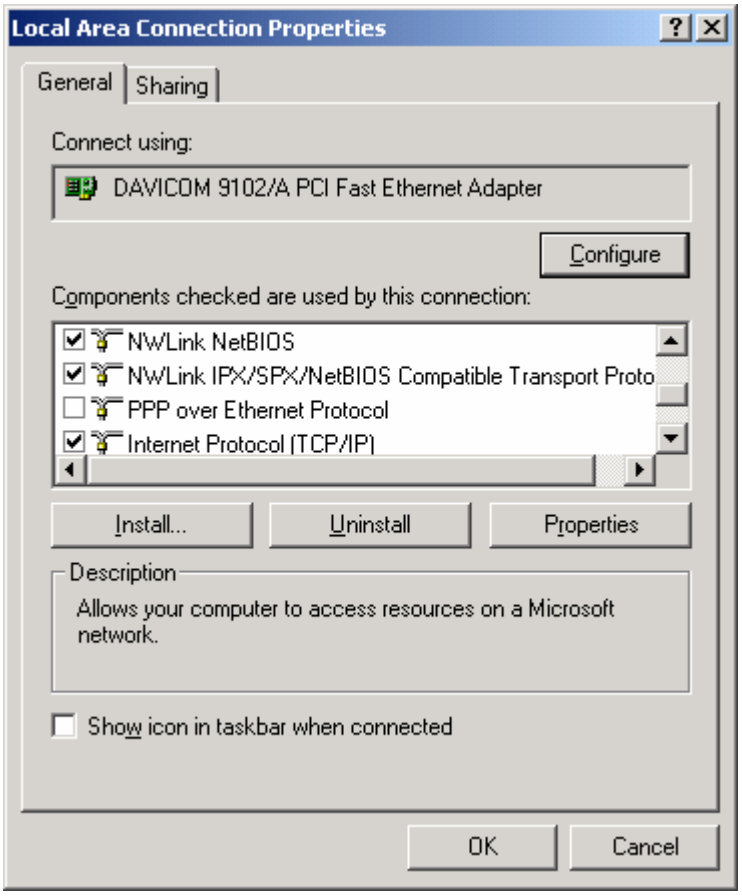


Figure 1.

Note: Using IPX to communicate between a Windows PC and MINI-MAX/51-E in a PC-to-PC network when there is no Novell or NT server on the network.

-Since there is no server, the IPX-Frame-Type is NOT defined on the network, so the default FRAME-Type setting of AUTO may NOT work! So, in such cases, you MUST configure the Frame-type to 802.2 yourself. Also, you can use 0 (ZERO) for NET-number. Figure 2 shows the properties of IPX/SPX protocol.

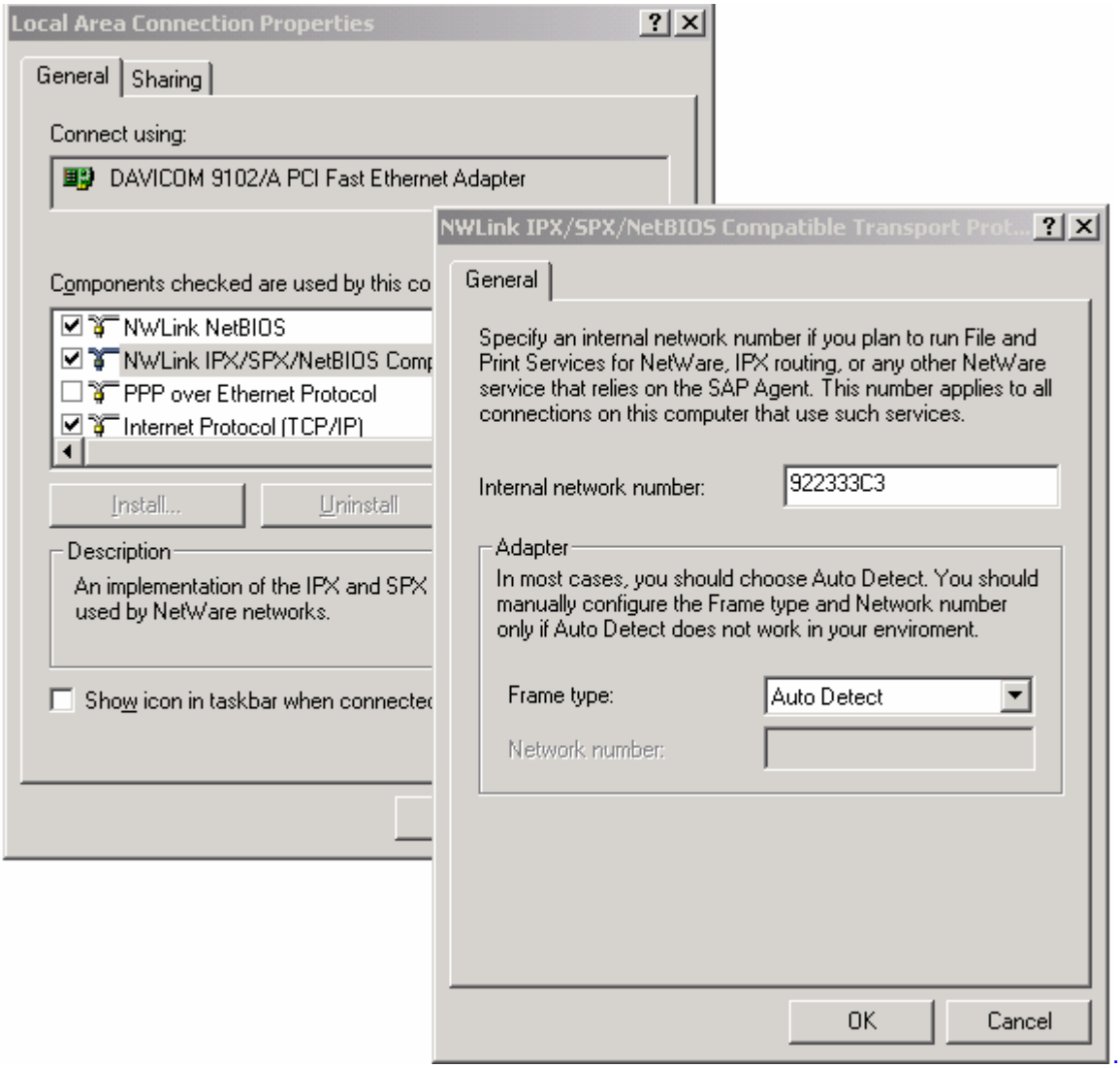


Figure 2

IPX Packet Structure

The IPX packet consists of two parts:

- *A* 30-byte IPX header*, which includes the network (NET-number), node (MAC-address), and socket addresses for both the destination and the source*
- *A* data *section, which often includes the header of a higher-level protocol, such as SPX*

The minimum IPX packet size (excluding the MAC header) is 30 bytes (IPX header only). Historically, the maximum size of routed IPX packets has been 576 bytes (IPX header and data). Until recently, all routed IPX packets were between 30 and 576 bytes.

Note. Current implementation of IPX server on MINI-MAX/51-E board limits the actual maximum packet size to 384 bytes:

14 bytes MAC –header
3 fixed bytes (0xE0,0xE0,0x03)
30 bytes IPX-header
337 bytes – data section

The IPX header is placed after the MAC header and before the data.

Structure of MAC header (14 bytes)

Destination MAC-address	(6 bytes)
Source MAC-address	(6 bytes)
Packet Type	(2 bytes)

0xE0,0xE0,0x03 (3 bytes)

Structure of IPX header (30 bytes)

Check Sum	(2 bytes)
Packet Length	(2 bytes)
Transport Control	(1 byte)
Packet Type	(1 byte)
Destination Network	(4 bytes)
Destination Node	(6 bytes)
Destination Socket	(2 bytes)
Source Network	(4 bytes)
Source Node	(6 bytes)
Source Socket	(2 bytes)

Data Section (337 bytes)

Check Sum - Packet integrity check.
The checksum is used by the NetWare SFT™ III™ software and NetWare 4 software. Older versions of NetWare did not use the IPX checksum and required that this field be set to 0xFFFF.

Packet Length - Length, in bytes, of the complete packet, which is the length of the IPX header plus the length of the data. The packet length is at least 30 bytes (for the IPX header).

Transport Control -Number of routers a packet has traversed on the way to its destination.

Packet Type -Type of service offered or required by the packet.

Destination Network - Number of the network to which the destination node is attached.

When a sending node sets this field to 0x0 (that is, 0x00000000), the destination node is assumed to be on the same network segment as the sending (or source) node.

Note. IPX does not have a broadcast network number (such as 0xFFFFFFFF).

In addition to network number 0, the numbers 0xFFFFFFFF and 0xFFFFFFFE are reserved for specific purposes. For this reason, they should not be assigned to any IPX network.

Destination Node - Physical address of the destination node (MAC-address).

Not all LAN topologies use the same size address field. A node on an Ethernet network requires all 6 bytes to define its address.
A node address of 0xFFFFFFFFFFFF (that is, 6 bytes of 0xFF) broadcasts the packet to all nodes on the destination network.
Destination Socket - Socket address of the packet destination process.
Sockets route packets to different processes within a single node. Novell reserves several sockets for use in the NetWare environment.
Note. IPX does not have a broadcast socket number (such as 0xFFFF).
Note. Current implementation of IPX server on MINI-MAX/51-E board uses 0x6200 socket to transmit the information.

Source Network - Number of the network to which the source node is attached. If a sending node sets this field to zero, the local network to which the source is connected is unknown. For routers, the rules that apply to the Destination Network field also apply to the Source Network field, except that routers can propagate packets that were received with this field set to zero.

Source Node - Physical address of the source node (MAC-address). Broadcast addresses are not allowed.
Source Socket - Socket address of the process that transmits the packet.
Processes communicating in a peer-to-peer fashion do not need to send and receive on the same socket number. On a network of workstations and servers, the server usually listens on a specific socket for service requests. In such a case, the source socket is not necessarily the same or even significant. All that matters is that the server reply to the source socket. For example, all NetWare file servers have the same socket address, but requests to them can originate from any socket number.
Source socket numbers follow the same conventions as those for destination sockets.
Note. Current implementation of IPX server on MINI-MAX/51-E board uses 0x6201 socket to receive the information.

Software Setup

Please download and install the latest release of 8051 Development System from http://www.bipom.com/8051dev.php Development software components include Micro-IDE, Micro C Compiler, simulator and debugger.

Micro-IDE is a Windows based Integrated Development Environment for micro-controller systems application development. Micro-IDE integrates essential components of software development including

- Multi File Editor with C and Assembly language syntax coloring
- Integration with toolkits including command line compilers, assemblers and linkers
- Project Manager
- Tools: Terminal program, Calculator, ASCII Chart
- Download Capability to target micro-controller boards including MINI-MAX/51-E
- Microcontroller simulator to simulate program execution without having the actual hardware

Micro C Compiler is powerful C compiler package for the 8051. Micro C includes all tools and utilities you need to develop 'C' and 'ASM' code for the 8051/52 family of processors.

Hardware Setup

Your PC should already be connected to your Ethernet hub. The included Ethernet cable cannot be used to connect the MINI-MAX/51-E board directly to the PC.
Note. You can use a BiPOM Electronics Ethernet cross cable (Part#: ECC-1) to connect MINI-MAX/51-E directly to a PC's network card.

1) Connect one end of the Ethernet cable to the Ethernet port of MINI-MAX/51-E (J6)
2) Connect the other end of the Ethernet cable to the hub
3) Connect one end of serial cable to an available serial (COM) port on your PC
4) Connect the other end of serial cable to MINI-MAX/51-E serial (COM) port (J2)
5) Connect the 6VDC Adapter to the MINI-MAX/51-E power connector (J5)

Testing the Hardware

BiPOM Electronics provides all the necessary software to test the hardware on a local network. TestIpx utility (which you can download from http://www.bipom.com/minimax51e.php) tests MINI-MAX/51-E on any local Ethernet network. TestIpx utility uses BROADCAST Ethernet MAC (0xFFFFFFFFFFFF), 0 (ZERO, 0x00000000) for NET-number, 0x6200 RX Socket and 0x6201 TX SOCKET.
MINI-MAX/51-E comes set from the factory as IPX server.

If the Flash memory of your board has been erased by your experiments, you should restore the IPX server:

1) Run Micro-IDE;
2) Open IPXSERV project (Project->Open Project menu option) under ..\ MicroC\BipomLib\MM51E\Servers\IPX_SERVER;
3) Download IPX server to the MM51E board ("Download" icon button)
   Figure 31 shows DOWNLOAD PROCESS.
4) Put board to RUN mode (change "Set Mode" button to GREEN LIGHT)

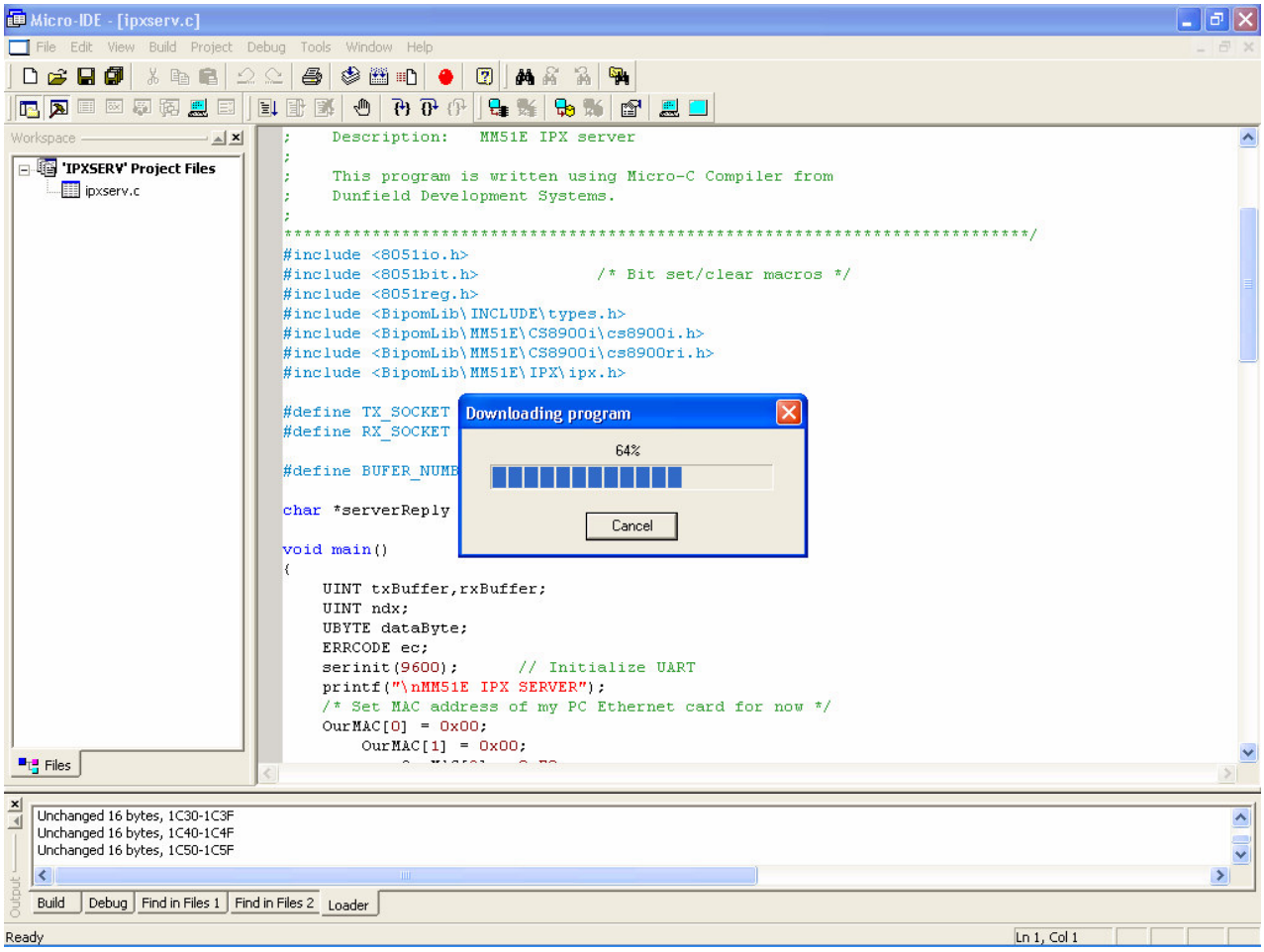
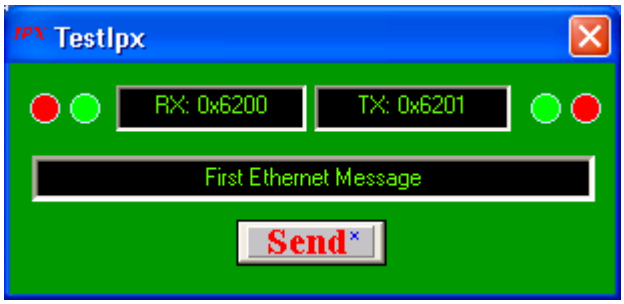
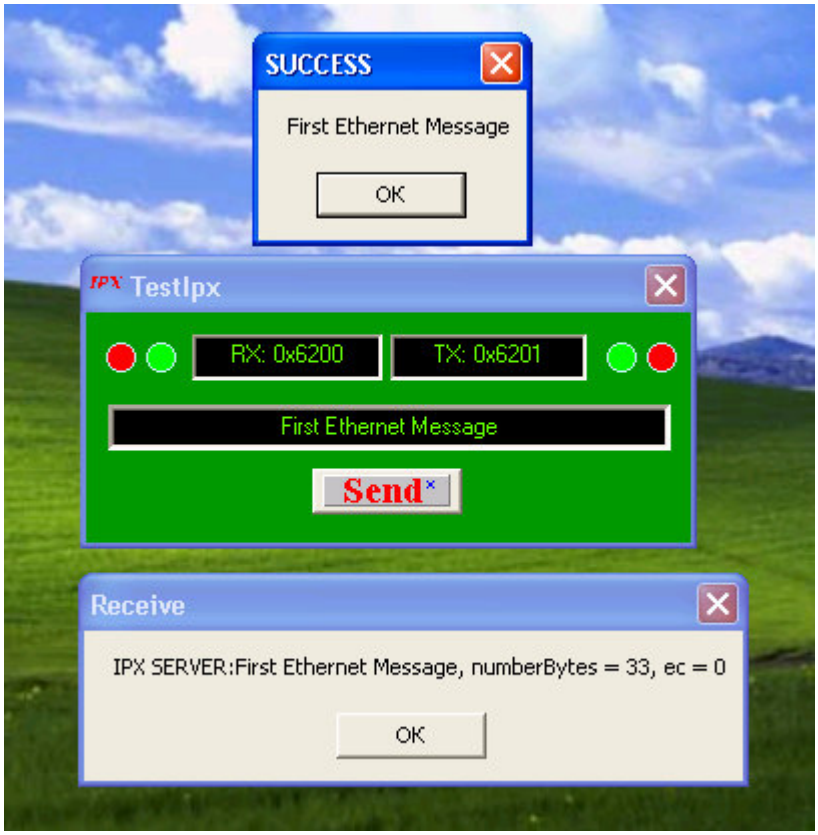


Figure 3

Now you can run TestIpx.exe:



You can edit “First Ethernet Message” string and send it to the MINI-MAX/51-E IPX server through Ethernet connection. MINI-MAX/51-E IPX server will echo to each message that you sent.
It will add a prefix at the beginning of your original message “IPX SERVER: First Ethernet Message”



If you are able to receive echo your hardware is OK.

If this is not the case, please make sure that IPX/SPX protocols are installed on your PC

How can you build your own IPX server?

It is very easy! BiPOM Electronics provides all sources of IPX server. Please download a full Visual C++ source of the TestIpx utility from http://www.bipom.com/minimax51e.php. This is the IPX client/server program that runs on the PC. This includes IPX.LIB and IPX.DLL. These libraries help design PC IPX applications that communicate with the MINI-MAX/51-E.

Also, the IPXSERV project is available under MicroC\BipomLib\MM51E\Servers\IPX_SERVER. This is the server MINI-MAX/51-E portion of IPX server. This project is very simple because most of the code has already been implemented for you inside the “mm51eti.lib” library.

This simple command line of the Linker (Project->Settings->Linker) has been pre-configured to allow you to use the IPX protocol:

{TOOLKITDIR}\slink {OBJFILES} l={TOOLKITDIR}\lib51 i=mm51eti.lib -q {TMPFILE}

For low-level access to the networking hardware on the MINI-MAX/51-E:

CS8900i project is available under MicroC\BipomLib\MM51E\CS8900i
IPX project is available under MicroC\BipomLib\MM51E\IPX.

Both projects are sources of the IPX layer, which is supported by “mm51eti.lib” library. You can see how it works and adapt to your specific needs.

How can you use IPX protocol?

IPX protocol is very simple and fast. IPX applications can communicate directly with other IPX nodes.
You can even run IPX network without server service using physical Ethernet connection only
(just connect MINI-MAX/51-E directly to a PC's network card).
It allows you to build very reliable and fast applications.
1.You can run multiple Remote View Systems in one PC with each Remote View connected to different MINI-MAX/51-E boards. Digital cameras should be connected to MINI-MAX/51E boards.
MINI-MAX/51-E board provides EXPANSION, KEYPAD, LCD connectors which can be used as an interface to digital camera (please see MINI-MAX/51-E manual). MINI-MAX/51-E + Digital Camera will act like Camera Server. Camera Server will translate a video through IPX connection.
2.You can build Remote Data Acquisition System on Ethernet network with many channels simultaneously.
3.You can build Remote Control System to drive some devices remotely.
BiPOM Electronics provides different Peripheral boards that can be used for such systems.
All the boards are compatible with MINI-MAX/51-E board.
To obtain more information on BiPOM Peripherals Boards please visit
http://www.bipom.com/periph_boards.php