User Manual

NCMZ-798 Emulation of PC1100 Local Area Network

Revision 1.2e

August 24, 1987

Revision History:

September 25, 1986	V1.2a
January 7, 1987	V1.2b
March 16, 1987	V1.2c
May 27, 1987	V1.2d
August 24, 1987	V1.2e

Table of Contents

1.0 Description

The NCMZ-798 is a microprocessor based I/O module that includes:

- Two RS-232 serial ports individually selectable from 150 Baud to 9.6K Baud
- 8 K of battery backed RAM
- 4 K of EPROM
- Built in routines allowing the module to communicate with the host PLC through the I/O bus.

Since the module is microprocessor based, a separate program can run independently from the PLC ladder program. This document describes the PC1100 Local Area Network communications package that can optionally be supplied with this module. The battery is not used with the PC1100 LAN package and can be either connected or disconnected.

The module gets configuration data from the I/O bus. Once the configuration data is loaded, the module can optionally talk to its host PLC through that PLC's RS-232 port. This optional serial link speeds data transfer from the module to the host PLC.

2.0 Installation

```
      +-----------+   I/O Rack
      |           |
+-----+-+         ++-+------------+
|       |         |A +--+         |                +-------------+
| PC700 |         |  |  |         |                |             |
|       +---------+B |  |         |                |  PC 1100    |
|       |         |  |  |         |                |             |
+-------+         +--+------------+                +------+------+
                /       |                                 |
NCMZ-798 -----       +--+--+                           +--+--+
module               |Modem|                           |Modem|
                     +-----+                           +-----+
                        |     4 wire switched carrier     |
              __________|_________________________________|______
                               multi-drop network
```

This module must be plugged into an unused slot of the host PLC's I/O rack. The host PLC can be a PC700,900,1500 or 1700 processor. The module communicates with the host over Input Register (IR) and Output Register (OR) pairs.

The sample program listed later shows how the module communicates with the host PLC using a short ladder program.

Before plugging the module into the I/O rack, determine the baud rate to be used on the PC1100 network, as well as, (optional) the second serial port of the 798 module. To set the baud rate, disassemble the 798 module by following the instructions found in the NCMZ-798 instruction leaflet. Set the appropriate jumpers on the module to select the required baud rate. Note that the second port (B) is ONLY used if:

1. The host PLC is a PC700 or a PC900, and
2. higher throughput from the module to the PLC is desired.

If a direct connection is made from the B port of the 798 module to the program loader port of the PLC, some of the register addresses change.

For more information on the hardware setup of this module, refer to the NCMZ-798 IL.

3.0 Loading the Host PLC ladder program

Since the module has no DIP switches to set (other than baud rate), all configuration information must be received from the host PLC. This configuration information is stored in six Holding Registers (HR's) in the host PLC.

These six registers are:

HR 1	Configuration register 1
HR 2	Configuration register 2
HR 3	Source Register
HR 4	Destination Register
HR 5	Transmit Delay / Data Format
HR 6	RTS Delay[1] / Receive time out

Although HR 1 through HR 6 are shown, the six registers can be ANY contiguous registers accessible in the host PLC using Table to Register and Register to Table functions. They do NOT have to be HR1 through HR6.

To change to a different group, simply change the TABLE END Holding Register in both the R-T and T-R functions.

Load the ladder program found in Appendix A (if using a PC700 or PC900, Appendix C if using an HPPC) into the host PLC. This ladder program will allow the 798 module to access these registers and configure itself. Next load these six registers with the proper values for your application.

<u>Memory Map of Communications Registers Data Table</u>: host PLC

```
Start of TABLE  +-------+
                !-------!       HR0001  Configuration Register 1
                !-------!       HR0002  Configuration Register 2
                !-------!       HR0003  Source Register
                !-------!       HR0004  Destination Register
                !-------!       HR0005  TX Delay/Data Format
                !-------!       HR0006  RTS Delay[2]/RX time out
                !       !
                !       !       Data Registers
                /       /       These registers are used if the
                /       /       optional program loader port is NOT
End of TABLE    +-------+       used.
```

4.0 Description of Configuration Registers

---

[1] The upper byte RTS delay is only found on PROM's dated after 5/10/87

[2] See footnote above.

Configuration Register 1: Control

```
16 15 14 13 12 11 10 9  8  7  6  5  4  3  2  1
 !  !  !  !          !  [  Error Register     ]
 !  !  !  !          +  Error bit.  This bit is set by the module
 !  !  !  !             whenever the module senses an error.  Refer
 !  !  !  !             to Appendix B for a complete description of
 !  !  !  !             the errors.
 !  !  !  +-----------  Read if 1, write if zero
 !  !  +--------------  Start.  Reload configuration data and
 !  !                   execute master to slave communications.  The
 !  !                   798 module clears this bit when the task is
 !  !                   completed.  If a slave, this bit and all of
 !  !                   the configuration data is only read once.
 !  +-----------------  RS-232 communications over port B is allowed
 !                      is this bit set to 1.
 +--------------------  Master if bit is set to 1, slave if zero.
```

Note that only bits 16 through 13 are set or cleared by the programmer. Bits 12 through 10 are not used and bits 9 through 1 are modified by the 798 module through the ladder program listed in Appendix A.

Configuration Register 2: Address/Count Register

```
16 15 14 13 12 11 10 9  8  7  6  5  4  3  2  1
[ # of registers     ]  [  Address of drop    ]
```

If this module is a Master:

The upper eight bits represent the number of registers to read or write and the lower eight bits represent address of the slave to read or write to. The maximum address is 127 decimal and the maximum number of registers is 64. The minimum address is 1 and the minimum number of registers is 1.

If this module is a Slave:

The upper eight bits are ignored and the lower eight bits represent the modules address on the network. Once this address is read, the only way to reset a new address is to cycle power to the 798 module or cycle the host PLC keyswitch from Run to Stop and back to Run.

Configuration Register 3: Source Register

Where does the information originate? This pointer shows the FIRST register to read.

Configuration Register 4: Destination Register

Where is the information placed? This pointer shows the FIRST register to receive data.

Information always moves from the Source to the Destination register. Refer to the programming examples in Appendix A for more information on the proper method of programming the unit.

Configuration Register 5: Transmit Delay / Data Format Register

Upper byte - Transmit delay

The module can be programmed to delay transmitting a message or response from 0.0 to 25.5 seconds (00 to FF hex) after it raises its Request to Send lead (RTS).

Lower byte - Data Format

The allowable data formats are:

11	8 data bits, 2 stop bits, no parity
15	8 data bits, 1 stop bit, no parity
19	8 data bits, even parity
1D	8 data bits, odd parity (Numa-Logic Default)

Baud rate is set by jumpers on the NCMZ-798.

Configuration Register 6: RTS Delay/Receive Time Out

Upper byte - RTS delay

The upper byte determines how many milliseconds after the last byte is transmitted does the RTS drop low. Typical values vary depending on the baud rate. 9600 - 02, 4800 - 03, 2400 - 05, 1200 - 0A, 300 - 23. All values are in hex.

Lower byte - Receive time out

The module will set the error bit of Configuration Register 1 if a response is not heard within 0.5 seconds. Since communications via the I/O bus sometimes requires more time, this register allows programming the module error time out for up to 25.5 seconds.

5.0 RS-232 Control Lines and Cabling

RS-232 Control Lines

pin 5	RTS	output	Goes high at beginning of transmission
pin 4	CTS	input	Resets the internal receive pointer. Connected to LAN sync pulse.
pin 20	DSR	input	Indicates the modem is ready
pin 6	DTR	output	pulled to +12 volts
pin 8	DCD	output	pulled to +12 volts
pin 2	RXD	input	Incoming data to module
pin 3	TXD	output	Outgoing data from module

<u>Wiring Diagrams</u>

<u>Direct Connect</u>: Cable C

```
NCMZ-798 A (male)       PC1100 B (female)
        2----------<----2
        3---------->----3
        4----------<--+-4
                      +-5
        5-+--------->----6
       20-+
        7---------------7
```

<u>Modem/Line Driver Connection NCMZ-798</u>: Cable A

```
NCMZ-798 A (male)       LD400MP or Racal-Vadic 1251K (male)
        2---------------3
        3---------------2
        4---------------8
        5---------------4
        7---------------7
       20---------------5
```

<u>Modem/Line Driver Connection PC1100</u>: Cable A*

```
PC1100 B (female)       LD400MP or Racal-Vadic 1251K (male)
 TxD   2---------------2
 RxD   3---------------3
 RTS   4---------------4
 CTS   5---------------5
 DSR   6---------------8
 Gnd   7---------------7
 DTR  20--------------- n.c.
```

<u>Modem/Line Driver Connection PC1100</u>: Cable F

Note: Numbers in parenthesis are the pin numbers of the PC1100 program loader RS-232 port if the NL-1075 or Quartech module splitters are not used.

```
PC1100 B (female)       LD400MP or Racal-Vadic 1251K (male)
 TxD (14) ---------------2
 RxD (16) ---------------3
 RTS (19) ---------------4
 CTS (13) ---------------5
 DSR (12) ---------------8
 Gnd ( 7) ---------------7
 DTR (22) ---------------5
```

<u>NCMZ to NCMZ direct - no modem</u>: Cable B

```
NCMZ slave (male)       NCMZ master (male)
        2------------------3
        3------------------2
        7------------------7
        4----------------+-5
                         +-20
        5-+----------------4
       20-+
```

<u>Optional Cable between NCMZ and PC700/900</u>: Cable D

```
NCMZ (male)             PC700/900 Program Loader port (female)

        2------------------2
        3------------------3
        7------------------7
        6-+              +-4
       20-+              +-6
                         +-20
        4-+
        5-+
```

Note: In order for the NCMZ798 to communicate with the PC700/900's RS-232 port:

1- The PC700/900 must have been powered up and the cable connected to the NCMZ module before the NCMZ module completes its initial power-on configuration (approx. 10 seconds).

2- The 15th bit of configuration register one in the PC700/900 must have been previously set before the NCMZ module has completed its power-up configuration.

3- If the NCMZ module does not seem to communicate via the RS-232 port, favoring the I/O bus instead, cycle the processor keyswitch from Run to Stop and back to Run. This will force the NCMZ module to perform a power-on configuration, rereading the configuration register one and again attempting to communicate over the optional RS-232 program loader connection.

For more types of cables, see the drawings at the end of this document.

<u>Using Modems or Line Drivers</u>

```
+--------+
|PC1100  |
|Master  |
+---+----+
    |
+---+----+
|   TX +|--------------+-----------------------+--------------
|   TX -|-----------+--|-------------------+--|----------------
|   RX +|--------+--|--|----------------+--|--|------------------
|   RX -|-----+--|--|--|-------------+--|--|--|--------------------
+--------+     |  |  |  |             |  |  |  |
4 wire       +-+--+--+--+-+          +-+--+--+--+-+
half duplex |TX TX RX RX |          |TX TX RX RX |
modems or   | -  +  -  + |          | -  + -   + |
line drivers+-----+------+          +-----+------+  I/O Cable
                  |                       |      +-------+
       +-------+  | A                   A | +-----++    ++------+
       |      +---+                       +-+      +-----+       |
       |PC1100|                             |NCMZ   |  D |PC700 |
       |Slave |                             |798    |    |Slave |
       | #1   |                             |       |    | #2   |
       +-------+                            +-------+    +-------+
                                            This slave is shown
                                            connected to the NCMZ
                                            module using the optional
                                            serial port connection.
```

6.0 Programming Examples

Example 1: Read 4 registers from the slave (HR100-HR103) and place in the master PLC at HR200-HR203. Assume the master NCMZ-798 module communicates with the master PLC via the optional program loader port connection. Assume further that the communications ladder as shown in Appendix A is used (HR0001 though HR0006 are the control registers.) Assume the slave address is 1. The slave is a PC1100. The baud rate is 1200.

Master

HR0001	Control	F000 hex	
HR0002	Address/Count	0401 hex	(# regs, address)
HR0003	Source Register	0063 hex	(99 decimal)
HR0004	Destination Register	00C7 hex	(199 decimal HR200)
HR0005	TX delay/Data format	051D hex	
HR0006	RTS Delay/RX delay time out	0A0B hex	

Note that the destination register is 199. 199 is the absolute memory address in the PC700/900/1100 of HR0200. When communicating over the optional program loader port connection, absolute addresses are used instead of offsets in the communications register data table defined in ladder. Refer to example 3 for more information on communicating between the NCMZ-798 module and its host PLC when the optional program loader port connection is not used. When this connection is not used, all communications between the NCMZ-798 module and its host PLC are done over the I/O bus.

```
Master PLC                                 Slave PLC

HR0001
HR0002
HR0003-0063 hex  This register points to source ----+ (63=99 dec.)
HR0004-00C7 hex  This points to destination --+     ! (C7=199 dec.)
HR0005                                        !     !
HR0006  +-------------------------------------+     !
:   :   !                                +------------+
HR0200 -+                                +-HR0100
HR0201                                     HR0101
HR0202                                     HR0102
HR0203                                     HR0103
```

Note that the pointers to the start of the source and the destination tables equals one less than the starting HR number.

NOTE!: The slave PLC (either PC1100 or NCMZ-798) <u>MUST</u> be configured for a transmit delay time of at least 200 mS. Use the configure port command in the 1100 to configure at least 200 mS of delay after receiving a message from the network before sending a response. If the PC1100 acknowledges sooner than this, the NCMZ-798 module may not be ready and ignore the response. If the PC1100 is seen (using a breakout box) to be answering the 798 module, but the 798 module signals that no response is heard (8A code), try more slave delay.

If the LD400MP line driver is used at 9600 baud or faster, additional parallel termination impedance may be necessary. Try using 1200 baud as an experiment with the normal termination impedance first to verify that everything else is correct.

Example 2: The slave is a PC700 with an NCMZ-798 module set for address 1. The slave NCMZ-798 module will communicate with its slave PLC over the optional program loader port connection. The baud rate is 9600.

Slave

HR0001	Configuration register 1	6000 hex
HR0002	Configuration register 2	0001 hex
HR0003	Source Register	0000
HR0004	Destination Register	0000
HR0005	TX delay/Data format	051D hex
HR0006	RX delay time out	02FF hex

Note that if the drop is configured as a slave, no value needs to be placed in the source or destination registers.

Example 3: The slave is a PC1100. The master is an NCMZ-798 module connected to a PC700. The optional program loader port connection between the NCMZ-798 module and the PC700 is NOT used. Program the module to read 10 registers from the slave (HR100-HR109) and place this information in the master PLC's HR200 through HR209. As with the previous examples, the master PLC is assumed to be using the ladder diagram shown in Appendix A to communicate with the NCMZ-798 module. The slave's address is assumed to be 1. The baud rate is 1200.

HR0001	Configuration register 1	B000 hex
HR0002	Configuration register 2	0A01 hex
HR0003	Source Register	0063 hex (99 decimal)
HR0004	Destination Register	00C7 hex (199 decimal)
HR0005	TX delay/Data Format	051D hex
HR0006	RX delay time out	0A0B hex

Note that in this example the destination register is also listed as 199. Same number, but a different reason.

When communicating via the I/O bus, information is transmitted to and from the NCMZ-798 module using TABLE TO REGISTER and REGISTER TO TABLE functions. The module cannot address memory in the master PLC directly. It can only access one block of registers called the communications registers.

The first six registers of this table are the configuration registers. The rest of the table is used to transfer data to and from the master PLC.

During a read (info from slave brought to master) the destination register points to a place in the master PLC's communication register table where the information will be placed. That location is an offset from the start of the master PLC's communications registers. The program listed in Appendix A shows that the table ends a HR0256 and is 256 registers long. This means that the start of the table is HR0001.

An offset of 0000 hex will place the information (in this example) into HR0001. An offset of 0006 will place data into HR0007 (the first register available after the configuration registers.

7.0 Special Considerations for Radio Modem Applications

A common type transmission media for longer distances is VHF/UHF (very high frequency/ultra high frequency) radios. Special modems are available that contain all the circuitry necessary to not only modulate and demodulate the data and tones, but also to actually key the radio's PTT (push to talk) circuitry.

However, for those installations that are using ordinary modems and radios, this section will give some guidelines.

Refer to the attached drawings at the end of this document for common wiring examples using RF (radio frequency) modems.

The major differences between a wireline based system and a radio based system are:

1. The programmable controllers on the network must also key the radio before sending data.

2. The system designer must incorporate additional delays in the communication sequence to account for the time lag introduced by the radio transmitter delay plus the additional holdup caused by the receiver squelch circuitry opening.

Keying the Radio

To key the radio, simply tie the programmable controller or 798 module's RTS (Request to Send) lead to a Rangor[3] RTS/PTT module. This module then closes a relay contact which is used to key the transmitter.

RTS/CTS Delays

When the master on the network wants to send a message, it must first key the transmitter connected to it. Since the transmitter does not activate immediately, any data sent from the programmable controller will be lost. Further, the remote radio's receiver cannot instantly process a signal when it first hears it. That receiver must first open it's squelch circuit.

For these two reasons, more delay must be incorporated in two places:

1. The master's transmit delay (RTS/CTS turnaround delay)

2. The slave's transmit delay (RTS/CTS turnaround delay)

---

[3] Rangor 8204 Doe Ave, Visalia CA. 93291 209-738-0849

If the master or slave is an NCMZ-798 then this extra delay is added to the upper byte of the fifth register in the configuration table, namely the Transmit Delay/Data Format register.

If the master or slave is a PC1100, then the delay is added by using the Configure Port command and selecting port "3". Refer to the PC1100 advanced functions manual describing this function in more detail.

Connecting Modem to Radio

The output of a 4 wire modem is connected to the speaker and microphone leads of the radio. Since the signal exiting the modem is connected to the microphone, certain levels must be matched with a resistor. Experience has shown that this value is around 100K ohms.

The connection from the radio speaker jack to the modem's receive line has not required any resistors for level or impedance matching. It seems to be only necessary to properly set the radio volume control.

Experience has shown that the radio's squelch can be either left open or adjusted to squelch the receiver when no signal is present. Apparently the modem has the ability to discriminate the white noise from the open squelch and not interpret this as data.

Drawings are included that show the proper wiring and DIP switch settings when the following equipment is used:

- Racal-Vadic[4] 1251K 4 wire half duplex modem (Bell 202T)
- Standard Communications[5] 768L UHF radio (remotes)
- Standard Communications FX60U UHF radio (master)
- PC1100 or NCMZ-798-LAN Slaves
- NCMZ-798-LAN network master

---

[4] Racal-Vadic 1525 McCarthy Blvd, Milpitas, CA. 95035
408-946-2227 — DISCONNECTED 922-3350 GARY FARIGONO

[5] Standard Communications, P.O. Box 92151, Los Angeles, CA. 90009-2152, 213-532-5300

8.0 Designing Networks with Master Rebroadcasters

When using modems, the limiting factor for range and data rates is based on the type of modem or line driver used. In some systems the network must be configured for more than the maximum recommended number of modems on a single line.

In these cases the network must be expanded to multiple lines, each carrying the same data at the same time.

Since the PC1100 network only conveniently supports one master, that one master must be able to rebroadcast simultaneously its message on multiple lines. Likewise, it must be able to listen to each of the multiple lines for a response from a slave, since the master will not know from which line the response will be heard.

To do all this, Black Box[6] Data Broadcast Units (DBU) are used. The DBU's take a single RS-232 signal and rebroadcast it on up to 8 other RS-232 ports. Similarly, the receive signals from all 8 channels is OR'ed and presented to the single master port.

In this way the network can be expanded to eight lines. If more than eight networks are required, a combination of DBU's and back-to-back modems or line drivers can be used.

A diagram is included with this document to show how a master rebroadcaster could be wired into the system.

---

[6] Black Box Corporation, P.O. Box 12800, Pittsburgh, PA., 15241, Support:412-746-5565, Order:412-746-5530

9.0 Designing a Redundant Master System

Since the PC1100 LAN is designed as a master-slave system, it may desirable to design an automatic transfer to a redundant master should the primary network master fail.

To do this, the redundant master must sense when the primary master has failed. Also, it may be required that the redundant master not only detect that the primary master has failed, but also to know something about the state of the network when the primary master failed.

Detecting that the primary master has failed can easily be accomplished with an output of the primary master connected to an input of the redundant master. When the primary master fails, the primary master would remove the signal from the output, signalling the redundant master to take over. The redundant master could seize the network using either an RS-232 fall back switch from Black Box, or else a custom box using a 3PDT relay that switches the TxD, RxD, and RTS leads of the master modem from the primary master to the redundant master.

This will accomplish the failover, but it does not show how to transfer network state information from the primary master to the redundant master. Before the transition occurs, the redundant master may be required to know something about the state of the network.

This can be accomplished by making the redundant master a slave on the network as well as a master. Note that a network master based on the NCMZ-798 module can be configured both as a master and as a slave just be using additional 798 modules. One module acts as the slave, receiving periodic updates from the primary network master. The other 798 module is the redundant master module. This module does nothing until the control is transferred over to the redundant master. Since the redundant master has been receiving periodic updates from the primary master, the redundant master can pick up control more smoothly.

Note that a PC1100 based master cannot be used as a redundant master if that master must also communicate with the network as a slave. This is because the PC1100 has no convenient method of switching between a slave and a master.

A drawing included with this document shows a possible configuration using redundant network masters.

10.0 HPPC Communications

The HPPC processor can provide several thousand Holding Registers and therefore makes an ideal "mailbox" for a network.

To connect the HPPC processor to the 1100 LAN network as either a master or a slave, just plug an NCMZ-798 module into the HPPC's I/O rack. A program must be loaded in the HPPC so that information can be passed between the module and the HPPC.

Note that since the HPPC MUST communicate via the I/O bus, a maximum of 256 unique registers can be transferred using the source and destination pointers of the NCMZ configuration table. Programming techniques can be used to extend this, however, up to the maximum number of registers allowed by the HPPC.

Refer to Appendix "C" for a listing of the "handshaking" program that must be loaded into the HPPC processor in order for communication to occurr with the NCMZ module.

Appendix A: Communications Ladder for Host PLC

```
R !BP0009                                                  !
U !HR0001                           +------------+UC0101!
N !-] [-----------------------------!            !--( )-!
G !                                 !PRESET      !      !
  !UC0101                           !  9999      !      !
# !-]/[-----------------------------!            !      !
  !#1                               !ACTUAL      !      !
  !                                 !HR0102      !      !
  !                                 !            !      !
1 !                                 +------------+      !
  ! This rung counts the number of communications errors !
  ! heard (number of times BP9 of HR1 is set).          !
  !  1,#1,                                              !
  !                                                     !
  !                                                     !
R !BP0009                                               !
U !HR0001                           +------------+MV0002!
N !-] [-----------------------------!            !--( )-!
G !                                 !SOURCE      !      !
  !                                 !HR0001      !      !
# !                                 !            !      !
  !                                 !DESTINATION !      !
  !                                 !HR0101      !      !
  !                                 !            !      !
2 !                                 +------------+      !
  ! This rung moves the error code into the temporary   !
  ! register HR101                                      !
  !  #2,                                                !
  !                                                     !
  !                                                     !
R !CR0200                                         CR0200!
U !-]/[--+----------------------------------------( )-!
N !#3    !                                              !
G !      !       This dummy rung produces a coil that is !
  !CR0200!       always on.                             !
# !-] [--+                                              !
  !#3                                                   !
  !                                                     !
  !                                                     !
3 !                                                     !
  !                                                     !
  !                                                     !
  !22,22,22,21,19,18,17,16,15,14,13,12,9,8,7,7,7,6,4,3,3,#3
  !
```

```
  !                                                                 !
R !CR0200                                      +------------+MV0201!
U !-] [----------------------------------------!            !--( )-!
N !#3                                          !SOURCE      !      !
G !                                            !HR0257      !      !
  !                                            !            !      !
# !                                            !DESTINATION !      !
  !                                            !OR0002      !      !
  !                                            !            !      !
  !                                            +------------+      !
4 ! This function writes HR257 (zero) into the output              !
  ! register.                                                      !
  !                                                                !
  !  #20,#19,#12,#9,#8,#5,#4,                                      !
  !                                                                !
  !                                                                !
R !BP0015                                                   SK0201!
U !HR0258                                                     0004!
N !-]/[-------------------------------------------------------( )-!
G !#6                                                              !
  !                                                                !
# ! If command not a write, skip                                   !
  !                                                                !
  !                                                                !
  !                                                                !
5 !                                                                !
  !                                                                !
  !                                                                !
  !  #20,#19,#12,#9,#8,#5,#4,                                      !
  !                                                                !
  !                                                                !
R !                                                         BC0015!
U !CR0200                                                   HR0258!
N !-] [-------------------------------------------------------( )-!
G !#3                                                              !
  !                                                                !
# !                                                                !
  !                                                                !
  !                                                                !
  !                                                                !
6 !                                                                !
  !                                                                !
  !                                                                !
  !  #6,                                                           !
  !                                                                !
```

```
    !                                                          !
R   !CR0200                              +------------+RT0202!
U   !-]/[--------------------------------!            !--( )-!
N   !#3                                  !TABLE LENGTH!      !
G   !                                    !  0256      !      !
    !CR0200                              !            !      !
#   !-] [--------------------------------!TABLE END   !      !
    !#3                                  !HR0256      !      !
    !                                    !            !      !
    !CR0200                              !POINTER     !      !
7   !-] [--------------------------------!HR0258      !      !
    !#3                                  !            !      !
    !                                    !SOURCE      !      !
    !                                    !IR0002      !      !
    !                                    !            !      !
    !   Write data from 798 module to    +------------+      !
    !   data table in host PLC                               !
    !  #7,                                                   !
    !                                                        !
    !                                                        !
R   !CR0200                              +------------+MV0201!
U   !-] [--------------------------------!            !--( )-!
N   !#3                                  !SOURCE      !      !
G   !                                    !IR0002      !      !
    !                                    !            !      !
#   !                                    !DESTINATION !      !
    !                                    !OR0002      !      !
    !                                    !            !      !
    !                                    +------------+      !
8   !                                                        !
    !  Echo data back to 798 module for integrity check      !
    !                                                        !
    !  #20,#19,#12,#9,#8,#5,#4,                              !
    !                                                        !
    !                                                        !
R   !                                                  SK0201!
U   !CR0200                                              0013!
N   !-] [--------------------------------------------------( )-!
G   !#3                                                      !
    !                                                        !
#   !                                                        !
    !                                                        !
    !                                                        !
    !                                                        !
9   !                                                        !
    !                                                        !
    !                                                        !
    !  #20,#19,#12,#9,#8,#5,#4,                              !
    !                                                        !
```

```
R !BP0015 BP0016                                          SK0204!
U !IR0002 IR0002                                            0012!
N !-] [----] [--+---------------------------------------( )-!
G !             !                                               !
  !BP0015 BP0016!                                               !
# !IR0002 IR0002!                                               !
  !-]/[----]/[--+    Check if command from 798 module           !
  !             !    is valid, otherwise skip next 12           !
1 !BP0014 BP0006!    rungs.                                     !
0 !IR0002 IR0002!                                               !
  !-] [----] [--+                                               !
  !             !                                               !
  !BP0014 BP0006!                                               !
  !IR0002 IR0002!                                               !
  !-]/[----]/[--+                                               !
  !             !                                               !
  !BP0013 BP0005!                                               !
  !IR0002 IR0002!                                               !
  !-] [----] [--+                                               !
  !             !                                               !
  !BP0013 BP0005!                                               !
  !IR0002 IR0002!                                               !
  !-]/[----]/[--+                                               !
  !             !                                               !
  !BP0012 BP0004!                                               !
  !IR0002 IR0002!                                               !
  !-] [----] [--+                                               !
  !                                                             !
  !  23,#11,#10,                                                !
```

```
  !                                                                  !
R !BP0012 BP0004                                               SK0204!
U !IR0002 IR0002                                                 0011!
N !-]/[----]/[--+--------------------------------------------( )-!
G !             !                                                    !
  !BP0011 BP0003!          Continue checking command from       !
# !IR0002 IR0002!          798 module and skip next 11          !
  !-] [----] [--+          rungs if invalid.                     !
  !             !                                                    !
1 !BP0011 BP0003!                                                    !
1 !IR0002 IR0002!                                                    !
  !-]/[----]/[--+                                                    !
  !             !                                                    !
  !BP0010 BP0002!                                                    !
  !IR0002 IR0002!                                                    !
  !-] [----] [--+                                                    !
  !             !                                                    !
  !BP0010 BP0002!                                                    !
  !IR0002 IR0002!                                                    !
  !-]/[----]/[--+                                                    !
  !             !                                                    !
  !BP0009 BP0001!                                                    !
  !IR0002 IR0002!                                                    !
  !-] [----] [--+                                                    !
  !             !                                                    !
  !BP0009 BP0001!                                                    !
  !IR0002 IR0002!                                                    !
  !-]/[----]/[--+                                                    !
  !                                                                  !
  !  23,#11,#10,                                                     !
  !                                                                  !
  !                                                                  !
R !CR0200                                  +------------+MV0201!
U !-] [------------------------------------!            !--( )-!
N !#3                                      !SOURCE      !      !
G !                                        !IR0002      !      !
  !                                        !            !      !
# !                                        !DESTINATION !      !
  !                                        !HR0258      !      !
  !                                        !            !      !
1 !                                        +------------+      !
2 ! Save command and place in pointer register                   !
  !                                                                  !
  !                                                                  !
  !  #20,#19,#12,#9,#8,#5,#4,                                        !
  !                                                                  !
```

```
   !                                                   !
R  !                                            BC0009!
U  !CR0200                                      HR0258!
N  !-] [-----------------------------------------( )-!
G  !#3                                                 !
   !          Reset checksum - part of specified address     !
#  !                                                   !
1  !                                                   !
3  !                                                   !
   !  #13,                                             !
   !                                                   !
   !                                                   !
R  !                                            BC0010!
U  !CR0200                                      HR0258!
N  !-] [-----------------------------------------( )-!
G  !#3                                                 !
   !                                                   !
#  !                                                   !
1  !                                                   !
4  !                                                   !
   !  #14,                                             !
   !                                                   !
   !                                                   !
R  !                                            BC0011!
U  !CR0200                                      HR0258!
N  !-] [-----------------------------------------( )-!
G  !#3                                                 !
   !                                                   !
#  !                                                   !
1  !                                                   !
5  !                                                   !
   !  #15,                                             !
   !                                                   !
   !                                                   !
R  !                                            BC0012!
U  !CR0200                                      HR0258!
N  !-] [-----------------------------------------( )-!
G  !#3                                                 !
   !                                                   !
#  !                                                   !
1  !                                                   !
6  !                                                   !
   !  #16,                                             !
```

```
    !                                                        !
    !                                                        !
R   !                                                 BC0013!
U   !CR0200                                           HR0258!
N   !-] [------------------------------------------------( )-!
G   !#3                                                      !
    !                                                        !
#   !                                                        !
1   !                                                        !
7   !                                                        !
    !  #17,                                                  !
    !                                                        !
    !                                                        !
R   !                                                 BC0014!
U   !CR0200                                           HR0258!
N   !-] [------------------------------------------------( )-!
G   !#3                                                      !
    !                                                        !
#   !                                                        !
1   !                                                        !
8   !                                                        !
    !  #18,                                                  !
    !                                                        !
    !                                                        !
R   !CR0200                            +-------------+MV0201!
U   !-] [------------------------------!             !--( )-!
N   !#3                                !SOURCE       !       !
G   !                                  !IR0002       !       !
    !                                  !             !       !
#   !                                  !DESTINATION  !       !
    !                                  !OR0002       !       !
    !                                  !             !       !
1   ! Echo command to module           +-------------+       !
9   !                                                        !
    !                                                        !
    !                                                        !
    !  #20,#19,#12,#9,#8,#5,#4,                              !
    !                                                        !
    !                                                        !
R   !BP0015                                           SK0201!
U   !HR0258                                             0002!
N   !-] [------------------------------------------------( )-!
G   !#6                                                      !
    !                                                        !
#   ! If command is a write, skip                            !
2   !                                                        !
0   !                                                        !
    !  #20,#19,#12,#9,#8,#5,#4,                              !
```

```
  !                                                           !
R !                                                     BC0016!
U !CR0200                                               HR0258!
N !-] [---------------------------------------------------( )-!
G !#3                                                         !
  !                                                           !
# !                                                           !
2 ! Reset read command bit                                    !
1 !                                                           !
  !  #21,                                                     !
  !                                                           !
  !                                                           !
R !CR0200                                  +-------------+TR0203!
U !-]/[------------------------------------!             !--( )-!
N !#3                                      !TABLE LENGTH!     !
G !                                        !  0256       !     !
  !CR0200                                  !             !     !
# !-] [------------------------------------!TABLE END    !     !
  !#3                                      !HR0256       !     !
  !                                        !             !     !
2 !CR0200                                  !POINTER      !     !
2 !-] [------------------------------------!HR0258       !     !
  !#3                                      !             !     !
  !                                        !DESTINATION  !     !
  !                                        !OR0002       !     !
  !                                        !             !     !
  ! Write data to 798 module               +-------------+     !
  !                                                           !
  !  #22,                                                     !
  !                                                           !
```

This module can also be used with the HPPC programmable controller. The ladder used to interface the NCMZ-798 module to the HPPC is different than what is shown here. For more information, contact Westinghouse.

Appendix B: Error Codes

The NCMZ-798 module communicates with the other drops in the network over its port A. The errors detected on the network are only detected by the master. These error codes are written into Configuration Register 1.

If a network error occurs, bit 9 of the Control configuration register is set to a 1. The lower 8 bits of that same register can then be examined for the reason of the error.

The following table shows what each 8 bit pattern means.

Master PC Errors:
(For errors occurring between the master NCMZ-798 and the master PLC, over the optional program loader link substitute a 4 instead of a 0 in the following error codes)

01	Attempted to write to write protected RAM with the master processor's keyswitch in the RUN/PROGRAM PROTECT position
02	Invalid data transfer detected. This error code (42) will occur if values used in the configuration table are out of range. Example: more than 64 or less than 1 register specified for read or write.
03	Checksum error detected on response from slave
04	Command overrun. The master received a subsequent response before it had processed the previous response
06	UART overrun
08	UART framing error
09	UART parity error
0B	Invalid port transmit status detected
0C	Master timeout. Master waited for CTS to go high before it transmitted data. CTS never went high.

Slave Errors:

81	Attempt to write to the slave PC ladder while in the RUN/PROGRAM protect mode
82	Invalid data transfer command detected. Slave not configured for multi-drop or slave does not support multi-point network.
83	Slave detected a message checksum

84	Slave command overrun. Slave had not responded to previous command before the next command was sent to it.
85	Command aborted. Slave could not respond at that time. One possible cause; a simultaneous attempt to block write both A and B ports of the PC1100.
86	UART overrun
87	Invalid address in slave PC
88	UART framing error
89	UART parity error
8A	Slave timeout. (No response heard from slave)
8F	Response received from slave PC upon initiating test. If the network master is a PC1100 this TEST function is used. The TEST function is not supported by an NCMZ-798 master.
9B	Slave in stop mode
9C	Slave in fault mode

If the error code seems to jump around, verify that the data formats and baud rates between the master and slave are the same.

Appendix C: Communications Ladder for Host PLC (HPPC version)

This program handshakes with the NCMZ-LAN module via IR0064 and OR0064. Notice that this program does not reserve the full 256 registers available (only 90 are used). This is allowed and may be useful when the full 256 register block will never be used, since those registers are available for use in other parts of the program.

Registers Used:

```
   R01568-R01657 Data Table (configuration registers R01568-R01573)
   R01658        Pointer
```

```
|          NETWORK 0311
|+------------++------------++------------+ BP0015 BP0016
||UPDATE I/O  ||  MOVE R-R  ||UPDATE I/O  | R01657 R01657 CR0695 CR0695
++ I/O Start  ++ Source Reg ++ I/O Start  ++--] [---]\[---]\[---( )-
||IR0064      ||IR0064      ||OR0064      ||        +------------+
||   Length   ||  Dest Reg  ||   Length   ||CR0695 |  MOVE R-T   |
|| 00001      ||R01657      || 00001      |+--] [--+ Source Reg +
||            ||            ||            ||        |R01657      |
|+------------++------------++------------+|        | Dest Table |
|                                          +        |R00001      |
|       +------------++------------+       |        |   Length   |
|CR0695 | BYTE MOVE  || ADD  A+B=C |       |        | 00090      |
+--]\[--+ Source Reg ++ Operand A  +-------+        |   Pointer  |
|       |R01657      ||R01658      |                |R01658      |
|       |Move pattern|| Operand B  |                |            |
|       | 00033      || 00001      +                |            |
|       |  Dest Reg  ||  Result C  |                |            |
|       |R01658      ||R01658      |                +------------+
|       |            ||            |
|       |            ||            |
|       +------------++------------+
|          NETWORK 0312
|+------------+          BP0016 BP0015 +------------+
||  MOVE R-R  |CR0695 R01657 R01657 |  MOVE T-R   |
++ Source Reg +--]\[----] [----]\[--+Source Table+
||R01657      |                     |R00001      |
||  Dest Reg  |                     |   Length   |
||OR0064      |                     | 00090      |
||            |                     |  Dest Reg  |
|+------------+                     |OR0064      |
|                                   |   Pointer  |
|                                   |R01658      |
|                                   |            |
|                                   |            |
|                                   +------------+
```

Index

ADDITIONS TO NUMALOGIC SPECIFICATION OF PC1100-LAN EMULATION WITH NCMZ-798

Error codes

Page 124 of the PC1100 instructions manual describes the errors, associated with the master pc. These errors can be found in the first HR of the configuration table. In addition to these codes we now have

- 9B = slave in stop mode
- 9C = slave in fault mode

The module is an interface between the Master pc and the network. For this reason also errors concerning communications between module and master pc may occur. They are coded as 4X (X is confirm the list on page 124)

ADDITIONAL CONGIGURATION REGISTERS

A fifth register was added to be able to specifie a delay and data format. When the RTS line goes high, the module waits for 0.5 seconds for the CTS line to go high before it times out. If CTS high within this period, transmission will begin after the specified delay (HR5 HOB). (0-25.5 sec) The transmission data format is specified in HR5 LOB. The data format is as specified in the following table:

11	8 data bits, 2 stop bits, no parity
15	8 data bits, 1 stop bit , no parity
19	8 data bits, even parity 1 stop bit
1D	8 data bits, odd parity , 1 stop bit
other codes are not permitted	

The default code is 1D.

Because interfacing via the IO-bus sometimes takes considerable time, the normal receive time-out for a master of 0.5 seconds may be insufficient. For this reason, a sixth register was added to the configuration table. HR6 LOB specifies an additional receive time-out of 0-25.5 seconds. If the module is configured as a slave, the receive time-out should be programmed to 25.5 sec. The receive time-out of the master is dependant of the slaves processing time.

RS232 CONTROL LINES

pin 5	RTS	output	Goes high at beginning of transmission
pin 20	CTS	input	Indicates that the modem is ready
pin 4	DCD	input	Resets the internal receive pointer (connected to LAN sync pulse)
pin 6	pulled up to 12V		
pin 8	pulled up to 12V		
pin 2	RXD	input	Incoming data for the module
pin 3	TXD	output	Transmitted data by the module

The third and fourth register of the configuration table specifie the source and destination of data. If the contents is 0000, the first entry of the table, specified with the RT and TR instructions of the ladder program, will be read or written to. If the module is communicating via the program-loader port, 0000 specifies HR1. The source and destination registers specifie the registers address, not the reference number!!

REQUIRED LADDER PROGRAM

The ladder listing gives a ladder program example. Additions to previous ladder programs, as used for example with the basic interpretor, are:

- the specified address in bit 1-8 of the associated IR is complemented in bit 9-14 of the same IR. This allows diagnostics in the PC-ladder program.
- The command in IR is echoed to the module, only if the command was a write. This allows for the module, to check integrity of the interface without affecting the exchange data rate.

HR6 high order byte is now for programming the HIGH state timing of the RTS line. RTS will remain high during the time in milliseconds, specified by HR6 HOB. A minimum time is required, and is dependant of the hardware selected baudrate of channel A.

Therefore, the values of HR6 HOB are:

decimal	hex	
35	$23..	at 300 baud
10	$0A..	at 1200 baud
5	$05..	at 2400 baud
3	$03..	at 4800 baud
2	$02..	at 9600 baud

When applying modems, these values might need tuning.

Regards,
D.Dokter
31 oct 1986

NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

Evaluation proceded using the following hardware and programing blocks:

HARDWARE:

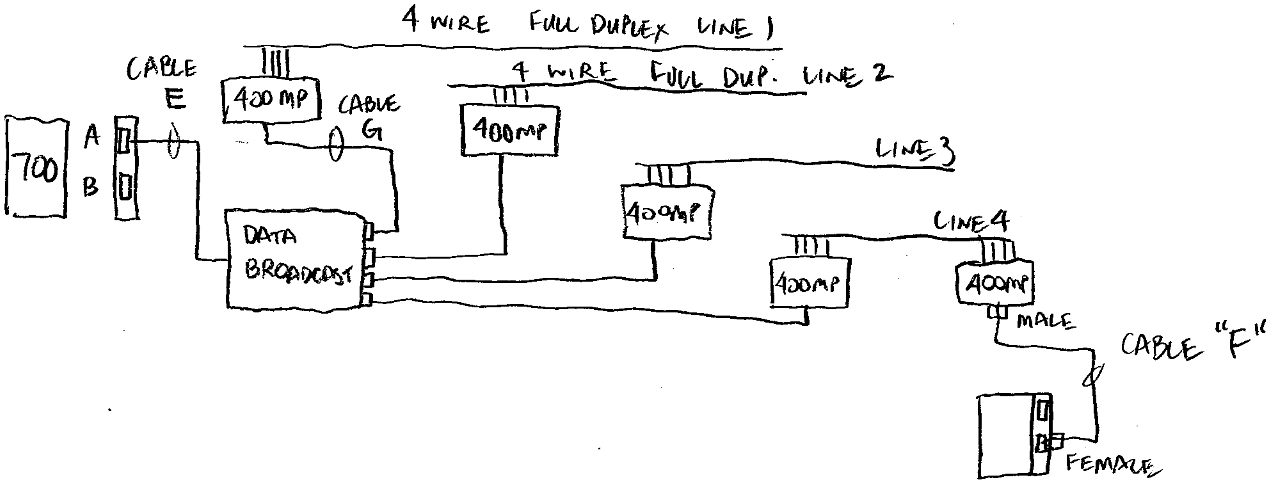
```
cabling:                                    OK FOR BLACK BOX LD 400 MP
        NCMZ-798 port A                     Racal-Vadic 1251/k 4 wire
                                            multi point modem (Bell Tel.
                                            spec. 202 T) DCE equ.
function                                            function
              Male                       Male
 input        Rx   2 <---------------< 3 Rx        output

 output       Tx   3 >---------------> 2 Tx        input

 input        DCD  4 <---------------< 8 DCD       output
                                                          "cable A"
 output       RTS  5 >---------------> 4 RTS       input

 reference    SG   7 ----------------- 7 SG        reference

 input        CTS 20 <---------------< 5 CTS       output
```

LD400MP CABLE "F" PC1100 PORT B

MODEM	
2	2
3	3
4	4
5	5
6	6
7	7
20	20
MALE	FEMALE

NCMZ-798 to NCMZ-798 direct - no modem

```
NCMZ-798 SLAVE                     NCMZ-798 MASTER
      male                              male
         2 >----\ /---------> 2 Rx
                 X
         3 <----/ \---------> 3 Tx

         4 <-----\ /--------> 4 DCD
                  X                              "cable B"
         5 >--o--/ \-----o-< 5 RTS
              |              |
         7 ---)-----------)-- 7 SG
              |              |
        20 <--|              |-->20 CTS
```



NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

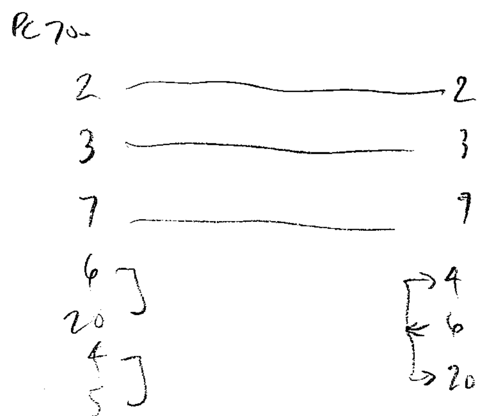
Cabling cont. :

NCMZ-798 to PC1100 direct - no modem

```
NCMZ-798 port A                PC1100 port B
      male                       female
      Rx  2 <---------------< 2  Tx

      Tx  3 >---------------> 3  Rx

      DCD 4 <-------------o-< 4  RTS        "cable C"
                          |
      RTS 5 >-o-\         |-> 5  CTS
              |  \
 (+12VDC) 6   |   \---------> 6  DSR (= DCD)
              |
      SG  7 --)-------------- 7  SG
              |
     CTS 20 <-|             < 20 DTR
```

Cable between NCMZ-798 and PC700/900

```
PC700/900                      NCMZ-798
Port A or B                    Port B
      female                   male
     Tx   2 >---------------> 2  Rx

     Rx   3 <---------------< 3  Tx

     RTS  4               |-< 4  RTS
                          |                 "cable D"
     CTS  5 <-------------o-> 5  CTS

     DSR  6 <---------------< 6  DSR

     SG   7 ----------------- 7  SG

     DTR 20 >---------------> 20 DTR
```

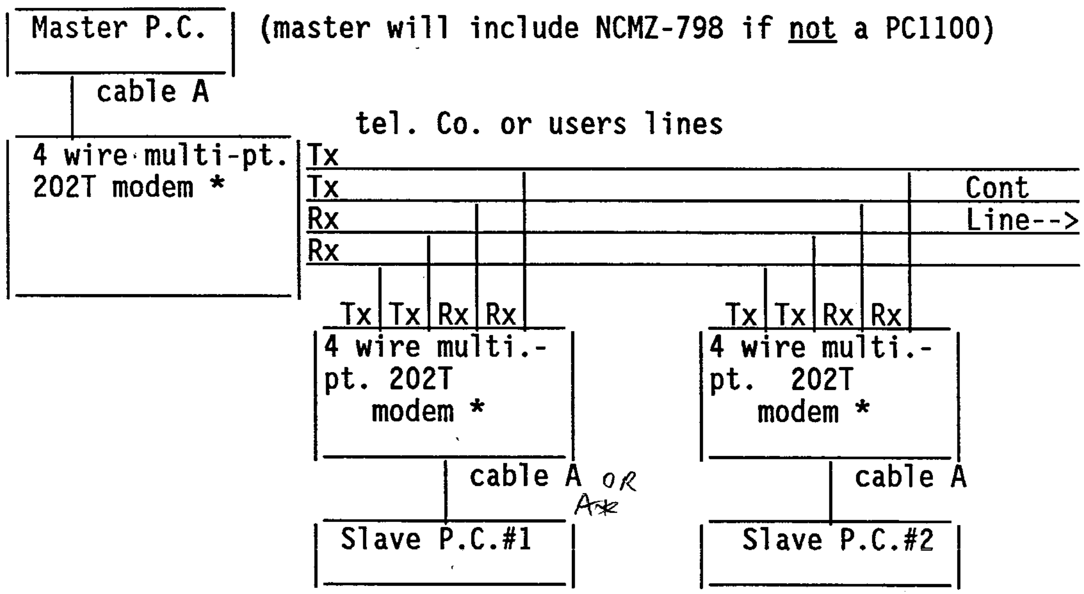


CABLE D

MUST SEE BIT 15 WHEN INITIALIZING (KEYSWITCH OR 798 POWER UP)

NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

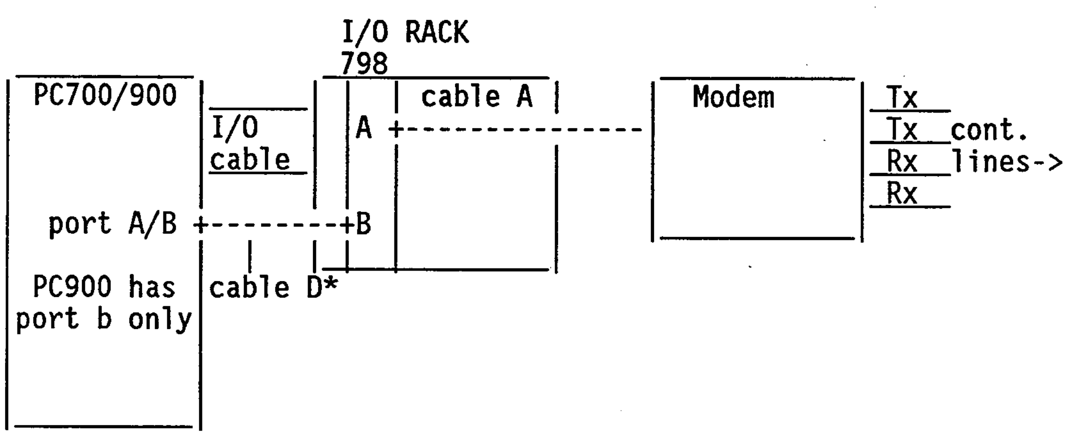
System Layout:



(slave will include NCMZ-798 if not a PC1100)

* A 4 wire multi.- pt. line driver with half duplex operation and Data Carrier Detect indicator can be substituted.

Individual Drop Layout:



* Cable D provides the quickest thru put but is not necessary. Info can transfer over I/O bus, however this will slow down the network and is not recomended unless there is no alternative

NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

NCMZ-798 direct to NCMZ-798 layout:

```
                       I/O RACK             I/O RACK
                       798                  798
 ____________         ___________          ___________          ___________
| PC700/900  | ____  | | cable B|        | |          |   ____ | PC700/900 |
|            | I/O   | |A +-------------- +A          |  I/O   |           |
|            | cable | |  |      |        | |          |  cable |           |
|            |       | |  |      |        | |          |        |           |
| * port A/B +--------+B  |      |        | B +-----------------+port A/B  |
|            |   |   |_|__|______|        |_|_|________| |      |           |
|            | cable D                                 cable D  |           |
|            |                                                  |           |
|____________|                                                  |___________|
```

NCMZ-798 direct to PC1100 layout:

```
                                                           1
                                                           0
                       I/O RACK                            7
                       798                                 5
 ____________         ___________          _____________  ___ ___ ___
| PC700/900  | ____  | | |      |         | PC1100      |A|   |   |   |
|            | I/O   | |A +---------\     | Adv.Ior2    | |   |   |   |
|            | cable | |  |      |   \-------------------+B|   |   |   |
|            |       | |  |      |     |  |             |_|   |   |   |
| * port A/B +--------+B  |      |  cable C|     ||     |4|   |   |   |
|            |   |   |_|__|______|        |     ||     |8|   |   |   |
|            | cable D                    |_____________|5|___|___|___|
|            |
|            |      * PC900 suports only one serial port = A
|____________|
```

NOTE RS232 serial interface suports only one to one communication. The modem or line driver allows expansion to a "party line system" where one master can connect to many slaves.

Module Baud Rate Selection:

Strap selectable for the range of 300 - 9600 on both ports must be set in acordance with modem limitation. The 202T limitation is 1200 baud. Set both ports to 1200 baud.

Programmable Controller Baud Rate Selection:

PC700 Suggest using two ports by adding the NLCI-792 computer interface board to the PC. Leave port A at 9600 for program loader while B is used for module/network communication which must be set up for 1200 baud. The NLPB I/O image -line solver board with in the PC700 has the port strap selections

PC900 Since there is only one RS-232 port it must be time shared between the program loader and module and set up with the baud rate of the network.

Please refer to the PC Communications manual for details.

NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

Racal-Vadic 1251/k switch selectable timing constraints:
(Bell Tel. 202T spec.)

Soft Carrier Timming - forces 900Hz soft carrier signal to be generated at end of data transmition to prevent data errors in remote reciever during carrier detector turn-off period

A2	A3	Timming
ON	OFF	24 ms
ON	OFF	8 ms
ON	OFF	Disabled

pick 24 or 8 ms

Turn Around Delay - Forces carrier detector off at end of data transmition to prevent modem from recieving echo of its own transmited signal.

A4	A5	Timming
ON	OFF	156 ms
ON	OFF	9 ms
OFF	OFF	Disabled

pick 9 ms

Clear to Send Delay - Selects time delay between RTS request and CTS response such that remote reciever has time to detect carrier before data is transmited.

B1	B2	B3	Timing
OFF	OFF	OFF	200 ms
ON	OFF	OFF	60 ms
OFF	ON	OFF	30 ms
OFF	OFF	ON	8.3 ms

Must be > Turn Around Delay or Carrier Detect Timming

pick 30 ms

Carrier Detect Timing - Selects delay period between time modem recieves valid carrier and Data Carrier Detect DCD (=Carrier Detext CxR) turns on. This delay inhibits nuisance toggling of DCD due to communication noise.

B5 ON = 6 ms to ON / 6 ms to OFF
OFF = 23 ms to ON / 6ms to OFF pick B5 ON

These timming constraints are in accordance with Bell Tel. Co. 202T 4-wire multi. - point modem specifications. Other modems or line drivers may work in a system. The device should be able to provide similar timing selections to insure data transfer integrity, particular attention given to the ability to give a clean Data Carrier Detect signal.

NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

PC Program Requirements:

```
PC700/900 slave - values used in eval.
                HR#
                  1   6000 or 2000 hex
                  2   0003 hex
                  3   0000 hex
                  4   0000 hex
                  5   051D hex
                  6   00FF hex
PC7009/900 Master - values used in evaluation
               HR#
                 1  F000 or B000 hex -> changes to DXXX after
                 2  0303 hex            transmition
                 3  000A hex
                 4  000A hex
                 5  051D hex  all data formats where tried
                 6  000B hex  various delay times where tried
```

Please refer to example program for details of how these values are transfered across the I/O bus

PC1100 Master- requires the special functions Port Transmit (PT) and Configure Port (CP)

```
   PT :
                                        ________________
                                       |     LT = 97     |
         |    comm. line               |OP1 Table Length |     PT9
         |--||---------------------    |HR00001(00003)   |-----()
         |    Rd/Wt                    |OP2 Source Table End
         |--||---------------------    |HR00002(00010)   |
         |    Test                     |OP3 Dest. Table End
         |--||---------------------    |HR00003(00010)   |
         |    Enable                   | implied Status Reg
         |--||---------------------    |HR00004(XXXXX)   |
         | Toggling comm. line         |OP4 Address      |
         | starts one comm. session    |HR00005(00003)   |
         | to desired slave.           |_________________|
CP:
         enable
         |--||---------------------    |  LT = 27        |   CP10
         | CP is read on power up,     |OP1 CV=3 (port B in|-----()-
         | keyswitch or enable         |multi.-pt. modem |
         | transistions                |mode)            |
         |                             |implied operand 3|
         |                             |(HR for operand 2|
         |                             |minus one)"S"delay
         |                             |CTS to Tx delay  |
         |                             |HR00009(00000)   |
         |                             |OP2 Configuration|
         |                             |HR00010(6D02 hex)|
```

NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

P.C. Program Requirements:

PC1100 Slave - requires the special functions Unit Address (UA) and Configure Port (CP)

```
UA:      enable
       |--||--------------------|      LT = 28       |     UA 1
       | UA is read on power up | OP1  Unit Address  |------()--
       | or key switch transis- |    CV = 1          |
       | tions. Toggling the    |                    |
       | enable resets the      | OP2 Dummy operand  |
       | communication blocking |    CV = 1          |
       | bit.                   |                    |
```

Slave's Unit Address <u>can not be changed</u> durring run

CP: as in the master PC1100

Please refer to the "PC1100 Advanced Functions" manual and the 1075 port adapter card instructural leaflet for details on programing and setting up a PC1100 in a Local Area Network.

The PC1100 in a multi.- pt. modem application must have a Configure Port special function in the program where the first operand is a constant value of 3 (CV = 3). Also the switches on the communication exspantion board within the PC must be closed for all units.
NOTE Some program loaders require entering the Configure Port special function as a literal function LT = 27 rather than the nemonic representation of CP, such that operand one will except an entery as a constant value equal to three.

<u>NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86</u>

Using the previously defined building blocks the present NCZM-798 software was tested in all practicle combinations with the following problems (issues) surfacing.

<u>NCMZ-798 as a Slave:</u>

The Status Register HR#1 must be read inorder to determine if RS-232 style communication over one of the P.C. communication ports is allowed and to indicate the need to reload the slaves address by referencing bit 14 "Start Function". It appears that the NCMZ-798 module only reads HR#1 thru HR6 upon power up of the module or keyswitch transition of the P.C., and leaves bit 14 set. In contrast the master gives an indication that the data has been read by clearing bit 14.

If the addition of clearing bit 14 for a slave is cumbersome we can do without it. I want to raise the issue so we can make note as to why
With a module as a slave and a PC1100 as master we found that the module would respond with the proper six byte response <u>except that the check sum was in error</u>. The master interperting this error would not proced beyond this point.
The error was found using two tools. The master PC1100's status register lower byte indicated 03 hex. Also we monitored the communication link using a Hewlet Packard 1640 A Serial Data Analyzer which can display transmitted and recieved information in hex format and clearly showed the check sum was in error.
e.g.

set PC address from master	0A 03 00 00 00 0D
Set PC address response from slave	0A 03 04 00 00 F1

The lower nibble of the checksum appears to be calculated properly while the upper nibble has its first three high order bits set high. Different addresses yield the same results.
Different data frames did not effect the problem nor did timing constraints. or the ability to communicate RS-232 to the PC700

The HP analyzer has a simulate mode where one can buffer up a desired string send it to a device and catch the response. Using the modems as an interface between the HP Analyzer and a slave, the HP can simulate the actions of a master. With a PC1100 as a slave one complete transaction could be accomplished by first sending out the set PC address catching the response and then sending out a block read comand and catching the response Trying this with a module as slave the only exchange possible was a response to set PC address. Ignoring the check sum error, a block read from the HP analyzer of appropriate registers yeilded no response. If there is a time out for slave to respond to block read comand after recieving set PC address which is faster than my ability to change the buffered command in the HP from set PC address to block read, then this lack of response is understandable.

NCMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

NCMZ-798 as a Master:

Check sum error causes failure of network.

When connected to a PC1100 over modems or cable C, the PC1100 would send back check sum error codes which the module inturn indicated with 83 hex in the status byte

Inorder to check wether the module as a master would continue with a block read comand after set PC address comand, the HP analyzer was set up as the slave. The HP unit can be set up to ignore the check sum error and transmit the set PC address acknowledgment after recieving the set PC address comand from the module. The module did follow by sending out a block read comand ,however the check sum was in error.

Set PC address from master module	0A 02 00 00 00 FC
HP analyzer response	0A 02 04 00 00 10
Block read from master module	05 0A 00 02 00 F1

Module as both slave and master:

If the module viewed the checksum the same way it was formated we could have communication even though the check sum was formed differently with respect to the PC1100. This was not the case. The modules respond to the format of the PC1100 check sum, but not to the their own corrupted check sum.

With either modems or cable B, the two modules would not communicate in a slave - master relationship.

CMZ-798 PC1100 LOCAL AREA NETWORK EMULATION SOFTWARE EVALUATION 7/86

Flow Chart of PC1100 Port Transmit Operation: