H E W L E T T

P A C K A R D

HP-CIO SIX CHANNEL TERMINAL MULTIPLEXER

FIRMWARE FOR HP-UX/RT TERMINAL I/O SUPPORT

EXTERNAL REFERENCE SPECIFICATION (ERS)

**** F I N A L ****

History

 date description

840117 Original document

840209 Major revision.

840223 Changes to the ARQ Asynchronous Event Sensed (AES) section,
       to the modem status bit assignment, and to the write
       data/configuration/control request with subfunction 1 and
       146.

840301 Minor clarification in the read download status with option
       1 section and added note in download order with option 1,
       subfunction 1 section.

840615 Added notes concerning the 2 characters FIFO in the SCC and
       the maximum record size that the host can receive on a read;
       added subfunction 8 to program the host controlled
       X-ON/X-OFF characters, subfunction 149 to send the host X-ON
       character immediately, and subfunction 150 to send the host
       X-OFF character immediately.

841219 Added section 3.8.2.11 concerning the potential modem
       problems and solutions; added additonal modem error codes
       for the modem status byte; added notes concerning the
       maximum length of the write data/configuration/control
       transaction block; added subfunctions 151, 152, and 153;
       added notes for subfunction 143; and added default values
       for the new subfunctions.

850104 Added notes to IC collision timer specifying when the timer
       is cleared; added additional notes to Potential Modem
       Problems & Solutions section; added additional notes to
       modem error code 9; added additional information to
       subfunction 1, 142, and 153; changed default values for the
       downloaded firmware; and added Appendix C describing EPROM
       log channel backplane protocol firmware.

850318 Changed status code 141 to return the state of modem signals
       TR and RS.

850401 Added note specifying that the firmware stops processing the
       write data/configuration/control transaction when an error
       is encountered.

850417 Changed and added modem error code for the modem quad.

850909 Changed the meaning of modem status code 7 and 9.

851009 Added subfunctions 154 and 155, updated meaning of modem
       status code 9.

860414 updated subfunction 1 concerning the switch from hardwire
       mode to modem mode when all modem signal are not true and
       update subfunction 144 concerning the maximum data byte
       count that may be used.

```
+---------------------------------------------+------------------+
|                                             |                  |
|    INTRODUCTION                             |   CHAPTER  1     |
|                                             |                  |
+---------------------------------------------+------------------+
```

1.1   Scope

This document describes the firmware that is available for the
HP-CIO RS-232-C 6 channel modem multiplexer card to support the
HP-UX/RT terminal I/O. Specifically, this document describes:

        o the firmware features available on the card
          with the EPROM shipped with the card,

        o the firmware features available after the
          modem Mux firmware is downloaded,

        o the host interface to the card, and

        o the card interface to RS-232-C and V.28
          devices.

The reader is referred to the following related documents.

        o Zilog Z80 CPU Technical Manual.

        o Zilog Z8030/Z8530 SCC Serial Communications
          Controller Technical Manual.

        o Zilog Z8036 Z-CIO/Z8536 CIO Counter/Timer and Parallel
          I/O Unit Technical Manual.

        o Backplane Interface Circuit (BIC) ERS by Bill Martin,
          Roseville Networks Division (RND).

        o HP-CIO Standard I/O Backplane by R. B. Haagens, Computer
          Systems Division (CSY).

        o Standard Datacom Driver by Greg Dolkas, Roseville
          Networks Division (RND).

        o CSG I/O Standard Backplane Protocol for Smart
          Cards by Greg Dolkas, Roseville Networks Division (RND).

o Guidebook to Data Communications, Part No. 5955-1715,
  Hewlett-Packard, 1977.

o HP-CIO SIX CHANNEL MODEM MULTIPLEXER, FIRMWARE
  EXTERNAL REFERENCE SPECIFICATION (ERS), by Terry
  Gong, Roseville Networks Division (RND).

1.2   Definition of Terms

ASI - Asynchronous Serial Interface. The single channel (vs 6 or
      8) interface.

Asynchronous - Asynchronous communication is a method of
               transmitting data without sending the clock signal.
               Each character consists of one start bit, 5 to 8
               data bits, an optional parity bit if any, and one or
               two stop bits.

Backplane - The interface between the host and the card.

Baud - "Baud" refers to the speed of signal units. Where one bit
       is used as a signal unit, baud speed and bps are the same.
       When two bits are combined to form a signal unit, then the
       baud rate would be half the bps. The signal unit is called
       a "dibit" (double bit), and has four levels or states. When
       three bits form one signal unit, it is called a "tribit"
       (triple bit) and has eight possible states.

BIC - Backplane Interface Circuit. A gate array chip used to
      interface the host backplane to the microprocessor on the
      interface card using the HP-CIO standard.

bps - Bits Per Second, the speed of data transmission over a
      communication channel.

Break - As used in asychronous serial data communication, a break
        condition is defined to be a signal "space" condition for
        at least 2 character time. (The 200 milliseconds
        definition is for device with a baud rate of 110.)

CCITT V.10 - Electrical standard for data communications similar to
             RS-423.

CCITT V.11 - Electrical standard for data communications similar to
             RS-422.

CCITT V.28 - An International Telegraph & Telephone Consultative
             Committee (CCITT) electrical standard for data
             communication which is similar to RS-232-C.

Channel - As used in data communication, a channel is a path within
          a line through which information flows.

Command - A command from the host computer to the card is defined
          to be a request to the card which is not prompted by a
          SRQ from the card.

CRC - Cyclic Redundancy Check is a method of using a polynomial to
      perform error checking. The polynomial is an algebraic
      function used to create a constant from the message bit
      pattern. This constant, generated and accumulated in both
      the trasmitter and receiver, is used to divide the binary
      numeric value of the character. The quotient is discarded
      and the remainder added to the next character, which is again
      divided. This continues until the last character, when the
      remainder is transmitted to the receiver for comparison with
      the receiver's remainder. An equal comparison indicates no
      errors, while an unequal comparison indicates an error in the
      transmission.

Cyclic redundancy check - See CRC.

Data communication equipment - See DCE.

Data set - Another name for modem.

Data terminating equipment - See DTE.

DCE - Data Communication Equipment is the equipment used to convey
      information between locations (e.g., a modem).

Download - the process of transferring Z80 executable code from the
           host to the card.

DTE - Data Terminal Equipment is the remote terminal where
      information enters and exits from the data link for a user or
      the computer where information is processed and stored.

Echoplex - This is a mode in the full duplex communication channel
           where any character transmitted by the terminal is
           echoed back to the terminal.

EIA RS-232-C - A North America data communication standard which
               specifies the mechanical and electrical
               requirements. It uses a 25-pin connector as the
               standard interface in Datacomm networks, with
               lettered pin assignments for ground, data, control

and timing circuits. The operating range is from 0 to 20,000 bps in bit-serial operation, synchronous and asynchronous.

EIA RS-423 - A data communication standard similar to EIA RS-232-C.

Frontplane - The interface between the card and the RS-232-C or CCITT V.28 device.

Full duplex - A channel where information can travel in both directions simultaneously between two locations. No turnaround time is required.

Half duplex - A channel which can transmit and receive but not simultaneously. The transmission flow must be halted each time the direction of travel is reversed. This halt is called "turn around time" and typically requires from 50 to 250 milliseconds (ms), depending upon line mileage. The halt is required to reverse the direction of the echo suppressors in the telephone line and to allow the modems to stabilize.

Hardwired direct connection - A direct connection between a computer and a terminal or between 2 computers over a relatively short distance using copper wire pairs or coaxial cable. The signals are transmitted in digital format.

Host - The computer in which the HP-CIO 6-Channel Modem Mux card is installed.

HP-CIO - A standard I/O backplane for the I/O interface cards for the Computer Systems Group. (HP Channel I/O)

Level-0 - For HP-CIO, the mechanical and electrical specification for the interface between the host backplane and the interface card.

Level-1 - For HP-CIO, the bus primitives between the channel and the device adaptor.

Level-2 - For HP-CIO, the command and order definition for the interaction between the host I/O backplane and the interface card.

Level-3 - For HP-CIO, the description of a series of related interactions between the channel and the device adaptor using the Level-2 protocol. This is often referred to as a transaction.

Line - As used in Datacomm, this refers to the physical equipment and configuration used in telecommunications.

Modem - Modulator/Demodulator. A device to convert the digital signal to an analog signal for transmission over the telephone line. At the receiving end the analog signal is converted back into the digital format. Modems are also often referred to as data sets by the phone companies.

Mux - An interface which multiplexes several independent data streams over a single backplane slot.

Multiplexer - see Mux

Order - A request from the host channel program to the card which requires the SRQ signal from the card before being sent.

Parity checking - Parity means that the bit count of character 1's should be either all odd or all even. A bit is inserted into the character to obtain parity where necessary. The insertions are called redundant because they can be removed from the message without loss of information. An all odd count per character is called odd parity. An all even count is called even parity. The output device supplies parity and the input device checks for parity.

PCA - Printed Circuit Assembly.

Polynomical checking - See CRC.

Redundancy checking - See parity checking.

Simplex - The message flow is always in one direction. An input terminal can only receive, never transmit; an output terminal can only transmit, never receive.

Synchronous - A transmission method where the characters are transmitted in a continuous stream with no intervals between each character. There are no start or stop bits like in asynchronous mode. Both the transmitter and receiver must maintain precise timing. It requires a clock be sent along with the data on a separate line.

Terminal - The device connected to the HP-CIO 6-Channel Modem Mux card.

Transaction - See Level-3.

```
+--------------------------------------------+------------------+
|                                            |                  |
|   OVERVIEW                                 |   CHAPTER  2     |
|                                            |                  |
+--------------------------------------------+------------------+
```

The 6 channel modem multiplexer card is used for interfacing up to
6 EIA RS-232-C devices to the HP-CIO backplane. The card firmware
is optimized for character-at-a-time I/O.

The Mux card uses a Z-80B microprocessor, along with three Z8530
SCCs (serial communication controller) and one Z3536 CIO
(counter/timer and parallel I/O unit) to achieve a low cost
solution while maintaining versatile functionality. It allows
direct interfacing to a wide variety of devices. It contains
firmware implemented capabilities that simplify interfacing tasks
for the host CPU.

2.1   Hardware

   o Z80B microprocessor with system clock of 3.6864 MHz

   o Z8530 SCC (x3)

   o Z8536 CIO

   o Backplane Interface Circuit (BIC) gate array chip
     and its support circuitry

   o Memory Interface Circuit (MIC) gate array chip for
     memory control, DMA, and peripheral chip support

   o Up to 16K x 8 EPROM for the firmware

   o 64K x 8 Dynamic RAM

   o RS-232-C transmitters and receivers (with compatibility
     for CCITT V.28) for 6 ports.

   o Standard HP-CIO connectors

   o Card and loop-back hood LEDs to indicate a go/no go
     self test

2.2   Features

   o EIA RS-232-C and CCITT V.28 compatibility

   o # of data bits/character: 5, 6, 7, or 8 plus parity

   o # of stop bits: 1, 1.5, 2

   o transmission mode: full duplex hardwire or modem

   o parity: none, odd, even

   o clock: internal

   o baud rate: 50, 75, 110, 134.5, 150, 300, 600, 900,
              1200, 1800, 2400, 3600, 4800, 7200,
              9600, or 19200

   o baud rate defaults to 9600, and is software programmable
     to any of 16 rates

   o break detection and generation

   o programmable device controlled X-ON/X-OFF handshaking

   o self-test

   o full duplex modem support to satisfy European
     licensing

   o timer for various modem functions

   o optional asynchronous event sensed interrupt

+---+--------------------+
CAPABILITIES DESCRIPTION	CHAPTER 3
+---+--------------------+

This chapter will describe each feature that is available on the card in detail for the downloaded firmware.

3.1   Device Controlled X-ON/X-OFF Handshake

This handshake protocol allows the device to pace the data transfer from the card to the device. The device will signal the card to stop transmitting data by sending an X-OFF character. The receiving device restarts data transmission by sending the X-ON character.

The mux firmware will stop data transmission as soon as the X-OFF character is received, however, up to two characters may be transmitted before the stopage due to the SCC FIFO buffer.

If the handshake is disabled while output is in process, a "restart transmitter" control request may be used to prevent outbound data from getting stuck on the card.

All handshake characters will be returned to the host in the order received. A 0FFH for the handshake character will disable the corresponding defined handshake. The handshake character for X-ON and X-OFF cannot be the same; that is, the firmware cannot use the same character to toggle the handshake.

3.2   Transmit Hardware Handshaking

The user must manually control all hardware handshaking like RS/CS. The handshake control is performed by using the control request defined in chapter 5.

The user can determine the state of the modem signal by polling or by waiting for a modem status change.

3.3   Stopping the Transmitter

The transmitter may be stopped from transmitting characters by enabling and using the X-OFF handshake or by using the control request. The control request will stop the transmitter like as if an X-OFF was received. Thus the transmitter may be restarted by using another control request or by using the X-ON handshake if it is enabled.

3.4   Force Restart of the Transmitter
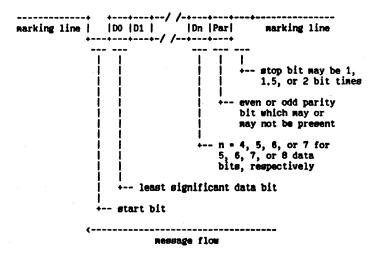
A control request is available for the host to restart the transmitter if it is stopped because of waiting for a device X-ON character from the device or terminal. This option is usual used when the handshaking is enabled and then disable by the host software without knowing the state of the card firmware or is used if the host tells the card to stop transmitting and wishes to restart the transmitter.

3.5   Programming the Receiver & Transmitter

The character size for the receiver and the transmitter may be specified at 5, 6, 7, or 8 bits per character not including an optional bit for even or odd parity. On transmit, the user data will be processed byte by byte passing the 5, 6, 7, or 8 least significant bits in each byte to the transmitter depending on the programmed character size. A parity bit will be added by the interface if even or odd parity is enabled. On receive, the incoming data will be passed to the user's buffer into the 5, 6, 7, or 8 least significant bits of each byte with the unused bits being zeros. The parity bit is never returned to the user.

```
--------------+   +---+---+--// /-+---+---+---+---------------
marking line  |   |D0 |D1 |     |Dn |Par|     marking line
              +---+---+---+--// /--+---+---+
              --- ---                --- --- ---
               |   |                  |   |   |
               |   |                  |   |   +-- stop bit may be 1,
               |   |                  |   |       1.5, or 2 bit times
               |   |                  |   |
               |   |                  |   +-- even or odd parity
               |   |                  |       bit which may or
               |   |                  |       may not be present
               |   |                  |
               |   |                  +-- n = 4, 5, 6, or 7 for
               |   |                      5, 6, 7, or 8 data
               |   |                      bits, respectively
               |   |
               |   +-- least significant data bit
               |
               +-- start bit


              <------------------------------------
                        message flow


                 Asynchronous Message Format
```

3.6   Break Detection & Generation

The firmware will notify the host of receiving a break on the next
read transaction done by the host after the detection.

The host can request the card to generate a break condition for the
device as soon as the control request is received. The break
generation length is determined by the host. The host does this by
using the control request to start the break generation and another
control request to stop the break generation.

3.7   Modem Timers

1. Modem connection timer. This is the time allowed to establish

---

MT CONNECT

modem connection by waiting for the DM, CS, and RR signals for
full duplex modem. If the DM, CS, and RR signals are not
asserted within the time given, the line connection request for
the originating mode will be aborted. For the auto answer
mode, the firmware will go back into the waiting for the IC
signal mode. The default value will be 25 seconds. The clock
resolution is 1 second. The clock range is from 0 to 255
seconds.

2. No activity disconnect timer. This timer is used to prevent
the incurred of excessive connect charges due to no activity
over a modem connection. The default value is 600 seconds.
The clock resolution is 1 second. The clock range is from 0 to
65535 seconds.

3. Lost receiver ready timer. For full duplex modem, this timer
is activiated by the firmware whenever the RR modem control
signal is lost. If the control signal should return before the
timer times out, the timer will be reset to 0. If the timer
times out before the control signal returns, the firmware will
lower TR and RS and inform the host of the disconnection.

   The mux firmware also uses this timer during the initial modem
connection phase. When the DM signal is first asserted, this
timer will be started. After the timer has timed out, the
firmware will check the DM signal. If the signal is still
asserted, the firmware will continue the modem connection
sequence by checking for the CS and RR modem signals.
Otherwise, the firmware will continue the modem connection
sequence by waiting for the DM signal for the originate mode or
by waiting for the IC signal for the auto answer mode.

   The resolution of the timer is 10 milliseconds. The default is
400 milliseconds. The clock range is from 0 to 2550
milliseconds.

For all of the timers above a zero value will defeat the timer;
that is, there will be no time out.

4. IC collision timer. When the IC modem signal is received and
if the auto answer mode is not active, the IC collision timer
is started. If the host should post an originate open modem
link request while the IC collision timer is active, a modem
change quad is generated with an modem error code 8 to abort
the open request.

   If there are several rings for the incoming call, the IC
collision timer is restarted for each ring. The timer is set
for 10 seconds.

The timer is cleared on expiration of the timer or by the host posting an auto answer open request.

3.8 Modem Support

The following modem lines will be supported by this firmware.

signal direction		modem line description	EIA RS-449 symbol	
card	device			
-->		send data	SD	T_x
<--		receive data	RD	R_x
-->		request to send	RS	RTS
<--		clear to send	CS	CTS
<--		receiver ready	RR	DCD
<--		incoming call	IC	RI
-->		terminal ready	TR	DTR
<--		data mode	DM	DSR
-->		signal rate selector	SR	DRS
<->		signal ground	SG	SG

3.8.1 Host Control of the Modem Signal

~VERSUS HARDWIRED

When the card is not set to full duplex modem transmission mode, the firmware will not provide any modem controls. However, the user may control all the modem lines by using the control request. Any timers for modem control will not be active. No error checking will be provided like checking for lost DM.

Any time the modem change state, the firmware will return a data block indicating the modem status change for that port. The user may use a control request to get the current modem state.

3.8.2 Firmware Control of the Modem Signal

When the card is enabled for full duplex modem transmission mode, the firmware will provide full modem control as described below.

There are three modem control timers implemented by the card firmware: the modem connection timer, the no activity disconnect timer, and the lost receiver ready timer. The modem connection timer is used to time the initial connection. It may be disabled or set to a value up to 255 seconds (default is 25 seconds). The no activity disconnect timer disconnects the modem if there is no transmit or receive data activity in the specified time interval. It may be disabled or set to a value up to 65535 seconds (default is 600 seconds). The lost receiver ready timer times the loss of the receiver ready (RR) signal for full duplex modem. Its default value is 400 milliseconds; however, it may be disabled or set to a new value up to 2.5 seconds.

The lost receiver ready timer is also used in the initial modem connection phase. Upon receiving the DM signal, the timer will be started. After the timer times out, the firmware will check the DM signal. If the DM signal is still asserted, the firmware will then continue the modem connection sequence by checking for the assertion of the clear-to-send (CS) and receiver ready (RR) modem signals. Otherwise, the firmware will continue the modem connection sequence by waiting for the DM signal for the originate mode or by restarting the IC poll for the answer mode.

The terminal ready (TR) and request-to-send (RS) signals will be set low whenever the data mode (DM), the receiver ready (RR), or the clear-to-send (CS) signals are lost after the link has been opened.

There are two ways of connecting a modem. The first is answer mode where the card will wait for an incoming call, and the second is origination mode where the card will initiate the call or connection.

3.8.2.1 Full Duplex Modem

The full duplex modem connection uses the send data (SD) line to send data to the modem and the receive data (RD) line to accept data from the modem. The request to send (RS), clear to send (CS), data mode (DM), receiver ready (RR), and incoming call (IC) lines are used to handshake with the modem. The signal rate selector (SR) line can be controlled by the host.

In answer mode, the firmware will wait for the IC line to be asserted. The firmware then respond by asserting the RS and TR

signal and starting the modem connection timer. These lines will be deasserted if DM, CS, and RR is not turned on when the timer times out. In this case the firmware will return to the mode of waiting for the IC signal. When the modem connection is made, the firmware will notify the host by returning a data block with the modem status change bit set with an open link for the modem error code.

To originate a call, TR and RS will be asserted and the modem connection timer will be started. These lines will be deasserted if DM, CS, and RR do not turn on within the time out of the timer. In this case the card will return a status indicating that the open request failed.

Once a connection has been established, the firmware will enable the no activity disconnect timers.

3.8.2.2 Full Duplex Modem Timing Diagram

The following are the various timing diagrams for full duplex modem functions.
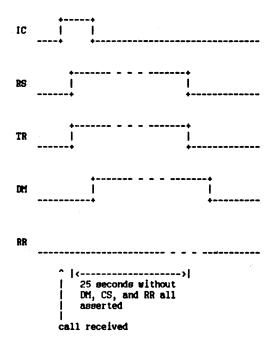
NOTE THAT THE TIME VALUES GIVEN IN THE TIMING DIAGRAMS ARE THE DEFAULT VALUES USED IN THE FIRMWARE TO OBTAIN THE NECESSARY DATA CUM LICENSES.

3.8.2.3   Normal Connection - Auto Answer

```
            +------+
IC     |      |
       ----+      +-------------------------------

               +-------------------------------
RS     |
       -------+

               +-------------------------------
TR     |
       -------+

                      +------------------------
DM     --------------------+
                           |

                      +--------------------
RR     ----------------------+
                             |

                      +--------------------
CS     ----------------------+
                             |

       ---------------------+-------------------
RD                          |/////////////////
                           +-------------------

       -------------------+---+-------------------
SD                          |/////////////////
                           +-------------------
            ^                    ^
            |                    |
          call               connection
          received           established
```

3.8.2.4 Wrong Number Call - Auto Answer

```
         +------+
IC       |      |
      ----+      +-----------------------------

           +--------- - - - -------+
RS         |                       |
      -----+                       +--------------

           +--------- - - - -------+
TR         |                       |
      -----+                       +--------------

             +--------- - - - -------+
DM           |                       |
      -------+                       +----------

RR
      ------------------------- - - - ---------------

           ^  |<--------------------->|
           |   25 seconds without
           |   DM, CS, and RR all
           |   asserted
           |
          call received
```
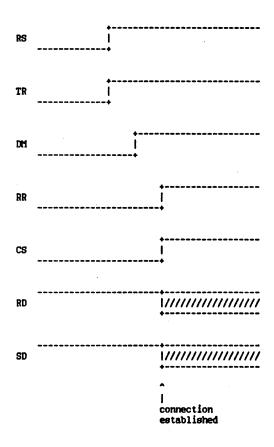
3.8.2.5 Normal Connection - Originate

```
                       +----------------------------
RS                     |
      -----------------+

                       +----------------------------
TR                     |
      -----------------+

                         +--------------------------
DM                       |
      -------------------+

                            +-----------------------
RR                          |
      ----------------------+

                            +-----------------------
CS                          |
      ----------------------+

      ----------------------+//////////////////////
RD                           /
                            +---------------------

      ----------------------+//////////////////////
SD                           /
                            +---------------------

                            ^
                            |
                          connection
                          established
```

3.8.2.6 Lost Clear-to-send Disconnect

```
        --------------+
RS                    |
                      +------------------------------

        --------------+
TR                    |
                      +------------------------------

        ------------------+
DM                        |
                          +----------------------

        ------------------+
RR                        |
                          +----------------------

        ------------+
CS                  |
                    +------------------------

        ------------+---------------------------
RD  //////////////\
        --------------+

        --------------+---------------------------
SD  //////////////\
        --------------+
                    ^
                    |
                    +-- loss of CS
```

3.8.2.7 Lost Receiver Ready Disconnect

```
        --------------+
RS                    |
                      +------------------------------

        --------------+
TR                    |
                      +------------------------------

        ------------------+
DM                        |
                          +----------------------

        -------+     +----+
RR          |     |    |
            +-----+    +----------------------

        ------------------+
CS                        |
                          +----------------------

        --------+---------------------------------
RD  ///////\
        --------+

        --------------+---------------------------
SD  ///////////////\
        --------------+

                 |<-+->|
                    |
                    +-- 400 ns loss of RR
```

3.8.2.8   Lost Data Mode Disconnect

```
         ------------+
RS                   |
                     +------------------------------

         ------------+
TR                   |
                     +------------------------------

         ----------+
DM                  |
                    +------------------------------

         ----------+
RR                  |
                    +------------------------------

         ----------+
CS                  |
                    +------------------------------

         ----------+-----------------------------
RD  /////////\
         ----------+

         ------------+-----------------------------
SD  ///////////\
         ------------+
                    ^
                    |
                    +-- lost DM
```

3.8.2.9   No activity Disconnect

```
         ------------------------+
RS                               |
                                 +--------------------

         ------------------------+
TR                               |
                                 +--------------------

         --------------------------+
DM                                 |
                                   +----------------

         --------------------------+
RR                                 |
                                   +----------------

         --------------------------+
CS                                 |
                                   +----------------

         ----------+----------------------------------
RD  /////////\
         ----------+

         ----------+----------------------------------
SD  ///////////\
         ----------+
                    |<---------->|
                    no transmit or received
                    data for 10 minutes
```

3.8.2.10 Firmware Actions on Modem Disconnect

After the modem is connected, the modem may be disconnected by using the disconnect modem control request or by the firmware detecting the lost of the DM, RR, or CS modem signals or by the no activity timer timing out. When the disconnect occurred, the firmware will flush all the data in the transmit buffer. In addition the firmware will not notify the host that a disconnect has occurred until both the DM and CS modem signals have gone low.

3.8.2.11 Potential Modem Problems & Solutions

When the user is controlling the modem line, the firmware normal operating mode is to generate a quad whenever the modem line changes state. This may be a problem if one of the modem line is noisy. Under this condition many quads may be generated by the firmware which may lead to silo buffer overflow. In addition if the noise persist for a long period, the modem line may prevent the normal processing of the receive data due to the large number of interrupts. This will lead to data losses on one or more ports.

There several solutions to the above problem. The first is to use subfunction 152 to enable the firmware to generate one quad for the specified port only when the modem line change state and only when the host has posted a read data/status transaction. Note that the quad will only be generated if the current state at the time of quad generation has changed from the state that was sent to the host in the last quad generated. This mean that if the modem line has changed state and then changed back to the original state, the host will not see this change. If it is necessary for the host to see every change, the modem quad generation control should not be enabled. Subfunction 151 may be used to disable the modem quad generation control. Subfunctions 151 and 152 are effective on a per port basis.

NOTE THAT WHEN THE MODEM QUAD GENERATION CONTROL IS ENABLED, THE QUAD WILL BE GENERATED AT THE TIME THE HOST POST A READ DATA/STATUS TRANSACTION AND THE QUAD WILL BE RETURNED IN THAT TRANSACTION. ADDITIONALLY, IF THE CARD TRANSMISSION MODE IS FULL DUPLEX MODEM, THE MODEM QUAD GENERATION CONTROL IS IGNORED.

The above method does not prevent the loss of data if one of the port really has a very noisy modem line which is constantly generating interrupts. To solve this problem subfunction 153 may be used to individually on a per port basis to enable or disable the frontplane interrupt for each of the modem signal. When the mask for the modem signal is set, the frontplane will generate an interrupt for every change in the signal. If the corresponding bit is cleared in the mask, the firmware will turn off the interrupt

[handwritten notes in margin: HARDWIRED MODE / MODEM LINE INTERRUPTS / SUBFUNCTIONS 151 & 152 USED ON PER PORT BASIS / SUBFUNCTION 153 DISABLE INTERRUPT ON PER PORT BASIS]

for the modem signal. This will also stop the firmware from reporting any changes in the modem signal. The user must use subfunction 141 to poll the modem lines to get the current state.

NOTE IF THE CARD TRANSMISSION MODE IS CHANGED TO FULL DUPLEX MODEM, THE CARD FIRMWARE WILL AUTOMATICALLY ENABLE ALL THE FRONTPLANE INTERRUPT FOR THE MODEM SIGNALS. If the user decide to switch back to the full duplex hardwired transmission mode, all the frontplane interrupt will remain enable. It is up to the user to use subfunction 153 to disable the desired modem signal interrupt.

When the card is controlling the modem lines only one quad will ever be generated. There will be one quad for modem connection and one quad for modem disconnection. However, no quad will be generated until an open modem link request is received from the host.

Another potential problem when the card is controlling the modem is the collision of an incoming call with the host initiated originate request. If the host has posted an originate open modem link request and if the firmware received an incoming call (indicated by the IC signal going high) or if there is an active incoming call, the firmware will abort the originate request and will not answer the incoming call. An active incoming call is indicated by the IC modem signal being high within the last 10 seconds. *[handwritten: - RI]*

If the host should switch to the auto answer mode and perform an open link request after an originate request abort, the card firmware will start the modem connection sequence the next time that the IC signal goes high. The firmware will not automatically start the modem connection if the IC collision timer is active. This procedure is necessary to prevent the card from answering a call when the call is no longer exist.

3.9 Diagnostics

Self-test is a portion of the mux firmware which performs diagnostic tests to detect card malfunctions. Self-test is executed 'offline'. That is, it does not run concurrent with the standard mux firmware. It is executed upon a hardware reset of the card following a HP-CIO "Addressed Device Clear" (DCL/DEN) or "RESet" (RES) assertion. It exercises the major components and data paths on the card. If no problems are found, the card is made functional and the standard mux firmware is invoked. If a hardware malfunction is detected, the card is left disabled indicating that self-test failed.

The following tests are performed by self-test:

o ROM test: To insure that no bits have changed on the ROM (EPROM),
  a cyclic redundancy check is done using the polynomial
  $X^{**}16+X^{**}2+X+1$. The test is performed in 4K segments to insure
  accuracy of the CRC.

o RAM test: RAM is checked for both stuck-at-0 and stuck-at-1
  conditions and address decoder failures.

o ASYNC SCC Loopback test: Verifies that each SCC can perform basic
  asynchronous internal transmit and receive functions. If a
  loopback hood is sensed, loopback is performed using the external
  line drivers. This verifies the external transmit and receive
  functions and the modem signals (RS, RR, CS, TR, and IC).

o CIO Loopback test: Detects stuck-at faults in the data lines,
  system control, interrupt control, and the three channel signals.
  If the loopback hood is sensed, the modem signal lines (DM and
  SR) are also check through its external line drivers.

o BIC test: Checks the Backplane Interface Circuit (BIC) for
  functional faults. Checks for some stuck-at faults in internal
  BIC registers. The BIC chip is tested using the internal
  loopback functionality built into it. Testing of backplane
  driver and receiver hardware external to the BIC is not done by
  self-test.

o MIC test: Checks the Memory Interface Circuit (MIC) for
  functional faults. The registers in the MIC which can be read
  and written to are checked for stuck-at-1 and stuck-at-0 faults.
  Only MIC channel B DMA capability is tested between the BIC and
  memory. Some MIC registers cannot be tested (they are write
  only) and the DMA test does not exercise all possible DMA
  configurations (time limitations).

Upon successful completion, self-test will set 'Passed self-test'
status (PST), turn off the failure LED on the card, and invoke the
standard card firmware. The standard firmware will then wait for
the host to 'teach' it its peripheral address (PA).

If self-test fails, the failure LED is left on, the 'Passed
self-test' status (PST) is not set, and the Z80 is 'Halted'.

If the loopback hood is installed and if the card firmware detected
the hood present, the hood LED will be turned on while the
self-test is executing. After a successful self-test completion,
the hood LED will be turned off.

After the self-test, the RAM area will be initialize by the EPROM
firmware. Any data in the RAM area before the self-test will be
lost.

The following time-line illustrates the host / card interactions
during the self-test sequence:

```
     Host              Card
------------      ----------------
              t0

sends card    |
DCL or asserts|
RES after a   |
power-on      |
              | Self-test begun
              |
              t1

Teach card    |
PA.           |
              |
Wait time t2  | Successful:
 or poll PST  |   Set PST.
              |   Turn off LED.
              |
              |
              | Unsuccessful:
              |   Halt Z80.
              |
              t2

where:
  t1 = 1.0 seconds

  t2 = 3 seconds if RES, or 15 seconds if DCL; may be less
```
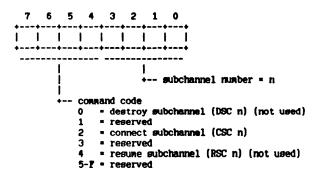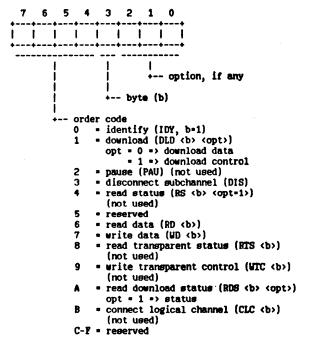
```
+------------------------------------------------+------------------+
|                                                |                  |
|   THE BACKPLANE PROTOCOL                       |   CHAPTER  4     |
|                                                |                  |
+------------------------------------------------+------------------+
```

The protocol being used is a subset of the HP-CIO STANDARD BACKPLANE PROTOCOL FOR SMART CARDS as defined by Greg Dolkas. This subset was created to minimize the significant overhead of the backplane transactions for small buffer sizes (mainly character-at-a-time I/O from terminals).

The backplane protocol is designed to use 1 command, 5 orders, and 3 ARQ's. The command which must always be issued before doing any transaction on the card is the Connect Subchannel (CSC) command. The command should contain the SRQ address in the low nibble of the byte. If this command is not issue, the card has no way of doing an SRQ. The command format is:

```
   7   6   5   4   3   2   1   0
 +---+---+---+---+---+---+---+---+
 |   |   |   |   |   |   |   |   |
 +---+---+---+---+---+---+---+---+

 ----------------   ----------------
        |                  |
        |                  +-- subchannel number = n
        |
        +-- command code
            0  = destroy subchannel (DSC n) (not used)
            1  = reserved
            2  = connect subchannel (CSC n)
            3  = reserved
            4  = resume subchannel (RSC n) (not used)
            5-F = reserved
```
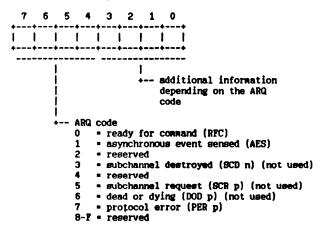
The 6 orders used by the card firmware are IDY, DLD, DIS, RD, WD, and RDS. The IDY is used to obtain the identification information from the card, and the RD and WD order is used to perform all transactions between the card and the host. The DLD order is used to send the download record containing the binary memory image to the card. The RDS order is used to read any download information generated by the card due to some host request. For example, the result of the download verification request. Lastly, the DIS order is used to disconnect the subchannel from the card. The order is not sent to the card until an SRQ has been sent by the card to the host. The format of the order is:

```
 7  6  5  4  3  2  1  0
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
---------------- --- ------------
            |         |
            |         |    +-- option, if any
            |         |
            |         +-- byte (b)
            |
            +-- order code
               0  = identify (IDY, b=1)
               1  = download (DLD <b> <opt>)
                    opt = 0 => download data
                        = 1 => download control
               2  = pause (PAU) (not used)
               3  = disconnect subchannel (DIS)
               4  = read status (RS <b> <opt=1>)
                    (not used)
               5  = reserved
               6  = read data (RD <b>)
               7  = write data (WD <b>)
               8  = read transparent status (RTS <b>)
                    (not used)
               9  = write transparent control (WTC <b>)
                    (not used)
               A  = read download status (RDS <b> <opt>)
                    opt = 1 => status
               B  = connect logical channel (CLC <b>)
                    (not used)
               C-F = reserved
```

If the b bit is set, all data transfer will be in byte mode (8-bit
at a time). If the b bit is clear, all data transfer will be in
word mode (16-bit at a time). The IDY will always transfer its
data in byte mode as specified by the standard.
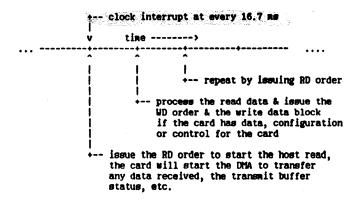
The 3 ARQ codes used are ready for command (RFC), asynchronous
event sensed (AES) and protocol error (PER) status.

```
 7  6  5  4  3  2  1  0
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
---------------- ----------------
            |             |
            |             +-- additional information
            |                 depending on the ARQ
            |                 code
            |
            +-- ARQ code
               0  = ready for command (RFC)
               1  = asynchronous event sensed (AES)
               2  = reserved
               3  = subchannel destroyed (SCD n) (not used)
               4  = reserved
               5  = subchannel request (SCR p) (not used)
               6  = dead or dying (DOD p) (not used)
               7  = protocol error (PER p)
               8-F = reserved
```

The mux card firmware assumes that one subchannel is always
permanently assigned to the card. The driver may ask for the IDY
information at anything. The host driver will periodically poll
the card with the RD order follow by the optional WD order.

The RD order is used to read any data received from the terminal
for all ports and status information from the mux firmware. The WD
order is used to send user data, configuration information, and
control request to the mux. The host is not required to send the
WD order if there is nothing to do.

The following is a simple time diagram of a normal read
follow by the optional write backplane protocol.

```
+-- clock interrupt at every 16.7 ms
|
v        time -------->
...  -----------+---------+---------+----------+---------  ....
           ^         ^         ^
           |         |         |
           |         |         +-- repeat by issuing RD order
           |         |
           |         +-- process the read data & issue the
           |             WD order & the write data block
           |             if the card has data, configuration
           |             or control for the card
           |
           +-- issue the RD order to start the host read,
               the card will start the DMA to transfer
               any data received, the transmit buffer
               status, etc.
```

The following sections describe the protocol implemented by this
card.

4.1   Subchannel Connection

The subchannel connection must always be performed first before
issuing any order.

host		card	description
	<--	ARQ,RFC	Card waits for the valid peripheral address before asserting RFC to allow the host to perform a subchannel connection request.
	.		
	.		
	.		
CSC n	-->		Channel initiates a subchannel connection.
	<--	SRQ	The card accepts the connection request and asks for the next order.
	.		
	.		
	.		

or

CSC n	-->		Channel initiates a subchannel connection.
	<--	ARQ,PER	The card rejects the connection request because it is already connected.
	.		
	.		
	.		

4.2   Subchannel Disconnect

This is provided for compatibility with the various computer system.

```
host          card    description
----          ----    -----------

       .
       .
       .
DIS    -->            The host wants to disconnect
                     the subchannel.  The card does
                     this by removing the SRQ address.
                     No SRQ is sent to the host for
                     the next order.
```

4.3   IDY Transaction

```
host          card    description
----          ----    -----------
       .
       .
       .
IDY    -->            Host sends IDY order to the
                     card to get the card ID
                     information block.

       <-- ID blk,   The card respond with the ID
           END        block information.

       <-- SRQ        Card requests next order.

       .
       .
       .
```

4.4 Read Data/Status Transaction

host	card	description
	.	
	.	
	.	
RD	-->	Host wants to read the data and any status information from the card.
	<-- data/ status,	Transfer the data/status to the host.
		Note that the card can send up to 8K of data. The host should allocate enough buffer space for this worse case; otherwise, data will be lost and the block terminator will be missing from the buffer.
	<-- SRQ	Card requests next order.
	.	
	.	
	.	

4.5 Write Data/Configuration/Control Transaction

This transaction is optional. The host does not need to send this transaction if there is no data, configuration, or control request for the card.

host	card	transaction
	.	
	.	
	.	
WD	-->	The host wants to write data, configuration, or control to the card.
	<-- SRQ	The card will send an SRQ when its DMA is setup and ready to receive data.
data, config, control, END	-->	Transfer the data, configuration, and control to the card.
	<-- SRQ	Card requests the next order.
	.	
	.	
	.	

4.6   Download Transaction

```
host          card    description
----          ----    -----------
                .
                .
                .

DLD,1  -->            The host wants to send a
                      download control request
                      to the card.

       <-- SRQ        The card will send an SRQ
                      when its DMA is setup and
                      ready to receive the data.

crtl,  -->            The host sends the download
END                   control request block to the
                      card requesting the card to
                      initialize for downloading.

       <-- SRQ        Card requests next order.

+-->DLD,1  -->        The host wants to send a
|                     download control request
|                     to the card.
|
|      <-- SRQ        The card will send an SRQ
|                     when its DMA is setup and
|                     ready to receive the data.
|
|  crtl,  -->         The host sends the download
|  END                control request block to the
|                     card to set the beginning
|                     RAM address and the download
|                     record length for the data
|                     block that will be sent by
|                     DLD,0 order.
|
|      <-- SRQ        Card requests next order.
|
|  DLD,0  -->         The host wants to send the
|                     download record.
|
|      <-- SRQ        The card will send an SRQ
|                     when its DMA is setup and
|                     ready to receive the data.
|
```

.
.
.

The downloaded firmware should be
running at this point.

4.7 ARQ Asynchronous Event Sensed (AES)

The ARQ, asynchronous event sensed status will be sent to the host
whenever the modem line changes state or whenever a data character
is received for any port which has the corresponding interrupt
enabled. However, only one ARQ will be sent between each read
data/status transaction for any event. That is, if the modem line
on one port changes state and if data was received on another port
or the same port, only one ARQ will be sent to the host. The host
read buffer may contain more than one event. The ARQ,AES is
basically used to wake up the host or to tell the host to do a read
before the next poll.

The low nibble of the ARQ code will specify which event was sensed
to generate the interrupt.

    ARQ
    code   description

    11H   data is available

    14H   modem signal changed state

Note for the data available asynchronous event sensed ARQ there
could be a delay of up to 10 milliseconds from the time the card
receive the first character to the time the host sees the ARQ. The
reason is that the data available ARQ is sent in the clock service
routine instead of in the receive service routine. This is done to
keep the receive service routine exectuion time to a minimum.

The two events may be selectively enable or disable by using the
write control request to set or clear the corresponding bit in the
host interrupt mask, see chapter 5 for additional details.

4.8 ARQ Protocol Error (PER)

The ARQ, protocol error status will be sent to the host for the
following conditions. The low nibble of the byte will specify the
type of error.

    ARQ
    code   description

    71H   Received an unimplemented command or
            an invalid command.

72H	Received an unimplemented order or an invalid order.
73H	Subchannel connect request when the card is already connected.
74H	Invalid port ID.
75H	Invalid subfunction code.
76H	Invalid length or length error.
77H	Transmit buffer overflow.
78H	Invalid parameter.

NOTE   THE   FIRMWARE   STOPS   PROCESSING   THE   WRITE
DATA/CONFIGURATION/CONTROL TRANSACTION UPON ENCOUNTERING THE FIRST
ERROR.
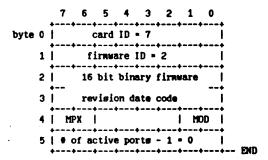
```
+-----------------------------------------------+------------------+
|                                               |                  |
|  THE TRANSACTION DATA BLOCK FORMAT            |   CHAPTER  5     |
|                                               |                  |
+-----------------------------------------------+------------------+
```

This chapter will describe the data block format for each of the 3
available transaction.

5.1   IDY Data Block Format

The following identity information blocks may be returned by the
Mux card for the IDY order.

For the EPROM firmware:

```
          7   6   5   4   3   2   1   0
        +---+---+---+---+---+---+---+---+
byte 0 |          card ID = 7          |
        +---+---+---+---+---+---+---+---+
     1 |        firmware ID = 2        |
        +---+---+---+---+---+---+---+---+
     2 |      16 bit binary firmware   |
        +--                          --+
     3 |      revision date code       |
        +---+---+---+---+---+---+---+---+
     4 | MPX |               | MOD |
        +---+---+---+---+---+---+---+---+
     5 | # of active ports - 1 = 0     |
        +---+---+---+---+---+---+---+---+-- END
```

        MPX = 3 for other backplane protocol

        MOD = 2 for both byte and word mode data transfers

For the downloaded Mux firmware:

```
           7   6   5   4   3   2   1   0
         +---+---+---+---+---+---+---+---+
byte 0 |        card ID = 7            |
         +---+---+---+---+---+---+---+---+
     1 |        firmware ID = 3        |
         +---+---+---+---+---+---+---+---+
     2 |      16 bit binary firmware   |
         +--                         --+
     3 |       revision date code      |
         +---+---+---+---+---+---+---+---+
     4 | MPX   |           | MOD |
         +---+---+---+---+---+---+---+---+
     5 | # of active ports - 1 = 5     |
         +---+---+---+---+---+---+---+---+--END
```

MPX = 3 for other backplane protocol

MOD = 2 for both byte and word mode data transfers

5.2   Read Data/Status Transaction

The format of the read data/status block send to the host after receiving the RD order is:

```
+---+---+---+---+---+---+---+\
| 0 |      port number      | \
+---+---+---+---+---+---+---+ |
|       attribute           | |   data block -
+---+---+---+---+---+---+---+ >-- each character
|     # of data char = 1    | |   receive from
+---+---+---+---+---+---+---+ |   each port consists
|      data character       | /   of 4 bytes as
+---+---+---+---+---+---+---+/    described here
| 0 |      port number      |
+---+---+---+---+---+---+---+
|       attribute           |
+---+---+---+---+---+---+---+
|     # of data char = 1    |
+---+---+---+---+---+---+---+
|      data character       |
+---+---+---+---+---+---+---+

               .
               .
               .

+---+---+---+---+---+---+---+\
| 1 |      port number      | \
+---+---+---+---+---+---+---+ |
|       status code         | |
+---+---+---+---+---+---+---+ |
| # of bytes in additional info | >-- status block -
+---+---+---+---+---+---+---+ |   bit 7 is always set in
|      additional           | |   the port number byte,
/      information          / |   additional information
/      for status code,    / |   will follow the count
|      if any               | /   if any
+---+---+---+---+---+---+---+/

               .
               .
               .

+---+---+---+---+---+---+---+
| 1   1   1   1   1   1   1   1 | <-- end of block
+---+---+---+---+---+---+---+
```

The read data/status block contains data and status information for

one or more multiplexer ports. Each byte received from the terminal will be formatted into a 4 byte quad containing the port number, the attribute, and the data character itself as shown above. The data block has no order and will appear in the order that it was received from the terminal connected to the mux card.

If no data character is available, the read data/status block will never be empty. There will always be the status information for each port containing the transmit buffer status. The additional information for each status block is variable length. The length will depend on the status code as described below.
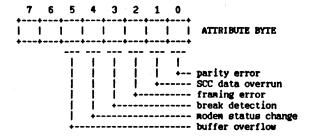
The read data/status block will always be terminated by a byte containing OFFH.

5.2.1 The Data Block Portion of the Read Transaction

The attribute byte gives the attributes of the data character that was just received from the terminal or other condition that occurred on the card. If the attribute is zero, then the data character has no error or other special condition.

In the future if the number of data character count is greater than 1, then the attribute byte applys to the last character in the data block. In this case all characters before the last one will have an attribute of zero indicating no errors.

The attribute byte is defined as follow:

```
     7   6   5   4   3   2   1   0
   +---+---+---+---+---+---+---+---+
   |   |   |   |   |   |   |   |   |   ATTRIBUTE BYTE
   +---+---+---+---+---+---+---+---+
       --- --- --- --- --- ---
        |   |   |   |   |   |
        |   |   |   |   |   +-- parity error
        |   |   |   |   +------ SCC data overrun
        |   |   |   +---------- framing error
        |   |   +-------------- break detection
        |   +------------------ modem status change
        +---------------------- buffer overflow
```

Usually only one bit in the attribute byte is set at any given time per data block.

If the parity error bit is set, the character received contains a parity error.

If the SCC data overrun bit is set, the card firmware was not able
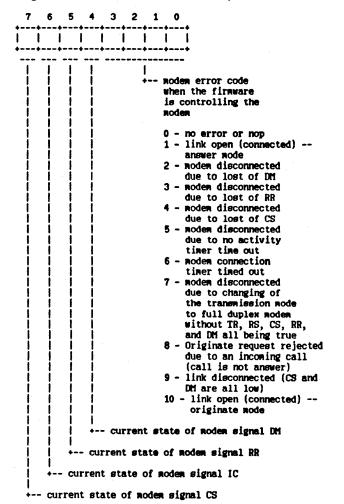
to read the character from the SCC fast enough. This means that a character was received when the SCC FIFO buffer is full leading to the loss of a character. The character associated with this attribute is the last good character received before data loss started.

If the framing error is set, the character received contains a framing error.

If the break detection bit is set, the firmware has detected a break condition. The data character will be a null in this case.

If the modem status change bit is set, the firmware will return the following information in the data character byte.

```
 7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
 --- --- --- --- ----------------
  |   |   |   |       |
  |   |   |   |       +-- modem error code
  |   |   |   |          when the firmware
  |   |   |   |          is controlling the
  |   |   |   |          modem
  |   |   |   |
  |   |   |   |          0 - no error or nop
  |   |   |   |          1 - link open (connected) --
  |   |   |   |              answer mode
  |   |   |   |          2 - modem disconnected
  |   |   |   |              due to lost of DM
  |   |   |   |          3 - modem disconnected
  |   |   |   |              due to lost of RR
  |   |   |   |          4 - modem disconnected
  |   |   |   |              due to lost of CS
  |   |   |   |          5 - modem disconnected
  |   |   |   |              due to no activity
  |   |   |   |              timer time out
  |   |   |   |          6 - modem connection
  |   |   |   |              timer timed out
  |   |   |   |          7 - modem disconnected
  |   |   |   |              due to changing of
  |   |   |   |              the transmission mode
  |   |   |   |              to full duplex modem
  |   |   |   |              without TR, RS, CS, RR,
  |   |   |   |              and DM all being true
  |   |   |   |          8 - Originate request rejected
  |   |   |   |              due to an incoming call
  |   |   |   |              (call is not answer)
  |   |   |   |          9 - link disconnected (CS and
  |   |   |   |              DM are all low)
  |   |   |   |          10 - link open (connected) --
  |   |   |   |               originate mode
  |   |   |   |
  |   |   |   +-- current state of modem signal DM
  |   |   |
  |   |   +-- current state of modem signal RR
  |   |
  |   +-- current state of modem signal IC
  |
  +-- current state of modem signal CS
```

When the firmware is controlling the modem by setting the

transmission mode to full duplex modem and when the modem status change bit is set, the modem error code will always be nonzero. If the card is not controlling the modem, the modem status change bit will indicate that there was a change in the modem signal. The data character will contain the current state of the modem signal after the change.

If enabled, the firmware will return an ARQ,AES interrupt when any modem line changes state for any of the port.

If the buffer overflow bit is set, the card ran out of buffer space when the character was received for that port. The data character in the quad is the last valid character for the port. There will be data lost afterward.

5.2.2   The Status Block Portion of the Read Transaction

The status block returns the transmit buffer status and any other status information requested by the host.

The following is a summary of all the status that may be returned by the card.

status code	description
128	the number of bytes remaining in the transmit buffer for transmission
141	the current modem signal state
142	the current link connection status
143	the requested RAM content
144	backplane drivers verification

5.2.2.1   Status 128. Remaining transmission count.

This status is returned to the host to allow the driver to pace its write data to the card. The card will use 1 byte in the additonal information portion of the status block. The 8-bit quantity will indicate the number of bytes remaining in the transmit buffer that has not been transmitted to the device.

The frequency of sending this status is controlled by subfunctions 154 and 155, described below. Subfunction 154 enable sending this status on every read data/status transaction. Subfunction 155 enable sending the status only when a change occurred. These subfunctions are mutually exclusive and only one will be active.

SUBFUNCTION 154 # 155
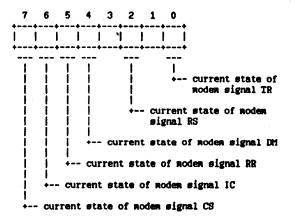
NOTE THAT THE SCC HAS THE CAPABILITY TO BUFFERED UP TO 2 DATA BYTES. This means that when the remaining count is zero, there may be up to 2 bytes not transmitted to the connected device.

The significant of the above statement is that the host must wait at least 2 character time before making any changes that will affect the SCC. This includes changing the character length, the parity, the stop bits, the baud rate or generating a break. If the changes were made to the SCC before all the characters are transmitted, the data bytes remaining in the buffer may be lost.

5.2.2.2   Status 141. Modem signal state.

This status is returned only when the control request with subfunction code 141 is issued by the host in the write data/configuration/control transaction. The card will return 1 byte containing the current state of the modem lines. The format of the byte:

```
  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
 --- --- --- ---     ---     ---
  |   |   |   |       |       |
  |   |   |   |       |       +-- current state of
  |   |   |   |       |           modem signal TR
  |   |   |   |       |
  |   |   |   |       +-- current state of modem
  |   |   |   |           signal RS
  |   |   |   |
  |   |   |   +-- current state of modem signal DM
  |   |   |
  |   |   +-- current state of modem signal RR
  |   |
  |   +-- current state of modem signal IC
  |
  +-- current state of modem signal CS
```

5.2.2.3   Status 142. Link connection status.

This status is returned only when the control request with subfunction code 142 is issued by the host in the write data/configuration/control transaction. The card will return 1 byte containing the current connection status. If the value is zero, the link is disconnected. If the value is an one, the link is connected.

5.2.2.4   Status 143. RAM content.

This status is returned only when the control request with subfunction code 143 is issued by the host in the write data/configuration/control transaction. The card will return the content of RAM requested by the host
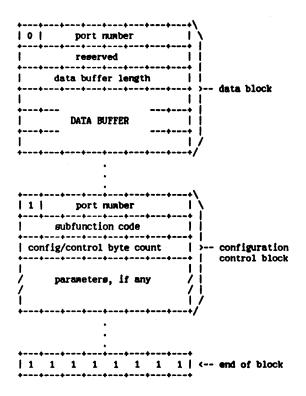
5.2.2.5   Status 144. Backplane drivers verification.

This status is returned only when the control request with subfunction code 144 is issued by the host in the write data/configuration/control transaction. The card will return the data that was written to the card with the control request. This is used to verify the proper operation of the backplane drivers.

Note that there will always be data returned. A data byte count of zero implys 256 for this status.

5.3   The Write Data/Configuration/Control Transaction

The format of the write data/configuration/control block is:

```
+---+---+---+---+---+---+---+\
| 0 |     port number     | |\
+---+---+---+---+---+---+---+ |
|        reserved           | |
+---+---+---+---+---+---+---+ |
|     data buffer length    | |
+---+---+---+---+---+---+---+ >-- data block
|                           | |
+---+---             ---+---+ |
|        DATA BUFFER        | |
+---+---             ---+---+ |
|                           | /
+---+---+---+---+---+---+---+/

              .
              .
              .

+---+---+---+---+---+---+---+\
| 1 |     port number     | |\
+---+---+---+---+---+---+---+ |
|      subfunction code     | |
+---+---+---+---+---+---+---+ |
|   config/control byte count| >-- configuration
+---+---+---+---+---+---+---+ |    control block
|                           | |
/    parameters, if any    / |
/                          / |
|                           | /
+---+---+---+---+---+---+---+/

              .
              .
              .

+---+---+---+---+---+---+---+
| 1   1   1   1   1   1   1   1 | <-- end of block
+---+---+---+---+---+---+---+
```

The write data/configuration/control block contains data and/or configuration and/or control information for one or more multiplexer ports. The write transaction is optional. It is not required by the firmware when the host has nothing for the card. The block is terminated by a byte containing a 0FFH.

Subfunctions and parameters are used to specify the configuration

(e.g. baud rates, parity etc.) and the control information (e.g. set TR modem line high).

The total length of the write data/configuration/control transaction block should not exceed 4096 bytes. This includes the end of block terminating character.

5.3.1   The Data Block Portion of the Write Transaction

The data block is specified when bit 7 is cleared. The port number specify which port will receive the transmit data. The data buffer length specify the number of characters for the transmitter. The length may be from 1 to 255 characters. If the host should specify an invalid port number or if the data buffer length in the data block is zero or if the data buffer length should exceed the write transaction block length or if the transmit buffer is full, the firmware will return the ARQ protocol error. NOTE, IT IS LEFT UP TO THE HOST NOT TO EXCEED THE CARD TRANSMIT BUFFER OF 255 BYTES.

5.3.2   The Configuration/Control Portion of the Write Transaction

The configuration/control block is specified when bit 7 is set. The port number specify which port will be configure or will receive the control request. If the subfunction code is invalid or if the configuration/control byte count is invalid or if the parameter for the subfunction is invalid, an ARQ protocol error will be return.

The following is a summary of all the subfunction available:
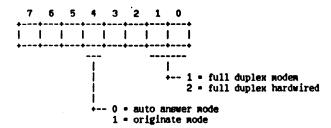
subfunction	description
	configuration
1	modem connection & transmission mode
2	baud rate
3	char len, stop bit, & parity
4	modem connection timer
5	no activity timer
6	lost receiver ready timer
7	device controlled X-ON/X-OFF characters
8	host controlled X-ON/X-OFF characters
	control
128	open (connect) the modem link
129	close (disconnect) the modem link
130	start the break generation
131	stop the break generation
132	clear the RS modem signal
133	set the RS modem signal
134	clear the TR modem signal
135	set the TR modem signal
136	clear the SR modem signal
137	set the SR modem signal
138	flush the transmit buffer
139	stop the transmitter
140	restart the transmitter
141	get the current modem signal state
142	get the link connection status
143	get the card RAM content
144	backplane drivers verification
145	card write register
146	set the host interrupt mask
147	disable the SCC receiver
148	enable the SCC receiver
149	send the host controlled X-ON character immediately
150	send the host controlled X-OFF character immediately
151	disable the modem quad generation control
152	enable the modem quad generation contrl
153	frontplane modem interrupt control mask
154	send status 128 on every read
155	send status 128 on changes only

5.3.2.1   Subfunction 1. Modem connection and transmission mode.

Parameter

```
    7   6   5   4   3   2   1   0
  +---+---+---+---+---+---+---+---+
  |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
            ---         -------
             |             |
             |             +-- 1 = full duplex modem
             |                 2 = full duplex hardwired
             |
             +-- 0 = auto answer mode
                 1 = originate mode
```

When the user switch the transmission mode from full duplex modem to full duplex hardwired, the firmware will clear all the modem timers, enable the transmitter and receiver, and clear the IC polling if it is active. The firmware will not change the state of any of the modem signals.

When the user switch from full duplex hardwired to full duplex modem, the firmware do one of 2 things depending on the state of the modem signals. If modem signals TR, RS, DM, CS, and RR are all asserted, the firmware will send back the modem change state data block with the link connected for the modem error code. No ARQ,AES will be sent to the host for this implied connection. The user will receive the modem change state data block with the link connected code in the next read data/status transaction.
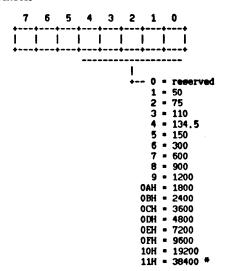
If any of the modem signals are false, the firmware will perform a modem disconnect and return a modem status (error code 7) indicating the link was disconnected due to no TR and RS being true. No ARQ,AES will be sent to the host for this implied disconnection. The user will receive the modem change state data block with the link disconnected code in the next read data/status transaction.

NOTE THE FRONTPLANE MODEM CONTROL INTERRUPT IS AUTOMATICALLY ENABLED FOR ALL MODEM SIGNALS WHEN THE TRANSMISSION MODE IS CHANGED TO FULL DUPLEX MODEM.

IS TX BUFFER FLUSHED?

5.3.2.2 Subfunction 2. Baud rate.

Parameter

```
     7   6   5   4   3   2   1   0
   +---+---+---+---+---+---+---+---+
   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+

                 -------------------
                 |
                 +-- 0 = reserved
                     1 = 50
                     2 = 75
                     3 = 110
                     4 = 134.5
                     5 = 150
                     6 = 300
                     7 = 600
                     8 = 900
                     9 = 1200
                   0AH = 1800
                   0BH = 2400
                   0CH = 3600
                   0DH = 4800
                   0EH = 7200
                   0FH = 9600
                   10H = 19200
                   11H = 38400 *
```
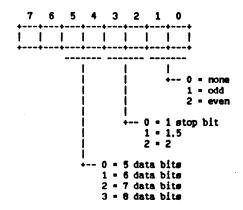
Note that the mux card does not offically support 38.4 Kbaud. This
high speed rate is included for experimental purposes.

5.3.2.3 Subfunction 3. Character length, stop bits, & parity.

Parameter

```
     7   6   5   4   3   2   1   0
   +---+---+---+---+---+---+---+---+
   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+

             -------- ------- -------
             |        |       |
             |        |       +-- 0 = none
             |        |           1 = odd
             |        |           2 = even
             |        |
             |        +-- 0 = 1 stop bit
             |            1 = 1.5
             |            2 = 2
             |
             +-- 0 = 5 data bits
                 1 = 6 data bits
                 2 = 7 data bits
                 3 = 8 data bits
```

5.3.2.4 Subfunction 4. Modem connection timer.

The clock resolution is 1 second. One byte of the parameter
portion of the subfunction block is required to programmed the
clock from 0 to 255 seconds.

5.3.2.5 Subfunction 5. No activity timer.

The clock resolution is 1 second. Two byte of the parameter
portion of the subfunction block is required to programmed the
clock from 0 to 65,535 seconds. The high byte of the 16-bit
quantity should be sent to the card first follow by the low byte.

5.3.2.6 Subfunction 6. Lost receiver timer.

The clock resolution is 10 milliseconds. One byte of the parameter
portion of the subfunction block is required to program the clock
from 0 to 2550 milliseconds (0 to 255 counts of 10's milliseconds).

5.3.2.7 Subfunction 7. Device controlled X-ON/X-OFF characters.

parameters

    byte 0: character for device X-ON function
         1: character for device X-OFF function

Note that a OFFH character disable the corresponding function for handshaking. For example, a OFFH for the X-ON function will disable the receiver from performing the X-ON function to restart the transmitter. In addition the OFFH character can never be use as a handshaking character.

5.3.2.8 Subfunction 8. Host controlled X-ON/X-OFF characters.

parameters

    byte 0: character for host X-ON function
         1: character for host X-OFF function

5.3.2.9 Subfunction 128. Open (connect) the modem link.

This control request requires no parameter. It is used to tell the firmware to start the modem connection sequence. If the transmission mode is not full duplex modem, this request will be a nop. The firmware will return a data block with the modem change status bit set when the connection is made or when an error is encountered.

If the generate ARQ,AES for modem signal change state bit is set in the host interrupt mask, the firmware will send an ARQ,AES when the connection is made or when an error condition occurred.

5.3.2.10 Subfunction 129. Close (disconnect) the modem link.

This control request requires no parameter. It is used to disconnect the modem by clearing the TR and RS modem signal if the modem was connected. If the transmission mode is not full duplex modem or if the modem link is disconnected already, this request will be treated as a nop.

If the generate ARQ,AES for modem signal change state bit is set in the host interrupt mask, the firmware will send an ARQ,AES when the link is disconnected.

5.3.2.11 Subfunction 130. Start the break generation.

This control request requires no parameter. It is used to start the break generation. The length of the break generation is determined by host.

5.3.2.12 Subfunction 131. Stop the break generation.

This control request requires no parameter. It is used to stop the break generation.

5.3.2.13 Subfunction 132. Clear the RS modem signal.

This control request requires no parameter. The firmware will clear the RS modem line.

5.3.2.14 Subfunction 133. Set the RS modem signal.

This control request requires no parameter. The firmware will set the RS modem line.

5.3.2.15 Subfunction 134. Clear the TR modem signal.

This control request requires no parameter. The firmware will clear the TR modem line.

5.3.2.16 Subfunction 135. Set the TR modem signal.

This control request requires no parameter. The firmware will set the TR modem line.

5.3.2.17 Subfunction 136. Clear the SR modem signal.

This control request requires no parameter. The firmware will clear the SR modem line.

5.3.2.18 Subfunction 137. Set the SR modem signal.

This control request requires no parameter. The firmware will set the SR modem line.

5.3.2.19   Subfunction 138. Flush the transmit buffer.

This control request requires no parameter. All data in the transmit buffer which has not been transmitted will be flushed. After this control request has been executed, the transmit buffer will be empty.

Note that if transmit data are included in the same write transaction which contains this control request, that data will be flushed if it occurs before this control request.

5.3.2.20   Subfunction 139. Stop the transmitter.

This control request requires no parameter. The firmware will stop transmitting data when this control request is received. The transmitter will be put into a state like if an X-OFF was received.

5.3.2.21   Subfunction 140. Restart the transmitter.

This control request requires no parameter. The firmware will restart the transmitter to continue transmitting data from where it was stopped.

5.3.2.22   Subfunction 141. Get the current modem signal state.

This control request requires no parameter. The firmware will return the modem line state on the next read data/status transaction with status code of 141.

Note that the modem line state will be the state at the time the control request was executed not when the status is returned to the host.

5.3.2.23   Subfunction 142. Get the link connection status.

This control request requires no parameter. The firmware will return the current link connection state in the next read data/status transaction. If the transmission mode is full duplex modem, the status will be zero if the link is not connected or is in the process of being disconnected. It will be a one if the link is connected. If the transmission mode is full duplex hardwired, the status will always be an one.

WHAT ABOUT THE 2 BYTES IN THE SCC?

5.3.2.24   Subfunction 143. Get the card RAM content.

This control request requires 3 bytes in the parameter portion of the contorl block.

parameter

```
     7   6   5   4   3   2   1   0
   +---+---+---+---+---+---+---+---+
   |  high byte of RAM address     |
   +--                           --+
   |  low byte of RAM address      |
   +---+---+---+---+---+---+---+---+
   |            length             |
   +---+---+---+---+---+---+---+---+
```

The first 2 bytes are used to specify what portion of the card RAM is to be returned to the host. The last byte is used to specify how much data is to be returned. A zero implys 256 bytes.

This control request is being used for firmware debugging and maintenance purposes and should not be used by the user in general.

Note there should be only one get the card RAM content request per write transaction block regardless of the port ID. If this restriction is not observed, then the status buffer may overflow without warning leading to firmware failure.

5.3.2.25   Subfunction 144. Backplane drivers verification.

This control request is used mainly for diagnostic purposes to verify the backplane drivers. The host will write a known pattern to the card. The card will return the data on the next read transaction to the host for verification. Note that the parameter block will always have data, and the parameter block is limited to 255 bytes.

5.3.2.26   Subfunction 145. Card write register.

This subfunction should not be accessible by the general users. The purpose of this subfunction is to allow for on-line card diagnostics capability.

With this request the state of several hardware functions can be changed from the power-up default states. These are:

   o On-card LED
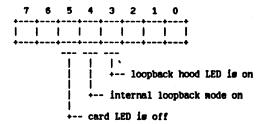
o Hood LED

o Internal loopback mode (loopback of data on the card)

There are two useful loopback modes available. If an on-line test is desired, the internal loopback mode can be enabled and the line drivers disabled. This loops data back within the card (all 6 channels together but transmit independently) and does not send data out the front edge connector.
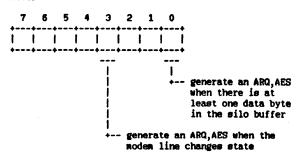
The other mode is used when a loopback hood is installed on the card. For this mode the internal loopback mode should be disabled. This loops data through the data line drivers and back through the receivers.

parameter

```
  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
    --- --- ---
     |   |   | `
     |   |   +-- loopback hood LED is on
     |   |
     |   +-- internal loopback mode on
     |
     +-- card LED is off
```

5.3.2.27   Subfunction 146. Set the host interrupt mask.

parameter

```
  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
        ---         ---
         |           |
         |           +-- generate an ARQ,AES
         |               when there is at
         |               least one data byte
         |               in the silo buffer
         |
         +-- generate an ARQ,AES when the
             modem line changes state
```

5.3.2.28   Subfunction 147. Disable the SCC receiver.

This control request requires no parameter. The SCC receiver enable bit will be cleared to disable the receiver from receiving any incoming data.

5.3.2.29   Subfunction 148. Enable the SCC receiver.

This control request requires no parameter. The SCC receiver enable bit will be set to allow the SCC to process all incoming data.

5.3.2.30   Subfunction 149. Send host X-ON character immediately.

This control request requires no parameter. The card firmware will send the host X-ON character immediately upon receiving this request. The X-ON will be sent ahead of any transmit data that might be queued for transmission. This request will ignore any X-OFF that may have been sent by the connected device.

Since the SCC can buffered up to 2 characters, there may be up to 2 character time delay before the X-ON is actually transmitted.

5.3.2.31   Subfunction 150. Send host X-OFF character immediately.

This subfunction is similar to subfucntion 149, except the host X-OFF character will be sent instead of the host X-ON character.

5.3.2.32   Subfunction 151. Disable modem quad generation control.

This control request requires no parameter. Upon receiving this request, the firmware will generate a quad whenever the modem signal changes state subject to the frontplane modem interrupt control mask. This is the normal operating state of the firmware.

5.3.2.33   Subfunction 152. Enable modem quad generation control.
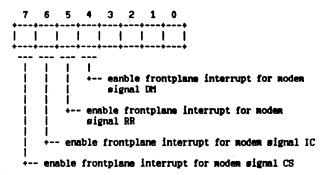
This control request requires no parameter. Upon receiving this request the firmware generate only one quad per port per read data/status transaction and only if the modem line changed state from the previous quad generated.

5.3.2.34 Subfunction 153. Frontplane modem interrupt ctrl mask.

parameter

```
      7   6   5   4   3   2   1   0
    +---+---+---+---+---+---+---+---+
    |   |   |   |   |   |   |   |   |
    +---+---+---+---+---+---+---+---+
    --- --- --- ---
     |   |   |   |
     |   |   |   +-- eanble frontplane interrupt for modem
     |   |   |       signal DM
     |   |   |
     |   |   +-- enable frontplane interrupt for modem
     |   |       signal RR
     |   |
     |   +-- enable frontplane interrupt for modem signal IC
     |
     +-- enable frontplane interrupt for modem signal CS
```

NOTE ALL FRONTPLANE MODEM CONTROL INTERRUPT ARE ENABLED AUTOMATICALLY WHEN THE TRANSMISSION MODE IS SET TO FULL DUPLEX MODEM AND THE ABOVE MASK IS SET TO ALL ONES.

5.3.2.35 Subfunction 154. Send Status 128 on every read.

This control request requires no parameter. Upon receiving this request the software will send the Status 128, the transmission remaining count, on every read data/status transaction. This is the default.

5.3.2.36 Subfunction 155. Send Status 128 only on changes.

This control request requires no parameter. Upon receiving this request the software will generate the Status 128 if a change has occurred from the last transmission remaining count sent to the host.

5.4 The Download Transaction

Two orders are used to handle all the download transaction. The option field of the download (DLD) and the read download status (RDS) is used to specify what type of data is being transfer on the backplane.

The RDS order has only 1 option defined for the mux card. When the option field is equal to 1, the return data is a status block.

The DLD order has 2 options defined for the mux card. If the option field is equal to 0, the data block being written to the card contains the memory image of the Z80 code for the card RAM. If the option field is equal to 1, the written data is a control request.

All other undefined options will return an ARQ protocol error.

5.4.1 Read download status with option 1.

The format of the status block is:

```
+---+---+---+---+---+---+---+
|      status code          |
+---+---+---+---+---+---+---+
|  additional info byte count |
+---+---+---+---+---+---+---+
|                           |
/   additional info, if any /
/                           /
|                           |
+---+---+---+---+---+---+---+
```

Only one status code is defined for the mux card. When the status code is 3, the card is returning the result of the download verification.
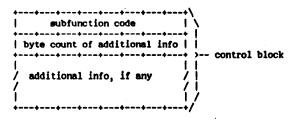
If the read download status order is issued without a previous download control request with subfunction 3, the firmware will return one byte with all one's in place of the status code indicating no status.

5.4.2 Download order with option 0.

When the option code is zero, the data block for the download order will contain the memory image Z80 code for the card RAM. The length of the data block must be the same as the length specified in the download control request used to set the beginning RAM address and the download record length.

5.4.3 Download order with option 1.

The format of the control block is:

```
+---+---+---+---+---+---+---+\
|      subfunction code     | \
+---+---+---+---+---+---+---+  |
| byte count of additional info |  |
+---+---+---+---+---+---+---+  >-- control block
|                           |  |
/ additional info, if any   /  |
/                           /  |
|                           |  /
+---+---+---+---+---+---+---+/
```

The following is a summary of all the subfunction for the download control request:

subfunction	description
1	initialize for downloading
2	set beginning RAM address & length
3	verify the download firmware
4	start RAM execution

5.4.3.1 Subfunction 1. Initialize for downloading.

This control request requires no parameter. The firmware will initialize the card for downloading by resetting the SCC to turn off the transmitters and receivers on the frontplane and by initializing the RAM variables required by the downloader.

NOTE: AFTER GIVING THIS REQUEST NO FURTHER READ AND WRITE ORDER SHOULD BE SENT TO THE CARD UNTIL AFTER THE DOWNLOAD IS DONE AND THE RAM CODE IS STARTED.

5.4.3.2 Subfunction 2. Set the starting RAM address and length.

This control request requires 8 bytes in the additional information portion of the control block to set the starting RAM address and the length for the next download record that will be sent by the DLD order. If this request does not precede each download record, the download record will be loaded into the same location specified by the previous control request thus overwriting previous data.

additional information

```
  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|              0                |
+--                          --+
|              0                |
+--                          --+
|    high byte of RAM location  |
+--                          --+
|    low byte of RAM location   |
+---+---+---+---+---+---+---+---+
|              0                |
+--                          --+
|              0                |
+--                          --+
|    high byte of record length |
+--                          --+
|    low byte of record length  |
+---+---+---+---+---+---+---+---+
```

5.4.3.3 Subfunction 3. Verify download.

This control request requires no parameter. The host should send this request after all the download records have been sent to the card. The card will perform a CRC checksum of the firmware and return the result to the host.

The host reads the result by using the read download status (RDS) order. The firmware returns 1 byte in the additional information portion of the status block. A zero value implys no error.

5.4.3.4   Subfunction 4.   Start RAM execution.

This control request requires 4 bytes in the parameter portion of
the block. The host uses this request to tell the card to start
executing the downloaded RAM code at the starting address given in
the additional information block. This control request should be
in a write transaction block by itself. Once this control request
is executed by the card, no further subfunction in the same
transaction block will be executed.

additional information

```
    7   6   5   4   3   2   1   0
  +---+---+---+---+---+---+---+---+
  |                0              |
  +--+                         +--+
  |                0              |
  +--+                         +--+
  | high byte of starting RAM addr|
  +--+                         +--+
  | low byte of starting RAM addr |
  +---+---+---+---+---+---+---+---+
```

```
+-----------------------------------------------+------------------+
|                                               |                  |
|   DEFAULT CARD CONFIGURATION                  |    CHAPTER  6    |
|                                               |                  |
+-----------------------------------------------+------------------+
```

Upon a reset or power up condition, the card will be set to a known
state defined below.

   o modem lines TR and RS are cleared

   o modem line SR is set

   o modem connection mode: answer mode

   o transmission mode: full duplex hardwired

   o baud rate: 9600

   o character length: 8 bits/char

   o # of stop bits: 1

   o parity: none

   o modem connection timer: 25 seconds

   o no activity timer: 600 seconds

   o lost receiver ready timer: 400 milliseconds (40 counts)

   o device controlled X-ON character: 0FFH imply disable

   o device controlled X-OFF character: 0FFH imply disable

   o host controlled X-ON character: 11H (DC1)

   o host controlled X-OFF character: 13H (DC3)

   o asynchronous event sensed interrupt: disable

   o SCC receiver: enable for EPROM firmware
                   disable for downloaded firmware

   o modem quad generation control: disable

   o frontplane modem interrupt control mask: all disable

```
+--------------------------------------------+-------------------+
|                                            |                   |
|  DOWNLOADABLE MUX CARD                     |  CHAPTER  7       |
|                                            |                   |
+--------------------------------------------+-------------------+
```

When the downloadable Mux card is installed in the system, the EPROM firmware does not contain the full Mux firmware. The firmware included on the EPROM is enough to do the following:

    o a subset of the Mux firmware to handle port 0,

    o the self-test to check the hardware of the card
      to give a go/no-go condition, and

    o the downloading code to load the regular Mux
      firmware.

The reason for making the Mux card downloadable is for maintainability. Whenever a bug is found in the Mux firmware or whenever there is some new enhancement, a new firmware code file can be generated rather quickly for the system division without having to burn a new EPROM. This allows the users to run the latest firmware whenever there is a new update.

This chapter will describe the code file format as generated by RND and the firmware features available for port 0 on initial power up.

7.1 The Code File Format

The code file distributed to each system division will be in the Intel Intellec 8/MDS format or commonly called the Intel hex format. It is up to each system division to convert the information into the binary form required by the Mux card.

The Intel hex formatted file contains 3 different record types. The first type is a comment record which is distinguished from the remaining 2 records by not having a ":" (colon) in column 1. The other 2 record types which are the data and end-of-file records will alway have a ":" in column 1.

Intel Hex File Format

```
comment record 1 ...
comment record 2 ...

       .

       .

comment record n ...
:xxyyzz00data bytes...aa     <-- data record (record n + 1)

       .

       .

       .
:xxyyzz00data bytes...aa     <-- data record (record n + i)
:00uuvv01bb                  <-- end-of-file record with
                                 transfer vector address
                                 (starting execution
                                 address for the downloaded
                                 firmware) (record n + i + 1)
```

where xx is the number of data bytes in the record
      (remember the number of ASCII hex digits making up
      the number of data bytes is 2 * xx -- each data
      byte consists of 2 hex digits)
    yy is the starting high byte address for the data
    zz is the starting low byte address for the data
    aa is the 2's complement checksum of the record
      computed as follow:

$$aa = -((xx + yy + zz + data\ bytes)\ .and.\ 0FFH)$$

    uu is the high byte of the transfer vector address
    vv is the low byte of the transfer vector address
    bb is the 2's complement checksum of the record
      computed as follow:

$$bb = - ((uu + vv + 01) .and. 0FFH)$$

Every record length will be less than or equal to 80 characters. Every character in the file except for the data in the comment record and the colon character used for starting the data and end-of-file records is a hexidecimal ASCII character. A pair of hex digits will form a byte of data.

7.2 The Download Record Format

The download record that is sent to the card by using the DLD order should be a binary image of the firmware. The control request in the download transaction specify the starting RAM address and the length of the download record. The control request should precede each DLD order. See chapter 4 for details on the the download transaction.

Note that the whole download firmware can be sent in one transaction if the host buffer is large enough to do this. The host must take the code file described above and convert it to the binary image required by the card.

7.3 Power-up Mux Firmware

The firmware available on power-up is just enough to allow the host to talk to the terminal connected to port 0. The full Mux firmware features are available only in the downloaded code. The reader is referred to chapter 4 and 5 for the following discussion to get additional details on the available feature.

No modem support will be provided in the EPROM firmware.

NOTE: THE EPROM FIRMWARE STOPS PROCESSING THE WRITE DATA/CONFIGURATAION/CONTROL TRANSACTION UPON ENCOUNTERING AN ERROR.

7.3.1 IDY transaction.

Available.

7.3.2 Read data/status transaction.

Only data from port 0 will be available with status 128, 143, and 144.

7.3.3 Write data/configuration/control transaction.

Only write data to port 0 is valid. All write data to other ports will generate an ARQ protocol error. Subfunctions 2, 3, 7, 130, 131, 138, 139, 140, 143, 144, and 145 will be available.

7.3.4 Download transaction.

All available.

```
+-------------------------------------------+-------------------+
|                                           |                   |
|   ASCII CHARACTERS & BINARY CODES         |   APPENDIX  A     |
|                                           |                   |
+-------------------------------------------+-------------------+
```

```
        0     1     2     3     4     5     6     7
      +-----+-----+-----+-----+-----+-----+-----+-----+
  0   | NUL | DLE | sp  | 0   | @   | P   | `   | p   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  1   | SOH | DC1 | !   | 1   | A   | Q   | a   | q   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  2   | STX | DC2 | "   | 2   | B   | R   | b   | r   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  3   | ETX | DC3 | #   | 3   | C   | S   | c   | s   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  4   | EOT | DC4 | $   | 4   | D   | T   | d   | t   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  5   | ENQ | NAK | %   | 5   | E   | U   | e   | u   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  6   | ACK | SYN | &   | 6   | F   | V   | f   | v   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  7   | BEL | ETB | '   | 7   | G   | W   | g   | w   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  8   | BS  | CAN | (   | 8   | H   | X   | h   | x   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  9   | HT  | EM  | )   | 9   | I   | Y   | i   | y   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  A   | LF  | SUB | *   | :   | J   | Z   | j   | z   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  B   | VT  | ESC | +   | ;   | K   | [   | k   | {   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  C   | FF  | FS  | ,   | <   | L   | \   | l   | |   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  D   | CR  | GS  | -   | =   | M   | ]   | m   | }   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  E   | SO  | RS  | .   | >   | N   | ^   | n   | ~   |
      +-----+-----+-----+-----+-----+-----+-----+-----+
  F   | SI  | US  | /   | ?   | O   | _   | o   | DEL |
      +-----+-----+-----+-----+-----+-----+-----+-----+
```

```
+---------------------------------------------------+-------------------+
|                                                   |                   |
|   EIA RS-232-C CONNECTOR PIN ASSIGNMENT           |   APPENDIX  B     |
|                                                   |                   |
+---------------------------------------------------+-------------------+
```

pin no.	description	CCITT V.24 Equivalent	EIA RS-449 Equivalent
1	AA Protective Ground	101	
2	BA Transmitted Data Send Data	103	SD
3	BB Received Data	104	RD
4	CA Request to Send	105	RS
5	CB Clear to Send Ready for Sending	106	CS
6	CC Data Set Ready Data Mode	107	DM
7	AB Signal Ground (Common Return)	102	SG
8	CF Received Line Signal Detector Data Channel Received Line Signal Detector Receiver Ready	109	RR
9	(Data Set Testing)		
10	(Data Set Testing)		
11	Unassigned		
12	SCF Secondary Received Line Signal Detector Backward Channel Received Line Signal Detector Secondary Receiver Ready	122	SRR
13	SCB Secondary Clear to Send Backward Channel Ready	121	SCS
14	SBA Secondary Transmitted Data		

		Transmitted Backward Channel Data	118	
		Secondary Send Data		SSD
		Not supported by firmware		
15	DB	Transmission Signal Element Timing (DCE Source)	114	
		Send Timing		ST
		Not available on this card		
16	SBB	Secondary Received Data		SRD
		Received Backward Channel Data	119	
		Not supported by firmware		
17	DD	Receiver Signal Element Timing (DCE Source)	115	
		Receive Timing		RT
		Not available on this card		
18		Unassigned		
19	SCA	Secondary Request to Send		SRS
		Transmit Backward Channel Line Signal	120	
20	CD	Data Terminal Ready	108.2	
		Terminal Ready		TR
21	CG	Signal Quality Detector		
		Data Signal Quality Detector	110	
		Signal Quality		SQ
		Not available on this card		
22	CE	Ring Indicator		
		Calling Indicator	125	
		Incoming Call		IC
23	CH/CI	Data Signal Rate Selector (DTE/DCE Source)	111/112	
		Signaling Rate Selector		SR
		Signaling Rate Indicator		SI
24	DA	Transmit Signal Element Timing (DTE Source)	113	
		Terminal Timing		TT

		Not available on this card		
25		Not defined		

The following is a historical note on the EIA RS-232-C symbol name. The original pin definition was defined by the pin assignment which consists of a matrix of 3 rows and 6 columns giving a total of 18 pins, see below.

```
     A  B  C  D  E  F
A .  .  .  .  .  .
B .  .  .  .  .  .
C .  .  .  .  .  .
```

Pin AA was used for protective ground, pin AB was for signal ground, and so on. The pin assignment was carried over to the current EIA RS-232-C symbol definition.

```
+----------------------------------------------+-------------------+
|                                              |                   |
|   EPROM LOG CHANNEL SUPPORT                  |   APPENDIX  C     |
|                                              |                   |
+----------------------------------------------+-------------------+
```

The EPROM firmware contains code for both the umux protocol as described in this document and the log channel protocol as described in the HP-CIO 27128 ASI ERS. The card firmware automatically comes up assuming the umux backplane protocol. The log channel protocol is invoked when the host or channel program issue the read transparent status (RTS) or the connect logical channel (CLC) order.

Once the EPROM firmware has detected either the RTS or the CLC order, the card will switch from the umux backplane protocol to the log channel backplane protocol. After switching to the log channel backplane protocol, it will remain using this protocol until the card is reset or until the downloaded firmware starts execution.

The firmware available for the log channel protocol is a subset of the ASI firmware. Enough firmware is available to allow the host to talk to the terminal connected to port 0 with no modem support. Any ASI feature which requires a timer are not available. Also, the force "0" or "1" for parity is unavailable.

Note that the blocking factor for the block read and write device data is 200 bytes.

For the remaining discussion in this appendix, the reader is referred to the ASI ERS.

C.1   Read Device Data Request

None of the subfunction feastures described in the ASI ERS will be available.

C.2   Write Device Data Request

Only the "automatic output separators appendage" option is available.

C.3   Control Device Request

None of the subfunctions defined are available.

C.4   Read Card Information

Subfunctions 0 and 254 will be available.

C.5   Write Card Configuration

Subfunctions 0, 1, 5, 9, 10, 11, 12, 13, 21, 32, and 33 will be available with NO ERROR CHECKING.

NOTE: The transmission mode is set to full duplex hardwired and cannot be changed even when subfunction 5 is used.

NOTE: The single text terminator will become effective immediately upon completion of the request. This means that if there is an active fronplane record, it will not wait for the record to be terminated before enabling the new single text terminators.

C.6   Control Card Request

Subfunctions 0, 1, 2, 3, 4, and 5 are available.

C.7   Default Card Configuration

The log channel portion of the firmware will have the following default values.

   o enable software handshake: no

   o edit mode: disabled

   o echoing: disabled

   o terminate on prompt sequence: not available

   o terminate on double text terminator: not available

o terminate on single text terminator: disabled

o end on count: disabled

o end on count length: 200

o alert 1 read mode: disabled

o modem lines SR, TR, and RS are not available

o transmission mode: full duplex hardwired only

o backspace character: BS

o line delete character: DEL

o strip FF hex: not available

o strip DEL: not available

o strip nulls: not available

o strip text terminators: enabled

o type of backspace notification: backspace echo

o echo CR-LF for a specified single text terminator: enabled

o send message after ENQ timeout: not available

o echo single text terminator: disabled

o host controlled X-ON/X-OFF handshake: disabled

o device controlled X-ON/X-OFF handshake: disabled

o host controlled ENQ/ACK handshake: disabled

o device controlled ENQ/ACK handshake: disabled

o baud rate: 9600

o character length: 8 bits/char

o # of stop bits: 1

o parity: none

o ignore parity errors: no

o discard characters with errors: no

o modem connection timer: not available

o no activity timer: not available

o lost receiver ready timer: not available

o gap timer: not available

o host ENQ/ACK timer: not available

o break generation: not available

o host interrupt mask: all cleared to disable interrupts

o host controlled X-ON character: DC1

o host controlled X-OFF character: DC3

o device controlled X-ON character: DC1

o device controlled X-OFF character: DC3

o single text terminator character for echoing CR-LF: CR

o host controlled ENQ character: ENQ

o host controlled ACK character: ACK

o device controlled ENQ character: ENQ

o device controlled ACK character: ACK

o host controlled ENQ/ACK pacing counter: 80

o number of output separator: 2

o output separator characters: CR-LF

o double text terminator: not available

o number of prompt sequence character: not available

o prompt sequence character: not available

o signal character detection: disabled

o do not terminate the text record on error: disabled

o "quoting" character mode: disabled

o conditional output separators appendage: disabled

o insert null in receive buffer for break: no

o repalce character for bad incoming character: DEL

o "quoting" character: \

o record separator to send output separators: LF

o signal character 1: OFF hex

o signal character 2: OFF hex

o signal character 3: OFF hex

o signal character 4: OFF hex

o quotable single text terminator char: EOT

o number of single text terminator: 1

o single text terminator: 1

o single text terminator character: CR

o send status 128, the remaining transmission count, on
  every read data/status transaction

Table of Contents