Space Science
UW SSEC Publication
No. 85.06.M1

MIDDS
HOST TO TERMINAL-TERMINAL TO HOST
SYSTEM PROTOCOL DESCRIPTION

26 June 1985

Rev. level: 1

4/MCIZ02/05

TABLE OF CONTENTS

MCIDAS HOST TO TERMINAL-TERMINAL TO HOST SYSTEM PROTOCOL DESCRIPTION
20 July 1983

OUTLINE

VI. Routing Codes - Terminal to Host

A. Negative Acknowledge (NAK)
B. Reserved
C. 40H, 41H Asymmetric One Bit Protocol IDS
D. 81H Keyboard Characters
E. 8AH ID Response
F. 91H TV Image State
G. 92H Graphic Image State
H. 93H Primary Cursor State
I. 96H Multi-Video Channel State
J. 98H Buffer Available Status
K. 99H Idle Traffic
L. 9AH Raw Joystick Data
M. 9BH Data Tablet Data
N. 9EH Data Tablet Pen State
O. 9FH Windco Response
P. A0H Stereo Cursor State

I. INTRODUCTION

All communications between the host CPU and the MIDDS terminal CPU are accomplished via message packets of a defined structure called a message protocol. This is a description of that protocol. Not all hardware configurations employ all aspects of this definition.

II. GENERAL CHARACTER DEFINITIONS

Communications between the host CPU and the MIDDS terminal are half duplex, byte oriented, binary synchronous, and employ a subset of the IBM bisync protocol. The following are the general classes of bytes:

1) All control characters are in the range of 00H to 3FH (H designates hexadecimal numeric notation)

2) All logical device routing characters are in the range of 40H to FFH

3) All data characters are in the range of 40H to FFH

A. Control Character Definitions

CHAR	NAME	DEFINITION
02H	STX	Indicates the next byte is beginning of message data
03H	ETX	End of message, checksum follows.
1DH	PENUP	Pen lift control for graphics message packets (routing codes 8BH & 8FH)
1EH	IRS	Internal record separator, defines end of data for current logical device
26H	ETB	End of transmitted block, defines end of message, checksum follows
2DH	ENQ	Used during initial loggon for IBM hardware environments
30H	MODSW	Mode switch control for graphics routing code 90H
31H	PENUP	Pen lift control for graphics mode switch packets (routing code 90H)
32H	SYN	Sync pattern used by hardware to frame message

37H	EOT	End of transmission, sent in response to a WACK (6BH)
3DH	NAK	Indicates checksum error, retransmission is requested

B. Control Character Exceptions (AKO & AK1)

AKO is a 2 byte response code.

CHAR	NAME	DESCRIPTION
10H	AKO	2 byte response to ENQ from IBM hardware environments
70H	AKO	Used during initial loggon, sent by IBM hosts in some bisync modes, terminal response is EOT (37H).

Example:

Host Sends;	SYN,ENQ	(i.e. 32H,2DH)
Terminal Responds;	SYN,AKO	(i.e. 32H,10H,70H)

AK1 is a 2 byte response code.

CHAR	NAME	DESCRIPTION
10H	AK1	2 byte response sent by IBM hosts in some bisync modes,
61H	AK1	terminal response is EOT (37H).
10H	WACK	Wait acknowledge, sent by IBM hosts.
6BH	WACK	Indicates wait and last message is acknowledged. Terminal responds with EOT (37H)

III. COMMUNICATION MODES

There are several modes of communication supported by the host software. For MIDDS only the transactional mode pertains.

During transactional mode, the terminal responds to the host message at a variable rate that depends upon the nature of the current system activity.

Transactional mode optionally operates under a one-bit asymmetric bisync protocol. Under the protocol, messages between the host and the terminal use the first routing code for message identification. The terminal and the host each switch between two states. Each state expects a message to include a predetermined id as the first routing code. If the correct id is detected, a new message is sent by the receiver, an incorrect id indicates a communication error and the receiver resends the last message.

Char	Name	Definition
40H	IDØ	One bit protocol id routing code
41H	ID1	

The protocol operates under the following rules:

Terminal state	Message Received		
	ID=Ø	ID=1	CS ERROR
state Ø expect ID=Ø	accept rcv buf send new xmt buf with ID=Ø go to state 1	discard rcv buf resend old xmt buf with ID=1 go to state Ø	discard rcv buf resend old xmt buf with ID=1 go to state Ø
state 1 expect ID=1	accept rcv buf resend old xmt buf with ID=Ø go to state 1	accept rcf buf send new xmt buf with ID=1 go to state Ø	discard rcv buf resend old xmt buf with ID=Ø go to state 1

Host state	Message Received		
	ID=Ø	ID=1	CS ERROR
state Ø expect ID=Ø	accept rcv buf send new xmt buf with ID=1 go to state 1	discard rcv buf resend old xmt buf with ID=Ø go to state Ø	discard rcv buf resend old xmt buf with ID=Ø go to state Ø
state 1 expect ID=1	accept rcv buf resend old xmt buf with ID=1 go to state 1	accept rcf buf send new xmt buf with ID=Ø go to state Ø	discard rcv buf resend old xmt buf with ID=1 go to state 1

Example:

```
host      SYN  SYN  STX  IDØ  IRS  <RT1> <DATA>  IRS  ...  ETX  P1  P2
terminal  SYN  SYN  STX  IDØ  IRS  ...   ETX  P1  P2

host      SYN  SYN  STX  ID1  IRS  ...   ETX  P1  P2
terminal  SYN  SYN  STX  ID1  IRS  ...   ETX  P1  P2
```

IV. GENERAL MESSAGE FORM - HOST TO TERMINAL AND TERMINAL TO HOST

The general form of a message between the host and the terminal is

SYN SYN STX <RT1> <DATA> IRS <RT2> <DATA> IRS ...
... <RTN> <DATA> ETX P1 P2

where <RTX> is a routing code to or from a logical device and <DATA> is zero or more bytes routed to or from that device. The message body from STX to ETX will always be ≤ 768 characters. P1 and P2 are a 16 bit checksum (CRC-16 inverse) for the message.
NOTE: All routing codes (host to terminal) send data to logical devices and not physical devices, with no assumptions about hardware implementation within a terminal.

V. ROUTING CODES - HOST TO TERMINAL

All routing codes are in the range of 40H to FFH. All routing codes are preceded by an STX or an IRS. All routing codes are followed by an IRS or an ETX. Terminals that do not have a particular routing code implemented, will treat that particular code as a no-op.

A. Routing Codes 40H and 41H are Used for Asymmetric One Bit Protocol

See communications modes.

B. Routing Codes 42H to 80H are Reserved for Future Expansion

C. Routing Code 81H CRT Data

This routing code sends ASCII text to the terminal console. All data must be ASCII bytes OR'd with 80H. Data may be from 1 to 82 bytes long.

Example: This is a CRT message consisting of a line feed, A, carriage return.

02H(1EH) 81H 8AH C1H 1EH(03H)

D. Routing Code 82H Terminal Printer

All data is ASCII OR'd with 80H. First data character is form control according to FORTRAN conventions, i.e.,

CHAR	FUNCTION
BLANK	single space before printing
0	double space
1	top of form
+	no space before printing

The form control character may be followed by from 0 to 132 characters which define one line for the printer.

Example: This is a message that top of forms the line printer and prints an A followed by a carriage return.

02H(1EH) 82H B1H C1H 1EH(03H)

E. Routing Code 83H Single Graphics Device (Delta Mode)

This routing code sends data to a terminal with single frame graphics. This function allows erasing the graphics frame or writing a line segment of 1 or more points. The graphics frame origin is at the upper left corner of the image space.

1. Graphics Erase Command

The empty message causes the graphics frame to be erased.

Example: 02H(1EH) 83H 1EH(03H)

2. Graphics Line Segment Command

Four header bytes are required following the 83H (routing code):

Byte 1	1	1	L4	L3	L2	L1	L0	Y8
Byte 2	1	1	Y7	Y6	Y5	Y4	Y3	Y2
Byte 3	1	1	Y1	Y0	X9	X8	X7	X6
Byte 4	1	1	X5	X4	X3	X2	X1	X0

where:

L4-L0 define the graphics value to write (0-1FH)
Y8-Y0 define the TV line to set the pen down (0-499)
X9-X0 define the TV pixel to set the pen down (0-639)

In all cases *0 is the least significant bit. If a single point is to be written, no additional bytes are required. If a line segment is to be drawn, additional data must be packed (using 2 bytes) as five 3-bit values. These 3-bit values define relative delta moves, each in one of the either cardinal directions.

3-bit Data Value	**Relative Pen Motion**
0	X to X+1 ; Y to Y
1	X to X-1 ; Y to Y
2	X to X ; Y to Y+1
3	X to X ; Y to Y-1
4	X to X+1 ; Y to Y+1
5	X to X-1 ; Y to Y+1
6	X to X+1 ; Y to Y-1
7	X to X-1 ; Y to Y-1

Furthermore, these 3-bit values must be packed as 5 data values into two bytes according to the following format:

If the two bytes are made of these bits,

Byte 1 = A1 A2 B1 B2 B3 C1 C2 C3
Byte 2 = A3 A4 D1 D2 D3 E1 E2 E3

then,

A1	A2	A3	A4	3-bit Delta Code
0	1	0	1	0
0	1	1	0	1
0	1	1	1	2
1	0	0	1	3
1	0	1	0	4
1	0	1	1	5
1	1	0	1	6
1	1	1	0	7
1	1	1	1	Unused

Bit groups B, C, D, and E are themselves the 3-bit delta codes. If the number of delta codes is not a multiple of five, a five 3-bit multiple must be padded out by making remaining deltas go back upon themselves. Example: The following device data draws a 4 pixel horizontal line at TV line 100 and TV pixel 200 at graphics level 3.

02H(1EH) 83H C6H D0H C3H C8H 40H 48H 1EH(03H)

F. Routing Code 84H - TV Image Line Load 3-Bit Mode

All data is packed as 3-bit codes with five 3-bit codes per two bytes according to the following format:

If the two bytes are made of these bits,

Byte 1 = A1 A2 B1 B2 B3 C1 C2 C3
Byte 2 = A3 A4 D1 D2 D3 E1 E2 E3

then,

A1	A2	A3	A4	3-bit code
0	1	0	1	0
0	1	1	0	1
0	1	1	1	2
1	0	0	1	3
1	0	1	0	4
1	0	1	1	5
1	1	0	1	6
1	1	1	0	7
1	1	1	1	Unused

Bit groups B, C, D, and E are themselves the 3-bit codes. The first 3-bit code is a blowup factor to be applied to both lines and pixels. One means no repeat.

The next three 3-bit codes are concatenated to form a 9-bit group (MSB's 1st) that define the TV starting line number.

All other 3-bit codes are an encoding of 6-bit data along the horizontal line. Image pixel data is assumed to be six bits. The starting pixel on each line is assumed to be zero. Each 3-bit code defines how to derive the next pixel from the last one as follows:

CODE	EFFECT
0	Next 6-bit pixel same as last
1	Next 2 same as last
2	Next 3 same as last
3	Last+1
4	Last-1
5	Last+2
6	Last-2
7	Escape, next two 3-bit codes form new pixel

There are 640 pixels assumed in a horizontal line. If fewer are sent, the remainder are set to 0. If more are sent, they are ignored. (More may be sent to comply with the 3-bit packing rule.)

The worst case for this encoding scheme (constant escapes) will produce 768 bytes of data for one line, actual image data will result in sending a much smaller number of bytes, thus increasing the effective transmission rate of the communication line.

G. Routing Code 85H - 6-bit Enhancement Table Load

The enhancement table is considered to be a 64 cell memory, each cell consisting of 15-bit data values (5 blue, 5 green, 5 red) bit inverted. Each 6-bit video pixel is used as an index into these cells. The three color output of a cel is used to define the color of each pixel. Each byte sent must have one or both of the two MSB's set to 1's. An enhancement table load message consists of one byte containing the starting cell number, and one or more sets of three bytes each (1 blue, 1 green, 1 red) defining 15-bit data to be written into the first cell specified and increasingly higher order cells.

H. Reserved

I. Routing Code 87H - TV Image Line Load 6-bit Mode

All data is in 6 bit mode (six LSB's are data bits and two MSB's are set to ones) bytes 1 and 2 are a 9-bit TV line number where:

Byte 1 = 1 1 * * * Y8 Y7 Y6
Byte 2 = 1 1 Y5 Y4 Y3 Y2 Y1 Y0

* = don't care
Y = TV line number to load

Byte 3 to N where N < 642 are 6-bit pixel values along the TV line. If N < 642, the remaining pixels on the TV line are set to 0.

J. Routing Code 88H - TV Frame to Load

This is a 2 byte message in 6-bit mode where 12 bits (right justified) form a 12 bit TV frame to send data to/from routing codes 84H or 87H.

K. Routing Code 89H - 12-bit Enhancement Table Load

The enhancement table is an array of 64x64 cells, each cell consisting of 15 bit data values (5 blue, 5 green, 5 red) bit inverted. The table is used to display two different image frames simultaneously. The intensity of a pixel in frame A (six bits) indexes into one dimension of the table, the intensity of the corresponding pixel in frame B indexes into the other dimension. The enhancement cell selected defines the color of the pixel pair.

Row 0 and column 0 are reserved for use with the six bit enhancement tables, consequently the software that loads the enhancement table remaps the table into an array of 63x63 cells. A message that specifies loading element (0,0), for example, is actually loading element (1,1) in the enhancement table memory.

An enhancement table load message consists of one byte containing the starting enhancement row number (6 LSB's), one byte containing the column number (6 LSB's), and one or more sets of three bytes each (1 blue, 1 green, 1 red) defining 15 bit data to be written into the first cell specified and increasingly higher column numbers. The five LSB's in each byte of the three color sets will contain the data. Enhancement data must be loaded one row at a time. Data that is written off the end of a row will alter the six bit enhancement tables. All data in the message must have one or both of the 2 MSB's set to ones.

L. **Routing Code 8AH - ID Request**

This is a request for terminal ID and image configuration sent by the host. There is no data in this message (ex. 02H, 8AH, 1EH[03H]).

The terminal will respond with ID, number of image frames, and number of graphic frames.

M. **Routing Code 8BH - Graphics Device (Delta Mode)**

This routing code sends data to a terminal with multiple frame graphics. This function allows erasing the graphics frame or writing a line segment of 1 or more points.

The graphics frame origin is at the upper left corner of the image space.

1. **Graphics Erase Command**

The message causes the graphics frame specified to be erased. The message contains only one byte (MSB=1) containing the graphics frame to erase.

Example: 02H(1EH) 8BH 81H 1EH(03H)

The above message will cause graphics frame 1 to be erased.

2. **Graphics Line Segment Command**

Five header bytes are required following the 8BH (routing code):

Byte 1	1	F6	F5	F4	F3	F2	F1	F0
Byte 2	1	1	L4	L3	L2	L1	L0	Y8
Byte 3	1	1	Y7	Y6	Y5	Y4	Y3	Y2
Byte 4	1	1	Y1	Y0	X9	X8	X7	X6
Byte 5	1	1	X5	X4	X3	X2	X1	X0

where:

F6-F0 define graphics frame to write
L4-L0 define the graphics value to write (0-1FH)
Y8-Y0 define the TV line to set the pen down (0-499)
X9-X0 define the TV pixel to set the pen down (0-639)

In all cases, *0 is the least significant bit.

If a single point is to be written, no additional bytes are required. If a line segment is to be drawn, additional data must be packed (using 2 bytes) as five 3-bit values. These 3-bit values define relative delta moves, each in one of the eight cardinal directions.

3-bit Data Value	Relative Pen Motion
0	X to X+1 ; Y to Y
1	X to X-1 ; Y to Y
2	X to X ; Y to Y+1
3	X to X ; Y to Y-1
4	X to X+1 ; Y to Y+1
5	X to X-1 ; Y to Y+1
6	X to X+1 ; Y to Y-1
7	X to X-1 ; Y to Y-1

Furthermore, these 3-bit values must be packed as five data values into two bytes according to the following format;

If the two bytes are made of these bits,

Byte 1 = A1 A2 B1 B2 B3 C1 C2 C3
Byte 2 = A3 A4 D1 D2 D3 E1 E2 E3

then,

A1	A2	A3	A4	3-bit Delta Code
0	1	0	1	0
0	1	1	0	1
0	1	1	1	2
1	0	0	1	3
1	0	1	0	4
1	0	1	1	5
1	1	0	1	6
1	1	1	0	7
1	1	1	1	Unused

Bit groups B, C, D, and E are themselves the 3-bit delta codes. If the number of delta codes is not a multiple of five, a five 3-bit multiple must be padded out by making remaining deltas go back upon themselves. Upon completion of 3-bit data for 1 line segment (padded to a two byte group), a pen lift control character (1DH), followed by all data normally sent following the 8BH routing code, may be sent thus allowing multiple line segments in one message packet.

N. Reserved

O. Reserved

P. Routing Code 8EH - Graphics Fill Mode

This message causes a specified graphics frame to have a specified number of pixels along a specified horizontal line starting at a specified pixel to be written to a desired graphics level. Five header bytes are required following the 8EH (routing code):

Byte 1	1	F6	F5	F4	F3	F2	F1	F0
Byte 2	1	1	L4	L3	L2	L1	L0	Y8
Byte 3	1	1	Y7	Y6	Y5	Y4	Y3	Y2
Byte 4	1	1	Y1	Y0	X9	X8	X7	X6
Byte 4	1	1	X5	X4	X3	X2	X1	X0

where F6-F0 define graphics frame to write
L4-L0 define the graphics value to write (0-1FH)
Y8-Y0 define the TV line to set the pen down (0-499)
X9-X0 define the TV pixel to set the pen down (0-639)

In all cases, *0 is the least significant bit.

Two bytes (in 6-bit mode) containing the right justified number of pixels to set to the desired graphics level must follow. A pen lift control (1DH) may follow if more lines are desired. If so, five header bytes and two pixel count bytes must follow.

Q. <u>Routing Code 8FH - Graphics End Point Mode</u>

This routing code causes graphics at the level specified to be written on the graphics frame desired given end points to line segments.

Five header bytes are required following the 8FH (routing code):

Byte 1	1	F6	F5	F4	F3	F2	F1	F0
Byte 2	1	1	L4	L3	L2	L1	L0	Y8
Byte 3	1	1	Y7	Y6	Y5	Y4	Y3	Y2
Byte 4	1	1	Y1	Y0	X9	X8	X7	X6
Byte 4	1	1	X5	X4	X3	X2	X1	X0

where F6-F0 define graphics frame to write
L4-L0 define the graphics value to write (0-1FH)
Y8-Y0 define the TV line to set the pen down (0-499)
X9-X0 define the TV pixel to set the pen down (0-639)

In all cases, *0 is the least significant bit.

Four bytes specifying point 2 of the end points are required where:

Byte 6	1	*	*	*	*	*	Y8	Y7
Byte 7	1	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Byte 8	1	*	*	*	*	X9	X8	X7
Byte 9	1	X6	X5	X4	X3	X2	X1	X0

where * means don't care
Y8-Y0 define the TV line number of the second end point
X9-X0 define the TV pixel number of the second end point

In all cases, *0 is the least significant bit.

If an additional segment connected to the last end point in desired, another four byte set as defined above is required. If the pen is to be lifted, a penup (1DH) must be sent followed by a 9 byte set under the starting format of byte 1 above.

R. **Routing Code 90H - Graphics in both Delta and End Point Mode**

This message packet allows data to be sent to any graphics frame in both end point and delta modes. This allows the data bits sent to the terminal to be minimized. A pixel repeat function (wide lines) dotted lines, and a pen lift function are also employed.

Eight header bytes are required following the 90H (routing code)

Byte 1	1	F6	F5	F4	F3	F2	F1	F0
Byte 2	1	L4	L3	L2	L1	L0	D5	D4
Byte 3	1	D3	D2	D1	D0	G4	G3	G2
Byte 4	1	G1	G0	S5	S4	S3	S2	S1
Byte 5	1	S0	E4	E3	E2	E1	E0	M
Byte 6	1	Y8	Y7	Y6	Y5	Y4	Y3	Y2
Byte 7	1	Y1	Y0	X9	X8	X7	X6	X5
Byte 8	1	X4	X3	X2	X1	X0	*	*

where

F6-F0	Define graphics frame to write
L4-L0	Define the graphics value to write (0-1FH)
D5-D0	Define length of dashed line segments in pixels. If D5-D0=0, solid lines are assumed and G4-G0, S5-S0 are not used.
G4-G0	Define the graphics value to write gaps between dashed line segments, i.e., if G4-G0 = 0, gaps are blank, otherwise gaps are at value G4-G0.
S5-S0	Define length of gap segments in pixels. If S5-S0=0, solid lines are assumed and G4-G0, D5-D0 are not used.
E4-E0	Define pixel repeat factor (0 means 1 pixel wide, 1 means 2 pixels wide, etc.
M	Define mode select (0 = delta, 1 = end point)
Y8-Y0	Define the TV line to set the pen down (0-499)
X9-X0	Define the TV pixel to set the pen down (0-639)
*	Means don't care

In all cases *0 is the least significant bit. If mode (M) = 1 (End Point Mode), the following data are expected:

Three bytes specifying point 2 of the end points are required where:

byte 9	1	Y8	Y7	Y6	Y5	Y4	Y3	Y2
byte 10	1	Y1	Y0	X9	X8	X7	X6	X5
byte 11	1	X4	X3	X2	X1	X0	*	*

where: * means don't care

Y8-Y0 define the TV line number of the second end point

X9-X0 define the TV pixel number of the second end point

In all cases, * is the least significant bit. If an additional segment connected to the last end point is desired, another 3 byte set as defined above is required. If mode (M) = 0 (Delta mode), the following data is expected: data must be packed (using 2 bytes) as five 3-bit values. These 3-bit values define relative delta moves, each in one of the eight cardinal directions.

3-bit Data Value	Relative Pen Motion
0	X to X+1 ; Y to Y
1	X to X-1 ; Y to Y
2	X to X ; Y to Y+1
3	X to X ; Y to Y-1
4	X to X+1 ; Y to Y+1
5	X to X-1 ; Y to Y+1
6	X to X+1 ; Y to Y-1
7	X to X-1 ; Y to Y-1

Furthermore, these 3-bit values must be packed as 5 data values into 2 bytes according to the following format:

If the 2 bytes are made of these bits,

Byte 1	-	A1	A2	B1	B2	B3	C1	C2	C3
Byte 2	-	A3	A4	D1	D2	D3	E1	E2	E3

then,

A1	A2	A3	A4	3-bit Delta Code
0	1	0	1	0
0	1	1	0	1
0	1	1	1	2
1	0	0	1	3
1	0	1	0	4
1	0	1	1	5
1	1	0	1	6
1	1	1	0	7
1	1	1	1	Unused

Bit groups B, C, D, and E are themselves the 3-bit delta codes if the number of delta codes is not a multiple of 5, a five 3-bit multiple must be padded out by making remaining deltas go back upon themselves if the pen is to be lifted, a penup (31H) must be sent followed by a 3 byte set under the starting format of byte 6.

If a mode switch is desired (end point to delta or delta to end point), a mode switch control (30H) must be sent followed by data in the opposite mode format.

S. **Routing Code 91H - TV Image Control**

This message is always 7 bytes long. This message allows the host CPU to define the state of TV images at the terminal. The following defines the format of the 7 data bytes where "BX" stands for bit "X", B7 is the MSB, and B0 is the LSB:

Byte 1 B7=1, B6=1, B5-B0 are 6 MSB's for frame to show
Byte 2 B7=1, B6=1, B5-B0 are 6 LSB's for frame to show
Byte 3 B7=1, B6=1, B5-B0 are 6 MSB's of upper bound frame in a movie loop
Byte 4 B7=1, B6=1, B5-B0 are 6 LSB's of upper bound frame in a movie loop
Byte 5 Same as byte 3 but for lower bounds
Byte 6 Same as byte 4 but for lower bounds
Byte 7 B7=1, B6=1, B5-B3=3 bit reserved group,
B2=Display key
B1=Movie Key
B0=Move Enable Key

If display key (B2) = 1, TV screen shows image defined by bytes 1 & 2
If display key (B2) = 0, TV image is not visible (black screen)
If movie key (B1) = 0, no movie looping
If movie key (B1) = 1, movie looping enabled
If move enable key = 0, frame cannot switch regardless of movie key state
If move enable key = 1, frame can be changed

T. **Routing Code 92H - Graphic Image Control**

This message is always 7 bytes long. This message allows the host CPU to define the state of graphic images at the terminal. The following defines the format of the 7 data bytes where "BX" stands for bit "X", B7 is the MSB, and B0 is the LSB:

Byte 1 B7=1, B6=1, B5-B0 are 6 MSB's for graphic frame to show
Byte 2 B7=1, B6=1, B5-B0 are 6 LSB's for graphic frame to show
Byte 3 B7=1, B6=1, B5-B0 are 6 MSB's of upper bound graphic frame in a movie loop
Byte 4 B7=1, B6=1, B5-B0 are 6 LSB's of upper bound graphic frame in a movie loop
Byte 5 Same as byte 3 but for lower bounds
Byte 6 Same as byte 4 but for lower bounds
Byte 7 B7=1, B6=1, B5-B3=3 bit reserved group,
B2=Display key
B1=Movie Key
B0=Move Enable Key

If display key (B2) = 1, TV screen shows graphic image defined by bytes 1 & 2
If display key (B2) = 0, graphic image is not visible
If movie key (B1) = 0, no movie looping of graphics
If movie key (B1) = 1, movie looping enabled
If move enable key = 0, graphic frame cannot switch regardless of movie key state
If move enable key = 1, graphic frame can be changed

U. Routing Code 93H - Primary Cursor State

This message allows the host to control the cursor. This message always contains 10 bytes where "BX" stands for bit "X", B7 is the MSB, and B0 is the LSB:

Byte 1 B7=1, B6=1, B5-B0 are 6 MSB's of 1/2 cursor line size
Byte 2 B7=1, B6=1, B5-B0 are 6 LSB's of 1/2 cursor line size
Byte 3 B7=1, B6=1, B5-B0 are 6 MSB's of 1/2 cursor element size
Byte 4 B7=1, B6=1, B5-B0 are 6 LSB's of 1/2 cursor element size
Byte 5 B7=1, B6=1, B5-B0 are 6 MSB's of cursor line position
Byte 6 B7=1, B6=1, B5-B0 are 6 LSB's of cursor line position
Byte 7 B7=1, B6=1, B5-B0 are 6 MSB's of cursor element position
Byte 8 B7=1, B6=1, B5-B0 are 6 LSB's of cursor element position
Byte 9 B7=1, B6-B0 are 7 bit cursor type (only 5 are currently implemented)

where:
Type 1 = Box
Type 2 = Cross hair
Type 3 = Box with cross hair
Type 4 = Solid rectangle
Type 5 = Star wars

Byte 10 B7=1, B6=1, B5-B3 are cursor color (1 bit each G, G, B), B2-B0 are a number implying cursor control where;

0 cursor unlinked (size and position according to bytes 1 to 8)
1 position with vernier (position determined via joysticks rather than bytes 5 to 8
2 size, no position
3 size and position (size and position determined by joysticks rather than bytes 1 to 8)
4 velocity, no position
5 velocity (position determined by joysticks according to velocity cursor algorithm)
6 stereo cursor, no joystick control
7 stereo cursor, joystick control

V. Routing Code 94H - Movie Loop Dwells

This message allows the host to choose dwell rates for frames in a movie loop. A dwell unit is considered to be 1/30 second. A maximum of 12 dwell values may be sent. Dwell value 1 pertains to image 1, dwell 2 to image 2, etc. If there are more images in a movie than dwells, the 12th dwell value is used repeatedly to the end of the movie loop. This message is optional and need be sent only if dwells different from the default are desired. Once dwell values are sent to the terminal, they will be retained for any subsequent movie loops until a new dwell is received by the terminal. The form of a dwell message is:

Byte 1 B7=1, B6-B0 define dwell for image 1
Byte 2 B7=1, B6-B0 define dwell for image 2 (optional)
Byte n Same as above where n must be ≤ 12 (optional)

W. Routing Code 95H - Graphics-Cursor Color Pallet

This message allows changing graphic and/or cursor colors in terminals equipped with the colorizing pallet hardware. The following is a description of the pallet lookup organization.

There are 32 graphic lookup cells (cells 0 to 31).
There are 32 cursor lookup cells (cells 32 to 63).
The total lookup table is 64 cells long (128 bytes).

Cells are addressed during display time by concatenating five graphic bits and 1 cursor bit (the cursor bit addresses the upper half of the lookup table).

The form of a cell is two bytes long where:

Byte 1	G2	G1	G0	B4	B3	B2	B1	B0
Byte 2	VE	R4	R3	R2	R1	R0	G4	G3

B4 to B0 Define blue component output.
G4 to G0 Define green component output.
R4 to R0 Define red component output.

VE is video enable (if VE=0 contents of cell is shown, if VE=1 video from image is shown (if VE=1 all bits in the cell must be set to 1's)).

The format of a routing code 95H is as follows:

Bytes must come in 3 byte groups with a minimum of 1 group (for cell 0) and a maximum of 64 groups.

Byte 1	1	1	G2	G1	G0	B4	B3	B2
Byte 2	1	1	R1	R0	G4	G3	B1	B0
Byte 3	1	1	*	*	VE	R4	R3	R2

* are don't cares
G's, B's, R's, and VE are loaded to hardware as indicated above under lookup table description.

Note: The first 3 byte group data is loaded to cell 0, the second 3 byte group to cell 1, etc.

X. Routing Code 96H - Multi-video Channel Enable-Disable

This message contains one data byte used to enable dual channel video display thru the 12 bit colorizer lookup table for terminals equipped with that hardware.

If data byte = C1H, dual channel enabled
If data byte = C0H, dual channel disabled

Y. Routing Code 97H - Terminal State Request

This message contains one byte of data specifying a delay count. Reception of the routing code by the terminal causes the terminal to send a complete state of the terminal to the host via routing codes 91H, 92H, 93H, 96H, 98H, 9AH, and 9BH as defined below under message formats for terminal to host.

Byte 1 B7=1, B6=1, B5-B0 are delay count before state response is sent (1/60 sec)

Z. Routing Code 99H - Transactional Mode Switch

This message has no data following the routing code. If a terminal is in intensive operating mode when this routing code is received, the terminal will switch to transactional mode. If the terminal is in transactional mode, it will delay its response to the host by an additional 1/15 second (until a maximum delay of 2½ sec). This allows less host interruption during idle periods. In transactional mode, the host uses this as an idle and always answers immediately.

Z1. Routing Code 9CH - Auto Prompt

This message consists of up to 80 ASCII characters OR'd with 80H. These data will be preceded by a line feed, displayed on the CRT, and stored in the keyboard data input buffer as though the user had keyed them in. The user may then use the keyboard to edit these characters or append to the character string. The host is allowed to send only one such message until the response string (from keyboard data) is received. Line feeds (0AH) and carriage returns (0DH) are not allowed. All other ASCII characters are allowed.

Z2. Routing Code 9DH - Velocity Increments

This message allows the host to set the velocity increments for the velocity cursor algorithm. A maximum of 12 increments may be sent. If there are more images than increments, the 12th increment will be used repeatedly to the end of the movie loop. This message is optional and is sent only if increments different than the default are desired. Increments sent to the terminal are used until a new velocity increment message is received by the terminal.

The form of the velocity increment is:

Value 1 for image 1
Value 2 for image 2
.
.
Value n for image n ($n \leq 12$)

Each value is two bytes long in the following form:

Byte 1 B7=1, B6-B0 are 7 MSB's of an increment
Byte 2 B7=1, B0 is LSB of an increment

Z3. Routing Code 9FH - Windco Mode State

This message is used to allow cloud tracking cursor control on the terminal. The space bar invokes this function when in Windco mode. The first spacebar causes the terminal to collect state information, change color and size of cursor, and step image frame. Second spacebar sends the selected state information to the host and returns cursor to starting color and size (see 9FH in terminal to host). It also steps the image frame if the frame was at the end of the movie loop. Lag size is change in cursor size from first to second frame in procedure.

Byte 1 B7=1, B6=1, B5 is mode (1 yes, 0 no)
Byte 2 B7=1, B6=1, B5-B3 are first color bits, B2-B0 are second color bits
Byte 3 B7=1, B6=1, B5-B0 are 6 MSB's of 1/2 cursor line lag size
Byte 4 B7=1, B6=1, B5-B0 are 6 LSB's of 1/2 cursor line lag size
Byte 5 B7=1, B6=1, B5-B0 are 6 MSB's of 1/2 cursor element lag size
Byte 6 B7=1, B6=1, B5-B0 are 6 LSB's of 1/2 cursor element lag size

Z4. Routing Code A0H - Stereo Cursor State

This message allows the host to control the cursor when in stereo cursor mode (see routing code 93H). This message always contains 10 bytes where:

Byte 1 B7=1, B6=1, B5-B0 are 6 MSB's of 1/2 cursor line size
Byte 2 B7=1, B6=1, B5-B0 are 6 LSB's of 1/2 cursor line size
Byte 3 B7=1, B6=1, B5-B0 are 6 MSB's of 1/2 cursor element size
Byte 4 B7=1, B6=1, B5-B0 are 6 LSB's of 1/2 cursor element size
Byte 5 B7=1, B6=1, B5-B0 are 6 MSB's of cursor line position
Byte 6 B7=1, B6=1, B5-B0 are 6 LSB's of cursor line position
Byte 7 B7=1, B6=1, B5-B0 are 6 MSB's of cursor element position
Byte 8 B7=1, B6=1, B5-B0 are 6 LSB's of cursor element position
Byte 9 B7=1, B6-B0 are 7 bit cursor type (only 5 are currently implemented)

where: type 1 = box
type 2 = cross hair
type 3 = box with cross hair
type 4 = solid rectangle
type 5 = star wars

Byte 10 B7=1, B6=1, B5-B0 are cursor color (bit each R,G,B)

VI. ROUTING CODES - TERMINAL TO HOST

In intensive mode, there are only two forms of messages sent from the terminal to host NAK and terminal state. In transactional mode, several routing codes are employed.

A. Return Message - Negative Acknowledge (NAK)

This message is sent in response to receiving a message with a bad checksum, thus requesting retransmission. The form of the message is:

SYN SYN NAK

B. Reserved

C. Routing Codes 40H and 41H are Used for Asymmetric One Bit Protocol

Sees communication modes.

D. Routing Code 81H - Keyboard Characters

This routing code allows the terminal to send user keyboard characters to the host. Two forms of user keyins a distinguished, single letter keyins and character strings preceded by a line feed and ended with a carriage return. Whenever a user keyin is complete, the terminal will send a routing code 81H message with that keyin. In addition, routing codes 91H, 92H, 93H, 96H, 98H, 9AH, and 9BH will be sent in the same physical message, thus establishing the terminal state at the time of the user keyin.

The form of routing code 81H message is:

One or more ASCII characters in the user keyin are OR'd with 80H and follow the 81H (routing code).

E. Routing Code 8AH - ID Response

This message is a 4 byte data response to the host giving the terminal ID, number of image frames and number of graphics frames the terminal is equipped with.

The form of the data following the 8AH is:

Byte 1	7 bit terminal ID with MSB set = 1
Byte 2	7 MSB's of number of image frames in terminal with MSB set = 1
Byte 3	7 LSB's of number of image frames in terminal with MSB set = 1
Byte 4	7 bits of number of graphics frames in terminal with MSB set = 1

F. Routing Code 91H - TV Image State

This message is always 7 bytes long. This message allows the host CPU to determine the state of TV images at the terminal. The following defines the format of the 7 data bytes where 'BX' stands for bit 'X', B7 is the MSB, and B0 is the LSB:

Byte 1	B7=1, B6=1, B5-B0 are 6 MSB's for frame showing
Byte 2	B7=1, B6=1, B5-B0 are 6 LSB's for frame showing
Byte 3	B7=1, B6=1, B5-B0 are 6 MSB's of upper bound frame in a movie loop
Byte 4	B7=1, B6=1, B5-B0 are 6 LSB's of upper bound frame in a movie loop
Byte 5	Same as byte 3 but for lower bounds
Byte 6	Same as byte 4 but for lower bounds
Byte 7	B7=1, B6=1, B5-B3 = 3 bit reserved group, B2 = display key B1 = movie key B0 = move enable key

If display key (B2) = 1, TV screen shows image defined by bytes 1 and 2.
If display key (b2) = 0, TV image is not visible (black screen).

If movie key (B1) = 0, no movie looping.
If movie key (B1) = 1, movie looping enabled.

If move enable key = 0, frame cannot switch regardless of movie key state.
If move enable key = 1, frame can be changed.

G. Routing Code 92H - Graphic Image State

This message is always 7 bytes long. This message allows the host CPU to determine the state of graphic images at the terminal. The following defines the format of the 7 data bytes where 'BX' stands for bit 'X', B7 is the MSB, and B0 is the LSB:

Byte 1	B7=1, B6=1, B5-B0 are 6 MSB's for graphic frame showing
Byte 2	B7=1, B6=1, B5-B0 are 6 LSB's for graphic frame showing
Byte 3	B7=1, B6=1, B5-B0 are 6 MSB's of upper bound graphic frame in a movie loop
Byte 4	B7=1, B6=1, B5-B0 are 6 LSB's of upper bound graphic frame in a movie loop
Byte 5	Same as byte 3 but for lower bounds
Byte 6	Same as byte 4 but for lower bounds
Byte 7	B7=1, B6=1, B5-B3 = 3 bit reserved group, B2 = display key B1 = movie key B0 = move enable key

If display key (B2) = 1, TV screen shows graphic image defined by bytes 1 and 2.
If display key (b2) = 0, graphic image is not visible (black screen).

If movie key (B1) = 0, no movie looping of graphics
If movie key (B1) = 1, movie looping enabled.

If move enable key = 0, graphic frame cannot switch regardless of movie key state.
If move enable key = 1, graphic frame can be changed.

H. Routing Code 93H - Primary Cursor State

This message allows the host to determine the state of the cursor. This message always contains 10 bytes where "BX" stands for bit "X", B7 is the MSB, and B0 is the LSB:

Byte 1	B7=1, B6=1,	B5-B0 are 6 MSB's of 1/2 cursor line size
Byte 2	B7=1, B6=1,	B5-B0 are 6 LSB's of 1/2 cursor line size
Byte 3	B7=1, B6=1,	B5-B0 are 6 MSB's of 1/2 cursor element size
Byte 4	B7=1, B6=1,	B5-B0 are 6 LSB's of 1/2 cursor element size
Byte 5	B7=1, B6=1,	B5-B0 are 6 MSB's of cursor line position
Byte 6	B7=1, B6=1,	B5-B0 are 6 LSB's of cursor line position
Byte 7	B7=1, B6=1,	B5-B0 are 6 MSB's of cursor element position
Byte 8	B7=1, B6=1,	B5-B0 are 6 LSB's of cursor element position
Byte 9	B7=1,	B6-B0 are 7 bit cursor type (only 5 are currently implemented)

where:
- type 1 = box
- type 2 = cross hair
- type 3 = box with cross hair
- type 4 = solid rectangle
- type 5 = star wars

Byte 10 B7=1, B6=1, B5-B3 are cursor color (bit each R,G,B)
B2-B0 are a number implying cursor control

where:

0 cursor unlinked (size and position according to bytes 1 to 8)
1 position with vernier (position determined via joysticks rather than bytes 5 to 8)
2 size, no position
3 size and position (size and position determined by joysticks rather than bytes 1 to 8)
4 velocity, no position
5 velocity (position determined by joysticks according to velocity cursor algorithm)
6 stereo cursor, no joystick control
7 stereo cursor, joystick control

I. Routing Code 96H - Multi-video Channel State

This message contains one data byte used to inform the host of the state of the dual channel video display for terminals equipped with that hardware.

If data byte = C1H, dual channel enabled
If data byte = C0H, dual channel disabled

J. Routing Code 98H - Buffer Available Status

This message allows the terminal to inform the host if buffers are available for data to be sent to CRT or line printer. This message is included in every physical message sent to the host while in transactional mode. This message is two bytes long:

Byte 1	CRT buffer status (C0H = No buffers, C1H = buffers available)
Byte 2	Line printer status (C0H = No buffers, C1H = buffers available)

This packet is the idle message.

K. Routing Code 99H - Idle Traffic

This message is sent to keep communications open whenever there is no information to exchange. This message has no data following the routing code.

This packet is currently unused.

L. Routing Code 9AH - Raw Joystick Data

This message sends the host the raw joystick data at the terminal. This message is 8 bytes long where:

Byte 1	Joystick 1 vertical 6 MSB's OR'd with C0H
Byte 2	Joystick 1 vertical 6 LSB's OR'd with C0H
Byte 3	Joystick 1 horizontal 6 MSB's OR'd with C0H
Byte 4	Joystick 1 horizontal 6 LSB's OR'd with C0H
Byte 5-8	Same as joystick 1 but for joystick 2

M. Routing Code 9BH - Data Tablet Data

The following data tablet data are sent if the terminal is implemented with a data tablet. 'BX' stands for 'X', B7 is the MSB, B0 is the LSB.

Byte 1 B7=1, B6 = Data tablet pen state, 0 means up, 1 means down
B5-B0 are 6 MSB's of current Y position

Byte 2	B7=1, B6	= Data tablet pen proximity state, 1 means pen is near tablet, 0 means pen is not near tablet. (If B6=0, all other data for bytes 1-4 is meaningless.) B5-B0 are 6 LSB's of current Y position
Byte 3	B7=1, B6=1,	B5-B0 are 6 MSB's of current X position
Byte 4	B7=1, B6=1,	B5-B0 are 6 LSB's of current X position

N. Routing Code 9EH - Data Tablet Pen State

This message contains no data bytes. It is sent if the pen from the graphics data tablet goes from up to down.

O. Routing Code 9FH - Windco Response

This message is in response to the second spacebar if in Windco mode (see 9FH host to terminal).

Byte 1	B7=1, B6=1,	B5-B0 are 6 MSB's of first image frame
Byte 2	B7=1, B6=1,	B5-B0 are 6 LSB's of first image frame
Byte 3	B7=1, B6=1,	B5-B0 are 6 MSB's of second image frame
Byte 4	B7=1, B6=1,	B5-B0 are 6 LSB's of second image frame
Byte 5	B7=1, B6=1,	B5-B0 are 6 MSB's of first 1/2 cursor line size
Byte 6	B7=1, B6=1,	B5-B0 are 6 LSB's of first 1/2 cursor line size
Byte 7	B7=1, B6=1,	B5-B0 are 6 MSB's of second 1/2 cursor line size
Byte 8	B7=1, B6=1,	B5-B0 are 6 LSB's of second 1/2 cursor line size
Byte 9	B7=1, B6=1,	B5-B0 are 6 MSB's of first 1/2 cursor element size
Byte 10	B7=1, B6=1,	B5-B0 are 6 LSB's of first 1/2 cursor element size
Byte 11	B7=1, B6=1,	B5-B0 are 6 MSB's of second 1/2 cursor element size
Byte 12	B7=1, B6=1,	B5-B0 are 6 LSB's of second 1/2 cursor element size
Byte 13	B7=1, B6=1,	B5-B0 are 6 MSB's of first cursor line position
Byte 14	B7=1, B6=1,	B5-B0 are 6 LSB's of first cursor line position
Byte 15	B7=1, B6=1,	B5-B0 are 6 MSB's of second cursor line position
Byte 16	B7=1, B6=1,	B5-B0 are 6 LSB's of second cursor line position
Byte 17	B7=1, B6=1,	B5-B0 are 6 MSB's of first cursor element position
Byte 18	B7=1, B6=1,	B5-B0 are 6 LSB's of first cursor element position
Byte 19	B7=1, B6=1,	B5-B0 are 6 MSB's of second cursor element position
Byte 20	B7=1, B6=1,	B5-B0 are 6 LSB's of second cursor element position
Byte 21	B7=1, B6=1,	B5-B0 are 6 MSB's of first graphic frame
Byte 22	B7=1, B6=1,	B5-B0 are 6 LSB's of first graphic frame
Byte 23	B7=1, B6=1,	B5-B0 are 6 MSB's of second graphic frame
Byte 24	B7=1, B6=1,	B5-B0 are 6 LSB's of second graphic frame

P. Routing Code A0H - Stereo Cursor State

This message allows the host to determine the state of the stereo cursor when in stereo cursor mode (see routing code 93H). This message always contains 10 bytes where:

Byte 1	B7=1, B6=1,	B5-B0 are 6 MSB's of 1/2 cursor line size
Byte 2	B7=1, B6=1,	B5-B0 are 6 LSB's of 1/2 cursor line size
Byte 3	B7=1, B6=1,	B5-B0 are 6 MSB's of 1/2 cursor element size
Byte 4	B7=1, B6=1,	B5-B0 are 6 LSB's of 1/2 cursor element size
Byte 5	B7=1, B6=1,	B5-B0 are 6 MSB's of cursor line position
Byte 6	B7=1, B6=1,	B5-B0 are 6 LSB's of cursor line position
Byte 7	B7=1, B6=1,	B5-B0 are 6 MSB's of cursor element position
Byte 8	B7=1, B6=1,	B5-B0 are 6 LSB's of cursor element position
Byte 9	B7=1, B6-B0	are 7 bit cursor type (only 5 are currently implemented)

where: type 1 = box
type 2 = cross hair
type 3 = box with cross hair
type 4 = solid rectangle
type 5 = star wars

Byte 10 B7=1, B6=1, B5-B3 are cursor color (bit each R,G,B)