40 CHARACTER VACUUM FLUORESCENT DISPLAY II and 2-SIDED VACUUM FLUORESCENT DISPLAY II PRODUCT ATTACH INFORMATION

Preface

This document is provided for use in conjunction with other relevant IBM publications, and IBM makes no warranty, either expressed or implied, relative to its completeness or accuracy. The information in this document is current as of the date of its publication, but is subject to change by IBM at any time without notice. This document is not intended to be a manufacturing or engineering specification; the user is assumed to understand the interrelationship among any affected systems, machines, programs, and media.

IBM may have patents or pending patent applications covering subject matter described in this document, including appearance design patents or applications. The furnishing of this document does not constitute or imply a grant of any license under any patents, patent applications, trademarks, copyrights, or other rights of IBM or of any third party; or any right to refer to IBM in any advertising or other promotional or marketing activities. IBM assumes no responsibility for any infringement of patents or other rights that may result from use of this document or from manufacture, use, lease, or sale of apparatus described herein.

Licenses under IBM′s utility patents are available on reasonable and non-discriminatory terms and conditions. IBM does not grant licenses under its appearance design patents. Inquiries relative to licensing should be directed in writing to:

IBM Director of Commercial Relations
International Business Machines Corporation
Armonk, New York 10504

April, 1988

Changes resulting in document revisions will be summarized in this table in reverse chronological sequence. Revision bars (|) will highlight the text changed in new document versions.

Version	Date	Change Description

Table 1. Change Summary

Contents

Serial Interface 1
Interface Connector 1
Input Data 1
Output Data 1

Voltage 3

Power-up/Reset 5

Display Interface 7
Serial I/O Addresses 7
Messages 7
Terminal to Display 7
Display to Terminal 13
Driver Programming Considerations 15

Serial Interface

Interface Connector

Signal and power connector - (4 pin shielded data link receptacle), (Amp P/N 1-520421-1).

The pin assignment for the connector will be as follows:

CONNECTOR PIN NO.	SIGNAL NAME
1	+V
2	Signal Line A
3	Signal Line B
4	Gnd

Input Data

This signal will be suitable to drive the DS3695 or equivalent RS422 transceiver module.

Output Data

This signal will be the output from the DS3695 or equivalent RS422 transceiver module.

Voltage

The voltage supplied to the display unit will be as follows:

Minimum	10.80 volts d. c.
Maximum	13.20 volts d. c.

Power-up/Reset

When the display is powered up, a **RESET** command is received from the terminal, or the display itself initiates an internal reset, the following sequence of events takes place:

1. The processor runs its BAT (RAS Basic Assurance Test). This checks out the processor, its internal ROM, its program load, its internal RAM, and the display RAM.
2. The display is cleared and ″U001″ is displayed on the bottom line of the display.
3. If the processor detects an error in its BAT, it will not talk to the terminal.
4. If no error is detected, the processor will look at the Serial I/O port. Since the display(s) can attach to the Serial I/O via either port 4A or 4B, the display processor must look for input from either 4A or 4B, since it doesn′t know which one it may be plugged into. It begins looking for a poll from the Serial I/O port. If a poll is not received or recognized within 50 milliseconds, the processor makes the assumption that it is plugged into the other SI/O port, and it effectively swaps the serial I/O lines (RS-485 A and B lines) and looks for a poll in this orientation (for up to 50 milliseconds). If it doesn′t see a poll in this orientation, it switches back. This cycle is repeated until a poll is recognized on display address, and the processor locks on that SI/O orientation. If the display is a two-sided display, it assigns the primary side to address X′24′ and the secondary side to address X′25′.
5. Once a poll is seen, the display processor establishes SI/O communications with the terminal by sending in a ″Request On Line″ (ROL) for each applicable address. The terminal then sends a ″Set Normal Response Mode″ (SNRM) for each address, and the display responds with an ″Non-Sequenced Acknowledge″ (NSA) for each address, indicating that normal, good communications have been established.
6. ″U003″ is displayed on the bottom line of the display, indicating that communications has been established.
7. The display driver should send a **DEVICE INFORMATION REQUEST** to the display processor. Based on the device ID in the device information byte, the driver can determine whether this is a single-sided VFD or a double-sided VFD.

Note: The preceding sequence of events is good for all ″reset″ situations, since, as a result of any reset, the display processor erases its RAM.

Display Interface

Serial I/O Addresses

The single-sided low cost 2x20, A/N display will respond to one of two serial I/O addresses, depending on which serial I/O display ports it's plugged into.

Table 2. Single-sided Display Serial I/O Addresses

PORT #	ADDRESS
4A	X'24'
4B	X'25'

The two-sided display will answer to both addresses (one for each side, of course).

Each display type (unique character sets, etc.) will have a unique identifier which will be sent to the terminal in response to the **DEVICE INFO REQUEST** system command. (See the "System Commands" section, below.)

Messages

Terminal to Display

There are four types of messages from the terminal to the display:

1. Set Normal Response Mode (SNRM)
2. Poll
3. Broadcast
4. Information Frame ("I-Frame")

A SNRM is a message from the terminal sent during power-up and is described in the Serial I/O Product Attachment Information spec.

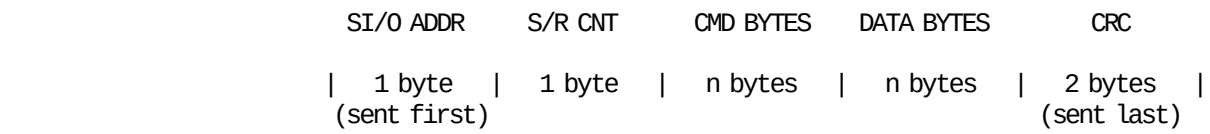
A poll is a request to the device for any status or data it has ready to be sent in to the terminal. Poll formats are described in the Serial I/O Product Attachment Information spec. Responses to polls are described below in the "Messages" section entitled "Display to Terminal".

A Broadcast message may be any "System" command, as described below, with a unique serial I/O address that is sent out to all I/O devices at the same time. The display will respond to a broadcast system command the same as if it were sent only to the display.

An "I-Frame" is a command to the display and is described below.

″I-Frame″ Message Format

The ″I-Frame″ messages from the terminal to the display will have the following format (this is what goes over the wires and not what is in the receive buffer):



Message Byte Definitions

NAME	DEFINITION
SI/O ADDR	Serial I/O address (one byte). See section ″Serial I/O Addresses″ above.
S/R CNT	Send/Receive Count (one byte). Also called ″Link Control Byte″. For description, see the Serial I/O Product Attachment Information spec.
CMD BYTES	Command Bytes (2 to 3 bytes). See section ″Command Definitions″ below.
DATA BYTES	Data Bytes (0 to 41 bytes). See section ″Data Definitions″ below.
CRC	Cyclic Redundancy Check (two bytes). For description, see the Serial I/O Product Attachment Information spec.

Command Definitions

Display Commands

The following **Display** commands can be sent from the terminal to the display:

Table 3 (Page 1 of 3). Display Commands

COMMAND	HEX CODE	DESCRIPTION
CLEAR TOP LINE	X′01 LL′	Clear all characters from the top line of the display and place the cursor at character location ″LL″. See cursor location parameter byte definition below.
CLEAR BOTTOM LINE	X′02 LL′	Clear all characters from the bottom line of the display and place the cursor at character location ″LL″. See cursor location parameter byte definition below.
UNDERLINE CURSOR	X′10 00′	No action will be taken on this command. A command reject will not be sent.

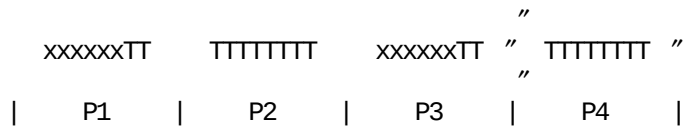
Table 3 (Page 2 of 3). Display Commands		
COMMAND	**HEX CODE**	**DESCRIPTION**
BOX CURSOR	X′10 01′	No action will be taken on this command. A command reject will not be sent.
MOVE CURSOR	X′20 LL′	No action will be taken on this command. A command reject will not be sent.
WRITE RAM	X′40 PP data′	Write the eight horizontal slice data bytes following ASCII value byte ′PP′ to RAM. The slice data will redefine the ASCII character located at X′PP′. Byte ″PP″ may have a value of between X′00′ and X′07′.
WRITE TOP LINE	X′81 LL data′	Write the 20 data bytes and one 8-bit sum byte following the cursor location parameter byte ′LL′ to the top line of the display, and place the cursor at cursor location ″LL″. See cursor location parameter byte definition below. The 8-bit sum byte in not displayed.
WRITE BOTTOM LINE	X′82 LL data′	Write the 20 data bytes and one 8-bit sum byte following the cursor location parameter byte ′LL′ to the bottom line of the display, and place the cursor at cursor location ″LL″. See cursor location parameter byte definition below. The 8-bit sum byte in not displayed.
WRITE TRI-ANGLE MARKS	X′21 P1 P2 P3 P4′	Turn the triangle marks on/off. P1 and P2 control the leftmost 10 marks and P3 and P4 control the rightmost 10 marks. Each bit in Px controls one triangle mark. All 8 bits in P2 and P4 are utilized; the lower 2 bits are utilized in P1 and P3. A 1 = triangle mark on and a 0 = tri-angle mark off. See figure below under Triangle Mark Definition Bytes.

Table 3 (Page 3 of 3). Display Commands		
COMMAND	**HEX CODE**	**DESCRIPTION**
SELECT CHARACTER SET	X′41 PP′	Select which character set will be used as follows: PP = 00 --- Character set 0 (US/European) PP = 01 --- Character set 1 (Katakana) If PP takes on any value other than ′00′ or ′01′, then a command reject will be sent.Note that the US/European character set contains 24 Greek, 6 Turkish, and 4 Hungarian characters.

Cursor Location Parameter Byte Definition

For all commands which require a cursor location parameter byte (indicated as byte ′LL′), that byte is ignored since the low cost VFD does not have a cursor.

Triangle Mark Definition Bytes



System Commands

The following **SYSTEM** commands can be sent from the terminal to the display:

Table 4 (Page 1 of 2). Display System Commands		
COMMAND	**HEX CODE**	**DESCRIPTION**
EXTENDED INDICATOR	X′00 00 00 LL data′	Tells the processor to display the ASCII message contained in the ″data″ bytes (up to 40 data bytes, which is X′LL′ bytes long) The data will be ASCII characters ″A″ - ″Z″ and ″0″ - ″9″. Note that only the first 20 characters will be displayed. The purpose of this command is to remain compatible with the 5x12 matrix VFD.

Table 4 (Page 2 of 2). Display System Commands		
COMMAND	**HEX CODE**	**DESCRIPTION**
SET INDICATOR	X′00 0X′	Tells the processor to display the IPL message ″U00X″. ″X″ may be from ″1″ to ″9″.
TEST REQUEST	X′00 10′	Causes the display processor to perform any internal tests and send the resulting status in to the terminal. The low cost VFD responds with the same data as if the status request command had been sent.
STATUS REQUEST	X′00 20′	Asks the processor to send current status in to the terminal.
RESET	X′00 40′	Causes the display processor to initiate a software reset from 80C52 address X′0000′. This simulates a POR.
EC LEVEL REQUEST	X′00 80′	Asks the processor to send current status and the processor′s EC level byte in to the terminal.
DEVICE INFORMATION REQUEST	X′00 00 01′	This command causes the display processor to send its display device ID and related information to the terminal. The description of the response is located in the ″Data Byte Definitions″ section, below.
(OTHERS)	X′00 XX′	Any other valid or invalid system command will be responded to with a command reject.

Display Command Summary

Table 5 (Page 1 of 2). Serial I/O Display Command Summary	
COMMAND	**HEX CODE**
CLEAR TOP LINE	01 LL
CLEAR BOTTOM LINE	02 LL
UNDERLINE CURSOR	10 00
BOX CURSOR	10 01
MOVE CURSOR	20 LL
WRITE RAM	40 PP data(8)
WRITE TOP LINE	81 LL data(20) sum(1)

Table 5 (Page 2 of 2). Serial I/O Display Command Summary	
COMMAND	**HEX CODE**
WRITE BOTTOM LINE	82 LL data(20) sum(1)
EXTENDED INDICATOR	00 00 00 LL data
SET INDICATOR	00 0X
TEST REQUEST	00 10
STATUS REQUEST	00 20
RESET	00 40
EC LEVEL REQUEST	00 80
DEVICE INFO REQUEST	00 00 01
WRITE TRIANGLE	21 P1 P2 P3 P4
SELECT CHARACTER SET	41 PP

Data Definition

Character ROM

The two character sets are stored in the 8052 ROM on the VFD logic board. The following table defines the ASCII character code and the character that ROM will display.

User-Definable Characters

Hex locations X′00′ to X′07′ are user definable characters. These locations may be loaded with whatever characters the OS/application wishes. The character is loaded into the character RAM via the **WRITE RAM** command, described above. They must be loaded after any reset by the display processor.

At power-up, the display processor loads two characters representing a ″key″ symbol (used to indicate that a password must be entered), and two characters representing an ″OK″ with a slash through it (indicating an error). They are available to the terminal′s POST code, and may be overwritten by the driver or application, when they load.

These characters are:

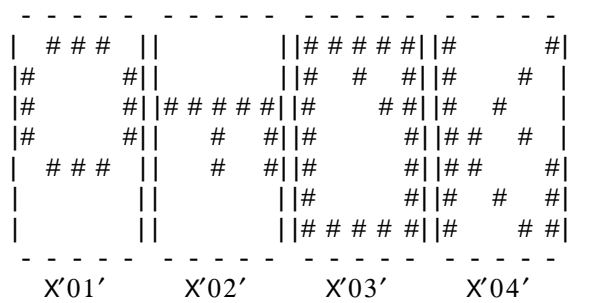


Figure 1. Operator Display POST Characters

Display to Terminal

There are three types of display responses to a poll from the terminal:

1. Non-Sequenced Acknowledgement (NSA)
2. ″End of Poll″ (EOP)
3. Information Frame (″I-Frame″)

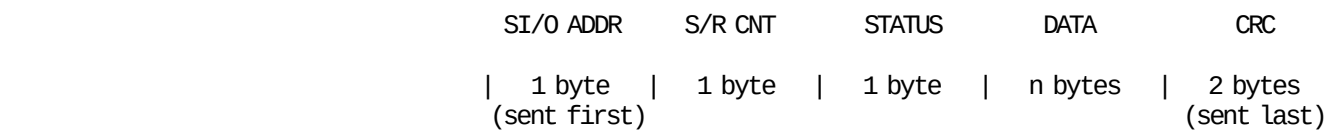
An NSA is sent to the terminal in response to a SNRM or Broadcast command. NSAs are described in the Serial I/O Product Attachment Information spec.

An EOP is returned to the terminal when the display has no information to send. It is an acknowledgement that the poll was received and that the display is still communicating on the SI/O interface. The EOP format is described in the Serial I/O Product Attachment Information spec.

An I-Frame is status or data from the display and is described below.

″I-Frame″ Message Format

The ″I-Frame″ messages from the display to the terminal will have the following format (this is what goes over the wires and not what is in the receive buffer):



Message Byte Definitions

NAME	DEFINITION
SI/O ADDR	Serial I/O Address (one byte). See section ″Serial I/O Addresses″ above.
S/R CNT	Send/Receive Count (one byte). Also called ″Link Control Byte″. For description, see the Serial I/O Product Attachment Information spec.
STATUS	Status Byte (one byte). See section ″Status Byte Definition″ below.
DATA	Data bytes (1 or 5 bytes). See section ″Data Byte Definitions″ below.
CRC	Cyclic Redundancy Check (two bytes). For description, see the Serial I/O Product Attachment Information spec.

Status Byte Definitions

Status Byte #0

(first status byte sent)

Table 6. Display Status Byte #0	
BIT	**DEFINITION**
7 (msb)	Command Reject '1' = The command sent from the terminal was **NOT** a valid display command. '0' = The last command sent from the terminal was a valid display command.
6	(RESERVED)
5	EC Level Response '1' = The EC level byte follows the status byte in the message '0' = EC level byte not in message
4	Device Info Response '1' = Device Information follows the status byte in the message '0' = Device Information not in message
3-0	(RESERVED)

Data Byte Definitions

The data bytes in the message from the display to the terminal may be defined as:

1. No data bytes
2. EC level byte (one byte), or
3. Device Information (five bytes)

They are defined below.

No Data Bytes

No data bytes in the message means that only status was sent to the terminal. This can only happen as a result of a command being sent from the terminal. No data bytes in the message indicate that the EC level byte or Device Info bytes are not being sent in.

EC Level Byte

As a result of the **EC LEVEL REQUEST** command from the terminal, the display will return the one EC level byte of the processor in the data byte field of the message from the display to the terminal. Also, the "EC Level Response" status bit will = "1" in this message.

Device Information

As a result of the **DEVICE INFO REQUEST** command from the terminal, the display will return the device information in the data byte field of the message from the display to the terminal. Also, status byte #0, bit #4 will = "1" in this message.

```
   DEV TYPE      DEV ID      FEATURES      CMD SET     EC LEVEL

|  1 byte   |   1 byte   |   1 byte   |   1 byte   |   1 byte   |

(sent first)                                          (sent last)
```

The device information bytes are defined as:

Table 7. Device Information Byte Definitions

BYTE	DEFINITION
Device Type	Always = '20'X (display device type)
Device ID	= '06'X (Single-sided VFD, Gull II) = '07'X (Double-sided VFD, Seagull)
Language	= '00'X (the character set defined above)
Command Set	= '02'X (the command set as defined in this spec.)
EC Level	Microcode EC level (same as what's returned from the **EC LEVEL REQUEST** command)

Driver Programming Considerations

See the "Power-up / Reset" section above for the sequence of events which takes place after ***ANY*** display reset (POR, **RESET** command, internal reset, etc.).

The display processor will load two characters (four bytes) into the user-definable RAM at power-up. These characters are required by 469x terminal POST code. They are the "key" symbol and the "not OK" symbol. They are described above.

Again, the differences between this display and the operator display on the 469x 50-Key/Display keyboard are:

1. The "Code Update" status bit (present on the keyboard display) is not implemented here.
2. The device ID (response to the **DEVICE INFO REQUEST** command) is different.

Since the "Code Update" status bit is not present (because the microcode can't be updated), the driver can't wait for that status to arrive before doing anything (as in the 50-Key/Display).

The A wait time should be a minimum of 60 HEX.

Everything else is the same.

DSMKIM520E IMBED OR APPEND FILE NOT FOUND.
DSMMOM395I ′.EDFIM′ LINE 690: .im gulliich
DSMMOM397I ′.EDFIM′ WAS IMBEDDED AT LINE 690 OF ′.EDFIM′
DSMMOM397I ′.EDFIM′ WAS IMBEDDED AT LINE 10 OF ′.IM′
DSMMOM397I ′.IM′ WAS IMBEDDED AT LINE 509 OF ′VFD-40-2′
DSMBEG323I STARTING PASS 2 OF 2.
DSMKIM520E IMBED OR APPEND FILE NOT FOUND.
DSMMOM395I ′.EDFIM′ LINE 690: .im gulliich
DSMMOM397I ′.EDFIM′ WAS IMBEDDED AT LINE 690 OF ′.EDFIM′
DSMMOM397I ′.EDFIM′ WAS IMBEDDED AT LINE 10 OF ′.IM′
DSMMOM397I ′.IM′ WAS IMBEDDED AT LINE 509 OF ′VFD-40-2′