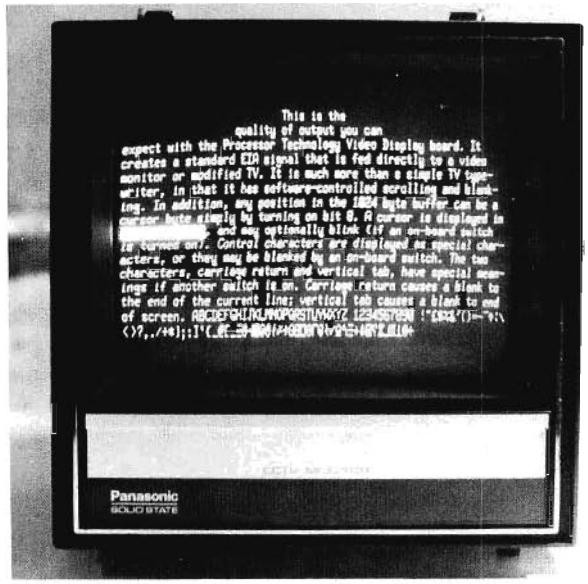

Photo 1: An example of the display output of the Processor Technology VDM-1 driving a standard video monitor purchased locally. Upper and lower case output with optional inverse video lends flexibility to the system. In this photograph, the letters in the inverse video rectangle do not show. A timing distortion in the line with inverse video was found in the author's VDM-1, evidence of which can be seen in this photograph.

Product Review:

Processor Technology VDM-1

D Anderson
755 Southmore Dr W
Ottawa Ontario CANADA

Processor Technology's Video Display Module for the Altair, IMSAI, and other Altair compatible machines is of excellent quality. The board has gold plated fingers, and solder resist (green lacquer) on both sides of the board. All component designations are silkscreened and are easily readable. The board displays 16 lines of 64 characters on a standard video monitor or modified TV.

The board has 48 integrated circuits, including 8 91L02As for 1024 bytes of visible memory, and a character generator ROM. A crystal oscillator generates the required frequencies for a standard video signal. Sockets are provided for all integrated circuits. A DIP switch is provided to set the board options.

The VDM-1 has a hardware cursor feature in its design, controlled by "cursor bytes" within the displayed text. A cursor byte is any byte having the high order bit on. A cursor byte may contain any character, and will be displayed in inverse video. That is, if

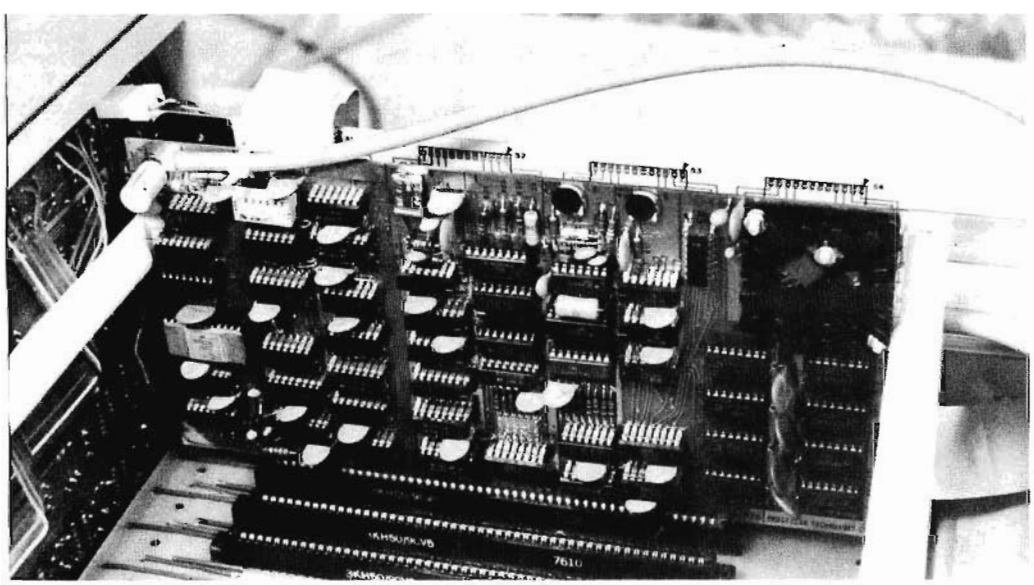

Photo 2: The VDM-1 shown mounted in the author's computer system. The thick coaxial cable at the top of the board runs to the monitor shown in photo 1.

Listing 1: A bootstrap loader, modified from Processor Technology's Teletype bootstrap loader, which allows entry of data in hexadecimal with instant display on the video output of the VDM-1. The author's ASCII keyboard is interfaced through 8080 ports 10 and 11 (hexadecimal). The VDM-1 control register is interfaced to port 8C. The VDM-1 memory in the author's system is located at hexadecimal addresses 8800 to 8BFE.

```
8080 MACRO ASSEMBLER, VER 2.3 BOOT LOADER ERRORS = 0 PAGE 1

                        TITLE 'BOOT LOADER'
0000                    ORG   0
005F            CURSR   EQU   5FH
8800            VDM     EQU   8800H
008C            VDMND   EQU   8CH           ; 8800H + 400H
0400            STACK   EQU   400H
000B            VT      EQU   11
000D            CR      EQU   13
                ;
                ;       ENTRY BOOT,DSPLY
                ;
0000  310004    BOOT:   LXI   SP,STACK
0003  CD3700            CALL  INIT          ; INITIALIZE KB & VDM
0006  110088            LXI   D,VDM         ; CLEAR SCREEN
0009  3E0B              MVI   A,VT          ; VERTICAL TAB CLEARS THE SCREEN
000B  12                STAX  D
000C  017000            LXI   B,0070H       ; SET LOAD ADDR
                ;
000F  2E00      ASHEX:  MVI   L,0           ; CLEAN HOUSE
0011  CD4300    LOOP1:  CALL  IN8           ; GET A BYTE
0014  CD4F00            CALL  DSPLY         ; DISPLAY IT
                ;
0017  DE30              SBI   '0'           ; CONVERT TO BCD
0019  FA2C00            JM    ONEWD
001C  FE0A              CPI   10
001E  DA2300            JC    DOIT
0021  C6F9              ADI   0F9H          ; CONVERT A-F DOWN
0023  29        DOIT:   DAD   H
0024  29                DAD   H
0025  29                DAD   H
0026  29                DAD   H
0027  85                ADD   L
0028  6F                MOV   L,A
0029  C31100            JMP   LOOP1
                ;
002C  FEF0      ONEWD:  CPI   0F0H          ; IS IT SPACE?
002E  C20F00            JNZ   ASHEX
0031  7D                MOV   A,L           ; GET CHR FROM L
0032  02                STAX  B             ; STORE BYTE
0033  03                INX   B             ; POINT TO NEXT BYTE
0034  C30F00            JMP   ASHEX
                ;
0037  AF        INIT:   XRA   A             ; SET CTL BITS FOR PIO DDR
0038  D310              OUT   10H
003A  D311              OUT   11H           ; INDICATE ALL LINES INPUT
003C  D38C              OUT   8CH           ; SET VDM CTL
003E  3E06              MVI   A,06H         ; TURN ON DATA REG
0040  D310              OUT   10H
0042  C9                RET
                ;
0043  DB10      IN8:    IN    10H           ; GET STATUS

8080 MACRO ASSEMBLER, VER 2.3 BOOT LOADER ERRORS = 0 PAGE 2

0045  E680              ANI   80H           ; DATA AVAILABLE?
0047  CA4300            JZ    IN8
004A  DB11              IN    11H           ; GET CHAR IN A
004C  E67F              ANI   7FH           ; DROP PARITY BIT
004E  C9                RET
                ;
                ;       DISPLAY CHAR IN ACCUMULATOR
004F  12        DSPLY:  STAX  D             ; DISPLAY IT
0050  13                INX   D
0051  EB        DSCRS:  XCHG
0052  365F              MVI   M,CURSR       ; SET UP CURSOR
0054  23                INX   H
0055  360B              MVI   M,VT          ; SET UP VERTICAL TAB TO CLEAR SCREEN
0057  2B                DCX   H
0058  EB                XCHG
0059  F5                PUSH  PSW           ; SAVE CHAR
005A  7A                MOV   A,D           ; CHECK FOR END OF SCREEN
005B  FE8C              CPI   VDMND         ; END OF VDM MEMORY?
005D  C26700            JNZ   $+10
0060  110088            LXI   D,VDM         ; RESET TO START
0063  F1                POP   PSW           ; RESTORE CHAR
0064  C35100            JMP   DSCRS
0067  F1                POP   PSW           ; RESTORE CHAR
0068  C9                RET
                ;
                        END
NO PROGRAM ERRORS

8080 MACRO ASSEMBLER, VER 2.3 BOOT LOADER ERRORS = 0 PAGE 3

                        SYMBOL TABLE

* 01

A       0007    ASHEX   000F    B       0000    BOOT    0000 *
C       0001    CR      000D *  CURSR   005F    D       0002
DOIT    0023    DSCRS   0051    DSPLY   004F    E       0003
H       0004    IN8     0043    INIT    0037    L       0005
LOOP1   0011    M       0006    ONEWD   002C    PSW     0006
SP      0006    STACK   0400    VDM     8800    VDMND   008C
VT      000B
```

the display is normally white characters on a black background, a cursor byte will be displayed as a black character on a white background. By setting one position on the DIP options switch, all cursor bytes, if any, will blink at about a one second rate.

Other positions on the DIP options switch cause control characters, such as STX, DEL, etc, to be blanked. If not blanked, they appear as strange characters. Another switch controls the two special blanking characters: carriage return and vertical tab. If the switch is on, a carriage return character will automatically blank itself and all data following it on the line. The vertical tab character will blank itself and all data following it in the memory buffer. In both cases, only the data on the screen is blanked; the data in memory is left unaltered. If unused portions of display memory are to be used as a program segment, this prevents "garbage" from appearing on the screen.

The 1 K static programmable memory buffer is directly addressable as memory in the Altair. Displaying data on the screen involves nothing more than storing data into the VDM's memory. Sounds easy, and it is. The display is essentially instantaneous. The sample dump program shown in listing 1 is impressive in that memory is dumped instantly.

The VDM board contains two 4 bit registers which control the scrolling and window shading of the display. "What is window shading?" you ask. Window shading is the process of blanking a display from the top down to a specified line. This is performed automatically by simply loading the window shade register. Scrolling is also easy and performed the same way. Both registers are loaded simultaneously by issuing an output instruction to the board.

This all sounds pretty simple, but it turns out that a fair amount of software is required to simulate the operation of a dumb CRT. This is because a carriage return and line feed sequence requires that you set your memory pointer back to the beginning of the line, add 64 to it, and check that you

Listing 2: A hexadecimal memory dump program which displays 256 bytes of memory formatted 16 bytes per line, 16 lines in all. This program displays one page starting at the location stored in address 0074 (hexadecimal) and then waits for a keyboard input before proceeding to display the next page.

```
8080 MACRO ASSEMBLER, VER 2.3 DUMP MEMORY ON SCREEN ERRORS = 0 PAGE 1

                        TITLE 'DUMP MEMORY ON SCREEN'
                    ;
                    ; THIS PROGRAM DISPLAYS MEMORY ON THE SCREEN, ONE PAGE
                    ; AT A TIME. 16 BYTES ARE DISPLAYED IN HEX ON EACH LINE.
                    ; 16 LINES ARE DISPLAYED AT ONCE. EACH LINE BEGINS
                    ; WITH THE HEX ADDRESS. AFTER EACH PAGE IS DISPLAYED, THE
                    ; PROGRAM HALTS INDEFINITELY, UNTIL ANY CHARACTER
                    ; IS TYPED: THEN THE NEXT PAGE IS DISPLAYED.
                    ;
  0074                      ORG   0074H
  8800              VDM     EQU   8800H
  0400              STACK   EQU   400H
  0043              INB     EQU   0043H
  000D              CR      EQU   13
                    ;
                    ;       ENTRY DUMP,BLANK,BINH,CRLF,CLEAR
                     ;
  0074 310004               LXI   SP,STACK
  0077 210000               LXI   H,0             ; ADDR OF MEMORY TO BE DISPLAYED
  007A CDCF00       DUMP:   CALL  CLEAR           ; CLEAR VDM MEMORY
  007D 110688               LXI   D,VDM+6         ; POINT TO VDM MEMORY + 6 OFFSET
  0080 0610         DMPGO:  MVI   B,16            ; 16 LINES
  0082 7C           DMLP1:  MOV   A,H             ; DISPLAY ADDR
  0083 CDA400               CALL  BINH
  0086 7D                   MOV   A,L
  0087 CDA400               CALL  BINH
  008A CDBF00               CALL  BLANK           ; FOLLOWED BY A BLANK
                    ;
  008D 0E10                 MVI   C,16            ; 16 BYTES PER LINE
  008F 7E           DMLP2:  MOV   A,M             ; GET BYTE TO CONVERT TO HEX
  0090 CDA400               CALL  BINH            ; CONVERT TO ASCII HEX
  0093 CDBF00               CALL  BLANK           ; DISPLAY A BLANK
  0096 23                   INX   H               ; POINT TO NEXT BYTE TO BE DISPLAYED
  0097 0D                   DCR   C
  0098 C28F00               JNZ   DMLP2           ; DO ALL 16 BYTES
                    ;
  009B CDC400               CALL  CRLF            ; POINT TO NEXT LINE ON SCREEN
  009E 05                   DCR   B
  009F C28200               JNZ   DMLP1           ; DO ALL 16 LINES
  00A2 CD4300               CALL  INB             ; WAIT FOR INPUT
  00A5 C37900               JMP   DUMP+3          ; DO NEXT PAGE
                    ;
                    ;       CONVERT ACCUM TO ASCII HEX WHERE D,E POINT
                    ;
  00A8 F5           BINH:   PUSH  PSW             ; SAVE BYTE TO BE CONVERTED
  00A9 1F                   RAR                   ; SHIFT RIGHT 4 BITS
  00AA 1F                   RAR
  00AB 1F                   RAR
  00AC 1F                   RAR
  00AD CDB500               CALL  BIN1            ; CONVERT TO ASCII
  00B0 12                   STAX  D               ; DISPLAY ASCII BYTE
1
8080 MACRO ASSEMBLER, VER 2.3 DUMP MEMORY ON SCREEN ERRORS = 0 PAGE 2

  00B1 13                   INX   D               ; POINT TO NEXT POSN ON SCREEN
  00B2 F1                   POP   PSW             ; GET ORIGINAL BYTE
  00B3 CDB500               CALL  BIN1            ; CONVERT TO ASCII
  00B6 12                   STAX  D               ; DISPLAY IT
  00B7 13                   INX   D               ; POINT TO NEXT POSN ON SCREEN
  00B8 C9                   RET
                    ;
                    ;       CONVERT A BYTE TO ASCII HEX
                    ;
  00B9 E60F         BIN1:   ANI   0FH             ; LOW 4 BITS
  00BB C630                 ADI   48              ; MODIFY FOR ASCII
  00BD FE3A                 CPI   58              ; DIGIT 0-9?
  00BF D8                   RC
  00C0 C607                 ADI   7               ; MODIFY FOR A-F
  00C2 C9                   RET
                    ;
                    ;
                    ;       DISPLAY A BLANK
                    ;
  00C3 3E20         BLANK:  MVI   A,' '           ; GET A BLANK
  00C5 12                   STAX  D               ; DISPLAY IT
  00C6 13                   INX   D               ; POINT TO NEXT POSN ON SCREEN
  00C7 C9                   RET
                    ;
                    ;       ISSUE CARRIAGE RETURN LINE FEED FOR VDM
                    ;
  00C8 7B           CRLF:   MOV   A,E             ; POINT TO NEXT LINE ON SCREEN
  00C9 E6C0                 ANI   0C0H
  00CB C646                 ADI   70              ; LENGTH OF LINE + 6 OFFSET
  00CD 5F                   MOV   E,A
  00CE 3E00                 MVI   A,0
  00D0 8A                   ADC   D
  00D1 57                   MOV   D,A
  00D2 C9                   RET
                    ;
                    ;       BLANK VDM MEMORY
                    ;
  00D3 110088       CLEAR:  LXI   D,VDM           ; POINT TO VDM BUFFER
  00D6 3E20                 MVI   A,' '           ; GET A BLANK
  00D8 0610                 MVI   B,16            ; 16 LINES
  00DA 0E40         DMLP3:  MVI   C,64            ; OF 64 BYTES
  00DC 12           DMLP4:  STAX  D               ; BLANK 1 BYTE
  00DD 13                   INX   D               ; POINT TO NEXT BYTE IN BUFFER
  00DE 0D                   DCR   C
  00DF C2DC00               JNZ   DMLP4           ; DO ALL 64 BYTES
  00E2 05                   DCR   B
```

haven't exceeded the limit of the memory buffer. You then must store your cursor byte, and add 1 to your scrolling counter, and verify that it has not passed 16. All of this adds up in memory requirements and programming time. It might have been better to have an automatic carriage return, line feed sequence handled directly by the hardware.

Another unusual feature of the board is that it has a circuit that creates a pulse approximately four times per second. You can tie this to the interrupt line or vectored interrupt bus if you wish to try some real time programming. Or you can test this timing pulse by issuing an input command to the board: Data bit 0 will go high every

VDM-1 SUMMARY

Product:	Altair-compatible video display board.
Manufacturer:	Processor Technology.
Price:	$199 kit.
Power Consumption:	+8 V/1 A max; +16 V/50 mA typical; −16 V/30 mA typical.
Size:	5.3 x 10.0 inches (13.5 x 25.4 cm) (Altair/IMSAI card cage dimensions).
Display Size:	16 lines of 64 characters.
Storage Medium:	91L02A low power static RAMs.
Features:	• upper and lower case displayed, as well as many special characters. • instant updating of display. • lacquer protected board (both sides). • quarter-second timer on board. • scrolling and window-shading software controlled. • automatic blanking with CR and VT. • control characters may be blanked. • multiple (optionally blinking) cursors.
Auxiliary Equipment Required:	Television Monitor. Interconnection Cables for Monitor. To be used with an Altair compatible mainframe.
Board Quality:	Excellent.
Documentation:	Excellent.
Delivery:	Slow, 60 days minimum.
Comments:	1. More software required than dumb CRT or TTY. 2. Instant update of display. 3. Monitor may not be able to display very many inverse video bytes.

quarter second. Thus if you were scrolling through a large source program, you could use this to delay the display on each line in multiples of a quarter second, without writing any complicated timing loops.

The documentation provided with the board is excellent and includes sample photos of what should be displayed at various points in the assembly process. An oscilloscope should not be required, but will obviously be helpful if you happen to get a bad chip.

Incidentally, one of several possible character generator ROMs will be provided, depending on availability. You have no choice.

One thing that should be pointed out is that your video monitor may not be able to display very many inverse video characters, as the horizontal sync gets messed up.

Although the board contains its own on-board horizontal and vertical video controls, it may not be possible to correct the image. As an example, see photo 1. The white rectangle on the left of the screen contains the words "inverse video" (they don't show up very well in the photo). Note the resulting slant to the rest of the characters on that line and on the next. I was unable to adjust the set or the board to eliminate the problem. In summary, the ProTech VDM board is well worth the money. Expect a minimum of 60 days' delivery, but don't hold your breath. Demand for this product is brisk. Incidentally, video monitors can be purchased from audio visual supply dealers for $150 or so. An appropriate connector from the board's cable to the set can be obtained at Radio Shack.■

Listing 2, continued:

```
00DF   C2D600          JNZ   DMLP3       ; DO ALL 16 LINES
00E2   3E00            MVI   A,0
00E4   D38C            OUT   8CH         ; INITIALIZE VDM
00E6   C9              RET
                ;
8080 MACRO ASSEMBLER, VER 2.3 DUMP MEMORY ON SCREEN ERRORS = 0 PAGE 3

                       END
NO PROGRAM ERRORS
8080 MACRO ASSEMBLER, VER 2.3 DUMP MEMORY ON SCREEN ERRORS = 0 PAGE 4

                    SYMBOL TABLE

  * 01

  A       0007     B       0000     BINI    0085     BINH    00A4
  BLANK   00BF     C       0001     CLEAR   00CF     CR      000D *
  CRLF    00C4     D       0002     DMLP1   007E     DMLP2   008B
  DMLP3   00D6     DMLP4   00D8     DMPGO   007C *   DUMP    0076
  E       0003     H       0004     INB     0043     L       0005
  M       0006     PSW     0006     SP      0006     STACK   0400
  VDM     8800

R:
```