Product Review:

Processor Technology VDM-1

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
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 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 1K 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 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 BBFE.

```
Listing 1...
```

```
MKROM MACRO ASSEMBLER, V1.2.3 ROOM MAJOR ERRORMSG = PAGE 1

MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```

```
MKROM...
```
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.

 hasn'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
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.