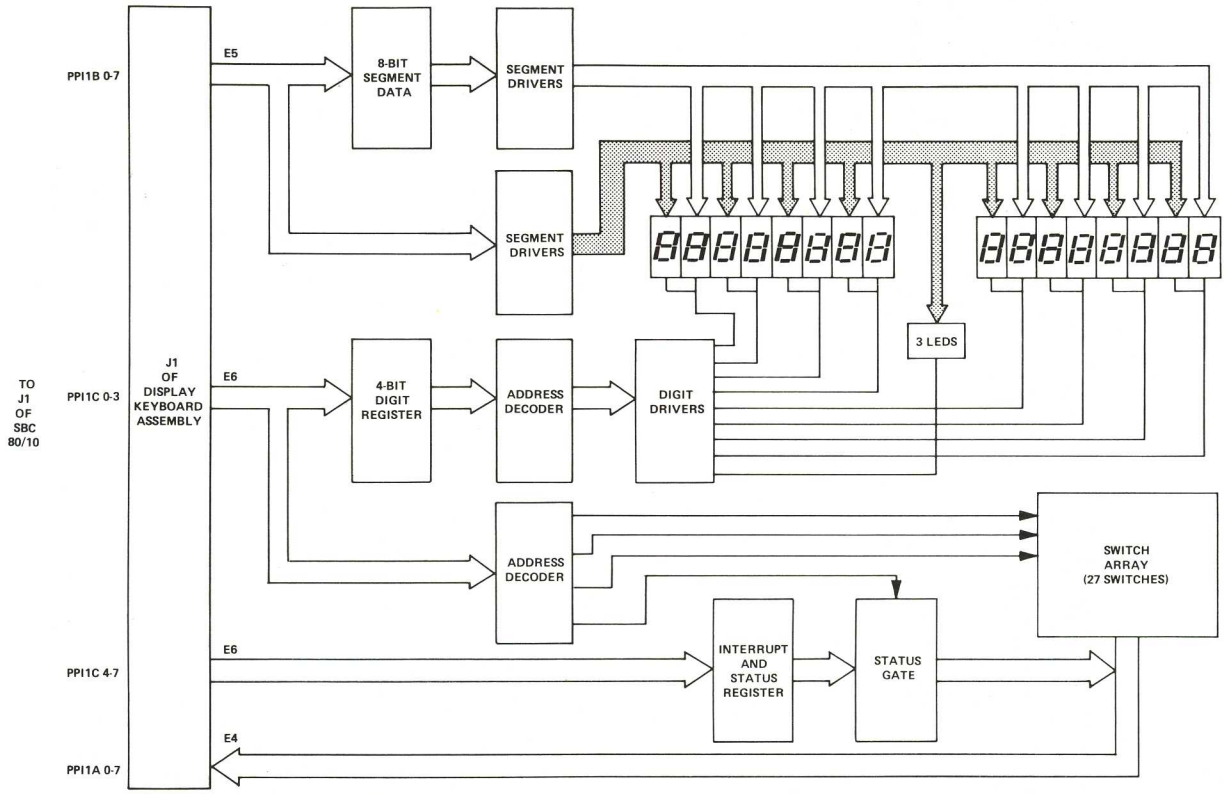
CARD-OF-THE-MONTH

CROMEMCO BYTESAVER™

by Roger Edelson



Display Keyboard Assembly, I/O & PROM Programming Functions, Block Diagram

Are you tired of reentering your favorite program everytime you turn off power? Does the thought of waiting 15 minutes for your BASIC to read in fill you with dread? Would you like to have your assembler/editor/monitor available within one second after turn-on? If you answered yes to any of these questions, then you ought to consider this month's card — the CROMEMCO BYTESAVER™.

If I could purchase one board (after some memory and an I/O card) the BYTESAVER would be my choice. The board is both an economical PROM (Programmable Read Only Memory), with the capacity for a full 8K bytes of storage, and a low-cost means of storing your programs in PROM. The BYTESAVER™ is an S-100 bus compatible memory board. Space is provided for eight 2708 U.V. erasable PROMs, for a full 8K bytes of memory. Using the 2708 allows your 8080 machine to operate no wait states, however, provision for a wait state is provided should you desire to purchase slower 2704 or 2708 PROMS.

The BYTESAVER™ also provides you with your own low cost PROM programmer. Using the software provided programs may be transferred from non-permanent RAM memory to the permanent PROM memory in the BYTESAVER™. Once your program is stored in PROM it's protected from power turn-offs, either accidental or intentional. The PROMs used are U.V. erasable and may be used again and again. Your program may be run either directly out of PROM, or using the software provided, it may be transferred to RAM memory beginning at any 1K segment selected. The software provided with the BYTESAVER™ controls both the transfer of data from the PROMs or to the PROMs. It is designed so there is no need for a keyboard. Just set the computer sense switches as instructed in the documentation. Transfer of RAM content to PROM takes less than a minute, and transfer of the 8K from PROM to RAM is complete in less than one second. The software controls the computer lights to provide verification of complete and accurate transfer of memory content.

As far as the design goes, the BYTESAVER™ is not particularly complex. The board is S-100 bus (ALTAIR/-IMSAI) compatible, and contains space for eight 2708/2704 PROMs. In order to program the eight 2708 PROMs a high voltage (+30V) supply is required, and this is generated from the regulated +5V supply by a DC to DC Converter. Switching (Protect/-Unprotect) is provided to disable this supply to avoid inadvertent overwrite of the memory in the case of a user program malfunction. The DC to DC converter appears to be a standard blocking oscillator configuration with a very simple feedback regulator. I say appears because the schematic does not show the internal arrangement of the pulse transformer used in the converter. Since there is hardly anything magic about a blocking oscillator it seems strange that Cromemco doesn't help those of us who might be able to service our own boards by including the transformer arrangement.

Distribution of the program voltage to the 2708s is by means of discrete transistors controlled by 7406 open-collector inverting buffers. A schematic of the BYTESAVER™ is shown in Figure 3. Control of the $\overline{CS}$/WE pin (which requires +12 V during WRITE) is done by a 7406 and a discrete transistor which allows

this pin to float up to +12V. During READ operation the $\overline{CS}$/WE pin is either a ground or +5V, as the +12V supply is disabled.

While we are discussing the schematic let me suggest to Cromemco that the 7406 is an inverting buffer and I would prefer it to be shown that way, always. It is difficult to follow a logic diagram if the inverting sign is left out, even if it is desired to show logical flow. If necessary an inverting circle could have been placed at the $\overline{CS}$/WE input pin at the 2708. A further help would have been to have the component designation on the schematic also. These are all minor quibbles, however, and they do not detract from the board's usefulness and operation.

Returning to the BYTESAVER™ circuit, the data bus lines are all buffered by 74367's (the standard 54/74 equivalent to the ubiquitous 8T97 series). Address decoding for the highest 8K of memory (A13-A15) is provided by jumper connections on the board. In order to provide addressing in this fashion using a three input NAND gate (rather than the slightly more expensive and easier to use octal decoder and dip switch) inverted address signals must be made available. Three inverters of a 74LS04 are used to derive $\overline{A13}$, $\overline{A14}$, and $\overline{A15}$. Since there were three inverters left over, these were used to buffer the A10 through A12 signals to the 7442. For this reason the low order address selects the output 7 pin. A bit strange but it works, and very well.

A 74123 dual retriggerable one-shot is used to provide the proper pulse width (N/MS) needed for programming the PROM. This pulse is also used to produce a wait state during PROM programming. As mentioned earlier, it is possible to jumper connect the capability for one wait state if slower PROMs are used. This is done at the input to the three input NAND gate whose output drives the clear pin on the second half of the 7474 'D' type Flip-Flop.

The board is high quality G-10 material with tinned solder pads and gold plated edge board connectors for reliability. The kit is fully socketed and assembly is suggested to be about one evening; it doesn't even take that long. A high quality soldermask contributes to the ease of assembly and minimizes the chance of solder splashes. The assembly instructions are explicit enough and the board is adequately screened to provide easy identification of component placement. Cromemco recommends no diode be installed in the position below QO. This will prevent inadvertant rewrite of the BYTEMOVER™ stored in PROM 0. The assembly instructions also warn that a pair of IC regulators look physically similar, BUT are NOT interchangeable. Also a nylon screw is provided to prevent shorting a regulator to ground.

One of the niceties of the BYTESAVER™ is the capabilities provided by its software. When you purchase a BYTESAVER™ with one 2704 PROM, the Bytemover software is pre-programmed in that PROM. For your information and assistance, Cromemco has allowed INTERFACE AGE to publish the BYTEMOVER™ software. This program is included at the end of this article.

The PROM containing the Bytemover software is normally inserted into PROM location Zero on the Bytesaver board.

The Bytemover software can be used to program a PROM in any of the PROM locations on the Bytesaver board. The Bytemover software can also be used to transfer programs from PROM to RAM.

The operation of the Bytemover software is controlled by setting front panel sense switches on any S-100 bus compatible computer. However, to use the Bytemover software there must be at least one RAM board in the computer beginning at location Zero in the memory map. Furthermore, this RAM board must be unprotected for proper execution of the Bytemover software.

Software can be loaded into a 2704 or a 2708 in as small increments as you desire provided it is added to previously unused areas in that PROM.

This is done by first using Bytemover to move the current contents of the PROM down to RAM which corresponds to the unused portion of the PROM and finally using Bytemover again to reprogram the PROM with the new software.

Although the entire PROM must always be programmed, it never hurts to re-write the same information over again. And, of course, an erased PROM in which all bits are "1" may be programmed at any time.

In general, it is OK to write a "1" over a "1", a "0" over a "0", or a "0" over a "1", but to write a "1" over a "0" the PROM must be completely erased.

If the PROM to which you want to add software is PROM zero on the Bytesaver board, turn off AC power to the computer and install a 1N914 diode just below QO. Turn the power back on and move Bytemover to RAM zero. Add the new software to an area of RAM which corresponds to an unused portion of PROM zero.

Re-program PROM zero by following the example in the manual supplied. Note you need not erase the PROM to do this. Turn the computer power off and remove the 1N914 diode below QO.

The Bytesaver software supplied with the board is designed to program the PROM in approximately 30 seconds. This is generally a sufficiently long period of programming time. However, to be completely within the manufacturer's specifications, the PROM should be programmed for two to three minutes.

If you wish to program your PROMs for longer than 30 seconds, the Bytemover software may be easily modified. Simply change the contents of memory location 77H (Hex) from 40H to 00H. Now you must manually time the programming operation and depress the stop switch at the end of the operation.

Before using the Bytesaver, you must install three jumper wires to set the location of the Bytesaver in memory. This adjustment is shown in Figure 1. The assembled Bytesaver comes with A_{13}, A_{14}, and A_{15} each tied to the corresponding 'HI' pad to position the board at the very top of memory. In the following instructions it is assumed this is the jumper connection used.

1. Turn off all power to the computer and plug in the Bytesaver board.
2. Be sure the program power on the Bytesaver is turned off (program power switch in the down position).
3. Turn on the computer. Raise the reset switch, the stop switch and then raise the reset switch once again to initialize the computer.

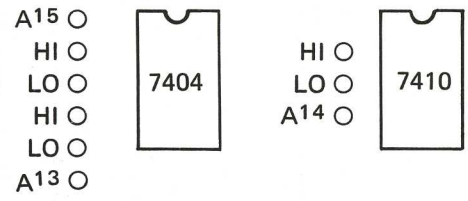


FIGURE 1. *This diagram illustrates the positioning of the Bytesaver address in memory. The preassembled Bytesaver comes with A^{15}, A^{14} and A^{13} connected to the corresponding HI terminals so memory address occurs when these lines are HI. Any or all of these address lines may be connected to the corresponding LO terminal to move the memory board lower in memory. Consequently, there are eight positions in which this board may be used.*

4. Raise the address switches A_{15}, A_{14}, and A_{13}. All other address switches should be in the down position.
5. Raise the examine switch. You are now examining the contents of the first byte of PROM in PROM location zero of the Bytesaver memory board (memory location 340 000). If the PROM supplied with your Bytesaver is in this PROM location, the data lights will read '061', the first byte of the Bytemover program.

The following examples show the steps needed to transfer and program the PROMs and the ease in which it is done.

EXAMPLE 1: Transfer the Bytemover Program from PROM to RAM beginning at RAM location Zero.

1. Raise the reset switch.
2. Depress the unprotect switch (on the Altair front panel).
3. Raise A_{15}, A_{14}, and A_{13}. Raise the examine switch. The data lights should read '061' octal.
4. Now set the sense switches for the task to be done, referring to Figure 2.
5. Push the run switch. In less than one second, the contents of PROM will be transferred to RAM. The contents of PROM are unaffected by this operation.
6. Raise the stop switch.
7. Raise the reset switch. Note that the data lights read '061.'

EXAMPLE 2: Program a 2708 PROM inserted in PROM location one. This PROM is to be programmed with the contents of the first 1K bytes of RAM beginning at location zero in memory. The Bytesaver software is still in the PROM installed in PROM location zero on the Bytesaver board.

A^{15}	Down	to transfer from Prom to Ram.
A^{14}	Down	for the transfer of 1K bytes.
A^{13}	Down	All down since we are transferring from the PROM that contains Bytemover (PROM 0).
A^{12}	Down	
A^{11}	Down	
A^{10}	Down	All down for storage to begin at location zero in RAM.
A^{9}	Down	
A^{8}	Down	

1. Raise the reset switch.
2. Depress the unprotect switch on the Altair front panel).
3. Raise A_{15}, A_{14}, and A_{13}. Raise the examine switch. The data lights should read '061' octal.
4. Raise the protect switch on the Bytesaver board (i.e. program power switch to the on position). The protect light on the front panel should turn off when this switch is raised.
5. Now set the sense switches for the task to be done.

A^{15}	Up	to program a PROM
A^{14}	Down	(always down for PROM programming).
A^{13}	Down	To select the PROM 1K higher in memory than the PROM that contains Bytemover
A^{12}	Down	
A^{11}	Up	
A^{10}	Down	All down for transfer to begin at location zero in RAM.
A^{9}	Down	
A^{8}	Down	

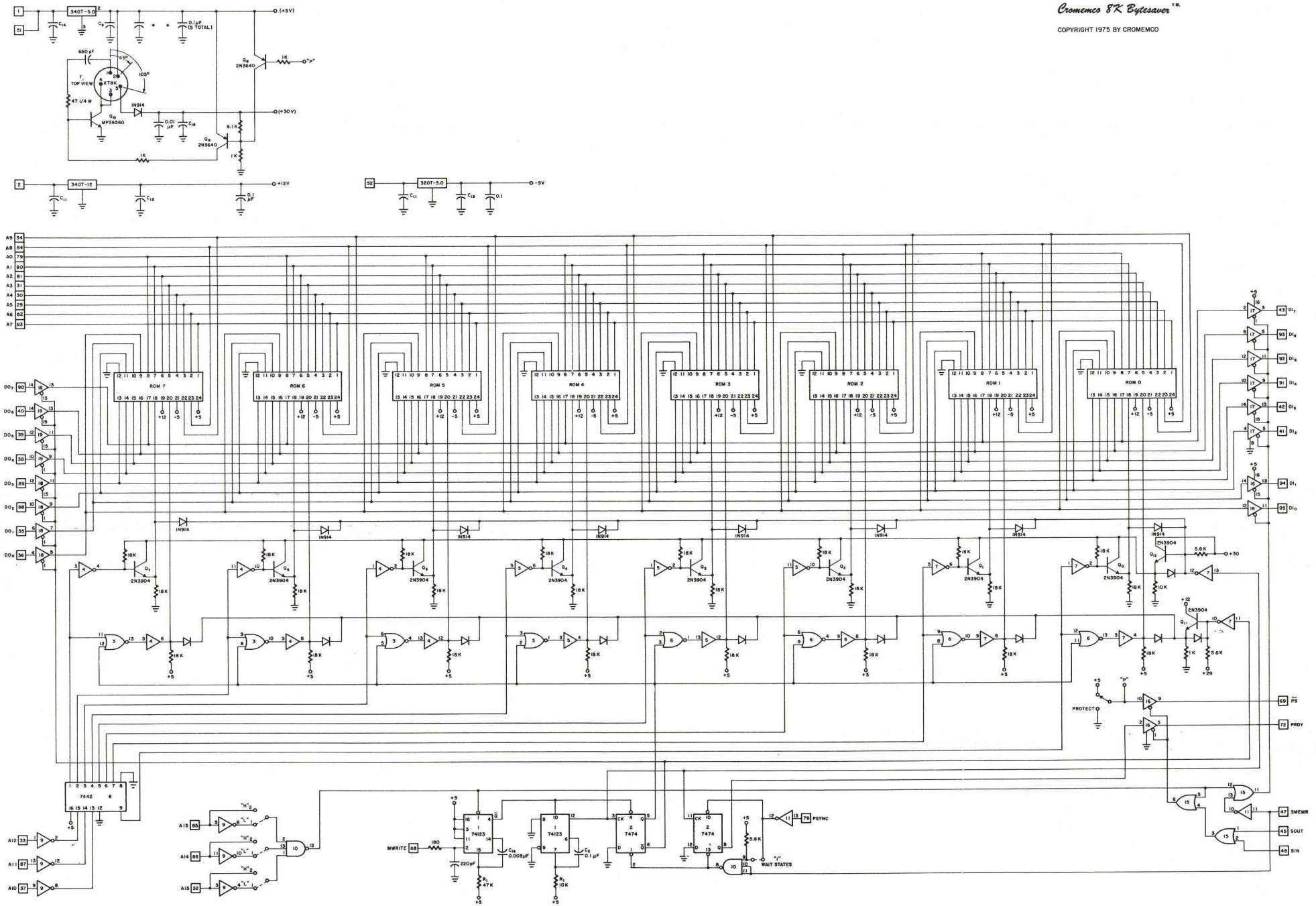
6. Push the run switch. Note that panel light A_9 is blinking at a rate of about twice per second. When this light stops blinking, the PROM programming is complete.

A^{15}	A^{14}	A^{13} A^{12} A^{11}	A^{10} A^{9} A^{8}
Up To program a PROM.	Up For a 7K transfer.	MSB LSB PROM address location in increments of 1K from the PROM in which BYTE	MSB LSB Selection of RAM address in 1K increments.
Down To move from PROM to RAM.	Down For a 1K transfer.		

FIGURE 2. FUNCTION OF THE SENSE SWITCHES IN BYTEMOVER.

BYTESAVER™ SCHEMATIC

Cromemco 8K Bytesaver™



7. Raise the stop switch.
8. Now note the INTE light on the front panel. If this light is on, the Bytemover Verifier has verified that the contents of the programmed PROM are indeed identical to the contents of the selected 1K bytes of RAM. If this light is off, the PROM is not programmed correctly. This could be due to a defective PROM.

EXAMPLE 3: Altair 8K BASIC can be stored in seven 2708 PROMs. Given that these seven PROMs are in PROM locations one through seven of the Bytesaver board, 8K BASIC can easily be transferred into RAM using the following procedure.

1. Raise the reset switch.
2. Depress the unprotect switch (on the Altair front panel).
3. Raise A_{15}, A_{14}, and A_{13}. Raise the examine switch. The data lights should read '061' octal.
4. Now set the sense switches for the task to be done.

A_{15}	Down	to transfer from PROM to RAM.
A_{14}	Up	for a 7K transfer.
A_{13}	Down	To begin transfer from the PROM 1K higher in memory than the Bytemover program.
A_{12}	Down	
A_{11}	Up	
A_{10}	Down	All down for storage to begin at location zero in RAM.
A_9	Down	
A_8	Down	

5. Push the run switch. In less than one second BASIC will be loaded into RAM (it sure beats paper tape!). Now raise the stop switch.

EXAMPLE 4: If you do not have Bytemover in PROM, you can program a PROM with Bytemover that is stored in RAM. The Bytemover software must be loaded into RAM beginning at location zero. The Bytemover software can then be burned into a PROM using the following procedure.

1. Raise the reset switch.
2. Depress the unprotect switch (on the Altair front panel).
3. Insert an erased PROM into PROM location zero.
4. Examine location 000 240 in memory.
5. Raise the program power switch on the Bytesaver board.
6. Set the sense switches with A_{15}, A_{14}, and A_{13} up.
7. Push the run switch. When the A_9 light stops blinking, the programming is complete. The INTE light will be on.
8. Turn off PROM program power by depressing the switch on the Bytesaver board.

That's about all there is to it to use a Bytesaver. It's both convenient and easy. The new instruction manual is a great improvement over the previous one. I would have liked to have a little more information on erasing PROMs. The PROM should be about one inch away and exposed for about twenty minutes. This assumes an integrated dosage of 10W-SECS/CM^2 is required. If shorter times or lower output devices are used, I would recommend placing the PROMs in the Bytesaver and running a memory check. All bits should be '1' for correct operation. PROM erasers are available from various computer stores around the country.

The Bytesaver is a tremendous adjunct to your computer capability (I have two, one with an assembler/monitor program and the other with BASIC residing in memory). If you expect to program a greater number of different PROMs you would be wise to replace the PROM sockets with low or zero insertion force types. Either that or place the PROM in carriers. This is to reduce the strain on their leads. The Bytesaver may be purchased with the Bytemover software already stored in either a 2704 or a 2708. By the way, the 2704 is just a 2708 with a bad bit in the upper 512 bytes. 2708 PROMs can also be purchased from Cromemco, but their price is a little higher than you can get elsewhere, but the convenience may be worthwhile. Anyway if you choose to buy it, I would heartily recommend considering the Cromemco Bytesaver as your next purchase.

```
0000                0000 * BYTEMOVER (T.M.) SOFTWARE FOR
0000                0001 * CROMEMCO 8K BYTESAVER (T.M.)
0000                0002 * VERSION 3. 1
0000                0003 * SELF-RELOCATING SOFTWARE LOCATABLE AT ANY
0000                0004 * 1024 BYTE (1K) BOUNDARY IN MEMORY
0000                0009 * ROUTINE TO FIND ONESELF IN MEMORY
0000                0010   SP EQU 6
0000                0019 * DEFINE FIRST 4 BYTES IN MEMORY AS STACK
0000 31 00 00       0020   LXI SP, 0
0003                0029 * SAVE FIRST FOUR BYTES IN REGISTERS
0003 C1             0030   POP B
0004 D1             0040   POP D
0005                0049 * REPLACE BYTE 0 WITH A 'RETURN'
0005 2E C9          0050   MVI L, 0C9H
0007 F3             0051   DI
0008 E5             0060   PUSH H
0009 E5             0070   PUSH H
000A 00             0080   NOP
000B 00             0081   NOP
000C 00             0082   NOP
000D 31 04 00       0090   LXI SP, 4
0010 CD 00 00       0100   CALL 0
0013                0101 * ROM LOCATION NOW IN BYTE 3
0013 31 02 00       0110   LXI SP, 2
0016 E1             0120   POP H
0017                0129 * RETURN BYTES 0-3
0017 31 04 00       0130   LXI SP, 4
001A D5             0140   PUSH D
001B C5             0150   PUSH B
001C                0159 * STORE ROM LOCATION IN SP
001C F9             0160   SPHL
001D 0E 00          0170   MVI C, 0
001F 59             0180   MOV E, C
0020 69             0190   MOV L, C
0021                0199 * INPUT SENSE SW COMMANDS
0021 DB FF          0200   IN 255
0023 57             0210   MOV D, A
0024                0219 * STRIP RAM ADDRESS
0024 E6 07          0220   ANI 7
0026 07             0230   RLC
0027 07             0240   RLC
0028                0249 * STORE RAM ADDRESS IN BC
0028 47             0250   MOV B, A
0029 7A             0260   MOV A, D
002A                0269 * STRIP ROM ADDRESS
002A E6 38          0270   ANI 56
002C 0F             0280   RRC
002D 00             0290   NOP
002E 67             0300   MOV H, A
002F 39             0310   DAD SP
0030 2E 00          0320   MVI L, 0
0032 7A             0330   MOV A, D
0033 EB             0340   XCHG
0034                0341 * ADDRESS OF ROM BEING PROCESSED IN DE
0034                0349 * BRANCH TO TRANSFER OR PROGRAM ROUTINE
```

```
0034 E6 80      0350   ANI 128
0036 0F         0360   RRC
0037 0F         0370   RRC
0038 C6 2D      0380   ADI 45
003A 21 00 00   0390   LXI H, 0
003D 6F         0400   MOV L, A
003E 39         0410   DAD SP
003F E9         0420   PCHL
0040            0500 * ROUTINE TO TRANSFER ROM TO RAM
0040 F9         0510   SPHL
0041 21 0B 00   0520   LXI H, 11
0044 39         0530   DAD SP
0045 EB         0550   XCHG
0046 F9         0560   SPHL  STACK CONTAINS ROM LOCATION
0047 EB         0570   XCHG  H&L CONTAIN LOOP ADDRESS
0048 11 00 00   0580   LXI D, 0
004B            0588 * START OF TRANSFER LOOP
004B            0589 * INCREMENT ROM ADDRESS
004B 3B         0590   DCX SP
004C            0599 * MOVE DATA FROM ROM TO RAM
004C F1         0600   POP 6
004D 02         0610   STAX B
004E            0619 * INCREMENT RAM ADDRESS
004E 03         0620   INX B
004F            0629 * INCREMENT BYTE COUNT
004F 13         0630   INX D
0050 7A         0640   MOV A, D
0051 E6 04      0650   ANI 4
0053 07         0660   RLC
0054 07         0670   RLC
0055 00         0680   NOP
0056 85         0690   ADD L
0057 6F         0070   MOV L, A
0058 E9         0710   PCHL
0059 00         0716   NOP
005A 00         0717   NOP
005B            0719 * JUMP TO 00B1 FROM TRANSFER ROUTINE
005B 3E 56      0720   MVI A, 56H
005D 85         0725   ADD L
005E 6F         0730   MOV L, A
005F E9         0740   PCHL
0060            1000 * ROUTINE TO PROGRAM ROM
0060 00         1010   NOP
0061            1019 * MOVE RAM ADDRESS INTO HL
0061 69         1020   MOV L, C
0062 7C         1030   MOV A, H
0063 60         1040   MOV H, B
0064            1049 * MOVE RAM ADDRESS INTO SP
0064 F9         1050   SPHL
0065 67         1060   MOV H, A
0066 2E 6B      1070   MVI L, 107
0068            1079 * INCREMENT RAM ADDRESS
0068 01 00 00   1080   LXI B, 0
006B            1089 * INCREMENT RAM ADDRESS
006B 3B         1090   DCX SP
006C            1098 * USE STAX AND POP 6 (PSW)
006C            1099 * TO MOVE DATA FROM ROM TO RAM
006C F1         1100   POP 6
006D 12         1110   STAX D
006E            1119 * INCREMENT ROM ADDRESS
006E 13         1120   INX D
006F            1129 * INCREMENT BYTE COUNT
006F 03         1130   INX B
0070            1138 * B STORES TWO CONSTANTS
0070            1139 * # COMPLETE PASSES & IN ROM CNT
0070 78         1140   MOV A, B
0071            1149 * # PASSES = 32 ?
0071 FE FC      1150   CPI 252
0073 3F         1160   CMC
0074 1F         1170   RAR
0075 1F         1180   RAR
0076            1198 * SET 64 TO 0 FOR TWO MINUTE TIMER VERSION
0076 E6 40      1200   ANI 64
0078            1201 * A=64 IF COMPLETED 32 PASSES
0078 2E 7D      1205   MVI L, 7DH
007A 85         1210   ADD L
007B 6F         1220   MOV L, A
007C E9         1225   PCHL
007D 2E 6B      1226   MVI L, 6BH
007F 78         1230   MOV A, B
0080 E6 04      1240   ANI 4
0082            1241 * A=4 IF END OF 1024 BYTE PASS
0082 07         1250   RLC
0083 07         1260   RLC
0084 07         1270   RLC
0085 85         1280   ADD L
0086 6F         1290   MOV L, A
0087            1291 * GO BACK TO 1090 UNLESS OVERFLOW
0087            1292 * THEN GO TO 1380 FOR
0087            1293 * ADDRESS SUBTRACTION
0087            1294 * OR 2135 FOR QUITS
0087 E9         1300   PCHL
```

```
0088 00          1350  NOP
0089 00          1360  NOP
008A 00          1370  NOP
008B             1378 * ANOTHER PROGRAM PASS TO BE DONE
008B             1379 * ADJUST ROM AND RAM ADDRESSES
008B 7C          1380  MOV A, H
008C 21 00 FC    1390  LXI H, 64512
008F             1399 * SUBTRACT 1024 FROM ROM ADDRESS
008F 39          1400  DAD SP
0090 F9          1410  SPHL
0091 21 00 FC    1420  LXI H, 64512
0094             1429 * SUBTRACT 1024 FROM RAM  ADDRESS
0094 19          1430  DAD D
0095 EB          1440  XCHG
0096 67          1450  MOV H, A
0097 2E 6B       1460  MVI L, 107
0099 78          1470  MOV A, B
009A E6 F8       1480  ANI 248
009C             1489 * INCREMENT PASS COUNTER BY ONE
009C C6 08       1490  ADI 8
009E 47          1495  MOV B, A
009F             1499 * GO BACK TO 1090
009F E9          1500  PCHL
00A0             2000 * ROUTINE TO LOAD BYTEMOVER INTO ROM
00A0 DB FF       2010  IN 255
00A2 47          2020  MOV B, A
00A3 E6 E0       2030  ANI 224
00A5 1E 00       2040  MVI E, 0
00A7 4B          2050  MOV C, E
00A8 57          2060  MOV D, A
00A9 78          2070  MOV A, B
00AA E6 1F       2080  ANI 31
00AC 47          2090  MOV B, A
00AD 67          2100  MOV H, A
00AE 2E 60       2110  MVI L, 96
00B0 E9          2120  PCHL
00B1             2121 * CHECK FOR 7K TRANSFER OF ROM TO RAM
00B1 C6 1A       2122  ADI 1AH
00B3 6F          2123  MOV L, A
00B4 DB FF       2124  IN 255
00B6 E6 40       2125  ANI 64
00B8 0F          2126  RRC
00B9 0F          2127  RRC
00BA 85          2128  ADD L
00BB 6F          2129  MOV L, A
00BC E9          2130  PCHL
00BD             2133 * PROGRAMMER VERIFICATION ROUTINE
00BD             2134 * PART 1
00BD 7C          2135  MOV A, H
00BE 21 00 FC    2145  LXI H, 64512
00C1 39          2155  DAD SP
00C2 F9          2165  SPHL
00C3 2E CD       2175  MVI L, 0CDH
00C5 67          2185  MOV H, A
00C6 E9          2195  PCHL
00C7 00          2205  NOP
00C8 00          2210  NOP
00C9 00          2215  NOP
00CA 00          2220  NOP
00CB             2229 * ROM TO RAM TRANSFER STOP ROUTINE
00CB FB          2230  EI
00CC E9          2240  PCHL
00CD             2248 * PROGRAMMER VERIFICATION ROUTINE
00CD             2249 * PART 2
00CD 7C          2250  MOV A, H
00CE 21 00 FC    2260  LXI H, 64512
00D1 19          2270  DAD D
00D2 EB          2280  XCHG
00D3 2E F1       2290  MVI L, 0F1H
00D5 67          2300  MOV H, A
00D6 01 00 00    2310  LXI B, 0
00D9 E9          2320  PCHL
00DA 00          2625  NOP
00DB             2629 * 7K TRANSFER COMPLETION CHECK
00DB D6 90       2630  SUI 90H
00DD 6F          2640  MOV L, A
00DE 7A          2650  MOV A, D
00DF C6 04       2660  ADI 4
00E1 57          2670  MOV D, A
00E2 FE 38       2680  CPI 56
00E4 3F          2685  CMC
00E5 3E 00       2690  MVI A, 0
00E7 1F          2700  RAR
00E8 85          2710  ADD L
00E9 6F          2720  MOV L, A
00EA E9          2730  PCHL
00EB             2879 * ROM PROGRAMMER STOP ROUTINE
00EB 00          2880  NOP
00EC 00          2881  NOP
00ED FB          2885  EI
00EE E9          2890  PCHL
00EF E9          2900  PCHL
```

```
00F0 E9     2906   PCHL
00F1        2918 * PROGRAMMER VERIFICATION ROUTINE
00F1        2919 * PART 3
00F1 3B     2920   DCX SP
00F2 F1     2930   POP 6
00F3 EB     2940   XCHG
00F4        2949 * COMPARE FOR GREATER
00F4 BE     2950   CMP M
00F5 EB     2960   XCHG
00F6 17     2970   RAL
00F7 E6 01  3000   ANI 1
00F9 2F     3010   CMA
00FA 3C     3011   INR A
00FB 85     3015   ADD L
00FC 6F     3020   MOV L, A
00FD 3B     3030   DCX SP
00FE 3B     3040   DCX SP
00FF        3050 * COMPARE FOR LESSER
00FF F1     3055   POP 6
0100 2F     3056   CMA
0101 EB     3058   XCHG
0102 86     3059   ADD M
0103 EB     3060   XCHG
0104 C6 07  3061   ADI A, 1
0106 3F     3065   CMC
0107 17     3070   RAL
0108 E6 01  3090   ANI 1
010A 2F     3100   CMA
010B 3C     3101   INR A
010C 85     3105   ADD L
010D 6F     3110   MOV L, A
010E 03     3130   INX B
010F 13     3140   INX D
0110 78     3150   MOV A, B
0111 E6 04  3180   ANI 4
0113 2F     3190   CMA
0114 3C     3191   INR A
0115 85     3195   ADD L
0116 6F     3200   MOV L, A
0117 E9     3210   PCHL
```


LET'S HEAR FROM YOU

Your letters are the lifeblood for us

We really do appreciate your letters and comments about INTERFACE AGE. We need to know what you think about the publication so that we know in what ways we can make it better.

If you like it, tell us, and if there are things you think could be improved, tell us that, too.

We do read all our mail and we appreciate your feedback.

The Editors