INTRODUCTION

Thank you for purchasing a Micro Works 2708 EPROM Programmer. Every effort has been made in the development of the B-08 to provide you with a long lasting, trouble free computer accessory. We suggest that you read this manual thoroughly before installing the B-08.

The Micro Works B-08 is a compact 2708 EPROM programmer that fits in a standard SWTPC 6800 I/0 slot. A safety switch and LED indicator provide control over the high programming voltage generated on the board. Your B-08, combined with the Micro Works 2708 Utility, provides full capability to program and copy 2708s efficiently and reliably.

The Micro Works is certain you will find your computer system more versatile with the use of the EPROM Programmer. We look forward to hearing any suggestions or comments from our customers.

UNPACKING AND INSTALLATION

Carefully remove the B-08 from the box and unwrap the packing material. Take time to inspect the PC board for any damage which may have been incurred in shipping. If there is any damage, save all packing materials and notify the carrier immediately.

Your B-08 contains MOS integrated circuitry which may easily be damaged by static electrical sources. Avoid over-handling and do not allow anything to come into contact with the conductors on the board. Never lift the board out of, or plug it into, a computer which is turned on.

Important: If your system has a +19 volt power supply, such as the Smoke Signal Broadcasting PS-1, make sure that your B-08 has a 7812 voltage regulator installed with a heat sink before plugging it in. If this optional voltage regulator has not been installed, a modification kit is available from The Micro Works for $5.00 plus postage and handling.

Scanned and edited by Michael Holley May 6, 2002
The Micro Works Document Circa 1978

We urge you to make sure that your SWTPC 6800 system has been completely tested before you install the B-08. You should at least ascertain that the CPU board operates properly with the on board RAM, in accordance with the SWTPC instruction manual.

For compatibility with the Micro Works 2708 Utility, your B-08 must be installed in I/0 slot 4 of the SWTPC mother board. Be sure that the power is off during installation.

U2708 PROGRAMMING HARDWARE

The 2708 EPROM is a 1024 word by 8 bit ultraviolet erasable programmable read only memory. The hardware required to read and program these memories is located on a single card designed to program, read and copy 2708 EPROMs with minimum effort. The EPROMs are erased by exposure to short wave UV light (2537 Angstroms) with an exposure of 10 watt-sec. per square cm. When erased, all locations contain ones. User selected bits are set to zero by application of a series of high voltage programming pulses, generated on the programmer card and timed by routines in software. Read the 2708 manufacturer's data sheet for more detailed timing information.

The 1K by 8 organization of the EPROMs requires ten address lines, in addition to three lines for programmer control. In order to utilize a single PIA, the two high order address and two of the control lines are multiplexed with the eight low order address lines using the A side of the PIA. The high order address and control signals are buffered by latches on the programmer card. Strobe pulses for the latches and signals to control the high voltage programming pulses are provided by the PIA's CA2 and CB2 lines, respectively. All data transfers take place through the B side of the PIA.

U2708 UTILITY SOFTWARE

The Micro Works provides three standard versions of the utility package, available either on 2708 EPROM or Kansas City standard cassette tape. The two EPROM versions have origins at $C000_{16}$ and $FC00_{16}$ and are designed to run with MIKBUG and SMARTBUG, respectively. The cassette tape version begins at 100016 and runs from RAM, calling several routines in MIKBUG.

STANDARD VERSION - U2708/$C000_{16}$

This program provides the user with a variety of functions useful for burning, verifying and reading the contents of 2708 EPROMs. The program is written in M6800 assembly language, and is supplied on a single 2708 EPROM. An image buffer of 1K bytes of RAM, starting at 0000_{16}, is also required. U2708 was originally developed for the SWTPC 6800 system, and as such, uses several of the subroutines found in the MIKBUG ROM. It expects to find the PROM burner PIA at locations 8010_{16} - 8013_{16}, but these vectors may be easily changed and the PIA moved elsewhere. The tape load routine uses the MP-C card in slot 1 and issues control characters for an AC-30/CT-1024 tape, interface.

When entered, 02708 prompts the user with "The Micro Works 2708 Utility", and awaits a command. Permissible commands are: BURN, VERIFY, ERASE TEST, MOVE, SET, XFER, LOAD, and MIKBUG. The user signifies the selected command by typing the first character of the command, or in the case of MIKBUG, typing a "*". The program responds by requesting further information where necessary, and executing the command. On completion of the command, except in the case of a MIKBUG call, control is returned to the utility and another command may be entered.

BURN COMMAND - "B"

The BURN command programs the EPROM with the data contained in the image buffer, located in computer RAM locations 0000_{16} - $03FF_{16}$. On completion of the BURN attempt, the contents of the EPROM are read and compared with the image buffer. If any errors are found, they are classified as "hard" or "soft" errors and displayed on the user's display device. Hard errors are signified by the letter "H" appearing in the "HS" position on the error list.

A hard error is one which cannot possibly be corrected by the bigger hammer and hit-it-three-times approach used by the BURN routine, to wit: if the EPROM has a bit, after programming, that should be high but is set low, we are out of luck. There is no choice but to re-erase and start over, since bits may only be set high by application of UV light. However, if an EPROM bit remained high when it should have gone low, maybe the power supply rippled at an inopportune millisecond or the EPROM was cold. Whatever, it is possible to recover from this type of "soft" error.

If hard errors are found, programming ceases and the user is notified of his/her plight. If only soft errors are found, another attempt to program the EPROM is made, up to a maximum of three tries. If soft errors persist after three tries, it is assumed that the EPROM is defective and, having notified the user, the burner gives up. If no errors are found, the EPROM is given a gold star and control is returned to the user.

VERIFY COMMAND - "V"

The verify command causes the EPROM to be read and compared with the data in the image buffer. A list of discrepancies, each one classified as hard or soft, is displayed for the user. This operation is a subroutine, also called by the BURN command.

ERASE TEST COMMAND - "E"

The ERASE TEST routine is used to verify the virginity of a supposedly erased EPROM. (N.B. It does not erase the PROM; UV light does that.) A well erased EPROM will contain FF16 in every location. Since EPROMs do age with repeated re-programming, it is wise to use this routine before attempting to teach new tricks to old dogs. An error list similar to the one produced by the BURN and VERIFY commands will be displayed, then control will be returned to the user. Note that all errors uncovered by this routine will be hard errors, and will be displayed as such. Again, re-erasure is the only revenge.

SET COMMAND - "S"

The SET command allows the user to set blocks of memory to a hex value input from the control console. When the command is executed, the routine requests that the user input two 4-digit hex numbers for the starting and ending memory addresses and a 2-digit hex data byte. On receipt of this information, the routine proceeds to set all memory locations from starting address to ending address, inclusive, to the data value. This feature is useful when only portions of the EPROM are to be programmed; other locations will be reserved for later expansion. In this case, the unused EPROM locations should be programmed to the "don't care" state, HFF. The image buffer may be set to all HFFs before overwriting a portion of it with the desired program. This command is also useful in implementing the "Halt and Clear Core" (HCC) instruction in systems not already possessing this capability.

MOVE COMMAND -"M"

The MOVE command, when executed, asks the user for three 4-digit hex addresses: the starting and ending addresses of the source block (data to be moved) and the starting address of the destination block (where to put it). It then proceeds to move the data contained in memory from source starting address to ending address, inclusive, to memory, starting at destination address. The word count is calculated from the difference in source addresses. This routine allows the user to move programs from the RAM area where they were debugged to the image buffer for programming into EPROM. As noted before, data blocks need not be a full 1024 words in length, thus conserving EPROM for future expansion.

XFER COMMAND - "X"

The transfer (XFER) command reads the data contained in the EPROM and transfers it to the image buffer located at 0000_{16} - $03FF_{16}$. This command is most useful in copying existing EPROMs. Simply place the EPROM to be copied in the programmer socket, execute "XFER", replace this EPROM with an erased one, and burn away.

LOAD COMMAND - "L"

The LOAD command allows the user to read MIKBUG format object tapes into the image buffer by specifying an offset value. The LOAD routine is equivalent to "load" in MIKBUG except for the subtraction of the offset value from the destination address on the tape. Thus, a tape assembled to start at C00016 will be loaded starting at 0000_{16} if the offset of $C000_{16}$ is entered in response to the "addr" query by U2708. The routine also asks for the length of the object program, in K, and uses this information to prevent clobbering areas of memory above and below the image buffer. Programs longer than 1K may be loaded in one pass; the MOVE command is then used to transfer upper 1K blocks to the image buffer for burning into EPROM.

MIKBUG COMMAND - "*"

The MIKBUG command is really a monitor call; it returns control to the user's operating system. Memory examination and modification functions are assumed to exist in the simplest of user operating systems, so they are not included in this program. They are the additional tools the user will need to edit the contents of the image buffer.

B-08 PARTS LIST

INTEGRATED CIRCUITS	
U1	6821 PIA
U2	2708 Socket
U3 and U4	74LS74 Dual Flip-flops
U5	7406 Hex Inverter
06	555 Timing Circuits
U7	7805 +5 Voltage Regulator
U8	7905 -5 Voltage Regulator
U9	7812 +12 Voltage Regulator (Optional)
RESISTORS	
R1, R3, and R7	4.7K
R2	470
R4	47K
R5	10K
R6	270
CAPACITORS	
C1, C2, C3, C8 and C10	22/35V Electrolytic
C4	.001 Mfd. Ceramic
C5, C7, C9, C11 and C12	.1 Mfd. Ceramic
C6	100/16V Electrolytic
DIODES	
CR1 through CR5	1N914
CR6	LED
TRANSISTORS	
Q1	2N3904

LIMITED WARRANTY

The Micro Works warrants its products to be free from defects in workmanship and materials for a period of ninety (90) days from the date of purchase. IT IS EXPRESSLY AGREED THAT THIS NINETY (90) DAY WARRANTY SHALL BE IN LIEU OF OTHER EXPRESS WARRANTIES, WARRANTIES OF FITNESS AND IN LIEU OF THE WARRANTY OF MERCHANTABILITY. No agent, representative, or employee of the Company has authority to increase or alter the obligation of this warranty.

This warranty shall not apply to any Micro Works product which has been modified, repaired or altered in any way. This warranty shall not apply to any product damaged as a result of abuse, misuse, accident or neglect. In no event shall The Micro Works be liable for consequential damages.

In order to make a claim against this warranty the defective board must be returned by private carrier or the U.S Postal Service to THE MICRO WORKS, P.O. BOX 1110, DEL MAR, CALIFORNIA, 92014. Boards must be accompanied by return shipping charges and the sales receipt showing date of purchase. It is suggested that boards shipped through the United States mails be insured.

REPAIRS

At any time after the expiration of the 90 day warranty period, The Micro Works will repair your PC board for a fee of $25.00, provided that the board is not physically damaged, and that not more than two chips require replacement. If the flat fee is not applicable, you will be notified before further repairs are made. If repairs are necessary, repack the board carefully and enclose a check to The Micro Works, P.O. Box 1110, Del Mar, CA, 92014.

```
                ****************************
                *     THE MICRO WORKS      *
                *     PROM UTILITY         *
                *     U2708/C000           *
                *                          *
                *     C. 1977 V1.0         *
                ****************************

                        NAM    U2708/C000
                        OPT    NOG
C000                    ORG    $C000

C000 8E A0 42  U2708    LDS    #$A042    GET STACKED
C003 86 38              LDA A  #$38      A SIDE DDR
C005 B7 80 11           STA A  PIACRA
C008 86 FF              LDA A  #$FF      TO OUTPUTS
C00A B7 80 10           STA A  PIAADR
C00D 86 3C              LDA A  #$3C      A SIDE DATA REC
C00F B7 80 11           STA A  PIACRA
C012 B7 80 13           STA A  PIACRB    B SIDE DATA REC
C015 86 04              LDA A  #$04      PROM TO READ
C017 B7 A0 52           STA A  CW
C01A BD C1 4F           JSR    DISCH     ADDR & CTL WORD
C01D 7E C1 01           JMP    CNTRL
C020 86 30     BURN     LDA A  #$30      ASCII #
C022 B7 A0 50           STA A  T         FOR TRY COUNTER
C025 CE C2 C5           LDX    #HVON     TURN ON SWITCH
C028 BD C1 42           JSR    STRING    TYPE SUMTHING
C02B BD E1 AC           JSR    INEEE     TO CONFIRM
C02E 7C A0 50  BLOOP3   INC    T         NEXT TRY
C031 CE C2 CC           LDX    #BHDR     PRINT HDR
C034 BD E0 7E           JSR    PDATA1
C037 B6 A0 50           LDA A  T         PRINT TRY COUNT
C03A BD E1 D1           JSR    OUTEE
C03D BD C1 45           JSR    CR
C040 86 08              LDA A  #$08      WE=12V
C042 B7 A0 52           STA A  CW        PROM TO WRITE
C045 C6 FF              LDA B  #$FF      PIA FOR OUTPUT
C047 8D 49              BSR    PCTLB     ALSO INITS PCNTR
C049 BD C1 4F  BLOOP2   JSR    DISCH     INIT ADRS
C04C A6 00     BLOOP1   LDA A  0,X       GET RAM DATA
C04E B7 80 12           STA A  PIADAT    PUT TO PROM
C051 86 34              LDA A  #$34      ZAP
C053 B7 80 13           STA A  PIACRB
C056 8D 48              BSR    DELAY     WAIT
C058 8D 40              BSR    PCTL1     ZAP OFF
C05A 8D 78              BSR    NEXT      NEXT ADR
C05C 24 EE              BCC    BLOOP1    END OF PASS?
C05E 5A                 DEC B            PASS=PASS-1
C05F 26 E8              BNE    BLOOP2    NEXT PASS
C061 8D 2F              BSR    PCTLB     PIA TO READ
C063 86 04              LDA A  #$04      PROM TO READ
C065 B7 A0 52           STA A  CW
C068 BD C1 65           JSR    VERIFY    CHECK IT OUT
C06B 7D A0 54           TST    E         ANY GOOFS?
C06E 27 17              BEQ    JMPC      NO, SPLIT
C070 7D A0 53           TST    H         HARD ERRORS?
C073 26 0C              BNE    ERRH
C075 B6 A0 50           LDA A  T
C078 81 33              CMP A  #$33      THREE TRIES
```

```
C07A 26 B2            BNE    BLOOP3    TRY HARDER
C07C CE C3 A1  ERRT   LDX    #TERR     TOO MANY TRIES
C07F 20 03            BRA    ERP       TOO BAD
C081 CE C3 B3  ERRH   LDX    #HERR     HARD ERRORS
C084 BD C1 42  ERP    JSR    STRING    PRINT UH OH
C087 CE C3 0F  JMPC   LDX    #HVOFF    TURN OFF ZAPPER
C08A BD C1 42         JSR    STRING    TYPE ANYTHING
C08D BD E1 AC         JSR    INEEE     TO CONFIRM
C090 20 6F            BRA    CNTRL     AND DIE

C092 86 38     PCTLB  LDA A  #$38      STUFF DDRB
C094 B7 80 13         STA A  PIACRB    WITH WHATEVER'S
C097 F7 80 12         STA B  PIADAT    IN ACC B
C09A 86 3C     PCTL1  LDA A  #$3C
C09C B7 80 13         STA A  PIACRB
C09F 39               RTS

C0A0 86 43     DELAY  LDA A  #$43      500 USEC DELAY
C0A2 4A        DLOOP  DEC A            (FOR 1 MHZ CLOCK)
C0A3 26 FD            BNE    DLOOP
C0A5 39               RTS

C0A6 B7 A0 58  CHECK  STA A  ROMDAT    SAVE TESTEE
C0A9 B8 A0 57         EOR A  RAMDAT    =TESTOR?
C0AC 27 26            BEQ    NEXT      ATTABOY
C0AE B4 A0 57         AND A  RAMDAT    HARD ERROR
C0B1 B7 A0 54         STA A  E         H FLAG FLAG
C0B4 27 03            BEQ    RECOV     WHEW
C0B6 B7 A0 53         STA A  H         HARD ERR FLAG
C0B9 CE A0 55  RECOV  LDX    #RAMAD    PRINT ERROR
C0BC BD E0 C8         JSR    OUT4HS    INFO FOR USERS
C0BF BD E0 CA         JSR    OUT2HS
C0C2 BD E0 CA         JSR    OUT2HS
C0C5 CE C3 DC         LDX    #CRLF     FOXY TRICK
C0C8 7D A0 54         TST    E         TO FLAG HARD
C0CB 27 01            BEQ    OUT       ERRORS W/H
C0CD 09               DEX              GET IT?
C0CE BD E0 7E  OUT    JSR    PDATA1    NO?
C0D1 B7 A0 54         STA A  E         BOO FLAG
C0D4 FE A0 55  NEXT   LDX    RAMAD     INC ADDRESS
C0D7 08               INX
C0D8 0C               CLC              SEE IF END
C0D9 BC A0 59         CPX    EA        OF BUFFER
C0DC 26 01            BNE    PUTAD
C0DE 0D               SEC              CARRY SET IF SO
C0DF FF A0 55  PUTAD  STX    RAMAD
C0E2 B6 A0 55         LDA A  RAMAD     MSB
C0E5 84 03            AND A  #$03
C0E7 BA A0 52         ORA A  CW        AND CONTROL
C0EA B7 80 10         STA A  PIAADR    TO LATCHES
C0ED 86 34            LDA A  #$34      STROBE 'EM
C0EF B7 80 11         STA A  PIACRA
C0F2 86 3C            LDA A  #$3C      WITH CA2
C0F4 B7 80 11         STA A  PIACRA
C0F7 B6 A0 56         LDA A  RAMAD+1   LSB TO PIA
C0FA B7 80 10         STA A  PIAADR
C0FD FE A0 55         LDX    RAMAD
C100 39               RTS

C101 8D 42     CNTRL  BSR    CR        CONTROL HANDLER
```

```
C103 8D 40              BSR    CR         FEED SOME LINES
C105 CE C2 A5           LDX    #IHDR      WAKE UP USER
C108 8D 38              BSR    STRING
C10A BD E1 AC           JSR    INEEE      GET USER WHIM
C10D 16                 TAB
C10E 8D 35              BSR    CR
C110 CE C1 26           LDX    #TABLE     LOOK UP COMMAND
C113 E1 00      CMDLP   CMP B  0,X        IN TABLE
C115 27 0A              BEQ    POUNCE
C117 08                 INX               NEXT ENTRY
C118 08                 INX
C119 08                 INX
C11A 8C C1 3E           CPX    #TABEND    IS ALL?
C11D 27 E2              BEQ    CNTRL      YES
C11F 20 F2              BRA    CMDLP      NO
C121 08         POUNCE  INX               CLIMB ONTO ADDR
C122 EE 00              LDX    0,X        GET IT
C124 6E 00              JMP    0,X        AND GO

C126 42         TABLE   FCC    'B'
C127 C0 20              FDB    BURN
C129 53                 FCC    'S'
C12A C1 C6              FDB    SET
C12C 4D                 FCC    'M'
C12D C1 9C              FDB    MOVE
C12F 58                 FCC    'X'
C130 C1 E6              FDB    XFER
C132 4C                 FCC    'L'
C133 C2 0E              FDB    LOAD
C135 2A                 FCC    '*'
C136 E0 E3              FDB    MIKBUG
C138 45                 FCC    'E'
C139 C1 80              FDB    ERASE
C13B 56                 FCC    'V'
C13C C1 3E              FDB    TABEND
C13E 8D 25      TABEND  BSR    VERIFY
C140 20 BF              BRA    CNTRL

C142 BD E0 7E   STRING  JSR    PDATA1     PRINT STRING
C145 CE C3 DC   CR      LDX    #CRLF      AND CRLF
C148 BD E0 7E           JSR    PDATA1     BIG DEAL
C14B FE A0 55           LDX    RAMAD
C14E 39                 RTS

C14F CE 04 00   DISCH   LDX    #$0400     READ 03FF
C152 FF A0 59           STX    EA         (RESTORE END)
C155 09                 DEX
C156 BD C0 DF           JSR    PUTAD      TO DISCHARGE
C159 BD C0 A0           JSR    DELAY      1/2 OF BUSSES
C15C CE 00 00           LDX    #$0000     THEN DO OTHER
C15F FF A0 53           STX    H          (CLEAR ERRORS)
C162 7E C0 DF           JMP    PUTAD      HALF OF BUSSES

C165 8D E8      VERIFY  BSR    DISCH      EXERCISE FOR
C167 CE C3 17           LDX    #VHDR      THE READER
C16A 8D D6              BSR    STRING
C16C CE C3 1E           LDX    #VHDR1
C16F 8D D1              BSR    STRING
C171 A6 00      VLOOP   LDA A  0,X
C173 B7 A0 57           STA A  RAMDAT
```

```
C176 B6 80 12          LDA A  PIADAT
C179 BD C0 A6          JSR    CHECK
C17C 24 F3             BCC    VLOOP
C17E 20 C5             BRA    CR

C180 8D CD     ERASE   BSR    DISCH     CHECK PROM FOR
C182 CE C3 2C          LDX    #EHDR     VIRGINITY
C185 8D BB             BSR    STRING    BEFORE BURNING
C187 CE C3 39          LDX    #EHDR1
C18A 8D B6             BSR    STRING
C18C C6 FF             LDA B  #$FF
C18E F7 A0 57          STA B  RAMDAT
C191 B6 80 12  ELOOP   LDA A  PIADAT
C194 BD C0 A6          JSR    CHECK
C197 24 F8             BCC    ELOOP
C199 7E C1 01          JMP    CNTRL

C19C CE C3 47  MOVE    LDX    #MHDR     SHUFFES RANDOM
C19F BD C1 42          JSR    STRING
C1A2 CE C3 52          LDX    #MHDR1
C1A5 8D 55             BSR    TWOADS    BLOCKS OF STUFF
C1A7 08                INX
C1A8 FF A0 59          STX    EA        AROUND IN RAM
C1AB 8D 5B             BSR    ONEADS
C1AD FF A0 5B          STX    DA
C1B0 FE A0 55          LDX    RAMAD
C1B3 A6 00     MLOOP   LDA A  0,X
C1B5 FE A0 5B          LDX    DA
C1B8 A7 00             STA A  0,X
C1BA 08                INX
C1BB FF A0 5B          STX    DA
C1BE BD C0 D4          JSR    NEXT
C1C1 24 F0             BCC    MLOOP
C1C3 7E C1 01          JMP    CNTRL

C1C6 CE C3 5F  SET     LDX    #SHDR     SET BLOCKS OF
C1C9 BD C1 42          JSR    STRING
C1CC CE C3 6A          LDX    #SHDR1
C1CF 8D 2B             BSR    TWOADS    RAM TO INPUT
C1D1 08                INX
C1D2 FF A0 59          STX    EA        VALUE
C1D5 BD E0 55          JSR    BYTE
C1D8 16                TAB
C1D9 FE A0 55          LDX    RAMAD
C1DC E7 00     SLOOP   STA B  0,X
C1DE BD C0 D4          JSR    NEXT
C1E1 24 F9             BCC    SLOOP
C1E3 7E C1 01          JMP    CNTRL

C1E6 BD C1 4F  XFER    JSR    DISCH     TRANSFERS FROM
C1E9 CE C3 77          LDX    #XHDR     CONTENTS TO
C1EC BD C1 42          JSR    STRING    RAM 0000-3FFF
C1EF B6 80 12  XLOOP   LDA A  PIADAT
C1F2 A7 00             STA A  0,X
C1F4 BD C0 D4          JSR    NEXT
C1F7 24 F6             BCC    XLOOP
C1F9 7E C1 01          JMP    CNTRL

C1FC BD C1 42  TWOADS  JSR    STRING    GET 2 2 BYTES ADS
C1FF BD E0 47          JSR    BADDR     ONE FOR RAMAD
```

```
C202 FF A0 55          STX    RAMAD
C205 BD E0 CC          JSR    OUTS      (PRINT A SPACE)
C208 BD E0 47  ONEADS  JSR    BADDR     AND ONE FOR X
C20B 7E E0 CC          JMP    OUTS      'NOTHER SPACE
C20E CE C3 8C  LOAD    LDX    #LHDR     MIKBUG TAPE READER
C211 BD C1 42          JSR    STRING    WITH OFFSET
C214 CE C3 9A          LDX    #LHDR1    ADDR PROMPT
C217 BD C1 42          JSR    STRING
C21A BD E0 47          JSR    BADDR     GET PGM START ADR
C21D FF A0 5D          STX    OFFSET    AND SAVE IT
C220 BD E0 CC          JSR    OUTS      SPACE OUT
C223 BD E0 AA          JSR    INHEX     GETS LENGTH
C226 84 0F             AND A  #$0F      MASK OFF GARBAGE
C228 48                ASL A            TIMES 4
C229 48                ASL A            TO CONVERT
C22A B7 A0 51          STA A  LEN       AND PUT AWAY
C22D BD C1 45          JSR    CR        FEED A LINE
                 * SMARTBUG PATCH GOES HERE
C230 86 3C             LDA A  #$3C      READER ON *SB PATCH INC ECHO
C232 B7 80 07          STA A  PIASB     *SB PATCH NOP NOP
C235 86 11             LDA A  #$11
C237 BD E1 D1          JSR    OUTEE
C23A BD E1 AC  LOAD3   JSR    INEEE     READ CHARACTER
C23D 81 53             CMP A  #'S'      TEST FOR VALID
C23F 26 F9             BNE    LOAD3
C241 BD E1 AC          JSR    INEEE     RECORD OR END
C244 81 39             CMP A  #'9'      OF TAPE
C246 27 50             BEQ    LOAD21    END OF TAPE = S9
C248 81 31             CMP A  #'1'      RECORD = S1
C24A 26 EE             BNE    LOAD3
C24C 7F A0 60          CLR    CKSUM     INT CHECKSUM
C24F BD E0 55          JSR    BYTE      GET WORD COUNT
C252 80 02             SUB A  #2
C254 B7 A0 5F          STA A  BYTCT
C257 BD E0 47          JSR    BADDR     GET TAPE ADDR
C25A F6 A0 0D          LDA B  XLOW
C25D B6 A0 0C          LDA A  XHI
C260 F0 A0 5E          SUB B  OFFSET+1  SUBTRACT OFFSET
C263 B2 A0 5D          SBC A  OFFSET    TO PUT CODE
C266 25 20             BCS    OFERR     IN PROM BUFFER
C268 F7 A0 0D          STA B  XLOW      AND TEST FOR
C26B B7 A0 0C          STA A  XHI       WRAP AROUND TO
C26E FE A0 0C          LDX    XHI       HIGH MEMORY
C271 BD E0 55  LOAD11  JSR    BYTE      READ DATA RECORD
C274 7A A0 5F          DEC    BYTCT
C277 27 14             BEQ    LOAD15    'TILL END
C279 A7 00             STA A  0,X       STORING IT AND
C27B 08                INX              TESTING FOR
C27C FF A0 0C          STX    XHI       END OF BUFFER
C27F B6 A0 0C          LDA A  XHI       TO AVOID CLOBBER
C282 4A                DEC A            (FIX A SINCE INX)
C283 B1 A0 51          CMP A  LEN       OF PROTECTED AREA
C286 2D E9             BLT    LOAD11
C288 CE C3 BF  OFERR   LDX    #OERR     HERE ON OFFSET
C28B 20 08             BRA    LOAD20    ERROR
C28D 7C A0 60  LOAD15  INC    CKSUM     CHECK CHECKSUM
C290 27 A8             BEQ    LOAD3     IF OK, NEXT RECORD
C292 CE C3 CC  LOAD19  LDX    #CERR     HERE ON CKSUM ERR
C295 BD C1 42  LOAD20  JSR    STRING
C298 86 13     LOAD21  LDA A  #$13      RDR OFF
```

```
C29A BD E1 D1          JSR    OUTEE
                * SMARTBUG PATCH GOES HERE ALSO
C29D 86 34             LDA A  #$34      *SB PATCH DEC ECHO
C29F B7 80 07          STA A  PIASB     *SB PATCH NOP NOP
C2A2 7E C1 01          JMP    CNTRL

                * STRINGS, BELLS AND WHISTLES

C2A5 54        IHDR    FCC    'THE MICRO WORKS'
C2A6 48 45
C2A8 20 4D
C2AA 49 43
C2AC 52 4F
C2AE 20 57
C2B0 4F 52
C2B2 4B 53
C2B4 0D                FCB    $D,$A,0,0
C2B5 0A 00
C2B7 00
C2B8 32                FCC    '2708 UTILITY'
C2B9 37 30
C2BB 38 20
C2BD 55 54
C2BF 49 4C
C2C1 49 54
C2C3 59
C2C4 04                FCB    $4
C2C5 48        HVON    FCC    'HV ON?'
C2C6 56 20
C2C8 4F 4E
C2CA 3F
C2CB 04                FCB    $4
C2CC 42        BHDR    FCC    'BURNING, TRY#'
C2CD 55 52
C2CF 4E 49
C2D1 4E 47
C2D3 2C 20
C2D5 54 52
C2D7 59 23
C2D9 04                FCB    $4
C30F                   ORG    U2708+$30F
C30F 48        HVOFF   FCC    'HV OFF?'
C310 56 20
C312 4F 46
C314 46 3F
C316 04                FCB    $4
C317 56        VHDR    FCC    'VERIFY'
C318 45 52
C31A 49 46
C31C 59
C31D 04                FCB    $4
C31E 41        VHDR1   FCC    'ADDR RA PR HS'
C31F 44 44
C321 52 20
C323 52 41
C325 20 50
C327 52 20
C329 48 53
C32B 04                FCB    $4
C32C 45        EHDR    FCC    'ERASURE TEST'
```

```
C32D 52 41
C32F 53 55
C331 52 45
C333 20 54
C335 45 53
C337 54
C338 04                 FCB     $4
C339 41         EHDR1   FCC     'ADDR FF PR HS'
C33A 44 44
C33C 52 20
C33E 46 46
C340 20 50
C342 52 20
C344 48 53
C346 04                 FCB     $4
C347 42         MHDR    FCC     'BLOCK MOVE'
C348 4C 4F
C34A 43 4B
C34C 20 4D
C34E 4F 56
C350 45
C351 04                 FCB     $4
C352 53         MHDR1   FCC     'SRC END DEST'
C353 52 43
C355 20 45
C357 4E 44
C359 20 44
C35B 45 53
C35D 54
C35E 04                 FCB     $4
C35F 53         SHDR    FCC     'SET MEMORY'
C360 45 54
C362 20 4D
C364 45 4D
C366 4F 52
C368 59
C369 04                 FCB     $4
C36A 46         SHDR1   FCC     'FROM TO  HEX'
C36B 52 4F
C36D 4D 20
C36F 54 4F
C371 20 20
C373 48 45
C375 58
C376 04                 FCB     $4
C377 54         XHDR    FCC     'TRANSFER PROM TO RAM'
C378 52 41
C37A 4E 53
C37C 46 45
C37E 52 20
C380 50 52
C382 4F 4D
C384 20 54
C386 4F 20
C388 52 41
C38A 4D
C38B 04                 FCB     $4
C38C 4F         LHDR    FCC     'OFFSET LOADER'
C38D 46 46
C38F 53 45
```

```
C391 54 20
C393 4C 4F
C395 41 44
C397 45 52
C399 04             FCB    $4
C39A 41        LHDR1   FCC    'ADDR K'
C39B 44 44
C39D 52 20
C39F 4B
C3A0 04             FCB    $4
C3A1 45        TERR    FCC    'ERRORS PERSIST...'
C3A2 52 52
C3A4 4F 52
C3A6 53 20
C3A8 50 45
C3AA 52 53
C3AC 49 53
C3AE 54 2E
C3B0 2E 2E
C3B2 04             FCB    $4
C3B3 48        HERR    FCC    'HARD ERRORS'
C3B4 41 52
C3B6 44 20
C3B8 45 52
C3BA 52 4F
C3BC 52 53
C3BE 04             FCB    $4
C3BF 4F        OERR    FCC    'OFFSET ERROR'
C3C0 46 46
C3C2 53 45
C3C4 54 20
C3C6 45 52
C3C8 52 4F
C3CA 52
C3CB 04             FCB    $4
C3CC 43        CERR    FCC    'CHECKSUM ERROR'
C3CD 48 45
C3CF 43 4B
C3D1 53 55
C3D3 4D 20
C3D5 45 52
C3D7 52 4F
C3D9 52
C3DA 04             FCB    $4
C3DB 48        CRLF1   FCC    'H'
C3DC 0D        CRLF    FCB    $D,$A,0,0,0,$4
C3DD 0A 00
C3DF 00 00
C3E1 04

               * RESTART VECTORS

C3F8                   ORG    U2708+$3F8
C3F8 E0 00     IO      FDB    $E000     *E28B FOR RT-68
C3FA E1 13     SFE     FDB    $E113     *E280
C3FC E0 05     POWDWN  FDB    $E005     *E298
C3FE E0 D0     START   FDB    $E0D0     *E147

               * EQUATES AND STORAGE
```

```
A050                ORG    $A050
A050          T     RMB    1
A051          LEN   RMB    1
A052          CW    RMB    1
A053          H     RMB    1
A054          E     RMB    1
A055          RAMAD RMB    2
A057          RAMDAT RMB   1
A058          ROMDAT RMB   1
A059          EA    RMB    2
A05B          DA    RMB    2
A05D          OFFSET RMB   2
A05F          BYTCT RMB    1
A060          CKSUM RMB    1
A00C          XHI   EQU    $A00C
A00D          XLOW  EQU    $A00D

              * I/O ADDRESSES
8010          PIAADR EQU   $8010     I/O ADDRESSES
8011          PIACRA EQU   $8011
8012          PIADAT EQU   $8012
8013          PIACRB EQU   $8013
8007          PIASB  EQU   $8007

              *EQUATE FOR SMARTBUG PATCH
A00B          ECHO   EQU   $A00B

              * VECTORS TO MIK MOUSEBUG
E0E3          MIKBUG EQU   $E0E3
E0C8          OUT4HS EQU   $E0C8
E0CA          OUT2HS EQU   $E0CA
E07E          PDATA1 EQU   $E07E
E055          BYTE   EQU   $E055
E0CC          OUTS   EQU   $E0CC
E047          BADDR  EQU   $E047
E1AC          INEEE  EQU   $E1AC
E1D1          OUTEE  EQU   $E1D1
E0AA          INHEX  EQU   $E0AA     THA'S ALL FOLKS...

                     END   U2708

NO ERROR(S) DETECTED
```

SYMBOL TABLE:

BADDR	E047	BHDR	C2CC	BLOOP1	C04C	BLOOP2	C049	BLOOP3	C02E
BURN	C020	BYTCT	A05F	BYTE	E055	CERR	C3CC	CHECK	C0A6
CKSUM	A060	CMDLP	C113	CNTRL	C101	CR	C145	CRLF	C3DC
CRLF1	C3DB	CW	A052	DA	A05B	DELAY	C0A0	DISCH	C14F
DLOOP	C0A2	E	A054	EA	A059	ECHO	A00B	EHDR	C32C
EHDR1	C339	ELOOP	C191	ERASE	C180	ERP	C084	ERRH	C081
ERRT	C07C	H	A053	HERR	C3B3	HVOFF	C30F	HVON	C2C5
IHDR	C2A5	INEEE	E1AC	INHEX	E0AA	IO	C3F8	JMPC	C087
LEN	A051	LHDR	C38C	LHDR1	C39A	LOAD	C20E	LOAD11	C271
LOAD15	C28D	LOAD19	C292	LOAD20	C295	LOAD21	C298	LOAD3	C23A
MHDR	C347	MHDR1	C352	MIKBUG	E0E3	MLOOP	C1B3	MOVE	C19C
NEXT	C0D4	OERR	C3BF	OFERR	C288	OFFSET	A05D	ONEADS	C208
OUT	C0CE	OUT2HS	E0CA	OUT4HS	E0C8	OUTEE	E1D1	OUTS	E0CC
PCTL1	C09A	PCTLB	C092	PDATA1	E07E	PIAADR	8010	PIACRA	8011
PIACRB	8013	PIADAT	8012	PIASB	8007	POUNCE	C121	POWDWN	C3FC
PUTAD	C0DF	RAMAD	A055	RAMDAT	A057	RECOV	C0B9	ROMDAT	A058
SET	C1C6	SFE	C3FA	SHDR	C35F	SHDR1	C36A	SLOOP	C1DC
START	C3FE	STRING	C142	T	A050	TABEND	C13E	TABLE	C126
TERR	C3A1	TWOADS	C1FC	U2708	C000	VERIFY	C165	VHDR	C317
VHDR1	C31E	VLOOP	C171	XFER	C1E6	XHDR	C377	XHI	A00C
XLOOP	C1EF	XLOW	A00D						