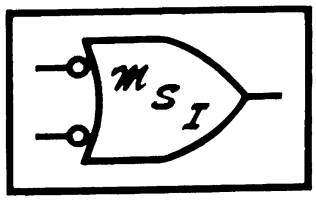

MSI-BUG MONITOR ROUTINES

Midwest Scientific Instruments
Olathe, Kansas

MSI-BUG MONITOR PROM, MODEL MT-1

INTRODUCTION

The MSI-BUG Monitor is provided on a 2708 (or equivalent) EPROM in order to provide the MSI Computer System with an immediate means of communication with a control terminal. Upon power up, the monitor routine is automatically entered which results in the printing of an asterisk (*) on the control terminal as a prompt character. The monitor is then ready to receive any of several input character commands which will permit execution of the functions described below.

FEATURES OF THE MSI-BUG MONITOR

Several features are provided in the MSI-Bug Monitor which include the following operations:

1. Memory Examine & Change
2. CPU Register Dump
3. List in Instruction Format
4. Checksum Block of Memory
5. Punch to a Tape Device
6. Load from a Tape Device
7. Execute User Program Function

In addition, many other features are included in the monitor which allow an I/O port number to be specified on punch or load functions, and control characters can be used to momentarily stop the output character routine or to return to monitor control. Each of these functions is described in detail below:

MONITOR COMMANDS:

Memory Examine & Change
Command: M XXXX

Typing a capital "M" followed by a four digit hex address will open that particular address for examination and the content of that memory location will be printed on the terminal in hex. Typing a slash (or almost any other character) will automatically advance to the next memory location and display the contents of that location. Typing a period will display the previous memory location. In order to deposit a new byte at a given memory location, type a space followed by the hex byte to be deposited. The byte will be deposited and the memory location will automatically increment and display the contents of the next location. One may advance to a new memory location by simply typing N XXXX at which time the content of the new location will be displayed. Typing either two consecutive spaces or a carriage return will return to the MSI-BUG Monitor.

CPU Register Dump
Command: R

Typing a capital "R" will print the CPU registers in the form CC BB AA XXXX PPPP SSSS where CC is the condition code register, BB is the B accumulator, AA is the A accumulator, XXXX is the index register, PPPP is the program counter, and SSSS is the stack pointer. The values printed are taken from the machine stack and are the values which the CPU will have restored when a G command is entered.

Lister Function
Command: T XXXX XXXX

Typing a capital "T" followed by a beginning memory location followed by an ending memory location, in hex, will result in the content of that block of memory being printed out on the control terminal in instruction format. One, two, and three byte instructions are displayed on a single line in order to facilitate examination by the programmer. While the listing is in progress, typing a CONTROL S will momentarily halt the print out, allowing examination of the screen. Typing any character will allow the print out to resume until the last address has been reached.

While the listing is in progress, typing a CONTROL E will allow the function to continue but no characters are printed on the control terminal. Typing another CONTROL E will cause the characters to be printed again. The function runs much faster when no time is taken to print to the control terminal, so a section of the listing can be skipped by typing CONTROL E to turn off the listing, and then another CONTROL E to turn it back on. Typing a CONTROL D will cause the monitor to stop the current function, print the asterisk (*) prompt character, and wait for an input character command. CONTROL S, CONTROL E, and CONTROL D may be used whenever a program communicates with the control terminal through the monitor's input and output routines.

Checksum Function
Command: C XXXX XXXX

Typing a capital "C" followed by a beginning memory location followed by an ending memory location, in hex, will result in a three byte checksum being calculated for that particular block of memory. This procedure is valuable in order to determine whether or not a particular byte of memory has changed during program testing or debug operations.

Punch to a Tape Device
Command: P 0 XXXX XXXX

Typing a capital "P" followed by a port number (0 or 1), followed by a beginning memory location, followed by an ending memory location, results in the content of that

memory block being punched to the specified output device in Motorola format. Either PORT 0 or PORT 1 may be specified immediately following the P. This allows a tape cassette device operating at 300 baud to be used on PORT 1 while maintaining a control terminal on PORT 0 which maybe operating at a higher baud rate. This allows punch and load functions to be carried out without alteration of the control terminal baud rate. Following the entry of the last address, the program will ask whether or not "ECHO" is desired. Typing a "Y" in response to the question will result in the output characters being displayed on the control terminal simultaneously as they are output to the punch device.

Load Function
Command: L 0

Typing a capital "L" followed by a port number (0 or 1) will allow a memory load function to be carried out from a tape input device. The program expects to receive data in standard Motorola format. PORT 0 or PORT 1 may be specified. An optional echo is also available. Type either "Y" or "N" in response to the "echo" question.

Execute User Program Function
Command: G XXXX

Typing a capital "G" followed by a beginning memory location results in the monitor jumping to that beginning memory location and executing a program which is contained at that location. Program execution will continue until halted by either a software interrupt command (3F) or returning to monitor control.

HARDWARE CONFIGURATION

The MSI-BUG Monitor is designed to communicate with a serial interface (6850 ACIA, MSI Model SI-1 Interface) on I/O PORT 0 of the MSI-6800 which has a base address of $F500. The monitor also requires 128 bytes of RAM memory to be located $F000.

A second version of the MSI-Bug Monitor, the Model MT-2, is available for use with the MSI CPU card when installed in SWTP 6800 computer systems. The Model MT-2 Monitor communicates with a serial interface (6850 ACIA) on I/O PORT 0, of a SWTP 6800 computer system, which has a base address of $8000. This version of the monitor expects 128 bytes of RAM memory to be available at $A000.

FREQUENTLY USED MONITOR ROUTINES

This is a list of monitor routines that can be called from a machine language program with a JSR instruction. These routines may be used to perform many different input and output functions. The name, address, and description of each routine is given. The "registers changed" column

lists the registers altered by each routine.

NAME	ADDRESS	REGISTERS CHANGED	DESCRIPTION
BADDR	$E047	A,B,X	accept 4 hex digits from terminal, return 2 bytes in X.
BYTE	$E055	A,B	accept 2 hex digits from terminal, return 1 byte in A.
OUTHL	$E067	A	print left hex digit of byte in A.
OUTHR	$E06B	A	print right hex digit of byte in A.
OUTCH	$E075		print character in A on interface whose address is in $F018
INCH	$E078	A	return in A the character taken from interface whose address is in $F016
PDATA1	$E07E	A,X	print bytes from memory, starting at address in X, until $04 is found
INHEX	$E089	A	accept 1 hex digit from terminal, return in right half of A, left half is zero.
OUT2H	$E09E	A,X	take 1 byte at the address in X print 2 hex digits.
OUTS	$E0CC	A	print 1 space.
INEEE	$E1AC	A	accept character from terminal, return in A.
OUTEEE	$E1D1		print the character A
OUT2HS	$E0CA	A,X	take 1 byte at the address in X, print 2 hex digits and 1 space.
OUT4HS	$EDC8	A,X	take 2 bytes at the address in X, print 4 hex digits and 1 space.

MONITOR SOFTWARE INTERRUPT FUNCTIONS

The Software Interrupt instruction may be used to transfer control from a main routine to a subroutine and back to the main routine by using the monitor's Software Interrupt Function. Execution of an SWI instruction causes the processor registers to be pushed on the stack and execution continued at the address stored in locations $FFFA

and $FFFB. This is the address of the monitor's SWI entry point. The monitor jumps to the address stored in location $F014 and $F015 where execution begins. Those locations are initialized by the monitor with the register print routine address, so execution of an SWI instruction causes the CPU registers to be printed. Putting the address of a subroutine in locations $F014 and $F015 causes that routine to be entered when an SWI is encountered. The routine is exited with a return from Interrupt (RTI) instruction which causes the processor registers to be pulled off the stack and execution resumed in the main program immediately following the SWI instruction. An example program is included which illustrates how a user's own SWI routine might be implemented.

CONVENTION FOR STOP BITS ON PUNCH AND LOAD

The MSI-BUG PUNCH Function outputs data with the interface initialized for 2 stop bits and the LOAD Function initializes for 1 stop bit. This is the convention because the number of stop bits cannot be dynamically changed and this allows the PUNCH and LOAD functions to operate with Teletype terminals as well as K.C. Standard Cassettes. Teletype output requires 2 stop bits whereas the monitor can load that data with the interface initialized for 1 stop bit. The ACIA initialization must be for 1 stop bit on LOAD in order for the function to work with K.C. Standard Cassette tapes which are normally punched with 1 stop bit.

USING NMI AND IRQ WITH THE MONITOR

The Interrupt Request and Non-maskable Interrupt may be used to transfer control to a machine language routine. When an NMI occurs, the CPU takes the NMI vector from locations $FFFC and $FFFD and jumps there. The NMI vector is $E005, the monitor's NMI routine address. The monitor then looks at its own NMI vector which is in monitor RAM at $F006 and $F007 and jumps to the address stored there. Placing the address of a machine language routine in locations $F006 and $F007 will cause execution of that routine when an NMI occurs.

When an IRQ occurs, and if the interrupt mask has been cleared, the CPU takes the IRQ vector from locations $FFF8 and $FFF9, which is $E000, the monitor's IRQ routine. The monitor loads an address from memory locations $F000 and $F001 and then jumps to that address to begin execution. Placing a machine language routine at memory locations $F000 and $F001 will result in the CPU executing that routine when an IRQ occurs. Execution will continue until a return from interrupt (RTI) instruction is encountered, at which time the CPU will return to its original program. The CPU registers are always pushed onto the stack prior to entering any interrupt routine.

An IRQ or NMI is initiated by grounding the IRQ or the NMI bus lines. Front panel switches on the MSI 6800 Computer System allow simulation of these functions.

```
00010                          NAM     SWI
00020                          OPT     O,NOG
00030 0100                     ORG     $0100
00040                  *
00050                  *MIDWEST SCIENTIFIC INSTRUMENTS INC.
00060                  *
00070       F014       SWIVEC  EQU     $F014    SWI VECTOR IN RAM
00080       E133       SFE1    EQU     $E133    NORMAL SWI ROUTINE
00090       E1D1       OUTEEE  EQU     $E1D1    OUTPUT CHARACTER ROUTINE
00100                  *
00110 0100 CE 011A     START   LDX     #PRCH    LOAD ADDR OF PRINT ROUTINE
00120 0103 FF F014             STX     SWIVEC   PUT IN SWI VECTOR
00130 0106 CE 011E             LDX     #MSG
00140 0109 A6 00       PRINT   LDA A   0,X      PRINT CHARACTERS UNTIL A $04
00150 010B 81 04               CMP A   #4
00160 010D 27 04               BEQ     PRINT1
00170 010F 3F                  SWI             DO A SWI TO CALL THE PRINT
00180 0110 08                  INX             ROUTINE
00190 0111 20 F6               BRA     PRINT
00200 0113 CE E133     PRINT1  LDX     #SFE1    PUT SWI ROUTINE ADDRESS
00210 0116 FF F014             STX     SWIVEC   BACK IN SWI VECTOR
00220 0119 3F                  SWI             RETURN TO MONITOR
00230                  *
00240 011A BD E1D1     PRCH    JSR     OUTEEE   PRINT THE CHARACTER
00250 011D 3B                  RTI             GO BACK TO CALLING ROUTINE
00260                  *
00270 011E 0D          MSG     FCB     $0D,$0A
00280 0120 48                  FCC     /HELLO/
00290 0125 0D                  FCB     $0D,$0A,$04
00300                          END

TOTAL ERRORS 00000

ENTER PASS : 1P,2P,2L,2T
```

L

```
00010                          NAM    MSIBUG
00020                  *
00021                  *MIDWEST SCIENTIFIC INSTRUMENTS INC.
00022                  *
00030                  *MSIBUG MONITOR MT-1 FOR USE WITH THE
00040                  *MSI 6800 COMPUTER.
00050                  *VERSION 1.7
00060                  *ACIA INTERFACES ARE AT $F500 AND $F503.
00070                  *MONITOR RAM IS AT $F000.
00080                  *
00090                  *MSIBUG MONITOR MT-2 FOR USE WITH THE
00100                  *SWTPC 6800 COMPUTER USES ACIA INTERFACES
00110                  *AT $8000 AND $8004, AND RAM AT $A000.
00120                  *THE MT-2 LISTING IS THIS LISTING WITH
00130                  *THE FOLLOWING CHANGES:
00140                  *  ACIAS EQU $8000
00150                  *  ACIAT EQU $8004
00160                  *  THE EQU FOR RAM IS  EQU $A000
00170                  *  STACK IS AT $A049
00180                  *
00190                  *WRITTEN BY ED WELLS AND HAL HOFFMAN
00200                  *LAST CHANGE 1-17-78 BY HAL HOFFMAN
00210                  *
00220                          OPT    O
00230          F500    ACIAS   EQU    $F500
00240          F508    ACIAT   EQU    $F508
00250          E400    PRMNXT  EQU    $E400
00260  E000                    ORG    $E000
00270  E000 FE F000    IO      LDX    IOV
00280  E003 6F 00              JMP    0,X
00290  E005 FE F006    POWDWN  LDX    NIO
00300  E008 6E 00              JMP    0,X
00310  E00A BD E15C    LOAD    JSR    LOADI
00320  E00D BD E1DF            JSR    OUT3
00330  E010 BD E078    LOAD3   JSR    INCH
00340  E013 81 53              CMP A  #'S
00350  E015 26 F9              BNE    LOAD3
00360  E017 BD E078            JSR    INCH
00370  E01A 81 39              CMP A  #'9
00380  E01C 27 26              BEQ    LOAD21
00390  E01E 81 31              CMP A  #'1
00400  E020 26 EE              BNE    LOAD3
00410  E022 7F F00A            CLR    CKSM
00420  E025 8D 2E              BSR    BYTE
00430  E027 80 02              SUB A  #2
00440  E029 B7 F00B            STA A  BYTECT
00450  E02C 8D 19              BSR    BADDR
00460  E02F 8D 25    LOAD11  BSR    BYTE
00470  E030 7A F00B            DEC    BYTECT
00480  E033 27 05              BEQ    LOAD15
00490  E035 A7 00              STA A  0,X
00500  E037 08                INX
00510  E038 20 F4              BRA    LOAD11
00520  E03A 7C F00A  LOAD15  INC    CKSM
```

```
00530 E03D 27 D1          BEQ    LOAD3
00540 E03F 86 3F   LOAD19 LDA A  #'?
00550 E041 BD E1D1        JSR    OUTEEE
00560 E044 7E E0E3 LOAD21 JMP    CONTRL
00570 E047 8D 0C   BADDR  BSR    BYTE
00580 E049 B7 F00C        STA A  XHI
00590 E04C 8D 07          BSR    BYTE
00600 E04E B7 F00D        STA A  XLOW
00610 E051 FE F00C        LDX    XHI
00620 E054 39             RTS
00630 E055 8D 32   BYTE   BSR    INHEX
00640 E057 48      BYTE2  ASL A
00650 E058 48             ASL A
00660 E059 48             ASL A
00670 E05A 48             ASL A
00680 E05B 16             TAB
00690 E05C 8D 2B          BSR    INHEX
00700 E05E 1B             ABA
00710 E05F 16             TAB
00720 E060 FB F00A        ADD B  CKSM
00730 E063 F7 F00A        STA B  CKSM
00740 E066 39             RTS
00750 E067 44      OUTHL  LSR A
00760 E068 44             LSR A
00770 E069 44             LSR A
00780 E06A 44             LSR A
00790 E06B 84 0F   OUTHR  AND A  #$0F
00800 E06D 8B 30          ADD A  #$30
00810 E06F 81 39          CMP A  #$39
00820 E071 23 02          BLS    OUTCH
00830 E073 8B 07          ADD A  #7
00840 E075 7E E1D9 OUTCH  JMP    OUT2
00850 E078 7E E1B4 INCH   JMP    IN2
00860 E07B 8D F8   PDATA2 BSR    OUTCH
00870 E07D 08             INX
00880 E07E A6 00   PDATA1 LDA A  0,X
00890 E080 81 04          CMP A  #4
00900 E082 26 F7          BNE    PDATA2
00910 E084 39             RTS
00920 E085 8D 31   PDATAC BSR    PORTAC
00930 E087 20 F5          BRA    PDATA1
00940 E089 8D ED   INHFX  BSR    INCH
00950 E08B 80 30   INHEX2 SUB A  #$30
00960 E08D 2B 54          BMI    CONTRL
00970 E08F 81 09          CMP A  #9
00980 E091 2F 0A          BLE    IN1HG
00990 E093 81 11          CMP A  #$11
01000 E095 2B 4C          BMI    CONTRL
01010 E097 81 16          CMP A  #$16
01020 E099 2E 48          BGT    CONTRL
01030 E09B 80 07          SUB A  #7
01040 E09D 39      IN1HG  RTS
01050 E09E A6 00   OUT2H  LDA A  0,X
01060 E0A0 8D C5   OUT2HA BSR    OUTHL
```

```
01070 E0A2 A6 00          LDA A  0,X
01080 E0A4 08             INX
01090 E0A5 20 C4          BRA    OUTHR
01100 E0A7 BD E1AC WAIT   JSR    INEEE
01110 E0AA 81 0D          CMP A  #$0D
01120 E0AC 27 03          BEQ    *+5
01130 E0AE 7E E0E3        JMP    CONTRL
01140 E0B1 CE E17D WAIT2  LDX    #CRLF
01150 E0B4 BD E07E WAIT1  JSR    PDATA1
01160 E0B7 39             RTS
01170 E0B8 FF F012 PORTAC STX    XTEMP
01180 E0BB CE F500        LDX    #ACIAS
01190 E0BE FF F018        STX    OUTADD
01200 E0C1 FF F016        STX    INADD
01210 E0C4 FE F012        LDX    XTEMP
01220 E0C7 39             RTS
01230 E0C8 8D D4   OUT4HS BSR    OUT2H
01240 E0CA 8D D2   OUT2HS BSR    OUT2H
01250 E0CC 86 20   OUTS   LDA A  #$20
01260 E0CE 20 A5          BRA    OUTCH
01270 E0D0 8E F072 START  LDS    #STACK
01280 E0D3 BF F008        STS    SP
01290 E0D6 CE E133        LDX    #SFE1
01300 E0D9 FF F014        STX    SWI
01310 E0DC 86 13          LDA A  #$13
01320 E0DE B7 F500        STA A  ACIAS
01330 E0E1 01             NOP
01340 E0E2 01             NOP
01350 E0E3 86 11   CONTRL LDA A  #$11
01360 E0E5 B7 F500        STA A  ACIAS
01370 E0E8 7F F01A        CLR    OUTSW
01380 E0EB BE F008        LDS    SP
01390 E0EE CE E172        LDX    #MCLOFF
01400 E0F1 8D 92          BSR    PDATAC
01410 E0F3 BD E1AC        JSR    INEEE
01420 E0F6 16             TAB
01430 E0F7 8D D3          BSR    OUTS
01440 E0F9 C1 4C          CMP B  #'L
01450 E0FB 27 69          BEQ    TAPE
01460 E0FD C1 4D          CMP B  #'M
01470 F0FF 27 68          BEQ    MEMORY
01480 E101 C1 50          CMP B  #'P
01490 E103 27 61          BEQ    TAPF
01500 E105 C1 52          CMP B  #'R
01510 E107 27 36          BEQ    PRINT
01520 E109 C1 43          CMP B  #'C
01530 E10B 27 5F          BEQ    CHECK
01540 E10D C1 54          CMP B  #'T
01550 E10F 27 5E          BEQ    TYPE
01560 E111 C1 47          CMP B  #'G
01570 E113 27 0A          BEQ    GO
01580 E115 B6 E400        LDA A  PRMNXT
01590 E118 81 7E          CMP A  #$7E
01600 E11A 26 C7          BNE    CONTRL
```

```
01610 E11C 7E E400          JMP     PRMNXT
01620 E11F BD E318 GO       JSR     INPUT1
01630 E122 30               TSX
01640 E123 B6 F01E          LDA A   ENDA
01650 E126 A7 05            STA A   5,X
01660 E128 B6 F01F          LDA A   ENDA+1
01670 E12B A7 06            STA A   6,X
01680 E12D 3B               RTI
01690 E12E FE F014 SFE      LDX     SWI
01700 E131 6E 00            JMP     0,X
01710 E133 BF F008 SFE1     STS     SP
01720 E136 30               TSX
01730 E137 6D 06            TST     6,X
01740 E139 26 02            BNE     *+4
01750 E13B 6A 05            DEC     5,X
01760 E13D 6A 06            DEC     6,X
01770 E13F BD E0B8 PRINT    JSR     PORTAC
01780 E142 FE F008          LDX     SP
01790 E145 08               INX
01800 E146 C6 03            LDA B   #3
01810 E148 BD E0CA PRINT1   JSR     OUT2HS
01820 E14B 5A               DEC B
01830 E14C 26 FA            BNE     PRINT1
01840 E14E BD E0C8          JSR     OUT4HS
01850 E151 BD E0C8          JSR     OUT4HS
01860 E154 CE F008          LDX     #SP
01870 E157 BD E0C8          JSR     OUT4HS
01880 E15A 20 87            BRA     CONTRL
01890 E15C FE F016 LOADI    LDX     INADD
01900 E15F 86 55            LDA A   #$55
01910 E161 A7 00            STA A   0,X
01920 E163 86 11            LDA A   #$11
01930 E165 39               RTS
01940 E166 7E E3B6 TAPE     JMP     TAPEIO
01950 E169 7E E212 MEMORY   JMP     CHANGE
01960 E16C 7E E31F CHECK    JMP     CKSUM
01970 E16F 7E E34F TYPE     JMP     LISTER
01980 E172 13      MCLOFF   FCB     $13
01990 E173 0D      MCL      FCB     $0D,$0A,$14,0,0,0,0,0,'*,4
      E174 0A
      E175 14
      E176 00
      E177 00
      E178 00
      E179 00
      E17A 00
      E17B 2A
      E17C 04
02000 E17D 0D      CRLF     FCB     $0D,$0A,0,0,0,0,0,0,4
      E17E 0A
      E17F 00
      E180 00
      E181 00
      E182 00
```

```
        E183 00
        E184 00
        E185 04
02010   E186 0D       MTAPE1  FCB     $0D,$0A,0,0,0,0,0,0,'S,'1,4
        E187 0A
        E188 00
        E189 00
        E18A 00
        E18B 00
        E18C 00
        E18D 00
        E18E 53
        E18F 31
        E190 04
02020   E191 20       ECHO    FCB     $20,$20,'E,'C,'H,'O,'?,$20,4
        E192 20
        E193 45
        E194 43
        E195 48
        E196 4F
        E197 3F
        E198 20
        E199 04
02030   E19A FE F016  ACIN    LDX     INADD
02040   E19D 20 03            BRA     *+5
02050   E19F FE F018  ACOUT   LDX     OUTADD
02060   E1A2 86 13            LDA A   #$13
02070   E1A4 A7 00            STA A   0,X
02080   E1A6 86 11            LDA A   #$11
02090   E1A8 A7 00            STA A   0,X
02100   E1AA 39               RTS
02110   E1AB 00               FCB     0
02120   E1AC FF F012  INEEE   STX     XTEMP
02130   E1AF CE F500          LDX     #ACIAS
02140   E1B2 20 06            BRA     *+8
02150   E1B4 FF F012  IN2     STX     XTEMP
02160   E1B7 FE F016          LDX     INADD
02170   E1BA 8D 08    IN3     BSR     INCHP
02180   E1BC 84 7F            AND A   #$7F
02190   E1BE 81 7F            CMP A   #$7F
02200   E1C0 27 F8            BEQ     IN3
02210   E1C2 20 18            BRA     OUT2+3
02220   E1C4 A6 00    INCHP   LDA A   0,X
02230   E1C6 47               ASR A
02240   E1C7 24 FB            BCC     INCHP
02250   E1C9 A6 01            LDA A   1,X
02260   E1CB 39               RTS
02270   E1CC 00               FCB     0,0
        E1CD 00
02280   E1CE 7E E0E3  QUIT    JMP     CONTRL
02290   E1D1 FF F012  OUTEEE  STX     XTEMP
02300   E1D4 CE F500          LDX     #ACIAS
02310   E1D7 20 06            BRA     *+8
02320   E1D9 FF F012  OUT2    STX     XTEMP
```

```
02330 E1DC FE F018          LDX     OUTADD
02340 E1DF 37       OUT3    PSH B
02350 E1E0 8D 2C            BSR     POLL
02360 E1E2 24 18            BCC     INOUT
02370 E1E4 0C       INBRK   CLC
02380 E1E5 36               PSH A
02390 E1E6 A6 01            LDA A   1,X
02400 E1E8 84 7F            AND A   #$7F
02410 E1EA 81 05            CMP A   #5
02420 E1EC 26 03            BNE     *+5
02430 E1EE 73 F01A          COM     OUTSW
02440 E1F1 81 13            CMP A   #$13
02450 E1F3 26 02            BNE     *+4
02460 E1F5 8D CD            BSR     INCHP
02470 E1F7 81 04            CMP A   #4
02480 E1F9 27 D3            BEQ     QUIT
02490 E1FB 32               PUL A
02500 E1FC 7D F01A INOUT    TST     OUTSW
02510 E1FF 26 08            BNE     OUT4
02520 E201 E6 00    OUTC1   LDA B   0,X
02530 E203 57               ASR B
02540 E204 57               ASR B
02550 E205 24 FA            BCC     OUTC1
02560 E207 A7 01            STA A   1,X
02570 E209 33       OUT4    PUL B
02580 E20A FE F012          LDX     XTEMP
02590 E20D 39               RTS
02600 E20E E6 00    POLL    LDA B   0,X
02610 E210 57               ASR B
02620 E211 39               RTS
02630 E212 BD E047 CHANGE   JSR     BADDR
02640 E215 20 32            BRA     UP1
02650 E217 FE F00C CHANG    LDX     XHI
02660 E21A BD E0CA          JSR     OUT2HS
02670 E21D 09               DEX
02680 E21E 8D 8C    CHA1    BSR     INEEE
02690 E220 81 2F            CMP A   #'/
02700 E222 27 21            BEQ     DWN
02710 E224 81 2E            CMP A   #'.
02720 E226 27 20            BEQ     UP
02730 E228 81 4E            CMP A   #'N
02740 E22A 26 05            BNE     *+7
02750 E22C BD E0CC          JSR     OUTS
02760 E22F 20 E1            BRA     CHANGE
02770 E231 81 0D            CMP A   #$0D
02780 E233 27 99            BEQ     QUIT
02790 E235 81 20            CMP A   #$20
02800 E237 26 0C            BNE     DWN
02810 E239 BD E055          JSR     BYTE
02820 E23C A7 00            STA A   0,X
02830 E23E A1 00            CMP A   0,X
02840 E240 27 03            BEQ     DWN
02850 E242 7E E03F          JMP     LOAD19
02860 E245 08       DWN     INX
```

```
02870 E246 20 01          BRA    UP1
02880 E248 09       UP    DEX
02890 E249 FF F00C  UP1   STX    XHI
02900 E24C CE E17D        LDX    #CRLF
02910 E24F BD E07E        JSR    PDATA1
02920 E252 CE F00C        LDX    #XHI
02930 E255 BD E0C8        JSR    OUT4HS
02940 E258 20 BD          BRA    CHANG
02950 E25A 86 12  PUNCH   LDA A  #$12
02960 E25C BD E075        JSR    OUTCH
02970 E25F FE F01C        LDX    BEGA
02980 E262 FF F00F        STX    TW
02990 E265 B6 F01F PUN11  LDA A  ENDA+1
03000 E268 B0 F010        SUB A  TW+1
03010 E26B F6 F01E        LDA B  ENDA
03020 E26E F2 F00F        SBC B  TW
03030 E271 26 04          BNE    PUN22
03040 E273 81 10          CMP A  #16
03050 E275 25 02          BCS    PUN23
03060 E277 86 0F  PUN22   LDA A  #15
03070 E279 8B 04  PUN23   ADD A  #4
03080 E27B B7 F011        STA A  MCONT
03090 E27E 80 03          SUB A  #3
03100 E280 B7 F00E        STA A  TEMP
03110 E283 F6 F01A        LDA B  OUTSW
03120 E286 37             PSH B
03130 E287 7F F01A        CLR    OUTSW
03140 E28A CE E186        LDX    #MTAPE1
03150 E28D BD E07E        JSR    PDATA1
03160 E290 33             PUL B
03170 E291 F7 F01A        STA B  OUTSW
03180 E294 7D F01B        TST    SAVE
03190 E297 27 0C          BEQ    PUN30
03200 E299 CE E196        LDX    #MTAPE1
03210 E29C BD E085        JSR    PDATAC
03220 E29F CE F508        LDX    #ACIAT
03230 E2A2 FF F018        STX    OUTADD
03240 E2A5 5F       PUN30 CLR B
03250 E2A6 CE F011        LDX    #MCONT
03260 E2A9 8D 32          BSR    PUNT2
03270 E2AB CE F00F        LDX    #TW
03280 E2AE 8D 2D          BSR    PUNT2
03290 E2B0 8D 2B          BSR    PUNT2
03300 E2B2 FE F00F        LDX    TW
03310 E2B5 8D 26  PUN32   BSR    PUNT2
03320 E2B7 7A F00E        DEC    TEMP
03330 E2BA 26 F9          BNE    PUN32
03340 E2BC FF F00F        STX    TW
03350 E2BF 53             COM B
03360 E2C0 37             PSH B
03370 E2C1 30             TSX
03380 E2C2 8D 19          BSR    PUNT2
03390 E2C4 33             PUL B
03400 E2C5 FE F00F        LDX    TW
```

```
03410 E2C8 09              DEX
03420 E2C9 BC F01E         CPX     ENDA
03430 E2CC 26 97           BNE     PUN11
03440 E2CE 7F F01A         CLR     OUTSW
03450 E2D1 CE E2DA         LDX     #S9
03460 E2D4 BD E07E         JSR     PDATA1
03470 E2D7 7E E0E3         JMP     CONTRL
03480 E2DA 53       S9     FCB     'S,'9,4
      E2DB 39
      E2DC 04
03490 E2DD EB 00    PUNT2  ADD B   0,X
03500 E2DF B6 F01A         LDA A   OUTSW
03510 E2E2 36              PSH A
03520 E2E3 7F F01A         CLR     OUTSW
03530 E2E6 BD E09E         JSR     OUT2H
03540 E2E9 32              PUL A
03550 E2EA B7 F01A         STA A   OUTSW
03560 E2ED 7D F01B         TST     SAVE
03570 E2F0 27 1C           BEQ     P3
03580 E2F2 09              DEX
03590 E2F3 FF F012         STX     XTEMP
03600 E2F6 CE F500         LDX     #ACIAS
03610 E2F9 FF F018         STX     OUTADD
03620 E2FC FE F012         LDX     XTEMP
03630 E2FF BD E09E         JSR     OUT2H
03640 E302 FF F012         STX     XTEMP
03650 E305 CE F508         LDX     #ACIAT
03660 E308 FF F018         STX     OUTADD
03670 E30B FE F012         LDX     XTEMP
03680 E30E 39       P3     RTS
03690 E30F BD E047 INPUT   JSR     BADDR
03700 E312 FF F01C         STX     BEGA
03710 E315 BD E0CC         JSR     OUTS
03720 E318 BD E047 INPUT1  JSR     BADDR
03730 E31B FF F01E         STX     ENDA
03740 E31E 39              RTS
03750 E31F 8D EE   CKSUM   BSR     INPUT
03760 E321 BD E0B1         JSR     WAIT2
03770 E324 7F F00A         CLR     CKSM
03780 E327 5F              CLR B
03790 E328 4F              CLR A
03800 E329 FE F01C         LDX     BEGA
03810 E32C 09              DEX
03820 E32D 08       CKSUM1 INX
03830 E32E AB 00           ADD A   0,X
03840 E330 C9 00           ADC B   #0
03850 E332 24 03           BCC     *+5
03860 E334 7C F00A         INC     CKSM
03870 E337 BC F01E         CPX     ENDA
03880 E33A 26 F1           BNE     CKSUM1
03890 E33C F7 F00B CKSUM2  STA B   CKSM+1
03900 E33F B7 F00C         STA A   CKSM+2
03910 E342 CE F00A         LDX     #CKSM
03920 E345 C6 03           LDA B   #3
```

```
03930 E347 BD E0CA         JSR    OUT2HS
03940 E34A 5A              DEC B
03950 E34B 26 FA           BNE    *-4
03960 E34D 20 64           BRA    STOP
03970 E34F 8D BE   LISTER  BSR    INPUT
03980 E351 BD E0B1         JSR    WAIT2
03990 E354 CE F01C LISTA   LDX    #BEGA
04000 E357 BD E0C8         JSR    OUT4HS
04010 E35A FE F01C         LDX    BEGA
04020 E35D A6 00           LDA A  0,X
04030 E35F 36              PSH A
04040 E360 BD E0CA         JSR    OUT2HS
04050 E363 FF F01C         STX    BEGA
04060 E366 5F              CLR B
04070 E367 32              PUL A
04080 E368 81 8C           CMP A  #$8C
04090 E36A 27 18           BEQ    THREE
04100 E36C 81 8E           CMP A  #$8E
04110 E36E 27 14           BEQ    THREE
04120 E370 81 CE           CMP A  #$CE
04130 E372 27 10           BEQ    THREE
04140 E374 84 F0           AND A  #$F0
04150 E376 81 20           CMP A  #$20
04160 E378 27 0B           BEQ    TWO
04170 E37A 81 60           CMP A  #$60
04180 E37C 25 08           BCS    ONE
04190 E37E 84 30           AND A  #$30
04200 E380 81 30           CMP A  #$30
04210 E382 26 01           BNE    TWO
04220 E384 5C      THREE   INC B
04230 E385 5C      TWO     INC B
04240 E386 F7 F00E ONE     STA B  TEMP
04250 E389 27 0D           BEQ    LISTB
04260 E38B 7A F00E         DEC    TEMP
04270 E38E 27 05           BEQ    ONLYON
04280 E390 BD E0C8         JSR    OUT4HS
04290 E393 20 03           BRA    LISTB
04300 E395 BD E0CA ONLYON  JSR    OUT2HS
04310 E398 FF F01C LISTB   STX    BEGA
04320 E39B CE E17D         LDX    #CRLF
04330 E39E BD E07E         JSR    PDATA1
04340 E3A1 B6 F01C         LDA A  BEGA
04350 E3A4 B1 F01E         CMP A  ENDA
04360 E3A7 25 AB           BCS    LISTA
04370 E3A9 2E 08           BGT    STOP
04380 E3AB F6 F01D         LDA B  BEGA+1
04390 E3AE F1 F01F         CMP B  ENDA+1
04400 E3B1 25 A1           BCS    LISTA
04410 E3B3 7E E0E3 STOP    JMP    CONTRL
04420 E3B6 C0 4C   TAPEIO  SUB B  #'L
04430 E3B8 F7 F00E         STA B  TEMP
04440 E3BB BD E1AC         JSR    INEEE
04450 E3BE 16              TAB
04460 E3BF BD E0CC         JSR    OUTS
```

```
04470 E3C2 7D F00E          TST     TEMP
04480 E3C5 27 08            BEQ     *+10
04490 E3C7 37               PSH B
04500 E3C8 BD E30F          JSR     INPUT
04510 E3CB 33               PUL B
04520 E3CC 7F F01A          CLR     OUTSW
04530 E3CF CE E191          LDX     #ECHO
04540 E3D2 BD E085          JSR     PDATAC
04550 E3D5 BD E1AC          JSR     INEEE
04560 E3D8 81 59            CMP A   #'Y
04570 E3DA 27 03            BEQ     *+5
04580 E3DC 73 F01A          COM     OUTSW
04590 E3DF CE F500          LDX     #ACIAS
04600 E3E2 C0 30            SUB B   #$30
04610 E3E4 27 06            BEQ     *+8
04620 E3E6 CE F508          LDX     #ACIAT
04630 E3E9 BD E1A2          JSR     ACOUT+3
04640 E3EC F7 F01B          STA B   SAVE
04650 E3EF 7D F00E          TST     TEMP
04660 E3F2 26 06            BNE     *+8
04670 E3F4 FF F016          STX     INADD
04680 E3F7 7E E00A          JMP     LOAD
04690 E3FA FF F018          STX     OUTADD
04700 E3FD 7E E25A          JMP     PUNCH
04710 F000                  ORG     $F000
04720 F000 0002     IOV     RMB     2
04730 F002 0004             RMB     4
04740 F006 0002     NIO     RMB     2
04750 F008 0002     SP      RMB     2
04760 F00A 0001     CKSM    RMB     1
04770 F00B 0001     BYTECT  RMB     1
04780 F00C 0001     XHI     RMB     1
04790 F00D 0001     XLOW    RMB     1
04800 F00E 0001     TEMP    RMB     1
04810 F00F 0002     TW      RMB     2
04820 F011 0001     MCONT   RMB     1
04830 F012 0002     XTEMP   RMB     2
04840 F014 0002     SWI     RMB     2
04850 F016 0002     INADD   RMB     2
04860 F018 0002     OUTADD  RMB     2
04870 F01A 0001     OUTSW   RMB     1
04880 F01B 0001     SAVE    RMB     1
04890 F01C 0002     BEGA    RMB     2
04900 F01E 0002     ENDA    RMB     2
04910 F020 0052             RMB     $52
04920 F072 0002     STACK   RMB     2
04930                       END

TOTAL ERRORS 00000

ENTER PASS : 1P,2P,2L,2T
```

L