KIM-1

EXTENDED MONITOR

USER'S MANUAL

JANUARY 1982

CONTENTS

INTRODUCTION

     This Extended Monitor is a combination of modifications of the Ohio Scientific (OSI) Extended Monitor 2PEM, originally written to run on the OSI C-2-4P and XIM by Pyramid Data Systems written to run on the KIM-1.
     Additional commands have been added and modified to run on the KIM-1 using the Forthought video driver routine. Load and save routines are in Ziptape format.
     <u>Only two (2) copies of this monitor are in existence and was intended for personal (private) use only.</u>

The Extended Monitor features:

    * 26 command entries
    * 1 User defined command
    * All commands are self-prompting
    * Resident disassembler
    * Resident de-bugging routines
    * Resident string search routines (HEX and ASCII)
    * Resident relocator
    * Resident mover
    * Resident memory test
    * Resident Hex calculator

SYSTEM REQUIREMENTS

     The Extended Monitor requires 3K of memory and uses 48 zero page locations. ($AD - $DC)
     The cassette format is programmed for Ziptape load and save routines.
     The video screen format is programmed for 64 characters by 16 lines.

INITIALIZATION

     Upon initialization the prompt (*) is issued, all break vectors are set for normal operation. Illegal entries default to "?" being issued.
     Leading zeros in all addresses must be used. Address formats are high byte, low byte order.
     The Extended Monitor is normally entered at $C400. This causes the stack pointer to be set (at $01FF) and the breakpoint registers to be initialized. Under certain circumstances, it may be desirable to re-enter the Extended Monitor without initialization. This may be done at $C41C.

MODIFICATIONS

There is one free command letter in the Extended Monitor - V. in advanced (future) applications, a user may desire an additional function. This function must be a machine language routine ending with a RTS. The function's call address is then inserted at the following address: $CCCF. For example, to call a routine at $A125 with V, $CCCF would be set to 25 and $CCD0 to A1.

OPERATION

All number entries into the Extended Monitor must be in hex. Leading zeros in all addresses must be used. All commands are self-prompting. Illegal entries return default prompt ?. If an error is made on entry of data a carriage return will exit the command and re-prompt. All commands are self-executing except the SAVE (S) command which requires pressing the space bar to execute, allowing set-up and/or editing of cassette before execution.
The Memory Test command (Z) requires the space bar depression to exit the command after execution.
Printer controls codes, PRINT (Control P) and NO PRINT (Control Z) are active in all commands. Clear Screen (Control V) is also active.

******* CAUTIONS *******

The Extended Monitor is not relocatable due to the tables contained within.
Zero page locations $C5 through $DC are reserved for breakpoint addresses and removed opcodes. Care should be taken no to damage these locations with the program being debugged.
The Continue command (C) MUST ONLY be used with breakpoint entry into the Extended Monitor.

COMMAND SUMMARY

@   Open memory location NNNN

A   Print breakpoint Accumulator

B   Enter breakpoint n

C   Continue from last breakpoint

D   Dump memory FRM,TO

E   Eliminate breakpoint n

F   Fill memory block FRM,TO=DATA

G   Go/execute at address NNNN

H   Hexadecimal calculator

I   Print location of last break entry

J   Compare memory block START=FRM,TO

K   Print breakpoint Stackpointer

L   Load memory from cassette ID=nn (SPACE sets nn to ID 01)

M   Move memory block NEW=FRM,TO

N   Number/HEX string search HEX>FRM,TO

P   Print breakpoint Status Register

Q   Disassemble from address NNNN

R   Relocate code NEW=FRM,TO

S   Save on cassette memory block ID=nn FRM,TO DUMPS?n

T   Print breakpoint address table

U   Jump to assembler

V   ** USER DEFINED **

W   Word/ASCII string search   ASCII>FRM,TO

X   Print breakpoint X Register

Y   Print breakpoint Y Register

Z   Test Memory FRM,TO

```
********************
@NNNN                                   (Open memory location NNNN)
********************
```

This Command opens 16 bit memory location NNNN.

RESTRICTIONS: None

EXAMPLE: @0200/CC

DISCUSSION:

The Monitor responds with /CC, where CC is the contents of location NNNN. Contents of location NNNN may be changed by typing in 8 bit (byte) hexadecimal data DD. Additional commands in this mode are:

    (SPACE) -   Increments to next location.

    (^)   -   Decrements to previous location.

    (")   -   Prints ASCII or graphic character.

    (CR)   -   Carriage Return exits OPEN MODE.

```
********************
A                                       (Print breakpoint Accumulator)
********************
```

This command prints the contents of the Accumulator at the breakpoint.

RESTRICTIONS: None

EXAMPLE: A/CC

DISCUSSION:

The Monitor responds with /CC, where CC is the contents of the Accumulator. The OPEN MODE is entered and the contents of the Accumulator may then be changed prior to continuing execution of the program.

```
********************
Bn,NNNN                                 (Install breakpoint n at address NNNN)
********************
```

This command installs breakpoint n at hexadecimal address NNNN.

RESTRICTIONS: Only 8 breakpoints may be entered.

EXAMPLE: B1,C415

DISCUSSION:

Number breakpoints 1 through 8. After breakpoint n is encountered it can be re-installed and used again later in the program.

```
********************
C                                       (Continue program execution)
********************
```

        This command is used to continue program execution after a breakpoint
is encountered.

RESTRICTIONS: This command MUST ONLY be used with breakpoint entry.

EXAMPLE: C

DISCUSSION:

        The program in question will continue from the last breakpoint until
another break is encountered or a programming error is encountered.

```
********************
D                                       (Dump memory FRM,TO)
********************
```

        This command dumps a block of memory between addresses specified by FRM
and TO.

RESTRICTIONS: None

EXAMPLE: DC415,D515

DISCUSSION:

        The format displayed/printed is by columns and rows.

```
********************
En                                      (Eliminate breakpoint n)
********************
```

        This command eliminates breakpoint specified by n.

RESTRICTIONS: None

EXAMPLE: E1

DISCUSSION:

        This command restores removed opcodes and eliminates the breakpoint n,
(n being 1 through 8).

```
********************
F FRM,TO=DD                          (Fill memory block with constant)
********************
```

This command fills memory block specified by addresses FRM and TO with hexadecimal date byte DD.

RESTRICTIONS: NONE

EXAMPLE: F0200,0300=FF

DISCUSSION:

Data FF will not be filled into address 0300 in the above example. Data is filled up to and NOT including the ending address.


```
********************
G NNNN                               (Go/execute at address NNNN)
********************
```

This command begins program execution at hexadecimal address specified by NNNN.

RESTRICTIONS: None

EXAMPLE: G0200

DISCUSSION:

The program at location NNNN must be an executable machine code.


```
********************
HDT1,DT2(+,-,/,*)=ANS                (Hexadecimal calculator)
********************
```

This command is a four function hexadecimal calculator.

RESTRICTIONS: None

EXAMPLE: H0200,0200+=0400 or H0200,0200-=0000

DISCUSSION:

This command is a four function hexadecimal calculator, where DT1 and DT2 are 16 bit data fields followed by an operator (+,-,/,*) and ANS is the 16 bit result of the operation.

```
********************
I                                       (Break entry address/registers)
********************
```

This command prints the address at which the Monitor was last entered by a break.

RESTRICTIONS: Defaults to FFFF

EXAMPLE: I

DISCUSSION:

  The Monitor responds with:

@NNNN
A/CC X/CC Y/CC P/CC K/CC

  Where NNNN is the address at which the Extended Monitor was entered and CC is the contents of the Accumulator, X index, Y index, Status Register and Stack-pointer respectively.


```
********************
JNNNN=FRM,TO                            (Compare memory blocks)
********************
```

  This command compares memory with specified blocks.

RESTRICTIONS: None

EXAMPLE: J0200=0300,0400

DISCUSSION:

  Compare memory starting at address NNNN to memory block specified by FRM and TO. The Monitor will respond with the ending address of block NNNN followed by the number of bytes in each block. Corresponding addresses with differing data are then displayed.


```
********************
K                                       (Print stack-pointer contents)
********************
```

  This command prints the contents of the stack-pointer at the breakpoint.

RESTRICTIONS: None

EXAMPLE: K/CC

DISCUSSION:

  The Monitor responds with /CC, where CC is the contents of the stack-pointer. The OPEN MODE is entered and the contents of the stack-pointer may then be changed prior to continuing execution of the program.

```
********************
L                                   (Load data from cassette tape)
********************
```

This command loads data from cassette tape after assigning the proper ID.

RESTRICTIONS: Tape format must be ZIPTAPE

EXAMPLE: L ID=02 or L ID=FF,2100

DISCUSSION:

The Monitor responds with ID=. Assign the appropriate ID. An ID of FF will prompt with a comma, then assign the starting address of the new location of the load. ID 00 will load any ID. If the desired ID is 01, a space bar depression will assume an ID of 01 and activate the load command automatically.

```
********************
M NEW=FRM,TO                        (Move memory block NEW=FRM,TO)
********************
```

This command moves a block of memory non-destructively.

RESTRICTIONS: Ram availability

EXAMPLE: M0200=0400,0450

DISCUSSION:

Move the block of memory between FRM and TO to a block of equal length beginning at the address specified by NEW. The MOVE command is non-destructive and data will not be changed.
Care must be taken that the distance of the forward move is greater than the length of the original block. Otherwise, data in the original block will be overwritten.

```
********************
N HEX>FRM,TO                        (Hex data search)
********************
```

This command searches for a hexadecimal data string.

RESTRICTIONS: 8 byte maximum string

EXAMPLE: N 4C EF 26>2000,4FFF

DISCUSSION:

Searches memory block between addresses FRM and TO for data string HEX. If a string match is found, the Monitor responds with @NNNN/CC, where NNNN is the address of the string and CC is the first byte of the string. The Monitor enters the OPEN MODE and all commands in the OPEN MODE are available.

```
********************
O                                       (Overflow from hexcalculator)
********************
```

        This command prints the overflow/remainder of the hexcalculator.

RESTRICTIONS: None

EXAMPLE: 0

DISCUSSION:

        This command prints the overflow of remainder of the 16 bit
multiplication or division respectively.

```
********************
P                                       (Print breakpoint Status Register)
********************
```

        This command prints the contents of the Status Register at the
breakpoint.

RESTRICTIONS: None

EXAMPLE: P/CC

DISCUSSION:

        The Monitor responds with /CC, where CC is the contents of the Status
Register. The OPEN MODE is entered and the contents of the Status Register
may be changed prior to continuing execution of the program.

```
********************
QNNNN                                   (Disassemble starting at address NNNN)
********************
```

        This command is a 6502 machine code disassembler

RESTRICTIONS: The program must be executable machine code.

EXAMPLE: Q2000

DISCUSSION:

        Disassembles executable 6502 machine code into 6502 mnemonic code.
Disassembly starts at the address specified by NNNN and continues for 15
lines. Non-executable opcodes are assigned a mnemonic of ???. Disassembly may
be continued in 15 line blocks by pressing the space bar. Carriage return
exits the command.

```
********************
R NEW=FRM,TO                            (Relocate a block memory)
********************
```

        This command relocates a block of memory specified by FRM and TO, to a
block of equal length specified by NEW.

RESTRICTIONS: Block must be executable machine code.

EXAMPLE: R0500=0300,0380

DISCUSSION:

        Relocate a block of executable machine code maintaining code
executability. Format is the same as the Move command.
        If the command R0500=0300,0380 was given and the original data read:

```
                    0300  A520         LDA $20
                    0302  206E03       JSR $036E
                    036E  8DC6D0       STA $D0C6
                    037F  60           RTS
```

        At 0500 it will read:

```
                    0500  A520         LDA $20
                    0502  206E05       JSR $056E
                    056E  8DC6D0       STA $D0C6
                    057F  60           RTS
```

        Note that the subroutine call at 0502 was changed to fit the program's
new location. However, if that subroutine had had an address outside the
range of the location (0300 to 0380), it would not have been changed. Such
was the case with the code at location 056E. Note that data tables will not
relocate properly. Cautions for Relocate are the same as MOVE.


```
********************
S ID=nnFRM,TO                           (Save a block of memory to cassette tape)
********************
```

        This command saves a block of memory between addresses specified by FRM
and TO including address TO+1 with Idnn, nn being 01 to FE.

RESTRICTIONS: ZIPTAPE format only. Assign ID of 01 through FE. Maximum number
dumps is 3.

EXAMPLE: S ID=01,0200,03FF
         NUMBER OF DUMPS? 2

DISCUSSION:

        The Monitor will respond with ID=. Type starting address of dump then
the ending address followed by a space. The Monitor will then respond with
NUMBER OF DUMPS?. Type number of dumps (1 to 3). A space bar depression will
then activate the command. A carriage return will exit the command in case of
typing error. ID's of FF and 00 will default with ?.

```
********************
T                                    (Print table of breakpoint addresses)
********************
```

       This command prints the table of breakpoint addresses.

RESTRICTIONS: None

EXAMPLE: T

DISCUSSION:

       Addresses of non-specified breakpoints default with FFFF. The table
prints addresses of B1 through B8.


```
********************
U                                    (Jump to Assembler warm start)
********************
```

       This command jumps to the Assembler/Editor.

RESTRICTIONS: The Assembler/Editor must be preloaded.

EXAMPLE: U

DISCUSSION:

       Break vectors used by the Assembler/Editor are reset with this command.
The file address is set to $4500. If the file is at any location other than
$4500, it then must be initialized by typing FIL ANNNN in the
Assembler/Editor. NNNN being the address of the active file.


```
********************
W ASCII>FRM,TO                       (word/ASCII string search)
********************
```

       This command is identical to N search except the search is made for an
ASCII literal string match.

RESTRICTIONS: 8 byte maximum string

EXAMPLE: W KIM>0200,17FF

DISCUSSION:

       Searches memory block between hexadecimal address FRM and TO for ASCII
string. If a string match is found, the Monitor enters @NNNN/CC, where NNNN
is the address of the string and CC is the first byte of the string. The
Monitor enters into the OPEN MODE and all commands in the OPEN MODE are
available.

```
*******************
X                                       (Print contents of X Register)
*******************
```

        This command prints the contents of the X Register at the breakpoint.

RESTRICTIONS: None

EXAMPLE: X/CC

DISCUSSION:

        The Monitor responds with /CC, where CC is the contents of the X
Register. The OPEN MODE is entered and the contents of the X Register may be
changed prior to continuing execution of the program.


```
********************
Y                                       (Print contents of Y Register)
********************
```

        This command prints the contents of the Y Register at the breakpoint.

RESTRICTIONS: None

EXAMPLE: Y/CC

DISCUSSION:

        Same as for A,X,P and K commands respectively.

```
********************
Z FRM,TO                              (Memory Test)
********************
```

        This command is a #COOK'S MEMORY TEST.

RESTRICTIONS: None

EXAMPLE: Z0200,0501

DISCUSSION:

        Tests block of memory specified by addresses FRM and TO. The Monitor
will respond with + on a good pass. Failure to pass test will issue the
following error codes:

        A - 0's cannot be loaded at address NNNN.
        B - Failure to retain 0's. Possible shorted address line at address

            NNNN
        C - Failure to pass walking bit test at address NNNN bit nn
        D - 1's cannot be loaded at address NNNN.

        If the command Z 0200,0501 was given and there was no memory at address
0500, due to failing ALL tests, the following response would be printed.

```
        A 0500
        B 0500
        C 0500 07
        C 0500 06
        C 0500 05
        C 0500 04
        C 0500 03
        C 0500 02
        C 0500 01
        C 0500 00
        D 0500
```

        Thus reporting memory location 0500 failed all tests.

The format is:

```
        Error code  Address      Faulty bit (Walking bit test only)
        (A-D)       NNNN         nn
```

        The command is exited by depressing the space bar. The error response
can also be exited by the space bar.

#COOK. Author of COOK'S MEMORY TEST.

```
KILOBAUD Magazine:      Oct. '77 p.58
                        Jul. '78 p.70
                        Jun. '80 p.178
```

(N O T E S)

(PROSPECTIVE CHANGES)