OSI BASIC for the KIM-1

BY JONATHAN M. PRIGOT
35 Cummings Road, #6
Boston, Massachusetts 02146

For some time, now, if a KIM-1 user wished to run Microsoft's 8K BASIC on their system, they had their choice of two cassettes. Unfortunately, since both versions seem to use self-modifying code, if the KIM crashes, then the tape must be reloaded. Putting it into ROM would prevent this, but, of course, this cannot be done.

Ohio Scientific Inc. (OSI) makes a line of 6502-based machines that contain their BASIC in four ROMs. If these ROMs could be used on other 6502 systems, the effects of a system crash would be less catastrophic. The problem, however, is to link the ROM BASIC with other machines. The purpose of this article is to present a way to do just that.

The OSI chip set consists of five chips: four 2316 ROMs which contain the 8K BASIC and reside at A000-BFFF, and a 1702 "support PROM" which resides at FF00-FFFF, and contains initialization routines, jumps to I/O routines, and the 6502's NMI, RST, and IRQ vectors. These support routines must be customized for the system the BASIC is to run on.

Some Important Addresses

```
0000-00FF: Zero Page Pointers and routines (USR vector=$B,$C
                                                    (L,H))
0100-01FF: Stack Area
0200-02FF: Normally used by OSI's I/O routines, otherwise
                                                    unused
0300-up  : Source file storage
A000-BFFF: BASIC ROMs
FF00-FFFF: OSI Support ROM which contains the following jumps:
FFEB-FFED: (4C) (Console In Low) (Console In High)
FFEE-FFF0: (4C) (Console Out Low) (Console Out High)
FFF1-FFF3: (4C) (Ctl/C Handler Low) (Ctl/C Handler High)
FFF4-FFF6: (4C) (Cassette Load Low) (Cassette Load High)
FFF7-FFF9: (4C) (Cassette Save Low) (Cassette Save High)
```

Console I/O Routines

The Console In and Out routines must preserve the Y-register. The BASIC also assumes the use of a full-duplex system. If your system, like the KIM, is half-duplex, then the echoed character must be suppressed. Both these routines must end with an RTS (60).

The action of the Control-C Handler depends on the action of the Console In routine. If the Console In routine does not halt and wait for a character input, then simply JMP to $A633. The BASIC will then input a character and test it. If the Console In routine does wait for input, then some means of polling the input port for a character must be used. Upon detection of a character, it must be passed to the BASIC by a JMP to $A636.

Cassette Routines

I decided to use the KIM's resident cassette save/load routines due to their accuracy. Unfortunately, both these routines exit to location $0000. Re-writing these routines to end with an RTS (60) so as to return to the BASIC would take too much space. I therefore decided to set up the save/load routines for two phases each. Phase one gets a file number (01-FE) from the user and saves/loads the BASIC's pointers to the source file. Phase two saves/loads the actual source file.

Since the save/load routines exit to location $0000, each phase sets up this location to point to the next phase. After each phase exits, one simply enters a "G" to go to the next phase. Phase two sets up a jump to the BASIC's warm-start point.

Source File Storage Relocation

The BASIC assumes uninterrupted memory from $0000 on up. If the bulk of your RAM is located elsewhere (like $2000-up), then the BASIC's pointers must be changed to reflect this. For example, if we assume a 4K RAM at $2000-$2FFF, then the locations to change are:

```
$79,$7A-Start of RAM + 1 (01,20)
$7B,$7C-Start of RAM + 4 (04,20)
$7D,$7E-  "    "  "   "     "
$7F,$80-  "    "  "   "     "
$81,$82-End of RAM + 1 (00,30)
$85,$86-  "    "  "   "     "
$8F,$90-Start of RAM (00,20)
$C3,$C4-  "    "  "     "
```

In addition, the first three locations in the text area must be set to zero. Initialize, do a warm-start, and type NEW. This resets all the links to the source file.

I have found that the most useful way to use this program is to set the KIM's NMI vector to point to this monitor. All that is then necessary is to press the ST key to activate the monitor.

```
                        0010  ;SIMPLE OSI/MICROSOFT BASIC SUPPORT MONITOR
                        0020  ;FOR THE KIM-1, VERSION 1.1
                        0030  ;WRITTEN 12/29/78 BY J.M. FRIGOT
                        0040  ;ALL RIGHTS RESERVED
                        0050  ;
                        0060        .OR FE00
                        0070  ;INITIALIZE BY CLEARING BCD FLAG, AND SETING
                        0080  ;STACK POINTER TO $0128
                        0090  ;
FE00  D8        0100  STAR CLD
FE01  A228      0110        LDX 28
FE03  9A        0120        TXS
                        0130  ;
                        0140  ;CLEAR SCREEN AND PRINT MENU, THEN GET USER'S
                        0150  ;CHOICE AND BRANCH APPROPRIATELY.
                        0160  ;
FE04  AD35FE    0170        LDA HEDR          ;LOAD TEXT POINTER
FE07  85EC      0180        STA *TEXT
FE09  AD36FE    0190        LDA HEDR+01
FE0C  85ED      0200        STA *TEXT+01
FE0E  20C9FE    0210        JSR PMPT          ;PRINT MENU
FE11  20E6FE    0220        JSR GETC          ;GET USER'S CHOICE
FE14  20FAFE    0230        JSR ZERO          ;ZERO Y-REG. AND ECHO-SUPPRESS FLAG
                        0240  ;
FE17  C94D      0250        CMP 'M            ;MONITOR?
FE19  D003      0260        BNE WARM
FE1B  4C001C    0270        JMP $1C00         ;JUMP TO KIM MONITOR
FE1E  C957      0280  WARM  CMP 'W            ;WARM-START THE BASIC?
FE20  D003      0290        BNE COLD
FE22  4C0000    0300        JMP $0000         ;BASIC WARM-START ENTRY
FE25  C943      0310  COLD  CMP 'C            ;COLD-START THE BASIC?
FE27  D0D7      0320        BNE STAR          ;INVALID ENTRY, SO RESTART
FE29  4C11BD    0330        JMP $BD11         ;BASIC COLD-START ENTRY
                        0340  ;
FE2C  1AFF      0350  MENU  .HS 1AFF          ;CLEAR SCREEN AND DELAY
FE2E  432F57    0360        .AS 'C/W/M?'      ;MESSAGE TEXT
FE31  2F4D3F
FE34  00        0370        .HS 00            ;END-OF-MESSAGE INDICATOR
FE35  2CFE      0380  HEDR  .SA MENU          ;POINTER TO TEXT
                        0390  ;
                        0400  ;CONTROL-C HANDLER
                        0410  ;
FE37  2C4017    0420  CNTC  BIT $1740         ;POLL TTY INPUT PORT
FE3A  3005      0430        BMI SKIP          ;IF NOTHING THERE, LEAVE
FE3C  A903      0440        LDA 03            ;LOAD CTL/C CHARACTER,
FE3E  4C36A6    0450        JMP $A636         ;AND JUMP TO BASIC HANDLER
FE41  60        0460  SKIP  RTS
                        0470  ;
                        0480  ;CASSETTE SAVE/LOAD ROUTINES
                        0490  ;
FE42  209AFE    0500  SAVE  JSR FILE          ;GET/STORE FILE NUMBER
FE45  A979      0510        LDA 79            ;STORE POINTER AREA ADDRESSES
FE47  8DF517    0520        STA $17F5
FE4A  A991      0530        LDA 91
FE4C  8DF717    0540        STA $17F7
FE4F  A900      0550        LDA 00
FE51  8DF617    0560        STA $17F6
FE54  8DF817    0570        STA $17F8
FE57  AD7FFE    0580        LDA SVAD          ;REDIRECT WARM-START VECTOR
FE5A  8501      0590        STA $01           ;TO PHASE-2 OF SAVE ROUTINE
FE5C  AD80FE    0600        LDA SVAD+01
FE5F  8502      0610        STA $02
FE61  4C..18    0620        JMP $1800         ;STORE POINTER AREA ON CASSETTE
                        0630  ;
FE64  D8        0640  PH2S  CLD               ;SAVE ROUTINE PHASE-2
FE65  A58F      0650        LDA $8F           ;LOAD SOURCE FILE POINTERS
FE67  8DF517    0660        STA $17F5
FE6A  A590      0670        LDA $90
FE6C  8DF617    0680        STA $17F6
FE6F  A57F      0690        LDA $7F
FE71  8DF717    0700        STA $17F7
FE74  A580      0710        LDA $80
FE76  8DF817    0720        STA $17F8

FE79  20C0FE    0730        JSR RSTR          ;RESTORE THE WARM-START VECTOR
FE7C  4C0018    0740        JMP $1800         ;SAVE THE SOURCE FILE ON CASSETTE
                        0750  ;
FE7F  64FE      0760  SVAD  .SA PH2S          ;POINTER TO SAVE PHASE-2 ROUTINE
                        0770  ;
                        0780  ;
FE81  209AFE    0790  LOAD  JSR FILE          ;GET FILE NUMBER
FE84  AD98FE    0800        LDA LOAD          ;REDIRECT WARM-START VECTOR
FE87  8501      0810        STA $01           ;TO PHASE-2 OF LOAD ROUTINE
FE89  AD99FE    0820        LDA LOAD+01
FE8C  8502      0830        STA $02
FE8E  4C7318    0840        JMP $1873         ;RETRIEVE POINTERS FROM CASSETTE
                        0850  ;
FE91  D8        0860  PH2L  CLD               ;LOAD ROUTINE PHASE-2
FE92  20C0FE    0870        JSR RSTR          ;RESTORE THE WARM-START VECTOR
FE95  4C7318    0880        JMP $1873         ;RETRIEVE THE SOURCE FILE
                        0890  ;
FE98  91FE      0900  LOAD  .SA PH2L          ;POINTER TO LOAD PHASE-2 ROUTINE
                        0910  ;
FE9A  D8        0920  FILE  CLD               ;GET/STORE FILE NUMBER ROUTINE
FE9B  ADBEFE    0930        LDA GETF          ;LOAD TEXT POINTERS
FE9E  85EC      0940        STA *TEXT
FEA0  ADBFFE    0950        LDA GETF+01
FEA3  85ED      0960        STA *TEXT+01
FEA5  20C9FE    0970        JSR PMPT          ;PRINT MESSAGE
FEA8  20F2FE    0980        JSR GETB          ;GET THE FILE NUMBER
FEAB  8DF917    0990        STA $17F9         ;STORE IN KIM CASSETTE FILE LOC.
FEAE  60        1000        RTS
                        1010  ;
FEAF  0D0A      1020  FLNM  .HS 0D0A          ;CR/LF
FEB1  46494C    1030        .AS 'FILE NUMBER='
FEB4  45204E
FEB7  554D42
FEBA  45523D
FEBD  00        1040        .HS 00
FEBE  AFFE      1050  GETF  .SA FLNM          ;POINTER TO TEXT
                        1060  ;
FEC0  A974      1070  RSTR  LDA 74            ;RESTORE WARM-START VECTOR
FEC2  8501      1080        STA $01
FEC4  A9A2      1090        LDA 0A2
FEC6  8502      1100        STA $02
FEC8  60        1110        RTS
                        1120  ;
                        1130  ;CONSOLE I/O ROUTINES
                        1140  ;
FEC9  20FAFE    1150  PMPT  JSR ZERO          ;ZERO Y-REG. AND ECHO-SUPP. FLAG
FECC  B1EC      1160  FTCH  LDA (TEXT),Y     ;GET A CHARACTER
FECE  F006      1170        BEQ DONE          ;IF END-OF-MESSAGE, LEAVE
FED0  20D7FE    1180        JSR PRNT          ;PRINT THE CHARACTER
FED3  C8        1190        INY               ;INCREMENT THE POINTER
FED4  D0F6      1200        BNE FTCH          ;GET NEXT CHARACTER
FED6  60        1210  DONE  RTS
                        1220  ;
                        1230  ;
FED7  84F4      1240  PRNT  STY *YSAV         ;SAVE Y-REGISTER
FED9  A4EE      1250        LDY *MFLG         ;GET THE ECHO-SUPPRESS FLAG
FEDB  D003      1260        BNE RSET          ;IF SET, DO NOT PRINT
FEDD  20A01E    1270        JSR $1EA0         ;OTHERWISE, PRINT THE CHARACTER
FEE0  20FAFE    1280  RSET  JSR ZERO          ;RESET ECHO-SUPPRESS FLAG
FEE3  A4F4      1290        LDY *YSAV         ;RESTORE Y-REGISTER
FEE5  60        1300        RTS
                        1310  ;
                        1320  ;
FEE6  84F4      1330  GETC  STY *YSAV         ;SAVE Y-REGISTER
FEE8  A0FF      1340        LDY #FF           ;SET THE ECHO-SUPPRESS FLAG
FEEA  84EE      1350        STY *MFLG
FEEC  205A1E    1360        JSR $1E5A         ;GET CHARACTER FROM TERMINAL
FEEF  A4F4      1370        LDY *YSAV         ;RESTORE Y-REGISTER
FEF1  60        1380        RTS
                        1390  ;
                        1400  ;
```

```
                        * The third pointer opens a file on a
                        * file oriented device
                        * The fourth closes a file on a file
                        * oriented device.
                        * Location $02C5 is a RTS

                        * Device 2 first

010E                            ORG     $010E
010E 3D 76                      FDB     DISKIN
0110 02 C5                      FDB     $02C5
0112 3D 0A                      FDB     OPNRED
0114 3D C0                      FDB     DISKCL

                        * Device Three

0116 02 C5                      FDB     $02C5
0118 3D AA                      FDB     DISKOU
011A 3D D1                      FDB     OPNWRT
011C 3D C0                      FDB     DISKCL

                        * Device Four
                        * pointers to my IBM routines

011E 02 C5                      FDB     $02C5
0120 3E B0                      FDB     $3EB0
0122 3E 80                      FDB     $3E80
0124 02 C5                      FDB     $02C5

                        * Device Five is set up
                        * to return to FLEX

0126 71 03                      FDB     $7103
0128 71 03                      FDB     $7103
012A 71 03                      FDB     $7103
012C 71 03                      FDB     $7103

0140                            ORG     $0140
0140 12 78                      FDB     $1278

0149                            ORG     $0149
0149 3C FE                      FDB     $3CFE

                                END

NO ERROR(S) DETECTED

* Device 1 is terminal
* Its pointers remain the
* same.  We change the
* pointers to the other devices
* in the following code.

* The first pointer returns a character in A.
* The second pointer takes a character from A

   SYMBOL TABLE:
ARG2   002E   CONT1  3D3D   CONTIN 3D3C   DISKCL 3DC0   DISKIN 3D76
DISKOU 3DAA   EOFCHK 3D87   ERROR  3D6B   FCB    7740   FILEXT 3D43
FILNAM 3D06   FMS    7806   FMSCLS 7803   GETNAM 3D14   LISPD  3D00
LSPNAM 3D04   MSG1   3DDD   NIL    0000   OPNFIL 3D5A   OPNRED 3D0A
OPNWRT 3DD1   PSTRNG 7118   RDNAM  3D21   RDORWR 3D03   RERROR 3DA5
RET10  3D82   RETLSP 3D9C   RPTERR 713C   SETEXT 712D   TRUE   06EB
VN     3D02   WARMS  7103   WERROR 3DBB   WRET10 3DB6   XSAV   3D08
```

Continued from pg. 36

```
FEF2  84F4      1410  GETB STY *YSAV          ;SAVE Y-REGISTER
FEF4  209D1F    1420       JSR $1F2D          ;GET BYTE FROM TERMINAL
FEF7  A4F4      1430       LDY *YSAV          ;RESTORE Y-REGISTER
FEF9  60        1440       RTS
                1450  ;
                1460  ;
FEFA  A000      1470  ZERO LDY 00             ;ZERO Y-REGISTER
FEFC  84EE      1480       STY *PFLG          ;RESET ECHO-SUPPRESS FLAG
FEFE  60        1490       RTS
                1500  ;
                1510  ;SYSTEM EQUATES
                1520  ;
                1530  TEXT .DL 00EC
                1540  PFLG .DL 00EE
                1550  YSAV .DL 00F4
                1560  ;
                1570  ;JUMP VECTORS FOR BASIC
                1580  ;
                1590       .OR FFEB
FFEB  4CE6FE    1600  GETV JMP GETC
FFEE  4CD7FE    1610  PUTV JMP PRNT
FFF1  4C37FE    1620  CTLC JMP CNTC
FFF4  4C81FE    1630  LODV JMP LOAD
FFF7  4C42FE    1640  CSAV JMP SAVE
                1650  ;
                1660  ;6502/KIM VECTORS
                1670  ;
FFFA  1C1C      1680  NMIV .HS 1C1C
FFFC  221C      1690  RSTV .HS 221C
FFFE  1F1C      1700  IRQV .HS 1F1C
                1710       .EN

SYMBOL TABLE
STAR    FE00
WARM    FE1E
COLD    FE25
MENU    FE2C
HEDR    FE35
CNTC    FE37
SKIP    FE41
SAVE    FE42
PH2S    FE64
SVAD    FE7F
LOAD    FE81
PH2L    FE91
LDAD    FE98
FILE    FE9A
FLNM    FEAF
GETF    FEBE
RSTR    FEC0
PMPT    FEC9
FTCH    FECC
DONE    FED6
PRNT    FED7
RSET    FEE0
GETC    FEE6
GETB    FEF2
ZERO    FEFA
TEXT    00EC
PFLG    00EE
YSAV    00F4
GETV    FFEB
PUTV    FFEE
CTLC    FFF1
LODV    FFF4
CSAV    FFF7
NMIV    FFFA
RSTV    FFFC
IRQV    FFFE

:
```