```
;********************************************************************************
;
;
;    This file defines EXOS variables, function codes, error codes and software
; interrupt codes as externals.  It should be assembled to a .REL file and
; then linked with any code which uses EXOS.  It produces no object code, just
; defines symbols.
;
;
err_code          defl      100h
;
err               MACRO     name
err_code          defl      err_code - 1
                  PUBLIC    name
name              equ       err_code
                  ENDM
;
;
;
var       MACRO name,number
name      equ       number
          PUBLIC    name
          ENDM
;
;
;
;-------------------------------------------------------------------------------
;
;                      EXOS FIXED VARIABLES
;                      ========================
;
;
          var       USR_P3,         0BFFFh   ;Four segments which were in Z-80 space
          var       USR_P2,         0BFFEh   ; when EXOS was last called.
          var       USR_P1,         0BFFDh
          var       USR_P0,         0BFFCh
;
          var       STACK_LIMIT,    0BFFAh   ;Bottom limit of stack for devices.
;
          var       RST_ADDR,       0BFF8h   ;Warm reset address
;
          var       ST_POINTER,     0BFF6h   ;Address of status line RAM
          var       LP_POINTER,     0BFF4h   ;Address of start of video LPT.
;
          var       PORTB5,         0BFF3h   ;Current contents of Z-80 port 0B5h.
;
          var       FLAG_SOFT_IRQ,  0BFF2h   ;Flag <>0 to cause software interrupt.
;
;
;
;-------------------------------------------------------------------------------
```

```
        var     @RESET,  0              ; Reset system
        var     @OPEN,   1              ; Open channel
        var     @CREAT,  2              ; Create channel
        var     @CLOSE,  3              ; Close channel
        var     @DEST,   4              ; Destroy channel
        var     @RDCH,   5              ; Read character
        var     @RDBLK,  6              ; Read block
        var     @WRCH,   7              ; Write character
        var     @WRBLK,  8              ; Write block
        var     @RSTAT,  9              ; Read status
        var     @SSTAT,  10             ; Set channel status
        var     @SFUNC,  11             ; Special vartion
        var     @EVAR,   16             ; Set/read/toggle EXOS variable
        var     @CAPT,   17             ; Capture channel
        var     @REDIR,  18             ; Re-direct channel
        var     @DDEV,   19             ; Set default device
        var     @SYSS,   20             ; Return system status
        var     @LINK,   21             ; Link device
        var     @READB,  22             ; Read EXOS boundary
        var     @SETB,   23             ; Set USER boundary
        var     @ALLOC,  24             ; Allocate segment
        var     @FREE,   25             ; Free segment
        var     @ROMS,   26             ; Locate ROMs
        var     @BUFF,   27             ; Allocate channel buffer
        var     @ERRMSG, 28             ; Return error message.
```

```
;*************************************************************************
;
;
;                              ERROR CODES
;                              ==================
;
;
;         General errors returned by the EXOS kernel
;
;            err      .IFUNC   ;0FFh   Invalid function code
;            err      .ILLFN   ;0FEh   EXOS function call not allowed
;            err      .INAME   ;0FDh   Invalid name string
;            err      .STACK   ;0FCh   Insufficient stack
;
;            err      .ICHAN   ;0FBh   Invalid channel number (0FFh) or channel
;                              ;          does not exist.
;            err      .NODEV   ;0FAh   Device does not exist   (OPEN/CREATE)
;            err      .CHANX   ;0F9h   Channel already exists (OPEN/CREATE)
;            err      .NOBUF   ;0F8h   No ALLOCATE BUFFER call made (OPEN/CREATE)
;            err      .BADAL   ;0F7h   Bad ALLOCATE BUFFER params. (OPEN/CREATE)
;            err      .NORAM   ;0F6h   Insufficient RAM for buffer.
;            err      .NOVID   ;0F5h   Insufficient video RAM.
;
;            err      .NOSEG   ;0F4h   No free segments (ALLOCATE SEG)
;            err      .ISEG    ;0F3h   Invalid segment (FREE SEG, SET BOUNDARY)
;            err      .IBOUN   ;0F2h   Invalid user boundary (SET USER BOUND)
;            err      .IVAR    ;0F1h   Invalid EXOS variable number
;            err      .IDESC   ;0F0h   Invalid device descriptor type (LINK DEV)
;
;
;
;         General errors returned by various devices
;
;            err      .ISPEC   ;0EFh   Invalid special function code
;            err      .2NDCH   ;0EEh   Attempt to open second channel
;            err      .NOFN    ;0EDh   Function not supported
;            err      .ESC     ;0ECh   Invalid escape character
;            err      .STOP    ;0EBh   Stop key pressed
;
;         public   .SWI
;
;
;
;         File related errors
;
;            err      .NOFIL   ;0EAh   File does not exist
;            err      .EXFIL   ;0E9h   File exists (CREATE)
;            err      .FOPEN   ;0E8h   File already open
;            err      .EOF     ;0E7h   End of file met in read
;            err      .FSIZE   ;0E6h   File is too big
;            err      .FPTR    ;0E5h   Invalid file pointer value.
;            err      .PROT    ;0E4h   Protection violation
;
;
;
```

```
;       Keyboard errors
;***************************************************************
        err     .KFKEY  ;0E3h    Invalid function key number
        err     .KFSPC  ;0E2h    Run out of function key space
;
;
;
;       Sound errors
;
        err     .SENV   ;0E1h    Envelope is too big or number 255.
        err     .SENBF  ;0E0h    Not enough room to define envelope
        err     .SENLO  ;0DFh    Envelope storage requested too small (ie. <2)
        err     .SQFUL  ;0DEh    Sound queue is full (and WAIT_SND <> 0)
;
;
;       Video errors
;
        err     .VROW   ;0DDh    Invalid row number to scroll
        err     .VCURS  ;0DCh    Attempt to move cursor off page
        err     .VCOLR  ;0DBh    Invalid colour passed to INK or PAPER
;
        err     .VSIZE  ;0DAh    Invalid X or Y size to OPEN
        err     .VMODE  ;0D9h    Invalid video mode to OPEN
        err     .VDISP  ;0D8h    Naff parameter to DISPLAY
        err     .VDSP2  ;0D7h    Not enough rows in page to DISPLAY
;
        err     .VBEAM  ;0D6h    Attept to move beam off page
        err     .VLSTY  ;0D5h    Line style too big
        err     .VLMOD  ;0D4h    Line mode too big
        err     .VCHAR  ;0D3h    Can't display character on graphics page
;
;
;
;       Serial errors
;
        err     .BAUD   ;0D2h    Invalid baud rate
;
;
;
;       Editor errors
;
        err     .EVID   ;0D1h    Invalid video page for OPEN.
        err     .EKEY   ;0D0h    Trouble in communicating with keyboard.
        err     .ECURS  ;0CFh    Invalid co-ordinates for positioning cursor.
;
;
;
;       Cassette errors
;
        err     .CCRC   ;0CEh    CRC error from cassette driver
;
```

```
;
;           Network errors
;

        err     .SEROP  ;0CDH    Serial device open - cannot use network
        err     .NOADR  ;0CCH    ADDR_NET not set up
;
;
;
;
printe  macro   e
        .printx         * Smallest error code is e *
        endm
;
;

        IF1
          printe        %err_code
        ENDIF
;
;
;---------------------------------------------------------------------------
;
;                   WARNING CODES
;                   =================
;

        var     .SHARE, 07Fh            ;Shared segment allocated
```

```
;********************************************************************************
;
;
;                           EXOS VARIABLE NUMBERS
;                           =====================
;
;
;
;
;
        var     $IRQ_ENABLE,    0       ; Interrupt enable bits.
;
        var     $FLAG_SIRQ,     1       ; Flag to cause a software interrupt.
        var     $CODE_SIRQ,     2       ; Software Interrupt code.
;
        var     $DEF_TYPE,      3       ; Type of default device.
        var     $DEF_CHAN,      4       ; Default channel number
;
        var     $LOCK_KEY,      5       ; Keyboard lock status.
        var     $CLICK_KEY,     6       ; Key click enable/disable.
        var     $STOP_IRQ,      7       ; Software interrupt on STOP key.
        var     $KEY_IRQ,       8       ; Software interrupt on any key press.
        var     $RATE_KEY,      9       ; Keyboard auto-repeat rate.
        var     $DELAY_KEY,     10      ; Delay before auto-repeat starts.
;
        var     $TAPE_SND,      11      ; Tape sound enable/disable.
;
        var     $WAIT_SND,      12      ; Sound driver waiting if buffer full
        var     $MUTE_SND,      13      ; Sound mute enable/disable.
        var     $BUF_SND,       14      ; Sound envelope storage size.
;
        var     $BAUD_SER,      15      ; Serial baud rate.
        var     $FORM_SER,      16      ; Serial word format.
        var     $ADDR_NET       17      ; Network address of this machine
        var     $NET_IRQ,       18      ; Software interrupt on network.
        var     $CHAN_NET,      19      ; Channel of network interrupt
;
        var     $MODE_VID,      20      ; Video mode.
        var     $COLR_VID,      21      ; Video colour mode.
        var     $X_SIZ_VID,     22      ; Video X page size.
        var     $Y_SIZ_VID,     23      ; Video Y page size.
;
        var     $ST_FLAG,       24      ; Status line displayed flag.
        var     $BORD_VID,      25      ; Border colour.
        var     $BIAS_VID,      26      ; Fixed bias colour.
;
        var     $VID_EDIT,      27      ; Video channel number.
        var     $KEY_EDIT,      28      ; Keyboard channel number.
        var     $BUF_EDIT,      29      ; Size of edit buffer.
        var     $FLG_EDIT,      30      ; Editor control flags
;
        var     $SP_TAPE,       31      ; Cassette I/O speed.
        var     $PROTECT,       32      ; Cassette protection control
        var     $REM1,          33      ; Cassette remote 1
        var     $REM2,          34      ; Cassette remote 2
```

```
;*****************************************************************************
;
;
;               SOFTWARE INTERRUPT CODES
;               ==============================
;
;
        var     ?FKEY,  10h             ;Function keys 10h...1Fh
        var     ?STOP,  20h             ;Stop key
        var     ?KEY,   21h             ;'any key' 21h
;
        var     ?NET,   30h             ;Network data received.
;
;
;
;
```