DISK I/O APPLICATION NOTE

In order to interface your disk to FLEX-FORTH, you will need to create a new R/W routine. The purpose of this routine is to take the screen # off the stack, and determine the corresponding track/sector position on the disk. For example, if your disk has 77 tracks with 16 sectors of 256 bytes per track, then screen 0 would coorespond to track 00, sectors 0 through 3 (4 sectors per screen) and screen 18 would be stored on track 04, sectors 8 through 11. The R/W routine that does this conversion is usually written in FORTH, and supplies the track/sector information to a CODE (machine language) routine. The CODE routine makes this data available to your DOS, and does a JSR, to a DOS subroutine that writes or reads 4 sectors (1024 bytes) to disk.

There are two routines in FORTH which use R/W: BLOCK and BUFFER. When you have completed your R/W routine (and CODE routine), you must insert it's code field address into BLOCK and BUFFER. To get this address, type

```
HEX  <return>
' R/W  <return>
2 - .  <return>
```

The ' (tick) leaves the parameter field address of R/W, which is two bytes ahead of the code field address. The 2 - . sequence will print the actual code field address. Then you can break out of FLEX-FORTH by typing

```
MON  <return>   (IRQ vector must be set up)
```

and store the code field address (CFA) in BUFFER and BLOCK (addr 31E8,31E9 and addr 31AC,31AD). As usual, store the low byte followed by the hi byte.

BUFFER and BLOCK were originally pointing to the KIM tape R/W routine. Now, FLEX-FORTH will use your disk routine, with no other modifications.

SCR59 318C   SCR60 31C2

BUFFER and BLOCK leave three data items on the stack when they call R/W. These are:

each item 2 bytes ie 16 bit signed

Stack position	name	description
Top	flag	flag=0 for write, 1 for read.
2nd	blk	the screen # (block #).
3rd	addr	the address of the block buffer (1024 bytes)

Note: FLEX-FORTH uses most of Z-page for the data stack and pointers, and also uses $0100-$01FF for the terminal input buffer and return stack. If your DOS uses this area, you will need to write a routine to save it.