

Joel Gluck asked me how to use interrupts from an ACTION program. Here are two programs which, respectively, deal with the DLI and VBLD interrupts.

Notice that, since Action is not re-entrant, you can't call subroutines, or do anything else which might mess up a memory location that the main program is depending upon. For example, multiply and divide are both done by subroutines, so you can't use them within a VBLANK routine. It is, however, safe to add, subtract, index an array, and store. (But be sure to clear the decimal flag if your main program might be calling the floating point ROMs!)

```
;
; Example of use of display list
; interrupt from Action
;
```

```
BYTE    VCOUNT = $D40B,
        WSYNC = $D40A,
        NMIEN = $D40E,
        CH = $2FC,
        COLPF2 = $D018
```

```
CARD    VDSLST = $200,
        SDLST = $230,
        OLDVEC
```

```
DEFINE PHA = "$48",
        PLA = "$68",
        TAX = "$AA",
        TAY = "$A8",
        TXA = "$8A",
        TYA = "$98",
        RTI = "$40"
```

```
;
; During a DLI you can't call any
; other functions, nor multiply,
; nor divide
;
```

```
PROC DLI()
  [PHA TYA PHA TXA PHA]
  WSYNC = VCOUNT
  COLPF2 = VCOUNT
  [PLA TAX PLA TAY PLA RTI]
```

```
PROC MAIN()
  BYTE I
  BYTE POINTER TEMP
  PRINTE("Setting up DLI")

  NMIEN = $40 ;DISABLE DLI

  OLDVEC = VDSLST
  VDSLST = DLI

  TEMP = SDLST+3
  TEMP^ = $C2

  FOR I = 1 TO 23
  DO
```

```

    TEMP = SDLST + I + 5
    TEMP^ = $82
OD

NMIEN = $C0 ;ENABLE DLI

WHILE CH = $FF DO
    PRINTE("Press any key to quit")
OD
CH = $FF ;Swallow key press

PrintE("Restoring DLI")

NMIEN = $40 ;DISABLE DLI
TEMP = SDLST+3
TEMP^ = $42

FOR I = 1 TO 23
DO
    TEMP = SDLST + I + 5
    TEMP^ = $02
OD

VDSLST = OLDVEC
PRINTE("Returning")
RETURN

```

VBL.ACT

```

;
; Example of using the vertical blank
; deferred interrupt from Action
;

BYTE    RTCLOCK = 20,
        CH = $2FC,
        COLOR2 = $2C6

CARD    VVBLKD = $224,
        SDLST = $230,
        OLDVEC

DEFINE JMPI = "$6C"

;
; Within a VBI you can't call any
; subroutines, nor can you multiply
; or divide. . . .

PROC VBLANKD()
    COLOR2 = RTCLOCK
    [JMPI OLDVEC]

;Simulate the OS call SETVBV

PROC SETVBV(BYTE WHICH
            CARD ADDR)
    CARD POINTER TEMP

```

```
BYTE V
TEMP = $216 + (WHICH LSH 1)
V = RTCLOCK+ 1
WHILE V <> RTCLOCK DO OD
TEMP^ = ADDR
RETURN
```

```
PROC MAIN()
  BYTE OLDC2

  OLDC2 = COLOR2
  PRINTE("Setting up Vblank")

  OLDVEC = VVBLKD
  SETVBV(7, VBLANKD)

  WHILE CH = $FF DO
    PRINTE("Press any key to quit")
  OD
  CH = $FF ;Swallow key press

  PrintE("Restoring Vblank")

  Setvbv(7, OLDVEC)
  COLOR2 = OLDC2
  PRINTE("Returning")
RETURN
```