

Handling IRQs macros

Assemble with ACME.

Just notice how the macro ENTER and EXIT are used, to make nice clean demosource with as many IRQ as you need.

```
;some macros to use for easy raster handling, by rambones

!to "part1.prg"

!zone mainprogram
*=$1000

;----- MACROS -----
; !macro INIT .inadd, .pladd{
; (code here)
;}

!macro ENTER{
pha
tya
pha
txa
pha
}

!macro EXIT .intvector, .rasterline{
LDX #>.intvector
LDY #<.intvector
STX $FFFF
STY $FFFE
LDA #.rasterline
STA $D012
SEC
ROL $D019
JMP _quitirq
}

!macro POKE .value, .address{
LDA .value
STA .address
}

!macro XDEL .pausex{
LDX #.pausex
_xxpause DEX
```

```
BNE _xxpause
}

!macro YDEL .pausey{
  LDY #.pausey
  _pause2 DEY
  BNE _pause2
}

;-----
----
; start of program..

JMP START

; utilities and pointers..

_quitirq
pla
tax
pla
tay
pla
_freeze
rti

_spritepoint
!BYTE 200,201,202,203,204,205,206,207

_xsprite
!BYTE 100,120,140,160,180,200,220,240

_ysprite
!BYTE 100,100,100,100,100,100,100,100

SCREEN=$0400
ZP=$2B

;----- MAIN START -----

START

jsr _clearscreen
jsr _setuplogo
jsr _setlogocolor
jsr SSINIT          ;charscroll
jsr _clearline
```

```
SEI
LDA #35
STA 01
LDX #>INT1
LDY #<INT1
STX $FFFF
STY $FFFE
ldx #>_freeze
ldy #<_freeze
stx $FFFA
sty $FFFB
LDX #0
STX $DC0E
INX
STX $D01A
LDA #1B
STA $D011
LDA #LINE1
STA $D012
CLI
LOCK
JMP LOCK

;-----
LINE1=32
INT1
+ENTER

    ldx #7
    .time5 dex
    bne .time5

    lda #1
    sta $d020
    lda #0
    sta $d021

    JSR SSSET2           ;stop charscroll
;set logofont $2800
LDA $D018
AND #240
ORA #10
STA $D018

;set multicolors on charlogo
lda #2
sta $d022
lda #4
sta $d023
```

```
;set multi color text mode
lda $d016
ora #16
sta $d016

;enable extended text background color
;lda $d011
;ora #64
;sta $d011

+EXIT INT2,LINE2

;-----
;set sprites here

LINE2=$4a
INT2
+ENTER

ldx #7
.time1 dex
bne .time1

JSR SSSET2          ;stop charscroll

lda #5
sta $d020
sta $d021

lda #255
sta $d015

lda #1
sta $d027
sta $d028
sta $d029
sta $d02a
sta $d02b
sta $d02c
sta $d02d
sta $d02e

;ok
ldx #0
.spri2 lda _spritepoint,x
sta $07f8,x
inx
```

```

cpx #7
bne .spri2

lda #100
sta $d000
lda #100
sta $d001

; ldx #0
;.spri4 lda _ysprite,x
; sta $d001,x
; inx
; inx
; cpx #7
; bne .spri4

+EXIT INT3,LINE3

;-----
LINE3=$c8
INT3
+ENTER

ldx #7
.time2 dex
bne .time2

lda #1
sta $d020
lda #0
sta $d021

JSR SSSET2      ;stop charscroll
JSR SSCALC      ;calc charscroll

+EXIT INT4,LINE4

;-----
LINE4=$f1
INT4
+ENTER

ldx #7
.time7 dex
bne .time7

lda #2
sta $d020
lda #6
sta $d021

```

```
lda #22
sta $d018

;set single color text mode
lda $d016
and #239
sta $d016

jsr SSSET1          ;scroll char

+EXIT INT1,LINE1

;-----
_clearscreen

;here go all the subroutines...

!endoffile
```

From:
<https://codebase64.org/> - Codebase 64 wiki

Permanent link:
https://codebase64.org/doku.php?id=base:handling_irqs_with_some_simple_macros

Last update: **2015-04-17 04:32**

