Misc useful ACTION! Functions#

General Information

Author: Carsten Strotmann
Language: ACTION!
Compiler/Interpreter: ACTION!

```
;*****************************
;**                         **
;** PHOENIX SOFTCREW         **
;** STANDARTROUTINEN         **
;** DIVERSES "DIVERS.INC"    **
;*****************************


MODULE

BYTE err,iostat=$23
;-----------------------------------
; clear screen

PROC Cls ()

    Put (125)

RETURN

;-----------------------------------
; cursor enabled

PROC C_On ()

    BYTE crsin=752

    crsin=0

RETURN

;-----------------------------------
; cursor disabled

PROC C_Off ()

    BYTE crsin=752

    crsin=1

RETURN

;-----------------------------------
; wait some time

PROC Pause (CARD times)

    BYTE wsync=$14,q
    CARD u
```

```
    FOR u=1 TO times
    DO
      FOR q=1 TO 200
      DO
    wsync=q
      OD
    OD

RETURN

;-----------------------------------
; beep tone

PROC Beep (BYTE times)

    BYTE u

    FOR u= 1 TO times
    DO
      PutD (0,253)
      Pause (10)
    OD

RETURN

;-----------------------------------
; ATARI Rainbow effekt

PROC Rainbow (BYTE col,INT direc)

BYTE rtclok=$14,
      wsync=$D40A,
      vcount=$D40B,
      key=764,d,e

BYTE ARRAY color(4)=$D016

key=255

WHILE key=255
DO
  IF e#rtclok THEN
     e=rtclok
     d=0
  FI
  IF direc=0 THEN
    d=vcount
    e=0
  ELSE
    d==+direc
  FI
  wsync=rtclok
  color(col)=e+d
OD

RETURN

;-----------------------------------
; Coldstart
```

```
PROC Boot=$E477 ()

;-----------------------------------
;  enable / disable ANTIC DMA

PROC Scr ()

  BYTE sdmctl=$22F

  sdmctl==!$20

RETURN

;-----------------------------------
; query HELP-Function Key

BYTE FUNC Help ()

BYTE hplflg=$2DC

RETURN (hplflg)

;-----------------------------------
; get key

BYTE FUNC Inkey ()

  BYTE atascii=$2FB,chasci=$2FC
  BYTE POINTER keydefp
  CARD keydef=$79

  chasci=$FF
  keydefp=keydef

  DO
  ;
  UNTIL chasci#$FF
  OD

  keydefp==+chasci
  atascii=keydefp^
  chasci=$FF

RETURN (atascii)

;-----------------------------------
; disable attract mode
; (screensaver)

PROC Noattr ()

BYTE attr=$4D

attr=0

RETURN

;-----------------------------------
```

```
; swap two card vars

PROC Swap (CARD a,b)

  CARD x

  x=a
  a=b
  b=x

RETURN

;-----------------------------------
; returns lowest

CARD FUNC Mini (CARD a,b)

  CARD result

  IF a<b THEN
    result=a
  ELSE
    result=b
  FI

RETURN (result)

;-----------------------------------
; return highest

CARD FUNC Maxi (CARD a,b)

  CARD result

  IF a>b THEN
    result=a
  ELSE
    result=b
  FI

RETURN (result)

;-----------------------------------
; set VBI Routine

PROC SETVBV=$E45C (BYTE mode,high,low)

PROC SetVbi (CARD vektor)

 SETVBV (7,vektor/$100,vektor MOD $100)

RETURN

;-----------------------------------
; reset VBI to OS VBI Vector

PROC RstVbi ()

  SetVbi ($E462)
```

```
RETURN

;---------------------------------
; set a bit in a byte

BYTE FUNC SetBit (BYTE value,bit)

  BYTE dumm

  dumm=1
  dumm==LSH bit
  value==%dumm
  PrintBE (value)

RETURN (value)

;---------------------------------
; clears a bit in a byte

BYTE FUNC ClearBit (BYTE value,bit)

  BYTE dumm

  dumm=1
  dumm==LSH bit
  dumm==!$FF
  value==&dumm

RETURN (value)

;---------------------------------
; query if bit is set

BYTE FUNC AskBit (BYTE value,bit)

  BYTE dumm

  dumm=1
  dumm==LSH bit
  value==&dumm
  IF value>0 THEN
    value=1
  FI

RETURN (value)

;---------------------------------
; jump to DOS

PROC Dosin ()


 [ $6C $0C $00 ]

;---------------------------------
; disable BREAK-Key

PROC BreakOff ()
```

```
  CARD brkky=$236

  brkky==+$C

RETURN

;-----------------------------------
; Error-Handler

PROC Errhand ()

  IF iostat>$7F THEN
    err=iostat
  ELSE
    err=0
  FI

RETURN

;-----------------------------------
; print Error Message with Errorcode

PROC Errmess (CARD x,BYTE y)

 IF err>0 THEN
  PutE ()
  Print ("Error - ")
  PrintB (err)
 FI

RETURN

;-----------------------------------
; close all CIO Buffer

PROC AllClose ()

  BYTE u

  FOR u=1 TO 7
  DO
    Close (u)
  OD

RETURN

;-----------------------------------
; ASCII to Internal

BYTE FUNC Inter (BYTE b)
 IF b>=0 AND b<32 THEN
  b==+64
 ELSEIF b>31 AND b<96 THEN
  b==-32
 ELSEIF b>127 AND b<160 THEN
  b==+64
 ELSEIF b>159 AND b<224 THEN
  b==-32
```

```
 FI
RETURN (b)

;-----------------------------------
; Internal to ASCII

BYTE FUNC Ascii (BYTE b)
 IF b>=0 AND b<64 THEN
  b==+32
 ELSEIF b>63 AND b<96 THEN
  b==-64
 ELSEIF b>127 AND b<192 THEN
  b==+32
 ELSEIF b>191 AND b<224 THEN
  b==-64
 FI
RETURN (b)

;-----------------------------------
; Wait for VBI

PROC Wvbi ()

 BYTE rtclk=$14,u

 u=rtclk
 DO
 ;
 UNTIL rtclk#u
 OD

RETURN

;-----------------------------------
; Return Highbyte

BYTE FUNC High (CARD value)

 BYTE ret

 ret=value/$100

RETURN (ret)

;-----------------------------------
; return Lowbyte

BYTE FUNC Low (CARD value)

RETURN (value)

;-----------------------------------
; Clicksound
PROC Click (BYTE time)

 CARD u

 Sound (0,30,10,8)
 FOR u=1 TO time*100
```

```
 DO : OD
 Sound (0,0,0,0)

RETURN

;---------------------------------
; Scan Screen at position

BYTE FUNC Scan (BYTE x,y)

 BYTE chr
 BYTE POINTER msc
 CARD savmsc=$58

 msc=savmsc+y*40+x
 chr=msc^

RETURN (chr)
```