Help for the 'ConLib.bas' include file

This is my 'ConLib.bas' include file.

It is a console mode emulator that works in RFO-BASIC! graphic mode.
using <gr.bitmap.create>

Works with 'typeface 2' monospace as this font as a fixed-pitch font, the letters
and characters each occupy the same amount of horizontal space

It is important to insert the 'ConLib.bas' include file in the right place, as shown below.

```
!-----------------------------------------------------------------------
di_height = 1120 % set to your own device
di_width = 720
!-----------------------------------------------------------------------
gr.open 255,0,0,0 % Black background colour

gr.color 255,255,255,255,1 % White foreground colour

gr.orientation 1  %  1 = Portrait  0 = Landscape.

pause 1000

WakeLock 3
!-----------------------------------------------------------------------
gr.screen screenWidth,screenHeight

scale_x=screenWidth/di_width

scale_y=screenHeight/di_height

gr.scale scale_x,scale_y
!-----------------------------------------------------------------------
!
!-----------------------------------------------------------------------
! Setup screen for drawing.
!-----------------------------------------------------------------------
gr.bitmap.create screen, di_width, di_height

gr.bitmap.draw    nn, screen, 0, 0

gr.bitmap.drawinto.start screen
!-----------------------------------------------------------------------
!
```

!--
! Include ConLib.bas after opening graphics
! and setting up the bitmap screen and before creating any bundles

include ConLib.bas % include 'ConLib.bas' here
!--
!
!--
! if you are using bundle's put them here, after the ConLib.bas included file

! 'global' bundle pointer = 2.
Example Below.
!--
bundle.create global % bundle pointer = 2.

bundle.put global, "Divice_W", di_width
!--
!
!--
The Comands:

In RFO Basic console mode a Newline characters (a CR/LF, or carriage return/line feed, combination) may be inserted into a string with the escape sequence **\n**.
Print "Jim\nGiudice"

In graphic mode this causes problems, so to enter a Newline characters use /n.
Example: Con_Print("RFO/nBasic")

!--
Con_TextSize(status)
!--
Sets the text size. It also sets some internal variables, so set this before anything else.
Example: Con_TextSize(30).
!--
!
!--
Con_ChrPerLine(status)
!--
An alternative to Con_TextSize(status) is Con_ChrPerLine(status)
Sets the number of characters per line. It also sets some internal variables, so set this before anything else
Example: Con_ChrPerLine(40). Will set the text size so that it will print 40 characters across the screen.
!--
!
!--

Con_Locate(text_x, text_y)

!----------

Move to text_x and text_y. The next characters will be printed there.

Example: Con_Locate(5, 10)

!----------

!

!----------

Con_Print(t$)

!----------

Prints the string 't$' at the current location, this could have been set with Con_Locate(x, y) or at the top of the screen if just started up, or at the top of the screen if it has just been cleared.

If the text gets to the end of a line it carrys on, on the line below. If the text gets to the bottom of the screen, the screen scroll up a line, providing Con_Scroll(s) is set to on.

Example: Con_Print("RFO Basic!")

!----------

!

!----------

Con_PrintAt(x, y, t$)

!----------

Prints the string 't$' at the location, x, y

Example: Con_PrintAt(10 ,5, "RFO Basic!")

!----------

!

!----------

Con_PrintString(n, s$)

!----------

You can use this function to print repeated copies of a string. It is useful for printing headings or underlinings.

Example: Con_PrintString(4,"-=*=-")

!----------

!

!----------

Con_PrintSpc(n)

Print n number of spaces in the current background colour.

Example: Con_PrintSpc(10)

!----------

!

!----------

Con_RenderOff()

!----------

Turn rendering off to speeds up printing.

Nothing will be seen until rendering is turned on

Example: Con_RenderOff()

!----------

!

!----------

Con_RenderOn()

!----------

Turn rendering on. Printing will be seen.

Example: Con_RenderOn()

!----------

!

!----------

Con_Render()

!----------

If rendering is off, this command will render the screen once, but not turn rendering on permanently.

Example: Con_Render()

!----------

!

!----------

Con_Italics(status)

!----------

Turn italics printing on or off

Example: Con_Italics(1), turns on italics.

Con_Italics(0), turns off italics.

!----------

!

!----------

Con_Underline(status)

!----------

Turns on or off Underline printing

Example: Con_Underline(1), turns on underline.

Con_Underline(0), turns off underline.

!----------

!

!----------

Con_Strike(status)

!----------

Turns on or off strike printing

Example: Con_Strike(1), turns on Strike.

Con_Strike(0), turns off Strike.

!----------

!

!----------

Con_Bold(status)

!----------

Turns on or off Bold printing

Example: Con_Bold(1), turns on Bold.

Con_Bold(0), turns off Bold.

!----------

!

!----------

Con_Scroll(status)

!----------

At start up scrolling is on. Use this command to turn scrolling on and off.

You may want to turn scrolling off, if you and going to print on the bottom right of the screen and do not want the screen to scroll.

Example: Con_Scroll(1), turns on Scrolling.

Con_Scroll(0), turns off Scrolling.

!----------

!

!----------

Con_Cls()

!----------

Clears the screen to the current background colour.

Example: Con_Cls()

!----------

!

!----------

Con_Background(colourNumber)

!----------

Set the background colour to one of the 18 background colours.

After a change of background colour any characters printed will have that colour background. The whole of the screen will not change to the new background colour until the command Con_Cls() is used.

Example: Con_Background(13), (Blue)

!----------

!

!----------

Con_BackgroundName(ColourName$)

!----------

As Con_Background(colourNumber), but use one of the colour names.

Example: Con_BackgroundName("Blue")

!----------

!

!----------

Con_Colour(colourNumber)

!----------

Set the foreground colour to one of the 18 foreground colours.
Example: Con_Colour(17), (Gold)
!----------
!
!----------
Con_ColourName(colourName$)
!----------
As Con_Colour(colourNumber), but use one of the colour names.
Example: Con_Colour("Red")
!----------
!
!----------
Con_SaveScreen()
!----------
Save the screen to a file at "../../ConScreens/"
The file name is set with the time functions, so will be unique.
The file will be saved as a PNG file.
Example: Con_SaveScreen()
!----------
!
!----------
Con_GetTouchX()
!----------
Returns X position touch of the screen.
This command is scaled to the text width, so touching column 10,
will return 10.
Example: Con_GetTouchX()
!----------
!
!----------
Con_GetTouchY()
!----------
Returns Y position touch of the screen.
This command is scaled to the text height , so touching row 12,
will return 12.
Example: Con_GetTouchY()
do : x = Con_GetTouchX() : until x
do : y = Con_GetTouchX() : until y
Con_Print("You touched column " + int$(x) + " and row " + int$(y))
!----------
!
!----------
Con_GetMaxX()

!---
Returns the maxim character per line.
Call this command after setting Con_TextSize() or Con_ChrPerLine()
Example: x = Con_GetMaxX()
!---
!
!---
Con_GetMaxY()
!---
Returns the maxim rows from the top to the bottom of the screen.
Call this command after setting Con_TextSize() or Con_ChrPerLine()
Example: y = Con_GetMaxY()
!---
!
!---
Con_GetCurrenX()
!---
Returns the current column number.
Example: x = Con_GetCurrentX()
!---
!
!---
Con_GetCurrenY()
!---
Returns the current row number.
Example: y = Con_GetCurrentY()
Con_Locate(10, 12)
column = Con_GetCurrenX() : row = Con_GetCurrenY()
column would hold the value 10 and row would hold the value 12
!---
!
!---
The are 18 colours:
!---

0 = Black, 1 = Grey, 2 = Silver, 3 = White, 4 = Maroon, 5 = Red, 6 = Olive, 7 = Yellow,

8 = Green, 9 = Lime, 10 = Teal, 11 = Aqua, 12 = Navy, 13 = Blue, 14 = Purple,

15 = Fuchsia, 16 = Brown, 17 = Gold.
!---