Print64x32.bas

This 64 column printing routine allows text to be 4 pixels wide instead of 8. It is NOT proportional printing, but this is still useful for lining things up in columns. This further enhances the screen by allowing 32 rows of 6 pixels instead of 24 rows of 8 pixels. This allows 2048 character positions on the screen.

Note that the screen tables file, needed for inclusion, can be downloaded from the forum thread http://www.boriel.com/forum/wishlist/64-char-print-32-lines-version-t680.html

Usage

```
Print64x32At(y,x)
```

Moves the print64 system's print cursor to row Y, column X. Note that 0<= x <= 63 - that is the range of values for X can be up to 63. The range of values for Y is 0-31. * Note that the `print64x32` system's cursor position is independent from that of the ZX Basic Print routine, or any other, such as the print42 system.

```
Print64x32(STRING)
```

Prints the string to the screen at the current Print64 co-ordinates. It does so in the current permanent colours.

```
Print64x32StringAt(y,x,text$)
```

An All in one function - sets the AT position, and prints the string.

> NOTE: The ZX Spectrum's attribute system is encoded into the hardware as a 32x24 character grid. This current version of print64x32 does not touch the attributes, as they would not line up well with the text.

CODE

```
SUB Print64x32At(Y as uByte,X as UByte)
 Poke(@Print64x32XCoord),X
 Poke (@Print64x32XCoord+1),Y
END SUB

SUB Print64x32StringAt(y as ubyte,x as ubyte,text$ as string)
   Print64x32At(y,x)
   Print64x32(text$)
END SUB

SUB Print64x32 (info as String)
ASM
; HL Points to string. Length, followed by bytes.
    LD C,(HL)
    INC HL
    LD B,(HL)
    INC HL
; BC Now contains our string length.
; HL Now points to first character.

Print64x32StringLoop:
    LD A,(HL)

    PUSH BC
    PUSH HL
    CALL Print64x32Char
    POP HL
    POP BC
    Call Print64x32UpdateCoordinates

    DEC BC
    LD A,B
    OR C
    JP Z,Print64x32End

    INC HL
    JP Print64x32StringLoop
    JP Print64x32End

Print64x32Char:
; Arrives with Character in A
    LD L,A                      ; HL=A*3
    LD H,0                      ; |
    ADD HL,HL                   ; |
    ADD A,L                     ; |
    LD  L,A                     ; |
    JR NC, Print64x32CharHop1   ; |
    INC H                       ; |

Print64x32CharHop1:             ; v
    LD DE,Print64x32CharSet-96  ;Offset is because space=32, and it's a 3 byte table - so we go 96 byt
    ADD HL,DE
; HL now points to Correct Character.

    LD BC,(Print64x32_X_Coord) ; Loads Y,X into BC
    LD A,B    ; B=B*6
    ADD A,A   ; |
    LD B,A    ; |
    ADD A,A   ; |
    ADD A,B   ; |
    LD B,A    ; v
; B now has 0-191 value for Y. C has X value in 0-63 format.

    CALL Print64x32ScreenAddress
; DE now points at the screen address we need.
    LD A,C
    AND 1
    LD A,%00001111
    JP NZ,Print64x32RightSide

Print64x32LeftSide:
    EXX
    LD D,3

Print64x32LeftSideLoop:
    EXX
    LD A,(HL)
    AND %11110000
    EXX
    LD E,A
    EXX
    LD A,(DE)
    AND %00001111
    EXX
    OR E
    EXX
    LD (DE),A
    INC B
    CALL Print64x32ScreenAddress
    ; HL Now has 2nd Line

    LD A,(HL)
    AND %00001111 ; Grab second four bits.
    RLCA          ; Push to left side of byte.
    RLCA
    RLCA
    RLCA

    EXX
    LD C,A
    EXX
    LD A,(DE)
    AND %00001111
    EXX
    OR C
    EXX
    LD (DE),A

    INC HL
    INC B
    CALL Print64x32ScreenAddress

    EXX
    DEC D
    JR NZ, Print64x32LeftSideLoop
    EXX

    RET

Print64x32RightSide:
    EXX
    LD D,3

Print64x32RightSideLoop:
    EXX

    LD A,(HL)
    AND %11110000
    RRCA        ;Push to right side.
    RRCA
    RRCA
    RRCA

    EXX
    LD E,A
    EXX
    LD A,(DE)
    AND %1111000
    EXX
    OR E
    EXX
    LD (DE),A

    INC B
    CALL Print64x32ScreenAddress
    ; HL Now has 2nd Line

    LD A,(HL)
    AND %00001111 ; Grab second four bits.

    EXX
    LD C,A
    EXX
    LD A,(DE)
    AND %11110000
    EXX
    OR C
    EXX
    LD (DE),A

    INC HL
    INC B
    CALL Print64x32ScreenAddress

    EXX
    DEC D
    JR NZ, Print64x32RightSideLoop
    EXX
RET


; Screen address.
Print64x32ScreenAddress:

    EX DE,HL
    LD H,ScreenTables/256
    LD L,B
    LD A,(HL)
    INC H
    LD L,(HL)
    LD H,A

    LD A,C
    SRL A ; Divide A(xcoord) by 2.
    ADD A,L
    LD L,A
    EX DE,HL
    RET

; Update Coordinates
Print64x32UpdateCoordinates:
    LD A,(Print64x32_X_Coord)
    INC A
    CP 64
    JR Z,Print64x32OffLine
    LD (Print64x32_X_Coord),A
    RET

Print64x32OffLine:
    XOR A

    LD (Print64x32_X_Coord),A
    LD A,(Print64x32_Y_Coord)
    INC A
    CP 33
    JR Z, Print64x32OffScreen
    LD (Print64x32_Y_Coord),A
    RET

; Could scroll instead? Right now go back to top.
Print64x32OffScreen:
    XOR A
    LD (Print64x32_Y_Coord),A
    RET

Print64x32End:
```

```
END ASM

RETURN

Print64x32XCoord:
ASM
; Variables
Print64x32_X_Coord:
DEFB 1
Print64x32_Y_Coord:
DEFB 10

#INCLUDE ONCE "ScreenTables.asm"

Print64x32CharSet:
DEFB 0,0,0      ; SPACE
DEFB 34,32,32   ; !
DEFB 85,0,0     ; "
DEFB 87,87,80   ; #
DEFB 54,35,96   ; $
DEFB 65,36,16   ; %
DEFB 53,37,96   ; &
DEFB 18,0,0     ; '
DEFB 36,68,32   ; (
DEFB 33,17,32   ; )
DEFB 82,114,80  ; *
DEFB 2,114,0    ; +
DEFB 0,2,64     ; ,
DEFB 0,112,0    ; -
DEFB 0,0,32     ; .
DEFB 17,36,64   ; /
DEFB 37,85,32   ; 0
DEFB 38,34,112  ; 1
DEFB 37,18,112  ; 2
DEFB 97,97,96   ; 3
DEFB 19,87,16   ; 4
DEFB 116,97,96  ; 5
DEFB 52,101,32  ; 6
DEFB 113,18,64  ; 7
DEFB 37,37,32   ; 8
DEFB 37,49,96   ; 9
DEFB 2,2,0      ; :
DEFB 2,2,64     ; ;
DEFB 18,66,16   ; <
DEFB 7,7,0      ; =
DEFB 66,18,64   ; >
DEFB 97,32,32   ; ?
DEFB 97,53,112  ; @
DEFB 37,117,80  ; A
DEFB 101,101,96 ; B
DEFB 37,69,32   ; C
DEFB 101,85,96  ; D
DEFB 116,116,112; E
DEFB 116,116,64 ; F
DEFB 37,69,112  ; G
DEFB 85,117,80  ; H
DEFB 114,34,112 ; I
DEFB 17,21,32   ; J
DEFB 85,102,80  ; K
DEFB 68,68,112  ; L
DEFB 87,85,80   ; M
DEFB 101,85,80  ; N
DEFB 117,85,112 ; O
DEFB 101,100,64 ; P
DEFB 37,85,48   ; Q
DEFB 101,102,80 ; R
DEFB 52,33,96   ; S
DEFB 114,34,32  ; T
DEFB 85,85,96   ; U
DEFB 85,85,32   ; V
DEFB 85,87,80   ; W
DEFB 85,37,80   ; X
```

v: latest

```
DEFB 85,34,32   ; Y
DEFB 113,36,112 ; Z
DEFB 100,68,96  ; [
DEFB 68,33,16   ; \
DEFB 49,17,48   ; ]
DEFB 39,34,32   ; ^
DEFB 0,0,15 ; _
DEFB 37,100,112 ; £
DEFB 6,53,112   ; a
DEFB 70,85,96   ; b
DEFB 3,68,48    ; c
DEFB 19,85,48   ; d
DEFB 3,86,48    ; e
DEFB 37,70,64   ; f
DEFB 3,83,96    ; g
DEFB 68,117,80  ; h
DEFB 64,68,96   ; i
DEFB 16,17,48   ; j
DEFB 69,102,80  ; k
DEFB 68,68,48   ; l
DEFB 5,117,80   ; m
DEFB 6,85,80    ; n
DEFB 7,85,112   ; o
DEFB 7,87,64    ; p
DEFB 7,87,16    ; q
DEFB 7,68,64    ; r
DEFB 6,66,96    ; s
DEFB 71,68,48   ; t
DEFB 5,85,96    ; u
DEFB 5,85,32    ; v
DEFB 5,87,80    ; w
DEFB 5,34,80    ; x
DEFB 5,113,96   ; y
DEFB 7,36,112   ; z
DEFB 50,66,48   ; {
DEFB 34,34,32   ; |
DEFB 98,18,96   ; }
DEFB 2,80,0     ; ~
DEFB 3,67,0     ; ©
DEFB 0,0,0      ; <space>   <8> (Block Graphics)
DEFB 51,48,0    ; TR    <1> (Block Graphics)
DEFB 204,192,0  ; TL    <2> (Block Graphics)
DEFB 255,240,0  ; Top   <3> (Block Graphics)
DEFB 0,3,51     ; BR    <4> (Block Graphics)
DEFB 51,51,51   ; Right <5> (Block Graphics)
DEFB 204,195,51 ; TL&BR <6> (Block Graphics)
DEFB 255,243,51 ; TL + Right    <7> (Block Graphics)
DEFB 0,12,204   ; BL    <SH 7>  (Block Graphics)
DEFB 51,60,204  ; BL&TR <SH 6>  (Block Graphics)
DEFB 204,204,204; Left  <SH 5>  (Block Graphics)
DEFB 255,252,204; Left + TR <SH 4>  (Block Graphics)
DEFB 0,15,255   ; Bottom    <SH 3>  (Block Graphics)
DEFB 51,63,255  ; BL + Right    <SH 2>  (Block Graphics)
DEFB 204,207,255; Left + BR <SH 1>  (Block Graphics)
DEFB 255,255,255; All 4 <SH 8>  (Block Graphics)
END ASM
Print64x32Udg:
ASM
Print64x32Udg:
DEFB 218,138,175    ; UDG A
DEFB 154,154,159    ; UDG B
DEFB 218,186,223    ; UDG C
DEFB 154,170,159    ; UDG D
DEFB 139,139,143    ; UDG E
DEFB 139,139,191    ; UDG F
DEFB 218,186,143    ; UDG G
DEFB 170,138,175    ; UDG H
DEFB 141,221,143    ; UDG I
DEFB 238,234,223    ; UDG J
DEFB 170,153,175    ; UDG K
DEFB 187,187,143    ; UDG L
DEFB 168,170,175    ; UDG M
```

v: latest

```
DEFB 154,170,175    ; UDG N
DEFB 138,170,143    ; UDG O
DEFB 154,155,191    ; UDG P
DEFB 218,170,207    ; UDG Q
DEFB 154,153,175    ; UDG R
DEFB 203,222,159    ; UDG S
DEFB 141,221,223    ; UDG T
DEFB 170,170,159    ; UDG U
END ASM

END SUB
```

Example

```
CLS
BORDER 2

Print64x32StringAt(5,05,"T")
Print64x32StringAt(6,10,"U")
Print64x32StringAt(7,15,"V")
Print64x32StringAt(8,20,"W")
Print64x32StringAt(9,25,"X")
Print64x32StringAt(10,30,"Y")
Print64x32StringAt(11,35,"Z")

Print64x32StringAt(31,15,"Hello World! :) ")
Print64x32StringAt(15,5,"Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod temp
Print64x32At(0,0)

For n=32 to 164
Print64x32(CHR$(n)+" ")
next n

PAUSE 1
PAUSE 0
```

v: latest