15/9/06

So having the 8 to 16K mod working and able to have upto 16 exta character sets got me thinking, with just 12 extra character sets i'd have enough different characters that would allow you to put a different character at each location on the screen, thus true hi-res using a slightly unusual memory layout, all i would have to do is fill the screem with 12 complete character sets i.e chars 0 to 31 on line 0, chars 32 to 63 on line 1, then chars 0 to 31 again on line 2 and so on. Now the tricky bit. how to hook the video display routine so that the I register is changed evey 2 lines or 16 scanlines. Now NOT been a great understander of the ZX81 video display routines i though i'd take a look how WRX works and have based my routine on Wilf's Thus instead of filling D-file with a load of characters i'm using a dummy sequence of 32 bytes BUT instead of just 1 buffer, i have 2, and instead of just been 32 Nop's they are 00h,01h,02h.....1Fh,RET and 20h,21h,22h.....3Fh,RET each buffer called 8 times, 2 complete character lines, and then the I register is incremented by 2 and then we return to the first buffer for another 2 complete lines and do this 12 times over and **ARX816** is born. The upshot..it works but because my routine has to be sycn'd with the ULA's line counter there's quite a big delay at the begining so this will slow basic down a bit, about 7% of available time for program excution is lost.

```
0001   4082                    .org 16514
0002   4082
0003   4082             ;this is hi res routine based on
                       ;wrx16, except it uses 8-16K above rom
0004   4082             ;however this ram is addressed like rom,
```

```
                        ;during refresh, ie. Bits 0-2 are the line
                        ;ula row counter, bits 3-8 are the char from
                        ;d-file/my dummy d-file. Whats more is this
                        ;Ram can be used as Chr$ generator requiring
                        ;only the I register to change. But can
                        ;equally be used for code execution.
0005   4082
0006   4082
0007   4082
                              lbuf1
0008   4082 0001020304050607   .byte   00,01,02,03,04,05,06,07,
                                       08,09,10,11,12,13,14,15
0008   4088 08090A0B0C0D0E0F
0009   4092 1011121314151617   .byte   16,17,18,19,20,21,22,23,
                                       24,25,26,27,28,29,30,31
0009   4098 18191A1B1C1D1E1F
0010   40A2 C9                 ret
0011   40A3
0012   40A3
                              lbuf2
0013   40A3 2021222324252627   .byte
                                       32,33,34,35,36,37,38,39,
                                       40,41,42,43,44,45,46,47
0013   40A9 28292A2B2C2D2E2F
0014   40B3 3031323334353637   .byte
                                       48,49,50,51,52,53,54,55,
                                       56,57,58,59,60,61,62,63
0014   40B9 38393A3B3C3D3E3F
0015   40C3 C9                 ret

       ;because of the way this ram is addressed during refresh
       ;I need 2 dummy d-file lines, the first to do even character
       ;rows, the second to do odd character rows. So basically the
       ;h-file is laid out like 12 consecutive character maps.

0016   40C4
0017   40C4             ;each buffer is called 8 times as video bytes
0018   40C4             ;both repeated 12 times in total,
                       ;(8+8)*12=192 scanlines.
```

```
0019   40C4
0020   40C4
                           ;this is the main loop when its time to
                           ;display the video because I need to sync
                           ;my lines with the ULA row counter
                           ;the effect is to slow down basic execution
                           ;a bit, not sure how much but its
                           ;effectively 8 scanlline
0021   40C4
                     arx
0022   40C4 06 78          ld b,120              ;between here
0023   40C6 10 FE    delay djnz delay            ;and the first
0024   40C8 03             inc bc                ;execution of
0025   40C9 00             NOP                   ;the lbuf bytes
0026   40CA                                      ;total delay is
0027   40CA 16 08          ld d,8                ;equal to 8 scanlines
0028   40CC 06 0C          ld b,12
0029   40CE 4A             ld c,d                ;set up initial regs
0030   40CF 3E 20          ld a,020h
0031   40D1 ED 47          ld i,a
0032   40D3
0033   40D3
0034   40D3               arx1
0035   40D3 CD 82 C0       call lbuf1+8000h
                                  ;17     17 t states so far
0036   40D6 00             nop
                                  ;4      159 t states so far (138 from ;lbuf1) timing
0037   40D7
0038   40D7 0D             dec c
                                  ;4      163 t states
0039   40D8 CA E2 40       jp z,arx2             ;use conditinal jp uses
                                                  ;10 t states either way
                                  ;10     173 t states
0040   40DB ED 57          ld a,I                ;this way if not 8
                                                  ;scanlines
                                  ;9      182 t states timing
0041   40DD ED 57          ld a,I
                                  ;9      191 t states timing
0042   40DF 00             nop
```

```
                                        ;4      195 t states timing
0043   40E0 18 F1              jr arx1
                                        ;12     207 t states !!!
                                         ;this is the end of the first loop
                                       ;this route is taken if 8 scanlines
                                       ;have been completed.
0044   40E2
0045   40E2             arx2
0046   40E2 ED 57              ld a,I
                                        ;9      182 t states timing
0047   40E4 03                 inc bc
                                        ;6      188 t states timing
                                        ;doesn't matter about c, gets reloaded
0048   40E5 03                 inc bc
                                        ;6      194 t states timing
0049   40E6 4A                 ld c,d
                                        ;4      198 t states
0050   40E7 ED 57              ld a,i
                                        ;9      207 t states !!!
0051   40E9
                                        ;the start of the second inner loop
0052   40E9             arx3
0053   40E9 CD A3 C0           call lbuf2+8000h
                                        ;17     17 t states ; fire the second row
0054   40EC 0D                 dec c
                                        ;4      159 t states (138 from lbuf2)
0055   40ED CA F8 40           jp z,arx4
                                        ;10     169 t states
0056   40F0 ED 57              ld a,I
                                        ;9      178 t states timing
                                          ;this branch if not 8 scan lines
0057   40F2 ED 57              ld a,I
                                        ;9      187 t states timing
0058   40F4 3C                 inc a
                                        ;4      191 t states timing,
                                        ;for the most part
0059   40F5 3C                 inc a
                                        ;4      195 t states timing,
                                        ;but also increments A by 2
```

```
                                        ;ready for the other branch
0060   40F6 18 F1             jr arx3
                                        ;12     207 t states !!!
0061   40F8
0062   40F8             arx4
0063   40F8 4A                 ld c,d
                                        ;4      173 t states reset scan line counter
0064   40F9 ED 47              ld i,a
                                        ;9      182 t states set i to next 512 byte block#
0065   40FB 7E                 ld a,(hl)
                                        ;7      189 t states timing
0066   40FC 00                 nop
                                        ;4      193 t states timing
0067   40FD 05                 dec b
                                        ;4      197 t states
0068   40FE C2 D3 40           jp nz,arx1
                                        ;10     207 t states !!!

;the following section is virtually the same as ;wilfs wrx16

0069   4101 DD 21 07 41        ld ix,arx5
0070   4105 18 07              jr arx6
0071   4107
0072   4107 CD 20 02    arx5   call 0220h
0073   410A DD 21 C4 40        ld ix,arx
0074   410E
0075   410E
0076   410E
0077   410E 3A 28 40    arx6   LD A,(4028h)    ;33 or 19 blank lines in bottom MARGIN

0078   4111 D6 08              SUB 8           ;reduce by 8 scan lines

0079   4113 C3 9E 02           JP 029Eh        ;start NMI, POP registers and RETURN
0080   4116
0081   4116
0082   4116
0083   4116             stop

;STOP hires and return to normal, this is the same as wrx16
```

```
0084   4116 21 81 02          ld hl,0281h    ;pointer to rom video routine
0085   4119 3E 1E             ld a,1Eh       ;rom pattern table base address (1E00)
0086   411B ED 47             ld i,a         ;stick it in the I register
0087   411D 18 03             jr sync
0088   411F
0089   411F             start

;Starts the hires video

0090   411F 21 C4 40          ld hl,arx      ;pointer to the hires video routine
0091   4122
0092   4122             sync                  ;used by START and STOP to smoothly change video mode
0093   4122 E5                 push hl

**************************************            ;this check gets stuck
*** DO NOT START OR STOP IN FAST MODE ***
**************************************            ;if running in fast mode

0094   4123 21 34 40          ld hl,4034h    ;FRAMES counter
0095   4126 7E                ld a,(hl)             ;get old FRAMES
0096   4127             sloop
0097   4127 BE                 cp (hl)        ;compare to new FRAMES
0098   4128 28 FD             jr z,sloop     ;exit after a change is detected
0099   412A DD E1             pop ix
0100   412C C9                ret
0101   412D
0102   412D                    .end
```

the above code is assuming that the H-file is located in the first 6K of the 8-16K region and is mapped thus:- first byte scanline 0 addess 2000h, second byte scanline 0 address 2008h ect.

```
maybe a better way to show it.

scanline 0> 2000h 2008h 2010h 2018h ..... 20F8h
scanline 1> 2001h 2009h 2011h 2019h ..... 20F9h
```

```
.
.
scanline 7> 2007h 200Fh 2017h 201Fh ..... 20FFh
scanline 8> 2100h 2108h 2110h 2118h ..... 21F8h
.
.
scanline 190> 3706h 370Eh 3716h 371Eh ..... 37FEh
scanline 191> 3707h 370Fh 3717h 371Fh ..... 37FFh
```

So with minimal extra hardware and a little programming i get a 3 function hardware upgrade :) UDG upto 16 pages, True hi-res, extra memory for code or storage. and as a bonus as well as working on a real zx81 (issue 1 has not been tested on any other as i haven't got another) it works on Eightyone version 0.42(Test Z) with hi-res disabled character gen set to sinclair, and ram 8-16k checked, it appears that without WRX enabled the ram is addressed with the alternate address lines supplied by the ula, as in my real ZX81 mod. If you read this far you might be wondering why i bothered since as i have static ram in my ZX81 i could have just added the resistor to the /wr line and used WRX, but that would be no fun and i have learnt a heck of alot about the video routines, and timing this way. Oh yeah i get 16K available for my programs too :) will be adding a sample program to my programs section soon. (24/9/06, the program has been added to programs section)