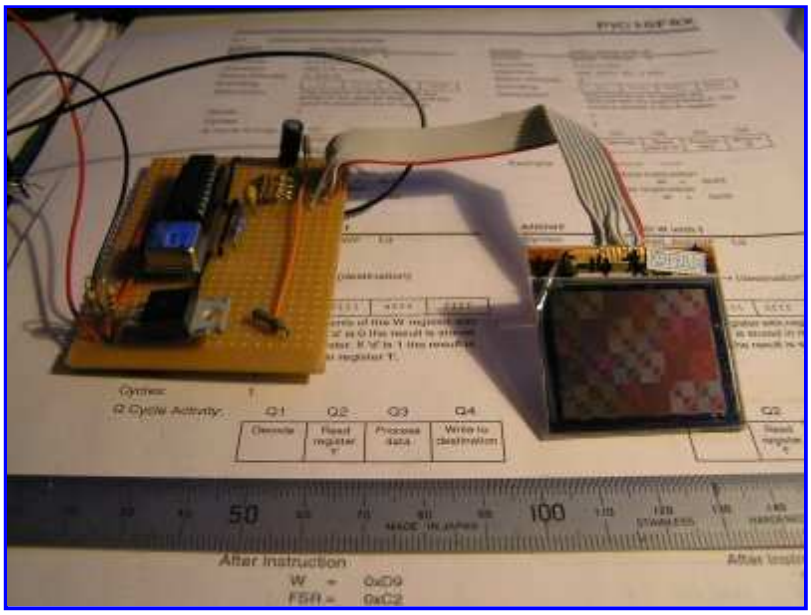
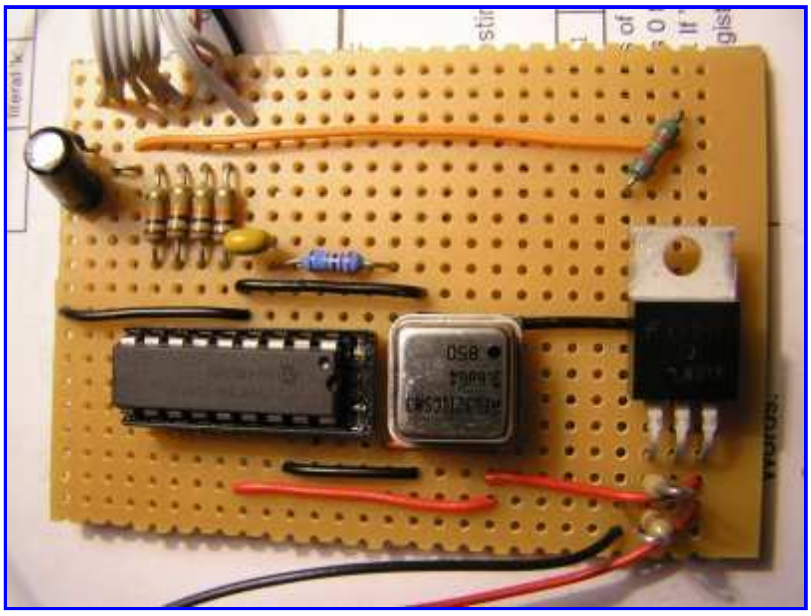
More (up a level) ...

Connecting the Nokia 3510i LCD to a Microchip PIC16F84 microcontroller



As with the FPGA board previously, the connections are made by soldering standard IDC ribbon cable directly to the glass substrate.
As the PIC will not work from 3.3V, I had to make a power supply for the LCD. For this I used an LM317, is complete overkill but easy and quick. Both LCD supply pins are tied to the output, which is around 3.0V. All four signal lines have 10K resistors in series, to protect the LCD inputs from the 5V signalling of the PIC.
This time the LCD decoupling is taken care of by a 1nF ceramic cap in parallel with a 4.7uF electrolytic, positive side to pad 8 of the LCD, negative to ground. This results in a rock steady image.



My lash-up for testing the PIC. Note the resistors in the signal paths. Crystal frequency is 3.6864MHz, well within the 16F84's maximum of 4MHz.

PIC is continuously writing this rolling, XOR pattern to the display. (Pixel brightness

= X co-ord XOR Y co-ord). Refresh rate is a little disappointing at only 2Hz! Next version will be with either a faster PIC, or a PIC with a hardware serial port, or both.

Here's the assembly code for PIC16F84 and MPASM. You can save this as a separate file from your own program, just make sure you put both this file and your own in the project. You will need to delcare any of the routines defined here you want to call at the top of your code as "extern". See the MPASM documentation on how to do this. It's pretty simple! All the routines I've exported are declared "global" (see declarations near the top of the listing).
The LCD starts up in 12 bit color mode (12BPP), and I've only made it work successfully in this mode. If you're only going to display text, then 8BPP mode will be adequate and probably much faster.
When using 12BPP mode, call "LCD3510_WritePixelPair" to write two pixels at a time. You pass a pointer to your 6-byte RAM array in the W-register. This procedure uses only the most significant four bits of each byte.

```
;
; Nokia 3510i colour LCD interface routines
;
; MPASM assembler, PIC16F84
;
; neil_manc@yahoo.com    11/10/2005
;
; 11/10 NS Created
;
;

        list    p = 16f84
        #include p16f84.inc

; ports and pins for display interface
#define NRESET  PORTB, 0
#define NCS             PORTB, 1
#define SCK             PORTB, 3
#define SDA             PORTB, 2


; declarations for exported functions
        global LCD3510_Initialise
        global LCD3510_BeginWritePixel
        global LCD3510_WritePixel
        global LCD3510_WritePixelPair
        global LCD3510_CASET
        global LCD3510_PASET
        global LCD3510_SET_12BPP
        global LCD3510_SET_8BPP
        global LCD3510_Command
        global LCD3510_Parameter


; local storage
lcd_if_data             udata
_ctr_i  res     1
;_ctr_j res 1
_ctr_n  res 1
_ctr_w1 res 1
_ctr_w2 res 1
_s_byte res 1
_c_byte res 1

; relocatable code starts here
lcd_interface   code


; look-up table for display initialisation sequence. call with index
```

```
; in W reg. each element consists of
; <parameter count> <command code>
;     [<parameter> [<parameter [<parameter> ... ]]]
_init_seq
        addwf   PCL, f

        retlw   .0
        retlw   0xC6    ; initial escape

        retlw   .1
        retlw   0xB9    ; refresh set
        retlw   .0

        retlw   .7
        retlw   0xB6    ; display control
        retlw   .128
        retlw   .128
        retlw   .129
        retlw   .84
        retlw   .69
        retlw   .82
        retlw   .67

        retlw   .15
        retlw   0xB3    ; setup greyscale map 0
        retlw   0x11
        retlw   0x22
        retlw   0x33
        retlw   0x44
        retlw   0x55
        retlw   0x66
        retlw   0x77
        retlw   0x88
        retlw   0x99
        retlw   0xAA
        retlw   0xBB
        retlw   0xCC
        retlw   0xDD
        retlw   0xEE
        retlw   0xFF

        retlw   .1
        retlw   0xB5    ; gamma curve set
        retlw   .1

        retlw   .1
        retlw   0xBD    ; common driver output
        retlw   .0

        retlw   .1
        retlw   0xBE    ; power supply setup
        retlw   .4

        retlw   .0
        retlw   0x11    ; sleep out

        retlw   .2
        retlw   0xBA    ; voltage control
        retlw   .127
        retlw   .3

        retlw   .1
        retlw   0x25    ; contrast
        retlw   .70

        retlw   .20
        retlw   0x2D    ; colour look up table for 8BPP mode
        retlw   0x00    ; red (8 levels)
        retlw   0x02
```

```
        retlw   0x04
        retlw   0x06
        retlw   0x09
        retlw   0x0B
        retlw   0x0D
        retlw   0x0F
        retlw   0x00    ; green (8 levels)
        retlw   0x02
        retlw   0x04
        retlw   0x06
        retlw   0x09
        retlw   0x0B
        retlw   0x0D
        retlw   0x0F
        retlw   0x00    ; blue (4 levels)
        retlw   0x05
        retlw   0x0A
        retlw   0x0F

        retlw   .13
        retlw   0xB7    ; setup temp gradient
        retlw   .0              ; -.05(n+1)%/K, 0 <= (n) < 4
        retlw   .0              ; all zeros(?) from here on:
        retlw   .0              ; used during factory testing
        retlw   .0
        retlw   .0
        retlw   .0
        retlw   .0
        retlw   .0
        retlw   .0
        retlw   .0
        retlw   .0
        retlw   .0
        retlw   .0

        retlw   .0
        retlw   0x03    ; booster voltage on

        retlw   0xED    ; end of sequence!

        retlw   .0
        retlw   0x21    ; inversion on

        retlw   .0
        retlw   0x29    ; display on



; short wait (for debugging)
_short_wait
        clrf    _ctr_w2
        incf    _ctr_w2
        clrf    _ctr_w1
        goto    __wait_loop

; waits 50ms (at 4MHz) TODO: find a way of automating this
_wait_5ms
        movlw   70
        movwf   _ctr_w2
        clrf    _ctr_w1
__wait_loop
        decfsz  _ctr_w1, F
        goto    __wait_loop
        decfsz  _ctr_w2, F
        goto    __wait_loop
        return

; exported function: LC3510_Command
; used for sending misc command bytes to the display
```

```
LCD3510_Command

; sends a command byte to the display. call with command stored in
; W register
_command
        movwf   _s_byte ; save W reg

                                        ; terminate any prior sequence of parameters
        bsf             NCS

        bcf             SCK
;       call    __per_cycle_wait
        bcf             NCS

                                        ; first bit is zero for command
        bcf             SDA
;       call    __per_cycle_wait
        bsf             SCK

        bsf             SDA             ; data pin high, ready for byte sending
        goto    __send_byte



; exported function: LC3510_WritePixel
; pixel value stored in W register
; writes one 8BPP pixel to display
LCD3510_WritePixel

; exported function: LC3510_Parameter
; used for sending misc parameter bytes to the display
LCD3510_Parameter

; sends a parameter byte to the display. call with parameter in W
; register
_parameter
        movwf   _s_byte ; save W reg

        bcf             SCK             ; clock in a one for parameter
        bsf             SDA
        bsf             SCK

__send_byte                     ; unrolled byte serialisation loop
        bcf             SCK
        btfss   _s_byte, 7
        bcf             SDA
        bsf             SCK

        bcf             SCK
        btfsc   _s_byte, 6
        bsf             SDA
        btfss   _s_byte, 6
        bcf             SDA
        bsf             SCK

        bcf             SCK
        btfsc   _s_byte, 5
        bsf             SDA
        btfss   _s_byte, 5
        bcf             SDA
        bsf             SCK

        bcf             SCK
        btfsc   _s_byte, 4
        bsf             SDA
        btfss   _s_byte, 4
        bcf             SDA
        bsf             SCK
```

```
        bcf             SCK
        btfsc   _s_byte, 3
        bsf             SDA
        btfss   _s_byte, 3
        bcf             SDA
        bsf             SCK

        bcf             SCK
        btfsc   _s_byte, 2
        bsf             SDA
        btfss   _s_byte, 2
        bcf             SDA
        bsf             SCK

        bcf             SCK
        btfsc   _s_byte, 1
        bsf             SDA
        btfss   _s_byte, 1
        bcf             SDA
        bsf             SCK

        bcf             SCK
        btfsc   _s_byte, 0
        bsf             SDA
        btfss   _s_byte, 0
        bcf             SDA
        bsf             SCK

        return


; exported function: LCD3510_initialise
; no paramters
LCD3510_Initialise
        bcf             STATUS, RP1
        bsf             STATUS, RP0

        bcf             NRESET  ; set ports to be outputs
        bcf             NCS
        bcf             SDA
        bcf             SCK

        bcf             STATUS, RP0
        bcf             NRESET
        bsf             NCS
        bsf             SDA
        bsf             SCK

        call    _wait_5ms

        bsf             NRESET
        movlw   0x01    ; send soft reset
        call    _command
        bsf             NCS
        call    _wait_5ms

        clrf    _ctr_i  ; begin display initialisation loop
__disp_init_loop
        movf    _ctr_i, W       ; get number of parameters
        call    _init_seq
        movwf   _ctr_n
        incf    _ctr_i, F

        xorlw   0xED            ; test for end of sequence
        btfsc   STATUS, Z
        goto    __init_end

        movf    _ctr_i, W       ; send command byte
```

```
        incf    _ctr_i, F
        call    _init_seq
        call    _command

        movf    _ctr_n, F       ; test if n = 0 (no parameters)
        btfsc   STATUS, Z
        goto    __disp_init_loop

__init_param_loop               ; get each parameter and send to display
        movf    _ctr_i, W
        incf    _ctr_i, F
        call    _init_seq
        call    _parameter
        decfsz  _ctr_n, F
        goto    __init_param_loop

        goto    __disp_init_loop

__init_end
        call    _wait_5ms       ; time for booster voltage to settle

        movlw   0x21    ; inversion on
        call    _command

        movlw   0x29    ; display on
        call    _command

        bsf             NCS             ; terminate final parameter sequence
        return


; exported function: LCD3510_BeginWritePixel
; no parameters
LCD3510_BeginWritePixel
        movlw   0x2C            ; write memory command
        call    _command
        return


; exported function: LC3510_WritePixelPair
; pointer to six-byte array of RGB values stored in W register
; writes two adjacent 12BPP pixels to display
LCD3510_WritePixelPair
        movwf   FSR
        movf    INDF, W
        andlw   0xF0
        movwf   _c_byte

        incf    FSR, F
        swapf   INDF, W
        andlw   0x0F
        iorwf   _c_byte, W
        call    _parameter

        incf    FSR, F
        movf    INDF, W
        andlw   0xF0
        movwf   _c_byte

        incf    FSR, F
        swapf   INDF, W
        andlw   0x0F
        iorwf   _c_byte, W
        call    _parameter

        incf    FSR, F
        movf    INDF, W
        andlw   0xF0
        movwf   _c_byte
```

```
        incf    FSR, F
        swapf   INDF, W
        andlw   0x0F
        iorwf   _c_byte, W
        call    _parameter
        return


; exported function: LC3510_CASET
; pointer to two-byte array stored in W register: start column no.
; and end column respectively.
LCD3510_CASET
        movwf   FSR
        movlw   0x2A
        call    _command

        movf    INDF, W
        call    _parameter

        incf    FSR, F
        movf    INDF, W
        call    _parameter
        return

; exported function: LC3510_PASET
; pointer to two-byte array stored in W register: start row no.
; and end row respectively.
LCD3510_PASET
        movwf   FSR
        movlw   0x2B
        call    _command

        movf    INDF, W
        call    _parameter

        incf    FSR, F
        movf    INDF, W
        call    _parameter
        return


; exported function: LC3510_SET_12BPP: pixels are packed RRRRGGGGBBBB
; no parameters
LCD3510_SET_12BPP
        movlw   0x3A
        call    _command
        movlw   0x03
        call    _parameter
        return


; exported function: LC3510_SET_8BPP. pixels are packed RRRGGGBB
; no parameters
LCD3510_SET_8BPP
        movlw   0x3A
        call    _command
        movlw   0x02
        call    _parameter
        return

        end
```


up one level