

Ensc 215 Lab 7 – Push-button interrupts.

Version 3.13, Copyright © W. Craig Scratchley 2008-2009

Updated by Liila Torabi

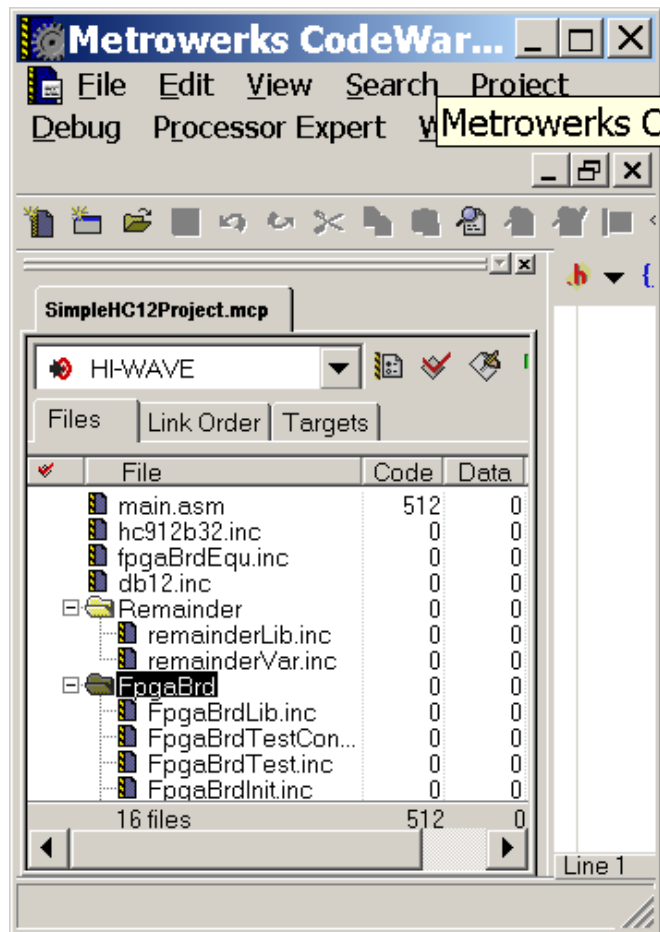
November 1st, 2010

Preparing for switch interrupts

Create a folder named “Lab7”, make there a copy of your project from Lab 5. Now we will set things up to get interrupts for switches.

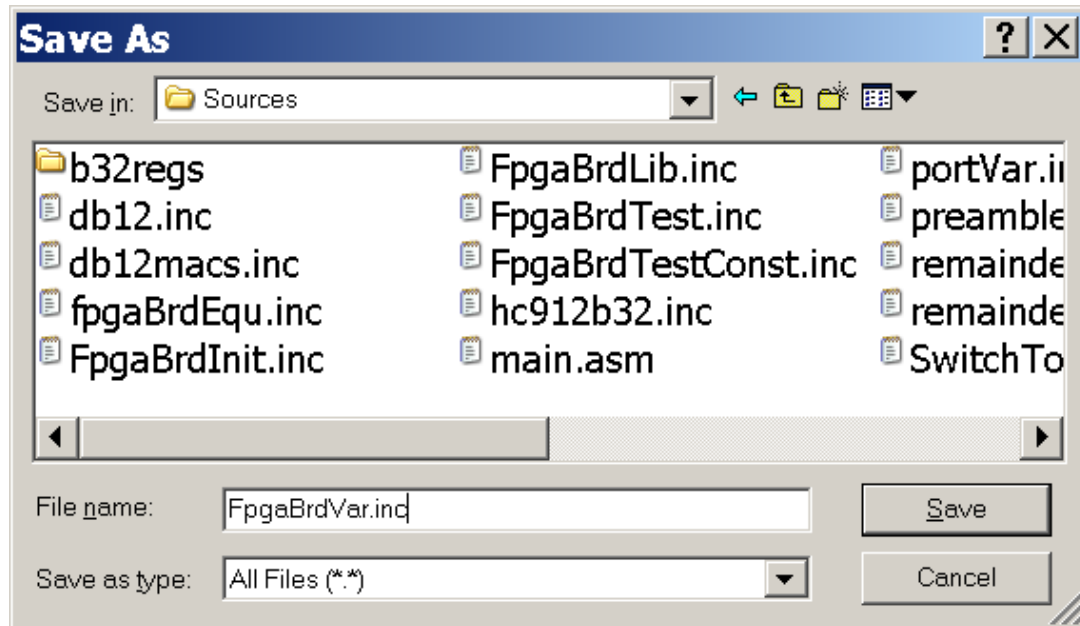
First, add a new file called FpgaBrdVar.inc to the FpgaBrd group:

1. Click on the FpgaBrd group

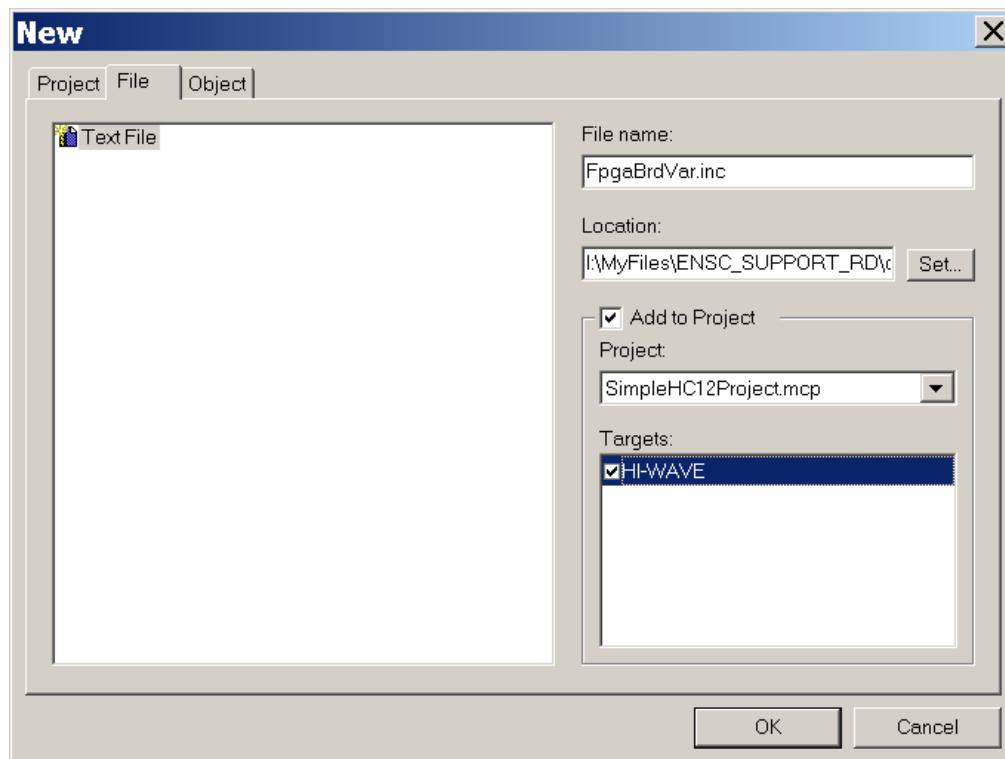


2. Click on the “New...” item from the File menu.
3. Choose the “File” tab.
4. Click on the “Set...” button. You may be taken to your project directory (215Lab.cw). If not, navigate there (make sure you choose the directory for the correct copy of the project).

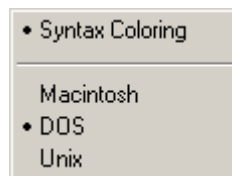
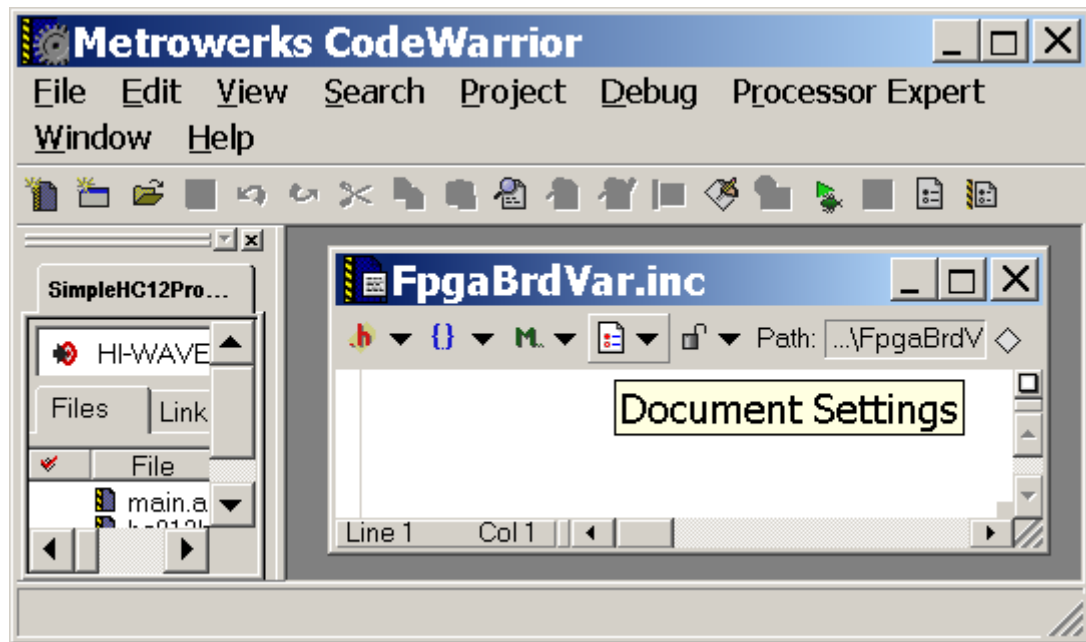
5. Double-click on the “Sources” folder to enter that folder.
6. Enter FpgaBrdVar.inc in the “File name:” field. Click on the Save button.



7. Check the “Add to Project” check box and then the HI-WAVE target check box and then click OK.



8. An editor for the new file will come up. Since this file will be for assembly programming, turn “Syntax Coloring” on after clicking on the “Document Settings” icon.



9. The file should now appear in your project window.

Then, do the following steps.

- Add the following lines to FpgaBrdVar.inc:

swIntCnt:	ds.b 1	; Switch interrupt counter.
swStat_old:	ds.b 1	; old Switch status

- Modify the file main.asm to include the FpgaBrdVar.inc file on the line before label Entry. “Make” (F7) your project so that the IDE can learn the variable names.

- Add the following lines to FpgaBrdLib.inc above the LCD-related subroutines. Since the IDE now knows about the variable names, the variables should turn light blue when you have typed them in properly.

```
;Interrupt Service routine For the External IRQ

ExtIsr:
    ;send reset pulse to push button(/LCD) flip flop
    bclr PORTE, nRST1
    bset PORTE, nRST1

    ;Check if the interrupt is from Push Buttons
    ldaa PORTB
    ; only the four button bits are important
    anda #(nSw4B | nSw3WG | nSw2RO | nSw1Y)
    cmpa swStat_old
    beq daughterChk

    ;deal with push buttons
    inc swIntCnt
    staa swStat_old

daughterChk:
    brclr PORTE, Status2, endExtIsr

    ;if we get here there is an unexpected interrupt
    bclr PORTE, nRST2
    bset PORTE, nRST2

endExtIsr:
    rti
```

“Make” (F7) your project so that the IDE can learn about the ExtIsr label.

In the initialization part of your main code , add the following lines before LCD initialization:

```

        bset  DDRE, nRST1 | nRST2
        ;; bclr  DDRE, Status2      ; default setting

        movb  #0, swIntCnt          ; initialize the interrupt counter
; initialize switch status variable (swStat_old)
        ldaa  PORTB
        ; only the four button bits are important
        anda  #(nSw4B | nSw3WG | nSw2RO | nSw1Y)
        staa  swStat_old

;Initialize the external interrupt
        movw  #ExtIsr, IRQIsrP

        bclr  PORTE, nRST1 | nRST2      ; reset flipflops
        ;enable external interrupts in default level-sensitive mode
        ; (not edge sensitive)
        bset  INTCR, IRQEN | !IRQE
        ;enable push button interrupt but not daughter brd interrupt
        bset  PORTE, nRST1 | !nRST2

```

- Now in your main file right after you have send a letter or a message to the LCD add the following part:

```

; LED1 should now be OFF. The LCD should now display the strings
; "The M68HC12 is" and "AWESOME".

LoopWaiting:
        wai

        ldaa  #255
        jsr   uS100Delay

; One or more buttons have been pressed or released.
; Check swIntCnt variable. Click "continue" and press or
; release a button.
        bra   LoopWaiting      ; wait for another button press

```

Run the program, follow directions in relevant comments when the program halts itself, and see what happens. Can you explain what you see? If not, don't worry.