AN811

MicroBolt

Level sensitive interrupts on the MicroBolt

10/7/2005

Introduction:

This application notes demonstrates how to use an external interrupt pin on the MicroBolt as a level sensitive interrupt.

Background:

The MicroBolt allows for external low level sensitive interrupts.

How it works:

This ImageCraft ICCARM demo project turns on the onboard MicroBolt LED, in the EINT1 interrupt handler, when the EINT1 push button switch is pressed down to create an interrupt. The onboard MicroBolt LED is then turned off in main. This demonstrates the external low level interrupt detection of the MicroBolt.

Note:

The EINT1 push button switch is labeled "ISP*" on the MicroBolt development board.

Program Listing:

```
/*
|-----------------------------------------------------------------------------------------
|-----------------------------------------------------------------------------------------
| File Name                      : MicroBoltExtIntLevel.c
| Author                         : Micromint, Inc.
| Copyright                      : Copyright © 2005, Micromint, Inc.
| Creation Date                  : 4/2/05
| Version                        : 1.00
| Spaces per tab                 : 2
| Description                    : Main C file
| Revision                       : Initial
|-----------------------------------------------------------------------------------------
|-----------------------------------------------------------------------------------------
*/

/*
|-----------------------------------------------------
|     Includes
|-----------------------------------------------------
*/

#include <ARM/philips/lpc210x.h>
#include <arm_macros.h>

#include "MicroBoltExtIntLevel.h"

/*
|-----------------------------------------------------------------------------------------
|     Function         :   main
|     Inputs           :   None
|     Outputs          :   None
|     Purpose          :   Main function for system
|     Author           :   Micromint, Inc.
|-----------------------------------------------------------------------------------------
*/
```

```
void main(void)
  {

/*
|----------------------------------------------------
|     MicroBolt hardware setup
|----------------------------------------------------
*/

  __DISABLE_INTERRUPT();                        // Disable all interrupts

  SCB_PLLCFG |= 0x23;                            // Turn on PLL, set to 59 MHz (0x03 is multiply value of 4)
  SCB_PLLCON |= 0x03;
  SCB_PLLFEED = 0xAA;                           // Shadow register copy for PLL
  SCB_PLLFEED = 0x55;

  PCB_PINSEL0=0x00000000;                       // JTAG is via secondary port
  PCB_PINSEL1=0x55400000;
  GPIO_IODIR=(0x00000000<<16)|
             0x00000000;

  GPIO_IOCLR=0xffffffff;
  GPIO_IOSET=(0x00000000<<16)|
            0x00000000;

  GPIO_IODIR |= MICROBOLT_LED;                  // Setup MicroBolt LED as output

  PCB_PINSEL0 |= P0_14_EXTERNAL_INTERRUPT_1;    // Setup P0.14 to alternate function EINT1

  VICVectAddr0 = (unsigned)ExtInt1_ISR;         // Assign the EINT1 ISR function to VIC priority 0
  VICVectCntl0 = INTERRUPT_CHANNEL_FOR_EINT1;   // Assign the VIC channel EINT1 to interrupt priority 0

  VICIntEnable |= INTERRUPT_ENABLE_FOR_EINT1;   // Enable the interrupt

  __ENABLE_INTERRUPT();                         // Enable all interrupts

/*
|----------------------------------------------------
|     Start of application
|----------------------------------------------------
*/

  while(1)                                      // Do this forever
    {
    GPIO_IOCLR = MICROBOLT_LED;                 // MicroBolt LED Off
    }
  }

/*
|----------------------------------------------------
|     External Interrupt-1 Interrupt Service Routine
|----------------------------------------------------
*/
#pragma interrupt_handler ExtInt1_ISR

void ExtInt1_ISR(void)                          // Come here whenever there is a low level on EINT1
  {
  GPIO_IOSET = MICROBOLT_LED;                   // MicroBolt LED On
  SCB_EXTINT |= EXTINT_CLR;                     // Clear interrupt

  VICVectAddr = VIC_ACK;                        // Acknowledge Interrupt
  }
```