Micromint, Inc	**AN809**
	MicroBolt
Timer Interrupts on the MicroBolt	10/7/2005

Introduction:
This application notes demonstrates how to use Timer-0 with an interrupt on the MicroBolt.

Background:
The MicroBolt has 2 32-bit timers that can be used with interrupts.

How it works:
This ImageCraft ICCARM demo project sets up Timer-0 and then toggles the onboard MicroBolt LED every Timer-0 interrupt. This demonstrates the timer interrupt via match capability of the MicroBolt.

Program Listing:

```
/*
|-----------------------------------------------------------------------------------------
|-----------------------------------------------------------------------------------------
| File Name                     : MicroBoltTimerInterrupt.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 "MicroBoltTimerInterrupt.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;                             // Set to 59 MHz (0x03 is multiply value of 4)
  SCB_PLLCON |= 0x03;                             // Enable and Connect the PLL
  SCB_PLLFEED = 0xAA;                             // Shadow register copy to enable changes
  SCB_PLLFEED = 0x55;                             // in PLLCON and PLLCFG

  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

  T0_MR0 = 0x00400000;                            // Match register 0 value for timer rate
  T0_MCR = 0x00000003;                            // Interrupt and Reset on MR0
  T0_TCR = 0000000001;                            // Timer0 Enable

  VICVectAddr0 = (unsigned)Timer0_ISR;            // Assign the Timer-0 ISR function to VIC priority 0
  VICVectCntl0 = INTERRUPT_CHANNEL_FOR_TIMER0;    // Assign the VIC channel Timer-0 to interrupt priority 0

  VICIntEnable |= INTERRUPT_ENABLE_FOR_TIMER0;    // Enable the Timer-0 interrupt

  __ENABLE_INTERRUPT();                           // Enable all interrupts
/*
|-----------------------------------------------------
|     Start of application
|-----------------------------------------------------
*/

  while(1)                                        // Do this forever
    {
    }
  }

/*
|-------------------------------------------------------------------------------------------
|     Function        :   Timer0_ISR
|     Inputs          :   None
|     Outputs         :   None
|     Purpose         :   Interrupt service routine for Timer-0
|     Author          :   Micromint, Inc.
|-------------------------------------------------------------------------------------------
*/

#pragma interrupt_handler Timer0_ISR

void Timer0_ISR (void)
  {
       static char LedFlip = 0;

  if (LedFlip == 0)                               // Toggle LED every interrupt
      {
      GPIO_IOSET = MICROBOLT_LED;                 // Turn on MicroBolt LED
      LedFlip = 1;
                      }
  else
            {
      GPIO_IOCLR = MICROBOLT_LED;                 // Turn off MicroBolt LED
      LedFlip = 0;
      }

  T0_IR = TIMER_CLR;                              // Clear Timer-0 interrupt flag
  VICVectAddr = VIC_ACK;                          // Acknowledge Interrupt
}
```