Micromint, Inc	**AN804**
	MicroBolt
Math examples on the MicroBolt	10/7/2005

Introduction:
This application notes demonstrates the math capabilities of the MicroBolt and the ImageCraft ICCARM compiler.

Background:
The ImageCraft ICCARM compiler supports various types of math operations that run on the MicroBolt (e.g., Multiply, Divide, and Floating point operations)

How it works:
This demo project performs various math operations and then checks the answers of those operations. If the answers turn out correct, the onboard MicroBolt LED is toggled.

Program Listing:

```
/*
|-------------------------------------------------------------------------------------------
|-------------------------------------------------------------------------------------------
| File Name                     : MicroBoltMath.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 "MicroBoltMath.h"

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


void main(void)
```

```
  {
       unsigned int Delay;
  float FloatNum;

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

  __DISABLE_INTERRUPT();                        // Disable all interrupts

  MAM_CR = 0x00;                                // Turn MAM off (default)
  MAM_TIM = 0x04;                               // Set flash timing to 4 clock cycles

  MAM_CR = 0x02;                                // Fully enable the Memory Accleration Module

  VICVectAddr8 = (unsigned)pll_isr;             // Assign the PLL lock ISR vector address
  VICVectCntl8 = INTERRUPT_CHANNEL_FOR_PLL;     // Assign the VIC address to the actual interrupt
  VICIntEnable = INTERRUPT_ENABLE_FOR_PLL;      // Enable the interrupt

  SCB_PLLCFG |= 0x23;                           // Set to 59 MHz (0x03 is multiply value of 4)
  SCB_PLLCON |= 0x01;                           // Enable 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)|                  // Make all inputs to start with
             0x00000000;

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

  __ENABLE_INTERRUPT();                         // Enable all interrupts

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

  while(1)                                          // Do this forever
    {

//-----------------------------------------------------
    FloatNum = 1 - 0.45;

    if (FloatNum == 0.55)
      {
      GPIO_IOSET = MICROBOLT_LED;                   // MicroBolt LED On
      }
//-----------------------------------------------------

    for (Delay = 0; Delay < 3700000; Delay++);      // Delay for 1 Second

//-----------------------------------------------------
    FloatNum = 1 - 0.3355;

    if (FloatNum == 0.6645)
      {
      GPIO_IOCLR = MICROBOLT_LED;                   // MicroBolt LED Off
      }
//-----------------------------------------------------

    for (Delay = 0; Delay < 3700000; Delay++);      // Delay for 1 Second

//-----------------------------------------------------
    FloatNum = 1.55 * 8.456789;

    if (FloatNum == 13.1080229)
      {
      GPIO_IOSET = MICROBOLT_LED;                   // MicroBolt LED On
      }
```

```
//-----------------------------------------------------

    for (Delay = 0; Delay < 3700000; Delay++);      // Delay for 1 Second

//-----------------------------------------------------
    FloatNum = 2.5 / 0.425;

    if (FloatNum == 5.8823529)
      {
      GPIO_IOCLR = MICROBOLT_LED;                   // MicroBolt LED Off
      }
//-----------------------------------------------------

    for (Delay = 0; Delay < 3700000; Delay++);      // Delay for 1 Second

    }
  }

/*
|--------------------------------------------------------------------------------------------
|     Function        :   pll_isr
|     Inputs          :   None
|     Outputs         :   None
|     Purpose         :   Once PLL has locked, connect it and use for system clock
|     Author          :   Micromint, Inc.
|--------------------------------------------------------------------------------------------
*/

#pragma interrupt_handler pll_isr
void pll_isr(void)
  {
  SCB_PLLCON |= 0x02;                           // Connect the PLL
  SCB_PLLFEED = 0xAA;                           // Shadow register copy to enable changes
  SCB_PLLFEED = 0x55;                           // in PLLCON and PLLCFG
  VICIntEnClear = PLL_CLR;                      // Disable the PLL interrupt, not needed anymore
  VICVectAddr = VIC_ACK;                        // Acknowledge Interrupt
  }
```