Micromint, Inc	**AN801**
	MicroBolt
Hello World using Printf on the MicroBolt	10/7/2005

Introduction:
This application notes demonstrates how to send ASCII text with Printf out from UART-0 on the MicroBolt.

Background:
The MicroBolt contains two UARTs for asynchronous serial communications.

How it works:
This ImageCraft ICCARM demo project outputs "Hello World..."X"...from MicroBolt!", via printf, every second out the UART-0 serial port at 115200 baud. The "X" is an incrementing counter value showing off the printf functionality. This demonstrates the UART serial capability of the MicroBolt when using printf functionality.

To add printf functionality to your program, copy the UART setup code and putchar function and paste it into your project.

To Test:
On the MicroBolt development board, setup the UART-0 configuration jumpers accordingly (see MicroBolt datasheet) and connect a serial cable from J1 to the PC.

Setup the terminal emulator in ICCARM via "Tools", "Environment Options", "Terminal". Select 115200 as the baud rate and select the correct PC COM port connected to the MicroBolt board. Now go to "Terminal" and select "Show Terminal Window". The terminal emulator window opens. Now hit "Open COM Port" and watch the serial data update on the screen from the MicroBolt. The data that will appear is "Hello World..."X"...from MicroBolt!".

Program Listing:

```
/*
|-------------------------------------------------------------------------------------------
|-------------------------------------------------------------------------------------------
| File Name                     : MicroBoltHelloWorld.c
| Author                        : Micromint, Inc.
| Copyright                     : Copyright © 2005, Micromint, Inc.
| Creation Date                 : 4/2/05
| Version                       : 2.00
| Spaces per tab                : 2
| Description                   : Main C file
| Revision                      : 1 - Initial
|                               : 2 - Add printf functionality
|-------------------------------------------------------------------------------------------
|-------------------------------------------------------------------------------------------
*/

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

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

```
#include "MicroBoltHelloWorld.h"

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

extern int _textmode;                                  // This is defined in the ICCARM library

void main(void)
  {

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

  unsigned int Delay;
  unsigned int Count = 0;

  __DISABLE_INTERRUPT();                          // Disable all interrupts

/*
|-------------------------
| Config MAM
|-------------------------
*/
  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

/*
|-------------------------
| Config PLL
|-------------------------
*/
  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

/*
|-------------------------
| Config GPIO
|-------------------------
*/
  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

/*
|-------------------------
| Config UART-O
|-------------------------
*/
  PCB_PINSEL0 |= P0_0_UART_0_TX;                  // Setup P0.0 to alternate function UART0-TX

  UART0_LCR = 0x00000083;                        // Enable the divisor
```

```
  UART0_DLM = 0;                                  // Divisor latch MSB (for baud rates < 4800)
  UART0_DLL = BAUD_RATE_115200;                    // Divisor latch LSB
  UART0_LCR = 0x00000003;                          // Close LCR, now UART works with divisor

  __ENABLE_INTERRUPT();                            // Enable all interrupts

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

  while(1)                                                // Do this forever
    {
    Count++;                                              // Increment test counter

    printf("Hello World...%d", Count);                    // Output string via UART-0
    GPIO_IOSET = MICROBOLT_LED;                           // MicroBolt LED On
    for (Delay = 0; Delay < 3700000; Delay++);            // Delay for 1/2 Second

    printf("...from MicroBolt!\n", Count);                // Output string via UART-0
    GPIO_IOCLR = MICROBOLT_LED;                           // MicroBolt LED Off
    for (Delay = 0; Delay < 3700000; Delay++);            // Delay for 1/2 Second
    }
  }

/*
|-------------------------------------------------------------------------------------------
|     Function        :   putchar
|     Inputs          :   Byte for transmission
|     Outputs         :   None
|     Purpose         :   Transmit a byte via UART0 to enable printf
|     Author          :   Micromint, Inc.
|-------------------------------------------------------------------------------------------
*/

int putchar(char SerialByte)
  {
  _textmode = 1;                                          // Turn carriage return line feed on for printf
  if (_textmode == 1 && SerialByte == '\n')
      {
      putchar('\r');
      }
  while (!(UART0_LSR & 0x20));
  UART0_THR = SerialByte;
  return SerialByte;
  }


/*
|-------------------------------------------------------------------------------------------
|     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
  }
```