

	AN802
	MicroBolt
Hello World on the MicroBolt	10/7/2005

**Introduction:**

This application notes demonstrates how to send ASCII text 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...from MicroBolt!" every second out the UART-0 serial port at 115200 baud. This demonstrates the UART serial capability of the MicroBolt.

**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...from MicroBolt!".

**Program Listing:**

```

/*
-----
File Name           : MicroBoltHelloWorld.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 "MicroBoltHelloWorld.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_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

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

while(1) // Do this forever
{
    unsigned int Delay;

    GPIO_IOSET = MICROBOLT_LED; // MicroBolt LED On
    SendSerialStringUart0("Hello World...", FALSE); // Output string via UART-0, No carriage return or
line feed
    for (Delay = 0; Delay < 780000; Delay++); // Delay for 1 Second

    GPIO_IOCLR = MICROBOLT_LED; // MicroBolt LED Off
    SendSerialStringUart0("from MicroBolt!", TRUE); // Output string via UART-0 with carriage return and
line feed
    for (Delay = 0; Delay < 780000; Delay++); // Delay for 1 Second
}
}

/*
-----
Function      : Uart0TxByte
Inputs       : Byte for transmission
Outputs      : None
Purpose      : Transmit a byte via UART0
Author       : Micromint, Inc.
-----
*/

void Uart0TxByte(unsigned char SerialByte)

```

```

{
  while (!(UART0_LSR & 0x20));
  UART0_THR = SerialByte;
}

/*
-----
Function      :  SendSerialStringUart0
Inputs       :  Serial string and option for CR and LF
Outputs      :  None
Purpose      :  Transmit a serial string via UART0
Author       :  Micromint, Inc.
-----
*/

void SendSerialStringUart0(char SerialString[], char SendCarriageReturnLineFeed)
{
  unsigned char i = 0;

  while (SerialString[i] != 0x00)
  {
    Uart0TxByte(SerialString[i]);
    i++;
  }

  if (SendCarriageReturnLineFeed == TRUE)
  {
    Uart0TxByte(13);
    Uart0TxByte(10);
  }
}

```