Micromint, Inc	**AN817**
	MicroBolt
MicroBolt Accurate Delay Routines	12/30/2005

Introduction:
This application notes demonstrates accurate delay routines for the MicroBolt.

Background:
Simple software delay routines are useful in many MicroBolt applications and do not require the use of onboard timers.

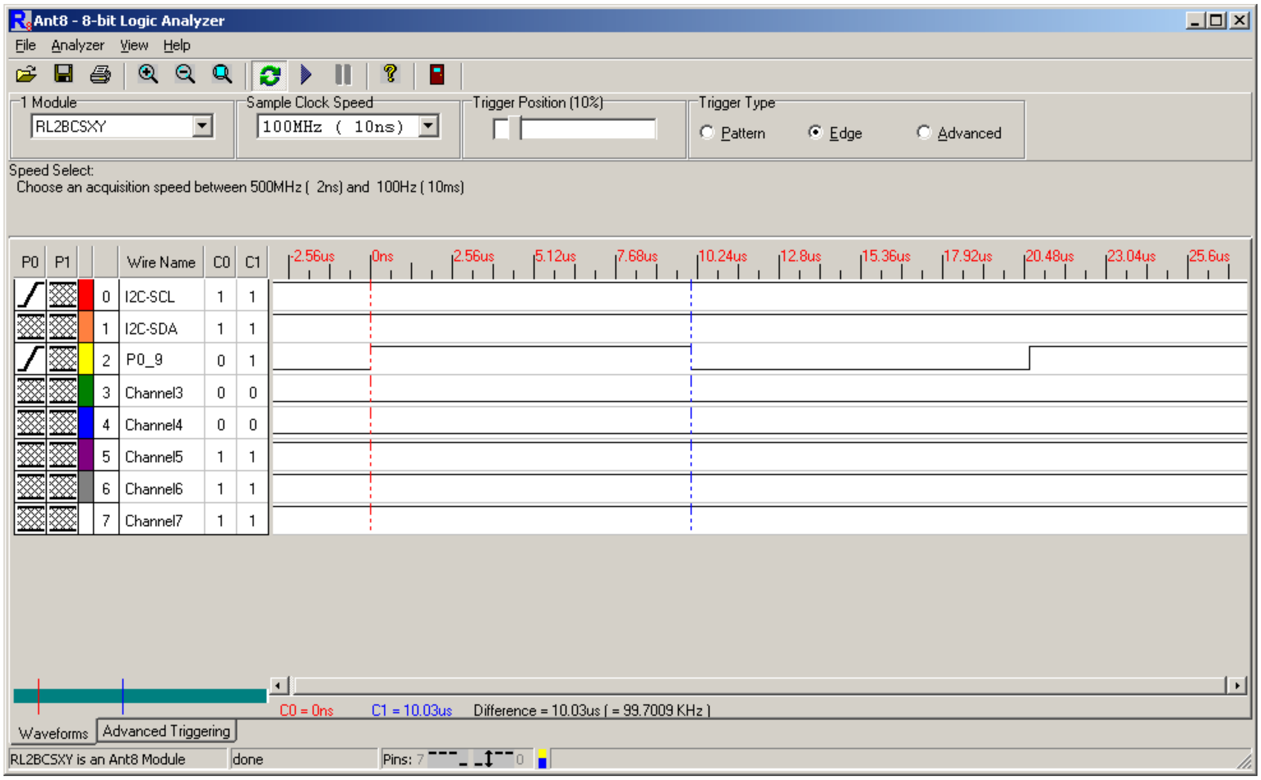
How it works:
This ImageCraft ICCARM demo project provides convenient delay functions for the MicroBolt. Simple software delay routines provide an easy way to insert delay functions into your program. Although the MicroBolt contains 2 high precision timer peripherals, using them for simple delays can be wasteful and disruptive to other interrupts within the system.

This example program creates 6 software delay routines for a 10 uS, 100 uS, 1 mS, 10 mS, 100 mS, and 1 Second delay. The MicroBolt P0_9 pin is toggled with these various delays as a way to show the delay accuracy.

It should be noted that software delay routines rely on the software compiler since it compiles the code to create the delay routines in the for-loop. For this reason, these delay routines would need to be verified on versions of ImageCraft ICCARM other than version 7.03 since they were developed on ICCARM version 7.03.

Waveforms:

The following waveform capture shows the 10 uS delay:



1 Module: RL2BCSXY

Sample Clock Speed: 100MHz (10ns)

Trigger Position (10%): □

Trigger Type: ○ Pattern ● Edge ○ Advanced

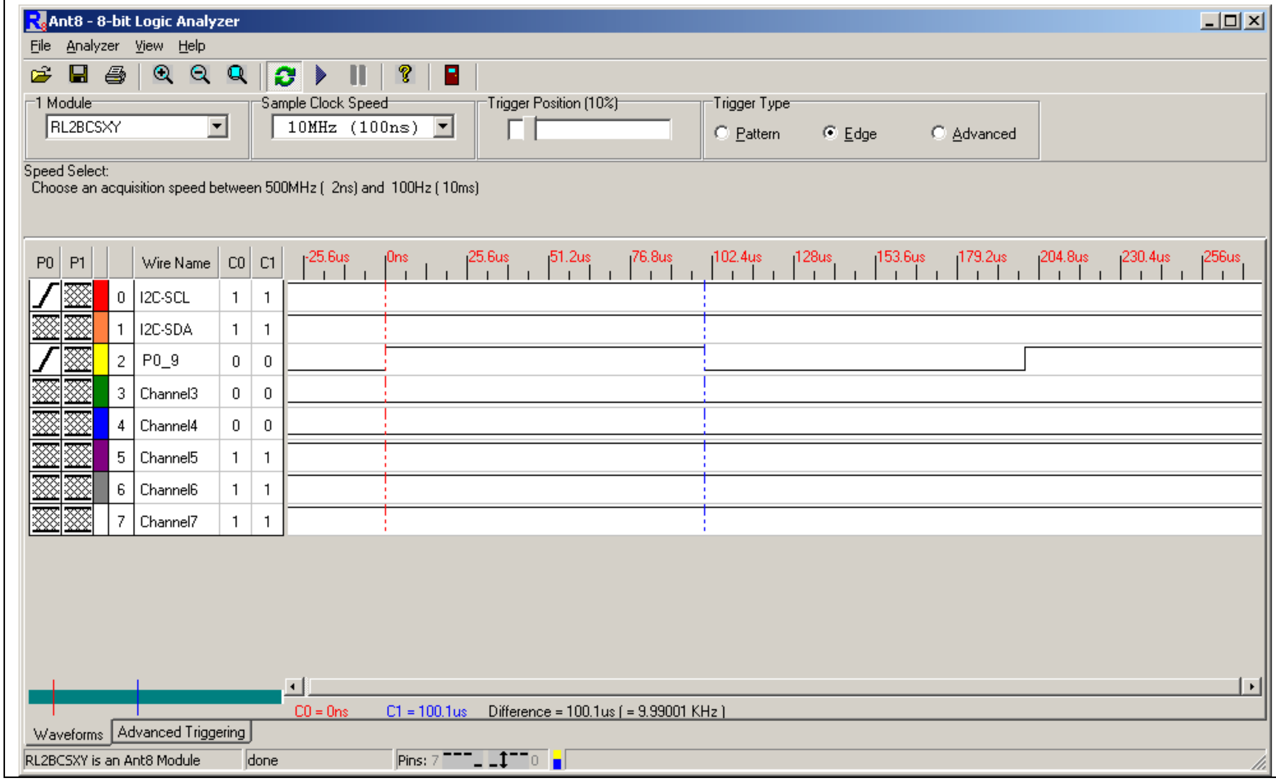
Speed Select:
Choose an acquisition speed between 500MHz (2ns) and 100Hz (10ms)

P0	P1			Wire Name	C0	C1
			0	I2C-SCL	1	1
			1	I2C-SDA	1	1
			2	P0_9	0	1
			3	Channel3	0	0
			4	Channel4	0	0
			5	Channel5	1	1
			6	Channel6	1	1
			7	Channel7	1	1

C0 = 0ns C1 = 10.03us Difference = 10.03us (= 99.7009 KHz)

RL2BCSXY is an Ant8 Module | done | Pins: 7 0

The following waveform capture shows the 100 uS delay:



1 Module: RL2BCSXY

Sample Clock Speed: 10MHz (100ns)

Trigger Position (10%): □

Trigger Type: ○ Pattern ● Edge ○ Advanced

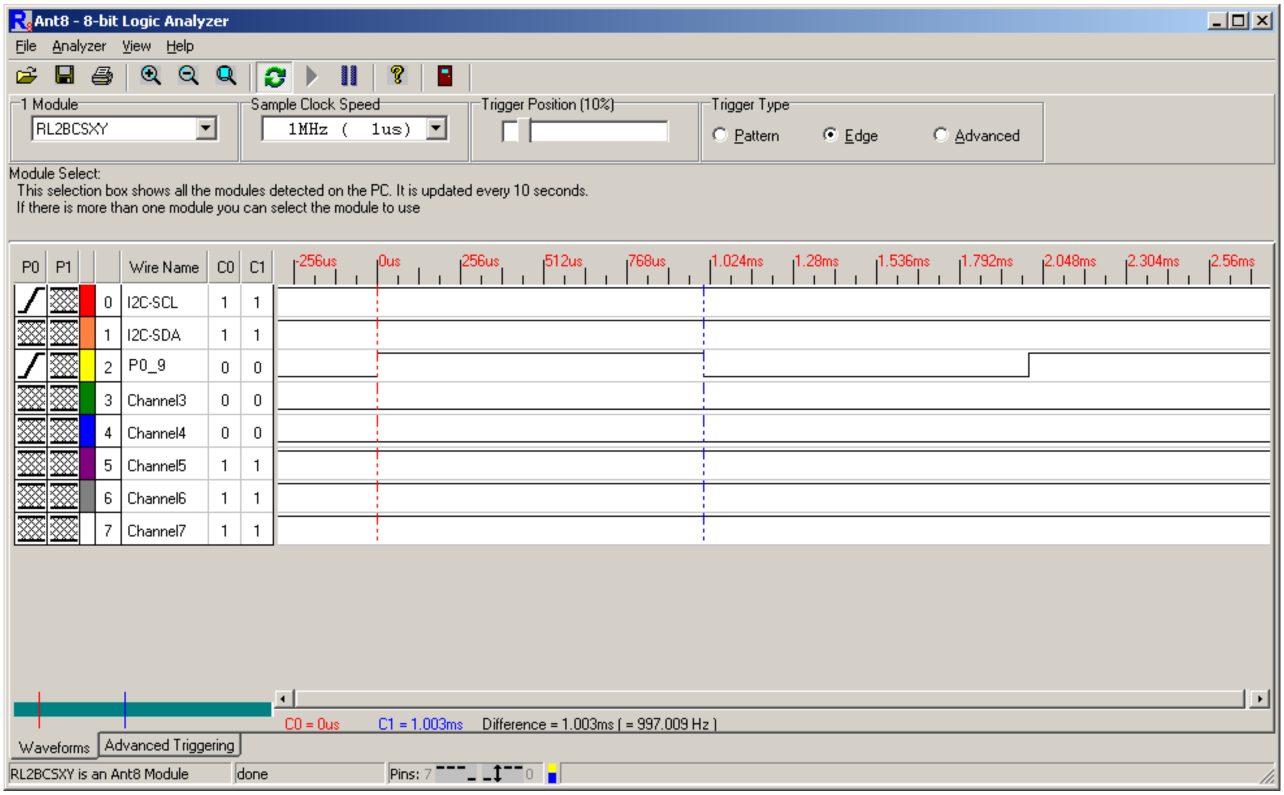
Speed Select:
Choose an acquisition speed between 500MHz (2ns) and 100Hz (10ms)

P0	P1			Wire Name	C0	C1
			0	I2C-SCL	1	1
			1	I2C-SDA	1	1
			2	P0_9	0	0
			3	Channel3	0	0
			4	Channel4	0	0
			5	Channel5	1	1
			6	Channel6	1	1
			7	Channel7	1	1

C0 = 0ns C1 = 100.1us Difference = 100.1us (= 9.99001 KHz)

RL2BCSXY is an Ant8 Module | done | Pins: 7 0

The following waveform capture shows the 1 mS delay:



1 Module: RL2BCSXY

Sample Clock Speed: 1MHz (1us)

Trigger Position (10%)

Trigger Type: ○ Pattern ● Edge ○ Advanced

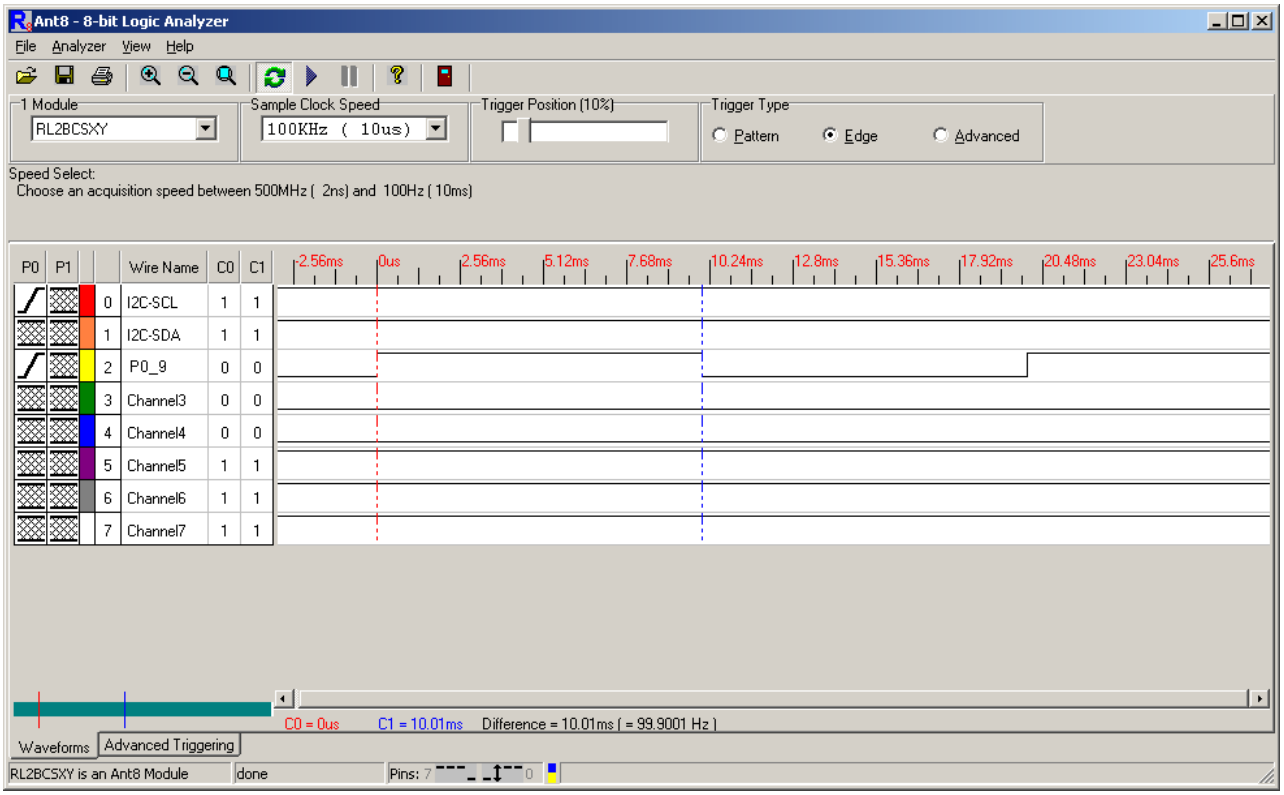
Module Select:
This selection box shows all the modules detected on the PC. It is updated every 10 seconds.
If there is more than one module you can select the module to use

P0	P1			Wire Name	C0	C1
/	▨	🟥	0	I2C-SCL	1	1
▨	▨	🟧	1	I2C-SDA	1	1
/	▨	🟨	2	P0_9	0	0
▨	▨	🟩	3	Channel3	0	0
▨	▨	🟦	4	Channel4	0	0
▨	▨	🟪	5	Channel5	1	1
▨	▨	⬛	6	Channel6	1	1
▨	▨	⬜	7	Channel7	1	1

C0 = 0us C1 = 1.003ms Difference = 1.003ms (= 997.009 Hz)

RL2BCSXY is an Ant8 Module | done | Pins: 7 0

The following waveform capture shows the 10 mS delay:



1 Module: RL2BCSXY

Sample Clock Speed: 100KHz (10us)

Trigger Position (10%)

Trigger Type: ○ Pattern ● Edge ○ Advanced

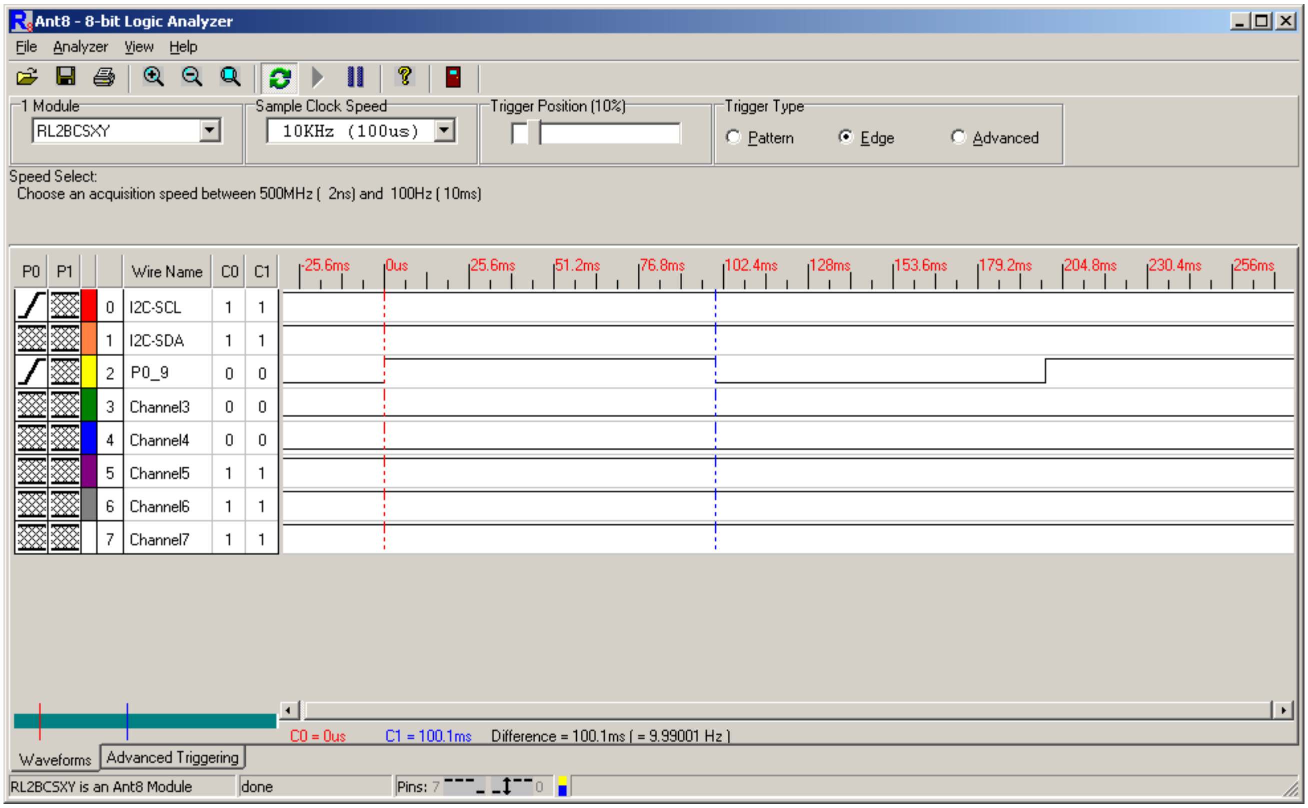
Speed Select:
Choose an acquisition speed between 500MHz (2ns) and 100Hz (10ms)

P0	P1			Wire Name	C0	C1
/	▨	🟥	0	I2C-SCL	1	1
▨	▨	🟧	1	I2C-SDA	1	1
/	▨	🟨	2	P0_9	0	0
▨	▨	🟩	3	Channel3	0	0
▨	▨	🟦	4	Channel4	0	0
▨	▨	🟪	5	Channel5	1	1
▨	▨	⬛	6	Channel6	1	1
▨	▨	⬜	7	Channel7	1	1

C0 = 0us C1 = 10.01ms Difference = 10.01ms (= 99.9001 Hz)

RL2BCSXY is an Ant8 Module | done | Pins: 7 0

The following waveform capture shows the 100 mS delay:



1 Module: RL2BCSXY

Sample Clock Speed: 10KHz (100us)

Trigger Position (10%)

Trigger Type: ○ Pattern ● Edge ○ Advanced

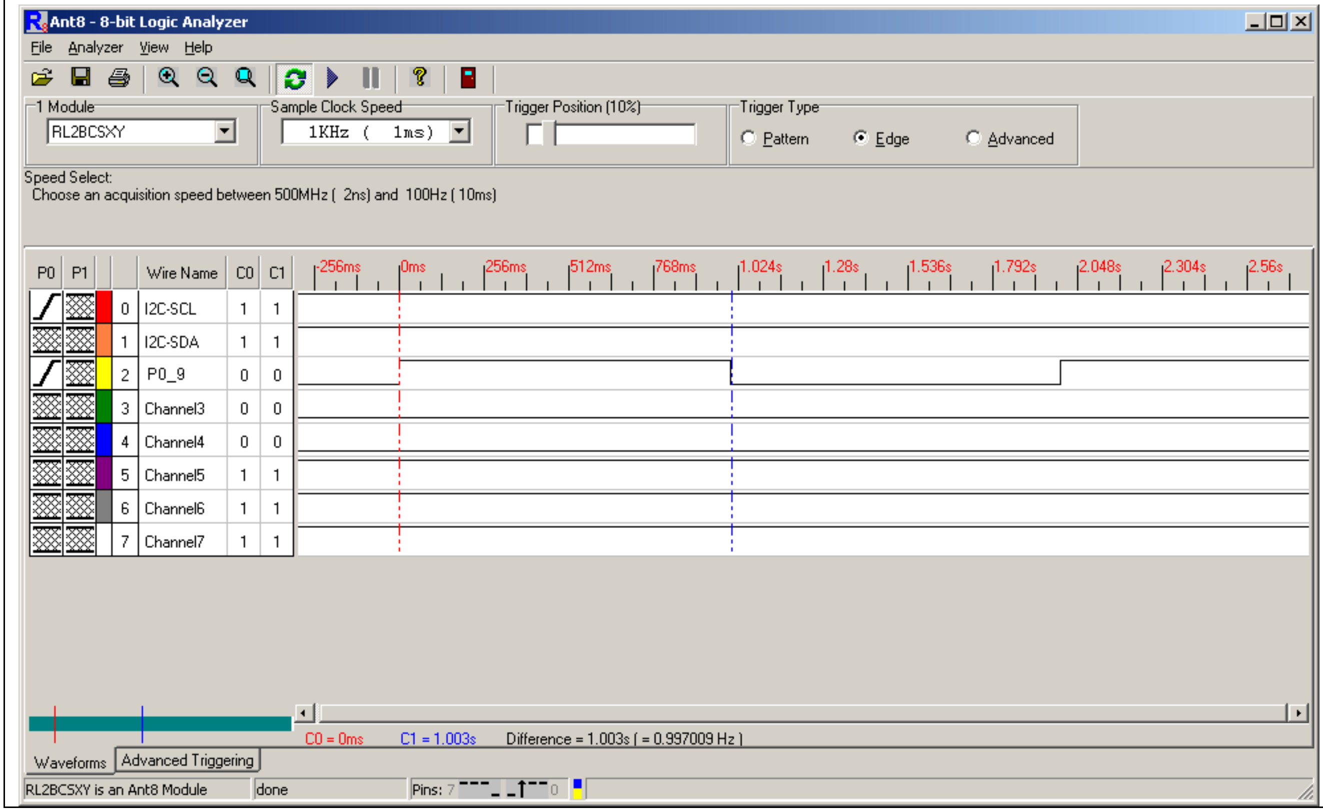
Speed Select:
Choose an acquisition speed between 500MHz (2ns) and 100Hz (10ms)

P0	P1			Wire Name	C0	C1
/	▨	🟥	0	I2C-SCL	1	1
▨	▨	🟧	1	I2C-SDA	1	1
/	▨	🟨	2	P0_9	0	0
▨	▨	🟩	3	Channel3	0	0
▨	▨	🟦	4	Channel4	0	0
▨	▨	🟪	5	Channel5	1	1
▨	▨	⬛	6	Channel6	1	1
▨	▨	⬜	7	Channel7	1	1

C0 = 0us C1 = 100.1ms Difference = 100.1ms (= 9.99001 Hz)

RL2BCSXY is an Ant8 Module | done | Pins: 7 0

The following waveform capture shows the 1 Second delay:



1 Module: RL2BCSXY

Sample Clock Speed: 1KHz (1ms)

Trigger Position (10%)

Trigger Type: ○ Pattern ● Edge ○ Advanced

Speed Select:
Choose an acquisition speed between 500MHz (2ns) and 100Hz (10ms)

P0	P1			Wire Name	C0	C1
/	▨	🟥	0	I2C-SCL	1	1
▨	▨	🟧	1	I2C-SDA	1	1
/	▨	🟨	2	P0_9	0	0
▨	▨	🟩	3	Channel3	0	0
▨	▨	🟦	4	Channel4	0	0
▨	▨	🟪	5	Channel5	1	1
▨	▨	⬛	6	Channel6	1	1
▨	▨	⬜	7	Channel7	1	1

C0 = 0ms C1 = 1.003s Difference = 1.003s (= 0.997009 Hz)

RL2BCSXY is an Ant8 Module | done | Pins: 7 0

Program Listing:

```
/*
|------------------------------------------------------------------------------------------
|------------------------------------------------------------------------------------------
| File Name                     : MicroBoltAccurateDelays.c
| Author                        : Micromint, Inc.
| Copyright                     : Copyright © 2005, Micromint, Inc.
| Creation Date                 : 12/30/05
| Version                       : 1.00
| Spaces per tab                : 2
| Description                   : Main C file
| Revision                      : Initial
|------------------------------------------------------------------------------------------
|------------------------------------------------------------------------------------------
*/

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

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

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

void main(void)
  {

  __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 and CCLK
|-------------------------
*/
  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 PCLK
|-------------------------
*/
  SCB_VPBDIV = 0;              // Peripheral clock is 1/4th Processor clock which equals 14.7456 MHz

/*
|-------------------------
| Configure VIC
|-------------------------
*/
  VICVectAddr0 = (unsigned)pll_isr;             // Assign the PLL lock ISR vector address
  VICVectCntl0 = INTERRUPT_CHANNEL_FOR_PLL;     // Assign the VIC address to the actual interrupt
  VICIntEnable = INTERRUPT_ENABLE_FOR_PLL;      // Enable the interrupt

/*
|-------------------------
| Config GPIO
|-------------------------
*/
  PCB_PINSEL0=0x00000000;
  PCB_PINSEL1=0x55400000;                       // Secondary JTAG pins

  GPIO_IODIR |= MICROBOLT_LED;                  // Setup MicroBolt LED as output
  GPIO_IODIR |= P0_9;                           // Setup P0_9 as output

  GPIO_IOSET = MICROBOLT_LED;                   // Turn on MicroBolt LED

  GPIO_IOCLR=0xffffffff;                        // Clear all pins to start with

/*
|----------------------------------------------------
|     Start of application
|----------------------------------------------------
*/
  while(1)                                      // Do this forever
    {
    GPIO_IOSET = MICROBOLT_LED;                 // Turn on MicroBolt LED

    GPIO_IOSET = P0_9;          // Measure delay via P0_9 On time (LPC2106 pin toggle takes about 1.4 uS)
    Delay10uS();                                // 10 uS Delay
    GPIO_IOCLR = P0_9;
    Delay10uS();

    GPIO_IOSET = P0_9;
    Delay100uS();                               // 100 uS Delay
    GPIO_IOCLR = P0_9;
    Delay100uS();

    GPIO_IOSET = P0_9;
    Delay1mS();                                 // 1 mS Delay
    GPIO_IOCLR = P0_9;
    Delay1mS();

    GPIO_IOSET = P0_9;
    Delay10mS();                                // 10 mS Delay
    GPIO_IOCLR = P0_9;
    Delay10mS();

    GPIO_IOSET = P0_9;
    Delay100mS();                               // 100 mS Delay
    GPIO_IOCLR = P0_9;
    Delay100mS();

    GPIO_IOCLR = MICROBOLT_LED;                 // Turn Off MicroBolt LED

    GPIO_IOSET = P0_9;
    Delay1Sec();                                // 1 Second Delay
    GPIO_IOCLR = P0_9;
    Delay1Sec();
    }
  }

/*
|----------------------------------------------------
|     Functions
|----------------------------------------------------
*/

/*
|-------------------------------------------------------------------------------------------
|     Function        :   Delay10uS
|     Inputs          :   None
|     Outputs         :   None
|     Purpose         :   Delay for 10 uS when PLLCFG is set to 0x23 (58.98 MHz)
|     Author          :   Micromint, Inc.
|------------------------------------------------------------------------------------------
*/
void Delay10uS(void)
  {
  volatile unsigned int Delay;

  for (Delay = 0; Delay < 7; Delay++);          // Delay for 10 uS including function call
  }

/*
|------------------------------------------------------------------------------------------
|     Function        :   Delay100uS
|     Inputs          :   None
|     Outputs         :   None
|     Purpose         :   Delay for 100 uS when PLLCFG is set to 0x23 (58.98 MHz)
|     Author          :   Micromint, Inc.
|------------------------------------------------------------------------------------------
*/
void Delay100uS(void)
  {
  volatile unsigned int Delay;

  for (Delay = 0; Delay < 109; Delay++);         // Delay for 100 uS including function call
  }

/*
|------------------------------------------------------------------------------------------
|     Function        :   Delay1mS
|     Inputs          :   None
|     Outputs         :   None
|     Purpose         :   Delay for 1 mS when PLLCFG is set to 0x23 (58.98 MHz)
|     Author          :   Micromint, Inc.
|------------------------------------------------------------------------------------------
*/
void Delay1mS(void)
  {
  volatile unsigned int Delay;

  for (Delay = 0; Delay < 1000; Delay++);       // Delay for 1 mS including function call
  for (Delay = 0; Delay < 100; Delay++);
  for (Delay = 0; Delay < 30; Delay++);
  }

/*
|------------------------------------------------------------------------------------------
|     Function        :   Delay10mS
|     Inputs          :   None
|     Outputs         :   None
|     Purpose         :   Delay for 10 mS when PLLCFG is set to 0x23 (58.98 MHz)
|     Author          :   Micromint, Inc.
|------------------------------------------------------------------------------------------
*/
void Delay10mS(void)
  {
  volatile unsigned int Delay;

  for (Delay = 0; Delay < 9200; Delay++);       // Delay for 10 mS including function call
  }

/*
|------------------------------------------------------------------------------------------
|     Function        :   Delay100mS
|     Inputs          :   None
|     Outputs         :   None
|     Purpose         :   Delay for 100 mS when PLLCFG is set to 0x23 (58.98 MHz)
|     Author          :   Micromint, Inc.
|------------------------------------------------------------------------------------------
*/
void Delay100mS(void)
  {
  volatile unsigned int Delay;
  for (Delay = 0; Delay < 92000; Delay++);      // Delay for 100 mS including function call
  }

/*
|------------------------------------------------------------------------------------------
|     Function        :   Delay1Sec
|     Inputs          :   None
|     Outputs         :   None
|     Purpose         :   Delay for 1 Second when PLLCFG is set to 0x23 (58.98 MHz)
|     Author          :   Micromint, Inc.
|------------------------------------------------------------------------------------------
*/
void Delay1Sec(void)
  {
  volatile unsigned int Delay;

  for (Delay = 0; Delay < 920000; Delay++);     // Delay for 1 Second including function call
  }

/*
|------------------------------------------------------------------------------------------
|     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;                      // Clear the PLL interrupt
  VICVectAddr = VIC_ACK;                        // Acknowledge Interrupt
  }
```