//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MSP-FET430P140 Demo - Timer\_A0, Capture of ACLK/8 using SMCLK

//

// Description: Input capture of ACLK/8 on P1.1(TA0)

// Run to breakpoint at the \_NOP() instruction to see 16 capture

// values and the differences.

// ACLK = 32768Hz, MCLK = SMCLK = default ~800kHz

// //\* An external watch crystal on XIN XOUT is required for ACLK \*//

// MSP430F149

// -----------------

// /|\| XIN|-

// | | | 32kHz

// --|RST XOUT|-

// | |

// | P2.0/ACLK|---+

// | | |

// | P1.1/TA0|<--+

// | |

// | P1.0|--->LED//

H. Grewal

// Texas Instruments Inc.

// Feb 2005

// Built with CCE Version: 3.2.0 and IAR Embedded Workbench Version: 3.21A

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <msp430x14x.h>

unsigned int new\_cap=0;

unsigned int old\_cap=0;

unsigned int cap\_diff=0;

unsigned int diff\_array[16]; // RAM array for differences

unsigned int capture\_array[16]; // RAM array for captures

unsigned char index=0;

unsigned char count = 0;

void main(void)

{

 volatile unsigned int i;

 WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer

 for (i=0; i<20000; i++) // Delay for crystal stabilization

 {

 }

 P1DIR = 0x01; // Set P1.0 out,1.1 input dir

 P1OUT &= ~0x01; // LED off

 P1SEL = 0x02; // Set P1.1 to TA0

 P2DIR = 0x01; // P2.0-ACLK

 P2SEL |= 0x01;

 BCSCTL1 |= DIVA\_3; // ACLK/8

 CCTL0 = CM\_1 + SCS + CCIS\_0 + CAP + CCIE; // Rising edge + CCI0A (P1.1)

 // + Capture Mode + Interrupt

 TACTL = TASSEL\_2 + MC\_2; // SMCLK + Continuous Mode

 \_BIS\_SR(LPM0\_bits + GIE); // LPM0 + Enable global ints

}

#pragma vector=TIMERA0\_VECTOR

\_\_interrupt void TimerA0(void)

{

 new\_cap = TACCR0;

 cap\_diff = new\_cap - old\_cap;

 diff\_array[index] = cap\_diff; // record difference to RAM array

 capture\_array[index++] = new\_cap;

 if (index == 16)

 {

 index = 0;

 P1OUT ^= 0x01; // Toggle P1.0 using exclusive-OR

 }

 old\_cap = new\_cap; // store this capture value

 count ++;

 if (count == 32)

 {

 count = 0;

 \_NOP(); // SET BREAKPOINT HERE

 }

}