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

/\* Lab 6 - Voltage ramp generator \*/

/\* \*/

/\* \*/

/\* \*/

/\* MSP430 Teaching ROM \*/

/\* \*/

/\* Exercise: Using the MSP-EXP430FG4618 Development Tool and the \*/

/\* MSP430FG4618 device implement a voltage ramp generator \*/

/\* Basic Timer 1 \*/

/\* \*/

/\* Instructions: \*/

/\* \*/

/\* The DAC module reference is obtained from the ADC module \*/

/\* \*/

/\* The DAC is configured with 12 bits resolution in straight binary format\*/

/\* \*/

/\* The DAC’s output value is updated every 1 msec by a Timer\_A ISR \*/

/\* \*/

/\* The buttons SW1 and SW2 are used to manually modify the DAC’s output \*/

/\* \*/

/\* Complete the code below: \*/

/\* - Configure FLL+ \*/

/\* - Select reference voltage \*/

/\* - Configure DAC12 \*/

/\* - Configure Timer\_A \*/

/\* - COnfigure P1 (buttons SW1 and SW2) \*/

/\* \*/

/\* \*/

/\* Copyright by Texas Instruments \*/

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

/\*Group #7 \*/

/\*Ron ben david\*/

/\*Yohav itzhak \*/

/\*Ben Reuven\*/

#include <msp430xG46x.h>

unsigned char flag;

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

// Port1 Interrupt Service Rotine

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

#pragma vector=PORT1\_VECTOR

\_\_interrupt void PORT1\_ISR (void)

{

if (P1IFG & 0x01) // SW1 generate interrupt

DAC12\_0DAT += 400; // DAC's output increases

if (P1IFG & 0x02) // SW2 generate interrupt

DAC12\_0DAT -= 400; // DAC's output decreases

P1IFG = 0x00; // clean all pending interrupts

}

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

// ISR to TACCRO from Timer A

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

#pragma vector=TIMERA0\_VECTOR

\_\_interrupt void TimerA0\_ISR (void)

{

DAC12\_0DAT++; // Increase DAC's output

if (DAC12\_0DAT == 0xfff)

DAC12\_0DAT = 0; // reset DAC's output

if (flag == 1) // if flag active exite LPM0

{

flag = 0;

LPM0\_EXIT;

}

}

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

// main

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

void main (void)

{

// Stop WatchDog

WDTCTL = WDTPW + WDTHOLD; // Stop WDT

// FLL+ configuration // ACLK - 32.768 kHz

FLL\_CTL0 |= DCOPLUS + XCAP18PF; // DCO+ set, freq = xtal x D x N+1

SCFI0 |= FN4; // x2 DCO freq, 8MHz nominal DCO

SCFQCTL = 121; // (121+1) x 32768 x 2 = 7.99 MHz

// DAC12 configuration

DAC12\_0DAT = 0; // DAC\_0 output 0V

ADC12CTL0 = REF2\_5V + REFON; // Internal 2.5V, Ref ON from AtoD12

TAR =0; // TAR reset

TACCR0 = 13600; // Delay to allow Ref to settle

TACCTL0 |= CCIE ; // Compare-mode CCR0 interrupt enabled

TACTL = 0x0210; // up mode, SMCLK, Timer\_A clear

flag = 1;

\_BIS\_SR(LPM0\_bits + GIE); // Enter LPM0, enable interrupts

DAC12\_0CTL =0x01c0; // DAC\_0 -> P6.6

// DAC\_1 -> P6.7

// DAC reference Vref

// 12 bits resolution

// Immediate load

// DAC full scale output

// Medium speed/current

// Straight binary

// Not grouped

// DAC12 enable

// Timer A configuration

TAR =0; // TAR reset

TACCTL0 =CCIE; // CCR0 interrupt enabled

TACCR0 =1; // 1 msec counting period

TACTL = 0x0110; // ACLK, div/1, up mode

// DAC0 outout port configuration (P6.6) // P6.6 configured as output DAC\_0

P6DIR &=0x040;

P6SEL |= 0x040;

// SW1 and SW2 ports configuration

P1SEL &= 0x03; // P1.0 and P1.1 I/O ports

P1DIR &= 0x03; // P1.0 and P1.1 digital inputs

P1IFG = 0; // clear all interrupts pending

P1IE |= 0x03; // enable port interrupts

// Interrupts enabled and enter in LPM3

\_BIS\_SR(LPM3\_bits + GIE);

}

