Assignment 8

Kyle Rosenthal 5/12/17 S17

Extra Credit done.

Youtube Link

https://www.youtube.com/watch?v=qDJF_OJ52iA

Code

main.c

```
//*****************************************************************************
//
// MSP432 main.c template - Empty main
//
//****************************************************************************

#include "msp.h"
#include "adc.h"
#include "uart.h"

void main(void)
{

    WDTCTL = WDTPW | WDTHOLD;           // Stop watchdog timer

    __enable_irq();

    Setup_UART();

    Setup_ADC(33,0);
      char arr[8];
      int i;
    while (1) {

        ADC_RequestNextSample();
        while (!ADC_CheckReady()) {}
        ADC_GetFormatedValue(arr);
        for (i = 0; i < 6; i++) {
            UART0Tx(arr[i]);
        }
        UART0Tx(' ');
        UART0Tx('V');
        UART0Tx(0x1b);
        UART0Tx('[');
        UART0Tx('H');
        for (i = 0; i < 20000; i++) {}
    }
}
```

uart.h

```c
/*
 * uart.h
 *
 *  Created on: May 8, 2017
 *      Author: kmrosent
 */

#ifndef UART_H_
#define UART_H_

int statusFlag;
int val;
void Setup_UART();
int readFlag();
void setFlag();
void clearFlag();
int readVal();
unsigned char UART0Rx(void);
unsigned char UART0Tx(unsigned char c);
void EUSCIA0_IRQHandler(void);

#endif /* UART_H_ */
```

uart.c

```c
/*
 * uart.c
 *
 *  Created on: May 8, 2017
 *      Author: kmrosent
 */

#include "uart.h"

#include "msp.h"

void Setup_UART() {
    val = 0;
    statusFlag = 0;

    __disable_irq();

    EUSCI_A0->CTLW0 |= BIT0;
    EUSCI_A0->MCTLW = 0;
    EUSCI_A0->CTLW0 = 0x0081;
    EUSCI_A0->BRW = 26;
    P1SEL0 |= (BIT2 + BIT3);
    P1SEL1 &= ~(BIT2 + BIT3);
    EUSCI_A0->CTLW0 &= ~BIT0;
    EUSCI_A0->IFG |= EUSCI_A_IFG_RXIFG;
    EUSCI_A0->IE |= EUSCI_A_IE_RXIE;
    //NVIC_SetPriority(EUSCIA0_IRQn, 4);
    NVIC_EnableIRQ(EUSCIA0_IRQn);
    __enable_irq();
}

int readFlag() {
    return statusFlag;
}

void setFlag() {
    statusFlag = 1;
}
void clearFlag() {
    statusFlag = 0;
}
int readVal() {
    int temp = val;
    val = -1;
    return temp;
}

/* read a character from UART0 */
unsigned char UART0Rx(void) {
    char c;
    while(!(EUSCI_A0->IFG & 0x01)) ;
    c = EUSCI_A0->RXBUF;
    return c;
}

/* write a character to UART */
unsigned char UART0Tx(unsigned char c) {
    while(!(EUSCI_A0->IFG&0x02)) ;
    EUSCI_A0->TXBUF = c;
    return c;
}

void EUSCIA0_IRQHandler(void) {
    char c = EUSCI_A0->RXBUF;

    if (c == '\r') {
        UART0Tx(c);
        c = '\n';
    }

    while(!(EUSCI_A0->IFG & 0x02)) {}
    EUSCI_A0->TXBUF = c;
}
```

adc.h

```
/*
 * adc.h
 *
 *  Created on: May 10, 2017
 *      Author: kmrosent
 */

#ifndef ADC_H_
```

```
#define ADC_H_

#define F_ADC_READ_ME 0
#define F_ADC_REQUEST 1
#define F_ADC_NO_OP   2

#define CAL     79 / 428 + 30

int vL, vH, adcflag;
unsigned long lastRead;

void Setup_ADC(int v_h, int v_l);
void ADC_RequestNextSample();
void ADC14_IRQHandler();

int ADC_CheckReady();
unsigned int ADC_GetRawValue();
void ADC_GetFormatedValue(char* value);

#endif /* ADC_H_ */
```

adc.c

```
/*
 * adc.c
 *
 *  Created on: May 10, 2017
 *      Author: kmrosent
 */

#include "adc.h"
#include "msp.h"

void Setup_ADC(int v_h, int v_l) {

    vL = v_l;
    vH = v_h;
    adcflag = F_ADC_NO_OP;

    P5->SEL0 |= BIT5;
    P5->SEL1 |= BIT5;

    //sample speed, sample and hold = 16, on
    ADC14->CTL0 = ADC14_CTL0_SHT0_2 | ADC14_CTL0_SHP | ADC14_CTL0_ON;
    //sample res = 14 bit
    ADC14->CTL1 = ADC14_CTL1_RES_3;

    ADC14->MCTL[0] |= ADC14_MCTLN_INCH_0;

    // int enable
    ADC14->IER0 |= ADC14_IER0_IE0;

    NVIC->ISER[0] = 1 << ((ADC14_IRQn) & 31);
    //wake on isr exit
    SCB->SCR &= ~SCB_SCR_SLEEPONEXIT_Msk;

}

void ADC_RequestNextSample() {
    //start sample
    if (adcflag != F_ADC_REQUEST) {
        ADC14->CTL0 |= ADC14_CTL0_ENC | ADC14_CTL0_SC;
    }
    adcflag = F_ADC_REQUEST;
}

void ADC14_IRQHandler() {
    lastRead = ADC14->MEM[0]; //output

    adcflag =F_ADC_READ_ME;
}

int ADC_CheckReady() {
    return adcflag == F_ADC_READ_ME;
}

unsigned int ADC_GetRawValue() {
    adcflag = F_ADC_NO_OP;
    return lastRead;
}

void ADC_GetFormatedValue(char* value) {
    adcflag = F_ADC_NO_OP;
    unsigned long long conversion = lastRead * CAL;
    int loc = 5;
    value[2] = '.';
    while (loc >= 0) {
        value[loc] = '0' + (conversion % 10);
        conversion /= 10;
        if (loc == 3) loc--;
        loc--;
    }
}
```