**Assignment 8:  
Analog to Digital**

**Cal Poly CPE 329-01**

**Spring 2017**

Brendan Baronia

Melinda Ong

Professor Gerfen, Jeffrey B.

**April 26, 2017**

***Code:***

adc.c:

#include "msp.h"  
#include "adc.h"  
  
void ADC\_init() {  
 ADC14->CTL0 = 0x00000010; /\*on, disabled during config\*/  
 ADC14->CTL0 |= 0x04080300; /\*SH pulse, sysclk, 32 samples, software trigger\*/  
  
 ADC14->CTL1 = 0x00000020; /\*12 bit resolution\*/  
 ADC14->MCTL[5] = 6; /\*A6 input, single ended, Vref = AVCC\*/  
 P4->SEL1 |= 0x80; /\*Configure P4.7\*/  
 P4->SEL0 |= 0x80;  
 ADC14->CTL1 |= 0x00050000; /\*convert for mem reg 5\*/  
 ADC14->CTL0 |= 2; /\*enable adc\*/  
}  
  
int adcReadVal() {  
 int result;  
  
 ADC14->CTL0 |= 1; /\*start conversion\*/  
 while (!ADC14->IFGR0); /\*wait until conversion complete\*/  
 result = ADC14->MEM[5]; /\*read result\*/  
  
 return result;  
}

adc.h:

#ifndef ADC\_H\_  
#define ADC\_H\_  
  
void ADC\_init();  
int adcReadVal();  
  
#endif /\* ADC\_H\_ \*/

main.c:

#include "msp.h"  
#include "uart.h"  
#include "adc.h"  
  
#define CONVERSION 1.3  
  
void main() {  
 WDT\_A->CTL = WDT\_A\_CTL\_PW | WDT\_A\_CTL\_HOLD; // Stop watchdog timer  
  
 /\*set up UART\*/  
 UART0\_init();  
 ADC\_init();  
  
 while(1) {  
 /\*get adc value\*/  
 int val = adcReadVal();  
  
 /\*convert adc value to mV\*/  
 val = val / CONVERSION;  
  
 /\*read value into a character array in reverse order  
 \* each character formatted into ascii  
 \*/  
 char values[20];  
 int counter;  
 for (counter = 0; val; counter++) {  
 values[counter] = (val % 10) + '0';  
 val = val / 10;  
 }  
  
 /\*echo characters in uart\*/  
 while (--counter > 3) {  
 uartPrint(values[counter]);  
 }  
  
 uartPrint('.');  
 uartPrint(values[counter--]);  
 uartPrint(values[counter--]);  
 uartPrint(' ');  
 uartPrint('V');  
 uartPrint('\r');  
 uartPrint('\n');  
  
 for(counter = 0; counter < 100000; counter++);  
 }  
}

uart.c:

#include "uart.h"  
#include "msp.h"  
  
/\*UART setup\*/  
void UART0\_init() {  
 EUSCI\_A0->CTLW0 |= 1; /\*Reset mode for config\*/  
 EUSCI\_A0->MCTLW = 0; /\*Disable oversampling\*/  
 EUSCI\_A0->CTLW0 = 0x0081; /\*1 stop bit, no parity, SMCLK, 8 bit data\*/  
 EUSCI\_A0->BRW = 26;  
 P1->SEL0 |= 0x0C; /\*P1.3, P1.2 for UART\*/  
 P1->SEL1 &= -0x0C;  
 EUSCI\_A0->CTLW0 &= ~1; /\*Out of reset mode\*/  
// EUSCI\_A0->IE |= 1; /\*enable receive interrupt\*/  
}  
  
/\*Sends c over uart\*/  
void uartPrint(char c) {  
 int counter;  
 /\*echo c to console\*/  
 EUSCI\_A0->TXBUF = c;  
 for(counter = 0; counter < 100; counter++);  
}

uart.h:

#ifndef UART\_H\_  
#define UART\_H\_  
  
/\* Sets up UART0  
 \*/  
void UART0\_init();  
  
/\* Sends c over uart  
 \*/  
void uartPrint(char c);  
  
#endif /\* UART\_H\_ \*/