#include "L4\_123.h"

#define F\_CPU 16000000

#include <asf.h>

#include <stdio.h>

#include <avr/delay.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

//Parity 0

#define OFF\_PARITY 0

#define ODD\_PARITY 1

#define EVEN\_PARITY 2

#define BIT\_FORMAT\_8bit 1

//StopBit

#define STOP\_BIT\_1bit 1

#define STOP\_BIT\_2bit 2

char\* ReadString( unsigned port);

int WriteString(unsigned port, char \*szOutput);

//Task 1 Functions

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

int ADC\_Read(void)

{

int result = 0;

// reset the converter

ADCSRA |= (1<<ADSC); //

while(!(ADCSRA & (1 << ADIF))); // ADIF turns on after reset

ADCSRA |= (1<<ADSC); // start conversion

while(!(ADCSRA & (1 << ADIF))); // ADIF turns on after conversion

result = ADCL; // read 1st lower part

result += (ADCH << 8); // read upper part

return result;

}

void ADC\_Init(char port)

{

if (port < 15)

{

ADMUX = (1 << REFS1) | (1 << REFS0) | port; // ref Volage – 2.5V

// set frequency 1MHz / 8 = 125kHz

ADCSRA = (1<<ADPS1) | (1<<ADPS0); //prescale

ADCSRA |= 1 << ADEN; // ADC Enable

}

}

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

int main (void)

{

int result; // variable to read integer ADC value

ADC\_Init(1); // initialize the port what we want

double voltage; // double type value to convert ADC value

char string[30]="ADC starts.. "; // print the string in the beginning

SetLineParameters( 0, 9600, EVEN\_PARITY, BIT\_FORMAT\_8bit, STOP\_BIT\_2bit ); // init board

WriteString((unsigned) 0, string); // write the started string in the terminal

//Task 1

while(1)

{

result=ADC\_Read();// read integer

voltage=(double)result; // convert to double

voltage=(voltage\*2.5)/1023; // convert to decimal

sprintf(string,"\rResult = %.5f Volts", voltage); // set double value to string

WriteString((unsigned) 0, string); // write the voltage on the screen

\_delay\_ms(1000); // wait 1 second before clean the screen

WriteString((unsigned) 0, "\033[2J"); // clean a screen to measure again

}

}