/\*

\* GccApplication1.cpp

\*

\* Created: 9/30/2015 12:57:29 AM

\* Author: Ismail

\*/

#ifndef F\_CPU

#define F\_CPU 1000000UL // set the CPU clock

#endif

//FOR LCD

#define RS 6

#define E 7

#define ctrl PORTD

#include <stdio.h>

#include <avr/io.h>

#include <util/delay.h>

#include <stdlib.h>

// for conversion from long to float

int i;

int j;

char arr[10];

char buf[10];

// ADC Variables

unsigned long volt;

unsigned long v[10];

double conv;

//.............................Begin LCD Funcations......................................................

int lcd\_data(char t)

{ ctrl |= (1<<RS);

PORTB=t;

ctrl |= (1<<E);

\_delay\_ms(1);

ctrl &= ~(1<<E);

\_delay\_ms(1);

t = t << 4;

PORTB=t;

ctrl |= (1<<E);

\_delay\_ms(1);

ctrl &= ~(1<<E);

\_delay\_ms(1);

return 0;}

int writecmd(char z)

{ctrl &= ~(1<<RS);

PORTB=z;

ctrl |= (1<<E);

\_delay\_ms(1);

ctrl &= ~(1<<E);

\_delay\_ms(1);

z = z << 4;

PORTB=z;

ctrl |= (1<<E);

\_delay\_ms(1);

ctrl &= ~(1<<E);

\_delay\_ms(1);

return 0;}

void lcd\_print(char const \*str)

{unsigned char k=0;

while (str[k]!=0)

{lcd\_data(str[k]);

k++;}}

void lcd\_init(void)

{writecmd(0x02);

writecmd(0x28);

writecmd(0x0c);

writecmd(0x01);

writecmd(0x06);}

void lcd\_gotoxy(unsigned char x, unsigned char y)

{

unsigned char firstcharadrs[] = {0x80, 0xC0,0x94,0xD4};

writecmd(firstcharadrs[y-1] + x - 1);

\_delay\_us(100);

}

//\_\_\_\_\_\_\_\_\_\_\_End of LCD Functions\_\_\_\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_UART Functions\_\_\_\_\_\_\_\_\_\_

void uart\_init(){

UBRRH=0x00;

UBRRL=12;

UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (1<<U2X) | (0<<MPCM);

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);

}

void uart\_transmit (unsigned char data){

while (!( UCSRA & (1<<UDRE)));

// wait for the register

UDR = data;

// loading the data in register

}

void string\_transmit(char const \*str){

unsigned char k=0;

while (str[k]!=0)

{

uart\_transmit (str[k]);

k++;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADC Functions Start \*\*\*\*\*\*\*\*\*\*\*\*/

void adc\_init()

{

ADMUX = (1<<REFS0) | (0<<REFS1);

// ADC Enable and prescaler of 128

// 8000000/128 = 62500

ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

}

// read adc value

int adc\_read(int ch)

{

// select the corresponding channel 0~7

// ANDing with '7' will always keep the value

// of 'ch' between 0 and 7

ch &= 0b00000111; // AND operation with 7

ADMUX = (ADMUX & 0xF8)|ch; // clears the bottom 3 bits before ORing

// start single conversion

// write '1' to ADSC

ADCSRA |= (1<<ADSC);

// wait for conversion to complete

// ADSC becomes '0' again

// till then, run loop continuously

while(ADCSRA & (1<<ADSC));

return (ADC);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADC Functions End \*\*\*\*\*\*\*\*\*\*\*\*/

void long\_float(){

volt = (int)conv;

if (volt < 1)

i=0;

if (volt < 10 && volt >= 1)

i=1;

if (volt >=10 && volt < 100)

i=2;

if (volt >=100)

i=3;

volt = conv\*1000;

ltoa(volt, buf,10);

for(j=0 ; i < 5 && j < 3;i++,j++)

arr[j] = buf[i];

volt = volt/1000;

ltoa(volt, buf,10);

string\_transmit(buf);

uart\_transmit('.');

string\_transmit(arr);

string\_transmit("\r\n");

lcd\_gotoxy(1,2);

lcd\_print(buf);

lcd\_print(".");

lcd\_print(arr);

lcd\_gotoxy(1,1);

}

int main(void)

{

// PORT initialization

DDRB = 0xff;

DDRC = 0x00;

DDRD = 0b11000000;

//\_\_LCD Initialization\_\_\_\_

lcd\_init();

lcd\_gotoxy(1,1);

//UART Initializaion

uart\_init();

// initialize ADC

adc\_init();

// loop forever

while(1){

for (i=0; i <= 9; i++){

v[i] = adc\_read(0);

volt = volt + v[i];

\_delay\_us(10);

}

volt = volt /10;

conv = (double)volt \* 5/1024;

conv = conv \* 110/10;

string\_transmit("Voltage = ");

lcd\_print("Voltage = ");

long\_float();

string\_transmit("\*\*\*END\*\*\*\n\r");

\_delay\_ms(700);

}

}