```
// miAT5
//Voltimetro vía RS232->PC
// SETA pruebas xsetaseta@gmail.com

#include <avr/io.h>
#include <stdio.h>

#define Set_Datos                       PORTB|=_BV(5)
#define Clear_Datos                 PORTB&=~_BV(5)
#define Set_Enable                  PORTB|=_BV(4)
#define Clear_Enable                PORTB&=~_BV(4)

#define F_CPU          1000000

void LCD_INI(void);
void LCD_CLS(void);
void LCD_HOME(void);
void LCD_AT(char x);
void Pon4bits(char x);
void ENABLE(void);
void SENDI(char x);
void SENDCHAR(char x);
void SENDCADE(char *x);
void PrintAtNum(unsigned char x,int numero);
void PrintAtStr(unsigned char x,char *string);

void TXchar(unsigned char c);
void TXNum(int numero);
void TXstr(char *string);

void startADC(void);



void delay_ms(unsigned char time_ms);
void delay_10us(unsigned char time_10us);


char VARI[17];  //Modificar para numeros mayores de 16


int main(void)
{
uint8_t countval;
unsigned char volH;
unsigned char volL;
unsigned int temp;
unsigned int temp2;
unsigned int temp3;

      PORTC=0X00;
      PORTB=0x00;
      PORTD=0x00;
      DDRD=0xff;      //puerto D como salida
      DDRC=0x00;      //puerto C como entrada
      DDRB=0xff; // puerto B como salida

      LCD_INI();
      LCD_CLS();
      LCD_HOME();

      UBRR0H = 0; // h=0 l=12 udx0=0 4800 bds. 1Mhz
      UBRR0L = 12; //h=0 l=12 udx0=1 9600 bds. 1Mhz
```

```
        UCSR0A = (1<<U2X0);
        UCSR0C = (1<<UCSZ00)|(1<<UCSZ01); //1bits de parada 8bits de datos
        UCSR0B = (1<<TXEN0); //Solo trasmisión     2bisParad=(1<<USBS0)|

        ADMUX=0xC0; //1.1V,derecha,ADC0
        ADCSRA=0x80; //dividido 2
                        //ADCSRA=0xA0 //Esto es para ATMEGA8
                        //o sin espera terminar conversión
        PrintAtStr(0,"Vol. 0-1V 10bits");

        countval = 0;
        while(1)
        {

                delay_ms(5);

                ADMUX&=~_BV(0); //Selecciona ADC0

                startADC();
                startADC();
                startADC();

                volL=ADCL;
                volH=ADCH;
                temp=ADCH;
                temp=temp << 8;
                temp=temp+volL;
                //temp2=(temp*54); //Para 1,1V
                //temp3=temp2/50;
                temp2=(temp*68);        //Para 1,080->67, Para 1,090->68
                temp3=temp2 >> 6;       // Dividido entre 64
                TXNum(temp3);
                TXchar(32);
                LCD_AT(64);
                sprintf(VARI,"%.4d",temp3);
                SENDCADE(VARI);
                //_____________________________________


                ADMUX|=_BV(0);  //Selecciona ADC1

                startADC();
                startADC();
                startADC();

                volL=ADCL;
                volH=ADCH;
                temp=ADCH;
                temp=temp << 8;
                temp=temp+volL;
                //temp2=(temp*54); //Para 1,1V
                //temp3=temp2/50;
                temp2=(temp*68);        //Para 1,080- Para 1,090 utilizar 68
                temp3=temp2 >> 6;       // Dividido entre 64
                TXNum(temp3);
                TXchar(13);
                LCD_AT(76);
                sprintf(VARI,"%.4d",temp3);
                SENDCADE(VARI);

                if(countval & 1) PORTD|=_BV(0); else PORTD&=~_BV(0);
                countval++;
        }
//---------
```

```
}

void startADC(void)
{

	ADCSRA|=_BV(6); // inicio ADC
	while( (ADCSRA & _BV(ADSC)) != 0);   // wait until conversion complete

}
void TXchar(unsigned char c)
{
	while ( !( UCSR0A & (1<<UDRE0)) ) ; // Wait for empty transmit buffer
	UDR0 = c;
}
void TXNum(int numero)
{
	unsigned char z;
	sprintf(VARI,"%d",numero);
	for(z=0;z<17 && VARI[z]!=0;z++)
		{TXchar(VARI[z]);}
}
void TXstr(char *string)
{
	unsigned char z;
	for(z=0;z<17 && *string!=0;z++,string++)
		{TXchar(*string);}
}
void PrintAtStr(unsigned char x,char *string)
{
	LCD_AT(x);
	SENDCADE(string);
}
void PrintAtNum(unsigned char x,int numero)
{
	LCD_AT(x);
	sprintf(VARI,"%d",numero);
	SENDCADE(VARI);
}
void LCD_INI(void)
{
	delay_ms(250);
	Pon4bits(0x03);
	Clear_Datos;
	ENABLE();
	Set_Datos;
	delay_ms(30);
	Pon4bits(0x03);
	Clear_Datos;
	ENABLE();
	Set_Datos;
	delay_ms(30);
	Pon4bits(0x03);
	Clear_Datos;
	ENABLE();
	Set_Datos;
	delay_ms(30);

	Pon4bits(0x02); // modo 4 bits
	Clear_Datos;
	ENABLE();
	Set_Datos;

	SENDI(0x2c); //modo 4 bits, dos lineas
```

```
        SENDI(0x0f); // cursor con parpadeo
        SENDI(0x04);

}

void Pon4bits(char x)
{
 char z;
 z=PORTB & 0xf0;
 x= x & 0x0f;
 PORTB= x |z;
}
void LCD_CLS(void)
{
        SENDI(1); //borra pantalla
}
void LCD_HOME(void)
{
        SENDI(2); //cursor al inicio
}
void LCD_AT(char x) //0 Comienzo línea1, 64 Comienzo línea2
{
        x=x | 128;
        SENDI(x);
}
void SENDCADE(char *x)
{
 char z;
        for(z=0;z<33 && *x!=0;z++,x++)
                {SENDCHAR(*x);}
}
void SENDCHAR(char x)
{
 char z;
        z=x >> 4;
        Pon4bits(z); //4 bytes de mas peso
        ENABLE();
        Pon4bits(x); //4 bytes de menos peso
        ENABLE();
}
//Manda datos de control
void SENDI(char x)
{
        Clear_Datos;
        SENDCHAR(x);
        delay_10us(250); //Tiempos para reformar
        Set_Datos;
}
void ENABLE(void)
{
        Set_Enable;
        delay_10us(250); //Tiempos para reformar
        Clear_Enable;
}

// 4 ciclos*delay*time_10us+5*time_10us
void delay_10us(unsigned char time_10us)
{
        unsigned short delay_count = F_CPU / 400000; //para 1Mhz->2

        unsigned short cnt;
        asm volatile ("\n"
                "L_dl1%=:\n\t"
                "mov %A0, %A2\n\t"
```

```
                "mov %B0, %B2\n"
                "L_dl2%=:\n\t"
                "sbiw %A0, 1\n\t"
                "brne L_dl2%=\n\t"
                "dec %1\n\t"
                        "brne L_dl1%=\n\t"
                        :"=&w" (cnt)
                :"r"(time_10us), "r"((unsigned short) (delay_count))
        );
}

// 4 ciclos*delay*time_ms+5*time_ms
void delay_ms(unsigned char time_ms)
{
        unsigned short delay_count = F_CPU / 4000; //para 1Mhz->250

        unsigned short cnt;
        asm volatile ("\n"
                "L_dl1%=:\n\t"
                "mov %A0, %A2\n\t"
                "mov %B0, %B2\n"
                "L_dl2%=:\n\t"
                "sbiw %A0, 1\n\t"
                "brne L_dl2%=\n\t"
                "dec %1\n\t"
                        "brne L_dl1%=\n\t"
                        :"=&w" (cnt)
                :"r"(time_ms), "r"((unsigned short) (delay_count))
        );
}
```