

```

1 // miAT4
2 // SETA pruebas xsetaseta@gmail.com
3
4 #include <avr/io.h>
5 #include <stdio.h>
6
7 #define Set_Datos          PORTB|=_BV(5)
8 #define Clear_Datos       PORTB&=~_BV(5)
9 #define Set_Enable        PORTB|=_BV(4)
10 #define Clear_Enable      PORTB&=~_BV(4)
11
12 #define F_CPU              1000000
13
14 void LCD_INI(void);
15 void LCD_CLS(void);
16 void LCD_HOME(void);
17 void LCD_AT(char x);
18 void Pon4bits(char x);
19 void ENABLE(void);
20 void SENDI(char x);
21 void SENDCHAR(char x);
22 void SENDCADE(char *x);
23 void PrintAtNum(unsigned char x,int numero);
24 void PrintAtStr(unsigned char x,char *string);
25
26 void TXchar(unsigned char c);
27 void TXNum(int numero);
28 void TXstr(char *string);
29
30
31
32
33
34 void delay_ms(unsigned char time_ms);
35 void delay_10us(unsigned char time_10us);
36
37
38 char VARI[17]; //Modificar para numeros mayores de 16
39
40
41 int main(void)
42 {
43     uint8_t countval;
44     unsigned char volH;
45     unsigned char volL;
46     unsigned int temp;
47     unsigned int temp2;
48     unsigned int temp3;
49
50     PORTC=0x00;
51     PORTB=0x00;
52     PORTD=0x00;
53     DDRD=0xff; //puerto D como salida
54     DDRc=0x00; //puerto C como entrada
55     DDRB=0xff; // puerto B como salida
56
57     LCD_INI();
58     LCD_CLS();
59     LCD_HOME();
60
61     UBRR0H = 0; // h=0 l=12 udx0=0 4800 bds. 1Mhz
62     UBRR0L = 12; //h=0 l=12 udx0=1 9600 bds. 1Mhz
63     UCSR0A = (1<<U2X0);
64     UCSR0C = (1<<UCSZ00)|(1<<UCSZ01); //1bits de parada 8bits de datos
65     UCSR0B = (1<<TXEN0); //solo trasmisión 2bisParad=(1<<USBS0)|
66
67     ADMUX=0xc0; //1.1v,derecha,ADC0
68     ADCSRA=0x80; //dividido 2
69     //ADCSRA=0xA0 //Esto es para ATMEGA8
70     //o sin espera terminar conversión
71     PrintAtStr(0,"Vol. 0-1V 10bits");
72
73     countval = 0;
74     while(1)
75     {

```

```

76     delay_ms(30);
77     ADCSRA|=_BV(6); // inicio ADC
78     while( (ADCSRA & _BV(ADSC)) != 0); // wait until conversion complete
79     //no se mira si ha terminado la conversión
80
81     volL=ADCL;
82     volH=ADCH;
83     temp=ADCH;
84     temp=temp << 8;
85     temp=temp+volL;
86
87     LCD_AT(76);
88     sprintf(VARI, "%.4d", temp);
89     SENDCADE(VARI);
90     //-----
91
92     //temp2=(temp*54); //Para 1,1v
93     //temp3=temp2/50;
94
95     temp2=(temp*67); //Para 1,080
96     temp3=temp2 >> 6; // Dividido entre 64
97
98     LCD_AT(64);
99     sprintf(VARI, "%.4d", temp3);
100    SENDCADE(VARI);
101    SENDCADE(" mv");
102
103    TXNum(temp3);
104    TXstr(" mv");
105    TXchar(13);
106
107    if(countval & 1) PORTD|=_BV(0); else PORTD&=~_BV(0);
108    countval++;
109    }
110    //-----
111
112    }
113
114    void TXchar(unsigned char c)
115    {
116        while ( !( UCSR0A & (1<<UDRE0)) ); // Wait for empty transmit buffer
117        UDR0 = c;
118    }
119    void TXNum(int numero)
120    {
121        unsigned char z;
122        sprintf(VARI, "%d", numero);
123        for(z=0; z<17 && VARI[z]!=0; z++)
124            {TXchar(VARI[z]);}
125    }
126    void TXstr(char *string)
127    {
128        unsigned char z;
129        for(z=0; z<17 && *string!=0; z++, string++)
130            {TXchar(*string);}
131    }
132    void PrintAtStr(unsigned char x,char *string)
133    {
134        LCD_AT(x);
135        SENDCADE(string);
136    }
137    void PrintAtNum(unsigned char x,int numero)
138    {
139        LCD_AT(x);
140        sprintf(VARI, "%d", numero);
141        SENDCADE(VARI);
142    }
143    void LCD_INI(void)
144    {
145        delay_ms(250);
146        Pon4bits(0x03);
147        Clear_Datos;
148        ENABLE();
149        Set_Datos;
150        delay_ms(30);

```

```

151 Pon4bits(0x03);
152 Clear_Datos;
153 ENABLE();
154 Set_Datos;
155 delay_ms(30);
156 Pon4bits(0x03);
157 Clear_Datos;
158 ENABLE();
159 Set_Datos;
160 delay_ms(30);
161
162 Pon4bits(0x02); // modo 4 bits
163 Clear_Datos;
164 ENABLE();
165 Set_Datos;
166
167 SENDI(0x2c); //modo 4 bits, dos lineas
168 SENDI(0x0f); // cursor con parpadeo
169 SENDI(0x04);
170
171 }
172
173 void Pon4bits(char x)
174 {
175     char z;
176     z=PORTB & 0xf0;
177     x= x & 0x0f;
178     PORTB= x |z;
179 }
180 void LCD_CLS(void)
181 {
182     SENDI(1); //borra pantalla
183 }
184 void LCD_HOME(void)
185 {
186     SENDI(2); //cursor al inicio
187 }
188 void LCD_AT(char x) //0 Comienzo línea1, 64 Comienzo línea2
189 {
190     x=x | 128;
191     SENDI(x);
192 }
193 void SENDCADE(char *x)
194 {
195     char z;
196     for(z=0; z<33 && *x!=0; z++, x++)
197         {SENDCHAR(*x);}
198 }
199 void SENDCHAR(char x)
200 {
201     char z;
202     z=x >> 4;
203     Pon4bits(z); //4 bytes de mas peso
204     ENABLE();
205     Pon4bits(x); //4 bytes de menos peso
206     ENABLE();
207 }
208 //Manda datos de control
209 void SENDI(char x)
210 {
211     Clear_Datos;
212     SENDCHAR(x);
213     delay_10us(250); //Tiempos para reformar
214     Set_Datos;
215 }
216 void ENABLE(void)
217 {
218     Set_Enable;
219     delay_10us(250); //Tiempos para reformar
220     Clear_Enable;
221 }
222
223 // 4 ciclos*delay*time_10us+5*time_10us
224 void delay_10us(unsigned char time_10us)
225 {

```

```

226     unsigned short delay_count = F_CPU / 40000; //para 1Mhz->2
227
228     unsigned short cnt;
229     asm volatile (
230         "\n"
231         "L_d11%=: \n\t"
232         "mov %A0, %A2 \n\t"
233         "mov %B0, %B2 \n\t"
234         "L_d12%=: \n\t"
235         "sbiw %A0, 1 \n\t"
236         "brne L_d12%=: \n\t"
237         "dec %1 \n\t"
238         "brne L_d11%=: \n\t"
239         : "=w" (cnt)
240         : "r"(time_10us), "r"((unsigned short) (delay_count))
241     );
242 }
243 // 4 ciclos*delay*time_ms+5*time_ms
244 void delay_ms(unsigned char time_ms)
245 {
246     unsigned short delay_count = F_CPU / 4000; //para 1Mhz->250
247
248     unsigned short cnt;
249     asm volatile (
250         "\n"
251         "L_d11%=: \n\t"
252         "mov %A0, %A2 \n\t"
253         "mov %B0, %B2 \n\t"
254         "L_d12%=: \n\t"
255         "sbiw %A0, 1 \n\t"
256         "brne L_d12%=: \n\t"
257         "dec %1 \n\t"
258         "brne L_d11%=: \n\t"
259         : "=w" (cnt)
260         : "r"(time_ms), "r"((unsigned short) (delay_count))
261     );
262 }
263

```