

```

1  // miAT3
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
27 void delay_ms(unsigned char time_ms);
28 void delay_10us(unsigned char time_10us);
29
30
31 char VARI[17]; //Modificar para numeros mayores de 16
32
33
34 int main(void)
35 {
36     uint8_t countval;
37     unsigned char volH;
38     unsigned char volL;
39     unsigned int temp;
40     unsigned int temp2;
41     unsigned int temp3;
42     unsigned int media;
43     unsigned char x;
44
45     PORTC=0x00;
46     PORTB=0x00;
47     PORTD=0x00;
48     DDRD=0xff; //puerto D como salida
49     DDRC=0x00; //puerto C como entrada
50     DDRB=0xff; // puerto B como salida
51
52     LCD_INI();
53     LCD_CLS();
54     LCD_HOME();
55
56
57     ADMUX=0xC0; //1.1V,derecha,ADC0
58     ADCSRA=0x80; //dividido 2
59                     //ADCSRA=0xA0 //Esto es para ATMEGA8
60                     //o sin espera terminar conversión
61     PrintAtStr(0,"Vol. 0-1V 10bits");
62
63     countval = 0;
64
65     while(1)
66     {
67         media=0;
68         for(x=0;x<16;x++) //Toma 16 valores y hace un promedio
69         {
70             delay_ms(10);
71             ADCSRA|=_BV(6); // inicio ADC
72             while( (ADCSRA & _BV(ADSC)) != 0); // wait until conversion complete

```

```

73                                     //no se mira si ha terminado la conversión
74         delay_ms(25);
75         volL=ADCL;
76         volH=ADCH;
77         temp=ADCH;
78         temp=temp << 8;
79         temp=temp+volL;
80
81         //
82         LCD_AT(76);
83         sprintf(VARI,"%d",temp);
84         SENDCADE(VARI);
85         //
86
87
88         //temp2=(temp*54); //Para 1,1V
89         //temp3=temp2/50;
90
91         temp2=(temp*67);           //Para 1,080
92         temp3=temp2 >> 6;         // Dividido entre 64
93         media+=temp3;
94     }
95     media=media >>4;             //Realiza el promedio
96
97     LCD_AT(64);
98     sprintf(VARI,"%d",media);
99     SENDCADE(VARI);
100    SENDCADE(" mV");
101
102
103    PORTD = countval++;
104    }
105    //-----
106
107 }
108
109 void PrintAtStr(unsigned char x,char *string)
110 {
111     LCD_AT(x);
112     SENDCADE(string);
113 }
114 void PrintAtNum(unsigned char x,int numero)
115 {
116     LCD_AT(x);
117     sprintf(VARI,"%d",numero);
118     SENDCADE(VARI);
119 }
120 void LCD_INI(void)
121 {
122     delay_ms(250);
123     Pon4bits(0x03);
124     Clear_Datos;
125     ENABLE();
126     Set_Datos;
127     delay_ms(30);
128     Pon4bits(0x03);
129     Clear_Datos;
130     ENABLE();
131     Set_Datos;
132     delay_ms(30);
133     Pon4bits(0x03);
134     Clear_Datos;
135     ENABLE();
136     Set_Datos;
137     delay_ms(30);
138
139     Pon4bits(0x02); // modo 4 bits
140     Clear_Datos;
141     ENABLE();
142     Set_Datos;
143
144     SENDI(0x2c); //modo 4 bits, dos lineas

```

```

145         SENDI(0x0f); // cursor con parpadeo
146         SENDI(0x04);
147     }
148 }
149
150 void Pon4bits(char x)
151 {
152     char z;
153     z=PORTB & 0xf0;
154     x= x & 0x0f;
155     PORTB= x |z;
156 }
157 void LCD_CLS(void)
158 {
159     SENDI(1); //borra pantalla
160 }
161 void LCD_HOME(void)
162 {
163     SENDI(2); //cursor al inicio
164 }
165 void LCD_AT(char x) //0 Comienzo línea1, 64 Comienzo línea2
166 {
167     x=x | 128;
168     SENDI(x);
169 }
170 void SENDCADE(char *x)
171 {
172     char z;
173     for(z=0; z<33 && *x!=0; z++, x++)
174         {SENDCHAR(*x);}
175 }
176 void SENDCHAR(char x)
177 {
178     char z;
179     z=x >> 4;
180     Pon4bits(z); //4 bytes de mas peso
181     ENABLE();
182     Pon4bits(x); //4 bytes de menos peso
183     ENABLE();
184 }
185 //Manda datos de control
186 void SENDI(char x)
187 {
188     Clear_Datos;
189     SENDCHAR(x);
190     delay_10us(250); //Tiempos para reformar
191     Set_Datos;
192 }
193 void ENABLE(void)
194 {
195     Set_Enable;
196     delay_10us(250); //Tiempos para reformar
197     Clear_Enable;
198 }
199
200 // 4 ciclos*delay*time_10us+5*time_10us
201 void delay_10us(unsigned char time_10us)
202 {
203     unsigned short delay_count = F_CPU / 400000; //para 1Mhz->2
204
205     unsigned short cnt;
206     asm volatile ("\n"
207         "L_d11%=: \n\t"
208         "mov %A0, %A2 \n\t"
209         "mov %B0, %B2 \n\t"
210         "L_d12%=: \n\t"
211         "sbw %A0, 1 \n\t"
212         "brne L_d12%=: \n\t"
213         "dec %1 \n\t"
214         "brne L_d11%=: \n\t"
215         : "=&w" (cnt)
216         : "r"(time_10us), "r"((unsigned short) (delay_count))

```

```
217         );
218     }
219
220     // 4 ciclos*delay*time_ms+5*time_ms
221     void delay_ms(unsigned char time_ms)
222     {
223         unsigned short delay_count = F_CPU / 4000; //para 1Mhz->250
224
225         unsigned short cnt;
226         asm volatile ( "\n"
227             "L_d11%=: \n\t"
228             "mov %A0, %A2 \n\t"
229             "mov %B0, %B2 \n\t"
230             "L_d12%=: \n\t"
231             "sbw %A0, 1 \n\t"
232             "brne L_d12%=: \n\t"
233             "dec %1 \n\t"
234             "brne L_d11%=: \n\t"
235             : "=&w" (cnt)
236             : "r"(time_ms), "r"((unsigned short) (delay_count))
237             );
238     }
239
```