```c
// miAT8
// Reloj
// SETA pruebas xsetaseta@gmail.com

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




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

char VARI[22];  //Modificar para numeros mayores de 16

void dalarma(void); //detecta alarma
void increho(void);
void incremi(void);
void increse(void);
void clearhora(void);
void printhora(char x);
void printalon(char x);
void printaloff(char x);



char ho,mi,se;
char hoon,mion,hoof,miof,al,alarma;
char sele;
char cambio;


//ISRs. Rutinas de atención a las interrupciones.
ISR(TIMER2_OVF_vect) // Comentarlo para no utilizar las interupciones
{
    increse();
     cambio=1;
}
//----------------------


int main(void)
{
 uint8_t countval= 0;


    PORTC=0X00;
    PORTB=0X00;
    PORTD=0X00;
    DDRD=0b00110000;    //puerto D 0011-0000 0=entrada 1=salida
    DDRC=0X00;  //puerto C como entrada
    DDRB=0xff; // puerto B como salida

    LCD_INI();
    LCD_CLS();
```



```c
    LCD_HOME();


    //----------
    TCCR2B=0x05; // divide entre 128
    ASSR= (1<<AS2); //Utiliza cristal
    //__________

    PrintAtStr(10,"TIME");
    TIFR2=1;

    alarma=0;
    cambio=1;

    TIMSK2=(1<<TOIE2);
    sei();  //Activa las interupciones
            // Comentarlo para no utilizar las interupciones


    while(1)
    {

/*      if(TIFR2 & 1) //Descomentarlo para no utilizar las interupciones
            {
             TIFR2=1;
             countval++;
             increse();
             if(al==1)dalarma();
             cambio=1;
            }
*/

        if(cambio==1)
            {
             if(al==1)dalarma();
             countval++;
             printhora(0);
             LCD_AT(90);
             if(countval & 1) PORTD|=_BV(5); else PORTD&=~_BV(5);
             cambio=0;
            }

        if((PIND & 128) || (PIND & 64) )
            {
             if(PIND & 128)
                { sele++;PrintAtStr(70,"           ");}
                else
                {
                 if(sele==0)
                    {
                     if(alarma==0)alarma=1; else alarma=0;
                     if(alarma==1) PORTD|=_BV(4); else PORTD&=~_BV(4);
                    }
                }

             switch(sele)
                {
                 case 1:
                    if(PIND & 64) increho();
                    PrintAtStr(64,"HORA + ");
                    printhora(0);
                 break;
                 case 2:
                    if(PIND & 64) {incremi(); se=0;}
                    PrintAtStr(64,"MIN. + ");
                    printhora(0);
                 break;
                 case 3:
                    if(PIND & 64) {++hoon; if(hoon>23)hoon=0;}
                    PrintAtStr(64,"AL.ON H");
                    printalon(75);
                 break;
                 case 4:
                    if(PIND & 64) {++mion; if(mion>59)mion=0;}
                    PrintAtStr(64,"AL.ON M");
```

```
151                         printalon(75);
152                     break;
153                     case 5:
154                         if(PIND & 64) {++hoof; if(hoof>23)hoof=0;}
155                         PrintAtStr(64,"AL.OFF H");
156                         printaloff(75);
157                     break;
158                     case 6:
159                         if(PIND & 64) {++miof; if(miof>59)miof=0;}
160                         PrintAtStr(64,"AL.OFF M");
161                         printaloff(75);
162                     break;
163                     case 7:
164                         if(PIND & 64){ ++al; if(al>1)al=0;}
165                         if(al==0)
166                             {PrintAtStr(64,"   ALARMA OFF   ");}
167                             else
168                             {PrintAtStr(64,"   ALARMA  ON   ");}
169                     break;
170                     default:
171                         sele=0;
172                         LCD_CLS();
173                         PrintAtStr(10,"TIME");
174                         printhora(0);
175                         if(al==1 && sele==0){printalon(64);SENDCADE("ON-");
    printaloff(72);SENDCADE("OFF");}
176 
177                   }
178                delay_ms(20);
179               }
180     }
181 }
182 
183 
184 void dalarma(void) //Detecta alarma
185 {
186     if(hoon==ho && mion==mi)alarma=1;
187     if(hoof==ho && miof==mi)alarma=0;
188     if(alarma==1) PORTD|=_BV(4); else PORTD&=~_BV(4);
189 
190 }
191 void printhora(char x)
192 {
193     sprintf(VARI,"%.2d:%.2d:%.2d",ho,mi,se);
194     PrintAtStr(x,VARI);
195 }
196 void printalon(char x)
197 {
198     sprintf(VARI,"%.2d:%.2d",hoon,mion);
199     PrintAtStr(x,VARI);
200 }
201 void printaloff(char x)
202 {
203     sprintf(VARI,"%.2d:%.2d",hoof,miof);
204     PrintAtStr(x,VARI);
205 }
206 void increse(void)
207 {
208     se++;
209     if(se>59){ se=0; incremi();}
210 }
211 
212 void incremi(void)
213 {
214     mi++;
215     if(mi>59){ mi=0; increho();}
216 }
217 
218 void increho(void)
219 {
220     ho++;
221     if(ho>23) ho=0;
222 }
223 void clearreloj(void)
224 {
```



```
225     ho=0;mi=0;se=0;
226 }
227 
228 void PrintAtStr(unsigned char x,char *string)
229 {
230     LCD_AT(x);
231     SENDCADE(string);
232 }
233 void PrintAtNum(unsigned char x,int numero)
234 {
235     LCD_AT(x);
236     sprintf(VARI,"%d",numero);
237     SENDCADE(VARI);
238 }
239 void LCD_INI(void)
240 {
241     delay_ms(250);
242     Pon4bits(0x03);
243     Clear_Datos;
244     ENABLE();
245     Set_Datos;
246     delay_ms(30);
247     Pon4bits(0x03);
248     Clear_Datos;
249     ENABLE();
250     Set_Datos;
251     delay_ms(30);
252     Pon4bits(0x03);
253     Clear_Datos;
254     ENABLE();
255     Set_Datos;
256     delay_ms(30);
257 
258     Pon4bits(0x02); // modo 4 bits
259     Clear_Datos;
260     ENABLE();
261     Set_Datos;
262 
263     SENDI(0x2c); //modo 4 bits, dos lineas
264     SENDI(0x0f); // cursor con parpadeo
265     SENDI(0x04);
266 
267 }
268 
269 void Pon4bits(char x)
270 {
271  char z;
272  z=PORTB & 0xf0;
273  x= x & 0x0f;
274  PORTB= x |z;
275 }
276 void LCD_CLS(void)
277 {
278     SENDI(1); //borra pantalla
279 }
280 void LCD_HOME(void)
281 {
282     SENDI(2); //cursor al inicio
283 }
284 void LCD_AT(char x) //0 Comienzo línea1, 64 Comienzo línea2
285 {
286     x=x | 128;
287     SENDI(x);
288 }
289 void SENDCADE(char *x)
290 {
291  char z;
292     for(z=0;z<33 && *x!=0;z++,x++)
293         {SENDCHAR(*x);}
294 }
295 void SENDCHAR(char x)
296 {
297  char z;
298     z=x >> 4;
299     Pon4bits(z); //4 bytes de mas peso
```

```
300         ENABLE();
301         Pon4bits(x); //4 bytes de menos peso
302         ENABLE();
303     }
304     //Manda datos de control
305     void SENDI(char x)
306     {
307         Clear_Datos;
308         SENDCHAR(x);
309         delay_10us(250); //Tiempos para reformar
310         Set_Datos;
311     }
312     void ENABLE(void)
313     {
314         Set_Enable;
315         delay_10us(250); //Tiempos para reformar
316         Clear_Enable;
317     }
318
319     // 4 ciclos*delay*time_10us+5*time_10us
320     void delay_10us(unsigned char time_10us)
321     {
322         unsigned short delay_count = F_CPU / 400000; //para 1Mhz->2
323
324         unsigned short cnt;
325         asm volatile ("\n"
326                       "L_dl1%=:\n\t"
327                       "mov %A0, %A2\n\t"
328                       "mov %B0, %B2\n"
329                       "L_dl2%=:\n\t"
330                       "sbiw %A0, 1\n\t"
331                       "brne L_dl2%=\n\t"
332                       "dec %1\n\t"
333                       "brne L_dl1%=\n\t"
334                       :"=&w" (cnt)
335                       :"r"(time_10us), "r"((unsigned short) (delay_count))
336         );
337     }
338
339     // 4 ciclos*delay*time_ms+5*time_ms
340     void delay_ms(unsigned char time_ms)
341     {
342         unsigned short delay_count = F_CPU / 4000; //para 1Mhz->250
343
344         unsigned short cnt;
345         asm volatile ("\n"
346                       "L_dl1%=:\n\t"
347                       "mov %A0, %A2\n\t"
348                       "mov %B0, %B2\n"
349                       "L_dl2%=:\n\t"
350                       "sbiw %A0, 1\n\t"
351                       "brne L_dl2%=\n\t"
352                       "dec %1\n\t"
353                       "brne L_dl1%=\n\t"
354                       :"=&w" (cnt)
355                       :"r"(time_ms), "r"((unsigned short) (delay_count))
356         );
357     }
358
359
```