

Código reloj con alarma semana 11 2025-2

```

1 //Reloj con alarma
2 //Hecho por Kalun Lau
3 //UPC 07 de Noviembre del 2025
4
5 #include <xc.h>
6 #include "cabecera.h"
7 #include "LCD.h"
8 #define _XTAL_FREQ 32000000UL
9 #define RELOJ 0
10 #define SET_CLK 1
11 #define SET_AL 2
12 #define APAGADA 3
13 #define ENCENDIDA 4
14
15 unsigned char horas=10,minutos=10,segundos=10,centesimas=10; //hora inicial 10:10'10"10
16 unsigned char al_hora=17,al_mins=00; //alarma ajustada a 17:00 inicialmente
17 unsigned char centena,decena,unidad;
18 unsigned char MODO = RELOJ;
19 unsigned char ALARM = APAGADA;
20
21 void configuro(void){
22     //configuracion del oscilador
23     OSCCON = 0x00; //HFINTOSC, posts 1:1
24     OSCFREQ = 0x00; //HFINTOSC a 32Mhz
25     OSCEN = 0x00; //HFINTOSC enabled
26     //configuracion de E/S
27     TRISA = 0xFF; //RB2 RB1 RB0 como entradas
28     ANSELB = 0x0F; //RB2 RB1 RB0 como digitales
29     WPUB = 0x07; //RB2 RB1 RB0 pullup enabled
30     TRISEBITS.TRISEB = 0; //RE0 estara el buzzer
31     ANSELEBITS.ANSELEB = 0; //RE0 estara el buzzer
32     //configuracion del Timer1
33     T1CLK = 0x01; //clk source fosc/4
34     T1CON = 0x33; //TMR1 ON, pres 1:8
35     //configuracion del CCP1
36     CCP1CON = 0x01; //compare mode, clear TMR1
37     CCP1M = 0x27; //valor de referencia 10000
38     CCP1IL = 0x00; //valor de referencia 10000
39     //configuracion de las interrupciones
40     INTCOMMbits.INTFIEDG = 0; //falling edge en INT0
41     INTCOMMbits.INTFIEDG = 0; //falling edge en INT1
42     INTCOMMbits.INTFIEDG = 0; //falling edge en INT2
43     PIRbits.INT0IF = 1; //INT0 enabled
44     PIRbits.INT1IF = 1; //INT1 enabled
45     PIRbits.INT2IF = 1; //INT2 enabled
46
47     PIEbits.CCP1IE = 1; //CCP1 enabled
48     PIRbits.INT0IF = 0; //Flag INT0 bajada
49     PIRbits.INT1IF = 0; //Flag INT1 bajada
50     PIRbits.CCP1IF = 0; //Flag CCP1 bajada
51     INTCOMMbits.GIE = 1; //Global ints enabled
52     //condiciones iniciales
53     LATEbits.LATE0 = 0; //buzzer apagado
54 }
55
56 void lcd_init(void){
57     TRISO = 0x00;
58     ANSEL = 0x00;
59     LCD_CONF1G();
60     __delay_ms(21);
61     BORRAR_LCD();
62     CURSOR_HOME();
63     CURSOR_ONOFF(OFF);
64 }
65
66 void convierte(unsigned char numero){
67     centena = numero / 100;
68     decena = (numero % 100) / 10;
69     unidad = numero % 10;
70 }
71
72 void DISP_ALARMA(void){
73     POS_CURSOR(1,14);
74     ESCRIBE_MENSAJE("AL",2);
75     POS_CURSOR(2,14);
76     if(ALARM == APAGADA){
77         ESCRIBE_MENSAJE("OF",2);
78     }
79     else if(ALARM == ENCENDIDA){
80         ESCRIBE_MENSAJE("ON",2);
81     }
82 }
83
84 void CTL_AL(void){
85     if(al_hora == horas && al_mins == minutos && ALARM == ENCENDIDA){
86         LATEbits.LATE0 = 1; //suena el buzzer
87     }
88     else{
89
90
91
92
93     }
94 }
95
96 void main(void){
97     configuro();
98     lcd_init();
99     while(1){
100         switch(MODO){
101             case RELOJ:
102                 POS_CURSOR(1,2);
103                 ESCRIBE_MENSAJE("Relojon UPC",11);
104                 POS_CURSOR(2,2);
105                 convierte(horas);
106                 ENVIA_CHAR(decena+0x30);
107                 ENVIA_CHAR(unidad+0x30);
108                 ENVIA_CHAR(':');
109                 convierte(minutos);
110                 ENVIA_CHAR(decena+0x30);
111                 ENVIA_CHAR(unidad+0x30);
112                 ENVIA_CHAR(':');
113                 convierte(segundos);
114                 ENVIA_CHAR(decena+0x30);
115                 ENVIA_CHAR(unidad+0x30);
116                 DISP_ALARMA();
117                 CTL_AL();
118                 break;
119             case SET_CLK:
120                 POS_CURSOR(1,2);
121                 ESCRIBE_MENSAJE("Set Clock ",11);
122                 POS_CURSOR(2,2);
123                 convierte(horas);
124                 ENVIA_CHAR(decena+0x30);
125                 ENVIA_CHAR(unidad+0x30);
126                 ENVIA_CHAR(':');
127                 convierte(minutos);
128                 ENVIA_CHAR(decena+0x30);
129                 ENVIA_CHAR(unidad+0x30);
130                 ENVIA_CHAR(':');
131                 ESCRIBE_MENSAJE(" ",6);
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249

```

1

Código reloj con alarma semana 11 2025-2

```

132 void DISP_ALARMA(){
133     break;
134     case SET_AL:
135         POS_CURSOR(1,2);
136         ESCRIBE_MENSAJE("Set Alarm ",11);
137         POS_CURSOR(2,2);
138         convierte(al_hora);
139         ENVIA_CHAR(decena+0x30);
140         ENVIA_CHAR(unidad+0x30);
141         ENVIA_CHAR(':');
142         convierte(al_mins);
143         ENVIA_CHAR(decena+0x30);
144         ENVIA_CHAR(unidad+0x30);
145         ESCRIBE_MENSAJE(" ",6);
146         DISP_ALARMA();
147         break;
148     }
149 }
150 }
151
152 void __interrupt__(irq(CCPI)) CCP1_ISR(void){
153     PIRbits.CCP1IF = 0;
154     if(centesimas == 99){
155         centesimas = 0;
156         if(segundos == 59){
157             segundos = 0;
158             if(minutos == 59){
159                 minutos = 0;
160                 if(horas == 23){
161                     horas = 0;
162                 }
163                 else{
164                     horas++;
165                 }
166             }
167             else{
168                 minutos++;
169             }
170         }
171         else{
172             segundos++;
173         }
174     }
175 }
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249

```

2