

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=0; //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     OSCFRC = 0x00; //HFINTOSC a 32MHz
25     OSCEN = 0x00; //HFINTOSC enabled
26     //Configuracion de E/F
27     TRISB = 0xFF; //RB2 RB1 RB0 como entradas
28     ANSELB = 0x00; //RB2 RB1 RB0 como digitales
29     WPUB = 0x07; //RB2 RB1 RB0 pullup enabled
30     TRISEbits.TRISE0 = 0;
31     ANSELBbits.ANSEL0 = 0; //RE0 estara el buzzer
32     //Configuracion del Timer1
33     T1CON = 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     CCP1RH = 0x27;
38     CCP1IL = 0x00; //valor de referencia 10000
39     //Configuracion de las interrupciones
40     INTCON0bits.INT0DG = 0; //falling edge en INT0
41     INTCON0bits.INT1DG = 0; //falling edge en INT1
42     INTCON0bits.INT2DG = 0; //falling edge en INT2
43     PIR1bits.INT0IE = 1; //INT0 enabled
44     PIR1bits.INT1IE = 1; //INT1 enabled
45     PIR10bits.INT2IE = 1; //INT2 enabled
46     PIR1bits.CCP1IE = 1; //CCP1 enabled
47     PIR1bits.INT0IF = 0; //Flag INT0 bajada
48     PIR1bits.INT1IF = 0; //Flag INT1 bajada
49     PIR10bits.INT2IF = 0; //Flag CCP1 bajada
50     INTCON0bits.GIE = 1; //global ints enabled
51     //condiciones iniciales
52     LATEdbits.LATE0 = 0; //buzzer apagado
53 }
54
55 void lcd_init(void){
56     TRISD = 0x00;
57     ANSEL0 = 0x00;
58     LCD_CONFIG();
59     __delay_ms(21);
60     BORRAR_LCD();
61     CURSOR_HOME();
62     CURSOR_ONOFF(OFF);
63 }
64
65 void convierte(unsigned char numero){
66     centena = numero / 100;
67     decena = (numero % 100) / 10;
68     unidad = numero % 10;
69 }
70
71 void DISP_ALARMA(void){
72     POS_CURSOR(1,14);
73     ESCRIBE_MENSAJE("AL",2);
74     POS_CURSOR(2,14);
75     if(ALARM == APAGADA){
76         ESCRIBE_MENSAJE("OF",2);
77     } else if(ALARM == ENCENDIDA){
78         ESCRIBE_MENSAJE("ON",2);
79     }
80 }
81
82 void CTL_AL(void){
83     if(al_hora == horas && al_mins == minutos && ALARM == ENCENDIDA){
84         LATEdbits.LATE0 = 1; //suena el buzzer
85     }
86 }
87
88 void _interrupt(irq(IRQ_INT2)) INT2_ISR(void){
89     LATEdbits.LATE0 = 0; //buzzer apagado
90 }
91
92 void main(void) {
93     configuro();
94     lcd_init();
95     while(1){
96         switch(MODO){
97             case RELOJ:
98                 POS_CURSOR(1,2);
99                 ESCRIBE_MENSAJE("Relojon UPC",11);
100                POS_CURSOR(2,2);
101                convierte(horas);
102                ENVIA_CHAR(decena+0x30);
103                ENVIA_CHAR(unidad+0x30);
104                ENVIA_CHAR(':');
105                convierte(minutos);
106                ENVIA_CHAR(decena+0x30);
107                ENVIA_CHAR(unidad+0x30);
108                ENVIA_CHAR('.');
109                convierte(segundos);
110                ENVIA_CHAR(decena+0x30);
111                ENVIA_CHAR(unidad+0x30);
112                ENVIA_CHAR(':');
113                convierte(centesimas);
114                ENVIA_CHAR(decena+0x30);
115                ENVIA_CHAR(unidad+0x30);
116                ENVIA_CHAR('.');
117                DISP_ALARMA();
118                CTL_AL();
119                break;
120            case SET_CLK:
121                POS_CURSOR(1,2);
122                ESCRIBE_MENSAJE("Set Clock ",11);
123                POS_CURSOR(2,2);
124                convierte(horas);
125                ENVIA_CHAR(decena+0x30);
126                ENVIA_CHAR(unidad+0x30);
127                ENVIA_CHAR('.');
128                convierte(minutos);
129                ENVIA_CHAR(decena+0x30);
130                ENVIA_CHAR(unidad+0x30);
131                ESCRIBE_MENSAJE(" ",8);
132
133         }
134     }
135 }
136
137 void _interrupt(irq(IRQ_CCPI)) CCP1_ISR(void){
138     PIR1bits.CCP1IF = 0;
139     if(centesimas == 99){
140         centesimas = 0;
141         if(segundos == 59){
142             segundos = 0;
143             if(minutos == 59){
144                 minutos = 0;
145                 if(horas == 23){
146                     horas = 0;
147                 } else{
148                     horas++;
149                 }
150             } else{
151                 minutos++;
152             }
153         } else{
154             segundos++;
155         }
156     }
157 }
158
159 void _interrupt(irq(IRQ_INT1)) INT1_ISR(void){
160     PIR1bits.INT1IF = 0;
161     switch(MODO){
162         case RELOJ:
163             ALARM = ENCENDIDA;
164             break;
165         case SET_CLK:
166             if(al_hora == 23){
167                 al_hora = 0;
168             } else{
169                 al_mins++;
170             }
171         case SET_AL:
172             if(al_hora == 23){
173                 al_hora++;
174             }
175     }
176     centesimas++;
177 }
178
179 //cambio de modo
180 void _interrupt(irq(IRQ_INTE0)) INTO_ISR(void){
181     PIR1bits.INT0IF = 0;
182     switch(MODO){
183         case RELOJ:
184             MODO = SET_CLK;
185             break;
186         case SET_CLK:
187             MODO = SET_AL;
188             break;
189         case SET_AL:
190             MODO = RELOJ;
191             break;
192     }
193 }
194
195 void _interrupt(irq(IRQ_INTE1)) INT1_ISR(void){
196     PIR1bits.INT1IF = 0;
197     switch(MODO){
198         case RELOJ:
199             if(horas == 23){
200                 horas = 0;
201             } else{
202                 horas++;
203             }
204         case SET_CLK:
205             if(horas == 23){
206                 horas = 0;
207             } else{
208                 horas++;
209             }
210         case SET_AL:
211             if(al_hora == 23){
212                 al_hora = 0;
213             } else{
214                 al_hora++;
215             }
216     }
217     break;
218 }
219
220 void _interrupt(irq(IRQ_INTE2)) INT2_ISR(void){
221     PIR10bits.INT2IF = 0;
222     switch(MODO){
223         case RELOJ:
224             ALARM = APAGADA;
225             break;
226         case SET_CLK:
227             if(minutos == 59){
228                 minutos = 0;
229             } else{
230                 minutos++;
231             }
232         case SET_AL:
233             if(al_mins == 59){
234                 al_mins = 0;
235             } else{
236                 al_mins++;
237             }
238         break;
239     }
240 }
241
242 void _interrupt(irq(default)) DEFAULT_ISR(void){
243     // Unhandled interrupts go here
244 }
245
246
247
248
249 }
```

1

Código reloj con alarma semana 11 2025-2

```

132     DISP_ALARMA();
133     break;
134 }
135
136 case SET_AL:
137     POS_CURSOR(1,2);
138     ESCRIBE_MENSAJE("Set Alarm ",11);
139     POS_CURSOR(2,2);
140     convierte(al_hora);
141     ENVIA_CHAR(decena+0x30);
142     ENVIA_CHAR(unidad+0x30);
143     ENVIA_CHAR('.');
144     convierte(al_mins);
145     ENVIA_CHAR(decena+0x30);
146     ENVIA_CHAR(unidad+0x30);
147     ESCRIBE_MENSAJE(" ",8);
148     DISP_ALARMA();
149     break;
150 }
151
152 void _interrupt(irq(IRQ_CCPI)) CCP1_ISR(void){
153     PIR1bits.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                 } else{
163                     horas++;
164                 }
165             }
166         } else{
167             minutos++;
168         }
169     }
170     if(al_hora == 23){
171         al_hora = 0;
172     } else{
173         al_mins++;
174     }
175 }
176
177 void _interrupt(irq(IRQ_INTE0)) INTO_ISR(void){
178     PIR1bits.INT0IF = 0;
179     switch(MODO){
180         case RELOJ:
181             MODO = SET_CLK;
182             break;
183         case SET_CLK:
184             MODO = SET_AL;
185             break;
186         case SET_AL:
187             MODO = RELOJ;
188             break;
189     }
190 }
191
192 void _interrupt(irq(IRQ_INTE1)) INT1_ISR(void){
193     PIR1bits.INT1IF = 0;
194     switch(MODO){
195         case RELOJ:
196             if(horas == 23){
197                 horas = 0;
198             } else{
199                 horas++;
200             }
201         case SET_CLK:
202             if(horas == 23){
203                 horas = 0;
204             } else{
205                 horas++;
206             }
207         case SET_AL:
208             if(al_hora == 23){
209                 al_hora = 0;
210             } else{
211                 al_hora++;
212             }
213     }
214     break;
215 }
216
217 void _interrupt(irq(IRQ_INTE2)) INT2_ISR(void){
218     PIR10bits.INT2IF = 0;
219     switch(MODO){
220         case RELOJ:
221             ALARM = APAGADA;
222             break;
223         case SET_CLK:
224             if(minutos == 59){
225                 minutos = 0;
226             } else{
227                 minutos++;
228             }
229         case SET_AL:
230             if(al_mins == 59){
231                 al_mins = 0;
232             } else{
233                 al_mins++;
234             }
235         break;
236     }
237 }
238
239 void _interrupt(irq(default)) DEFAULT_ISR(void){
240     // Unhandled interrupts go here
241 }
242
243
244
245
246
247
248
249 }
```

2

1