

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 configura(void){
22     //configuración del oscilador
23     OSCCON = 0x08; //INTOSC, posts 1:1
24     OSCFRC = 0x06; //INTOSC a 32MHz
25     OSCEN = 0x08; //INTOSC enabled
26     //configuración de I/O
27     TRISB = 0xF; //RB2 RB3 RB0 como entradas
28     ANSELB = 0x08; //RB2 RB3 RB0 como digitales
29     WPUB = 0x07; //RB2 RB3 RB0 pullup enabled
30     TRISEbits.TRISE0 = 0; //RE0 entra el buzzer
31     ANSELEbits.ANSEL0 = 0; //RE0 entra el buzzer
32     //configuración del Timer1
33     T1CLK = 0x01; //Clk source fosc/4
34     T1CON = 0x3; //TMR1 On, pres 1:8
35     //configuración del CCP1
36     CCP1CN = 0x08; //Compare mode, clear TMR1
37     CCP1IN = 0x27;
38     CCP1IL = 0x10; //valor de referencia 10000
39     //configuración de las interrupciones
40     INTCON0bits.INT0EDG = 0; //Falling edge en INT0
41     INTCON0bits.INT1EDG = 0; //Falling edge en INT1
42     INTCON0bits.INT2EDG = 0; //Falling edge en INT2
43     PIEbits.INT0IE = 1; //INT0 enabled
44     PIEbits.INT1IE = 1; //INT1 enabled
45     PIEbits.INT2IE = 1; //INT2 enabled
46     PIEbits.CCP1IF = 1; //CCP1 enabled
47     PIRbits.INT0IF = 0; //Flag INT0 bajada
48     PIRbits.INT1IF = 0; //Flag INT1 bajada
49     PIR0bits.INT2IF = 0; //Flag INT2 bajada
50     PIRbits.CCP1IF = 0; //Flag CCP1 bajada
51     INTCONbits.GIE = 1; //global ints enabled
52     //condiciones iniciales
53     LATEBits.LATE0 = 0; //buzzer apagado
54 }
55
56 void lcd_init(void){
57     TRIS = 0x00;
58     ANSEL = 0x00;
59     LCD_CONFIG();
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 CIL_AL(void){
85     if(al_hora == horas && al_mins == minutos && ALARM == ENCENDIDA){
86         LATEBits.LATE0 = 1; //suena el buzzer
87     }
88 }
89
90 LAITbits.LATE0 = 0; //buzzer apagado
91 }
92
93 void main(void) {
94     configura();
95     lcd_init();
96     while(1){
97         switch(MODO){
98             case RELOJ:
99                 POS_CURSOR(1,2);
100                ESCRIBE_MENSAJE("Relojon UPC",11);
101                POS_CURSOR(2,2);
102                convierte(horas);
103                ENVIA_CHAR(decena+0x30);
104                ENVIA_CHAR(unidad+0x30);
105                ENVIA_CHAR(':');
106                convierte(minutos);
107                ENVIA_CHAR(decena+0x30);
108                ENVIA_CHAR(unidad+0x30);
109                ENVIA_CHAR(',');
110                convierte(segundos);
111                ENVIA_CHAR(decena+0x30);
112                ENVIA_CHAR(unidad+0x30);
113                ENVIA_CHAR('.');
114                convierte(centesimas);
115                ENVIA_CHAR(decena+0x30);
116                ENVIA_CHAR(unidad+0x30);
117                DISP_ALARMA();
118                CIL_AL();
119                break;
120         case 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(" ",6);
132         }
133     }
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(" ",6);
148     DISP_ALARMA();
149     break;
150 }
151
152 void __interrupt(irq(IRO_CCP1)) CCP1_ISR(void){
153     PIR3bits.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     }
172     else{
173         segundos++;
174     }
175     else{
176         centesimas++;
177     }
178 }
179
180 //cambio de modo
181 void __interrupt(irq(IRO_INT0)) INT0_ISR(void){
182     PIR0bits.INT0IF = 0;
183     switch(MODO){
184         case RELOJ:
185             MODO = SET_CLK;
186             break;
187         case SET_CLK:
188             MODO = SET_AL;
189             break;
190         case SET_AL:
191             MODO = RELOJ;
192             break;
193     }
194 }
195
196 void __interrupt(irq(IRO_INT1)) INT1_ISR(void){
197     PIR0bits.INT1IF = 0;
198     switch(MODO){
199         case RELOJ:
200             ALARM = ENCENDIDA;
201             break;
202         case SET_CLK:
203             if(horas == 23){
204                 horas = 0;
205             }
206             else{
207                 horas++;
208             }
209             break;
210         case SET_AL:
211             if(al_hora == 23){
212                 al_hora = 0;
213             }
214             else{
215                 al_hora++;
216             }
217             break;
218     }
219 }
220
221 void __interrupt(irq(IRO_INT2)) INT2_ISR(void){
222     PIR0bits.INT2IF = 0;
223     switch(MODO){
224         case RELOJ:
225             ALARM = APAGADA;
226             break;
227         case SET_CLK:
228             if(minutos == 59){
229                 minutos = 0;
230             }
231             else{
232                 minutos++;
233             }
234             break;
235         case SET_AL:
236             if(al_mins == 59){
237                 al_mins = 0;
238             }
239             else{
240                 al_mins++;
241             }
242             break;
243     }
244 }
245
246 void __interrupt(irq(default)) DEFAULT_ISR(void){
247     {
248     // Unhandled interrupts go here
249     }
250 }

```

1

Código reloj con alarma semana 11 2025-2

```

132     DISP_ALARMA();
133     break;
134 }
135 case SET_AL:
136     POS_CURSOR(1,2);
137     ESCRIBE_MENSAJE("Set Alarm ",11);
138     POS_CURSOR(2,2);
139     convierte(al_hora);
140     ENVIA_CHAR(decena+0x30);
141     ENVIA_CHAR(unidad+0x30);
142     ESCRIBE_MENSAJE(" ",6);
143     DISP_ALARMA();
144     break;
145 }
146
147 void __interrupt(irq(IRO_CCP1)) CCP1_ISR(void){
148     PIR3bits.CCP1IF = 0;
149     if(centesimas == 99){
150         centesimas = 0;
151         if(segundos == 59){
152             segundos = 0;
153             if(minutos == 59){
154                 minutos = 0;
155                 if(horas == 23){
156                     horas = 0;
157                 }
158                 else{
159                     horas++;
160                 }
161             }
162             else{
163                 minutos++;
164             }
165         }
166     }
167     else{
168         segundos++;
169     }
170 }
171
172 void __interrupt(irq(IRO_INT0)) INT0_ISR(void){
173     PIR0bits.INT0IF = 0;
174     switch(MODO){
175         case RELOJ:
176             MODO = SET_CLK;
177             break;
178         case SET_CLK:
179             MODO = SET_AL;
180             break;
181         case SET_AL:
182             if(al_hora == 23){
183                 al_hora = 0;
184             }
185             else{
186                 al_hora++;
187             }
188             break;
189     }
190 }
191
192 void __interrupt(irq(IRO_INT1)) INT1_ISR(void){
193     PIR0bits.INT1IF = 0;
194     switch(MODO){
195         case RELOJ:
196             ALARM = ENCENDIDA;
197             break;
198         case SET_CLK:
199             if(horas == 23){
200                 horas = 0;
201             }
202             else{
203                 horas++;
204             }
205             break;
206         case SET_AL:
207             if(al_hora == 23){
208                 al_hora = 0;
209             }
210             else{
211                 al_hora++;
212             }
213             break;
214     }
215 }
216
217 void __interrupt(irq(IRO_INT2)) INT2_ISR(void){
218     PIR0bits.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             }
227             else{
228                 minutos++;
229             }
230             break;
231         case SET_AL:
232             if(al_mins == 59){
233                 al_mins = 0;
234             }
235             else{
236                 al_mins++;
237             }
238             break;
239     }
240 }
241
242 void __interrupt(irq(default)) DEFAULT_ISR(void){
243     {
244     // Unhandled interrupts go here
245     }
246 }
247
248 void __interrupt(irq(IRO_TMR1)) TMR1_ISR(void){
249     LATEBits.LATE0 = 1;
250     if(ALARM == ON){
251         LATEBits.LATE0 = 0;
252     }
253 }
254
255 void __interrupt(irq(IRO_TMR0)) TMR0_ISR(void){
256     LATEBits.LATE0 = 1;
257     if(ALARM == ON){
258         LATEBits.LATE0 = 0;
259     }
260 }
261
262 void __interrupt(irq(IRO_CCPI)) CCP1_ISR(void){
263     PIR3bits.CCP1IF = 0;
264     if(CCPI == 1){
265         LATEBits.LATE0 = 1;
266     }
267     else{
268         LATEBits.LATE0 = 0;
269     }
270 }
271
272 void __interrupt(irq(IRO_CCPI)) CCP2_ISR(void){
273     PIR3bits.CCP2IF = 0;
274     if(CCPI == 1){
275         LATEBits.LATE0 = 1;
276     }
277     else{
278         LATEBits.LATE0 = 0;
279     }
280 }
281
282 void __interrupt(irq(IRO_CCPI)) CCP3_ISR(void){
283     PIR3bits.CCP3IF = 0;
284     if(CCPI == 1){
285         LATEBits.LATE0 = 1;
286     }
287     else{
288         LATEBits.LATE0 = 0;
289     }
290 }
291
292 void __interrupt(irq(IRO_CCPI)) CCP4_ISR(void){
293     PIR3bits.CCP4IF = 0;
294     if(CCPI == 1){
295         LATEBits.LATE0 = 1;
296     }
297     else{
298         LATEBits.LATE0 = 0;
299     }
300 }
301
302 void __interrupt(irq(IRO_CCPI)) CCP5_ISR(void){
303     PIR3bits.CCP5IF = 0;
304     if(CCPI == 1){
305         LATEBits.LATE0 = 1;
306     }
307     else{
308         LATEBits.LATE0 = 0;
309     }
310 }
311
312 void __interrupt(irq(IRO_CCPI)) CCP6_ISR(void){
313     PIR3bits.CCP6IF = 0;
314     if(CCPI == 1){
315         LATEBits.LATE0 = 1;
316     }
317     else{
318         LATEBits.LATE0 = 0;
319     }
320 }
321
322 void __interrupt(irq(IRO_CCPI)) CCP7_ISR(void){
323     PIR3bits.CCP7IF = 0;
324     if(CCPI == 1){
325         LATEBits.LATE0 = 1;
326     }
327     else{
328         LATEBits.LATE0 = 0;
329     }
330 }
331
332 void __interrupt(irq(IRO_CCPI)) CCP8_ISR(void){
333     PIR3bits.CCP8IF = 0;
334     if(CCPI == 1){
335         LATEBits.LATE0 = 1;
336     }
337     else{
338         LATEBits.LATE0 = 0;
339     }
340 }
341
342 void __interrupt(irq(IRO_CCPI)) CCP9_ISR(void){
343     PIR3bits.CCP9IF = 0;
344     if(CCPI == 1){
345         LATEBits.LATE0 = 1;
346     }
347     else{
348         LATEBits.LATE0 = 0;
349     }
350 }
351
352 void __interrupt(irq(IRO_CCPI)) CCP10_ISR(void){
353     PIR3bits.CCP10IF = 0;
354     if(CCPI == 1){
355         LATEBits.LATE0 = 1;
356     }
357     else{
358         LATEBits.LATE0 = 0;
359     }
360 }
361
362 void __interrupt(irq(IRO_CCPI)) CCP11_ISR(void){
363     PIR3bits.CCP11IF = 0;
364     if(CCPI == 1){
365         LATEBits.LATE0 = 1;
366     }
367     else{
368         LATEBits.LATE0 = 0;
369     }
370 }
371
372 void __interrupt(irq(IRO_CCPI)) CCP12_ISR(void){
373     PIR3bits.CCP12IF = 0;
374     if(CCPI == 1){
375         LATEBits.LATE0 = 1;
376     }
377     else{
378         LATEBits.LATE0 = 0;
379     }
380 }
381
382 void __interrupt(irq(IRO_CCPI)) CCP13_ISR(void){
383     PIR3bits.CCP13IF = 0;
384     if(CCPI == 1){
385         LATEBits.LATE0 = 1;
386     }
387     else{
388         LATEBits.LATE0 = 0;
389     }
390 }
391
392 void __interrupt(irq(IRO_CCPI)) CCP14_ISR(void){
393     PIR3bits.CCP14IF = 0;
394     if(CCPI == 1){
395         LATEBits.LATE0 = 1;
396     }
397     else{
398         LATEBits.LATE0 = 0;
399     }
400 }
401
402 void __interrupt(irq(IRO_CCPI)) CCP15_ISR(void){
403     PIR3bits.CCP15IF = 0;
404     if(CCPI == 1){
405         LATEBits.LATE0 = 1;
406     }
407     else{
408         LATEBits.LATE0 = 0;
409     }
410 }
411
412 void __interrupt(irq(IRO_CCPI)) CCP16_ISR(void){
413     PIR3bits.CCP16IF = 0;
414     if(CCPI == 1){
415         LATEBits.LATE0 = 1;
416     }
417     else{
418         LATEBits.LATE0 = 0;
419     }
420 }
421
422 void __interrupt(irq(IRO_CCPI)) CCP17_ISR(void){
423     PIR3bits.CCP17IF = 0;
424     if(CCPI == 1){
425         LATEBits.LATE0 = 1;
426     }
427     else{
428         LATEBits.LATE0 = 0;
429     }
430 }
431
432 void __interrupt(irq(IRO_CCPI)) CCP18_ISR(void){
433     PIR3bits.CCP18IF = 0;
434     if(CCPI == 1){
435         LATEBits.LATE0 = 1;
436     }
437     else{
438         LATEBits.LATE0 = 0;
439     }
440 }
441
442 void __interrupt(irq(IRO_CCPI)) CCP19_ISR(void){
443     PIR3bits.CCP19IF = 0;
444     if(CCPI == 1){
445         LATEBits.LATE0 = 1;
446     }
447     else{
448         LATEBits.LATE0 = 0;
449     }
450 }
451
452 void __interrupt(irq(IRO_CCPI)) CCP20_ISR(void){
453     PIR3bits.CCP20IF = 0;
454     if(CCPI == 1){
455         LATEBits.LATE0 = 1;
456     }
457     else{
458         LATEBits.LATE0 = 0;
459     }
460 }
461
462 void __interrupt(irq(IRO_CCPI)) CCP21_ISR(void){
463     PIR3bits.CCP21IF = 0;
464     if(CCPI == 1){
465         LATEBits.LATE0 = 1;
466     }
467     else{
468         LATEBits.LATE0 = 0;
469     }
470 }
471
472 void __interrupt(irq(IRO_CCPI)) CCP22_ISR(void){
473     PIR3bits.CCP22IF = 0;
474     if(CCPI == 1){
475         LATEBits.LATE0 = 1;
476     }
477     else{
478         LATEBits.LATE0 = 0;
479     }
480 }
481
482 void __interrupt(irq(IRO_CCPI)) CCP23_ISR(void){
483     PIR3bits.CCP23IF = 0;
484     if(CCPI == 1){
485         LATEBits.LATE0 = 1;
486     }
487     else{
488         LATEBits.LATE0 = 0;
489     }
490 }
491
492 void __interrupt(irq(IRO_CCPI)) CCP24_ISR(void){
493     PIR3bits.CCP24IF = 0;
494     if(CCPI == 1){
495         LATEBits.LATE0 = 1;
496     }
497     else{
498         LATEBits.LATE0 = 0;
499     }
500 }
501
502 void __interrupt(irq(IRO_CCPI)) CCP25_ISR(void){
503     PIR3bits.CCP25IF = 0;
504     if(CCPI == 1){
505         LATEBits.LATE0 = 1;
506     }
507     else{
508         LATEBits.LATE0 = 0;
509     }
510 }
511
512 void __interrupt(irq(IRO_CCPI)) CCP26_ISR(void){
513     PIR3bits.CCP26IF = 0;
514     if(CCPI == 1){
515         LATEBits.LATE0 = 1;
516     }
517     else{
518         LATEBits.LATE0 = 0;
519     }
520 }
521
522 void __interrupt(irq(IRO_CCPI)) CCP27_ISR(void){
523     PIR3bits.CCP27IF = 0;
524     if(CCPI == 1){
525         LATEBits.LATE0 = 1;
526     }
527     else{
528         LATEBits.LATE0 = 0;
529     }
530 }
531
532 void __interrupt(irq(IRO_CCPI)) CCP28_ISR(void){
533     PIR3bits.CCP28IF = 0;
534     if(CCPI == 1){
535         LATEBits.LATE0 = 1;
536     }
537     else{
538         LATEBits.LATE0 = 0;
539     }
540 }
541
542 void __interrupt(irq(IRO_CCPI)) CCP29_ISR(void){
543     PIR3bits.CCP29IF = 0;
544     if(CCPI == 1){
545         LATEBits.LATE0 = 1;
546     }
547     else{
548         LATEBits.LATE0 = 0;
549     }
550 }
551
552 void __interrupt(irq(IRO_CCPI)) CCP30_ISR(void){
553     PIR3bits.CCP30IF = 0;
554     if(CCPI == 1){
555         LATEBits.LATE0 = 1;
556     }
557     else{
558         LATEBits.LATE0 = 0;
559     }
560 }
561
562 void __interrupt(irq(IRO_CCPI)) CCP31_ISR(void){
563     PIR3bits.CCP31IF = 0;
564     if(CCPI == 1){
565         LATEBits.LATE0 = 1;
566     }
567     else{
568         LATEBits.LATE0 = 0;
569     }
570 }
571
572 void __interrupt(irq(IRO_CCPI)) CCP32_ISR(void){
573     PIR3bits.CCP32IF = 0;
574     if(CCPI == 1){
575         LATEBits.LATE0 = 1;
576     }
577     else{
578         LATEBits.LATE0 = 0;
579     }
580 }
581
582 void __interrupt(irq(IRO_CCPI)) CCP33_ISR(void){
583     PIR3bits.CCP33IF = 0;
584     if(CCPI == 1){
585         LATEBits.LATE0 = 1;
586     }
587     else{
588         LATEBits.LATE0 = 0;
589     }
590 }
591
592 void __interrupt(irq(IRO_CCPI)) CCP34_ISR(void){
593     PIR3bits.CCP34IF = 0;
594     if(CCPI == 1){
595         LATEBits.LATE0 = 1;
596     }
597     else{
598         LATEBits.LATE0 = 0;
599     }
600 }
601
602 void __interrupt(irq(IRO_CCPI)) CCP35_ISR(void){
603     PIR3bits.CCP35IF = 0;
604     if(CCPI == 1){
605         LATEBits.LATE0 = 1;
606     }
607     else{
608         LATEBits.LATE0 = 0;
609     }
610 }
611
612 void __interrupt(irq(IRO_CCPI)) CCP36_ISR(void){
613     PIR3bits.CCP36IF = 0;
614     if(CCPI == 1){
615         LATEBits.LATE0 = 1;
616     }
617     else{
618         LATEBits.LATE0 = 0;
619     }
620 }
621
622 void __interrupt(irq(IRO_CCPI)) CCP37_ISR(void){
623     PIR3bits.CCP37IF = 0;
624     if(CCPI == 1){
625         LATEBits.LATE0 = 1;
626     }
627     else{
628         LATEBits.LATE0 = 0;
629     }
630 }
631
632 void __interrupt(irq(IRO_CCPI)) CCP38_ISR(void){
633     PIR3bits.CCP38IF = 0;
634     if(CCPI == 1){
635         LATEBits.LATE0 = 1;
636     }
637     else{
638         LATEBits.LATE0 = 0;
639     }
640 }
641
642 void __interrupt(irq(IRO_CCPI)) CCP39_ISR(void){
643     PIR3bits.CCP39IF = 0;
644     if(CCPI == 1){
645         LATEBits.LATE0 = 1;
646     }
647     else{
648         LATEBits.LATE0 = 0;
649     }
650 }
651
652 void __interrupt(irq(IRO_CCPI)) CCP40_ISR(void){
653     PIR3bits.CCP40IF = 0;
654     if(CCPI == 1){
655         LATEBits.LATE0 = 1;
656     }
657     else{
658         LATEBits.LATE0 = 0;
659     }
660 }
661
662 void __interrupt(irq(IRO_CCPI)) CCP41_ISR(void){
663     PIR3bits.CCP41IF = 0;
664     if(CCPI == 1){
665         LATEBits.LATE0 = 1;
666     }
667     else{
668         LATEBits.LATE0 = 0;
669     }
670 }
671
672 void __interrupt(irq(IRO_CCPI)) CCP42_ISR(void){
673     PIR3bits.CCP42IF = 0;
674     if(CCPI == 1){
675         LATEBits.LATE0 = 1;
676     }
677     else{
678         LATEBits.LATE0 = 0;
679     }
680 }
681
682 void __interrupt(irq(IRO_CCPI)) CCP43_ISR(void){
683     PIR3bits.CCP43IF = 0;
684     if(CCPI == 1){
685         LATEBits.LATE0 = 1;
686     }
687     else{
688         LATEBits.LATE0 = 0;
689     }
690 }
691
692 void __interrupt(irq(IRO_CCPI)) CCP44_ISR(void){
693     PIR3bits.CCP44IF = 0;
694     if(CCPI == 1){
695         LATEBits.LATE0 = 1;
696     }
697     else{
698         LATEBits.LATE0 = 0;
699     }
700 }
701
702 void __interrupt(irq(IRO_CCPI)) CCP45_ISR(void){
703     PIR3bits.CCP45IF = 0;
704     if(CCPI == 1){
705         LATEBits.LATE0 = 1;
706     }
707     else{
708         LATEBits.LATE0 = 0;
709     }
710 }
711
712 void __interrupt(irq(IRO_CCPI)) CCP46_ISR(void){
713     PIR3bits.CCP46IF = 0;
714     if(CCPI == 1){
715         LATEBits.LATE0 = 1;
716     }
717     else{
718         LATEBits.LATE0 = 0;
719     }
720 }
721
722 void __interrupt(irq(IRO_CCPI)) CCP47_ISR(void){
723     PIR3bits.CCP47IF = 0;
724     if(CCPI == 1){
725         LATEBits.LATE0 = 1;
726     }
727     else{
728         LATEBits.LATE0 = 0;
729     }
730 }
731
732 void __interrupt(irq(IRO_CCPI)) CCP48_ISR(void){
733     PIR3bits.CCP48IF = 0;
734     if(CCPI == 1){
735         LATEBits.LATE0 = 1;
736     }
737     else{
738         LATEBits.LATE0 = 0;
739     }
740 }
741
742 void __interrupt(irq(IRO_CCPI)) CCP49_ISR(void){
743     PIR3bits.CCP49IF = 0;
744     if(CCPI == 1){
745         LATEBits.LATE0 = 1;
746     }
747     else{
748         LATEBits.LATE0 = 0;
749     }
750 }
751
752 void __interrupt(irq(IRO_CCPI)) CCP50_ISR(void){
753     PIR3bits.CCP50IF = 0;
754     if(CCPI == 1){
755         LATEBits.LATE0 = 1;
756     }
757     else{
758         LATEBits.LATE0 = 0;
759     }
760 }
761
762 void __interrupt(irq(IRO_CCPI)) CCP51_ISR(void){
763     PIR3bits.CCP51IF = 0;
764     if(CCPI == 1){
765         LATEBits.LATE0 = 1;
766     }
767     else{
768         LATEBits.LATE0 = 0;
769     }
770 }
771
772 void __interrupt(irq(IRO_CCPI)) CCP52_ISR(void){
773     PIR3bits.CCP52IF = 0;
774     if(CCPI == 1){
775         LATEBits.LATE0 = 1;
776     }
777     else{
778         LATEBits.LATE0 = 0;
779     }
780 }
781
782 void __interrupt(irq(IRO_CCPI)) CCP53_ISR(void){
783     PIR3bits.CCP53IF = 0;
784     if(CCPI == 1){
785         LATEBits.LATE0 = 1;
786     }
787     else{
788         LATEBits.LATE0 = 0;
789     }
790 }
791
792 void __interrupt(irq(IRO_CCPI)) CCP54_ISR(void){
793     PIR3bits.CCP54IF = 0;
794     if(CCPI == 1){
795         LATEBits.LATE0 = 1;
796     }
797     else{
798         LATEBits.LATE0 = 0;
799     }
800 }
801
802 void __interrupt(irq(IRO_CCPI)) CCP55_ISR(void){
803     PIR3bits.CCP55IF = 0;
804     if(CCPI == 1){
805         LATEBits.LATE0 = 1;
806     }
807     else{
808         LATEBits.LATE0 = 0;
809     }
810 }
811
812 void __interrupt(irq(IRO_CCPI)) CCP56_ISR(void){
813     PIR3bits.CCP56IF = 0;
814     if(CCPI == 1){
815         LATEBits.LATE0 = 1;
816     }
817     else{
818         LATEBits.LATE0 = 0;
819     }
820 }
821
822 void __interrupt(irq(IRO_CCPI)) CCP57_ISR(void){
823     PIR3bits.CCP57IF = 0;
824     if(CCPI == 1){
825         LATEBits.LATE0 = 1;
826     }
827     else{
828         LATEBits.LATE0 = 0;
829     }
830 }
831
832 void __interrupt(irq(IRO_CCPI)) CCP58_ISR(void){
833     PIR3bits.CCP58IF = 0;
834     if(CCPI == 1){
835         LATEBits.LATE0 = 1;
836     }
837     else{
838         LATEBits.LATE0 = 0;
839     }
840 }
841
842 void __interrupt(irq(IRO_CCPI)) CCP59_ISR(void){
843     PIR3bits.CCP59IF = 0;
844     if(CCPI == 1){
845         LATEBits.LATE0 = 1;
846     }
847     else{
848         LATEBits.LATE0 = 0;
849     }
850 }
851
852 void __interrupt(irq(IRO_CCPI)) CCP60_ISR(void){
853     PIR3bits.CCP60IF = 0;
854     if(CCPI == 1){
855         LATEBits.LATE0 = 1;
856     }
857     else{
858         LATEBits.LATE0 = 0;
859     }
860 }
861
862 void __interrupt(irq(IRO_CCPI)) CCP61_ISR(void){
863     PIR3bits.CCP61IF = 0;
864     if(CCPI == 1){
865         LATEBits.LATE0 = 1;
866     }
867     else{
868         LATEBits.LATE0 = 0;
869     }
870 }
871
872 void __interrupt(irq(IRO_CCPI)) CCP62_ISR(void){
873     PIR3bits.CCP62IF = 0;
874     if(CCPI == 1){
875         LATEBits.LATE0 = 1;
876     }
877     else{
878         LATEBits.LATE0 = 0;
879     }
880 }
881
882 void __interrupt(irq(IRO_CCPI)) CCP63_ISR(void){
883     PIR3bits.CCP63IF = 0;
884     if(CCPI == 1){
885         LATEBits.LATE0 = 1;
886     }
887     else{
888         LATEBits.LATE0 = 0;
889     }
890 }
891
892 void __interrupt(irq(IRO_CCPI)) CCP64_ISR(void){
893     PIR3bits.CCP64IF = 0;
894     if(CCPI == 1){
895         LATEBits.LATE0 = 1;
896     }
897     else{
898         LATEBits.LATE0 = 0;
899     }
900 }
901
902 void __interrupt(irq(IRO_CCPI)) CCP65_ISR(void){
903     PIR3bits.CCP65IF = 0;
904     if(CCPI == 1){
905         LATEBits.LATE0 = 1;
906     }
907     else{
908         LATEBits.LATE0 = 0;
909     }
910 }
911
912 void __interrupt(irq(IRO_CCPI)) CCP66_ISR(void){
913     PIR3bits.CCP66IF = 0;
914     if(CCPI == 1){
915         LATEBits.LATE0 = 1;
916     }
917     else{
918         LATEBits.LATE0 = 0;
919     }
920 }
921
922 void __interrupt(irq(IRO_CCPI)) CCP67_ISR(void){
923     PIR3bits.CCP67IF = 0;
924     if(CCPI == 1){
925         LATEBits.LATE0 = 1;
926     }
927     else{
928         LATEBits.LATE0 = 0;
929     }
930 }
931
932 void __interrupt(irq(IRO_CCPI)) CCP68_ISR(void){
933     PIR3bits.CCP68IF = 0;
934     if(CCPI == 1){
935         LATEBits.LATE0 = 1;
936     }
937     else{
938         LATEBits.LATE0 = 0;
939     }
940 }
941
942 void __interrupt(irq(IRO_CCPI)) CCP69_ISR(void){
943     PIR3bits.CCP69IF = 0;
944     if(CCPI == 1){
945         LATEBits.LATE0 = 1;
946     }
947     else{
948         LATEBits.LATE0 = 0;
949     }
950 }
951
952 void __interrupt(irq(IRO_CCPI)) CCP70_ISR(void){
953     PIR3bits.CCP70IF = 0;
954     if(CCPI == 1){
955         LATEBits.LATE0 = 1;
956     }
957     else{
958         LATEBits.LATE0 = 0;
959     }
960 }
961
962 void __interrupt(irq(IRO_CCPI)) CCP71_ISR(void){
963     PIR3bits.CCP71IF = 0;
964     if(CCPI == 1){
965         LATEBits.LATE0 = 1;
966     }
967     else{
968         LATEBits.LATE0 = 0;
969     }
970 }
971
972 void __interrupt(irq(IRO_CCPI)) CCP72_ISR(void){
973     PIR3bits.CCP72IF = 0;
974     if(CCPI == 1){
975         LATEBits.LATE0 = 1;
976     }
977     else{
978         LATEBits.LATE0 = 0;
979     }
980 }
981
982 void __interrupt(irq(IRO_CCPI)) CCP73_ISR(void){
983     PIR3bits.CCP73IF = 0;
984     if(CCPI == 1){
985         LATEBits.LATE0 = 1;
986     }
987     else{
988         LATEBits.LATE0 = 0;
989     }
990 }
991
992 void __interrupt(irq(IRO_CCPI)) CCP74_ISR(void){
993     PIR3bits.CCP74IF = 0;
994     if(CCPI == 1){
995         LATEBits.LATE0 = 1;
996     }
997     else{
998         LATEBits.LATE0 = 0;
999     }
1000 }
1001
1002 void __interrupt(irq(IRO_CCPI)) CCP75_ISR(void){
1003     PIR3bits.CCP75IF = 0;
1004     if(CCPI == 1){
1005         LATEBits.LATE0 = 1;
1006     }
1007     else{
1008         LATEBits.LATE0 = 0;
1009     }
1010 }
1011
1012 void __interrupt(irq(IRO_CCPI)) CCP76_ISR(void){
1013    
```