Microcontroladores

Profesor: Kalun José Lau Gan

Semestre 2023-2

Semana 10

1

¿Preguntas previas?

- Puede explicarnos sobre interrupciones vectorizadas en el XC8 con el Curiosity Nano?
  - Hoy atenderemos ese punto
- ¿Cómo puedo construir un reloj con el Curiosity Nano?
  - Empleando el Timer1 ó algún módulo RTC externo (DS1307, DS3231 ó similares)
- ¿MPLAB Xpress puede correr fuera de una PC?
  - Como se evidenció la semana pasada, MPLAB Xpress al ser una plataforma que corre en un navegador web, se puede emplear cualquier máquina que tenga ello (un web browser), pudiendo ser una PC desktop corriendo Windows, Linux, MacOS, Android. Hasta desde un celular puedes desarrollar una aplicación con el Curiosity Nano!

2

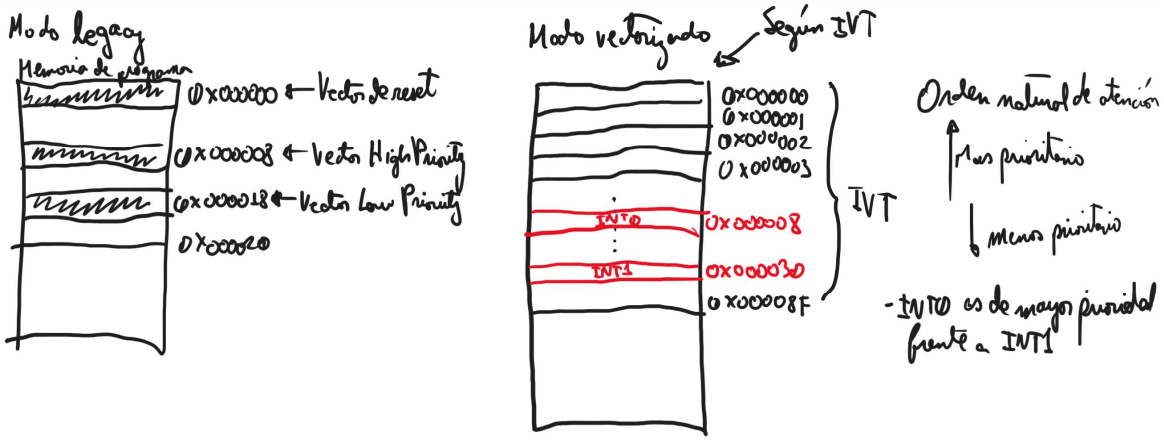
Agenda:

- Interrupciones vectorizadas en XC8
- El Timer1 para RTC
  - Timer1 junto con el CCP1 en modo comparador
- Interrupciones externas INT0 e INT1

3

Interrupciones vectorizadas a detalle

- Hay una tabla de interrupciones vectorizadas (IVT)
- Cada fuente de interrupción que posee el microcontrolador se encuentra en una dirección en particular en la IVT



4

Interrupciones vectorizadas a detalle

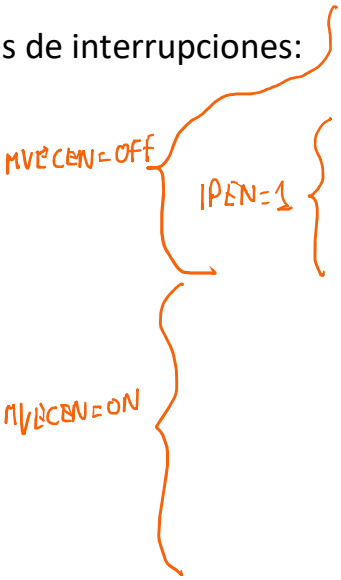
- La IVT:
- La atención de las interrupciones en el orden natural: menor valor de dirección mayor prioridad.

Vector Number	Interrupt source	Vector Number (cont.)	Interrupt source (cont.)
0x0	Software Interrupt	0x3E	PWM3RINT
0x1	HLVD (High/Low-Voltage Detect)	0x3F	PWM3GINT
0x2	OSF (Oscillator Fail)	0x40	U2RX
0x3	CSW (Clock Switching)	0x41	U2TX
0x4	-	0x42	U2E
0x5	CLC1 (Configurable Logic Cell)	0x43	U2
0x6	-	0x44	TMR5
0x7	IOC (Interrupt-On-Change)	0x45	TMR5G
0x8	INT0	0x46	CCP2
0x9	ZCD (Zero-Cross Detection)	0x47	SCAN
0xA	AD (ADC Conversion Complete)	0x48	U3RX
0xB	ACT (Active Clock Tuning)	0x49	U3TX
0xC	CM1 (Comparator)	0x4A	U3E
0xD	SMT1 (Signal Measurement Timer)	0x4B	U3
0xE	SMT1PRA	0x4C	-
0xF	SMT1PWA	0x4D	CLC4
0x10	ADT	0x4E - 0x4F	-
0x11 - 0x13	-	0x50	INT2
0x14	DMA1SCNT (Direct Memory Access)	0x51	CLC5
0x15	DMA1DCNT	0x52	CWG2 (Complementary Waveform Generator)
0x16	DMA1OR	0x53	NCO2
0x17	DMA1A	0x54	DMA3SCNT
0x18	SPI1RX (Serial Peripheral Interface)	0x55	DMA3DCNT
0x19	SPI1TX	0x56	DMA3OR
0x1A	SPI1	0x57	DMA3A
0x1B	TMR2	0x58	CCP3
0x1C	TMR1	0x59	CLC6
0x1D	TMR1G	0x5A	CWG3
0x1E	CCP1 (Capture/Compare/PWM)	0x5B	TMR4
0x1F	TMR0	0x5C	DMA4SCNT
0x20	U1RX	0x5D	DMA4DCNT
0x21	U1TX	0x5E	DMA4OR
0x22	U1E	0x5F	DMA4A
0x23	U1	0x60	U4RX
0x24 - 0x25	-	0x61	U4TX
0x26	PWM1RINT	0x62	U4E
0x27	PWM1GINT	0x63	U4
0x28	SPI2RX	0x64	DMA5SCNT
0x29	SPI2TX	0x65	DMA5DCNT
0x2A	SPI2	0x66	DMA5OR
0x2B	-	0x67	DMA5A
0x2C	TMR3	0x68	U5RX
0x2D	TMR3G	0x69	U5TX
0x2E	PWM2RINT	0x6A	U5E
0x2F	PWM2GINT	0x6B	U5
0x30	INT1	0x6C	DMA6SCNT
0x31	CLC2	0x6D	DMA6DCNT
0x32	CWG1 (Complementary Waveform Generator)	0x6E	DMA6OR
0x33	NCO1 (Numerically Controlled Oscillator)	0x6F	DMA6A
0x34	DMA2SCNT	0x70	-
0x35	DMA2DCNT	0x71	CLC7
0x36	DMA2OR	0x72	CM2
0x37	DMA2A	0x73	NCO3
0x38	I2C1RX	0x74 - 0x77	-
0x39	I2C1TX	0x78	NVM
0x3A	I2C1	0x79	CLC8
0x3B	I2C1E	0x7A	CRC (Cyclic Redundancy Check)
0x3C	-	0x7B	TMR6
0x3D	CLC3	0x7C - 0x8F	-

5

Interrupciones vectorizadas a detalle

- Plantilla de funciones de interrupciones:




```
//--------- un sola fuente de interrupcion
void __interrupt() INT_ISR(void){
//--
}
```

```
//--------- con prioridades en modo legacy
void __interrupt(high_priority) INT_HP_ISR(void){
//--
}

void __interrupt(low_priority) INT_LP_ISR(void){
//--
}
```

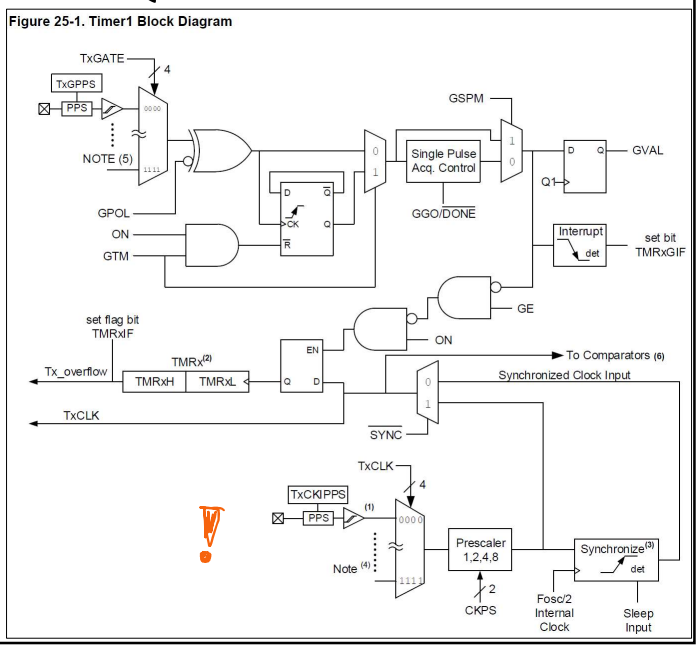
```
//--------- con prioridades de orden natural segun IVT
void __interrupt(irq(IRQ_INT0) INT0_ISR(void){
//--
}

void __interrupt(irq(IRQ_INT1) INT0_ISR(void){
//--
}

void __interrupt(irq(IRQ_INT2) INT0_ISR(void){
//--
}

void __interrupt(irq(IRQ_TMR0) INT0_ISR(void){
//--
}
```

6

El Timer 1 en el PIC18F57Q43

- Módulo temporizador de 16 bits
- Empleado para aplicaciones de RTC
- Función de gate
- Funciona con el CCP1 (modo captura ó modo comparación)
- Múltiples fuentes de reloj

Figure 25-1. Timer1 Block Diagram



7

El Timer 1 en el PIC18F57Q43

- Selección de la fuente de reloj (registro T1CLK)

Name: TxCLK
Address: 0x321,0x32D,0x339

Timer Clock Source Selection Register

Bit	7	6	5	4	3	2	1	0
				CS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – CS[4:0] Timer Clock Source Selection

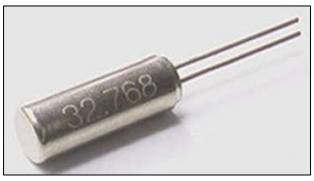
Table 25-4. Timer Clock Sources

CS	Clock Source		
	Timer1	Timer3	Timer5
11111-10110		Reserved	
10101		CLC8_OUT	
10100		CLC7_OUT	
10011		CLC6_OUT	
10010		CLC5_OUT	
10001		CLC4_OUT	
10000		CLC3_OUT	
01111		CLC2_OUT	
01110		CLC1_OUT	
01101	TMR5_OUT	TMR5_OUT	Reserved
01100	TMR3_OUT	Reserved	TMR3_OUT
01011	Reserved	TMR1_OUT	TMR1_OUT
01010		TMR0_OUT	
01001		CLKREF_OUT	
01000		EXTOSC	
00111		SOSC	
00110		MFINTOSC (31.25 kHz)	
00101		MFINTOSC (500 kHz)	
00100		LFINTOSC	
00011		HFINTOSC	
00010		Fosc	
00001		Fosc/4	
00000	Pin selected by T1CKIPPS	Pin selected by T3CKIPPS	Pin selected by T5CKIPPS

Reset States: POR/BOR = 00000
All Other Resets = uuuuu

8

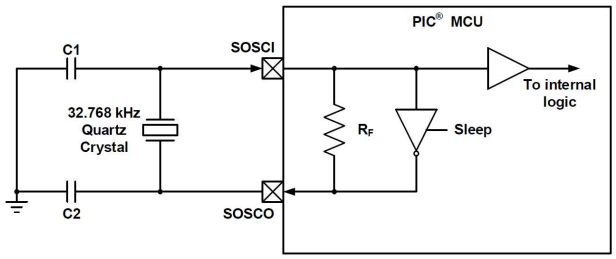
El Timer 1 en el PIC18F57Q43



- Oscilador secundario

12.1.1.5 Secondary Oscillator

The Secondary Oscillator (SOSC) is a separate external oscillator block that can be used as an alternate system clock source or as a Timer clock source. The SOSC is optimized for 32.768 kHz, and can be used with either an external quartz crystal connected to the SOSCI and SOSCO pins, or with an external clock source connected to the SOSCI pin as shown in the figures below.



9

El CCP1 en el PIC18F57Q43

- Módulo captura/comparación/PWM

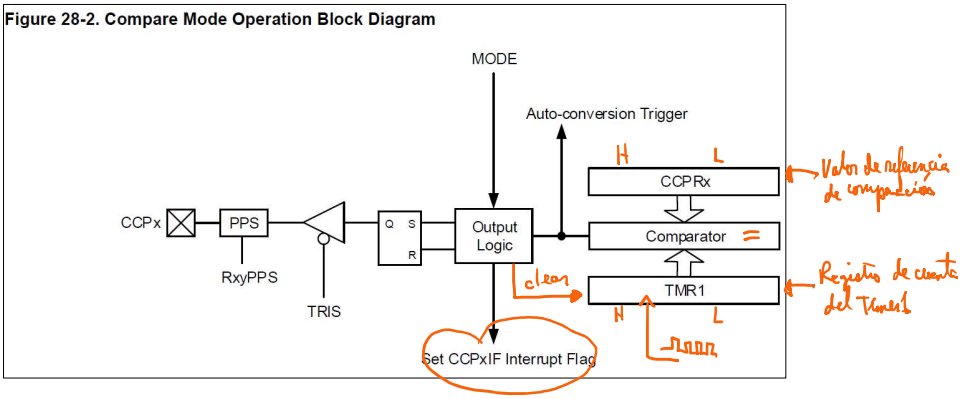
CCP Mode	Timer Resource
Capture	Timer1, Timer3 or Timer5
Compare	
PWM	Timer2, Timer4 or Timer6

10

El CCP1 en el PIC18F57Q43

- CCP1 en modo comparación
- Diagrama de bloques

Figure 28-2. Compare Mode Operation Block Diagram



11

El CCP1 en el PIC18F57Q43

- CCP1 en modo comparación
- Información importante:

28.3.2 Timer1 Mode for Compare

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

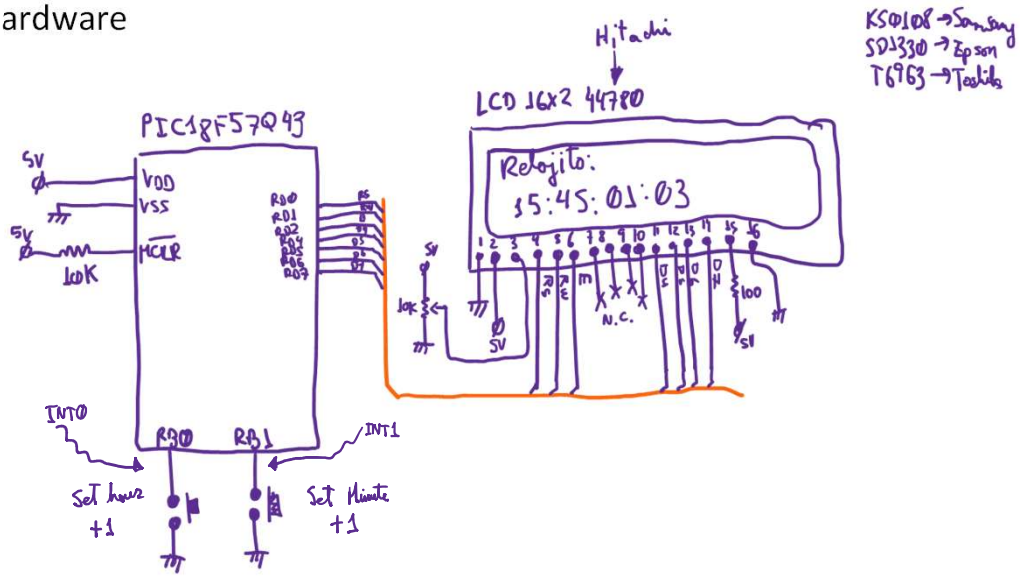
See the "**TMR1 - Timer1 Module with Gate Control**" section for more information on configuring Timer1.

Important: Clocking Timer1 from the system clock (F_{OSC}) must not be used in Compare mode. For Compare mode to recognize the trigger event on the CCPx pin, Timer1 must be clocked from the instruction clock ($F_{OSC}/4$) or from an external clock source.

12

Ejercicio: Reloj empleando el Timer1

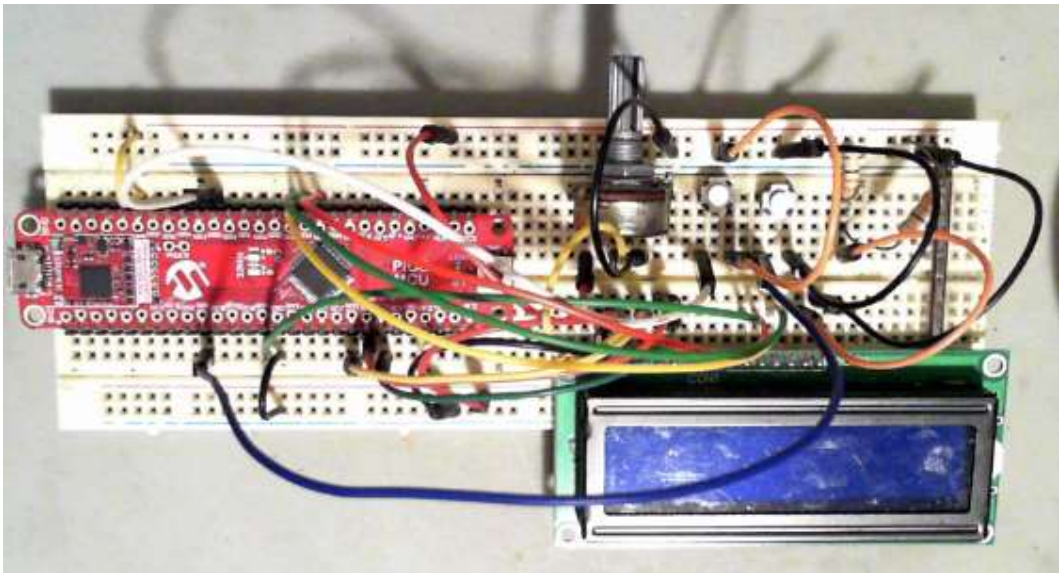
- Hardware



13

Ejercicio: Reloj empleando el Timer1

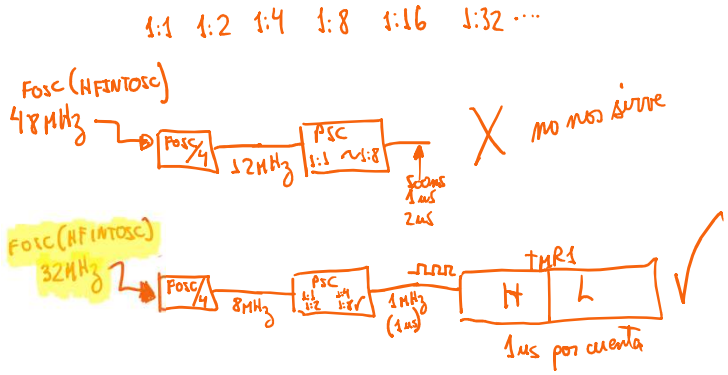
- Hardware



14

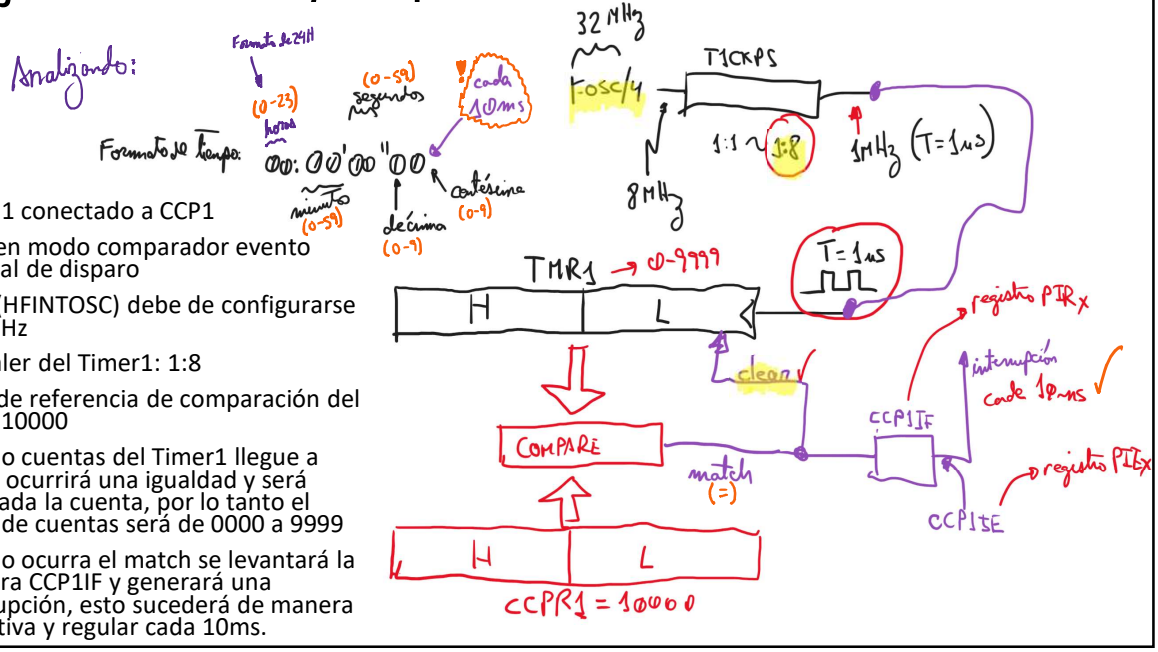
Ejercicio: Reloj empleando el Timer1

- A tener de cuenta con respecto a la frecuencia del reloj principal



15

Ejercicio: Reloj empleando el Timer1

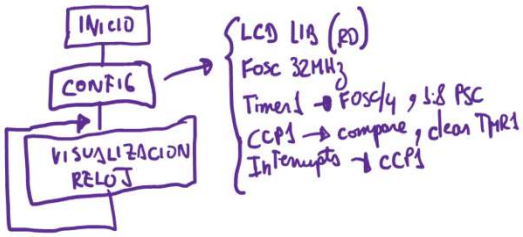


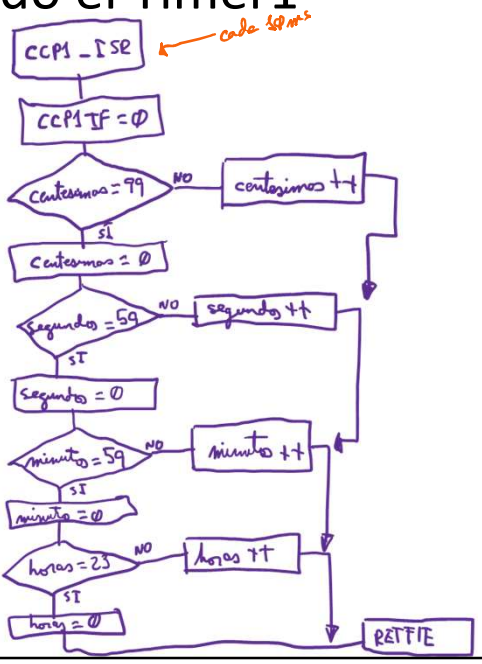
- Timer 1 conectado a CCP1
- CCP1 en modo comparador evento especial de disparo
- FOSC (HFINTOSC) debe de configurarse a 32MHz
- Prescaler del Timer1: 1:8
- Valor de referencia de comparación del CCP1: 10000
- Cuando cuentas del Timer1 llegue a 10000 ocurrirá una igualdad y será reiniciada la cuenta, por lo tanto el rengo de cuentas será de 0000 a 9999
- Cuando ocurra el match se levantará la bandera CCP1IF y generará una interrupción, esto sucederá de manera repetitiva y regular cada 10ms.

16

Ejercicio: Reloj empleando el Timer1

- Algoritmo





17

Ejercicio: Reloj empleando el Timer1

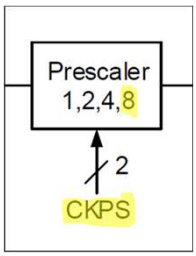
- Tres tareas principales a cumplir (base de tiempo):
  - Configurar el Timer1
  - Configurar el CCP1
  - Configurar las interrupciones

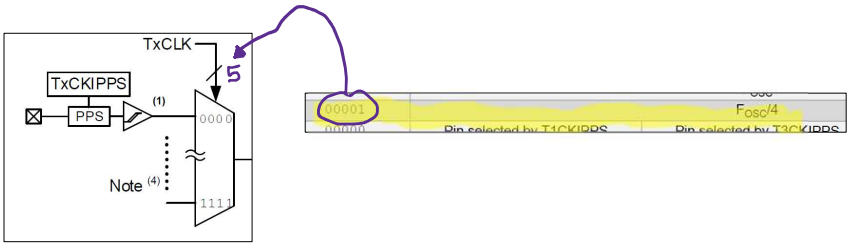
18

Ejercicio: Reloj empleando el Timer1

- Configuración del Timer1:

Address	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x031C	TMR1	7:0	TMR1[7:0]							
		15:8	TMR1[15:8]							
0x031E	T1CON	7:0			CKPS[1:0]			SYNC	RD16	ON
0x031F	T1GCON	7:0	GE	GPOL	GTM	GSPM	GGO/DONE	GVAL		
0x0320	T1GATE	7:0			GSS[5:0]					
0x0321	T1CLK	7:0						CS[4:0]		





19

Ejercicio: Reloj empleando el Timer1

- Registro T1CLK

T1CLK = 0x01

Source Fosc/4

Name: TxCLK
Address: 0x321,0x32D,0x339

Timer Clock Source Selection Register

Bit	7	6	5	4	3	2	1	0
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – CS[4:0] Timer Clock Source Selection

Table 25-4. Timer Clock Sources

CS	Clock Source		
	Timer1	Timer3	Timer5
11111–10110	Reserved		
10101	CLC8_OUT		
10100	CLC7_OUT		
10011	CLC6_OUT		
10010	CLC5_OUT		
10001	CLC4_OUT		
10000	CLC3_OUT		
01111	CLC2_OUT		
01110	CLC1_OUT		
01101	TMR5_OUT	TMR5_OUT	Reserved
01100	TMR3_OUT	Reserved	TMR3_OUT
01011	Reserved	TMR1_OUT	TMR1_OUT
01010	TMR0_OUT		
01001	CLKREF_OUT		
01000	EXTOSC		
00111	SOSC		
00110	MFINTOSC (31.25 kHz)		
00101	MFINTOSC (500 kHz)		
00100	LFINTOSC		
00011	HFINTOSC		
00010	Fosc		
00001	Fosc/4		
00000	Pin selected by T1CKIPPS	Pin selected by T3CKIPPS	Pin selected by T5CKIPPS

20

Ejercicio: Reloj empleando el Timer1

- Registro T1CON:

T1CON = 0x33

Prescaler 1:8

Timer1 ON

Name: TxCON
Address: 0x31E,0x32A,0x336

Timer Control Register

Bit	7	6	5	4	3	2	1	0
			CKPS[1:0]			SYNC	RD16	ON
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 5:4 – CKPS[1:0] Timer Input Clock Prescaler Select
Reset States: POR/BOR = 00
All Other Resets = uu

Value	Description
11	1:8 Prescaler value
10	1:4 Prescaler value
01	1:2 Prescaler value
00	1:1 Prescaler value

Bit 2 – $\overline{\text{SYNC}}$ Timer External Clock Input Synchronization Control
Reset States: POR/BOR = 0
All Other Resets = u

Value	Condition	Description
x	CS = $F_{OSC}/4$ or F_{OSC}	This bit is ignored. Timer uses the incoming clock as is.
1	All other clock sources	Do not synchronize external clock input
0	All other clock sources	Synchronize external clock input with system clock

Bit 1 – RD16 16-Bit Read/Write Mode Enable
Reset States: POR/BOR = 0
All Other Resets = u

Value	Description
1	Enables register read/write of Timer in one 16-bit operation
0	Enables register read/write of Timer in two 8-bit operations

Bit 0 – ON Timer On
Reset States: POR/BOR = 0
All Other Resets = u

Value	Description
1	Enables Timer
0	Disables Timer

21

Ejercicio: Reloj empleando el Timer1

- Configuración del CCP1:

Address	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0340	CCPR1	7:0	CCPR[7:0]							
		15:8	CCPR[15:8]							
0x0342	CCP1CON	7:0	EN		OUT	FMT	MODE[3:0]			
0x0343	CCP1CAP	7:0					CTS[3:0]			

CCPR1H = 0x27

CCPR1L = 0x10

Valor de referencia del comparador

10000 en decimal

CCPR1 = 10000; X

22

Ejercicio: Reloj empleando el Timer1

- Registro CCP1CON:

CCP1CON = 0x81

Name: CCPxCON
Address: 0x342,0x346,0x34A

CCP Control Register

Bit	7	6	5	4	3	2	1	0
	EN		OUT	FMT	MODE[3:0]			
Access	R/W		R	R/W	R/W	R/W	R/W	R/W
Reset	0		x	0	0	0	0	0

Bit 7 – EN CCP Module Enable

Value	Description
1	CCP is enabled
0	CCP is disabled

Bit 5 – OUT CCP Output Data (read-only)

Bit 4 – FMT CCPxRH:L Value Alignment (PWM mode)

Value	Condition	Description
x	Capture mode	Not used
x	Compare mode	Not used
1	PWM mode	Left aligned format
0	PWM mode	Right aligned format

Bits 3:0 – MODE[3:0] CCP Mode Select

Table 28-5. CCPx Mode Select

MODE Value	Operating Mode	Operation	Set CCPxIF
11xx	PWM	PWM operation	Yes
1011	Compare	Pulse output; clear TMR1[2]	Yes
1010		Pulse output	Yes
1001		Clear output[1]	Yes
1000		Set output[1]	Yes
0111	Capture	Every 16th rising edge of CCPx input	Yes
0110		Every 4th rising edge of CCPx input	Yes
0101		Every rising edge of CCPx input	Yes
0100		Every falling edge of CCPx input	Yes
0011		Every edge of CCPx input	Yes
0010	Compare	Toggle output	Yes
0001		Toggle output; clear TMR1[2]	Yes
0000	Disabled		—

23

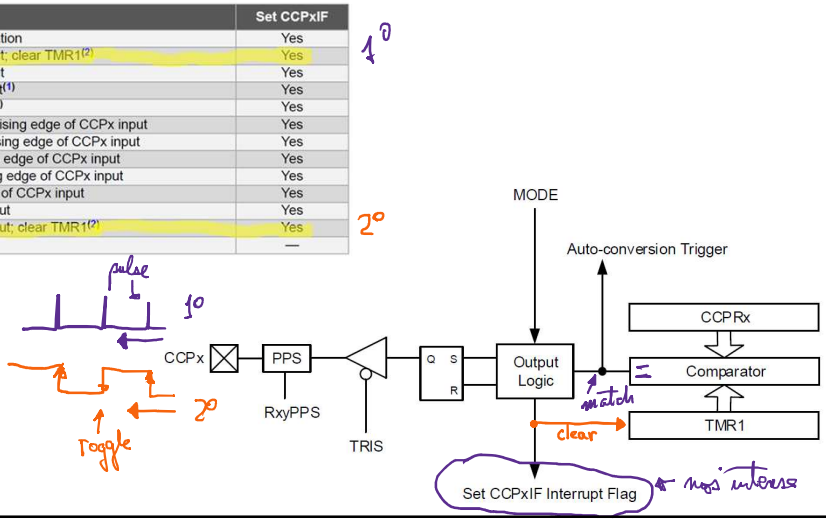
Ejercicio: Reloj empleando el Timer1

- Registro CCP1CON:

Bits 3:0 – MODE[3:0] CCP Mode Select

Table 28-5. CCPx Mode Select

MODE Value	Operating Mode	Operation	Set CCPxIF
11xx	PWM	PWM operation	Yes
1011	Compare	Pulse output; clear TMR1[2]	Yes
1010		Pulse output	Yes
1001		Clear output[1]	Yes
1000		Set output[1]	Yes
0111	Capture	Every 16th rising edge of CCPx input	Yes
0110		Every 4th rising edge of CCPx input	Yes
0101		Every rising edge of CCPx input	Yes
0100		Every falling edge of CCPx input	Yes
0011		Every edge of CCPx input	Yes
0010	Compare	Toggle output	Yes
0001		Toggle output; clear TMR1[2]	Yes
0000	Disabled		—



24

Ejercicio: Reloj empleando el Timer1

- Configuración de las interrupciones del CCP1:

0x04A1	PIE3	7:0	TMR0IE	CCP1IE	TMR1GIE	TMR1IE	TMR2IE	SPI1IE	SPI1TXIE	SPI1RXIE

1 → habilitación de la int del CCP1

0x04D6	INTCON0	7:0	GIE/GIEH	GIEL	IPEN			INT2EDG	INT1EDG	INT0EDG

1 → habilitación global de interrupciones

0x04B1	PIR3	7:0	TMR0IF	CCP1IF	TMR1GIF	TMR1IF	TMR2IF	SPI1IF	SPI1TXIF	SPI1RXIF

bandera

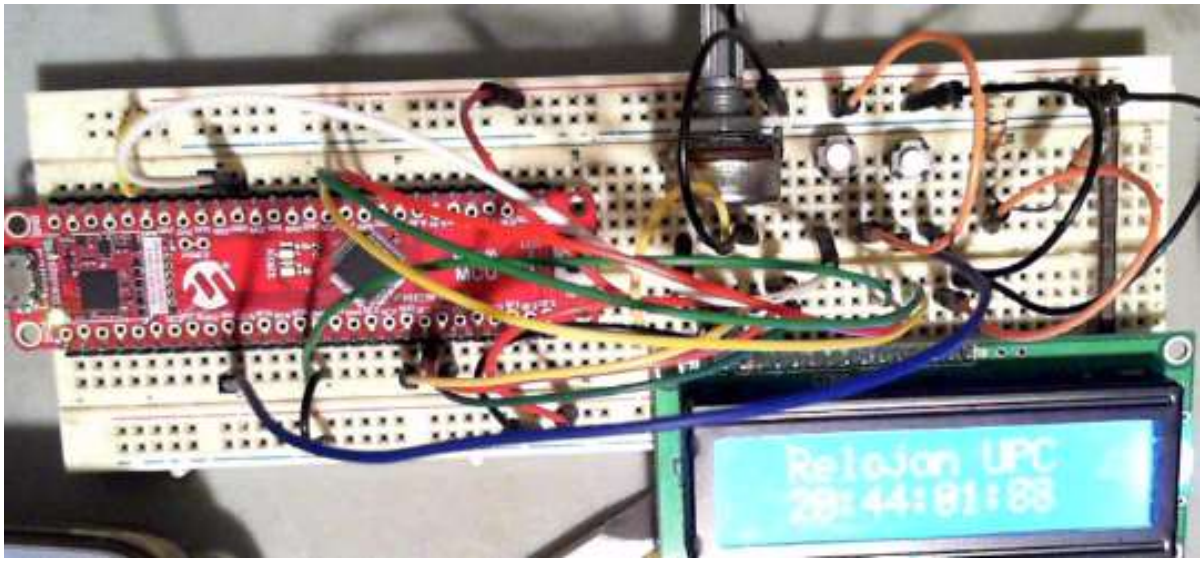
Ejercicio: Reloj empleando el Timer1

```c
#include <xc.h>
#include "cabecera.h"
#include "LCD.h"
#define _XTAL_FREQ 32000000UL

unsigned char horas=20,minutos=02,segundos=16,centesimas=58;
unsigned char centena,decena,unidad;

void configuro(void){
    //configuracion del oscilador
    OSCCON1 = 0x60;
    OSCFRQ = 0x06;  //HFINTOSC a 32MHz
    OSCEN = 0x40;
    //configuracion del Timer1
    T1CLK = 0x01;
    T1CON = 0x33;
    //configuracion del CCP1
    CCP1CON = 0x81;
    CCPR1H = 0x27;
    CCPR1L = 0x10;
    //configuracion de las interrupciones
    PIE3bits.CCP1IE = 1;
    INTCON0bits.GIE = 1;
}

void lcd_init(void){
    TRISD = 0x00;
    ANSELD = 0x00;
    LCD_CONFIG();
    __delay_ms(21);
    BORRAR_LCD();
    CURSOR_HOME();
    CURSOR_ONOFF(OFF);
}
```

```c
void convierte(unsigned char numero){
    centena = numero / 100;
    decena = (numero % 100) / 10;
    unidad = numero % 10;
}

void main(void) {
    configuro();
    lcd_init();
    POS_CURSOR(1,2);
    ESCRIBE_MENSAJE("Relojon UPC",11);
    while(1){
        POS_CURSOR(2,2);
        convierte(horas);
        ENVIA_CHAR(decena+0x30);
        ENVIA_CHAR(unidad+0x30);
        ENVIA_CHAR(':');
        convierte(minutos);
        ENVIA_CHAR(decena+0x30);
        ENVIA_CHAR(unidad+0x30);
        ENVIA_CHAR(':');
        convierte(segundos);
        ENVIA_CHAR(decena+0x30);
        ENVIA_CHAR(unidad+0x30);
        ENVIA_CHAR(':');
        convierte(centesimas);
        ENVIA_CHAR(decena+0x30);
        ENVIA_CHAR(unidad+0x30);
    }
}
```

```c
void __interrupt(irq(IRQ_CCP1)) CCP1_ISR(void){
    PIR3bits.CCP1IF = 0;
    if(centesimas == 99){
        centesimas = 0;
        if(segundos == 59){
            segundos = 0;
            if(minutos == 59){
                minutos = 0;
                if(horas == 23){
                    horas = 0;
                }
                else{
                    horas++;
                }
            }
            else{
                minutos++;
            }
        }
        else{
            segundos++;
        }
    }
    else{
        centesimas++;
    }
}
```

Ejercicio: Reloj empleando el Timer1



27

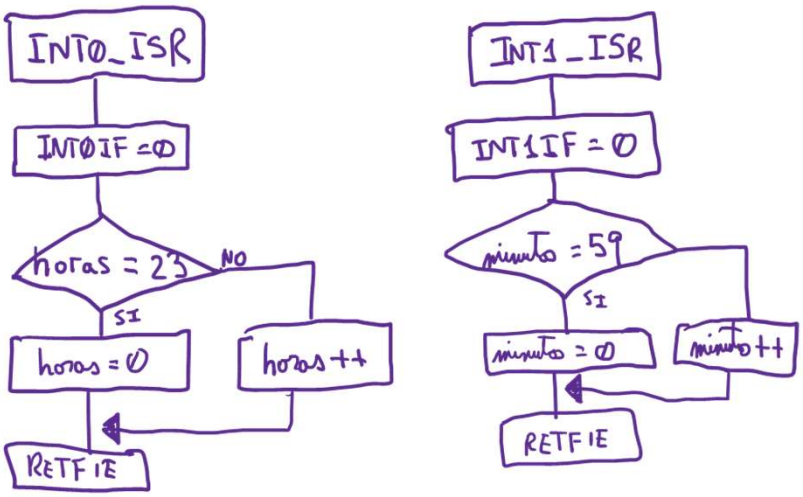
Ejercicio: Reloj empleando el Timer1

- Etapa de ajuste de la hora empleando los pulsadores en RB0/INT0 y RB1/INT1
  - Según el circuito implementado, los botones que están en INT0 e INT1 respectivamente son activos en bajo
  - Se debe de activar las pullup en ambos puertos
  - Ambas INTs deben de configurarse en flaco descendente (INT0EDG y INT1EDG deben de ser cero).

28

Ejercicio: Reloj empleando el Timer1

- Etapa de ajuste de la hora empleando los pulsadores en RB0/INT0 y RB1/INT1



29

Ejercicio: Reloj empleando el Timer1

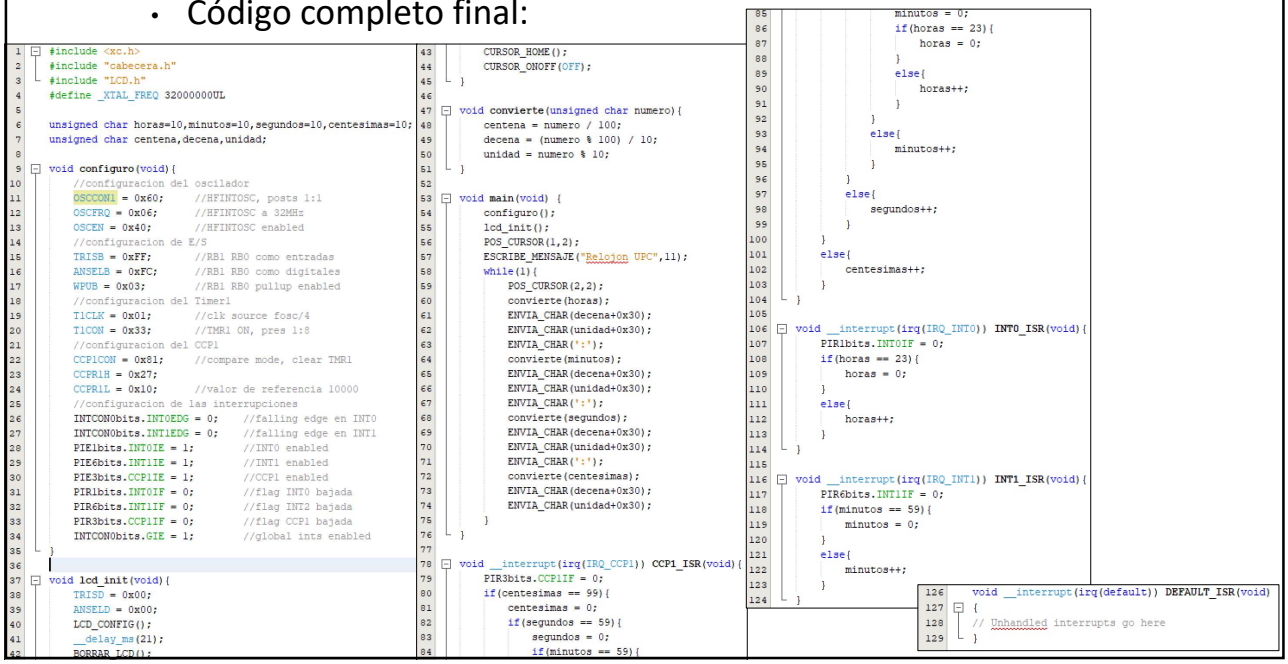
- Configuración de las interrupciones de INT0 e INT1:

Address	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x049F	PIE1	7:0	SMT1PWAIE	SMT1PRAIE	SMT1IE	CM1IE	ACTIE	ADIE	ZCDIE	INT0IE
0x04A4	PIE6	7:0	DMA2AIE	DMA2ORIE	DMA2DCNTIE	DMA2SCNTIE	NCO1IE	CWG1IE	CLC2IE	INT1IE
0x04D6	INTCON0	7:0	GIE/GIEH	GIEL	IPEN			INT2EDG	INT1EDG	INT0EDG
0x04AF	PIR1	7:0	SMT1PWAIF	SMT1PRAIF	SMT1IF	CM1IF	ACTIF	ADIF	ZCDIF	INT0IF
0x04B4	PIR6	7:0	DMA2AIF	DMA2ORIF	DMA2DCNTIF	DMA2SCNTIF	NCO1IF	CWG1IF	CLC2IF	INT1IF

30

Ejercicio: Reloj empleando el Timer1

- Código completo final:



```c
#include <xc.h>
#include "cabecera.h"
#include "LCD.h"
#define _XTAL_FREQ 32000000UL

unsigned char horas=10,minutos=10,segundos=10,centesimas=10;
unsigned char centena,decena,unidad;

void configuro(void){
    //configuracion del oscilador
    OSCCON1 = 0x60;     //HFINTOSC, posts 1:1
    OSCFRQ = 0x06;      //HFINTOSC a 32MHz
    OSCEN = 0x40;       //HFINTOSC enabled
    //configuracion de E/S
    TRISB = 0xFF;       //RB1 RB0 como entradas
    ANSELB = 0xFC;      //RB1 RB0 como digitales
    WPUB = 0x03;        //RB1 RB0 pullup enabled
    //configuracion del Timer1
    T1CLK = 0x01;       //clk source fosc/4
    T1CON = 0x33;       //TMR1 ON, pres 1:8
    //configuracion del CCP1
    CCP1CON = 0x81;     //compare mode, clear TMR1
    CCPR1H = 0x27;
    CCPR1L = 0x10;      //valor de referencia 10000
    //configuracion de las interrupciones
    INTCON0bits.INT0EDG = 0;    //falling edge en INT0
    INTCON0bits.INT1EDG = 0;    //falling edge en INT1
    PIE1bits.INT0IE = 1;        //INT0 enabled
    PIE6bits.INT1IE = 1;        //INT1 enabled
    PIE3bits.CCP1IE = 1;        //CCP1 enabled
    PIR1bits.INT0IF = 0;        //flag INT0 bajada
    PIR6bits.INT1IF = 0;        //flag INT2 bajada
    PIR3bits.CCP1IF = 0;        //flag CCP1 bajada
    INTCON0bits.GIE = 1;        //global ints enabled
}

void lcd_init(void){
    TRISD = 0x00;
    ANSELD = 0x00;
    LCD_CONFIG();
    __delay_ms(21);
    BORRAR_LCD();
    CURSOR_HOME();
    CURSOR_ONOFF(OFF);
}

void convierte(unsigned char numero){
    centena = numero / 100;
    decena = (numero % 100) / 10;
    unidad = numero % 10;
}

void main(void) {
    configuro();
    lcd_init();
    POS_CURSOR(1,2);
    ESCRIBE_MENSAJE("Relojon UPC",11);
    while(1){
        POS_CURSOR(2,2);
        convierte(horas);
        ENVIA_CHAR(decena+0x30);
        ENVIA_CHAR(unidad+0x30);
        ENVIA_CHAR(':');
        convierte(minutos);
        ENVIA_CHAR(decena+0x30);
        ENVIA_CHAR(unidad+0x30);
        ENVIA_CHAR(':');
        convierte(segundos);
        ENVIA_CHAR(decena+0x30);
        ENVIA_CHAR(unidad+0x30);
        ENVIA_CHAR(':');
        convierte(centesimas);
        ENVIA_CHAR(decena+0x30);
        ENVIA_CHAR(unidad+0x30);
    }
}

void __interrupt(irq(IRQ_CCP1)) CCP1_ISR(void){
    PIR3bits.CCP1IF = 0;
    if(centesimas == 99){
        centesimas = 0;
        if(segundos == 59){
            segundos = 0;
            if(minutos == 59){
                minutos = 0;
                if(horas == 23){
                    horas = 0;
                }
                else{
                    horas++;
                }
            }
            else{
                minutos++;
            }
        }
        else{
            segundos++;
        }
    }
    else{
        centesimas++;
    }
}

void __interrupt(irq(IRQ_INT0)) INT0_ISR(void){
    PIR1bits.INT0IF = 0;
    if(horas == 23){
        horas = 0;
    }
    else{
        horas++;
    }
}

void __interrupt(irq(IRQ_INT1)) INT1_ISR(void){
    PIR6bits.INT1IF = 0;
    if(minutos == 59){
        minutos = 0;
    }
    else{
        minutos++;
    }
}

void __interrupt(irq(default)) DEFAULT_ISR(void)
{
    // Unhandled interrupts go here
}
```

31

Ejercicio: Reloj empleando el Timer1

- Adicionales:
  - Cambio de formato 12H/24H
  - Sistema de alarma (activación/desactivación)
  - Función de cronómetro
  - Función de cuenta regresiva
  - Medición de temperatura / humedad con DHT11
  - Animación de secundero
  - Visualización de reloj analógico usando caracteres personalizados
  - Visualización de caracteres gigantes en el LCD
  - Sincronismo con NTP
  - Acelerómetro para encender la pantalla ante un movimiento

32

Fin de la sesión

33