

Microcontroladores

Semana 6

Semestre 2024-2

Por Kalun José Lau Gan

1

Preguntas previas

- ¿Cómo funcionan los sensores ultrasónicos?
 - Funcionando con señales ultrasónicas, se envia un pulso, éste rebota y regresa y se mide el tiempo recorrido por dicha señal
- ~~Para los del EL51, el siguiente miércoles cae 1 de mayo. ¿Cómo serán las clases?~~
 - ~~Ese mismo día serán las clases: 9:00AM - 11:00AM la teoría y laboratorio único 1PM - 5PM.~~
- ~~¿La semana 7 es LB2?~~
 - ~~Si, en el horario publicado en el punto anterior.~~
- No pude rendir la PC1. ¿Algo se podrá hacer?
 - Tienes posibilidad de rendir el examen de recuperación en la semana 17, para ello debes de hacer el trámite correspondiente, más información con tu profesor de campus

2

Agenda:

- Los periféricos temporizadores en el PIC18F57Q43
- El módulo Timer0
 - Modo 8bit
 - Modo 16bit
 - Registros de configuración y operación
- Cálculo de temporización
- Generación de ondas cuadradas periódicas

3

Los temporizadores en el PIC18F57Q43

- Según datasheet este microcontrolador tiene siete temporizadores: tres de 8 bits y cuatro de 16 bits.
- Pero en el índice del datasheet solo aparecen tres: Timer0, Timer1 y Timer2
- Timer3 y Timer5 tienen la misma funcionalidad que el Timer1
- Timer4 y Timer6 tienen la misma funcionalidad que el Timer2
- El Timer0 lo utilizamos para aplicaciones genéricas de temporización
- El Timer1 se emplea para aplicaciones de reloj en tiempo real, sigue trabajando cuando el CPU entra en modo SLEEP.
- El Timer2 está enfocado a la generación de señal PWM trabajando en conjunto con el módulo CCP.

4

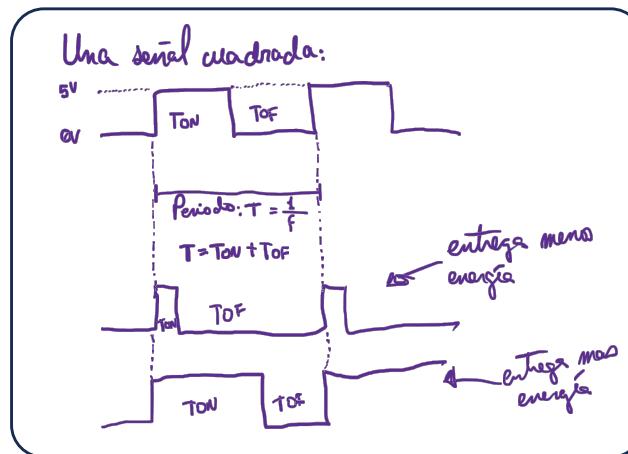
¿Para qué utilizamos temporizadores si con `delay()` se puede hacer también igual?

- La función `delay()`, en nuestro caso, un conjunto de instrucciones “nop”, ocasiona que el CPU se encuentre atendiendo dicha rutina de retardo, no dejando hacer otra cosa.
- Lo peor que puedes utilizar en un programa son retardos, éstos hacen que el desempeño global de tu aplicación se vea afectada.
- Una rutina de retardo en software es perder el tiempo en el CPU.

5

Señal PWM

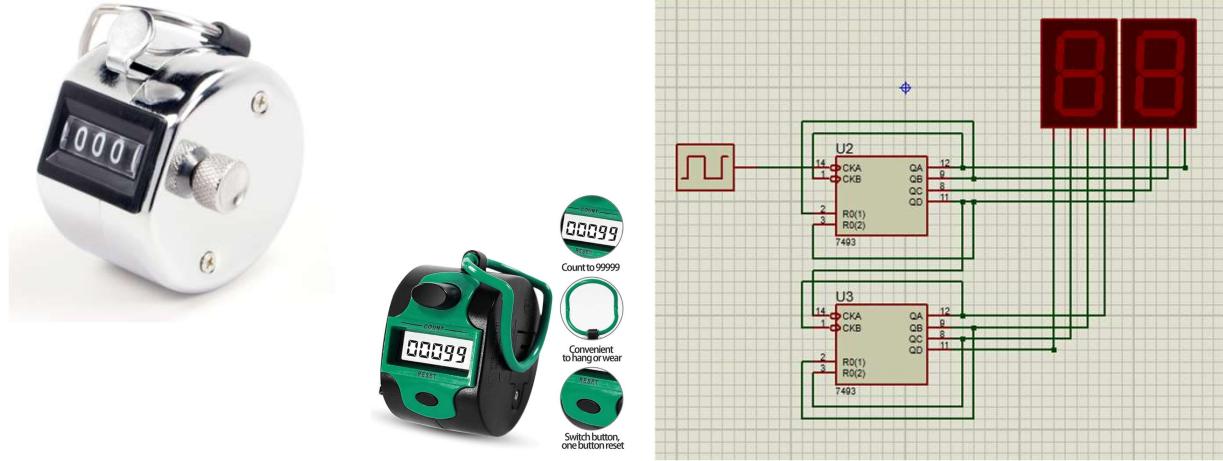
- Pulse Width Modulation: Modulación de ancho de pulso
- Se utilizar para obtener una señal de determinada cantidad de energía partiendo de una señal cuadrada.



6

La previa: contadores digitales

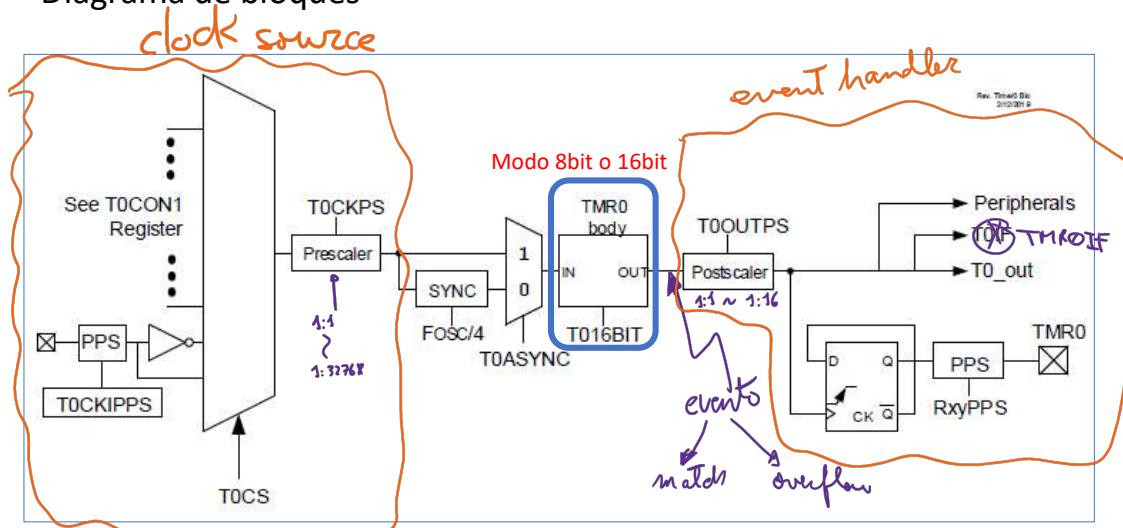
- Sistemas secuenciales, poseen reloj, un registro de cuenta, dependiendo de su función, puede incrementar o puede decrementar la cuenta según la entrada de reloj.



7

El módulo Timer 0:

- Diagrama de bloques



8

El módulo Timer 0

- (Ref. Item 24 de la hoja técnica del microcontrolador PIC18F57Q43)
- Temporizador de cuenta ascendente
- Modo de trabajo: 8 bits (0-254) ó 16 bits (0-65535)
- Las cuentas del Timer0 se alojan en:
 - TMROH:TMROL (16 bits)
 - TMROL (8 bits)
- Diversas fuentes de reloj (revisar TOCON1 y TOCS)
- Divisor de frecuencia al reloj de entrada PRESCALER (1:1 – 1:32768)
- POSTSCALER de 1:1 a 1:16 (incrementos de uno en uno)
- Si se trabaja en modo 16 bits y con carga de cuenta inicial, el POSTSCALER debe de ser solo de 1:1.
- Al activarse TMROIF=1 ó T0IF=1 se debe de bajar manualmente la bandera para que se pueda detectar un nuevo desborde ó evento de match (simplemente haciendo "bcf PIR3, 7"; siendo el bit 7 el TMROIF).
- En un evento de match o de desborde puede emitir interrupción al CPU si TMROIE=1, revisar capítulo 11 de la hoja técnica.

9

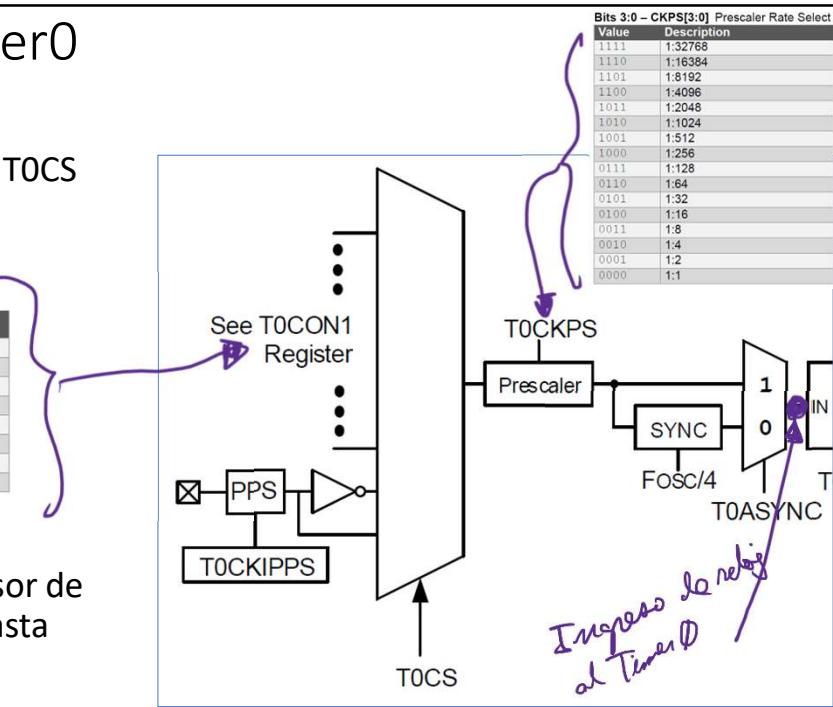
El módulo Timer0

• Fuente de reloj

- Hay que revisar los bits TOCS del registro TOCON1

Bits 7:5 – CS[2:0] Timer0 Clock Source Select

Value	Description
111	CLC1_OUT
110	SOSC
101	MFINTOSC (500 kHz)
100	LFINTOSC
011	HFINTOSC
010	Fosc/4
001	Pin selected by T0CKIPPS (Inverted)
000	Pin selected by T0CKIPPS (Non-inverted)

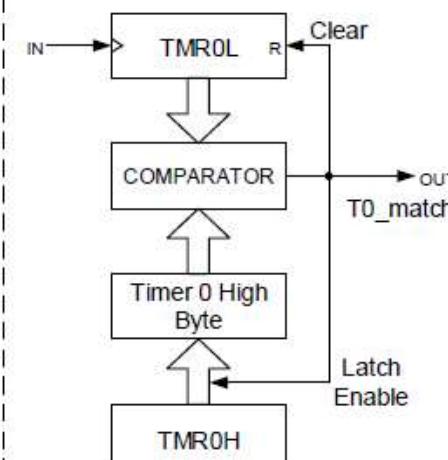


10

El módulo Timer 0:

- Modo de trabajo en **8 bits**:
 - Mejorado con respecto al Timer0 presente en el PIC18F4550 ó PIC18F45K50
 - El TMROL se usa como el registro de cuenta
 - El TMROH se utiliza como valor de comparación
 - Cuenta máxima es cuando TMROH es 255, haciendo que se mande a cero la cuenta actual, quiere decir que tendrás un rango de 0-254
 - Cuando TMROL es igual a TMROH se produce un **evento de "match"** (T0_match) el cual limpia la cuenta y actualiza el valor de comparación.
 - **Nunca se desborda**

8-bit TMRO Body Diagram (T016BIT = 0)

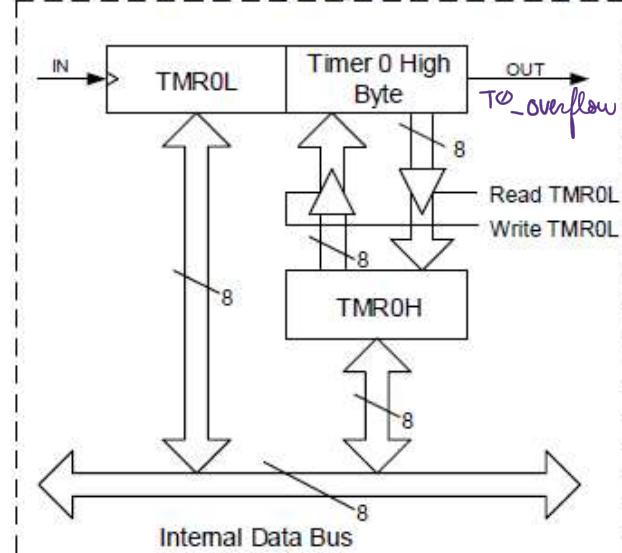


11

El módulo Timer 0:

- Modo de trabajo en **16 bits**:
- **Tener en consideración el procedimiento estricto sobre el proceso de carga de un valor en la cuenta en modo 16 bits:** Primero cargar en TMROH y luego en TMROL.
- El **evento de desborde** (T0_overflow) se produce cuando la cuenta esta en el valor mas alto (65535) y se recibe un pulso de reloj, ocasionando que la cuenta pase a 0 y levantándose la bandera de desborde (TMROIF=1 ó T0IF=1) siempre y cuando POSTSCALER 1:1

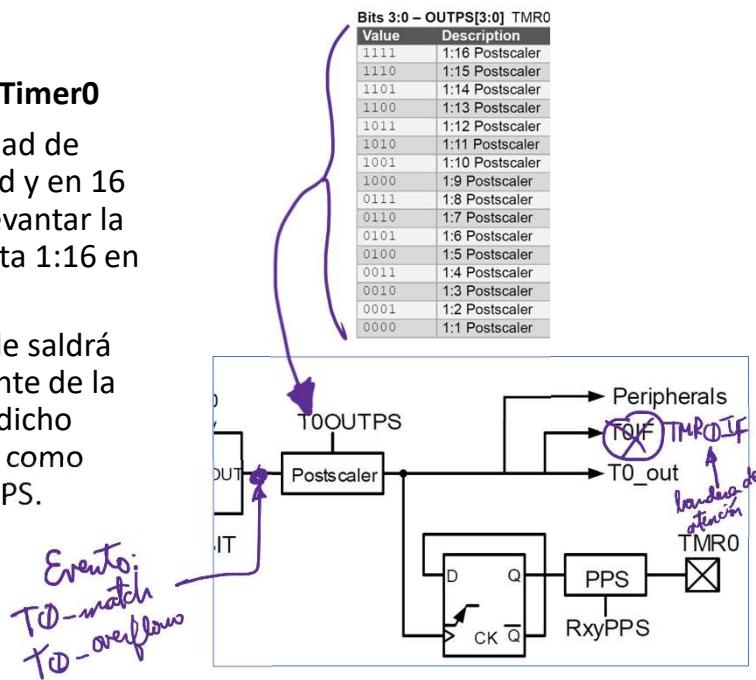
16-bit TMRO Body Diagram (T016BIT = 1)



12

El Timer0

- **El gestionador de eventos del Timer0**
- El postscaler va a contar cantidad de eventos (en 8 bits es la igualdad y en 16 bits es el desborde) antes de levantar la bandera TMROIF. Va de 1:1 hasta 1:16 en incrementos de uno en uno.
- Se tiene una salida TMRO donde saldrá una señal basculante proveniente de la salida del postscaler, para ello dicho puerto debe de configurarse como salida digital y también el RxyPPS.



13

El Timer0

- Ref. Capítulo 21 de la hoja técnica
- Salida de señal del Timer0 por un pin de E/S
- Se tiene que revisar RxyPPS para saber por cuál pin podemos sacar la señal.
- Solo se puede sacar la señal por RCx o por RFx

Table 21-2. PPS Output Selection Table

RxyPPS	Output Source	Available Output Ports											
		28-Pin Devices			40-Pin Devices			48-Pin Devices					
0x45	ADGRDB	A	—	C	A	—	C	—	—	A	—	—	F
0x44	ADGRDA	A	—	C	A	—	C	—	—	A	—	—	F
0x43	DSM1	A	—	C	A	—	—	D	—	A	—	D	—
0x42	CLKR	—	B	C	—	B	C	—	—	B	—	E	—
0x41	NCO3	—	B	C	—	B	—	E	—	B	—	E	—
0x40	NCO2	—	B	C	—	B	—	D	—	B	—	D	—
0x3F	NCO1	A	—	C	A	—	—	D	—	A	—	D	—
0x3E - 0x3A	Reserved	—	—	—	—	—	—	—	—	—	—	—	—
0x39	TMRO	—	B	C	—	B	C	—	—	C	—	—	F
0x38	I2C1 SDA ⁽¹⁾	—	B	C	—	B	C	—	—	B	C	—	—

14

El módulo Timer 0:

- Temporización máxima si HFINTOSC = 4MHz
 - Empleando Opción FOSC/4 = 1MHz ($1\mu s$) como fuente de reloj al TMRO
 - Prescaler = 1:32768 (32768 μs por cuenta)
 - Modo = 16bits (65536 cuentas)
 - En el evento de desborde: 2147483648 μs (35 minutos)
 - Postscaler = 1:16
 - Nos sale al final 34,359,738,368 μs de temporización máxima! (aprox 9.5 horas)
- Cambias el HFINTOSC a 2MHz y llegamos a 19 horas!
- Cambias el HFINTOSC a 1MHz y llegamos a 38 horas de temporizado!

15

El módulo Timer 0

- Registros implicados en la operación del Timer0:

- Registros de cuenta
 - TMROH:TMROL (16bits)
 - TMROH es valor de comparación y TMROL es el registro de cuenta (8bits)
- Registros de configuración TOCON0 y TOCON1
- Registros PIE3 (habilitadores), PIR3 (banderas) y IPR3 (prioridades) ubicando en la VIC (vectored interrupt controller cap 11)

} bank 3

Address	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0318	TMROL	7:0	TMROL[7:0]							
0x0319	TMROH	7:0	TMROH[7:0]							
0x031A	TOCON0	7:0	EN		OUT	MD16		OUTPS[3:0]		
0x031B	TOCON1	7:0		CS[2:0]		ASYNC		CKPS[3:0]		
0x04A1	PIE3	7:0	TMR0IE	CCP1IE	TMR1GIE	TMR1IE	TMR2IE	SPI1IE	SPI1TXIE	SPI1RXIE
0x04B1	PIR3	7:0	TMR0IF	CCP1IF	TMR1GIF	TMR1IF	TMR2IF	SPI1IF	SPI1TXIF	SPI1RXIF
0x0365	IPR3	7:0	TMR0IP	CCP1IP	TMR1GIP	TMR1IP	TMR2IP	SPI1IP	SPI1TXIP	SPI1RXIP

16

El módulo Timer 0

- Registro T0CON0:
 - Habilitador del módulo
 - Señal OUT
 - Modo (8 ó 16 bits)
 - Postscaler

Timer0 Control Register 0								
Bit	7	6	5	4	3	2	1	0
Access	R/W		R	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0
Bit 7 – EN TMRO Enable								
Value	Description							
1	The module is enabled and operating							
0	The module is disabled							
Bit 5 – OUT TMRO Output								
Bit 4 – MD16 16-Bit Timer Operation Select								
Value	Description							
1	TMRO is a 16-bit timer							
0	TMRO is an 8-bit timer							
Bits 3:0 – OUTPS[3:0] TMRO Output Postscaler (Divider) Select								
Value	Description							
1111	1:16 Postscaler							
1110	1:15 Postscaler							
1101	1:14 Postscaler							
1100	1:13 Postscaler							
1011	1:12 Postscaler							
1010	1:11 Postscaler							
1001	1:10 Postscaler							
1000	1:9 Postscaler							
0111	1:8 Postscaler							
0110	1:7 Postscaler							
0101	1:6 Postscaler							
0100	1:5 Postscaler							
0011	1:4 Postscaler							
0010	1:3 Postscaler							
0001	1:2 Postscaler							
0000	1:1 Postscaler							

17

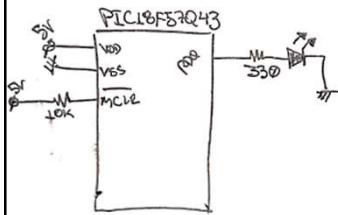
Timer0 Control Register 1								
Bit	7	6	5	4	3	2	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bits 7:5 – CS[2:0] Timer0 Clock Source Select								
Value	Description							
111	CLC1_OUT							
110	SOSC							
101	MFINTOSC (500 kHz)							
100	LFINTOSC							
011	HFINTOSC							
010	Fosc/4							
001	Pin selected by TUCKIPPS (Inverted)							
000	Pin selected by TOCKIPPS (Non-inverted)							
Bit 4 – ASYNC TMRO Input Asynchronization Enable								
Value	Description							
1	The input to the TMRO counter is not synchronized to system clocks							
0	The input to the TMRO counter is synchronized to Fosc/4							
Bits 3:0 – CKPS[3:0] Prescaler Rate Select								
Value	Description							
1111	1:32768							
1110	1:16384							
1101	1:8192							
1100	1:4096							
1011	1:2048							
1010	1:1024							
1001	1:512							
1000	1:256							
0111	1:128							
0110	1:64							
0101	1:32							
0100	1:16							
0011	1:8							
0010	1:4							
0001	1:2							
0000	1:1							

18

Ejercicios:

- Prender y apagar un LED en RDO con un periodo de 1000ms aproximadamente (TON/TOF = 500ms) empleando el Timer0 como fuente de temporización.

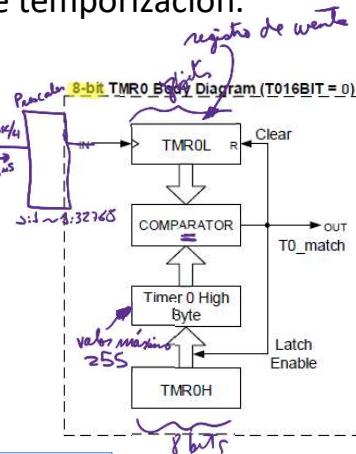
Círculo de la aplicación:



$$T = \frac{F}{m}$$

Si uso modo 8 bits y fuente de reloj para el Timer0 FOSC/4 sabiendo que FOSC=4MHz:

- Si PSC es 1:1 cada cuenta es $1\mu s$, voy a poder temporizar solo $255\mu s$ (evento match)
 - Si PSC es 1:32768 entonces cada cuenta es $32768\mu s$, la temporización máxima será 8.3 segundos aprox!
 - Si PSC es 1:8192, cada cuenta es ahora $8192\mu s$, la temporización máxima será 2 segundos aprox.
 - Si PSC es 1:2048, quiere decir que cada cuenta es $2048\mu s$, la temporización máxima será de aprox 500 ms (522.24 ms)!
 - Si valor de comparación lo reduzco a 244, obtendré 499.71 ms, mucho mas cercano al valor solicitado)



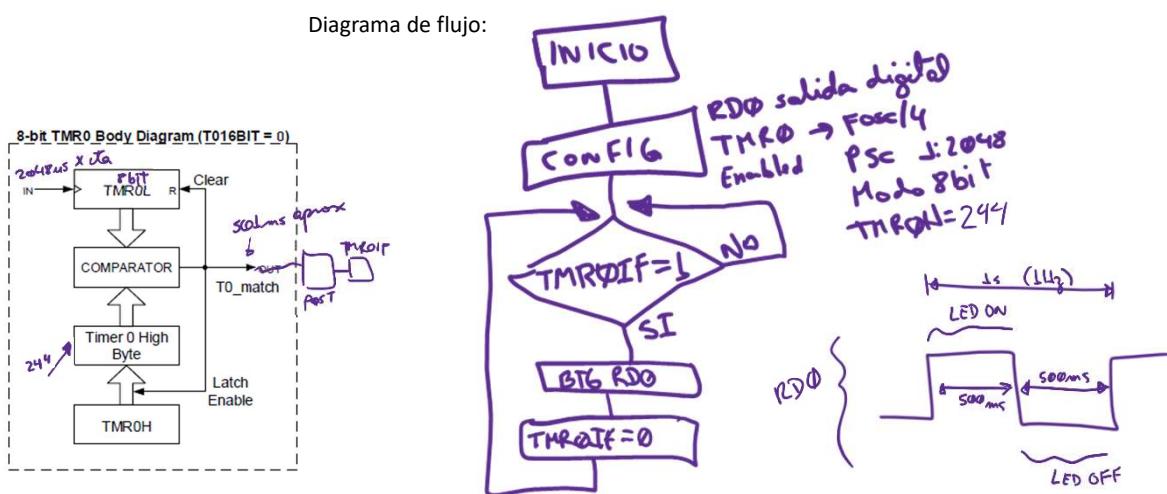
Lo que tenemos en configurar en el Timer0:
TOCON0: 80H (Timer0 enabled, POSTSCALER 1:1)
TOCON1: 4BH (FOSC/4, PRESCALER 1:2048)
TMR0H: 244

19

Ejercicios:

- Prender y apagar un LED en RDO con un periodo de 1000ms aproximadamente ($TON/TOF = 500ms$) empleando el Timer0 como fuente de temporización.

Diagrama de flujo:



20

Ejercicios:

- Prender y apagar un LED en RD0 con un periodo de 500ms aproximadamente empleando el Timer0 como fuente de temporización.

Código en XC8 PIC Assembler:

```

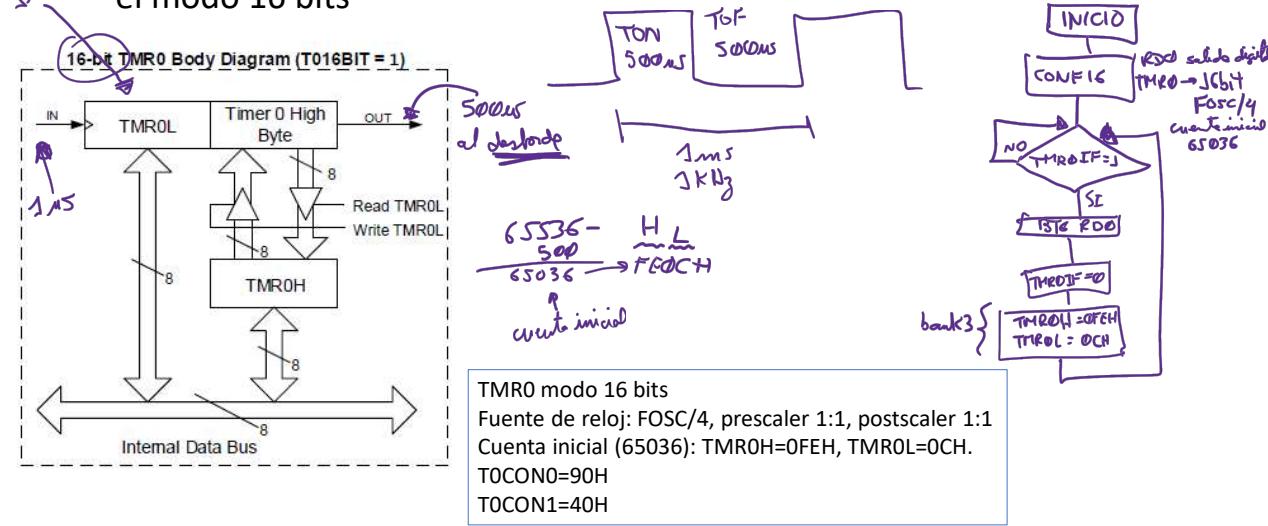
1      PROCESSOR 18F57Q43
2      #include "cabecera.inc"
3
4      PSECT upcino, class=CODE, reloc=2, abs
5      upcino:
6          ORG 000000H
7          bra configuro
8
9          ORG 000100H
10     configuro:
11         movlw 0H
12         movlw 60H
13         movwf OSCCON1, 1
14         movlw 02H
15         movwf OSCFRQ, 1
16         movlw 40H
17         movwf OSCEN, 1
18         movlb 3H
19         movlw 80H
20         movwf TOCON0, 1      ;TMR0 enabled, 8bit mode, postscaler 1:1
21         movlw 4BH
22         movwf TOCON1, 1      ;FOSC/4, async, prescaler 1:2048
23         movlw 245
24         movwf TMR0H           ;Valor de comparación 245
25         movlb 4H
26         bcf TRISD, 0, 1      ;RD0 como salida
27         bcf ANSEL0, 0, 1      ;RD0 como digital
28
29     inicio:
30         btfss PIR3, 7, 1    ;Pregunto si hubo match en TMR0, TMROIF=1
31         bra inicio
32         btg LATD, 0, 1      ;Complementamos RD0
33         bcf PIR3, 7, 1      ;Bajamos la bandera TMRO IF
34         bra inicio
35
36     end upcino

```

21

Ejercicios:

- Instante*
- Como generamos una señal cuadrada de **1KHz 50%DC** ahora usando el modo 16 bits



22

Ejercicios:

- Cómo generamos una señal cuadrada de 1KHz 50%DC ahora usando el modo 16 bits

Código en XC8 PIC Assembler:

```

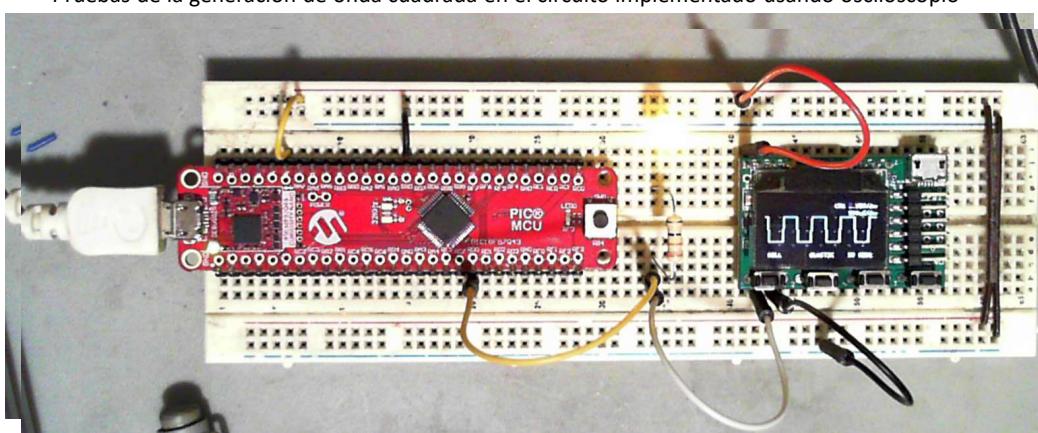
1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5  upcino:
6      ORG 000000H
7      bra configuro
8
9      ORG 000100H
10     configuro:
11         movlb 0H
12         movlw 60H
13         movwf OSCCON1, 1
14         movlw 02H
15         movwf OSCFRC, 1
16         movlw 40H
17         movwf OSCEN, 1
18         movlb 3H
19         movlw 90H
20         movwf T0CON0, 1      ;TMRO enabled, 16bit, postscaler 1:1
21         movlw 40H
22         movwf T0CON1, 1      ;FOSC/4, prescaler 1:1, async
23         movlb 4H
24         bcf TRISD, 0, 1    ;RD0 como salida
25         bcf ANSELD, 0, 1   ;RD0 como digital
26
27         inicio:
28             movlb 4H
29             btfs PIR3, 7, 1    ;Pregunto si TMROIF=1
30             bra inicio        ;Aun no se levanta TMROIF
31             btg LATD, 0, 1    ;Complemento a RD0
32             bcf PIR3, 7, 1    ;Bajamos bandera TMROIF
33             movlb 3H
34             movlw 0FEH
35             movwf TMROH, 1
36             movlw 0CH
37             movwf TMROL, 1      ;Precarga de cuenta 65036 a TMRO
38             bra inicio
39
40         end upcino

```

23

Ejercicios:

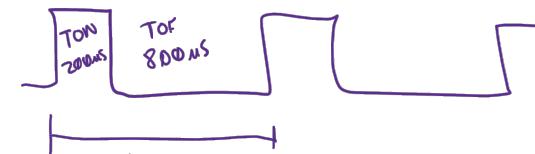
- Cómo generamos una señal cuadrada de 1KHz 50%DC ahora usando el modo 16 bits
 - Pruebas de la generación de onda cuadrada en el circuito implementado usando osciloscopio



24

Ejercicios:

- Cómo generamos una señal cuadrada de 1KHz 20%DC ahora usando el modo 16 bits



-Cuenta inicial:

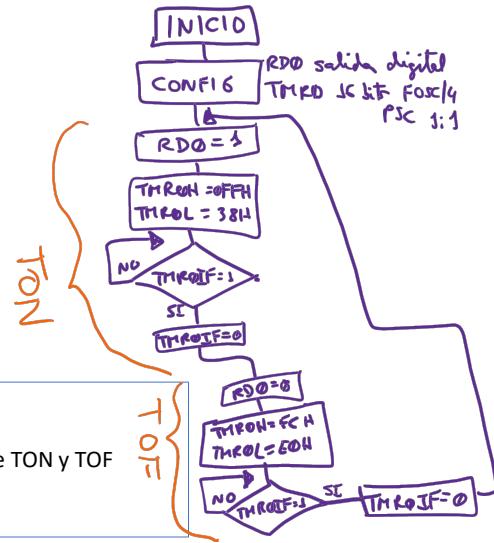
$$\begin{array}{r} \text{TON} \\ 65536 - \\ 200 \\ \hline 65336 \end{array}$$

$$\begin{array}{r} \text{TOF} \\ 65536 - \\ 800 \\ \hline 64736 \end{array}$$

$$\begin{array}{r} \text{H L} \\ 0FF38H \\ \hline \text{H L} \\ 0FCF0H \end{array}$$

RD0 salida digital
TMR0 16 bit Fosc/4
PSC 1:1

TMR0 modo 16 bits
Fuente de reloj: FOSC/4
Cuenta inicial: Depende de TON y TOF
T0CON0=90H
T0CON1=40H



25

Ejercicios:

- Cómo generamos una señal cuadrada de 1KHz 20%DC ahora usando el modo 16 bits

```

1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5  upcino:
6  ORG 000000H
7  bra configuro
8
9  ORG 000100H
10 configuro:
11  movlb 0H
12  movlw 60H
13  movwf OSCCON1, 1
14  movlw 02H
15  movwf OSCFRQ, 1
16  movlw 40H
17  movwf OSCEN, 1
18  movlb 3H
19  movlw 90H
20  movwf T0CON0, 1      ;TMRO enabled, 16bit, postscaler 1:1
21  movlw 40H
22  movwf T0CON1, 1      ;FOSC/4, prescaler 1:1, async
23  movlb 4H
24  bcf TRISD, 0, 1     ;RD0 como salida
25  bcf ANSEL0, 0, 1     ;RD0 como digital
  
```

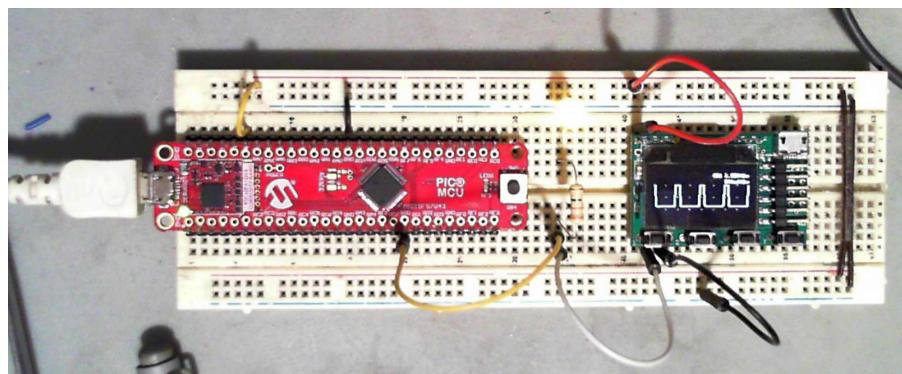
```

27  inicio:
28  movlb 4H
29  bsf LATD, 0, 1      ;RD0 a uno
30  movlb 3H
31  movlw 0FFH
32  movwf TMROH, 1
33  movlw 38H
34  movwf TMROL, 1      ;Precarga de cuenta 65036 a TMRO
35  movlb 4H
36  otro:
37  btfss PIR3, 7, 1    ;Pregunto si TMROIF=1
38  bra otro            ;Aun no se levanta TMROIF
39  bcf PIR3, 7, 1      ;Bajamos bandera TMROIF
40  bcf LATD, 0, 1      ;RD0 a cero
41  movlb 3H
42  movlw 0FCFH
43  movwf TMROH, 1
44  movlw 0E0H
45  movwf TMROL, 1      ;Precarga de cuenta 64736 a TMRO
46  movlb 4H
47  otro2:
48  btfss PIR3, 7, 1    ;Pregunto si TMROIF=1
49  bra otro2           ;Aun no se levanta TMROIF
50  bcf PIR3, 7, 1      ;Bajamos bandera TMROIF
51  bra inicio
52
53  end upcino
  
```

26

Ejercicios:

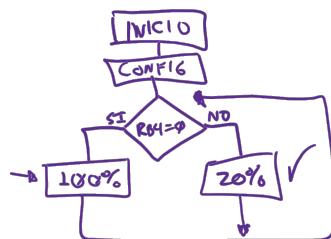
- Cómo generamos una señal cuadrada de 1KHz 20%DC ahora usando el modo 16 bits



27

¿Cómo intercambio el duty cycle de 20% a 100% mediante el pulsador en RB4?

- Preguntando al inicio por dicho botón



28

¿Cómo intercambio el duty cycle de 20% a 100% mediante el pulsador en RB4?

<pre> 1 PROCESSOR 18F57Q43 2 #include "cabecera.inc" 3 4 PSECT upcino, class=CODE, reloc=2, abs 5 upcino: 6 ORG 000000H 7 bra configuro 8 9 ORG 000080H 10 configuro: 11 movlb 0H 12 movlw 60H 13 movwf OSCCON1, 1 14 movlw 02H 15 movwf OSCFRQ, 1 16 movlw 40H 17 movwf OSCEN, 1 18 movlb 3H 19 movlw 90H 20 movwf T0CON0, 1 ;TMR0 enabled, postsc 1:1 modo 16bit 21 movlw 40H 22 movwf T0CON1, 1 ;FOSC/4 Presc 1:1 23 movlw 0FEH 24 movwf TMROH, 1 25 movlw 0CH 26 movwf TMROL, 1 ;cuenta inicial de 65036 al arranque 27 movlb 4H 28 bcf TRISD, 0, 1 ;RD0 como salida 29 bcf ANSEL0, 0, 1 ;RD0 como digital 30 bsf TRISB, 4, 1 ;RB4 como entrada 31 bcf ANSELB, 4, 1 ;RB4 como digital 32 bsf WFUEB, 4, 1 ;RB4 con pullup activado </pre>	<pre> 34 inicio: 35 btfs PORTB, 4, 1 ;preguntamos si presionaste el boton en RB4 36 bra nopresione 37 bsf LATD, 0, 1 ;RD0 en uno 38 bra inicio 39 40 nopresione: 41 bsf LATD, 0, 1 ;RD0 en uno 42 movlb 3H 43 movlw 0FFH 44 movwf TMROH, 1 45 movlw 38H 46 movwf TMROL, 1 ;cuenta inicial de 65036 al arranque 47 movlb 4H 48 btfs PIR3, 7, 1 ;pregunto si TMROIF=1 49 bra \$-2 ;falso vuelvo a preguntar 50 bcf PIR3, 7, 1 ;bajo la bandera TMROIF 51 bcf LATD, 0, 1 ;RD0 en cero 52 movlb 3H 53 movlw 0FCH 54 movwf TMROH, 1 55 movlw 0E0H 56 movwf TMROL, 1 ;cuenta inicial de 65036 al arranque 57 movlb 4H 58 btfs PIR3, 7, 1 ;pregunto si TMROIF=1 59 bra \$-2 ;falso vuelvo a preguntar 60 bcf PIR3, 7, 1 ;bajo la bandera TMROIF 61 bra inicio ;retorno a inicio 62 63 end upcino </pre>
---	--

29

Observaciones

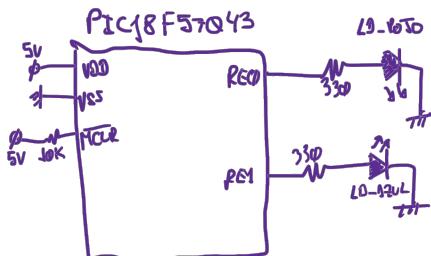
- No sale exacto en las pruebas, esto debido a que no se ha contemplado el tiempo en que se demora en ejecutar las instrucciones
- Se tiene que hacer una compensación haciendo que el TMRO cuente menos cuentas (periodo menor de temporizado) y se ajusta empleando nops
- Haciendo este tipo de compensaciones nos acercaremos a la frecuencia solicitada pero no será exacto.
- Esto nos hace pensar que no se va a poder hacer sistemas en tiempo real.

30

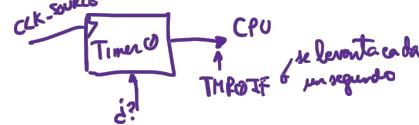
Ejercicio 2024-1

- Hacer una señal de cruce de tren empleando RE0 y RE1 como salidas hacia dos LEDs y el modulo Timer0 haciendo la base de tiempo de cambio de un segundo

a) Hardware



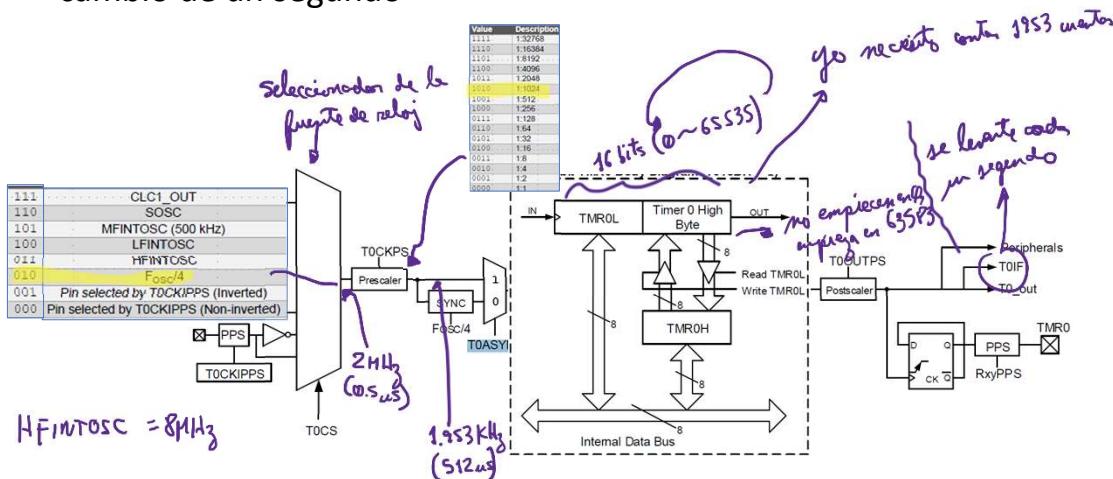
b) Análisis del Timer0 → Me avise cada 1 segundo



31

Ejercicio 2024-1

- Hacer una señal de cruce de tren empleando RE0 y RE1 como salidas hacia dos LEDs y el modulo Timer0 haciendo la base de tiempo de cambio de un segundo



32

Ejercicio 2024-1

- Hacer una señal de cruce de tren empleando RE0 y RE1 como salidas hacia dos LEDs y el modulo Timer0 haciendo la base de tiempo de cambio de un segundo

Detalles de la configuración del Timer0:

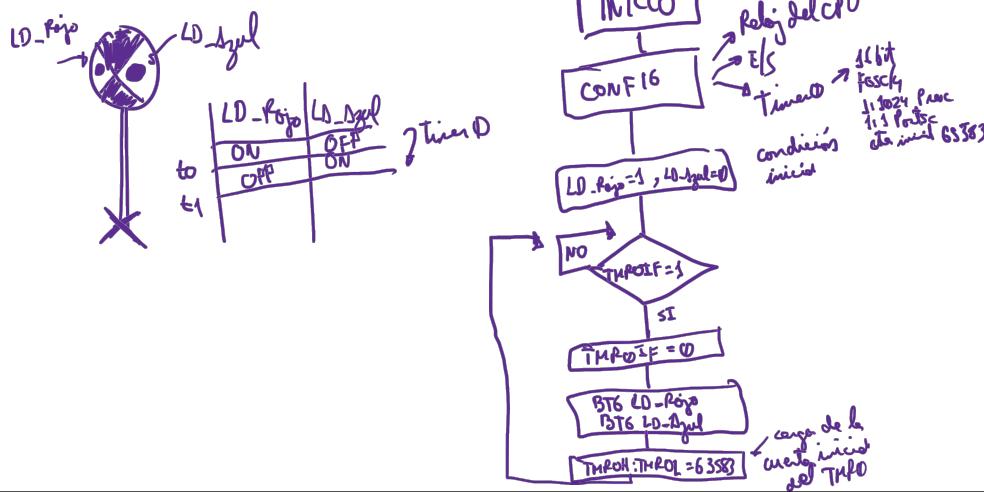
- Fuente de reloj: HFINTOSC ✓
- Modo: 16 bit ✓
- Prescaler: 1:1024 ✓
- Postscaler: 1:1 ✓
- Cuenta inicial: 63583

Name: T0CON0 Address: 0x31A							
Timer0 Control Register 0							
Bit 7 – E	0	6	5	M6	0	0	0
Access	0				0	0	0
Reset	0				0	0	0
Bit 5 – OUT Timer0 Output							90H
Bit 4 – M0:16 16-Bit Timer Operation Select							✓
Value Description							
1 TMRO is a 16-bit timer							
0 TMRO is an 8-bit timer							
Bits 3:0 – OUTPS[3:0] Timer0 Output Postscaler (Divider) Select							
Value Description							
1111 1:16 Postscaler							
1110 1:15 Postscaler							
1101 1:14 Postscaler							
1100 1:13 Postscaler							
1011 1:12 Postscaler							
1010 1:11 Postscaler							
1001 1:10 Postscaler							
1000 1:9 Postscaler							
0111 1:8 Postscaler							
0110 1:7 Postscaler							
0101 1:6 Postscaler							
0100 1:5 Postscaler							
0011 1:4 Postscaler							
0110 1:3 Postscaler							
0001 1:2 Postscaler							
0000 1:1 Postscaler							
Name: T0CON1 Address: 0x31B							
Timer0 Control Register 1							
Bit 7 – CS[2:0] Timer0 Clock Source Select	0	CPU	0	CKPSD	1	0	0
Access	0				✓	0	0
Reset	0				0	0	0
Bit 5 – OUT Timer0 Output							4AH
Bit 4 – ASYNC_TMR0 Input Asynchronization Enable							
Value Description							
1 The input to the TMRO counter is not synchronized to system clocks							
0 The input to the TMRO counter is synchronized to Fosc/4							
Bits 3:0 – CKPS[3:0] Prescaler Rate Selected							
Value Description							
1111 1:32768							
1110 1:65536							
1101 1:16384							
1100 1:4096							
1010 1:1024							
1001 1:512							
1000 1:256							
0111 1:128							
0110 1:64							
0101 1:32							
0100 1:16							
0011 1:8							
0010 1:4							
0001 1:2							
0000 1:1							

33

Ejercicio 2024-1

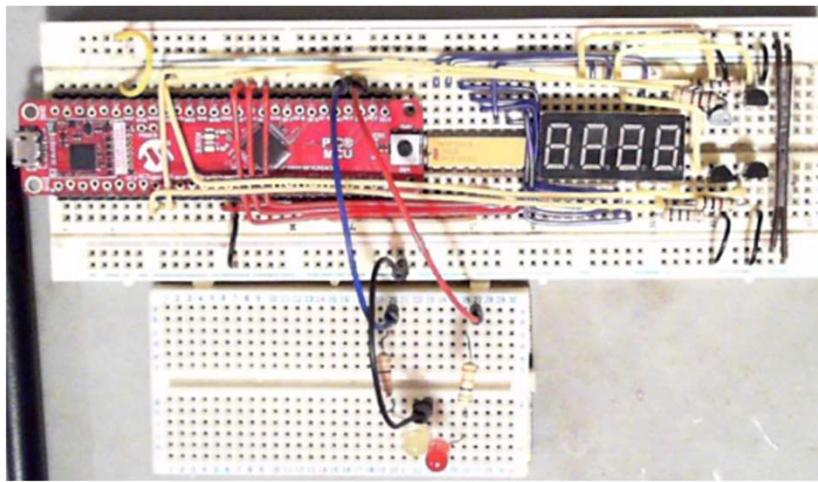
- Hacer una señal de cruce de tren empleando RE0 y RE1 como salidas hacia dos LEDs y el modulo Timer0 haciendo la base de tiempo de cambio de un segundo



34

Ejercicio 2024-1

- Hacer una señal de cruce de tren empleando RE0 y RE1 como salidas hacia dos LEDs y el modulo Timer0 haciendo la base de tiempo de cambio de un segundo



35

Ejercicio 2024-1

```

1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5  upcino:
6  ORG 000000H
7  bra configuro
8
9  ORG 000100H
10 configuro:
11  movlb 0H
12  movlw 60H
13  movwf OSCCON1, 1
14  movlw 03H
15  movwf OSCFRC, 1
16  movlw 40H
17  movwf OSCEN, 1      ;HFINTOSC a 8MHz
18  movlb 3H
19  movlw 90H
20  movwf T0CON0, 1
21  movlw 4AH           ;TMRO 16bit,psc 1:1024, post 1:1
22  movwf T0CON1, 1      ;TMRO HFINTOSC/4
23  movlw 0F8H
24  movwf TMROH, 1
25  movlw 5FH
26  movwf TMROL, 1      ;cuenta inicial de 63583
27  movlb 4H
28  bcf TRISE, 0, 1     ;RE0 como salida
29  bcf TRISE, 1, 1     ;RE1 como salida
30  bcf ANSEL0, 0, 1     ;RE0 como digital
31  bcf ANSEL0, 1, 1     ;RE1 como digital
32  bsf LATE, 0, 1       ;LD_Rojo encendido
33  bsf LATE, 1, 1       ;LD_Azul apagado

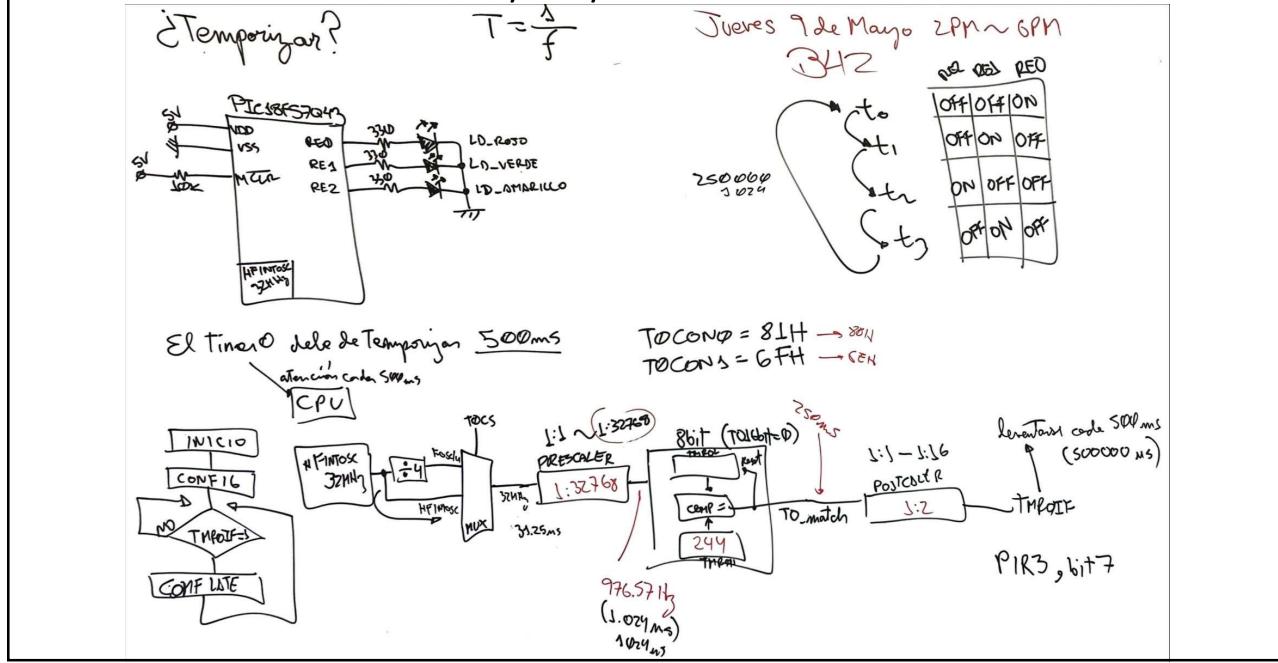
```

- Hacer una señal de cruce de tren empleando RE0 y RE1 como salidas hacia dos LEDs y el modulo Timer0 haciendo la base de tiempo de cambio de un segundo

35	inicio:	
36	btfss PIR3, 7, 1	;Pregunto si se levanto bandera de TMRO
37	bra \$-2	;Falso, vuelve a preguntar
38	bcf PIR3, 7, 1	;Verdad, bajo la bandera TMRO
39	btg LATE, 0, 1	;Basculo RE0
40	btg LATE, 1, 1	;Basculo RE1
41	movlb 3H	
42	movlw 0F8H	
43	movwf TMROH, 1	
44	movlw 5FH	
45	movwf TMROL, 1	;cuenta inicial de 63583
46	movlb 4H	
47	bra inicio	
48	end upcino	

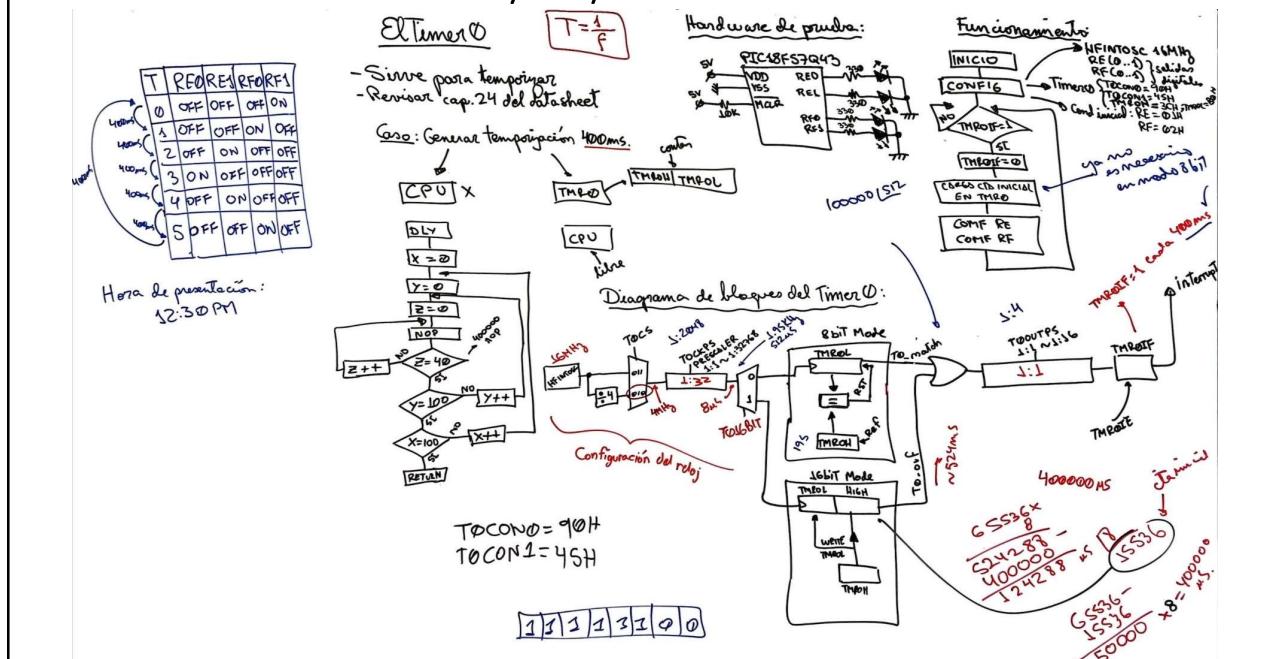
36

Pizarra EL54-2 25/04/2024



37

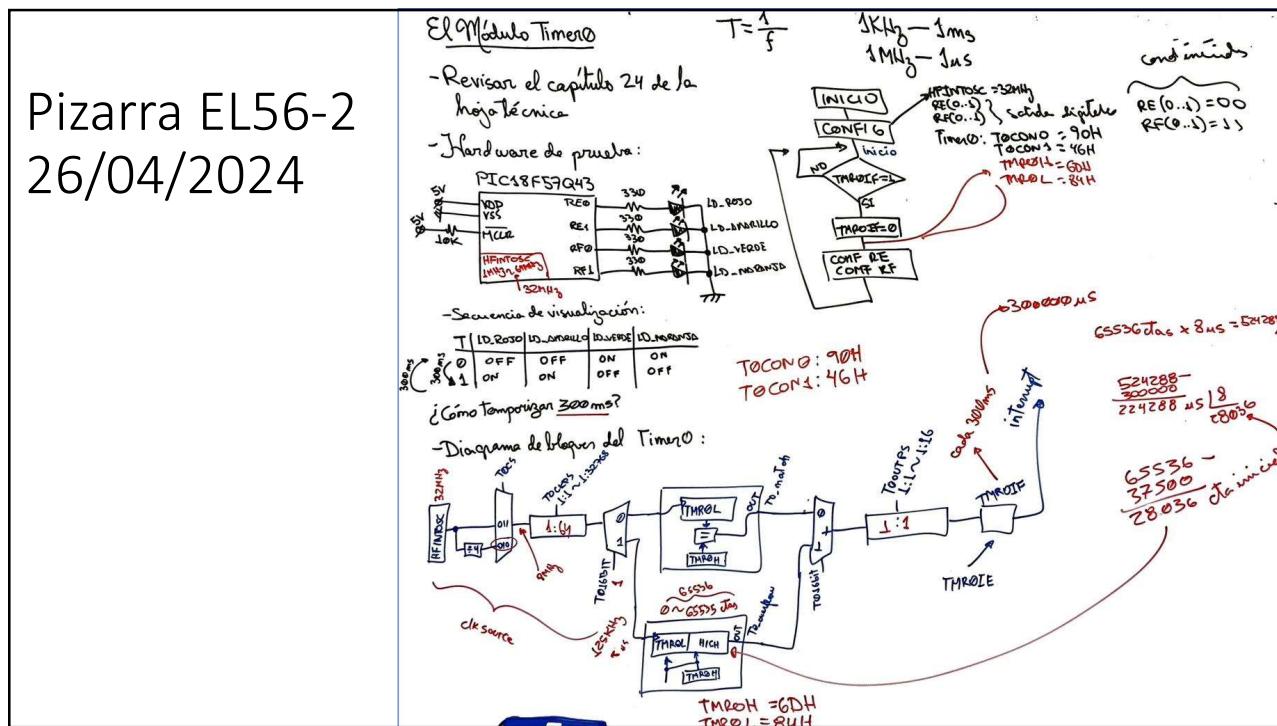
Pizarra EL56-1 26/04/2024



38

Pizarra EL56-2

26/04/2024



39

```

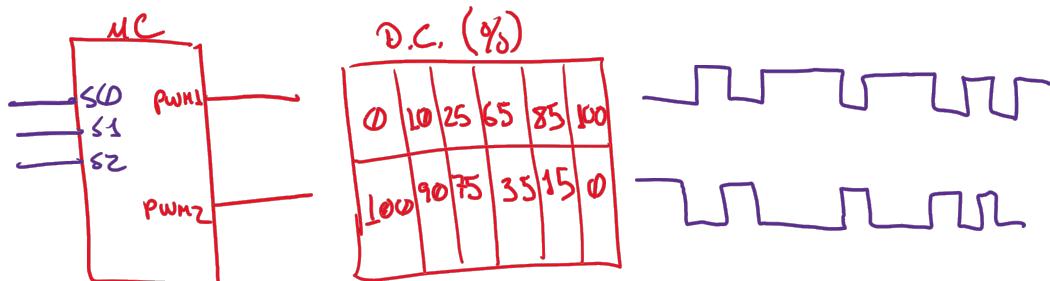
1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5
6  TAQUIS EQU 500H
7
8  upcino:
9    ORG 000000H
10   bra configuro
11
12   ORG 000700H
13   tabla1: DB 01H, 02H, 00H, 00H, 02H, 01H, 02H, 01H, 02H, 00H, 03H, 00H, 03H, 00H
14
15   ORG 000800H
16   tabla2: DB 00H, 00H, 01H, 02H, 01H, 00H, 02H, 01H, 02H, 01H, 00H, 03H, 00H, 03H, 00H
17
18   ORG 000100H
19  configuro:
20   movlb 0H
21   movlw 60H
22   movwf OSCCON1, 1
23   movlw 05H
24   movwf OSCFRQ, 1
25   movlw 40H
26   movwf OSCEN, 1      ;HFINTOSC a 32MHz
27   movlb 3H
28   movlw 90H
29   movwf TOCON0, 1
30   movlw 44H
31   movwf TOCON1, 1      ;TMR0 16bit, FOSC/4, PRESC 1:64, POSTS 1:1
32   movlw 6DH
33   movwf TMR0H, 1
34   movlw 84H
35   movwf TMR0L, 1      ;cuenta inicial 28036 (6D84H)
36   movlb 5H
37   clrf TAQUIS, 1
38   movlb 4H
39   movlw 0FCH
40   movwf TRISE, 1
41   movwf ANSEL, 1      ;RE(0..1) salidas digitales
42
43   movwf TRISF, 1
44   movwf ANSELE, 1      ;RF(0..1) salidas digitales
45   movlw 01H
46   movwf LATE, 1
47   movlw 02H
48   movwf LATF, 1      ;condiciones iniciales
49   clrf TBLPTRU, 1
50   movlw 07H
51   movwf TBLPTRH, 1
52   clrf TBLPTRL, 1
53
54   inicio:
55   btfss PIR3, 7, 1
56   bra S-2
57   btfss PIR3, 7, 1
58   movlw 60H
59   movwf TMR0H, 1
60   movlw 84H
61   movwf TMR0L, 1      ;cuenta inicial 28036 (6D84H)
62   movlw 07H
63   movwf TBLPTRH, 1
64   movlb 3H
65   movlw 14
66   cpfseq TAQUIS, 1
67   bra aunno
68   clrf TAQUIS, 1
69   bra siguiente
70
71   aunno:
72   incf TAQUIS, 1, 1
73   siguiente:
74   movf TAQUIS, 0, 1
75   movlb 4H
76   movwf TBLPTRL, 1
77   TBLRD+
78   movff TABLAT, LATF
79   movlw 08H
80   movwf TBLPTRH, 1
81   TBLRD+
82   movff TABLAT, LATF
83
84   end upcino

```

40

Ejercicios propuestos

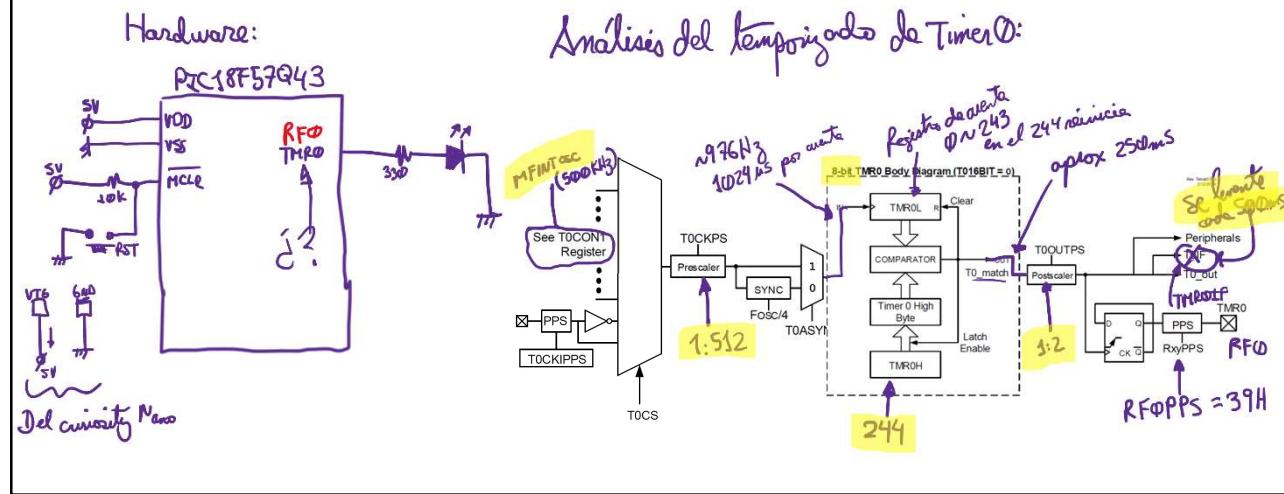
- En un PoR. ¿En qué estado se encuentra el Timer0, encendido o apagado?
- Si Fosc = 24MHz. ¿Cuál es la temporización máxima del Timer0 en modo 16 bits?
- Desarrollar un generador de PWM 2KHz con dos salidas complementarias y con opciones de dutycycle siguientes: 0%, 10%, 25%, 65%, 85% y 100%



41

Ejercicio 2024-2

- Realizar el parpadeo de un LED con TON y TOF de 500ms (el periodo total es de 1 segundo) aproximadamente y empleando el Timer0 en modo 8 bits, el LED estará conectado en el puerto RF0 y éste será la salida del TMRO del PIC18F57Q43.



42

Ejercicio 2024-2

<p>Name: T0CON0 → 84H Address: 0x31A</p> <p>Timer0 Control Register 0</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Bit</td> <td style="text-align: center; padding: 2px;">7</td> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">1</td> </tr> <tr> <td style="text-align: center; padding: 2px;">Access</td> <td style="text-align: center; padding: 2px;">R</td> <td style="text-align: center; padding: 2px;">W</td> <td style="text-align: center; padding: 2px;">R</td> <td style="text-align: center; padding: 2px;">W</td> <td style="text-align: center; padding: 2px;">R</td> <td style="text-align: center; padding: 2px;">W</td> <td style="text-align: center; padding: 2px;">R</td> <td style="text-align: center; padding: 2px;">W</td> </tr> <tr> <td style="text-align: center; padding: 2px;">Reset</td> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">1</td> </tr> </table> <p>Bit 7 – EN TMRO Enable <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Value</td> <td style="text-align: center; padding: 2px;">Description</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1</td> <td style="text-align: center; padding: 2px;">The module is enabled and operating ✓</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">The module is disabled</td> </tr> </table> </p> <p>Bit 5 – OUT TMRO Output</p> <p>Bit 4 – MD16 16-Bit Timer Operation Select <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Value</td> <td style="text-align: center; padding: 2px;">Description</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1</td> <td style="text-align: center; padding: 2px;">TMRO is a 16-bit timer ✓</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">TMRO is an 8-bit timer</td> </tr> </table> </p> <p>Bits 3:0 – OUTPS[3:0] TMRO Output Postscaler (Divider) Select <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Value</td> <td style="text-align: center; padding: 2px;">Description</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1111</td> <td style="text-align: center; padding: 2px;">1:16 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1110</td> <td style="text-align: center; padding: 2px;">1:15 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1101</td> <td style="text-align: center; padding: 2px;">1:14 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1100</td> <td style="text-align: center; padding: 2px;">1:13 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1011</td> <td style="text-align: center; padding: 2px;">1:12 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1010</td> <td style="text-align: center; padding: 2px;">1:11 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1001</td> <td style="text-align: center; padding: 2px;">1:10 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1000</td> <td style="text-align: center; padding: 2px;">1:9 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0111</td> <td style="text-align: center; padding: 2px;">1:8 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0110</td> <td style="text-align: center; padding: 2px;">1:7 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0101</td> <td style="text-align: center; padding: 2px;">1:6 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0100</td> <td style="text-align: center; padding: 2px;">1:5 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0011</td> <td style="text-align: center; padding: 2px;">1:4 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0010</td> <td style="text-align: center; padding: 2px;">1:3 Postscaler</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0001</td> <td style="text-align: center; padding: 2px;">1:2 Postscaler ✓</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0000</td> <td style="text-align: center; padding: 2px;">1:1 Postscaler</td> </tr> </table> </p>	Bit	7	0	0	0	0	0	0	1	Access	R	W	R	W	R	W	R	W	Reset	0	0	0	0	0	0	0	1	Value	Description	1	The module is enabled and operating ✓	0	The module is disabled	Value	Description	1	TMRO is a 16-bit timer ✓	0	TMRO is an 8-bit timer	Value	Description	1111	1:16 Postscaler	1110	1:15 Postscaler	1101	1:14 Postscaler	1100	1:13 Postscaler	1011	1:12 Postscaler	1010	1:11 Postscaler	1001	1:10 Postscaler	1000	1:9 Postscaler	0111	1:8 Postscaler	0110	1:7 Postscaler	0101	1:6 Postscaler	0100	1:5 Postscaler	0011	1:4 Postscaler	0010	1:3 Postscaler	0001	1:2 Postscaler ✓	0000	1:1 Postscaler	<p>Name: T0CON1 → A9H Address: 0x31B</p> <p>Timer0 Control Register 1</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Bit</td> <td style="text-align: center; padding: 2px;">7</td> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">1</td> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">1</td> </tr> <tr> <td style="text-align: center; padding: 2px;">Access</td> <td style="text-align: center; padding: 2px;">R</td> <td style="text-align: center; padding: 2px;">W</td> <td style="text-align: center; padding: 2px;">R</td> <td style="text-align: center; padding: 2px;">W</td> <td style="text-align: center; padding: 2px;">R</td> <td style="text-align: center; padding: 2px;">W</td> <td style="text-align: center; padding: 2px;">R</td> <td style="text-align: center; padding: 2px;">W</td> </tr> <tr> <td style="text-align: center; padding: 2px;">Reset</td> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">1</td> </tr> </table> <p>Bits 7:5 – CS[2:0] Timer0 Clock Source Select <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Value</td> <td style="text-align: center; padding: 2px;">Description</td> </tr> <tr> <td style="text-align: center; padding: 2px;">111</td> <td style="text-align: center; padding: 2px;">CLC1_OUT</td> </tr> <tr> <td style="text-align: center; padding: 2px;">110</td> <td style="text-align: center; padding: 2px;">OSC0</td> </tr> <tr> <td style="text-align: center; padding: 2px;">101</td> <td style="text-align: center; padding: 2px;">MFINTOSC (600 kHz)</td> </tr> <tr> <td style="text-align: center; padding: 2px;">100</td> <td style="text-align: center; padding: 2px;">LFINTOSC</td> </tr> <tr> <td style="text-align: center; padding: 2px;">011</td> <td style="text-align: center; padding: 2px;">HFINTOSC</td> </tr> <tr> <td style="text-align: center; padding: 2px;">010</td> <td style="text-align: center; padding: 2px;">Fosc/4</td> </tr> <tr> <td style="text-align: center; padding: 2px;">001</td> <td style="text-align: center; padding: 2px;">Pin selected by TOCKIPPS (Inverted)</td> </tr> <tr> <td style="text-align: center; padding: 2px;">000</td> <td style="text-align: center; padding: 2px;">Pin selected by TOCKIPPS (Non-inverted)</td> </tr> </table> </p> <p>Bit 4 – ASYNC TMRO Input Asynchronization Enable <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Value</td> <td style="text-align: center; padding: 2px;">Description</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1</td> <td style="text-align: center; padding: 2px;">The input to the TMRO counter is not synchronized to system clocks ✓</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0</td> <td style="text-align: center; padding: 2px;">The input to the TMRO counter is synchronized to Fosc/4</td> </tr> </table> </p> <p>Bits 3:0 – CKPS[3:0] Prescaler Rate Select <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Value</td> <td style="text-align: center; padding: 2px;">Description</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1111</td> <td style="text-align: center; padding: 2px;">1:32768</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1110</td> <td style="text-align: center; padding: 2px;">1:16384</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1101</td> <td style="text-align: center; padding: 2px;">1:8192</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1100</td> <td style="text-align: center; padding: 2px;">1:4096</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1011</td> <td style="text-align: center; padding: 2px;">1:2048</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1010</td> <td style="text-align: center; padding: 2px;">1:1024 ✓</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1001</td> <td style="text-align: center; padding: 2px;">1:512</td> </tr> <tr> <td style="text-align: center; padding: 2px;">1000</td> <td style="text-align: center; padding: 2px;">1:256</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0111</td> <td style="text-align: center; padding: 2px;">1:128</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0110</td> <td style="text-align: center; padding: 2px;">1:64</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0101</td> <td style="text-align: center; padding: 2px;">1:32</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0100</td> <td style="text-align: center; padding: 2px;">1:16</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0011</td> <td style="text-align: center; padding: 2px;">1:8</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0010</td> <td style="text-align: center; padding: 2px;">1:4</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0001</td> <td style="text-align: center; padding: 2px;">1:2</td> </tr> <tr> <td style="text-align: center; padding: 2px;">0000</td> <td style="text-align: center; padding: 2px;">1:1</td> </tr> </table> </p>	Bit	7	0	0	0	1	0	0	1	Access	R	W	R	W	R	W	R	W	Reset	0	0	0	0	0	0	0	1	Value	Description	111	CLC1_OUT	110	OSC0	101	MFINTOSC (600 kHz)	100	LFINTOSC	011	HFINTOSC	010	Fosc/4	001	Pin selected by TOCKIPPS (Inverted)	000	Pin selected by TOCKIPPS (Non-inverted)	Value	Description	1	The input to the TMRO counter is not synchronized to system clocks ✓	0	The input to the TMRO counter is synchronized to Fosc/4	Value	Description	1111	1:32768	1110	1:16384	1101	1:8192	1100	1:4096	1011	1:2048	1010	1:1024 ✓	1001	1:512	1000	1:256	0111	1:128	0110	1:64	0101	1:32	0100	1:16	0011	1:8	0010	1:4	0001	1:2	0000	1:1
Bit	7	0	0	0	0	0	0	1																																																																																																																																																							
Access	R	W	R	W	R	W	R	W																																																																																																																																																							
Reset	0	0	0	0	0	0	0	1																																																																																																																																																							
Value	Description																																																																																																																																																														
1	The module is enabled and operating ✓																																																																																																																																																														
0	The module is disabled																																																																																																																																																														
Value	Description																																																																																																																																																														
1	TMRO is a 16-bit timer ✓																																																																																																																																																														
0	TMRO is an 8-bit timer																																																																																																																																																														
Value	Description																																																																																																																																																														
1111	1:16 Postscaler																																																																																																																																																														
1110	1:15 Postscaler																																																																																																																																																														
1101	1:14 Postscaler																																																																																																																																																														
1100	1:13 Postscaler																																																																																																																																																														
1011	1:12 Postscaler																																																																																																																																																														
1010	1:11 Postscaler																																																																																																																																																														
1001	1:10 Postscaler																																																																																																																																																														
1000	1:9 Postscaler																																																																																																																																																														
0111	1:8 Postscaler																																																																																																																																																														
0110	1:7 Postscaler																																																																																																																																																														
0101	1:6 Postscaler																																																																																																																																																														
0100	1:5 Postscaler																																																																																																																																																														
0011	1:4 Postscaler																																																																																																																																																														
0010	1:3 Postscaler																																																																																																																																																														
0001	1:2 Postscaler ✓																																																																																																																																																														
0000	1:1 Postscaler																																																																																																																																																														
Bit	7	0	0	0	1	0	0	1																																																																																																																																																							
Access	R	W	R	W	R	W	R	W																																																																																																																																																							
Reset	0	0	0	0	0	0	0	1																																																																																																																																																							
Value	Description																																																																																																																																																														
111	CLC1_OUT																																																																																																																																																														
110	OSC0																																																																																																																																																														
101	MFINTOSC (600 kHz)																																																																																																																																																														
100	LFINTOSC																																																																																																																																																														
011	HFINTOSC																																																																																																																																																														
010	Fosc/4																																																																																																																																																														
001	Pin selected by TOCKIPPS (Inverted)																																																																																																																																																														
000	Pin selected by TOCKIPPS (Non-inverted)																																																																																																																																																														
Value	Description																																																																																																																																																														
1	The input to the TMRO counter is not synchronized to system clocks ✓																																																																																																																																																														
0	The input to the TMRO counter is synchronized to Fosc/4																																																																																																																																																														
Value	Description																																																																																																																																																														
1111	1:32768																																																																																																																																																														
1110	1:16384																																																																																																																																																														
1101	1:8192																																																																																																																																																														
1100	1:4096																																																																																																																																																														
1011	1:2048																																																																																																																																																														
1010	1:1024 ✓																																																																																																																																																														
1001	1:512																																																																																																																																																														
1000	1:256																																																																																																																																																														
0111	1:128																																																																																																																																																														
0110	1:64																																																																																																																																																														
0101	1:32																																																																																																																																																														
0100	1:16																																																																																																																																																														
0011	1:8																																																																																																																																																														
0010	1:4																																																																																																																																																														
0001	1:2																																																																																																																																																														
0000	1:1																																																																																																																																																														

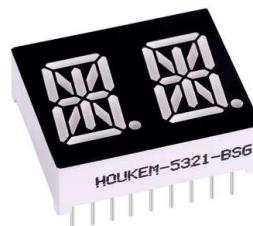
43

44

<pre> 1 PROCESSOR 18F57Q43 2 #include "cabecera.inc" 3 4 PSECT upcino, class=CODE, reloc=2, abs 5 upcino: 6 ORG 000000H 7 bra configuro 8 9 ORG 000100H 10 configuro: 11 ;configurar la fuente de reloj 12 movlb 0H 13 movlw 60H 14 movwf OSCCON1, 1 15 movlw 02H 16 movwf OSCFRQ, 1 17 movlw 40H 18 movwf OSCEN, 1 19 ;configuro el PPS 20 movlb 2H 21 movlw 39H 22 movwf RFOPPS, 1 ;el RF0 conectado a TMRO 23 ;configuro el Timer0 24 movlb 3H 25 movlw 244 26 movwf TMROH, 1 ;valor de ref de comparador 244 27 movlw 81H 28 movwf T0CON0, 1 ;TMRO enabled, 8bit, POSTS 1:2 29 movlw 0A9H 30 movwf T0CON1, 1 ;MFINTOSC, SYNC, PRESC 1:512 </pre>	<pre> 31 ;configurar las E/S 32 movlb 4H 33 bcf TRISF, 0, 1 ;RF0 es salida 34 bcf ANSELF, 0, 1 ;RF0 es digital 35 36 inicio: 37 bra inicio 38 ;el CPU no hace nada despues de la configuración! 39 end upcino </pre>
---	---

Ejercicios propuestos

- Conectar el microcontrolador PIC18F57Q43 con el siguiente dispositivo (display de dos dígitos de 14 segmentos multiplexados de ánodo común):



45

Fin de la sesión

46