

# Microcontroladores

## Semana 6

Semestre 2024-1

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.

2

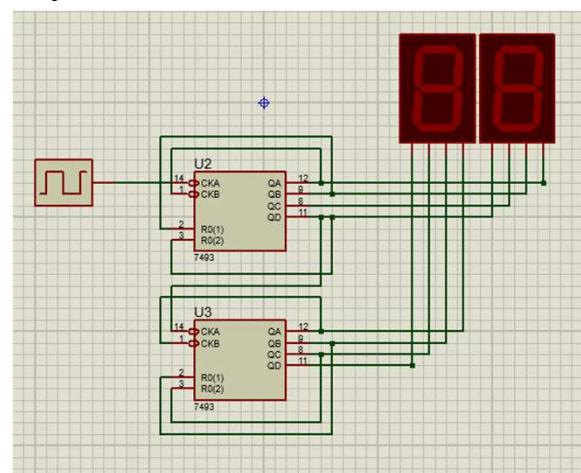
## Agenda:

- El módulo Timer0
  - Modo 8bit
  - Modo 16bit
  - Registros de configuración y operación
- Cálculo de temporización
- Generación de ondas cuadradas periódicas

3

## La previa: contadores digitales

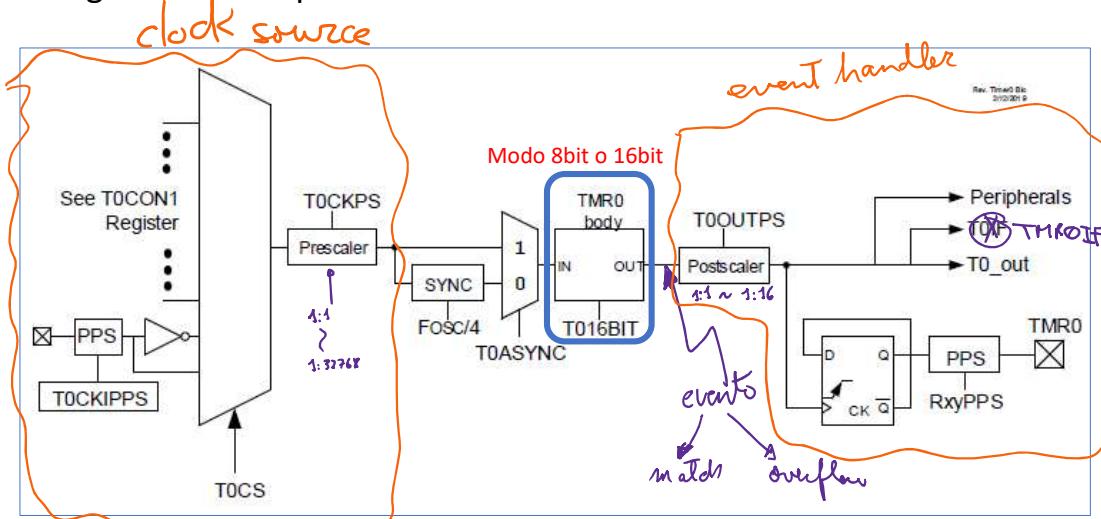
- Sistemas secuenciales, poseen reloj, un registro de cuenta, dependiendo de su función, puede incrementar o puede decrementar la cuenta según la entrada de reloj.



4

## El módulo Timer 0:

- Diagrama de bloques



5

## El módulo Timer 0

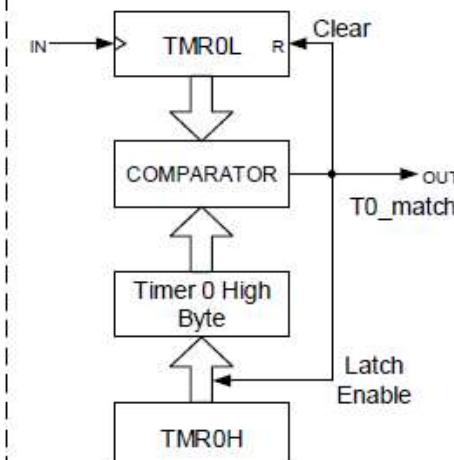
- (Ref. Item 24 de la hoja técnica del microcontrolador PIC18F57Q43)
- Temporizador de cuenta ascendente
- Modo de trabajo: 8 bits (0-254) ó 16 bits (0-65535)
- Las cuentas del Timer0 se alojan en:
  - TMROH:TMROL (16 bits)
  - TMROL (8 bits)
- Diversas fuentes de reloj (revisar T0CON1 y TOCS)
- Divisor 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.

6

## 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)

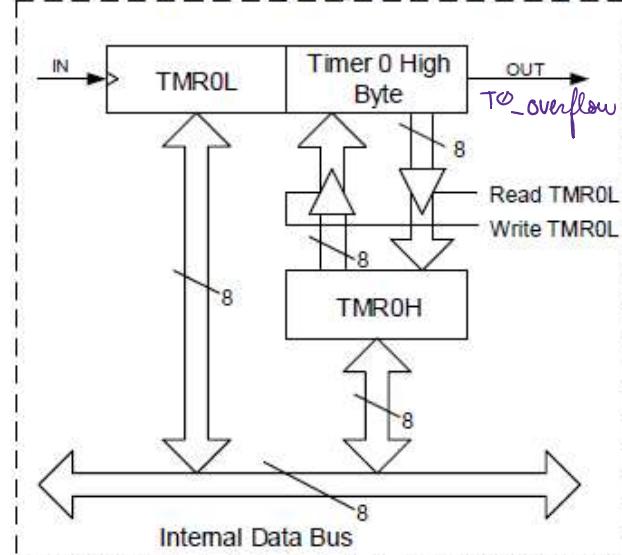


7

## 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 está en el valor más 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)



8

## 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  $\mu s$  por cuenta)
  - Modo = 16bits (65536 cuentas)
  - En el evento de desborde: 2147483648  $\mu s$  (35 minutos)
  - Postscaler = 1:16
  - Nos sale al final 34,359,738,368  $\mu s$  de temporización máxima! (aprox 9.5 horas)

9

## 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

10

## 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
<b>Bit 7 – EN</b> TMR0 Enable								
<b>Value</b>	<b>Description</b>							
1	The module is enabled and operating							
0	The module is disabled							
<b>Bit 5 – OUT</b> TMR0 Output								
<b>Bit 4 – MD16</b> 16-Bit Timer Operation Select								
<b>Value</b>	<b>Description</b>							
1	TMR0 is a 16-bit timer							
0	TMR0 is an 8-bit timer							
<b>Bits 3:0 – OUTPS[3:0]</b> TMR0 Output Postscaler (Divider) Select								
<b>Value</b>	<b>Description</b>							
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							

11

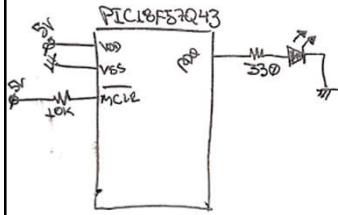
Timer0 Control Register 1								
Bit	7	6	5	4	3	2	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
<b>Bits 7:5 – CS[2:0]</b> Timer0 Clock Source Select								
<b>Value</b>	<b>Description</b>							
111	CLC1_OUT							
110	SOSC							
101	MFINTOSC (500 kHz)							
100	LFINTOSC							
011	HFINTOSC							
010	Fosc/4							
001	Pin selected by TUCKIPPS (Inverted)							
000	Pin selected by TOCKIPPS (Non-inverted)							
<b>Bit 4 – ASYNC</b> TMR0 Input Asynchronization Enable								
<b>Value</b>	<b>Description</b>							
1	The input to the TMR0 counter is not synchronized to system clocks							
0	The input to the TMR0 counter is synchronized to Fosc/4							
<b>Bits 3:0 – CKPS[3:0]</b> Prescaler Rate Select								
<b>Value</b>	<b>Description</b>							
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							

12

## Ejercicios:

- Prender y apagar un LED en RD0 con un periodo de 1000ms aproximadamente (TON/TOF = 500ms) empleando el Timer0 como fuente de temporización.

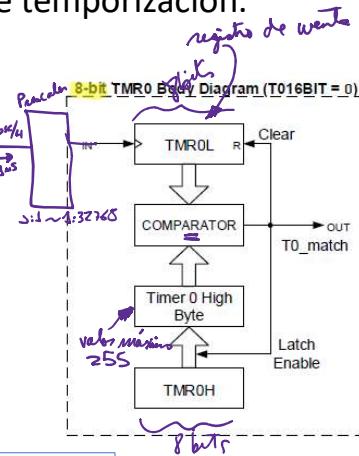
Circuito de la aplicación:



$$T = \frac{1}{f}$$

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 1μs, voy a poder temporizar solo 255μs (evento match)
- Si PSC es 1:32768 entonces cada cuenta es 32768μs, la temporización máxima será 8.3 segundos aprox!
- Si PSC es 1:8192, cada cuenta es ahora 8192μs, la temporización máxima será 2 segundos aprox.
- Si PSC es 1:2048, quiere decir que cada cuenta es de 2048μs, 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)



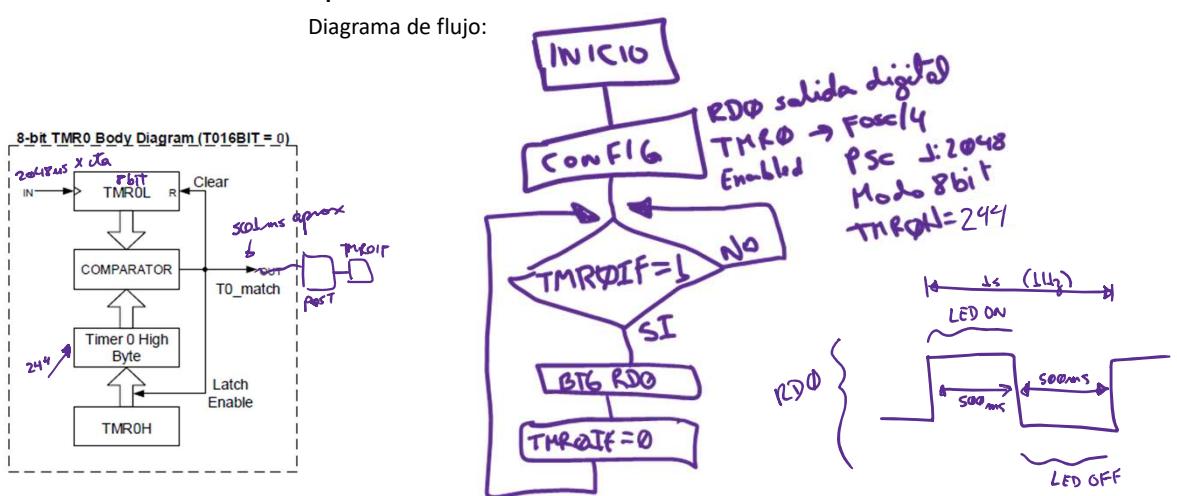
Lo que tenemos en configurar en el Timer0:  
 TOCON0: 80H (Timer0 enabled, POSTSCALER 1:1)  
 TOCON1: 4BH (FOSC/4, PRESCALER 1:2048)  
 TMROH: 244

13

## Ejercicios:

- Prender y apagar un LED en RD0 con un periodo de 1000ms aproximadamente (TON/TOF = 500ms) empleando el Timer0 como fuente de temporización.

Diagrama de flujo:



14

## 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:

```

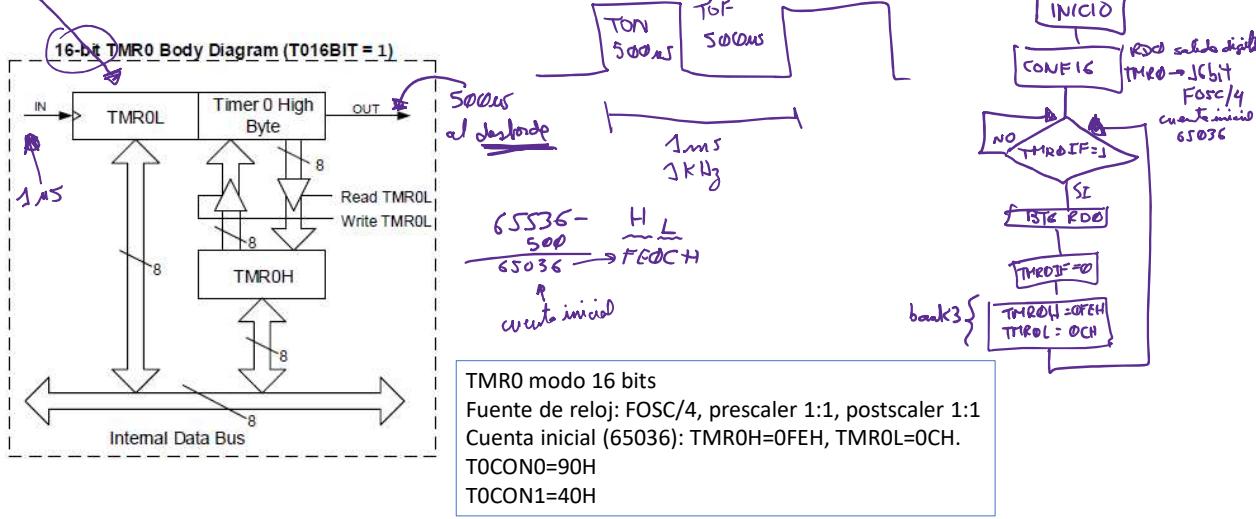
1      PROCESSOR 18F57Q43
2      #include "cabecera.inc"
3
4      PSECT upcino, class=CODE, reloc=2, abs
5      upcino:
6          ORG 000000H
7          bra configuro
8
9          ORG 000100H
10         configuro:
11             movlw 0H
12             moviw 60H
13             movwf OSCCON1, 1
14             movlw 02H
15             movwf OSCFRQ, 1
16             movlw 40H
17             movwf OSCEN, 1
18             movlb 3H
19             moviw 80H
20             movwf T0CON0, 1      ;TMR0 enabled, 8bit mode, postscaler 1:1
21             movlw 4BH
22             movwf T0CON1, 1      ;FOSC/4, async, prescaler 1:2048
23             movlw 245
24             movwf TMROH          ;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 TMRO, 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

```

15

## Ejercicios:

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



16

## 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

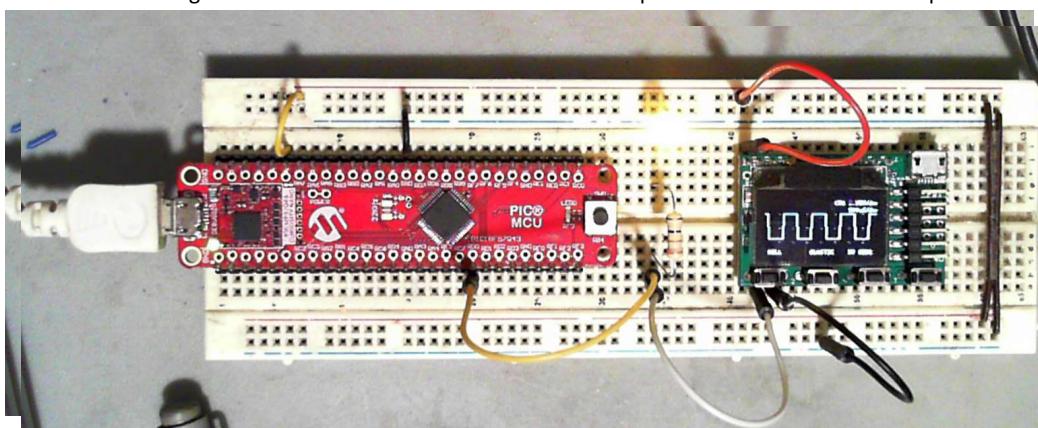
```

17

## Ejercicios:

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

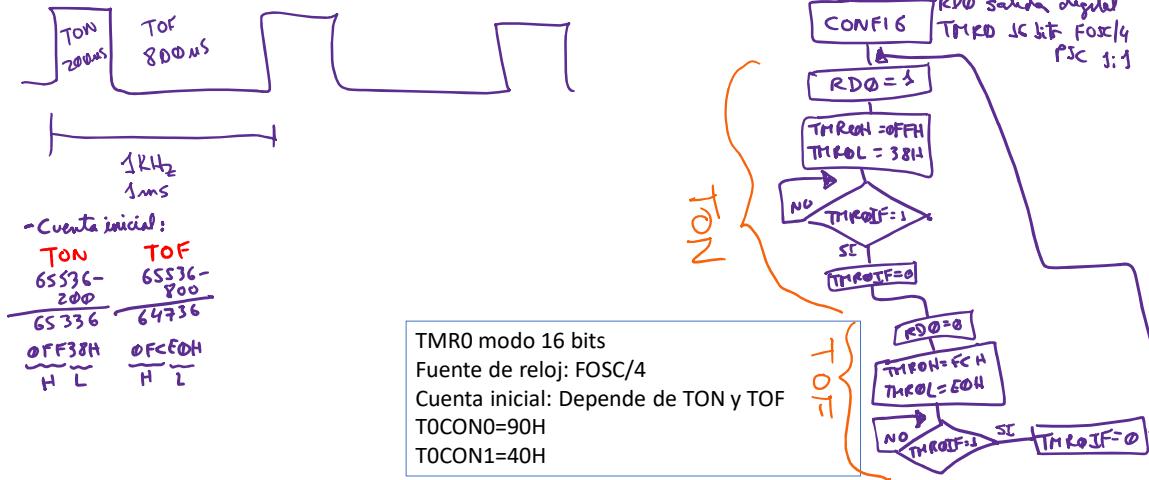
- Pruebas de la generación de onda cuadrada en el circuito implementado usando osciloscopio



18

## Ejercicios:

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



19

## 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

```

```

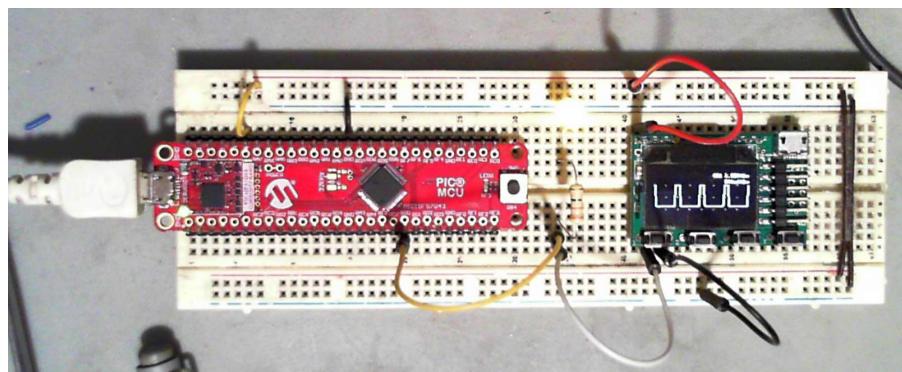
27  inicio:
28  movlb 4H
29  bsf LATD, 0, 1      ;RD0 a uno
30  movlb 3H
31  movlw OFFH
32  movwf TMROH, 1
33  movlw 38H
34  movwf TMROL, 1      ;Precarga de cuenta 65036 a TMRO
35  movlb 4H
36  otro:
37  btfss PIR3, 7, 1    ;Pregunto si TMROIF=1
38  bra otro            ;Aun no se levanta TMROIF
39  bcf PIR3, 7, 1      ;Bajamos bandera TMROIF
40  bcf LATD, 0, 1      ;RD0 a cero
41  movlb 3H
42  movlw OFCH
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

```

20

## Ejercicios:

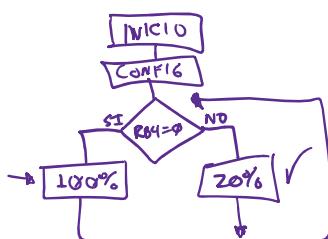
- Cómo generamos una señal cuadrada de 1KHz 20%DC ahora usando el modo 16 bits



21

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

- Preguntando al inicio por dicho botón



22

# ¿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    btfsc 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    btfss PIR3, 7, 1   ;pregunto si TMROIF=1 49    bra \$-2 50    bcf PIR3, 7, 1     ;falso vuelvo a preguntar 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    btfss PIR3, 7, 1   ;pregunto si TMROIF=1 59    bra \$-2 60    bcf PIR3, 7, 1     ;falso vuelvo a preguntar 61    bra inicio          ;retorno a inicio 62 63  end upcino </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

23

## Observaciones

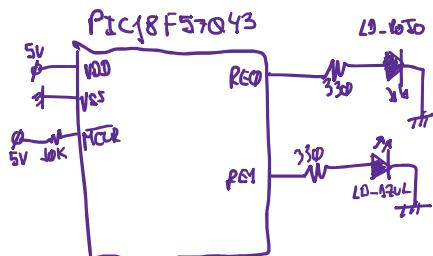
- No sale exacto en las pruebas, esto debido a que no se ha contemplado el tiempo en que se demora en ejecutar las instrucciones
- Se tiene que hacer una compensación haciendo que el TMRO cuente menos cuentas (periodo menor de temporizado) y se ajusta empleando nops
- Haciendo este tipo de compensaciones 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.

24

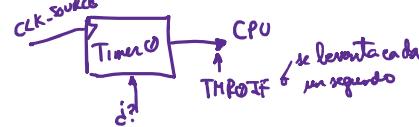
## Ejercicio 2024-1

- 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



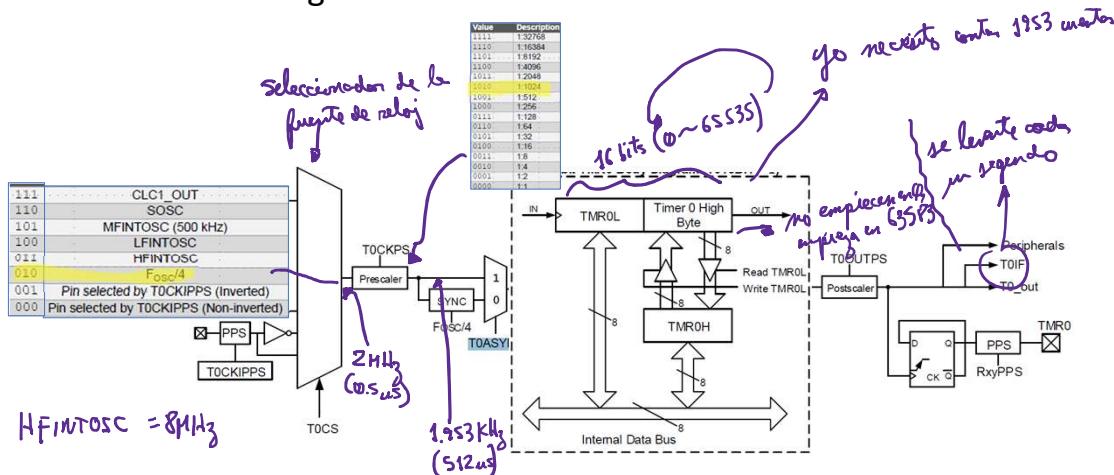
b) Análisis del Timer0 → Me avisa cada 1 segundo



25

## 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



26

## 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: LFINTOSC/4
- Modo: 16 bit ✓
- Prescaler: 1:1024 ✓
- Postscaler: 1:1 ✓
- Cuenta inicial: 63583

Name: T0CON0		Address: 0x31A	
Timer0 Control Register 0			
Bit	7	6	5
Access	0	X	0
Reset	0	0	0
Value	0	0	0
Description	TMRO Enable	TMRO Output	CKPS[3:0]
1	The module is enabled and operating	TMRO is an 8-bit timer	0000
0	The module is disabled	TMRO is a 16-bit timer	0000
Bit 5 – OUT TMRO Output			
Bit 4 – M0M16 16-Bit Timer Operation Select			
Value	1	0	
Description	TMRO is a 16-bit timer	TMRO is an 8-bit timer	
Bits 3:0 – OUTPS[3:0] TMRO Output Postscaler (Divider) Select			
Value	1111	1:16 Postscaler	CKPS[3:0]
Description	1110	1:15 Postscaler	0000
	1101	1:14 Postscaler	0001
	1100	1:13 Postscaler	0010
	1011	1:12 Postscaler	0011
	1010	1:11 Postscaler	0100
	1001	1:10 Postscaler	0101
	1000	1:9 Postscaler	0110
	0111	1:8 Postscaler	0111
	0110	1:7 Postscaler	1000
	0101	1:6 Postscaler	1001
	0100	1:5 Postscaler	1010
	0011	1:4 Postscaler	1011
	0110	1:3 Postscaler	1010
	0011	1:2 Postscaler	1011
	0000	1:1 Postscaler	1100

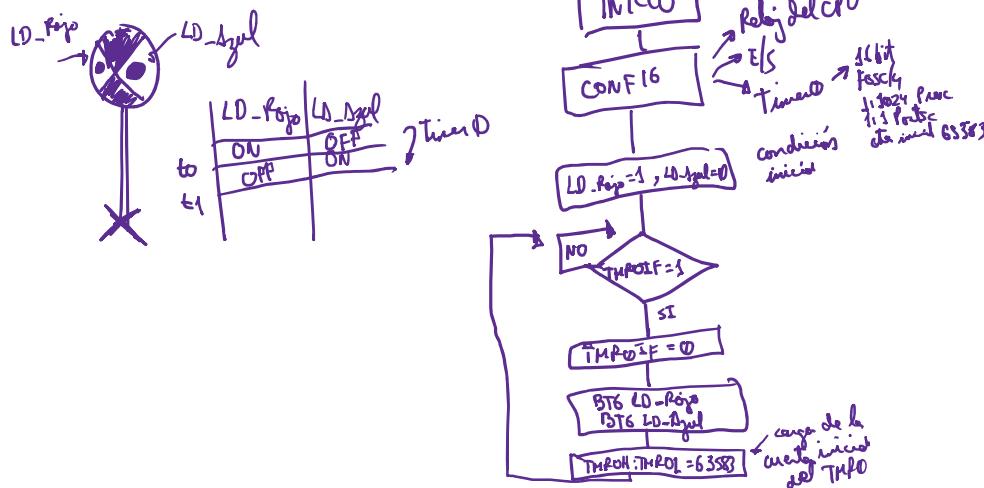
  

Name: T0CON1		Address: 0x31B	
Timer0 Control Register 1			
Bit	7	6	5
Access	0	X	0
Reset	0	0	0
Value	0	0	0
Description	CKPS[3:0]	CKPS[3:0]	CKPS[3:0]
1	0000	0000	0000
0	0001	0001	0001
Bit 5 – CS[2:0] Timer0 Clock Source Select			
Value	111	GLC_OUT	CKPS[3:0]
Description	110	SE[0]	0000
	101	MPINTOSC (600 kHz)	0001
	100	LPINTOSC	0010
	011	HFINTOSC	0011
	010	Fosc/4	0100
	001	Pin selected by T0CKIPPS (Inverted)	0101
	000	Pin selected by T0CKPPS (Not Inverted)	0110
Bit 4 – ASYNC_TMR0 Input Asynchronization Enable			
Value	1	The input to the TMRO counter is not synchronized to system clocks	
Description	0	The input to the TMRO counter is synchronized to Fosc/4	
Bits 3:0 – CKPS[3:0] Prescaler Rate Selected			
Value	1111	1:32768	CKPS[3:0]
Description	1110	1:16384	0000
	1101	1:8192	0001
	1100	1:4096	0010
	1010	1:2048	0011
	1010	1:1024	0100
	1001	1:512	0101
	0000	1:256	0110
	0111	1:128	0111
	0110	1:64	0100
	0101	1:32	0101
	0100	1:16	0110
	0011	1:8	0111
	0010	1:4	0100
	0001	1:2	0101
	0000	1:1	0110

27

## Ejercicio 2024-1

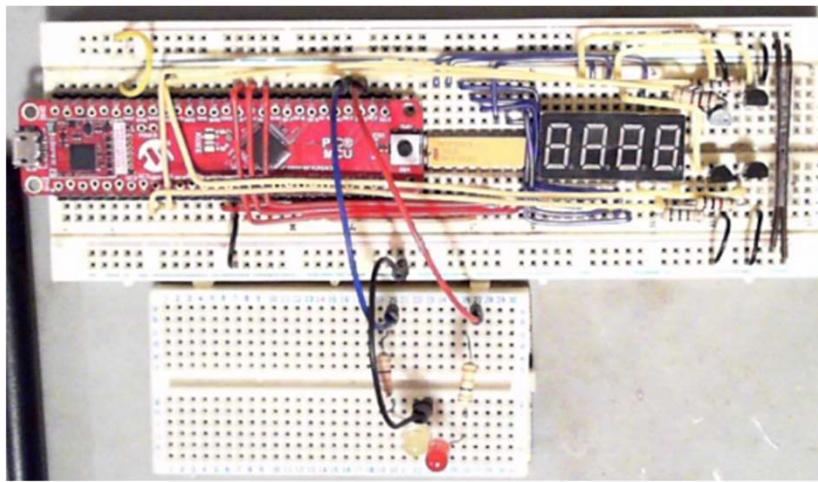
- Hacer una señal de cruce de tren empleando RE0 y RE1 como salidas hacia dos LEDs y el modulo Timer0 haciendo la base de tiempo de cambio de un segundo



28

## 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



29

## 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

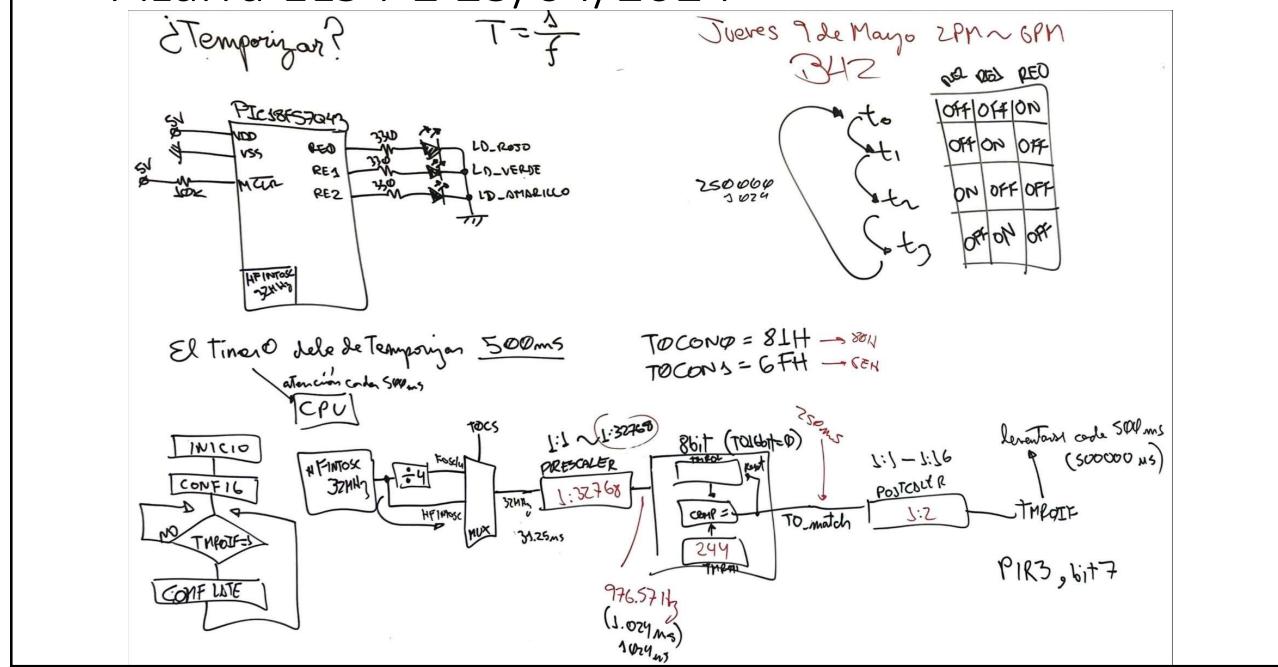
```

- 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	btfs s PIR3, 7, 1	;Pregunto si se levanto bandera de TMRO
37	bra \$-2	;Falso, vuelve a preguntar
38	baf 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		
49	end upcino	

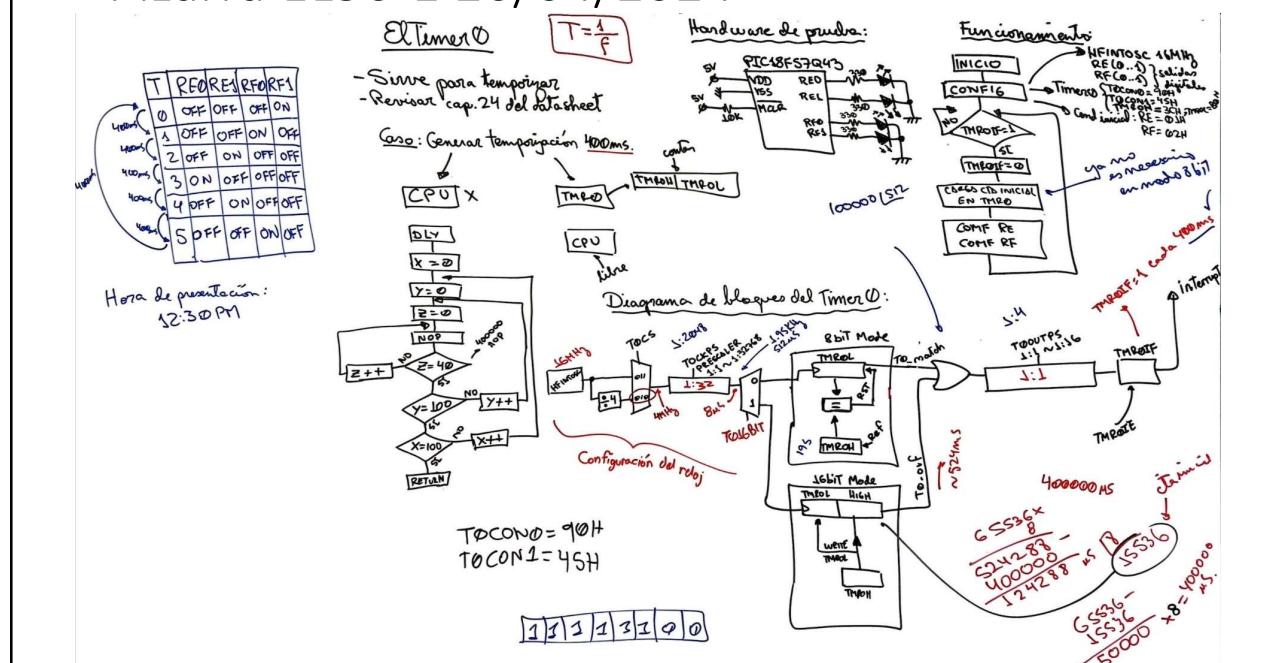
30

## Pizarra EL54-2 25/04/2024



31

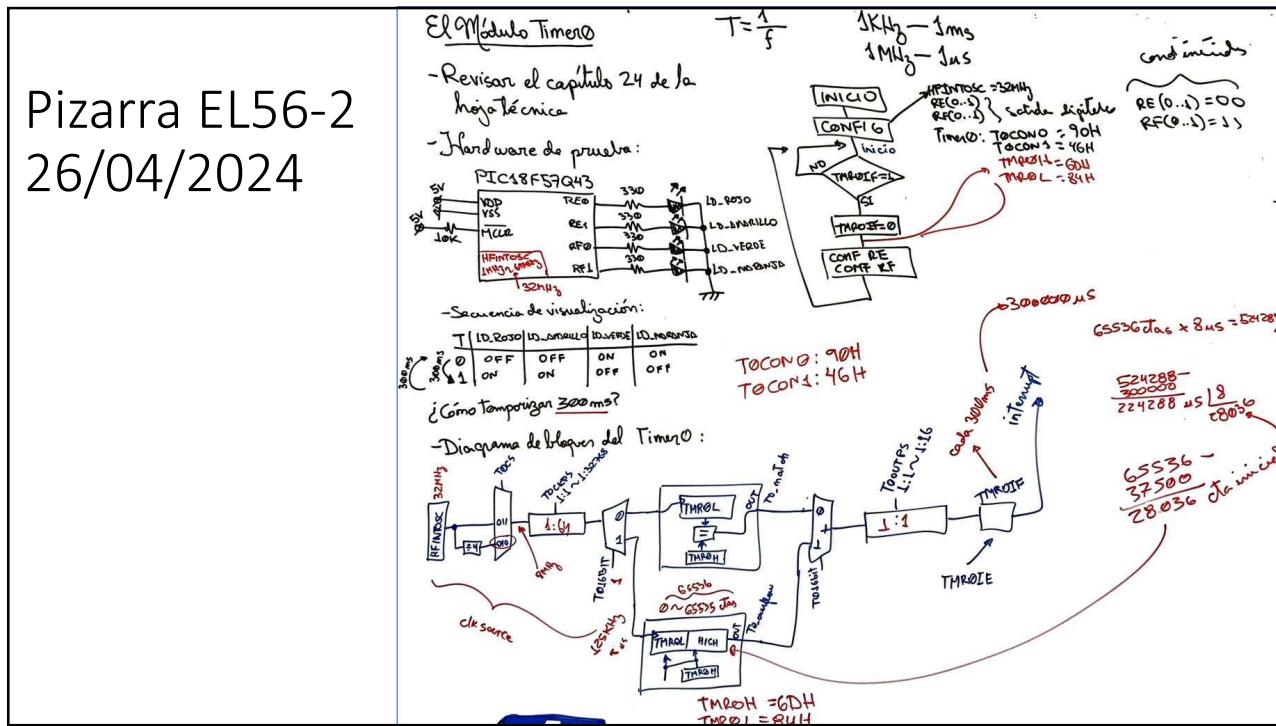
## Pizarra EL56-1 26/04/2024



32

# Pizarra EL56-2

26/04/2024



33

```

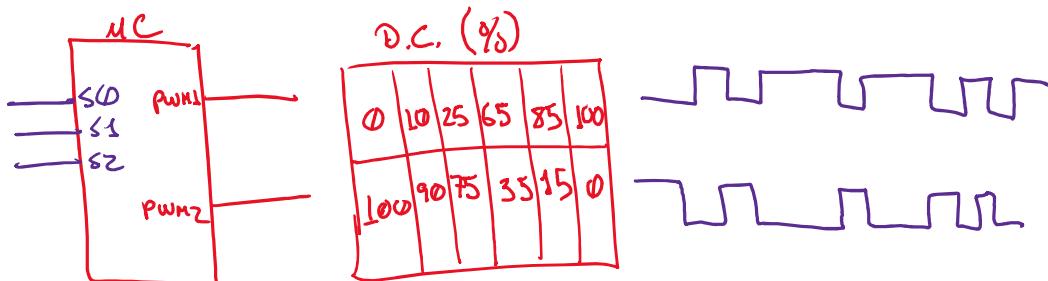
1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5
6  TAQUIS EQU 500H
7
8  upcino:
9    ORG 000000H
10   bra configuro
11
12   ORG 000700H
13   tabla1: DB 01H, 02H, 00H, 00H, 02H, 01H, 02H, 01H, 02H, 00H, 03H, 00H, 03H, 00H
14
15   ORG 000800H
16   tabla2: 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 TACONO, 1
30   movlw 44H
31   movwf TACON1, 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 TBLPTR, 1
53
54   inicio:
55     btfss PIR3, 7, 1
56     bra S-2
57     btfss PIR3, 7, 1
58     movlw 6DH
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
74   siguiente:
75     movf TAQUIS, 0, 1
76     movlb 4H
77     movwf TBLPTRL, 1
78     TBLRD+
79     movff TABLAT, LATF
80     movlw 0FH
81     movwf TBLPTRH, 1
82     TBLRD+
83     movff TABLAT, LATF
84
85   end upcino

```

34

## Ejercicios propuestos

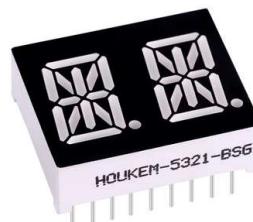
- En un PoR. ¿En qué estado se encuentra el Timer0, encendido o apagado?
- Si Fosc = 24MHz. ¿Cuál es la temporización máxima del Timer0 en modo 16 bits?
- Desarrollar un generador de PWM 2KHz con dos salidas complementarias y con opciones de dutycycle siguientes: 0%, 10%, 25%, 65%, 85% y 100%



35

## Ejercicios propuestos

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



36

Fin de la sesión