

Microcontroladores

Semestre: 2023-1

Profesor: Kalun José Lau Gan

Semana 4: Módulo Timer0 en los microcontroladores PIC18

1

¿Preguntas previas?

- El PC (program counter) solo se mencionó un ejemplo (decodificador de display de 7 segmentos). ¿Hay alguna otra utilidad?
 - El PC es un registro contador incremental el cuál almacena la dirección de memoria de la siguiente instrucción a ejecutar, si modificas dicho registro al término de la ejecución de la instrucción actual se irá a la dirección dada ahí.
 - Utilidad práctica es hacer saltos dentro de la memoria de programa pero no es muy conveniente ya que tenemos instrucciones dedicadas a ello (GOTO, BRA, CALL).
- ¿Cuáles serían las diferencias entre utilizar las instrucciones de comparación numérica (cpfs--) y las instrucciones de salto condicional?
 - Las instrucciones de comparación numérica hacen la operación del registro declarado contra Wreg, en cambio las instrucciones de salto condicional atienden a las banderas del CPU (registro STATUS)

2

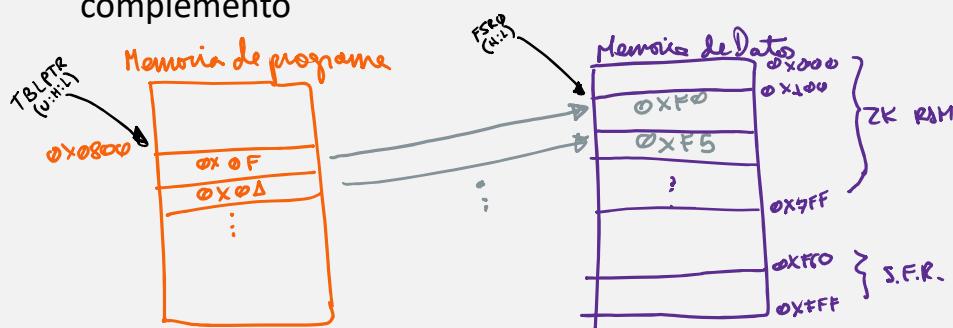
Preguntas previas:

- ¿Porqué las declaraciones de las etiquetas a los GPR difieren entre el 45K50 y el 57Q43?
 - Debido a que los GPR en el 45K50 inician en la dirección 000H de la memoria de datos, en cambio en el 57Q43 empiezan a partir de la dirección 500H de su memoria de datos.
- ¿Hay diferencias entre las weak pull-up del puerto RB del 45K50 con las del 57Q43?
 - Si, en el 45K50 la activación de dichas resistencias es con el bit 7 (RBPU) del registro INTCON y aplicados en conjunto todos los pines de dicho puerto, en cambio en el 57Q43 las resistencias se activan de manera individual por cada pin a través del registro WPUB.

3

Ejemplo sobre manipulación de datos entre memoria de programa y memoria de datos

- Memoria de programa dirección 0x0800
 - 0x0F, 0x0A, 0x08, 0x10, 0xFF, 0x3D, 0x4B, 0x15
- En la memoria de datos en la dirección 0x100 escribir los datos que están en la memoria de programa en dirección 0x0800 pero en complemento



```

inicio:    movlw 0x08
           movwf TBLPTRH
           movlw 0x00
           movwf TBLPTRL
           lfsr 0, 0x100
           movlw .8
           cpfseq TBLPTRL
           goto aunno
           goto yatermine
           TBLRD*
           comf TABLAT, w
           movwf INDFO
           incf TBLPTRL, f
           incf FSRL
           goto loop
           yatermine:nop
           end
loop:
aunno:

```

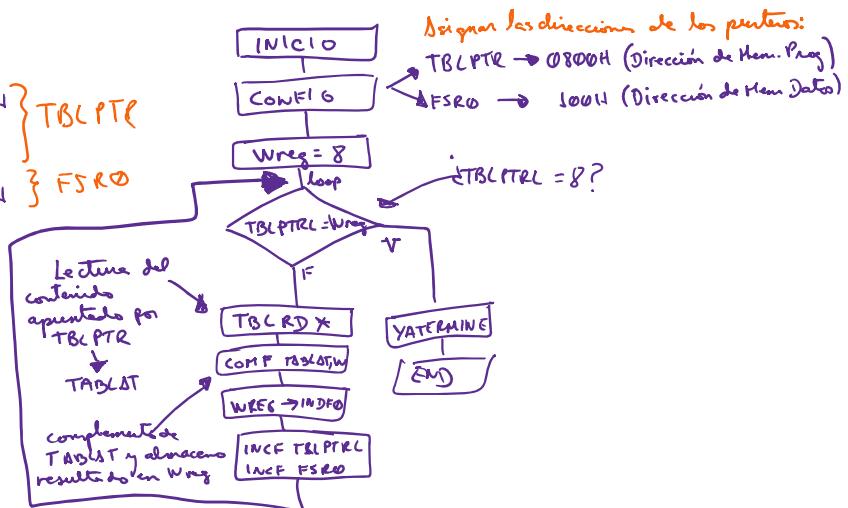
4

Aplicando ingeniería reversa para obtener el algoritmo en diagrama de flujo a partir del código XC8 PIC Assembler

```

início:    moviw 0x08
            movwf TBLPTRH
            movlw 0x00
            movwf TBLPTRL
            lfsr 0, 0x100
            movlw .8
            cpfseq TBLPTRL
            goto aunno
            goto yatermine
            TBLRD*
            comf TABLAT, w
            movwf INDFO
            incf TBLPTRL, f
            incf FSROL
            goto loop
            yatermine:nop
            end

```



5

Sobre algoritmos:



6

Agenda:

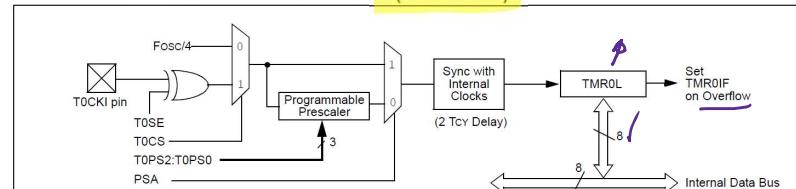
- El modulo Timer 0
- Aplicaciones con temporizadores
- Multiplexación de displays de siete segmentos
- Algoritmo para la obtención de los dígitos de centena, decena y unidad de un registro.

7

El Timer 0 en el PIC18F45K50

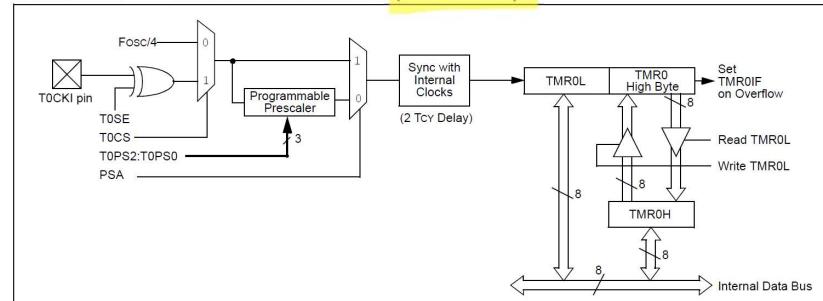
- Información extraída del datasheet del PIC18F45K50
- Diagrama de bloques:

FIGURE 11-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

8

El módulo Timer 0

- (Ref. Item 12 de la hoja técnica del microcontrolador PIC18F45K50)
- Temporizador de cuenta ascendente
- Resolución 8 bits (0-255) ó **16 bits (0-65535)**
- Las cuentas del Timer0 se alojan en:
 - TMROH:TMROL (16 bits)
 - TMROL (8 bits)
- **Tener en consideración el procedimiento estricto sobre el tratamiento de la cuenta en modo 16 bits.**
- Diversas fuentes de reloj: interno (FOSC/4) o externo (pin6 TOCKI)
- Divisor de frecuencia al reloj de entrada (1:2 – 1:256)
- El desborde se produce cuando la cuenta esta en el valor mas alto y se recibe un pulso de reloj, ocasionando que la cuenta pase a 0 y levantándose la bandera de desborde (TMROIF=1)
- Al activarse TMROIF=1 se debe de bajar manualmente la bandera para que se pueda detectar un nuevo desborde (simplemente haciendo bcf INTCON, 2, siendo el bit 2 el TMROIF).
- Al desbordarse puede emitir interrupción (TMROIF = 1), revisar interrupciones y sus 10 registros implicados
- Se usa el registro T0CON (SFR 0FD5H) para configurar el Timer0 (por defecto T0CON=0FFH)

9

Timer 0 – Modos de trabajo

- Modo temporizador (reloj interno para la cuenta: FOSC/4)
 - Ej. Generador de ondas cuadradas, base de tiempo para la multiplexación de los displays de siete segmentos, cuentas regresivas cortas, efectos de desplazamiento en displays, LED blinkers, etc
 - **No se usa para aplicaciones en tiempo real (relojes, cronómetros)**
- Modo contador (empleando pin externo TOCKI para la cuenta)
 - Ej. Velocímetro para bicicleta, medidor de RPMs de un motor

10

El Timer0 – Registro T0CON

- Tener en cuenta que los valores por defecto en un PoR son '1' en cada bit del registro
- TOSE es ignorado si TOCS = 0

REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER							
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	TOCS	TOSE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0
Legend:							
R = Readable bit -n = Value at POR	W = Writable bit '1' = Bit is set	U = Unimplemented bit, read as '0' '0' = Bit is cleared	x = Bit is unknown				
bit 7	TMR0ON: Timer0 On/Off Control bit 1 = Enables Timer0 0 = Stops Timer0						
bit 6	T08BIT: Timer0 8-Bit/16-Bit Control bit 1 = Timer0 is configured as an 8-bit timer/counter 0 = Timer0 is configured as a 16-bit timer/counter						
bit 5	TOCS: Timer0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO)						
bit 4	TOSE: Timer0 Source Edge Select bit 1 = Increment on high-to-low transition on T0CKI pin 0 = Increment on low-to-high transition on T0CKI pin						
bit 3	PSA: Timer0 Prescaler Assignment bit 1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler. 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.						
bit 2-0	T0PS2:T0PS0: Timer0 Prescaler Select bits 111 = 1:256 Prescale value 110 = 1:128 Prescale value 101 = 1:64 Prescale value 100 = 1:32 Prescale value 011 = 1:16 Prescale value 010 = 1:8 Prescale value 001 = 1:4 Prescale value 000 = 1:2 Prescale value						

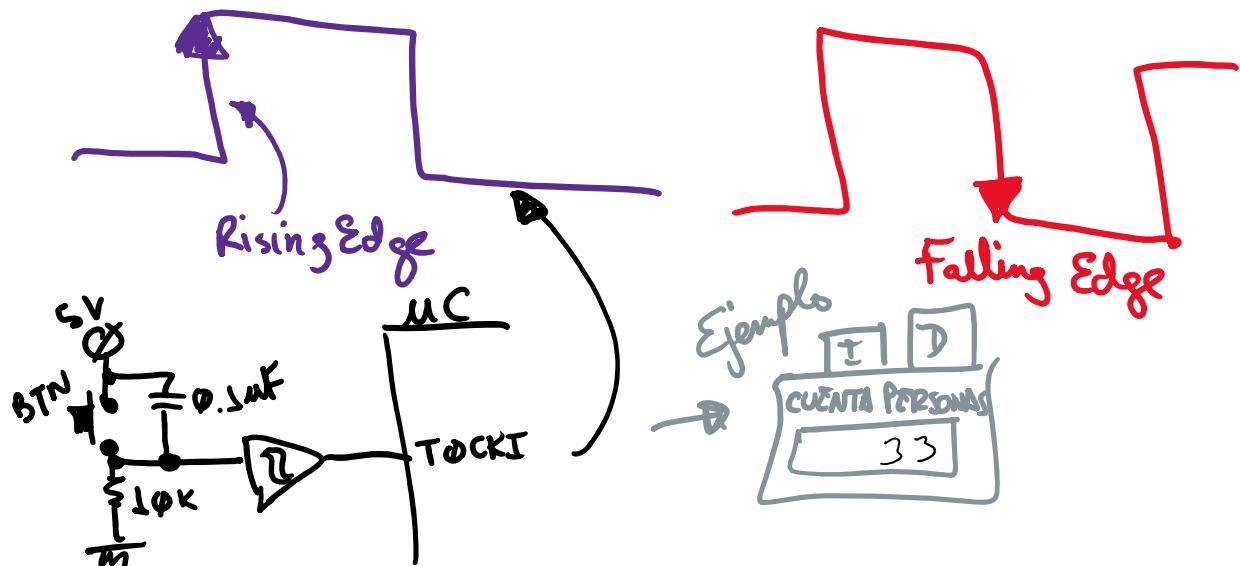
11

Procedimiento para ingresar una cuenta inicial al Timer 0 en modo 16 bits

1. Si por ejemplo se quiere ingresar el número 5536 como cuenta inicial, convertirlo a hexadecimal (DEC 5536 = HEX 15A0H)
2. Se ingresa el dato de 8 bit mas significativo a TMR0H, en el ejemplo 15H hacia TMR0H.
3. Se ingresa el dato de 8 bit menos significativo a TMR0L, haciendo esto se sube en simultáneo el TMR0H al registro de cuentas del Timer0, en el ejemplo 0A0H hacia TMR0L.
4. Recordar que luego del desborde se deberá ingresar nuevamente la cuenta inicial para preservar el temporizado de manera continua.
5. No olvidar que luego de un desborde se levanta la bandera y ésta hay que bajarla manualmente para que el módulo pueda activarla ante un nuevo desborde

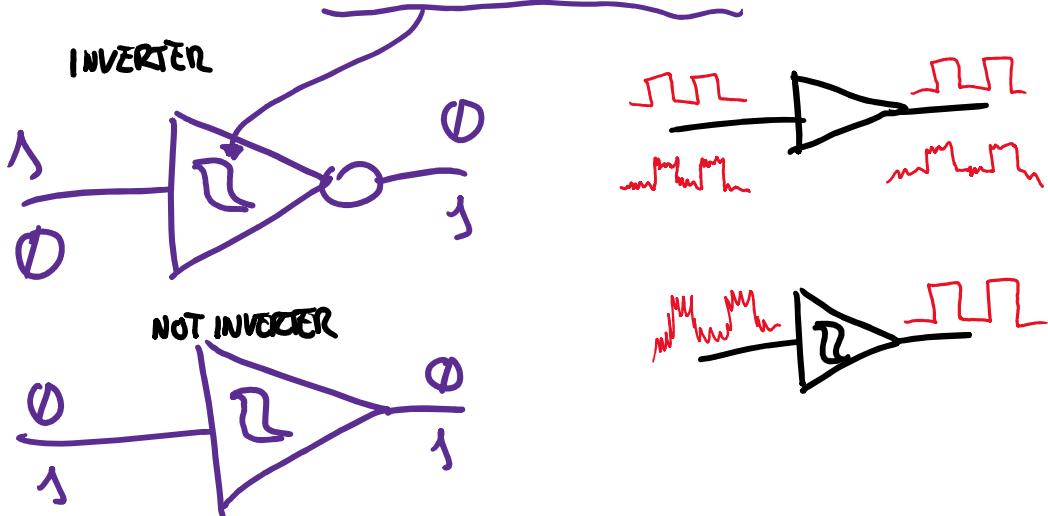
12

Recordando Rising Edge vs Falling Edge



13

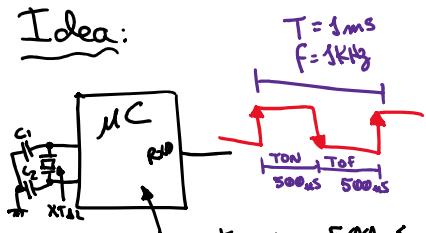
Recordando el SCHMITT TRIGGER



14

Ejemplo: Generar una onda cuadrada de 1KHz DC 50 % empleando el Timer0 modo 8 bits:

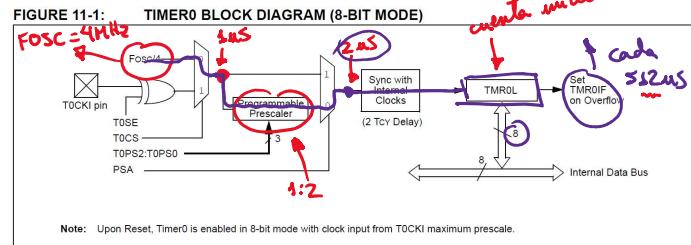
Idea:



Tiene que temporizar 500μs para que basalle RX0

Lo va a hacer el Timer0

¿Cómo hago para que el Timer0 temporice 500μs?



Resumen:

1: Modo 8 bit - temporizador ($F_{osc}/4$)

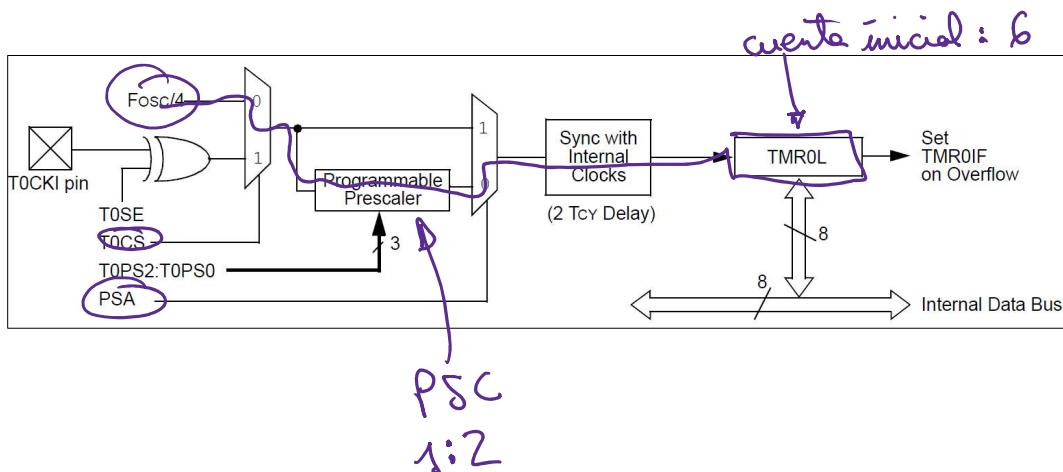
2: Prescaler 1:2

3: Cuenta inicial de 6 } $\rightarrow TMR0L$

$$T0CON = \emptyset \times C0 \\ 001$$

15

Continuación...



16

Configurar el TOCON:

REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-4	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	TOPS2	T0PS1	0PS0
bit 7							bit 0

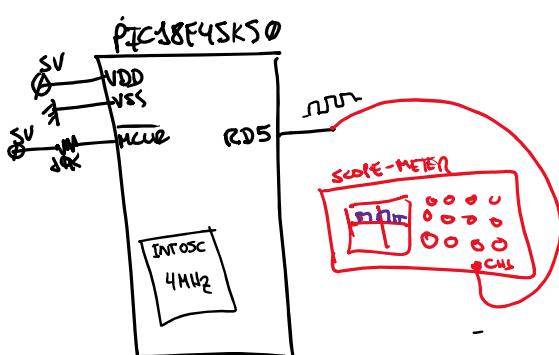
Legend:	
R = Readable bit	W = Writable bit
-n = Value at POR	'1' = Bit is set
	'0' = Bit is cleared
	X = Bit is unknown

- bit 7 → **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0
- bit 6 → **T08BIT:** Timer0 8-Bit/16-Bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter
- bit 5 → **T0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO) ($f_{osc}/4$)
- bit 4 → **T0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin
- bit 3 → **PSA:** Timer0 Prescaler Assignment bit
1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
- bit 2-0 → **TOPS2:TOPS0:** Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

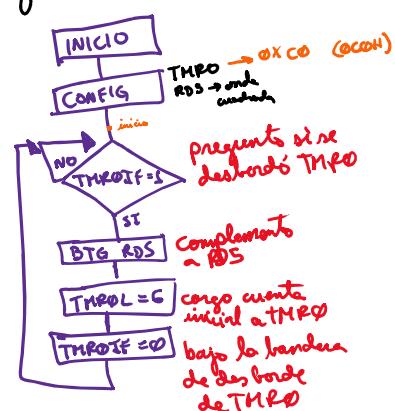
17

Ejemplo: Generar una onda cuadrada de 1KHz empleando el Timer0 modo 8 bits:

Circuito:

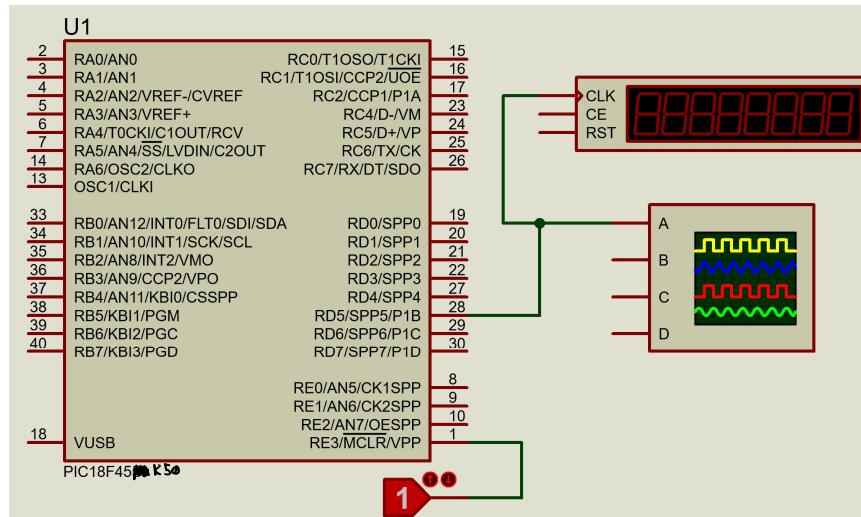


Algoritmo:



18

Hardware



19

Propuesta de código en XC8 PIC ASM de lo anterior

```

PROCESSOR 18F45K50
#include "cabecera.inc"

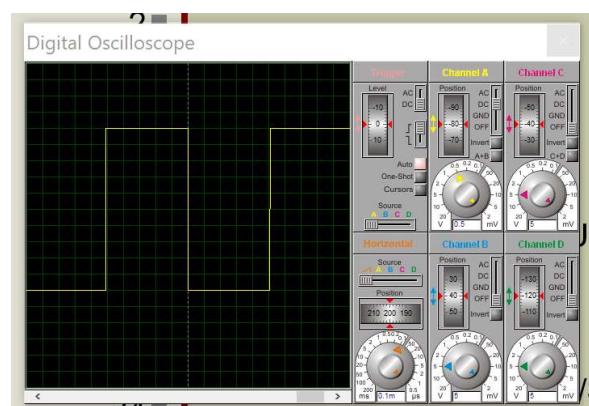
PSECT principal, class=CODE, reloc=2, abs
principal:
    ORG 000000H
    bra configuro
    ORG 000020H

configuro:
    movlw 0FH
    movlw 52H
    movwf OSCCON
; configuración de los puertos
    bcf TRISD, 5
    bcf ANSEL0, 5      ;RD5 como salida digital
    movlw 0C0H          ;Timer0 ON, modo 8bit, 1:2PSC, FOSC/4
    movwf T0CON

loop:   movlw 6
        movwf TMR0L      ;Cuenta inicial de 6
otro:   btfs INTCON, 2 ;Pregunta si TMR0IF=1
        goto otro
        btg LATD, 5      ;Basculamos RD5 para que genere la onda cuadrada
        bcf INTCON, 2    ;Bajando la bandera TMR0IF
        goto loop

end principal

```



20

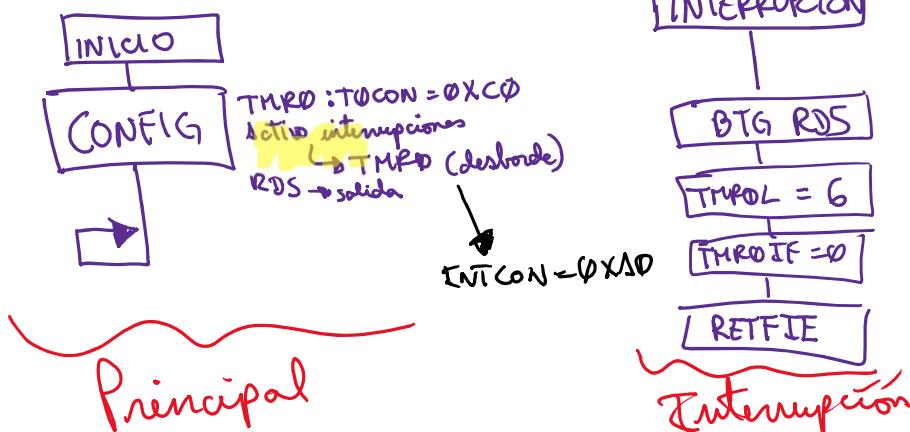
Observaciones

- 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 TMR0 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.

21

Mejora del generador de onda cuadrada de 1KHz empleando interrupciones

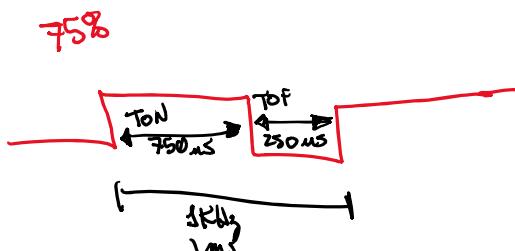
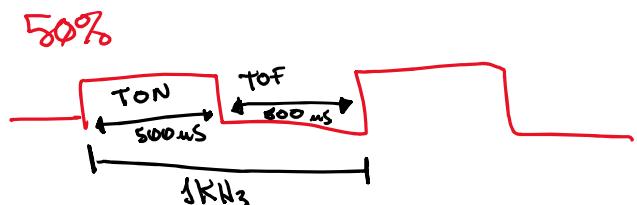
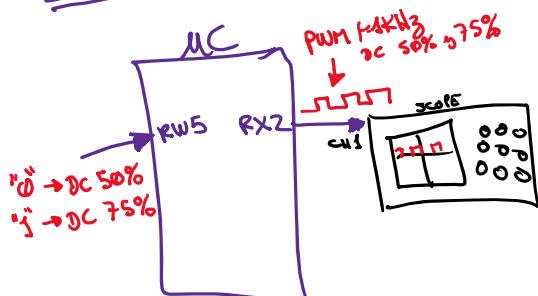
Diagramas de flujo:



22

Ejemplo: Desarrollar un generador de PWM con frecuencia 1KHz y con dos opciones de Duty Cycle 50% y 75%.

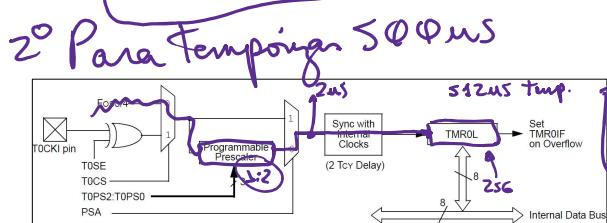
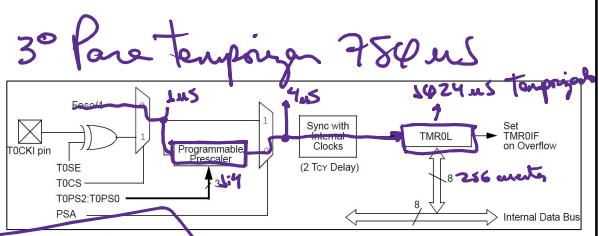
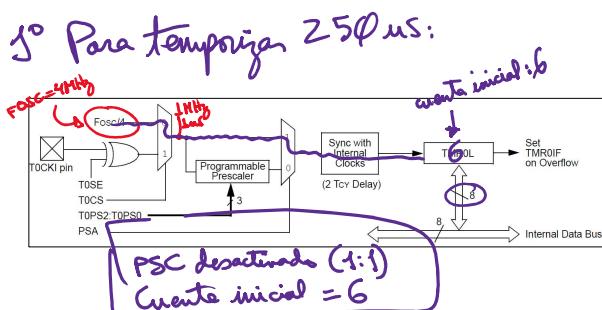
Idea:



Nota: Debemos de configurar tres temporizaciones en el Timer0:
250μs, 500μs y 750μs.

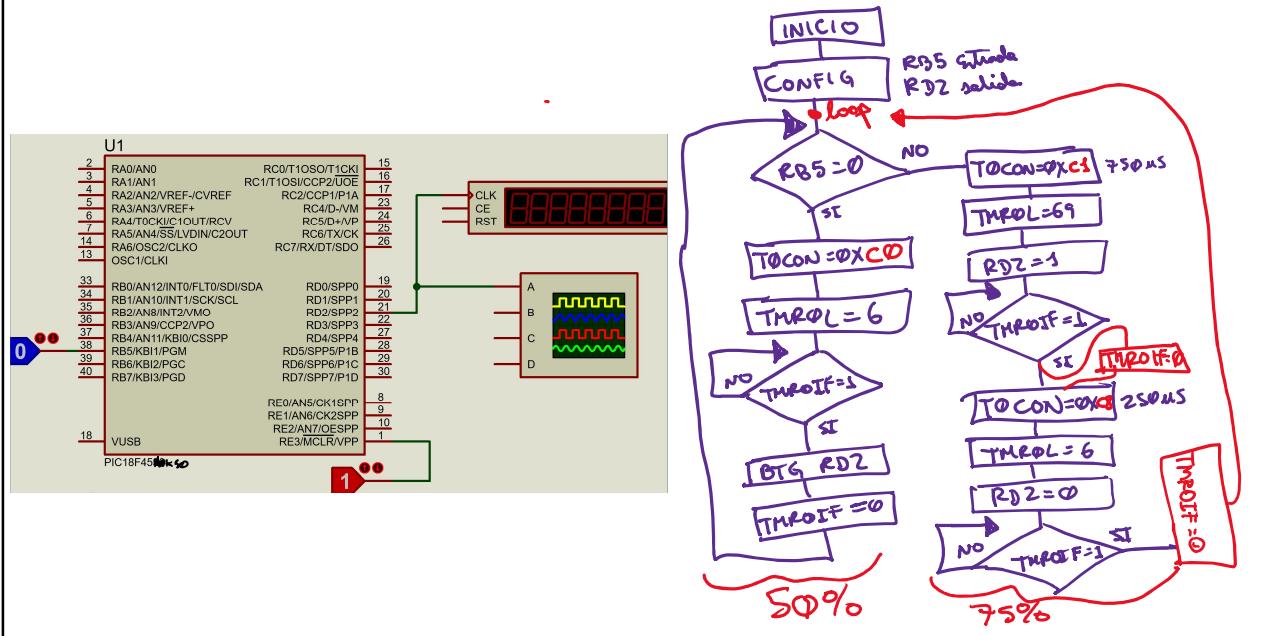
23

Cont. del ejemplo: Configuración del Timer0 según tiempos de temporización requeridos



24

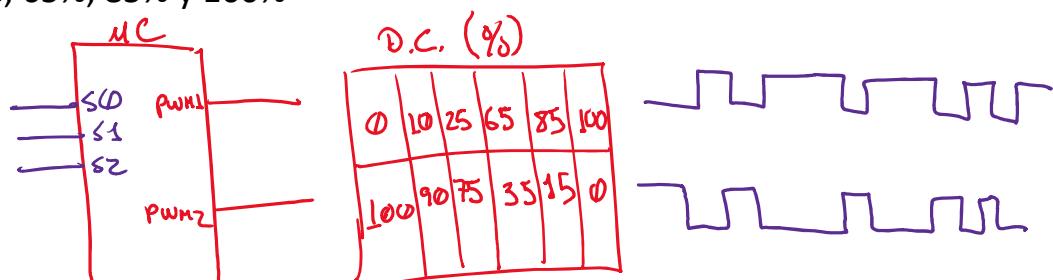
Cont. del ejemplo: Hardware y desarrollo del diagrama de flujo



25

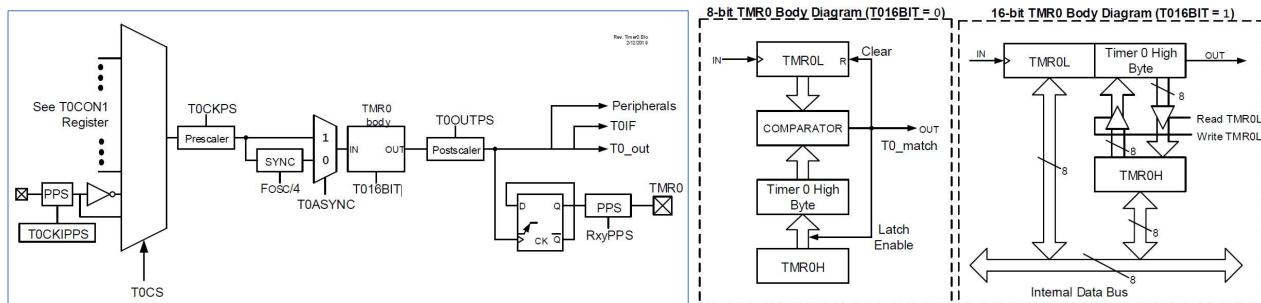
Fin de la sesión

- Links adicionales:
 - Microchip Timer0 Tutorial part1: <http://ww1.microchip.com/downloads/en/devicedoc/51682a.pdf>
 - Microchip Timer0 Tutorial part2: <http://ww1.microchip.com/downloads/en/DeviceDoc/51702a.pdf>
- Ejercicio: Desarrollar un generador de PWM 2KHz con dos salidas complementarias y con opciones de dutycycle siguientes: 0%, 10%, 25%, 65%, 85% y 100%



26

El Timer0 en el PIC18F57Q43



- Es diferente frente al modulo encontrado en el 45K50
- Posee mejoras como:
 - Comparador en modo 8 bits
 - Prescaler extendido de 1:1 hasta 1:32768
 - Postscaler 1:1 – 1:16
 - Varias fuentes de reloj para seleccionar

27

El Timer0 en el PIC18F57Q43

• Registros de configuración

Timer0 Control Register 0							
Bit	7	6	5	4	3	2	1
Access	R/W		OUT	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0
Bit 7 – TMR0 Enable							
Value	Description						
1	The module is enabled and operating						
0	The module is disabled						
Bit 5 – OUT TMR0 Output							
Value	Description						
1	TMR0 is a 16-bit timer						
0	TMR0 is an 8-bit timer						
Bits 3:0 – OUTPS[3:0] TMR0 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						
Name: T0CON0 Address: 0x31A							
Name: T0CON1 Address: 0x31B							
Timer0 Control Register 1							
Bit	7	6	5	4	3	2	1
Access	R/W	CS[2:0]	R/W	ASYNC	CKPS[3:0]	R/W	R/W
Reset	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 T0CKIPPS (inverted)						
000	Pin selected by T0CKIPPS (Non-inverted)						
Bit 4 – ASYNC TMR0 Input Asynchronization Enable							
Value	Description						
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						
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						

28

El Timer0 en el PIC18F57Q43

- Ejercicio: Desarrollar un generador de onda cuadrada de 1KHz y 50% DC con el Curiosity Nano PIC18F57Q43 con RD0 como salida.

29

Cuestionario:

- 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?
- ¿Qué es lo que hace la instrucción BTG?
- Si Fosc = 12MHz. ¿Cuánto se demorará en ejecutar la instrucción CPFSGT?
- Hacer un circuito de conexión entre el microcontrolador PIC18F4550 y el siguiente dispositivo:



30

Fin de la sesión