

# Microcontroladores

Sesión Teoría

Semana 3

Profesor: Kalun Lau

1

## Preguntas previas:

- PC1: martes 26 de enero 19:00, contenido hasta lo analizado el día anterior a la evaluación, desarrollo manual a mano en hoja bond y posteriormente escaneado o fotografiado, evaluación individual.
- LB1: jueves 28 de enero 10:00, desarrollado según procedimiento e implementado, trabajo individual

2

## Agenda

- El Timer 0
- Interrupciones

3

## El módulo Timer 0

- (Ref. Item 11 de la hoja técnica del microcontrolador PIC18F4550)
- Temporizador de cuenta ascendente
- Resolución 8 bits (0-255) ó 16 bits (0-65535)
- Las cuentas del Timer0 se alojan en:
  - TMR0H:TMR0L (16 bits)
  - TMR0L (8 bits)
- Tener en consideración el procedimiento estricto sobre el tratamiento de la cuenta en modo 16 bits.
- Diversas fuentes de reloj (interno o externo)
- Divisor de frecuencia al reloj de entrada (1:2 – 1:256)
- Al desbordarse puede emitir interrupción (TMR0IF = 1), revisar interrupciones y sus 10 registros implicados
- Se usa el registro TOCON (SFR 0xFD5) para configurar el Timer0 (por defecto TOCON=0xFF)

4

## Timer 0 – Modos de trabajo

- Modo temporizador (reloj interno)
  - Ej. Generador de ondas cuadradas, base de tiempo para la multiplexación de los displays de siete segmentos
  - No se usa para aplicaciones en tiempo real (relojes, cronómetros)
- Modo contador (empleando pin T0CKI)
  - Ej. Velocímetro para bicicleta

5

## El Timer 0

- Diagrama de bloques:

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

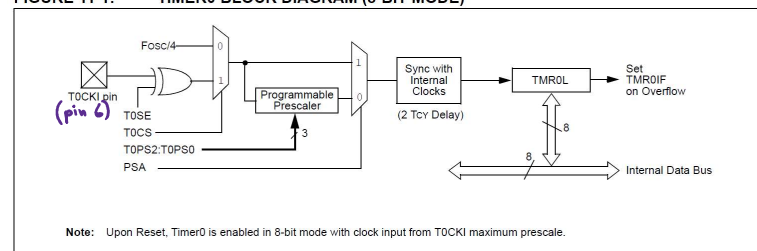
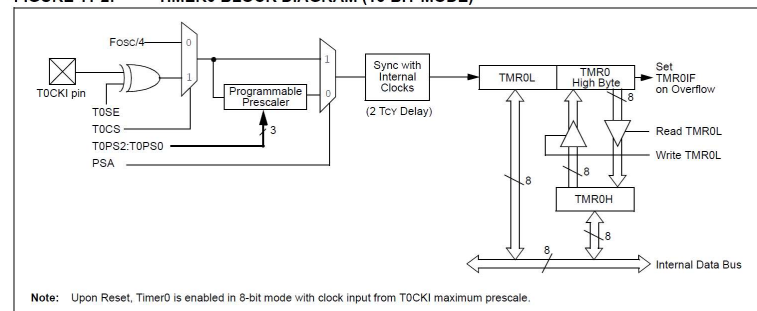


FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



6

## El Timer0 – Registro TOCON

REGISTER 11-1: TOCON: 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	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

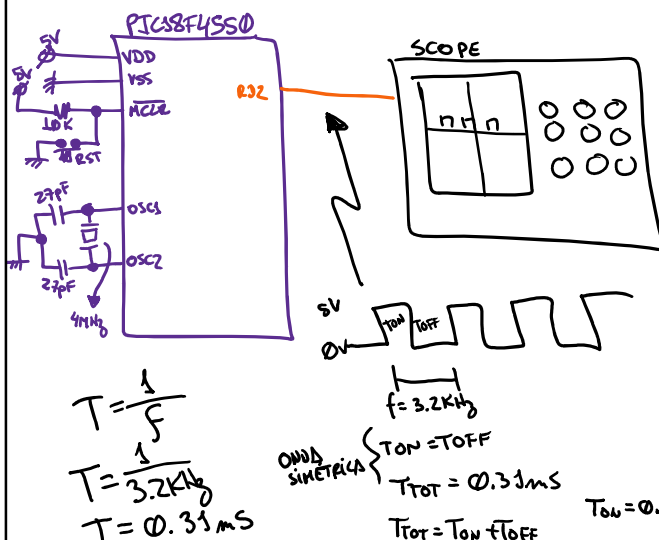
x = Bit is unknown

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

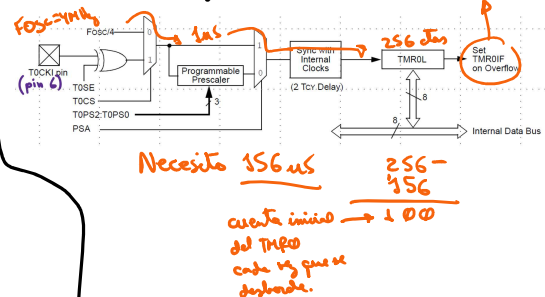
bit 7	<b>TMR0ON:</b> Timer0 On/Off Control bit 1 = Enables Timer0 0 = Stops Timer0
bit 6	<b>T08BIT:</b> Timer0 8-Bit/16-Bit Control bit 1 = Timer0 is configured as an 8-bit timer/counter 0 = Timer0 is configured as a 16-bit timer/counter
bit 5	<b>T0CS:</b> Timer0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO)
bit 4	<b>T0SE:</b> 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	<b>PSA:</b> 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	<b>T0PS2:T0PS0:</b> 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

7

## Ejemplo: Generar una onda cuadrada simétrica (DC 50 %) de 3.2KHz usando el Timer0



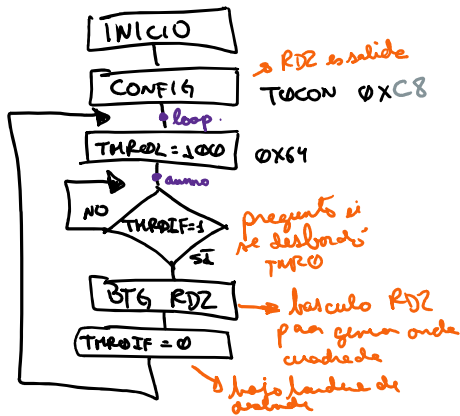
El Timer0 se debe de configurar para que temporice 0.15ms (150 micro segundos)



8

## Ejemplo: Generar una onda cuadrada simétrica (DC 50 %) de 3.2KHz usando el Timer0

Diagrama de Flujo:



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
TMR0ON	T08BIT	T0CS	T0SE	PSA	TOPS2	TMR0 bit 0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2-0	

Legend:  
 R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 -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)

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

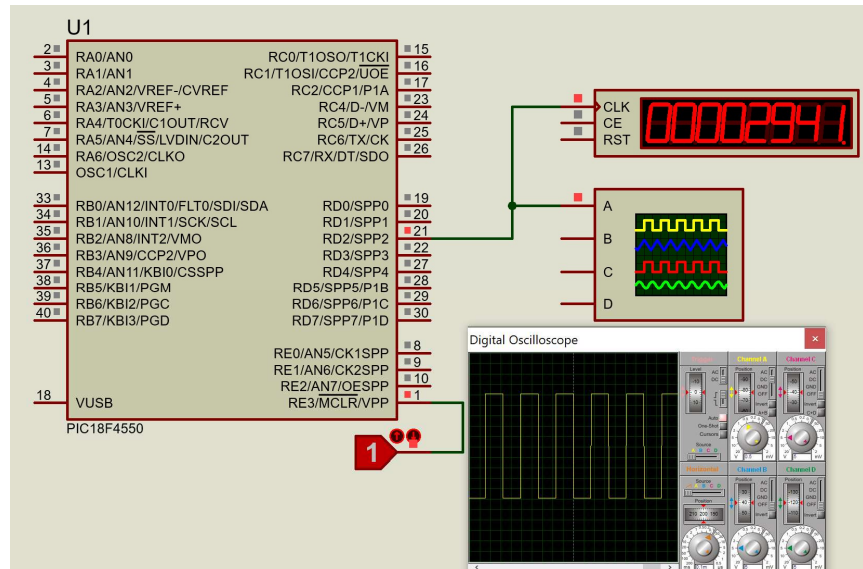
*Handwritten notes:*  
 T0CON = 0xC8  
 X (DON'T CARE)  
 X (DON'T CARE)

9

## Código y simulación en Proteus

```

21      org 0x0000
22      goto init_conf
23
24      ;Aquí se pueden declarar las
25
26      org 0x0020
27      init_conf:
28          bcf TRISD, 2
29          movlw 0xC8
30          movwf T0CON
31
32      loop:
33          movlw .100
34          movwf TMR0L
35      aunno:
36          btfss INTCON, TMR0IF
37          goto aunno
38          btg LATD, 2
39          bcf INTCON, TMR0IF
40          goto loop
41      end
  
```



10

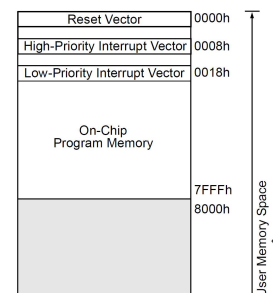
## Procedimiento para ingresar una cuenta inicial al Timer 0 en modo 16 bits

1. Si por ejemplo se quiere ingresar el número 5536 como cuenta inicial, convertirlo a hexadecimal (DEC 5536 = HEX 0x15A0)
2. Se ingresa el dato de 8 bit mas significativo a TMR0H, en el ejemplo 0x15 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 0xA0 hacia TMR0L.
4. Recordar que luego del desborde se deberá ingresar nuevamente la cuenta inicial para preservar el temporizado de manera continua.

11

## Interrupciones:

- Las interrupciones son eventualidades que detienen el flujo normal de operación del microcontrolador.
- En el PIC18F4550 tenemos dos vectores de interrupción:
  - Alta Prioridad (0x0008)
  - Baja Prioridad (0x0018)
- Las prioridades están desactivadas por defecto, si están desactivadas, todas van al 0x0008)
- La interrupción externa INTO solo es alta prioridad
- Todos los periféricos internos del microcontrolador (Timers, INTs externas, CCP, EUSART, A/D, comparadores analógicos, etc) pueden emitir interrupciones al CPU.
- Son 10 registros de configuración del sistema de interrupciones
- Las banderas que indican la fuente de interrupción deberán de bajarse una vez activados.



12

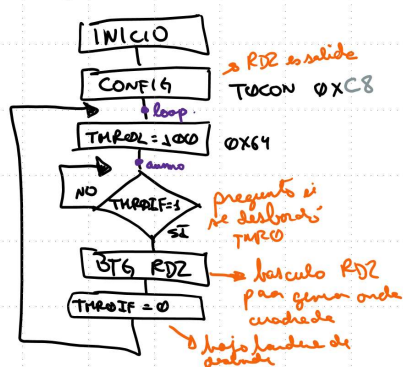
## Los diez registros para configurar las interrupciones:

- **INTCON** (están las interrupciones primarias incluyendo los habilitadores globales)
- **INTCON2** (configuración de flancos de las ints. exts., prioridad en TMR0 y RB)
- **INTCON3** (prioridades, habilitadores y banderas de INT1 e INT2)
- **PIE1** (habilitadores de interrupciones de periféricos parte 1)
- **PIE2** (habilitadores de interrupciones de periféricos parte 2)
- **PIR1** (banderas de interrupciones de periféricos parte 1)
- **PIR2** (banderas de interrupciones de periféricos parte 2)
- **IPR1** (configuración de prioridades de interrupciones de periféricos parte 1)
- **IPR2** (configuración de prioridades de interrupciones de periféricos parte 2)
- **RCON** (registro de control de reset del CPU, incluye el IPEN - habilitador de prioridades)

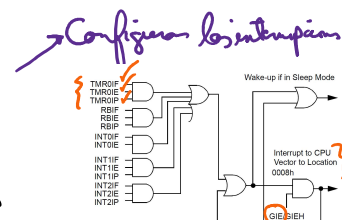
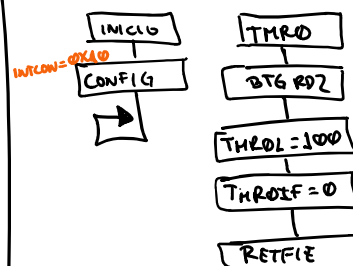
13

Modificando el ejemplo de la generación de la onda de 3.2KHz para que funcione con la interrupción por desborde de Timer0

Diagrama de Flujo Anterior:



Nuevo diagrama:



14

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF <sup>(1)</sup>
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'  
x = Bit is unknown

bit 7 **GIE/GIEH**: Global Interrupt Enable bit  
When IPEN = 0:  
1 = Enables all unmasked interrupts  
0 = Disables all interrupts  
When IPEN = 1:  
1 = Enables all high-priority interrupts  
0 = Disables all interrupts

bit 6 **PEIE/GIEL**: Peripheral Interrupt Enable bit  
When IPEN = 0:  
1 = Enables all unmasked peripheral interrupts  
0 = Disables all peripheral interrupts  
When IPEN = 1:  
1 = Enables all low-priority peripheral interrupts (if GIE/GIEH = 1)  
0 = Disables all low-priority peripheral interrupts

bit 5 **TMR0IE**: TMR0 Overflow Interrupt Enable bit  
1 = Enables the TMR0 overflow interrupt  
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE**: INT0 External Interrupt Enable bit  
1 = Enables the INT0 external interrupt  
0 = Disables the INT0 external interrupt

bit 3 **RBIE**: RB Port Change Interrupt Enable bit  
1 = Enables the RB port change interrupt  
0 = Disables the RB port change interrupt

bit 2 **TMR0IF**: TMR0 Overflow Interrupt Flag bit  
1 = TMR0 register has overflowed (must be cleared in software)  
0 = TMR0 register did not overflow

bit 1 **INT0IF**: INT0 External Interrupt Flag bit  
1 = The INT0 external interrupt occurred (must be cleared in software)  
0 = The INT0 external interrupt did not occur

bit 0 **RBIF**: RB Port Change Interrupt Flag bit<sup>(1)</sup>  
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)  
0 = None of the RB7:RB4 pins have changed state

INTCON = 0x0A0

Configuración del registro INTCON para habilitar la interrupción por desborde del Timer0

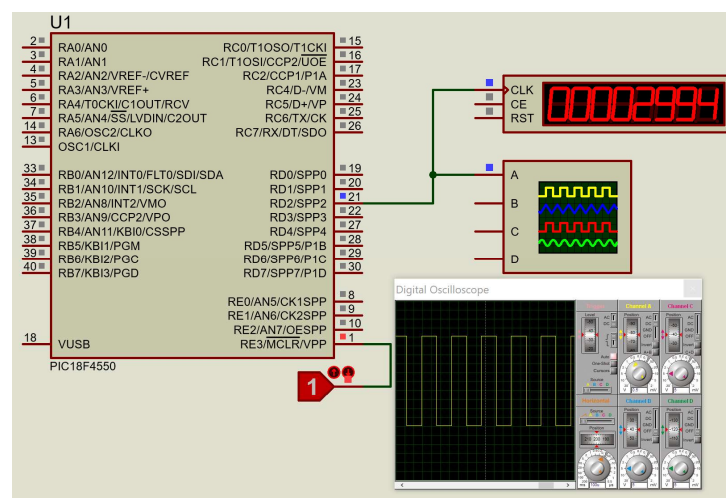
15

## Código mejorado con interrupciones

```

21      org 0x0000
22      goto init_conf
23
24      org 0x0008
25      goto TMR0_ISR
26
27      ;Aquí se pueden declarar las
28
29
30      org 0x0020
31      init_conf:
32      bcf TRISD, 2
33      movlw 0xC8
34      movwf T0CON
35      movlw 0xA0
36      movwf INTCON      ;Hab
37
38      loop:
39      goto loop
40
41      TMR0_ISR:
42      btg LATD, 2
43      movlw .100
44      movwf TMR0L
45      bcf INTCON, TMR0IF
46      retfie
47      end

```



16



## 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%

