

Microcontroladores

Semana 5

Preguntas previas:

- Llenar la encuesta inicial UPC del curso de Microcontroladores y participar en el sondeo de la sesión de videoconferencia

Agenda:

- Los periféricos temporizadores en los microcontroladores
- El módulo Timer0:
 - Características
 - Modos de trabajo
 - Proceso de configuración
 - Aplicaciones

¿Para qué sirven los módulos de temporización en un microcontrolador?

- Como un cronómetro, mide el tiempo!
- Aplicaciones relacionadas con medir el tiempo:
 - Cronómetro
 - Cuenta regresiva
 - Generación de ondas cuadradas (digital) simétricas o asimétricas



Ejemplo: Generar una señal cuadrada simétrica (Duty Cycle 50%) a una frecuencia de 1KHz

- Empleando Arduino

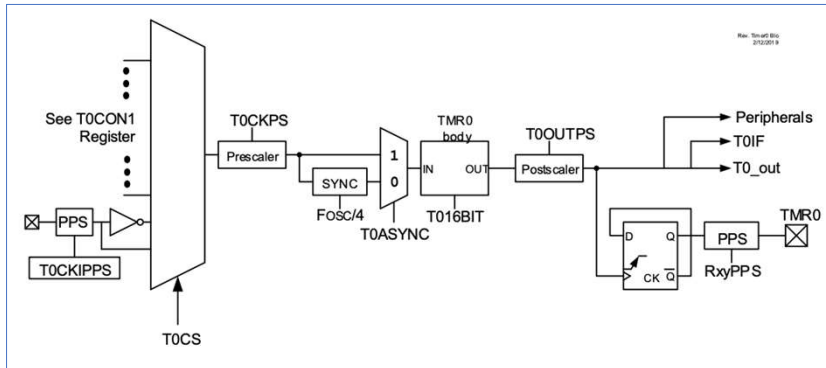
```
void loop(){  
  digitalWrite(3, HIGH);  
  delayus(500);  
  digitalWrite(3, LOW);  
  delayus(500);  
}
```

Hay siete temporizadores en el PIC18F57Q43

- Según la hoja técnica, se tiene Timer0, Timer1, Timer2, Timer3, Timer4, Timer5 y Timer6.
- En el índice de la hoja técnica solo aparecen Timer0, Timer1, Timer2, y el resto?
- Timer0
- Timer1, Timer3 y Timer5 son iguales
- Timer2, Timer4 y Timer6 son iguales

El Timer0

- De cuenta ascendente
- Tiene dos modos de trabajo:
- 8 bits con sistema de comparación
- 16 bits regular con desborde
- T0IF es la bandera TMR0IF que se encuentra en el VIC (registro PIR3 bit7)

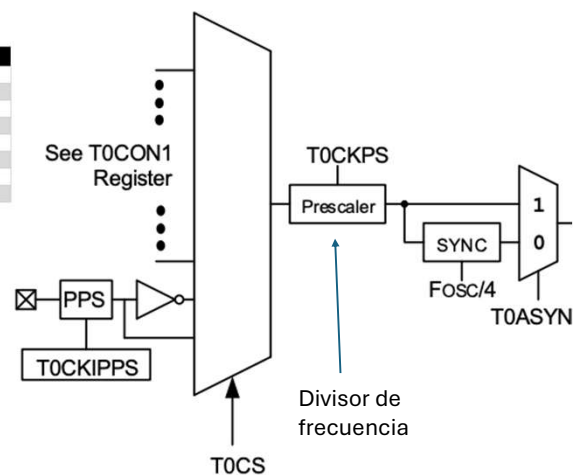


Configuración de la fuente de reloj del Timer0

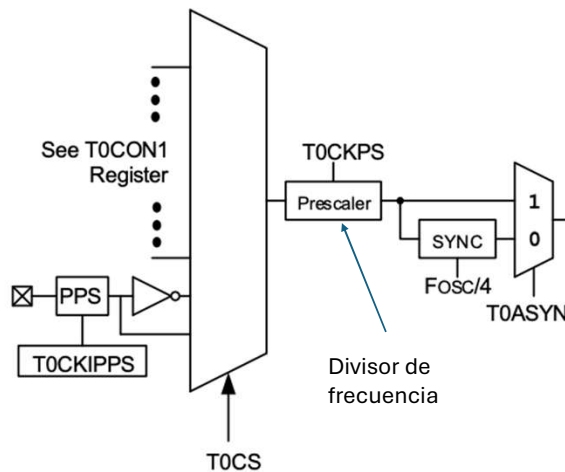
- CS[2:0] Timer0 Clock Source Select

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)

- Se tiene diferentes fuentes de reloj para alimentar al Timer0



Configuración de la fuente de reloj del Timer0



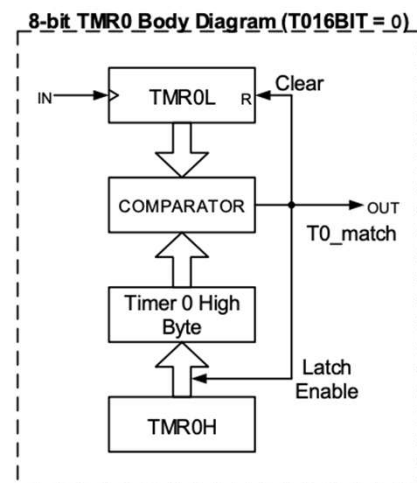
- El divisor de frecuencia hace que la señal de reloj sea mas lenta

- 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

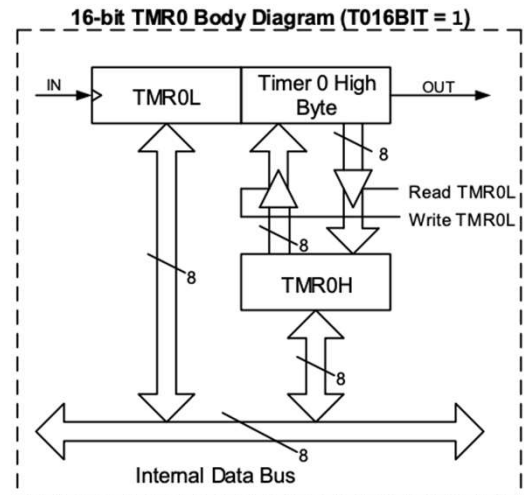
El modo 8 bit del Timer0

- En este modo, **no se desborda** la cuenta ya que tiene un sistema de comparación
- La cuenta máxima es de 0-254 (255 cuentas) cuando el valor de referencia de comparación es 255



El modo de 16 bits

- En este modo, las cuentas van de 0 al 65535 (65536 cuentas)
- Cuando estas en la cuenta máxima (65535) y recibes un pulso, la cuenta se desborda y pasa a 0.
- Mecanismo para precargar la cuenta
- Se temporiza al desborde, trabajar con cuenta inicial.
- La política de carga/lectura de datos de cuenta en este modo se realiza primero trabajando con el TMR0H (el mas significativo) y luego cuando se trabaja con el TMR0L (el menos significativo) se actualizan los 16 bits

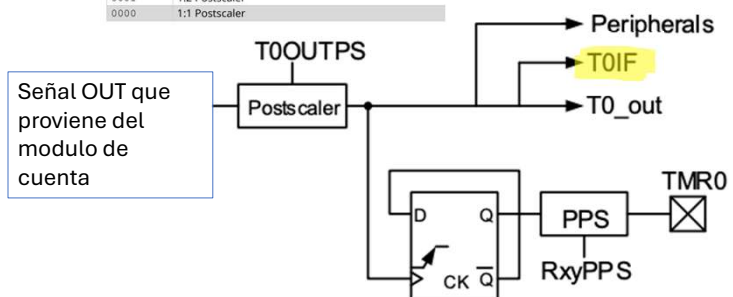


Configuración de el sistema de postscaler del Timer0

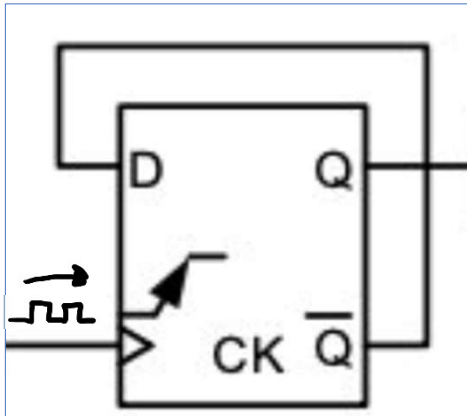
- El postscaler va a contar la cantidad de eventos producidos antes de enviar las atenciones al CPU y demas opciones
- Tenemos tambien una asignación de un pin de salida
- TMR0IF (que es lo mismo que T0IF) se encuentra en el registro PIR3 del módulo de interrupciones vectorizadas (VIC)

- 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



¿Flip flip D modo basculación?



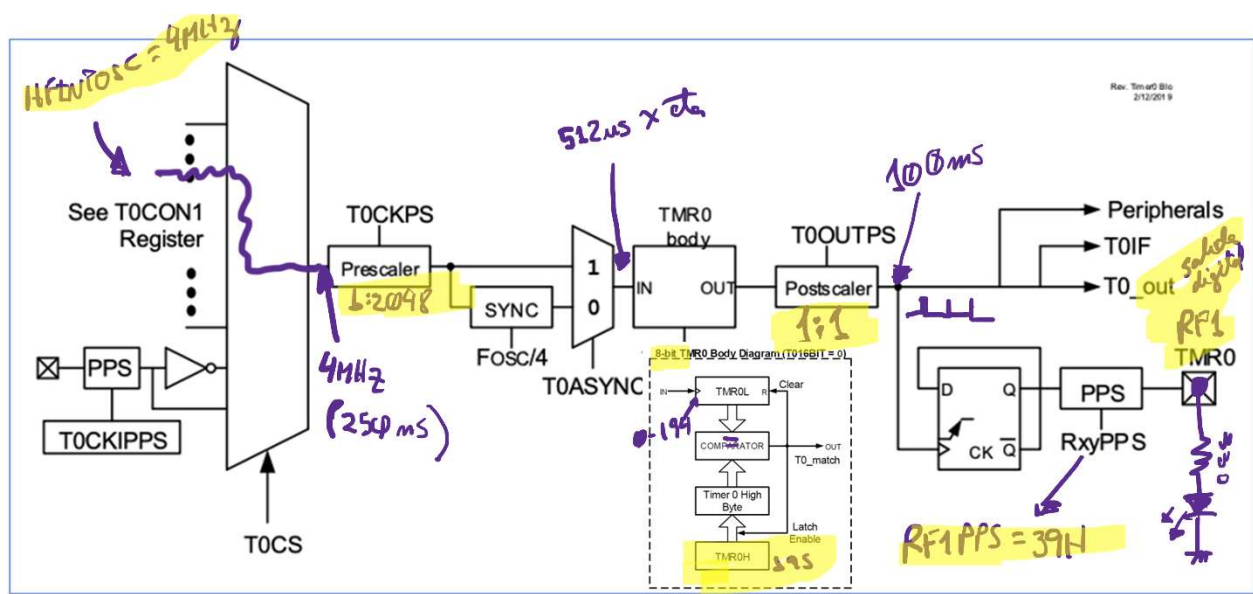
¿Cómo hago para titilar un LED con el Timer0 en modalidad periférico independiente?

- Se va a configurar el Timer0 para que funcione automáticamente y bascule un LED con un periodo de 200ms aproximadamente conectado en su salida (puerto TMR0)

¿Cómo titilar un LED a 200ms con solo el Timer0?

- Se debe de configurar:
 - Fuente de reloj del Timer0
 - Divisor de frecuencia PRESCALER
 - Modo de trabajo (8 o 16 bits), de preferencia emplear el de 8 bits
 - Configurar postscaler (1:1)
 - Configurar el PPS (RF0PPS = 39H en Bank2)
 - Configurar el puerto TMR0 como salida digital (TRISF.0 = 0 y ANSELF.0 = 0 en Bank4)

Configuración del Timer0 para titilar LED a 200ms



Conf de registros para el Timer0

• T0CON0

Name: T0CON0 → 80H
Address: 0x31A

Timer0 Control Register 0

Bit 7 - EN TMR0 Enable

Value	Description
1	The module is enabled and operating
0	The module is disabled

Bit 5 - OUT TMR0 Output

Bit 4 - MD16 16-Bit Timer Operation Select

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

Conf de registros para el Timer0

• T0CON1

Name: T0CON1 → 6BH
Address: 0x31B

Timer0 Control Register 1

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

Código fuente

- Revisar en el siguiente enlace:
 - https://github.com/tocache/Microchip-PIC18F57Q43/blob/main/1AEL0256%202025-1/Examples/NRC9350_Sem05_Deco7S.X/maincode0.s

Fin de la sesión