



Módulos CCP

Características generales

Dos módulos con idéntico funcionamiento.

Registro de captura de 16 bits.
Registro de comparación de 16 bits.
Registro de ciclo de trabajo PWM.

Módulo CCP1.

Consta de dos registros de 8 bits: CCPR1H y CCPR1L.
Registro de control: CCP1CON.
Acción especial: Generada mediante una comparación.
Resetea Timer1.

Módulo CCP2.

Consta de dos registros de 8 bits: CCPR2H y CCPR2L.
Registro de control: CCP2CON.
Acción especial: Generada mediante una comparación.
Resetea Timer1.
Lanza una conversión A/D.



Condiciones de funcionamiento

Temporizadores necesarios.

Modo captura \Rightarrow TMR1
 Modo comparación \Rightarrow TMR1
 PWM \Rightarrow TMR2

Interacción entre los dos módulos CCP.

CCPx	CCPy	Interacción
Captura	Captura	Misma base de tiempos definida en TMR1
Captura	Comparación	La comparación debe configurarse con la acción especial, que borra TMR1
Comparación	Comparación	La(s) comparación(es) deben configurarse con la acción especial, que borra TMR1
PWM	PWM	Las PWMs tendrán la misma frecuencia y tasa de actualización (interrupción TMR2)
PWM	Captura	Ninguna
PWM	Comparación	Ninguna



Registro CCP1CON / CCP2CON (17h / 1Dh)

-	-	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
---	---	-------	-------	--------	--------	--------	--------

bit 5-4 **CCPxX:CCPxY**: Bits menos significativos del PWM

En modo PWM:

Los dos bits menos significativos del ciclo de trabajo.

Los ocho bits más significativos están en CCPxL.

bit 3-0 **CCPxM3:CCPxM0**: Bits de selección del modo CCP a utilizar

0000 = Comparación/Captura/PWM desactivada (reset del módulo CCP)

0100 = Modo de captura, cada flanco descendente

0101 = Modo de captura, cada flanco ascendente

0110 = Modo de captura, cada 4º flanco ascendente

0111 = Modo de captura, cada 16º flanco ascendente

1000 = Modo de comparación, CCPx=1 al igualarse (CCPxIF=1)

1001 = Modo de comparación, CCPx=0 al igualarse (CCPxIF=1)

1010 = Modo de comparación, genera interrupción al igualarse (CCPx invariable, CCPxIF=1)

1011 = Modo de comparación, lanza acción especial (CCPxIF=1, CCPx invariable)

CCP1 resetea TMR1 / CCP2 resetea TMR1 y lanza una conversión A/D

11xx = Modo PWM



Modo captura

Captura en CCPxH:CCPRxL los 16 bits de TMR1.

Cada flanco de bajada en CCPx

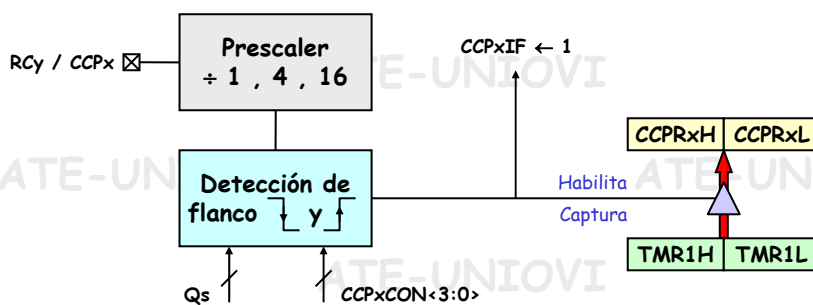
Cada flanco de subida en CCPx

Cada 4º flanco de subida en CCPx

Cada 16º flanco de subida en CCPx

Cada captura da lugar a CCPxIF=1

Diagrama de bloques del modo de captura.



Módulo CCP en modo PWM

©ATE-Universidad de Oviedo

5



Condiciones de funcionamiento en el modo captura

El pin CCPx debe configurarse como entrada.

TMR1 debe funcionar en modo temporizador o en modo contador síncrono.

En modo contador asíncrono puede que la captura no funcione.

Podría generarse una falsa interrupción de captura al cambiar el modo de captura.

Se aconseja deshabilitar CCPxIE al cambiar el modo de captura.

También podría generarse una falsa interrupción al cambiar el prescaler. Además, un cambio *directo* no resetea la cuenta de flancos.

Se evitan estas situaciones reseteando el módulo CCP (CCPxCON ← 0) antes de llevar a cabo un cambio de modo y/o un cambio de prescaler.

CLRf CCP1CON

; Apaga el módulo

CCP

MOVLW NEW_CAPT_PS

; Carga en W el

nuevo valor

; del prescaler

Módulo CCP en modo PWM

©ATE-Universidad de Oviedo

; Carga CCP1CON

6

con este valor



REGISTROS ASOCIADOS CON EL MODULO CCP EN MODO CAPTURA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000



Modo comparación

Cuando CCPRx coincide con TMR1 (en sus 16 bits):

El pin RCy/CCPx se pone a 1

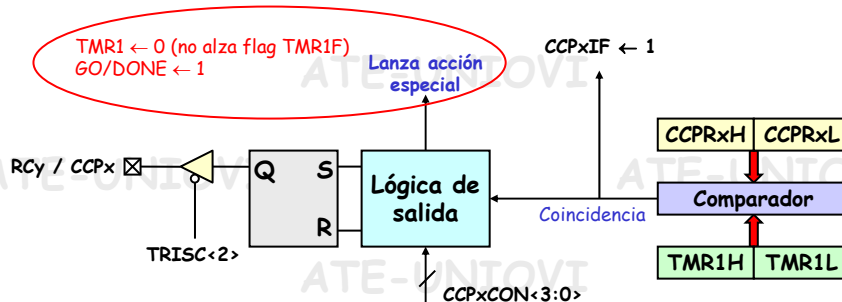
El pin RCy/CCPx se pone a 0

Se genera una interrupción

Lanza una acción especial

Cada coincidencia da lugar a
CCPxIF=1

Diagrama de bloques del modo comparación.





Condiciones de funcionamiento en el modo comparación

El pin CCPx debe configurarse como salida.

TMR1 debe funcionar en modo temporizador o en modo contador síncrono.

En modo contador asíncrono puede que la comparación no funcione.

Cuando se selecciona generar interrupción, el pin CCPx no se ve afectado.

Se puede generar una acción especial por hardware.

CCP1: Resetea TMR1.

CCP2: Resetea TMR1 y lanza una conversión A/D (si está activado).

En ningún caso se pone a 0 el flag TMR1IF



REGISTROS ASOCIADOS CON EL MODULO CCP EN MODO COMPARACION

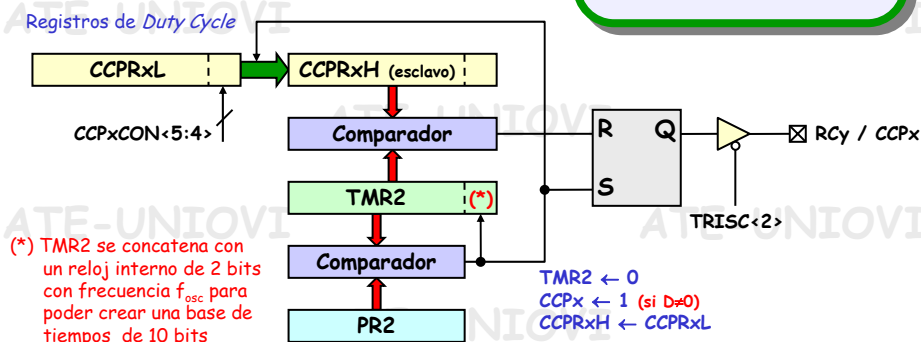
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000



Modo PWM

Genera una salida PWM de 10 bits de resolución

Diagrama de bloques del modo PWM.



Módulo CCP en modo PWM

©ATE-Universidad de Oviedo

11



Condiciones de funcionamiento en el modo PWM

El periodo T se indica en el registro PR2.

$$T = (PR2 + 1) \cdot 4 \cdot T_{OSC} \cdot PRESCALER_{TMR2}$$

El ciclo de trabajo D se indica en CCPRxL y en CCPxCON<5:4>.

$$D = (CCPRxL : CCPxCON<5:4>) \cdot T_{OSC} \cdot PRESCALER_{TMR2}$$

CCPRxH es de sólo lectura en PWM.

Cuando TMR2 (+ 2 bits) se iguala a CCPxH (+ 2 bits), $CCPx \leftarrow 0$.

Existe una resolución (número de bits) máxima que se puede obtener con el funcionamiento en modo PWM.

$$Res = \frac{\log\left(\frac{f_{OSC}}{f_{PWM} \cdot PS_{TMR2}}\right)}{\log(2)} \text{ bits}$$

Máximo número de bits que se pueden usar para definir el ciclo de trabajo.

Módulo CCP en modo PWM

©ATE-Universidad de Oviedo

12



f_{PWM}	1,22kHz	4,88kHz	19,53kHz	78,12kHz	156,3kHz	208,3kHz
PRESALER _{TMR2}	16	4	1	1	1	1
PR2	0xFFh	0xFFh	0xFFh	0x3Fh	0x1Fh	0x17h
Resolución máxima	10 bits	10 bits	10 bits	8 bits	7 bits	5,5 bits

(f_{osc} = 20MHz)

Pasos a seguir para definir una PWM

1. Establecer el periodo T escribiendo en el registro PR2.
2. Establecer el ciclo de trabajo D en CCPxL y en CCPxCON<5:4>.
3. Configurar el pin CCPx como salida.
4. Fijar el prescaler de TMR2 y activar el temporizador. (T2CON)
5. Configurar el módulo CCP para funcionar en modo PWM.



REGISTROS ASOCIADOS CON EL MODULO CCP EN MODO PWM

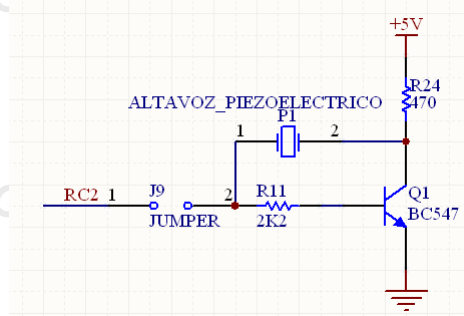
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 Module's Register								0000 0000	0000 0000
92h	PR2	Timer2 Module's Period Register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

PLANTEAMIENTO

Se trata de realizar el diseño del programa encargado de generar el "ring de llamada de un teléfono". Para eso empleamos el módulo CCP1 de un PIC16F877 y la placa de prácticas cuyo esquema eléctrico aparece a continuación (el jumper J9 debe estar puesto).



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

PLANTEAMIENTO

Se emplea un oscilador de 4 MHz para el microcontrolador. El timbre de llamada consiste en un intervalo de 1s de sonido (en el que se van alternando dos tonos de 320Hz y de 480Hz durante 25ms cada uno), seguido de un intervalo de silencio de 2s y así sucesivamente.

1s (sonido)							2s	1s(sonido)
25ms	25ms	25ms	25ms	25ms	25ms	25ms		25ms
320Hz	480Hz	320Hz	480Hz	320Hz	480Hz	Silencio	320Hz...



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

SOLUCION

Se utiliza el módulo CCP1 en modo PWM para generar dos ondas cuadradas con frecuencias de emisión 320Hz y 480Hz. Para reducir armónicos, se hace que éstas ondas sean cuadradas (t en estado alto = t en estado bajo)

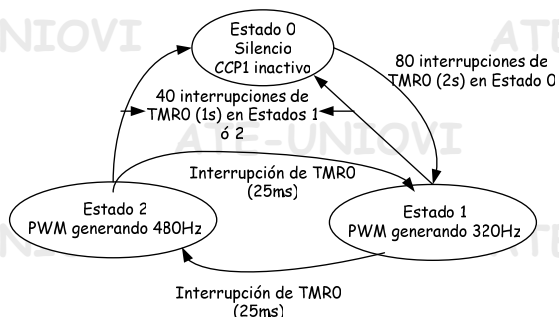
Se utiliza el temporizador TMRO para temporizar intervalos de 25ms, mediante interrupción. Un segundo se contabiliza por 40 interrupciones de TMRO y 2 segundos por 80 interrupciones de TMRO.



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

SOLUCION

Cuando entra una interrupción, se evalúa el estado actual y las posibles modificaciones en el mismo de acuerdo con el siguiente diagrama de estados:





EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

CALCULOS PARA LAS ONDAS PWM

Modo PWM

Frecuencia:

$$1/\text{frecuencia} = (4/\text{fosc}) * \text{Prescaler TMR2} * (\text{PR2}+1)$$

Con fosc=4MHz

Prescaler TMR2 = 16 (elegido)

Resulta: para 320Hz → PR2= 194 (0xC2)

para 480Hz → PR2=129 (0x81)

Ancho de pulso (tiempo en estado alto):

$$1/(2*\text{frecuencia}) = (1/\text{fosc}) * \text{Prescaler TMR2} * (\text{CCPR1L}+2\text{bits})$$

Tomamos los 2 bits a cero

Resultando: para 320Hz → CCPR1L = 97 (0x61)

para 480Hz → CCPR1L = 65 (0x41)



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

CALCULOS PARA TMRO

$$\text{Para temporizar 25ms} = (4/\text{fosc}) * \text{Prescaler TMRO} * (256-\text{Precarga})$$

Resulta: Precarga = 158 (0x9E)



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

PROGRAMA - INICIALIZACION

Algoritmo utilizado para la inicializacion:

Puertos: PORTC con RC2 como salida
TMR0 como temporizador y prescaler de 256
Carga de PR2 para generar 320Hz (0xC2)
CCP1 en modo PWM
TMR2 con prescaler de 16
Inicializamos variable ESTADO con código de onda de 320Hz
Ponemos a cero CONTADOR de interrupciones
Cargamos registro de ancho del pulso (duty) CCPR1L (0x61)
Precarga de TMR0 para desbordar tras 25ms (0x9E)
Borramos flag TOIF y activamos interrupciones de TMR0
Ponemos en marcha TMR2
BUCLE PRINCIPAL INFINITO



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

PROGRAMA - INTERRUPCION DE TMR0

Se comprueba que TOIF=1 si no salimos de inmediato
Salvamos el contexto del programa ppal. (W y STATUS)
Precargamos TMR0
Incrementamos CONTADOR de interrupciones
Verificamos ESTADO actual para evaluar posibles cambios
Si ESTADO==0 (silencio) y CONTADOR == 80 (2s)
ponemos CONTADOR a cero
cambiamos ESTADO a 1 (onda de 320Hz)
activamos módulo CCP en PWM
cargamos PR2 y CCPR1L adecuadamente para 320Hz
Si ESTADO ==1 (onda 320Hz)
cambiamos ESTADO a 2 (onda de 480Hz)
cargamos PR2 y CCPR1L adecuadamente para esa frecuencia



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

PROGRAMA - INTERRUPCION DE TMR0 (continuación)

Si ESTADO ==2 (onda 480Hz)
 cambiamos ESTADO a 1 (onda de 320Hz)
 cargamos PR2 y CCPR1L para generar 320Hz
 Si ESTADO era 1 ó 2
 Si CONTADOR ==40 (ha pasado 1s emitiendo)
 ponemos CONTADOR a cero
 cambiamos ESTADO a 0 (silencio)
 desactivamos CCP2
 Ponemos a cero el flag de TMR0
 Recuperamos contexto del programa ppal.
 Retornamos del programa de tratamiento de interrupción



EJEMPLO PRACTICO - GENERACION DE UN RING TELEFONICO

MODIFICACION PROPUESTA

