



4. Interrupciones Externas, Temporizadores y PWM

Microcontroladores

M. C. Felipe Santiago Espinosa

Abril / 2018



Introducción

- Los AVR tienen una gama amplia de recursos internos.
- En este capítulo revisaremos las interrupciones externas, interrupciones por cambios en los pines I/O y los temporizadores.
- Las señales PWM se generan con los temporizadores, sólo deben configurarse adecuadamente.
- Todos los recursos son manejados por medio de los Registros I/O o Registros I/O Extendidos.
- Los recursos generan eventos que interrumpen a la CPU.



4.1 Interrupciones Externas

Microcontroladores

M. C. Felipe Santiago Espinosa

Abril de 2018



Interrupciones Externas

- Sirven para detectar un estado lógico o un cambio de estado en algunas terminales de entrada del microcontrolador.
- Útiles para monitorear Interruptores, Botones o Sensores con salida a relevador.
- El ATmega328 tiene dos Interrupciones Externas:
 - INT0 (PortD.2, terminal 4)
 - INT1 (PortD.3, terminal 5)
- Se pueden activar por un nivel bajo de voltaje o por flancos de subida o bajada.

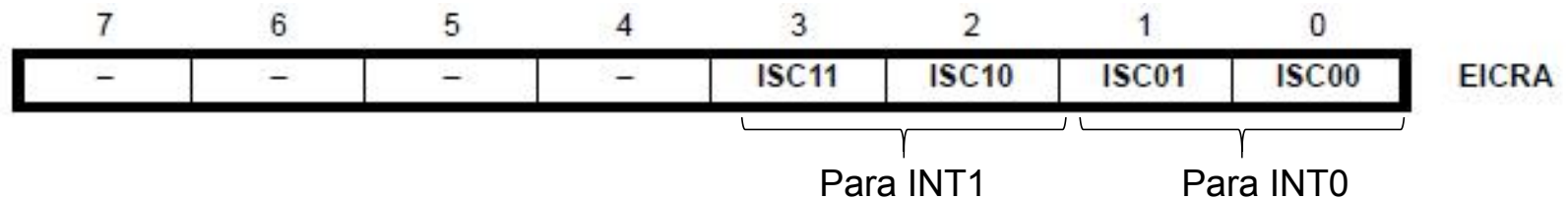


Interrupciones Externas

- Las interrupciones por flanco requieren una señal de reloj en los recursos I/O.
 - Anulado en la mayoría de los modos de bajo consumo.
- Las interrupciones por nivel bajo no requieren señal de reloj (son asíncronas).
 - Adecuadas para despertar al microcontrolador desde cualquier modo de bajo consumo.

Configuración

Las interrupciones externas se configuran con los 4 bits menos significativos del registro EICRA – (*External Interrupt Control Register A*):



ISCx1	ISCx0	Activación de la Interrupción
0	0	Por nivel bajo
0	1	Cualquier cambio lógico
1	0	Por flanco de bajada
1	1	Por flanco de subida

ISC - Interrupt Sense Control

EICRA es un Registro I/O extendido (se ubica en la dirección 0x69).

Habilitación

- El habilitador global de interrupciones es el bit I del registro de Estado (SREG), ubicado en la posición número 7.
- Los habilitadores individuales están en el registro EIMSK (*External Interrupt Mask Register*).



- EIMSK es un Registro I/O ubicado en la dirección 0x1D.

Estado

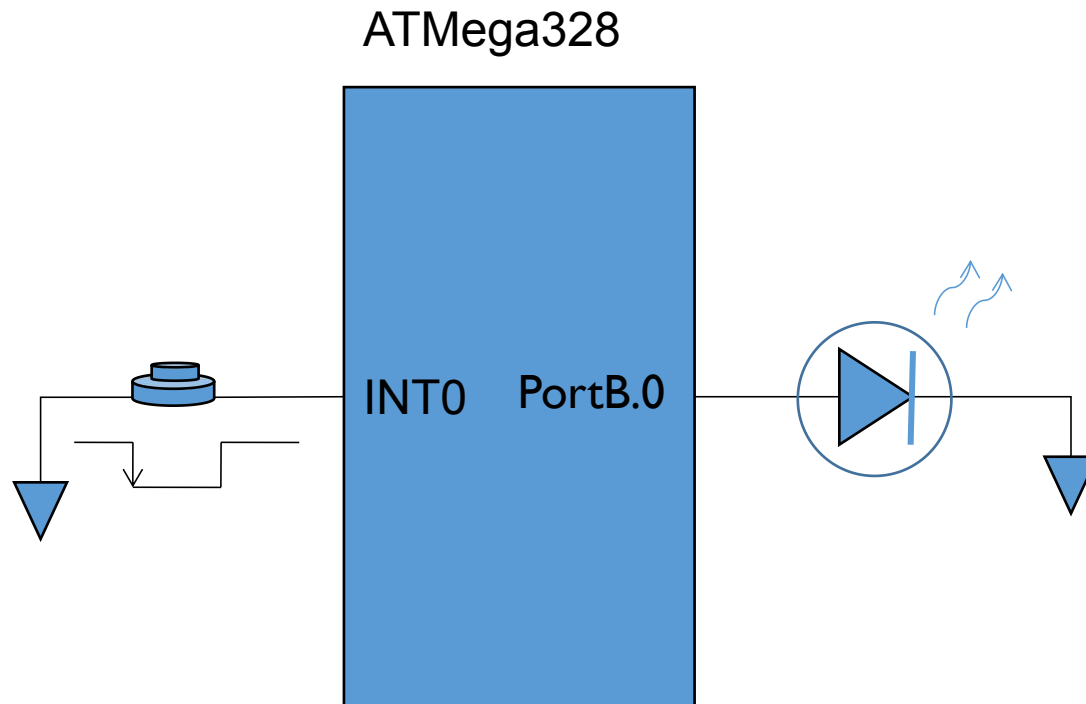
- El estado de las interrupciones se refleja en el registro EIFR (*External Interrupt Flag Register*):



- Las banderas se limpian automáticamente por hardware, cuando la ISR es ejecutada.
 - Aunque el registro puede ser consultado por software, no tiene ningún sentido práctico.
- EIMSK es un Registro I/O ubicado en la dirección 0x1C.

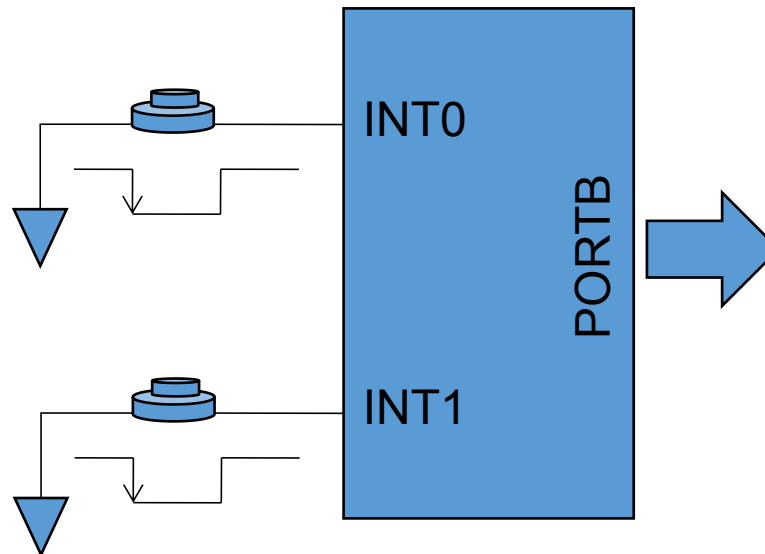
Ejercicio 1: Prueba de un botón

- Conmutar el bit menos significativo del puerto B cada que se presiona un botón.



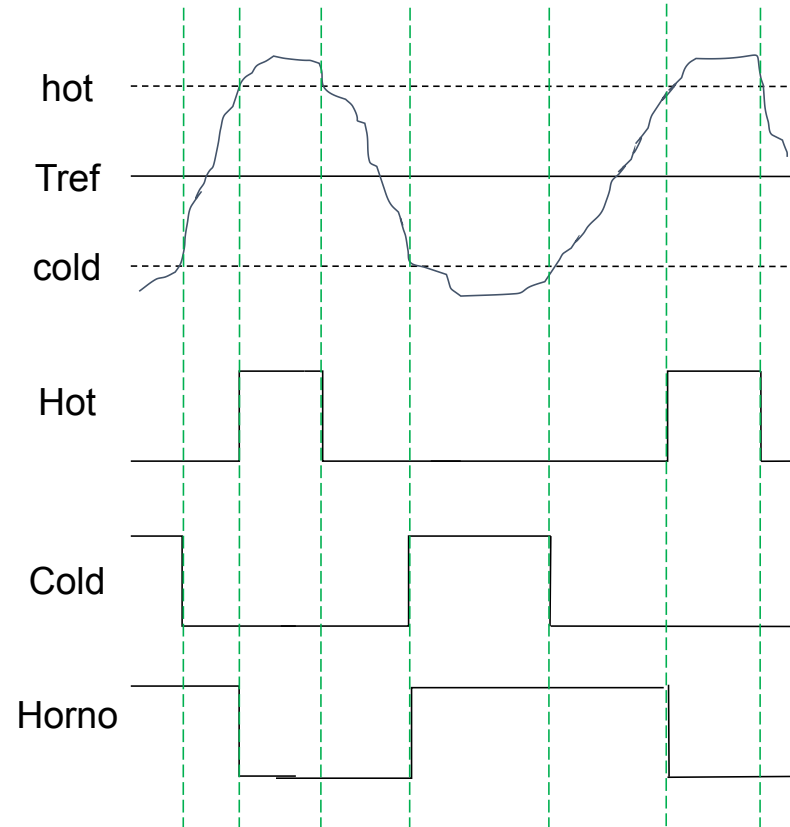
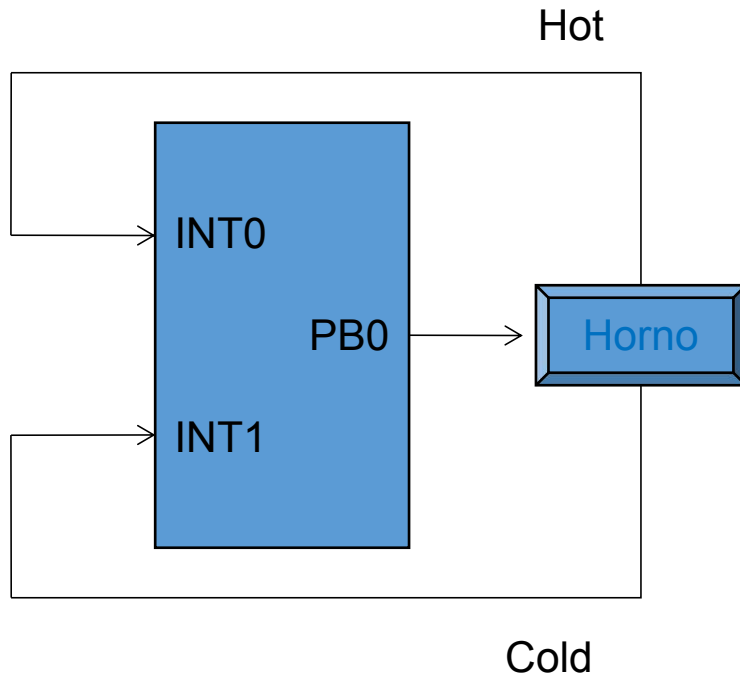
Ejercicio 2: Contador de eventos

- El puerto B se incrementará cada vez que se presiona el botón 1 y tendrá un decremento al presionar al botón 2.
- Deberá contar de 0 a 9, pasando de 9 a 0 (UP) ó de 0 a 9 (DOWN).



Ejercicio 3: Control simple

Se pretende mantener la temperatura de un horno en una temperatura de referencia.





4.2 Interrupciones por Cambios en los Pines

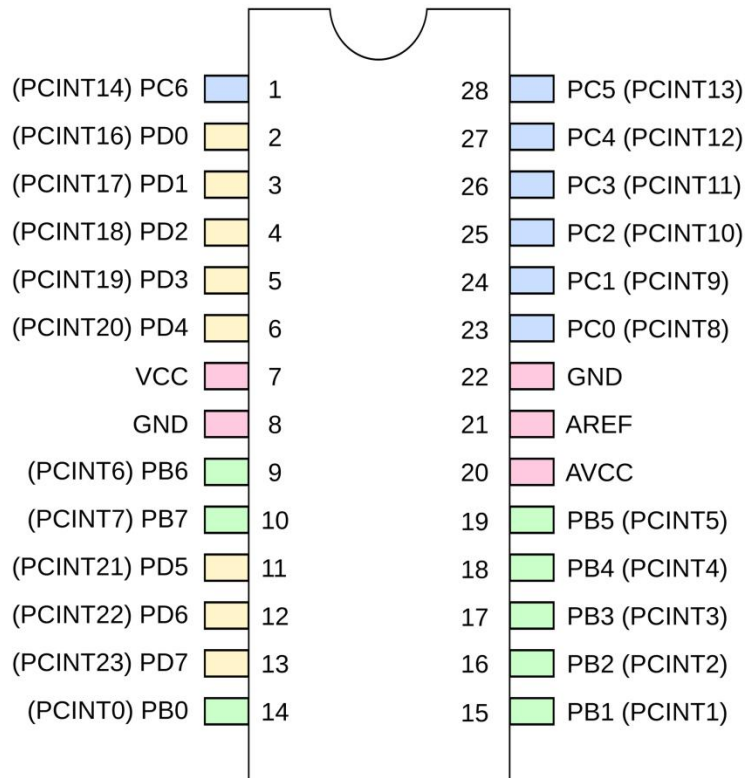
Microcontroladores

M. C. Felipe Santiago Espinosa

Abril de 2018

Interrupciones por cambios

Las interrupciones PCIE0, PCIE1 y PCIE2 son provocadas por cualquier cambio en las terminales de los puertos.



Interrupción	Pines que monitorea
PCINT0	PB0 .. PB7
PCINT1	PC0 .. PC6
PCINT2	PD0 .. PD7

Las interrupciones son asíncronas.

Habilitaciones individuales

- Un pin puede provocar una interrupción si está habilitada en forma individual.
- Se habilita en los registros PCMSKx (*Pin Change Mask*):

7	6	5	4	3	2	1	0	
PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2

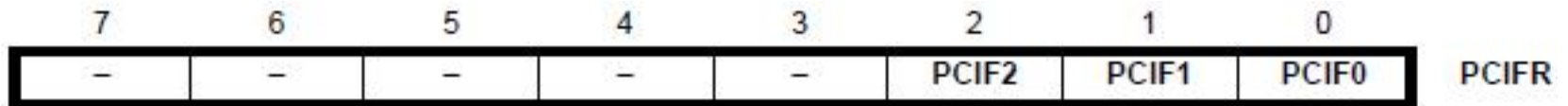
- En la ISR debe emplearse una máscara para determinar qué pin provocó el cambio.

Habilitación y estado

- Establecida la máscara, la interrupción por puerto se habilita en el registro PCICR (*Pin Change Interrupt Control Register*).

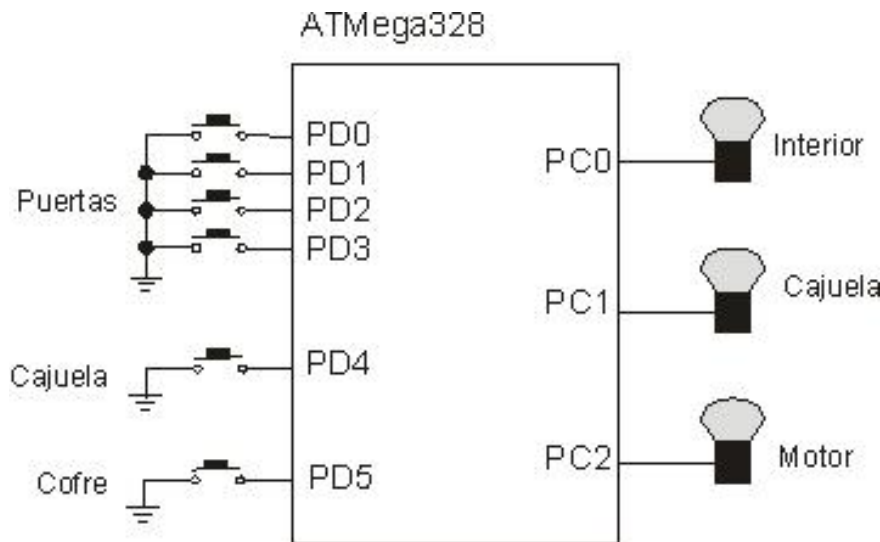


- El estado de cada interrupción puede verse en el registro de banderas PCIFR (*Pin Change Interrupt Control Register*).



Ejercicio 1: Puerta de un automóvil.

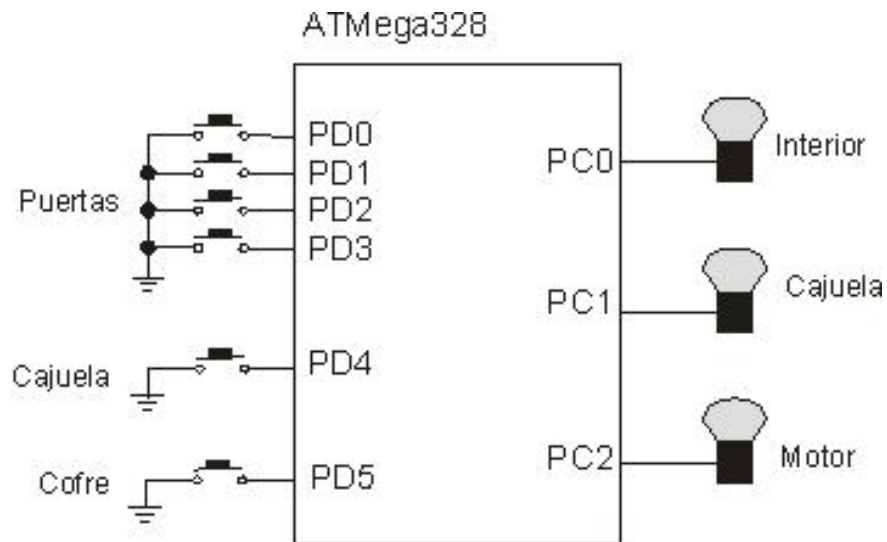
En las 4 puertas, cajuela y cofre de un automóvil se han puesto botones para detectar si están abiertas o cerradas. El botón está presionado con las puertas cerradas.



- Si alguna de las 4 puertas está abierta deberá encenderse la luz interior.
- Si la cajuela está abierta, además de la luz interior debe encenderse la luz de la cajuela.
- Si el cofre está abierto, además de la luz interior debe encenderse la luz del motor.

Suponga que todas las puertas están cerradas cuando el sistema se energiza.

Ejercicio 1: Puerta de un automóvil.

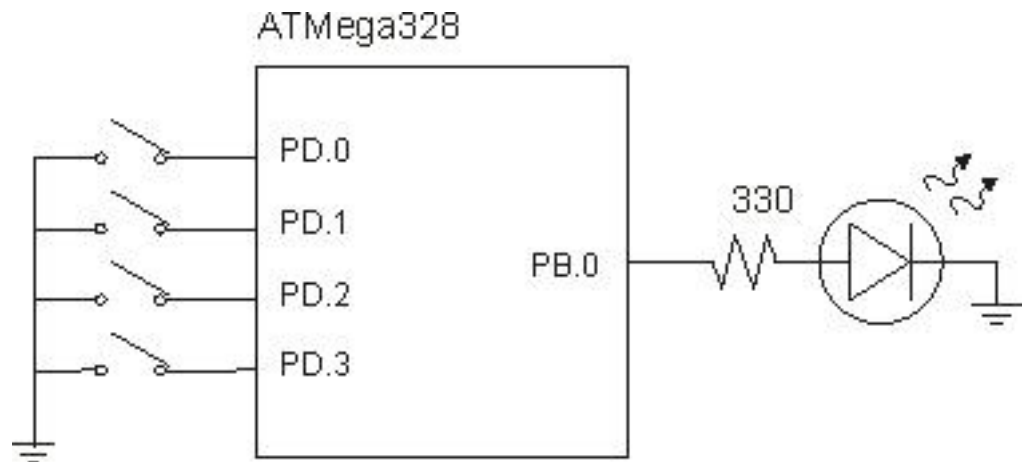


PD5	PD4	PD3-PD0	PC2	PC1	PC0
1	1	X	1	1	1
0	1	X	0	1	1
1	0	X	1	0	1
0	0	!= 0	0	0	1
0	0	0	0	0	0

Ejercicio 2: Clave en Código Gray

Realice un sistema que active una salida cuando detecte la secuencia en código Gray de 4 bits siguiente:

Sec.	Código
0	0000
1	0001
2	0011
3	0111
4	0110
5	1110
6	1111



Sugerencia: Considere 7 estados válidos. En cada cambio observe si el nuevo estado es válido y acorde al anterior, o si se ha perdido la secuencia y alcanzado un estado inválido.



4.3 Temporizadores

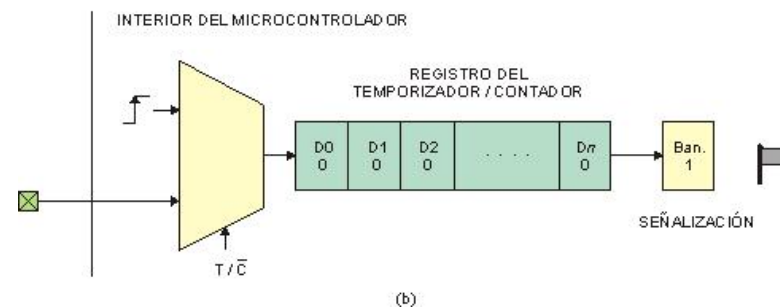
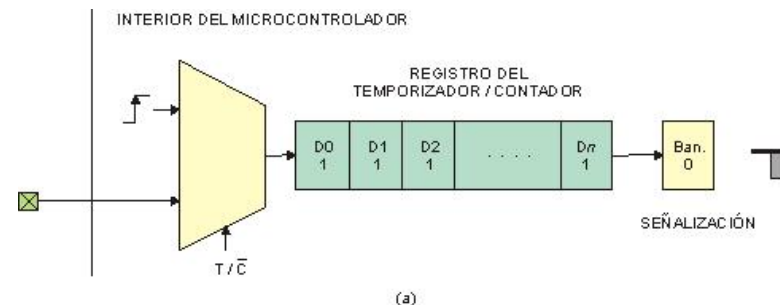
Microcontroladores

M. C. Felipe Santiago Espinosa

Abril de 2018

Temporizadores/Contadores

- Un Timer/Counter es un registro de n bits que se incrementa automáticamente en cada ciclo de reloj o por eventos externos.



- Sus desbordamientos indican que ha transcurrido un intervalo de tiempo o un número predefinido de eventos.

Temporizadores/Contadores

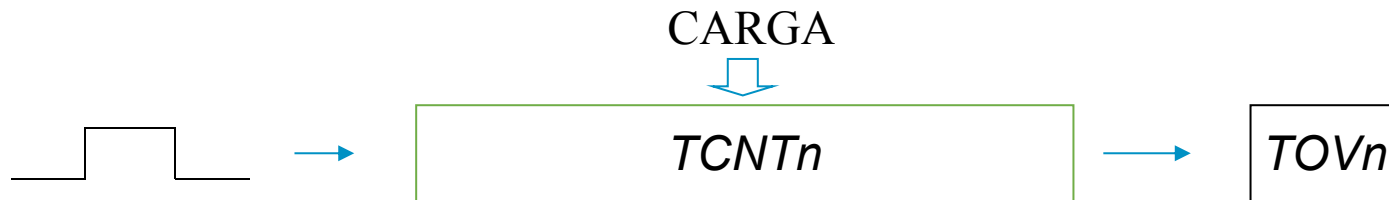
El ATmega328 tiene tres temporizadores:

Temporizador	Tamaño	Registros
Timer 0	8 bits	TCNT0
Timer 1	16 bits	TCNT1H, TCNT1L
Timer 2	8 bits	TCNT2



Eventos en los temporizadores

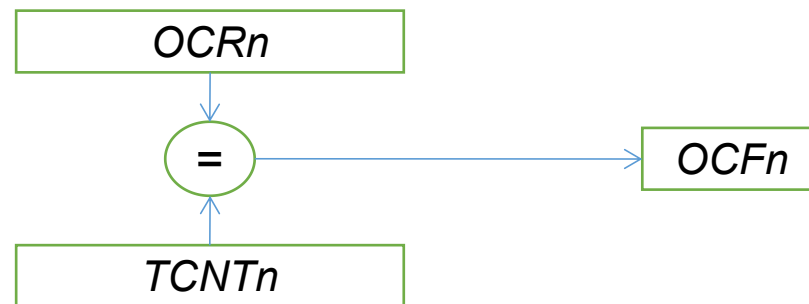
Desbordamiento:



$$MAXVAL = 2^{\text{Tamaño } (TCNTn)} - 1$$

La bandera $TOVn$ se pone en alto cuando el registro $TCNTn$ pasa de $MAXVAL$ a 0.

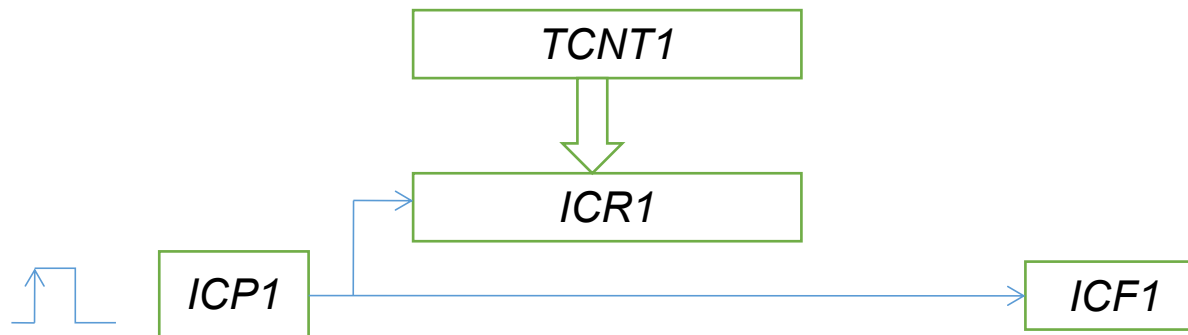
Coincidencia por comparación (Compare match):



La bandera $OCFn$ se pone en alto cuando los registros $OCRn$ y $TCNTn$ tienen el mismo valor.

Eventos en los temporizadores

Captura de entrada (Input Capture):



La bandera *ICF1* se pone en alto cuando ocurre un cambio predefinido en la terminal *ICP1* y el valor de *TCNT1* es guardado en *ICR1*.

Eventos en los temporizadores

Temporizador	Eventos	Banderas	Registro
TCNT0	Desbordamiento	TOV0	TIFR0
	Comparación (2)	OCF0A, OCF0B	
TCNT1	Desbordamiento	TOV1	TIFR1
	Comparación (2)	OCF1A, OCF1B	
	Captura de entrada	ICF1	
TCNT1	Desbordamiento	TOV0	TIFR2
	Comparación (2)	OCF2A, OCF2B	

Cada temporizador tiene su registro de banderas.



Detección y respuesta a los eventos

Existen 3 formas de detectar eventos y actuar ante ellos.

Sondeo (*polling*).

- El programa principal evalúa de manera frecuente el estado de las banderas.
- Ejemplo: Se espera un evento por desbordamiento del Timer 0.

```
LOOP:  LDS    R16, TIFR0  
        SBRS   R16, TOV0  
        RJMP   LOOP
```

```
while (!( TIFR0 & 1 << TOV0))  
    ;
```

- Bajo este esquema, las banderas se deben limpiar por software, re-escribiendo un 1 lógico.

Uso de Interrupciones

- Las interrupciones se habilitan en el registro TIMSKn (*Timer Interrupt Mask Register*) de cada temporizador.
- También se debe poner en alto al habilitador global de interrupciones (bit I en SREG).
- Ejemplo: Configurar el timer 2 para que pueda detectar eventos de coincidencia con el comparador A:

```
LDI  R16, 1 << OCIE2A
SDS  TIMSK2, R16
SEI
```

```
TIMSK2 = 1 << OCIE2A;
sei();
```

- La bandera que generó la interrupción se limpia automáticamente al concluir la ISR.

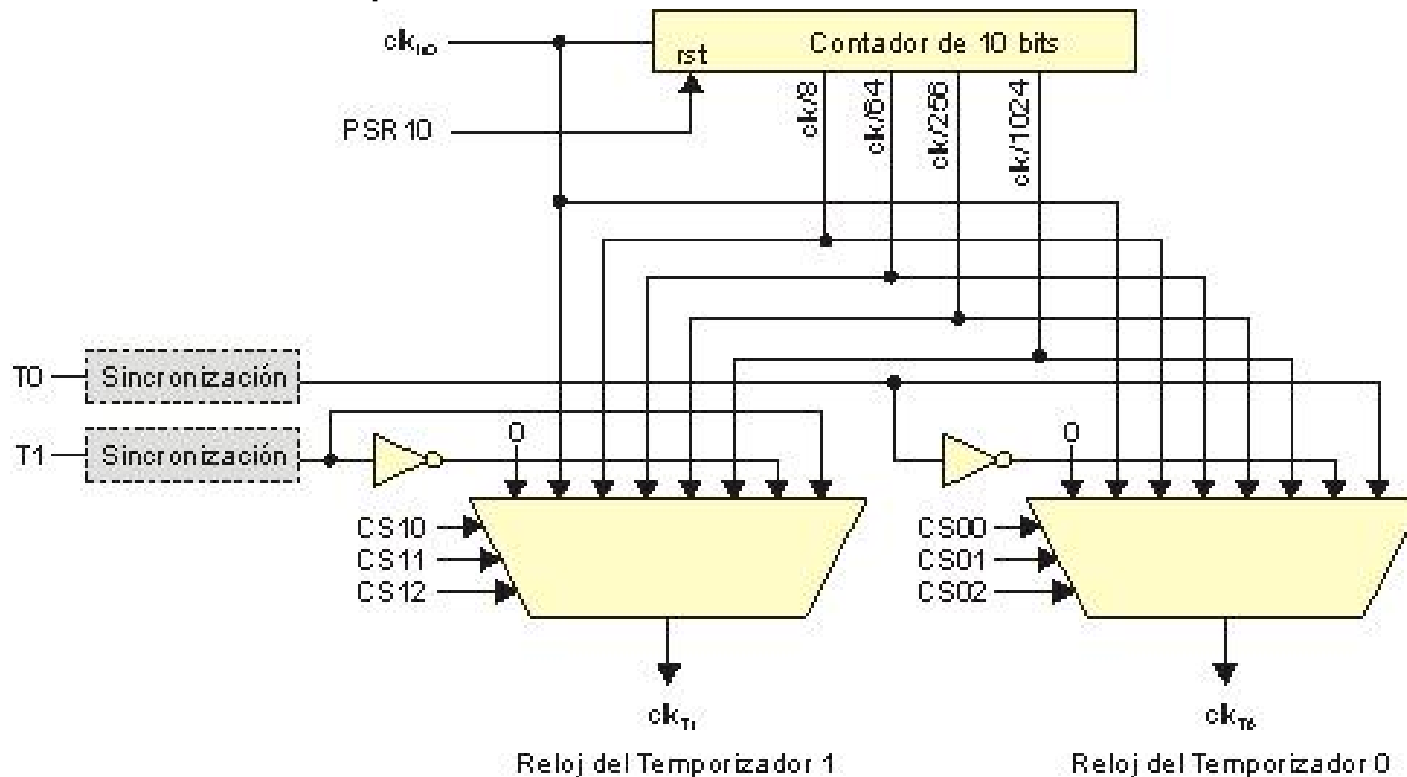


Reacción automática sobre eventos

- Los timers soportan la posibilidad de reaccionar sólo con hardware ante eventos de coincidencia.
- Algunas terminales relacionadas pueden ponerse en alto, en bajo o conmutarse de manera automática.
 - Útil para generar señales con frecuencia constante o para la generación de PWM.
- La atención a eventos se realiza de manera paralela a la ejecución del programa principal.

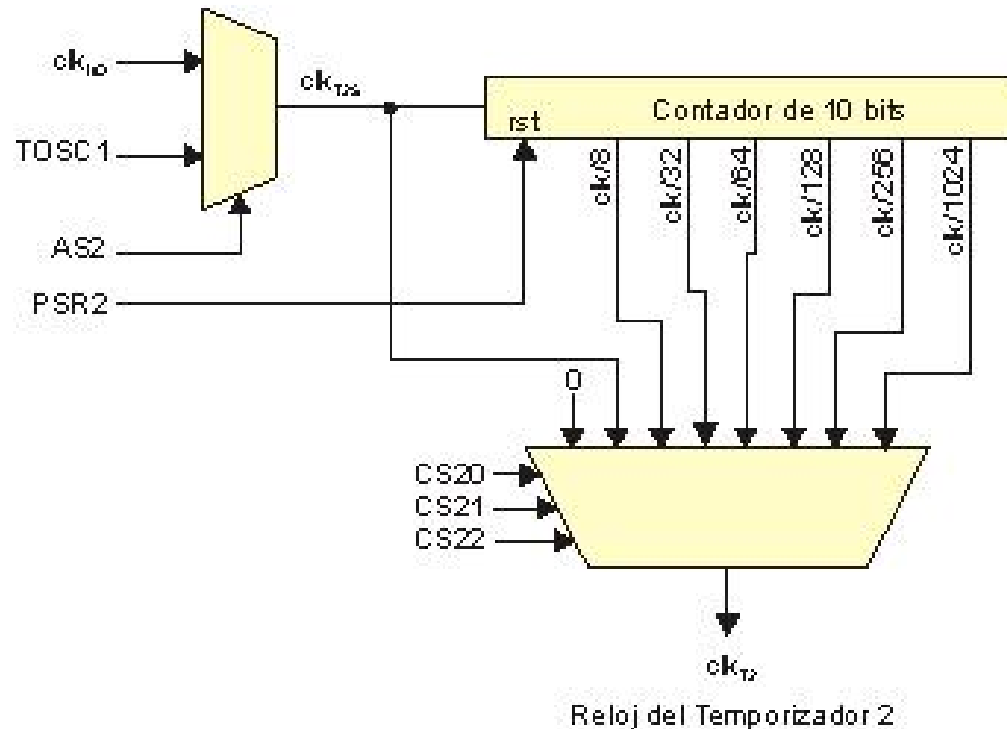
Pre-escalador

- Un pre-escalador es un divisor de frecuencia con diferentes salidas conectado a un multiplexor.
- Los temporizadores 0 y 1 comparten al pre-escalador, pero con selección independiente.



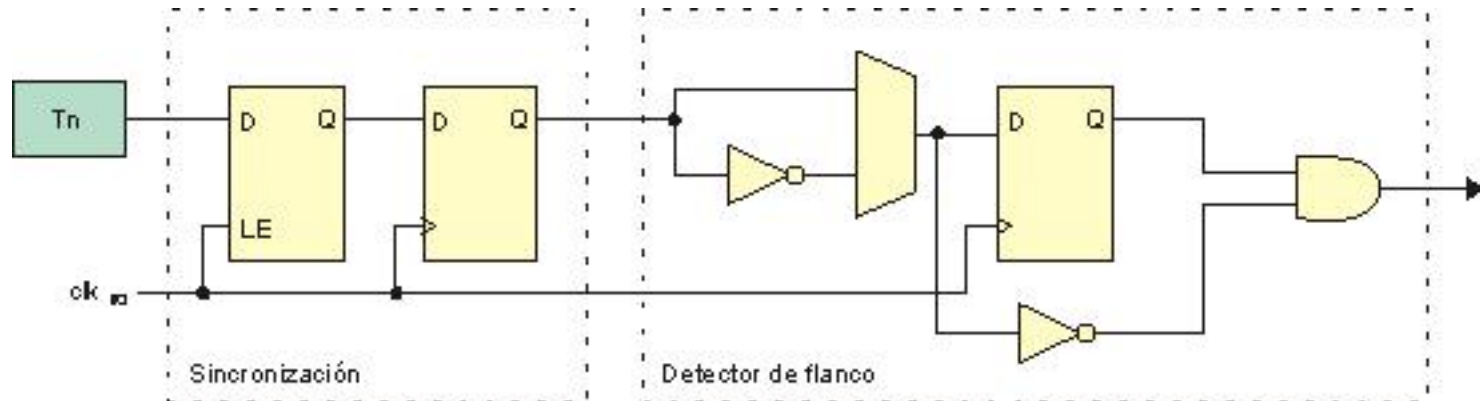
Pre-escalador

- El temporizadores 2 tiene su propio pre-escalador, con la posibilidad de ser manejado por un oscilador externo.



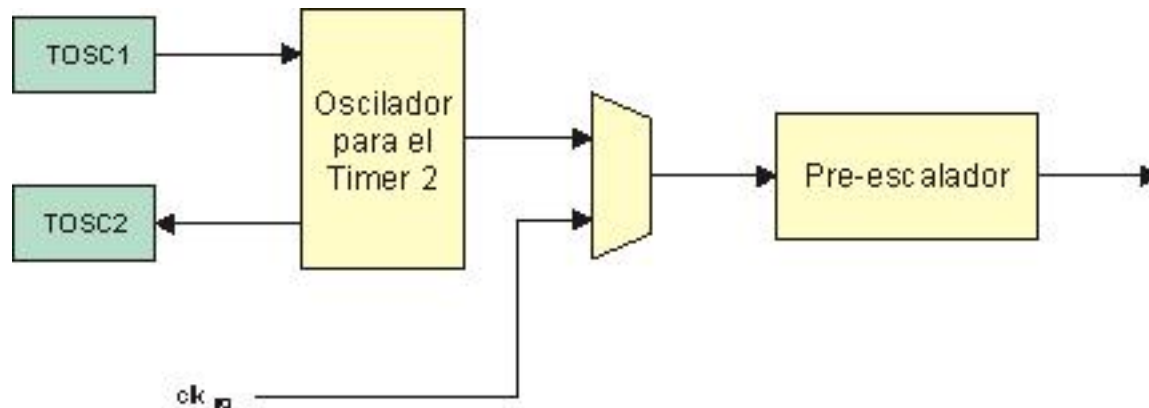
Temporización externa

- Los temporizadores 0 y 1 pueden operar con una señal externa (T0 ó T1). En estos casos se les denomina contadores de eventos.
- Se puede configurar para detectar flancos de bajada o subida.
- Los eventos externos se sincronizan con la señal de reloj interna, su frecuencia máxima es de $f_{CLKIO}/2$.



Temporización externa

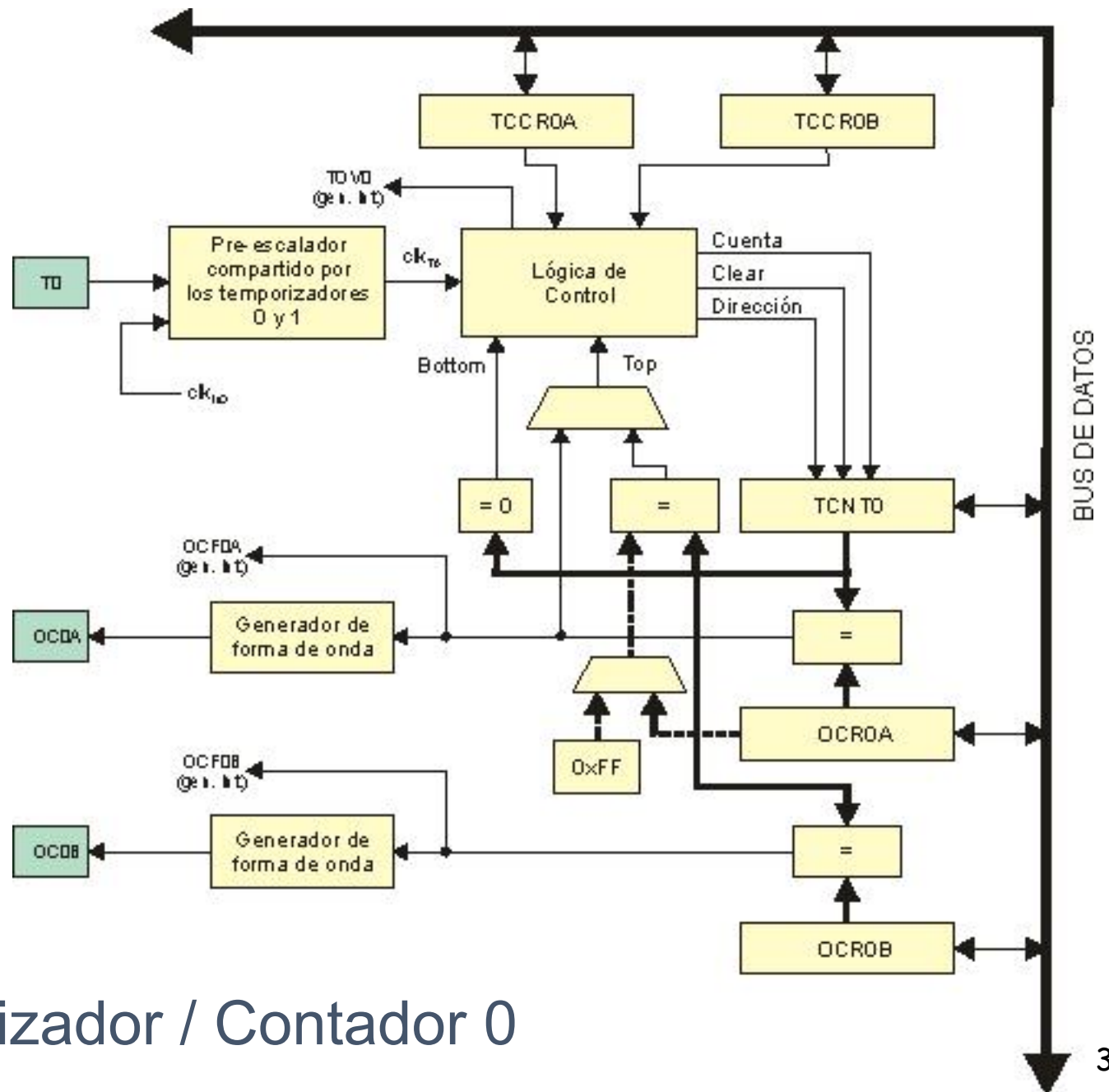
- El temporizador 2 puede ser manejado por un reloj asíncrono externo.
- Optimizado para trabajar con un cristal de 32.768 KHz. En combinación con su pre-escalador genera fracciones o múltiplos de segundos reales.
- El temporizador 2 opera a una frecuencia independiente al resto del sistema.





Organización y Registros del Temporizador/Contador 0

- **TCNT0** (*Timer/Counter 0*), es el registro de 8 bits del temporizador/ contador 0.
- **OCR0A** (*Output Compare Register A*), es uno de los registros con el que continuamente se compara el temporizador para generar eventos de coincidencias por comparación.
- **OCR0B** (*Output Compare Register B*), es el otro registro con el que continuamente se compara el temporizador para generar eventos de coincidencias por comparación.
- Los registros **OCR0A** y **OCR0B** también son usados en los modos PWM.



Temporizador / Contador 0

Registros de control

- Se tienen dos registros para el control del temporizador/contador 0:
TCCR0A y **TCCR0B** (*Timer/Counter Control Register*):

TCCR0A:

7	6	5	4	3	2	1	0
COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00

TCCR0B:

7	6	5	4	3	2	1	0
FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00

Modos de operación

- **Bits WGM0[2:0]** (Waveform Generation Mode): Para definir el modo de operación, se tienen las siguientes combinaciones:

Modo	WGM02	WGM01	WGM00	Descripción
0	0	0	0	Normal
1	0	0	1	PWM con fase correcta (MAX = 0xFF)
2	0	1	0	CTC (<i>clear timer on compare match</i>) – El máximo está en OCR0A
3	0	1	1	PWM rápido (MAX = 0xFF)
4	1	0	0	Reservado
5	1	0	1	PWM con fase correcta (MAX = OCR0A)
6	1	1	0	Reservado
7	1	1	1	PWM rápido (MAX = OCR0A)

Modos de operación

- **Modo 0:** En este modo es un temporizador sin alguna funcionalidad extra y sólo genera eventos de desbordamiento.
- **Modos 1 y 5:** Modos de PWM en fase correcta, el temporizador automáticamente se incrementa de cero a máximo y de máximo a cero, al coincidir con los registros de comparación conmutará una salida para la generación de una señal PWM.
- **Modo 2:** Es un modo CTC (*clear timer on compare match*), en donde el temporizador se limpia automáticamente después de una coincidencia con el registro **OCR0A**.
- **Modos 3 y 7:** Modos de PWM rápido, el temporizador automáticamente se incrementa de cero a máximo, al coincidir con los registros de comparación conmutará una salida para la generación de una señal PWM.

- Bits **COM0A[1:0]** y **COM0B[1:0]** (*Compare Output Mode*): Con estos bits se configura una respuesta automática en OC0x, ante eventos de comparación.

Para los modos que no son PWM se tiene:

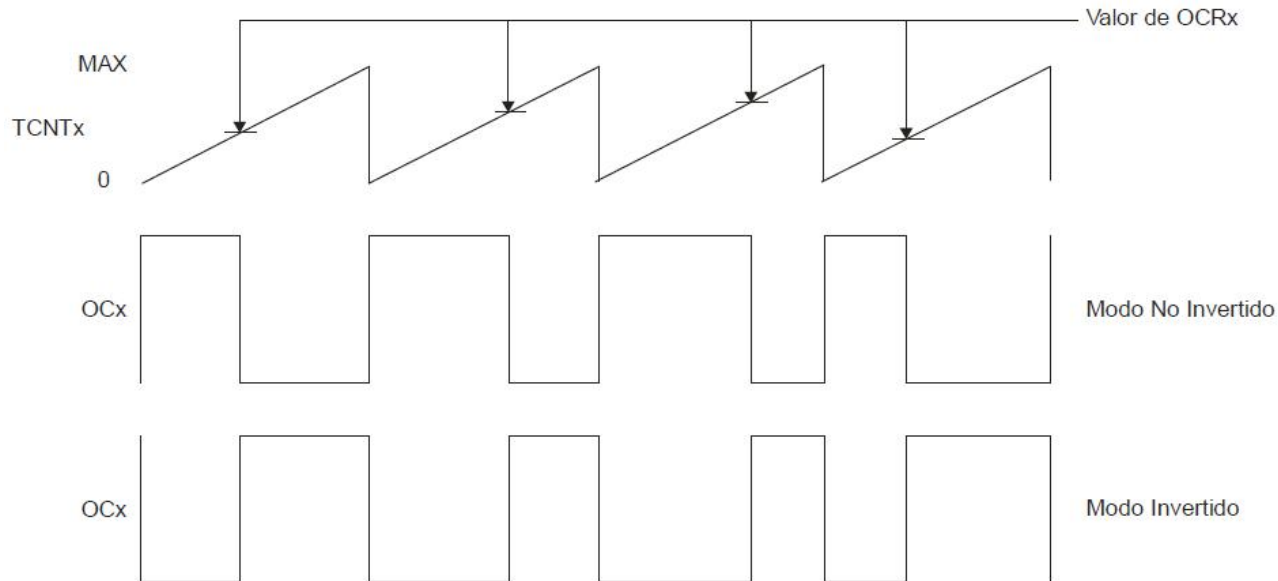
COM0x1	COM0x0	Descripción
0	0	Operación normal - OC0x desconectado
0	1	Conmuta OC0x, sobre una coincidencia por comparación.
1	0	Limpia OC0x, sobre una coincidencia por comparación.
1	1	Ajusta OC0x, sobre una coincidencia por comparación.

Para los modos PWM:

COM0x1	COM0x0	Descripción
0	0	Operación normal - OC0x desconectado, no hay salida PWM.
0	1	Reservado (sin uso)
1	0	Modo No Invertido
1	1	Modo Invertido

(x puede ser A o B)

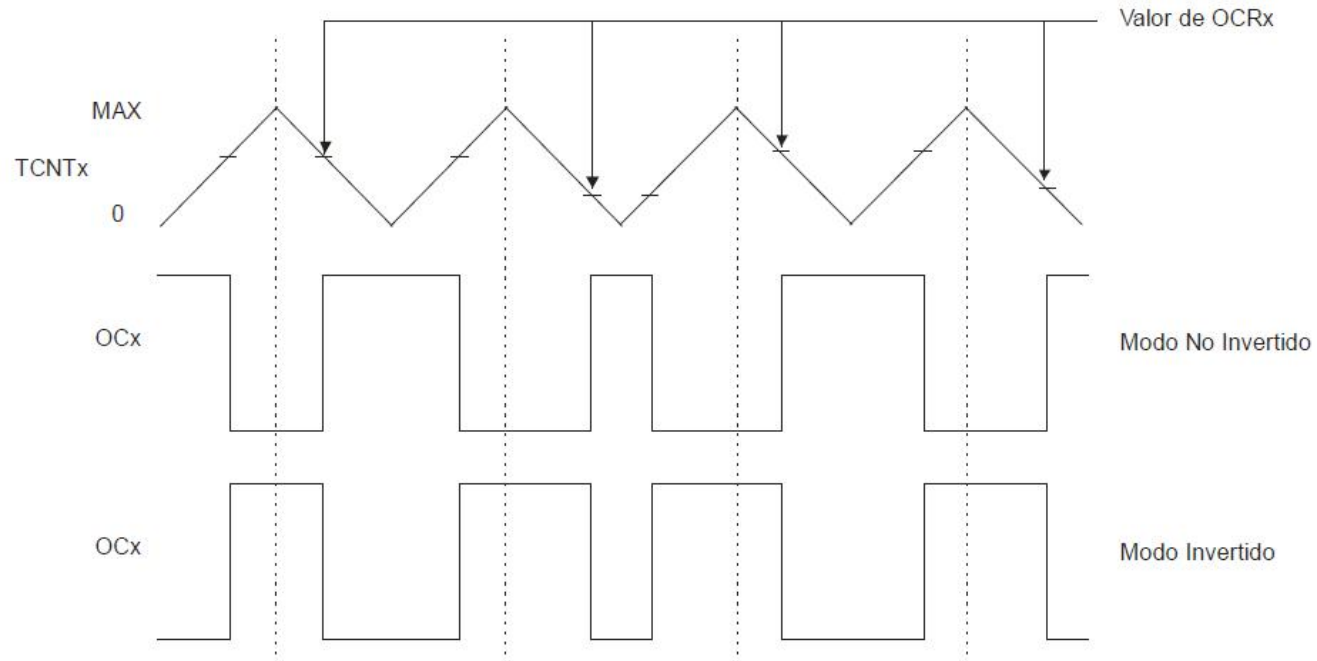
PWM Rápido



$$f_{PWM} = \frac{f_{clk}}{MAX + 1}$$

El registro OCRx se actualiza cuando el temporizador pasa de su valor máximo a cero.

PWM con Fase Correcta



$$f_{PWM} = \frac{f_{clk}}{2 (MAX + 1)}$$

Las escrituras en el registro OCRx se realizan cuando TCNTx alcanza su valor máximo.

- **Bits FOC0A y FOC0B:** forzan u obligan un evento de comparación, si se configuró una respuesta automática, ésta también va a realizarse. **FOC0A** se relaciona con el registro **OCR0A** y **FOC0B** con el registro **OCR0B**.
- **Bits CS0[2:0] (Clock Select):** Bits de selección de la fuente de reloj.

CS02	CS01	CS00	Descripción
0	0	0	Sin fuente de reloj (temporizador detenido)
0	0	1	CLK I/O (Sin pre-escalador)
0	1	0	CLK I/O/8 (Desde el pre-escalador)
0	1	1	CLK I/O/64 (Desde el pre-escalador)
1	0	0	CLK I/O/256 (Desde el pre-escalador)
1	0	1	CLK I/O/1024 (Desde el pre-escalador)
1	1	0	Fuente externa (en T0), con flanco de bajada
1	1	1	Fuente externa (en T0), con flanco de subida

Eventos del Temporizador/Contador 0

Los eventos se ven reflejados en el registro **TIFR0** (*Timer/Counter Interrupt Flag Register*), se tienen tres eventos:

7	6	5	4	3	2	1	0
-	-	-	-	-	OCF0B	OCF0A	TOV0

- **TOV0** (*Timer/Counter Overflow 0*): Desbordamiento del Temporizador/Contador 0.
- **OCF0x** (*Output Compare Match Flag 0x*): Indica una coincidencia en la comparación.
 - Se tienen dos banderas por que el temporizador es comparado con dos registros diferentes: **OCR0A** y **OCR0B**.

Nota: Las banderas se limpian automáticamente si se emplean interrupciones o escribiendo un 1 si se usa sondeo.

Habilitación de Interrupciones

Las interrupciones se activan en el registro **TIMSK0** (*Timer/Counter Interrupt Mask Register 0*). Los habilitadores individuales son:

7	6	5	4	3	2	1	0
-	-	-	-	-	OCIE0B	OCIE0A	TOIE0

- **TOIE0** (*Timer/Counter 0 Overflow Interrupt Enable*): Habilita la interrupción por desbordamiento del Temporizador/Contador 0.
- **OCIE0x** (*Timer/Counter 0 Output Compare Match x Interrupt Enable*): Habilita la interrupción por una coincidencia en la comparación.
 - Se tienen dos habilitaciones porque el temporizador es comparado con dos registros diferentes: **OCR0A** y **OCR0B**.



Vectores de interrupción

- Los **vectores de interrupción** para los eventos del temporizador/ contador 0 son:

Dirección	Evento	Descripción
0x001C	TIMER0_COMPA	El Timer 0 coincide con el comparador A
0x001E	TIMER0_COMPB	El Timer 0 coincide con el comparador B
0x0020	TIMER0_OVF	Desbordamiento del Timer 0



Ejercicio 1:

1. Escriba una secuencia de instrucciones para generar una señal de 5 KHz (Suponer que se está usando el oscilador interno a 1 MHz). Utilizando:
 - a. Desbordamientos atendidos por interrupción (salida en PD0).
 - b. Respuesta automática ante coincidencias por comparación y modo CTC (salida en OC0A, PD6).

$$F = 5 \text{ KHz} \Rightarrow T = 0.2 \text{ ms} = 200 \text{ } \mu\text{S}$$

$$T_H = T_L = 100 \text{ } \mu\text{S}$$



Ejercicio 2:

- Configure al ATmega328P para que genere un tono de 400 Hz (aproximadamente), mientras se mantenga presionado un botón.

$$F = 400 \text{ Hz} \Rightarrow T = 2.5 \text{ ms} = 2500 \mu\text{S}$$

$$TH = TL = 1250 \mu\text{S}$$

$$\text{Dividiendo entre 8: } TH / 8 = 156.25 \mu\text{S} \sim 156$$

Recursos a emplear:

Botón – Cambio en pin en PC0.

Tono – Respuesta automática en OC0A (PD6).



Ejercicio 3:

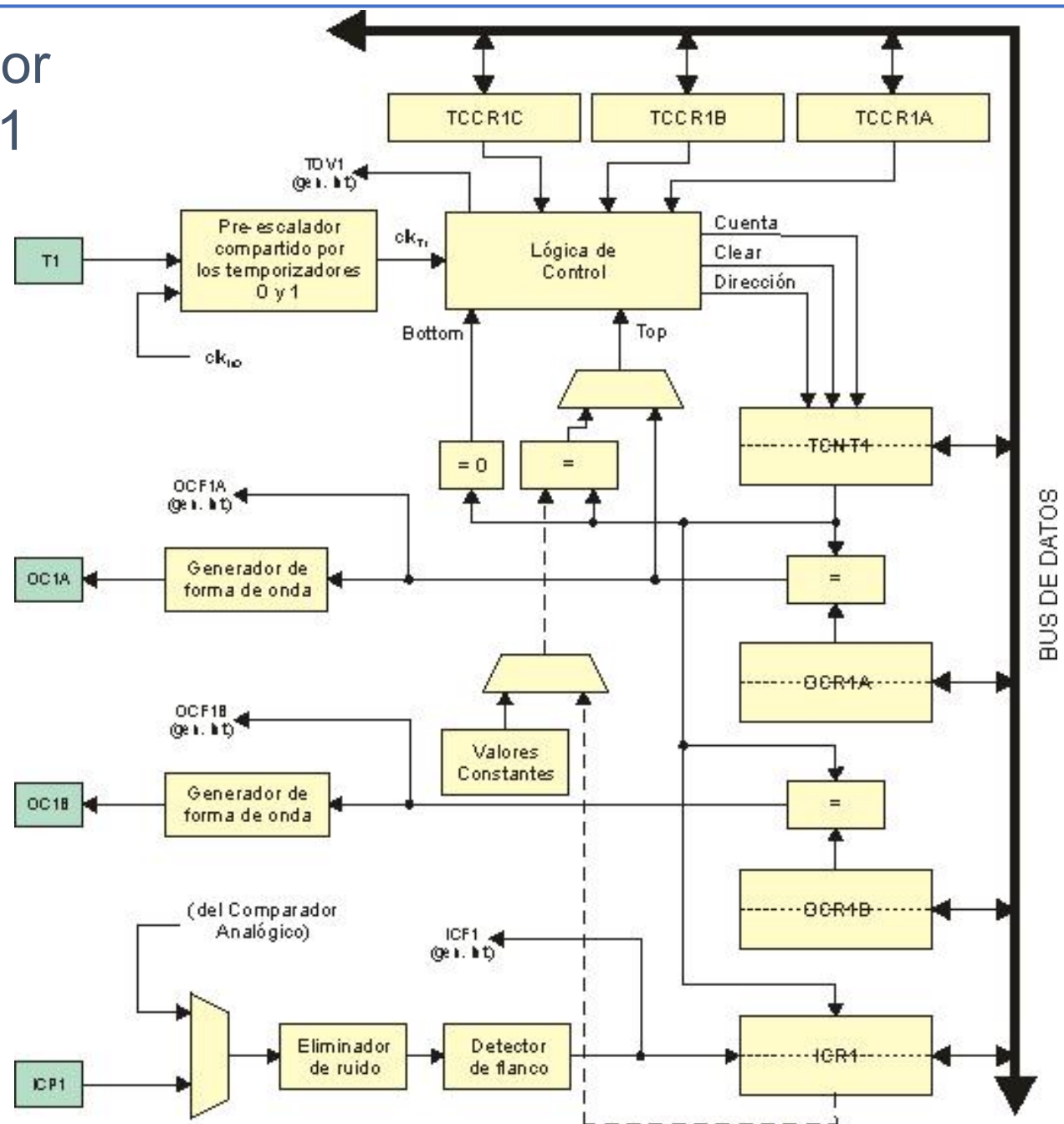
- Genere una señal PWM rápida con salida en modo No Invertido, donde el ancho del pulso esté determinado por el valor del puerto B.



Organización y Registros del Temporizador/Contador 1

- Puesto que el temporizador 1 es de 16 bits, muchos de sus registros se componen de dos registros de 8 bits. Puede manejar un evento de desbordamiento, dos de comparación y uno de captura.
- **TCNT1** (*Timer/Counter 1*), el temporizador/contador 1 se compone con dos registros de 8 bits: **TCNT1H** y **TCNT1L**.
- **OCR1A** (*Output Compare Register 1 A*), registro A para la continúa comparación del temporizador 1. También se forma con dos registros de 8 bits: **OCR1AH** y **OCR1AL**.
- **OCR1B** (*Output Compare Register 1 B*), registro B para la continúa comparación del temporizador 1. Formado por: **OCR1BH** y **OCR1BL**.
- **ICR1** (*Input Capture Register 1*), registro para capturar el valor de **TCNT1** ante un evento en la terminal **ICP1**. Se forma con: **ICR1H** y **ICR1L**.

Temporizador /Contador 1



Registros de control

- Se tienen tres registros para el control del temporizador/contador 1:
TCCR1A, **TCCR1B** y **TCCR1C** (Timer/Counter Control Register):

TCCR1A:

7	6	5	4	3	2	1	0
COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10

TCCR1B:

7	6	5	4	3	2	1	0
ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10

TCCR1C:

7	6	5	4	3	2	1	0
FOC1A	FOC1B	-	-	-	-	-	-

Bits WGM1[3:0] (Waveform Generation Mode):

Modo	WGM13	WGM12	WGM11	WGM10	Descripción	MAX
0	0	0	0	0	Normal	0xFFFF
1	0	0	0	1	PWM con Fase Correcta, 8 bits	0x00FF
2	0	0	1	0	PWM con Fase Correcta, 9 bits	0x01FF
3	0	0	1	1	PWM con Fase Correcta, 10 bits	0x03FF
4	0	1	0	0	CTC	OCR1A
5	0	1	0	1	PWM Rápido, 8 bits	0x00FF
6	0	1	1	0	PWM Rápido, 9 bits	0x01FF
7	0	1	1	1	PWM Rápido, 10 bits	0x03FF
8	1	0	0	0	PWM con Frecuencia y Fase Correcta	ICR1
9	1	0	0	1	PWM con Frecuencia y Fase Correcta	OCR1A
10	1	0	1	0	PWM con Fase Correcta	ICR1
11	1	0	1	1	PWM con Fase Correcta	OCR1A
12	1	1	0	0	CTC	ICR1
13	1	1	0	1	Reservado	-
14	1	1	1	0	PWM Rápido	ICR1
15	1	1	1	1	PWM Rápido	OCR1A

Modos de operación

- **Modo 0:** En este modo es un temporizador sin alguna funcionalidad extra y sólo genera eventos de desbordamiento.
- **Modos 1, 2, 3, 10 y 11:** Modos de PWM en fase correcta, el temporizador automáticamente se incrementa de cero a máximo y de máximo a cero, al coincidir con los registros de comparación conmutará una salida para la generación de una señal PWM.
- **Modo 4 y 12:** Son modos CTC (*clear timer on compare match*), en donde el temporizador se limpia automáticamente después de una coincidencia con el registro **OCR1A** e **ICR1**, según el modo.
- **Modos 5, 6, 7, 14 y 15:** Modos de PWM rápido, el temporizador automáticamente se incrementa de cero a máximo, al coincidir con los registros de comparación conmutará una salida para la generación de una señal PWM.
- **Modos 8 y 9:** Modos de PWM con frecuencia y fase correcta, son similares a los Modos de PWM en fase correcta, cambia el momento en que se actualiza el registro de comparación.

- Bits **COM1A[1:0]** y **COM1B[1:0]** (*Compare Output Mode*): Con estos bits se configura una respuesta automática en OC1x, ante eventos de comparación.

Para los modos que no son PWM se tiene:

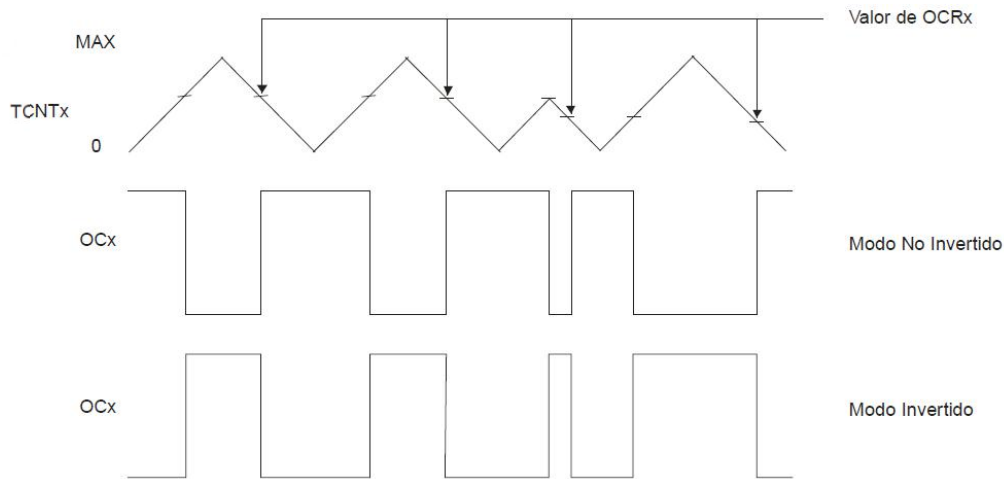
COM1x1	COM1x0	Descripción
0	0	Operación normal - OC1x desconectado
0	1	Conmuta OC1x, sobre una coincidencia por comparación.
1	0	Limpia OC1x, sobre una coincidencia por comparación.
1	1	Ajusta OC1x, sobre una coincidencia por comparación.

Para los modos PWM:

COM1x1	COM1x0	Descripción
0	0	Operación normal - OC1x desconectado, no hay salida PWM.
0	1	Reservado (sin uso)
1	0	Modo No Invertido
1	1	Modo Invertido

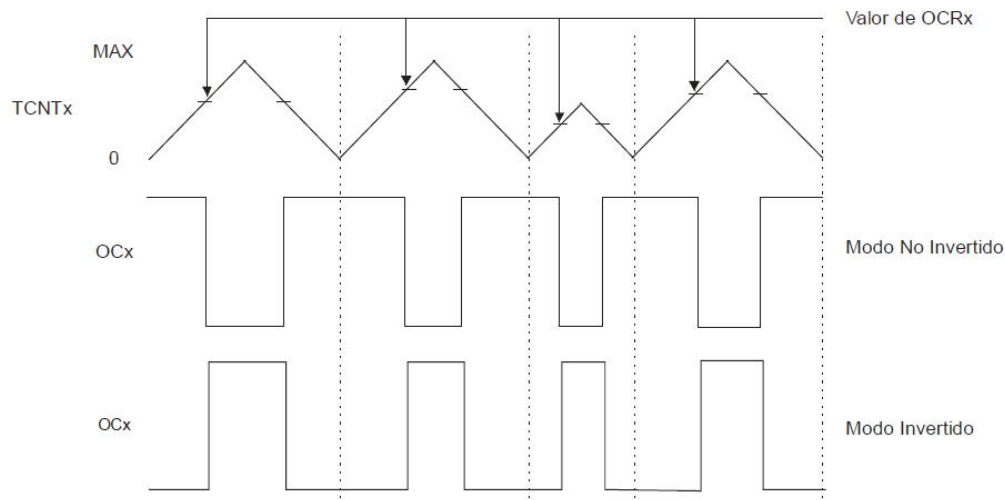
(x puede ser A o B)

PWM con Fase Correcta



OCR1x se actualiza cuando el temporizador llega al máximo valor.

PWM con Frecuencia y Fase Correcta



OCR1x se actualiza cuando el temporizador llega a cero.

- **Bits FOC1A y FOC1B:** forzan u obligan un evento de comparación, si se configuró una respuesta automática, ésta también va a realizarse. **FOC1A** se relaciona con el registro **OCR1A** y **FOC1B** con el registro **OCR1B**.
- **Bits CS1[2:0] (Clock Select):** Bits de selección de la fuente de reloj.

CS12	CS11	CS10	Descripción
0	0	0	Sin fuente de reloj (temporizador detenido)
0	0	1	CLK I/O (Sin pre-escalador)
0	1	0	CLK I/O/8 (Desde el pre-escalador)
0	1	1	CLK I/O/64 (Desde el pre-escalador)
1	0	0	CLK I/O/256 (Desde el pre-escalador)
1	0	1	CLK I/O/1024 (Desde el pre-escalador)
1	1	0	Fuente externa (en T1), con flanco de bajada
1	1	1	Fuente externa (en T1), con flanco de subida

Captura de Entrada

Un evento en la terminal **ICR1** hace que el temporizador se copie en **ICR1** y que la bandera **ICF1** sea puesta en alto.

Bits involucrados:

- **Bit ICNC1** (*Input Capture Noise Canceler*): Activa un eliminador de ruido. La entrada se filtra por cuatro muestras sucesivas.
- **Bit ICES1** (*Input Capture Edge Select*): Selecciona el flanco de activación de la captura.

0	Flanco de bajada
1	Flanco de subida

En los modos 8, 10, 12 y 14, el registro **ICR1** tiene otras funciones por lo que y la función de captura no puede utilizarse.

Eventos generados por el Temporizador/Contador 1

- Los eventos se ven reflejados en el registro **TIFR1** (*Timer/Counter Interrupt Flag Register*), se tienen cuatro eventos:

TIFR1, registro de estado del temporizador 1:

7	6	5	4	3	2	1	0
-	-	ICF1	-	-	OCF1B	OCF1A	TOV1

- TOV1** (*Timer/Counter Overflow 1*): Desbordamiento del Temporizador/Contador 1.
- OCF1x** (*Output Compare Match Flag 1x*): Indica una coincidencia en la comparación.
 - Se tienen dos banderas por que el temporizador es comparado con dos registros diferentes: **OCR1A** y **OCR1B**.
- ICF1** (*Input Capture Flag*): Captura de entrada con el Temporizador/Contador 1.

Nota: Las banderas se limpian automáticamente si se emplean interrupciones o escribiendo un 1 si se usa sondeo.

Habilitación de Interrupciones

Las interrupciones se activan en el registro **TIMSK1** (*Timer/Counter Interrupt Mask Register 1*). Los habilitadores individuales son:

7	6	5	4	3	2	1	0
-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1

- **TOIE1** (*Timer/Counter 1 Overflow Interrupt Enable*): Habilita la interrupción por desbordamiento del Temporizador/Contador 1.
- **OCIE1x** (*Timer/Counter 1 Output Compare Match x Interrupt Enable*): Habilita la interrupción por una coincidencia en la comparación.
 - Se tienen dos habilitaciones porque el temporizador es comparado con dos registros diferentes: **OCR1A** y **OCR1B**.
- **ICIE1** (*Input Capture Interrupt Enable*): Habilita las interrupciones por eventos de captura



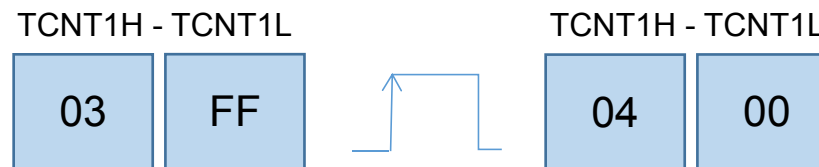
Vectores de interrupción

- Los **vectores de interrupción** para los eventos del temporizador/ contador 1 son:

Dirección	Evento	Descripción
0x0014	TIMER1_CAPT	Captura de entrada con el Timer 1
0x0016	TIMER1_COMPA	El Timer 1 coincide con el comparador A
0x0018	TIMER1_COMPB	El Timer 1 coincide con el comparador B
0x001A	TIMER1_OVF	Desbordamiento del Timer 1

Acceso a registros de 16 bits

- Puede ocurrir un problema al leer el registro **TCNT1** con accesos por bytes (principalmente en ensamblador), porque su valor cambia en cada ciclo de reloj.
- Por ejemplo, si el temporizador no se detiene y se va a leer cuando tiene 0x03FF, ¿qué pasa si el acceso es con una interfaz de 8 bits?



- Si se lee primero el byte menos significativo....
- Si se lee primero el byte más significativo....



Acceso a registros de 16 bits

- El hardware incluye un registro temporal de 8 bits para apoyar las lecturas y escrituras de 16 bits.
- Cuando se escribe o lee la parte alta, se hace la escritura o lectura en el registro temporal.
- Cuando se tiene acceso a la parte baja, se hacen lecturas o escrituras de 16 bits tomando como parte alta al registro temporal.
- Debe haber un orden en el acceso a los registros de 16 bits.
- Lectura....
- Escritura.....



Acceso a registros de 16 bits

Ejemplo 1: Leer el registro TCNT1 y dejar el resultado en R17:R16.

```
IN      R16, TCNT1L ; lee 16 bits, el byte alto queda en el
                        ; registro temporal
IN      R17, TCNT1H ; lee del registro temporal
```

Ejemplo 2: Escribe 1500 en TCNT1.

```
LDI     R16, HIGH (1500) ; escribe el registro temporal
OUT     TCNT1AH, R16
LDI     R16, LOW  (1500) ; escribe 16 bits
OUT     TCNT1AL, R16
```

Nota: En alto nivel no hay conflicto porque se pueden emplear variables o registros de 16 bits.



Ejercicio 1:

- Configure al ATmega328P para que genere un tono de 500 Hz (aproximadamente), mientras se mantenga presionado un botón.

$$F = 500 \text{ Hz} \Rightarrow T = 2 \text{ ms} = 2000 \mu\text{S}$$

$$TH = TL = 1250 \mu\text{S}$$

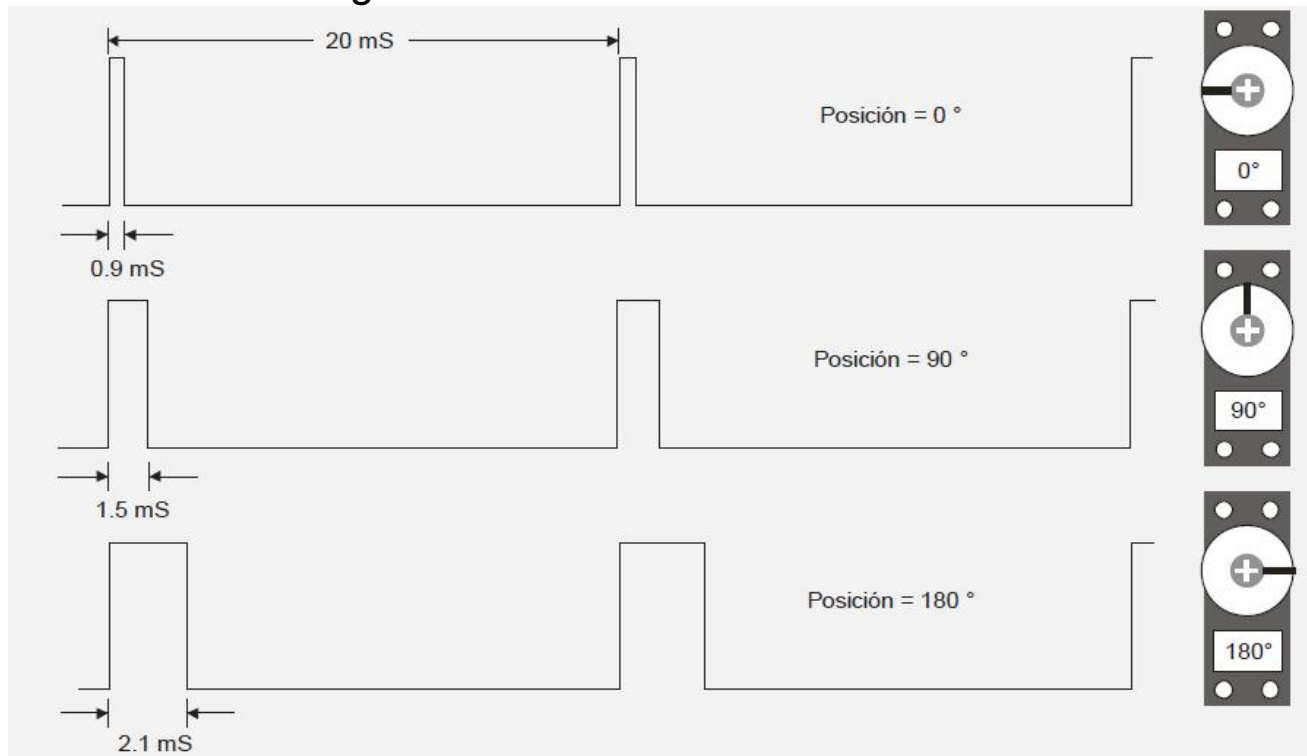
Recursos a emplear:

Botón – Cambio en pin en PC0.

Tono – Respuesta automática en OC1A (PB1).

Ejercicio 2:

- Un servomotor se maneja con una señal PWM a una frecuencia de 50 Hz, su comportamiento es el siguiente:

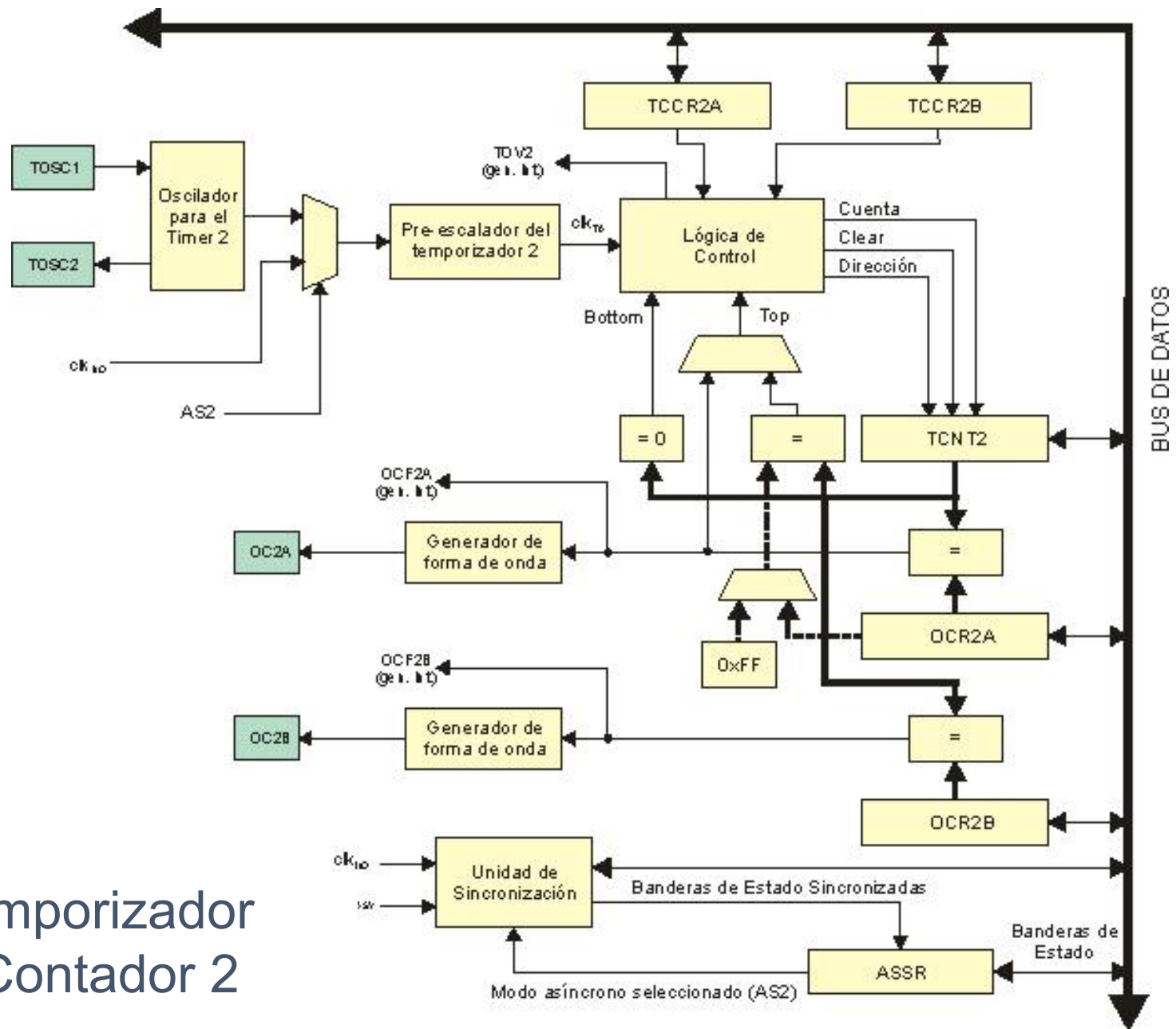


Realice un programa que mueva el servomotor de 0° a 90°, de 90° a 180°, de 180° a 90° y de 90° a 0°, con un periodo de 200 mS entre cada cambio. Repita la secuencia de movimiento de manera continua.



Organización y Registros del Temporizador/Contador 2

- **TCNT2** (*Timer/Counter 2*), es el registro de 8 bits del temporizador/ contador 2.
- **OCR2A** (*Output Compare Register A*), es uno de los registros con el que continuamente se compara el temporizador para generar eventos de coincidencias por comparación.
- **OCR2B** (*Output Compare Register B*), es el otro registro con el que continuamente se compara el temporizador para generar eventos de coincidencias por comparación.
- Los registros **OCR2A** y **OCR2B** también son usados en los modos PWM.



Temporizador / Contador 2

Registros de control

- Se tienen dos registros para el control del temporizador/contador 2: **TCCR2A** y **TCCR2B** (*Timer/Counter Control Register*):

TCCR2A:

7	6	5	4	3	2	1	0
COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20

TCCR2B:

7	6	5	4	3	2	1	0
FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20

Modos de operación

- Bits **WGM2[2:0]** (Waveform Generation Mode): Para definir el modo de operación, se tienen las siguientes combinaciones:

Modo	WGM22	WGM21	WGM20	Descripción
0	0	0	0	Normal
1	0	0	1	PWM con fase correcta (MAX = 0xFF)
2	0	1	0	CTC (<i>clear timer on compare match</i>) – El máximo está en OCR2A
3	0	1	1	PWM rápido (MAX = 0xFF)
4	1	0	0	Reservado
5	1	0	1	PWM con fase correcta (MAX = OCR2A)
6	1	1	0	Reservado
7	1	1	1	PWM rápido (MAX = OCR2A)

- Bits **COM2A[1:0]** y **COM2B[1:0]** (*Compare Output Mode*): Con estos bits se configura una respuesta automática en OC2x, ante eventos de comparación.

Para los modos que no son PWM se tiene:

COM2x1	COM2x0	Descripción
0	0	Operación normal - OC2x desconectado
0	1	Conmuta OC2x, sobre una coincidencia por comparación.
1	0	Limpia OC2x, sobre una coincidencia por comparación.
1	1	Ajusta OC2x, sobre una coincidencia por comparación.

Para los modos PWM:

COM2x1	COM2x0	Descripción
0	0	Operación normal - OC2x desconectado, no hay salida PWM.
0	1	Reservado (sin uso)
1	0	Modo No Invertido
1	1	Modo Invertido

(x puede ser A o B)

- **Bits FOC2A y FOC2B:** forzan u obligan un evento de comparación, si se configuró una respuesta automática, ésta también va a realizarse. **FOC2A** se relaciona con el registro **OCR2A** y **FOC2B** con el registro **OCR2B**.
- **Bits CS2[2:0] (Clock Select):** Bits de selección de la fuente de reloj.

CS22	CS21	CS20	Descripción
0	0	0	Sin fuente de reloj (temporizador detenido)
0	0	1	CLK _{T2S} (Sin pre-escalador)
0	1	0	CLK _{T2S} /8
0	1	1	CLK _{T2S} /32
1	0	0	CLK _{T2S} /64
1	0	1	CLK _{T2S} /128
1	1	0	CLK _{T2S} /256
1	1	1	CLK _{T2S} /1024

Eventos del Temporizador/Contador 2

- Los eventos se ven reflejados en el registro **TIFR2** (*Timer/Counter Interrupt Flag Register*), se tienen tres eventos:

7	6	5	4	3	2	1	0
-	-	-	-	-	OCF2B	OCF2A	TOV2

- TOV2** (*Timer/Counter Overflow 2*): Desbordamiento del Temporizador/Contador 2.
- OCF2x** (*Output Compare Match Flag 2x*): Indica una coincidencia en la comparación.
 - Se tienen dos banderas por que el temporizador es comparado con dos registros diferentes: **OCR2A** y **OCR2B**.

Nota: Las banderas se limpian automáticamente si se emplean interrupciones o escribiendo un 1 si se usa sondeo.

Habilitación de Interrupciones

Las interrupciones se activan en el registro **TIMSK2** (*Timer/Counter Interrupt Mask Register 2*). Los habilitadores individuales son:

7	6	5	4	3	2	1	0
-	-	-	-	-	OCIE2B	OCIE2A	TOIE2

- **TOIE2** (*Timer/Counter 2 Overflow Interrupt Enable*): Habilita la interrupción por desbordamiento del Temporizador/Contador 2.
- **OCIE2x** (*Timer/Counter 2 Output Compare Match x Interrupt Enable*): Habilita la interrupción por una coincidencia en la comparación.
 - Se tienen dos habilitaciones porque el temporizador es comparado con dos registros diferentes: **OCR2A** y **OCR2B**.

Vectores de interrupción

- Los **vectores de interrupción** para los eventos del temporizador/ contador 2 son:

Dirección	Evento	Descripción
0x000E	TIMER2_COMPA	El Timer 2 coincide con el comparador A
0x0010	TIMER2_COMPB	El Timer 2 coincide con el comparador B
0x0012	TIMER2_OVF	Desbordamiento del Timer 2

Reloj de Tiempo Real

El temporizador 2 puede operar con un cristal externo de 32.768 kHz. Con el pre-escalador se consiguen periodos de desbordamiento que son fracciones o múltiplos de segundos reales.

Factor de pre-escala	Frecuencia del temporizador 2	Periodo de desbordamiento
1	32. 768 kHz	1/128 s
8	4. 096 kHz	1/16 s
32	1. 024 kHz	1/4 s
64	512 Hz	1/2 s
128	256 Hz	1 s
256	128 Hz	2 s
1024	32 Hz	8 s

- El desbordamiento se realiza cada 256 ciclos.

Registro ASSR (*Asynchronous Status Register*)

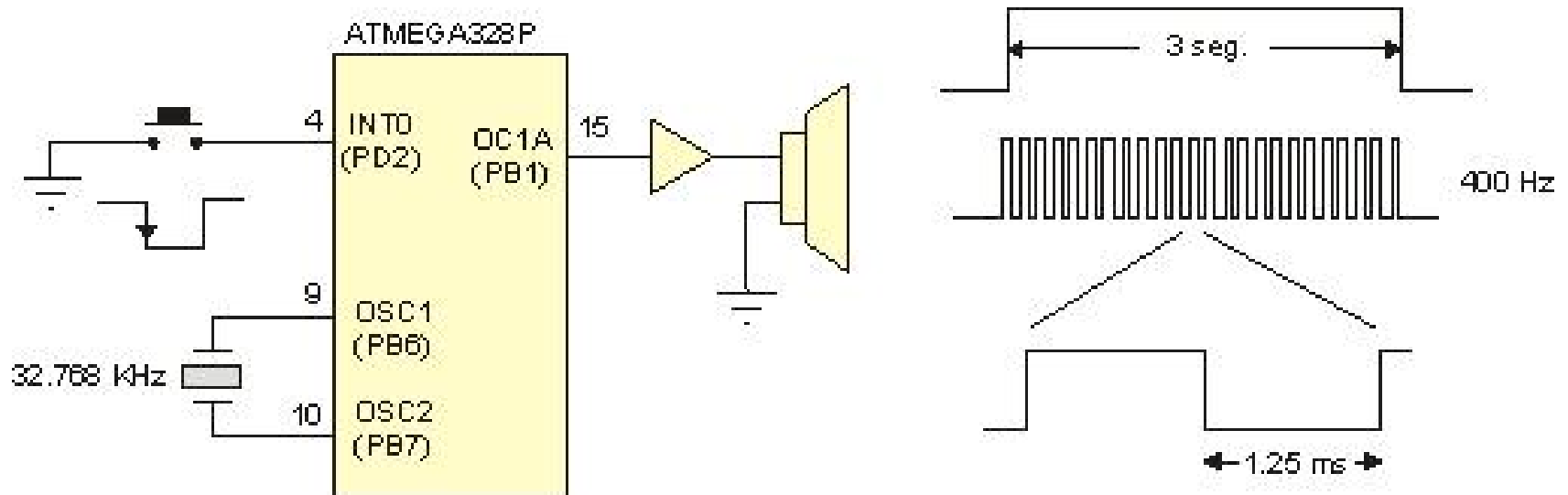
Registro para controlar la operación asíncrona del temporizador 2.

7	6	5	4	3	2	1	0
-	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB

- **Bit EXCLK:** Determina si la operación asíncrona será con un cristal de 32.768 KHz o a una frecuencia diferente.
- **Bit AS2** (*Asynchronous Timer/Counter*) Habilita la operación asíncrona. Puede ser un cristal de 32.768 kHz (EXCLK = 0) o una señal de reloj diferente (EXCLK = 1).
- **Bits TCN2UB, OCR2AUB, OCR2BUB, TCR2AUB, TCR2BUB** (UB: *Update Busy*) - significa que el registro correspondiente está **ocupado por actualización**.
- Estas banderas se ajustan o limpian de manera automática, útiles para sincronizar al temporizador 2 con el resto del sistema.

Ejercicio 1:

- Configure al ATmega328P para que funcione como un timbre, de manera que genere un tono de 400 Hz durante 3 segundos cuando se presione un botón. En la figura se muestra el acondicionamiento del hardware con la salida esperada.





Ejercicio 2:

- En un supermercado se ha determinado premiar a cada cliente múltiplo de 200.
- Desarrolle un sistema basado en un AVR, el cual debe detectar al cliente número 200 y generar un tono de 440 Hz (aproximadamente), por 5 segundos, cuando eso ocurra.
- Los clientes deben presionar un botón para ser considerados.
- Se trata de una aplicación para los temporizadores, utilice al temporizador 0, manejado con eventos externos, para llevar el conteo de clientes.
- Al temporizador 1 para generar el tono de 440 Hz y al temporizador 2 con un oscilador externo de 32.768 KHz para el conteo de segundos, .



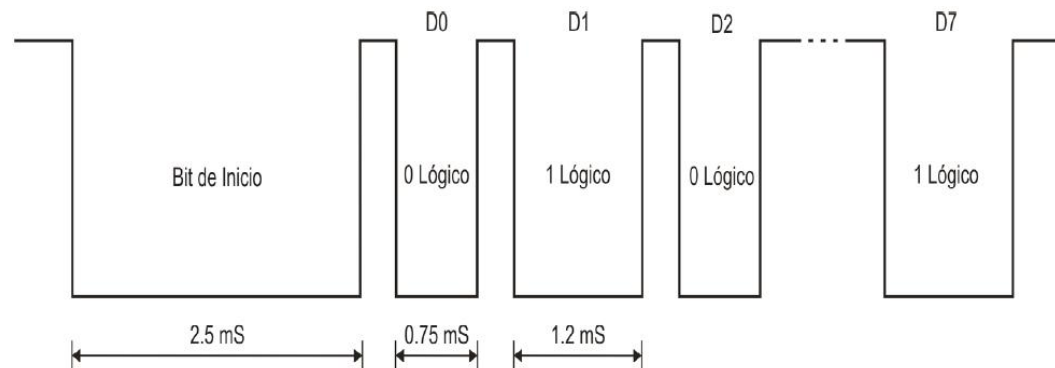
Ejercicio 3:

- Empleando PWM, controle el encendido de un conjunto de LEDs ultra-brillantes, manejando 5 niveles de intensidad. El sistema deberá contar con un botón para el cambio de intensidad, la cual será modificada con incrementos de un 20 % cada vez que el botón es presionado (al encender el sistema los LEDs estarán apagados).

Para una adecuada operación de los LEDs, configure para que la señal de salida tenga una frecuencia de 100 Hz.

Ejercicio 4:

- Empleando los recursos de captura del temporizador 1, desarrolle un programa que reciba y decodifique una secuencia serial de 8 bits de información modulada por el ancho de pulsos activos en bajo, como se muestra en la figura:



- Después de detectar al bit de inicio deben obtenerse los 8 bits de datos, iniciando con el bit menos significativo, concluida la recepción, debe mostrarse el dato en cualquiera de los puertos libres.
- Sugerencia:* utilice 2 mS y 1 mS como referencias, si el ancho del pulso es mayor a 2 mS es un bit de inicio, menor a 2 mS pero mayor a 1 mS es un 1 lógico, menor a 1 mS, se trata de un 0 lógico.