

Tema 6 – Periféricos de usuario

Profesor: Carlos Diego Moreno Moreno

Dpto. de Ingeniería Electrónica y de Computadores.
Área de Arquitectura y Tecnología de Computadores.
Escuela Politécnica Superior. Universidad de Córdoba.

Objetivos.

- Dar una idea general de los periféricos más sencillos que incorpora el LPC2378.
- Conocer las características básicas de los cinco puertos de 32 bits de este micro.
- Estudiar las características en los tres modos de funcionamiento de los cuatro *timers* de que dispone.
- Dar a conocer las características básicas del *WatchDog Timer* (WDT).
- Estudio del PWM y del interfaz que tiene el LPC2378.
- Conocimiento básico del reloj–calendario (RTC) incluido.
- Funcionamiento desde el punto de vista del programador de los conversores analógico/digital y digital/analógico.

TEMA 6: Periféricos de usuario.

6.1.– Puertos de entrada/salida digitales.

6.2.– Temporizadores.

6.3.– Temporizador de vigilancia (*WatchDog Timer*, WDT).

6.4.– Modulación por anchura de pulsos (PWM).

 6.4.1.– Definición.

 6.4.2.– Tipos.

 6.4.3.– Aplicaciones.

 6.4.4.– Modulador PWM del LPC23xx.

6.5.– Reloj en tiempo real (RTC).

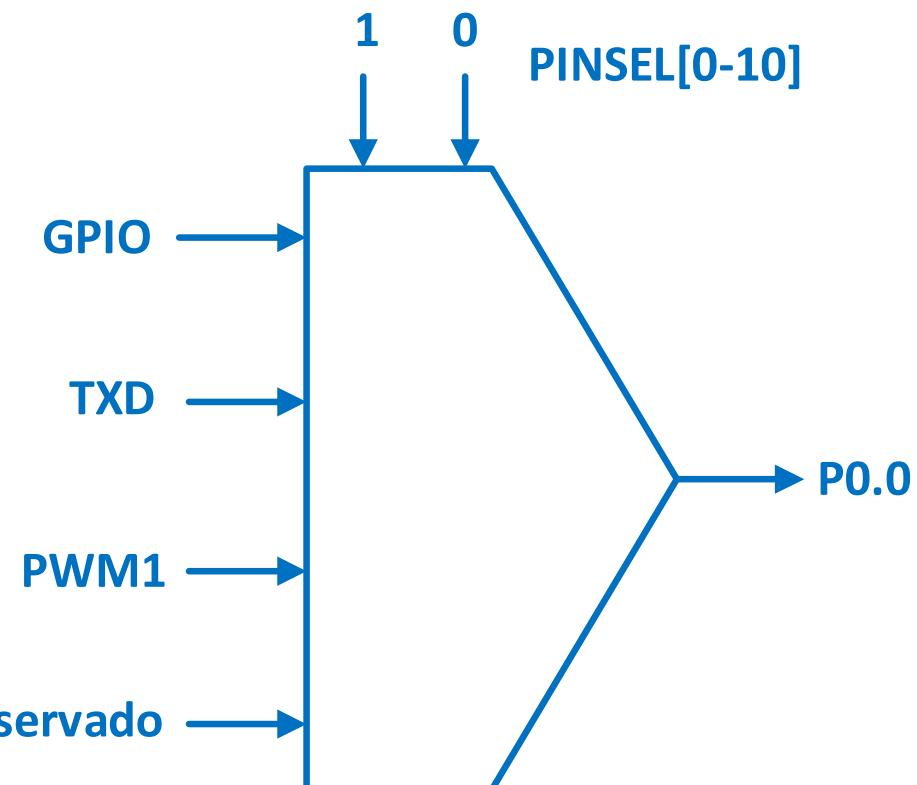
6.6.– Conversor Analógico/Digital.

6.7.– Conversor Digital/Analógico.

6.1.– Puertos de entrada/salida digitales (GPIO)

- El número de periféricos integrados en el LPC23XX es muy elevado, por lo que conectar todos los periféricos a pines externos del circuito sería prácticamente imposible, además de encarecer la complejidad y el coste del encapsulado. Por este motivo, los pines del circuito pueden conectarse a más de un periférico teniendo por tanto más de una función.
- Los pines externos se configuran con dos registros: PINSEL, que multiplexa entre una de las cuatro posibles funciones al pin, y PINMODE que selecciona el tipo de salida.

BLOQUE DE CONEXIÓN DEL PIN



6.1.– Puertos de entrada/salida digitales (GPIO)

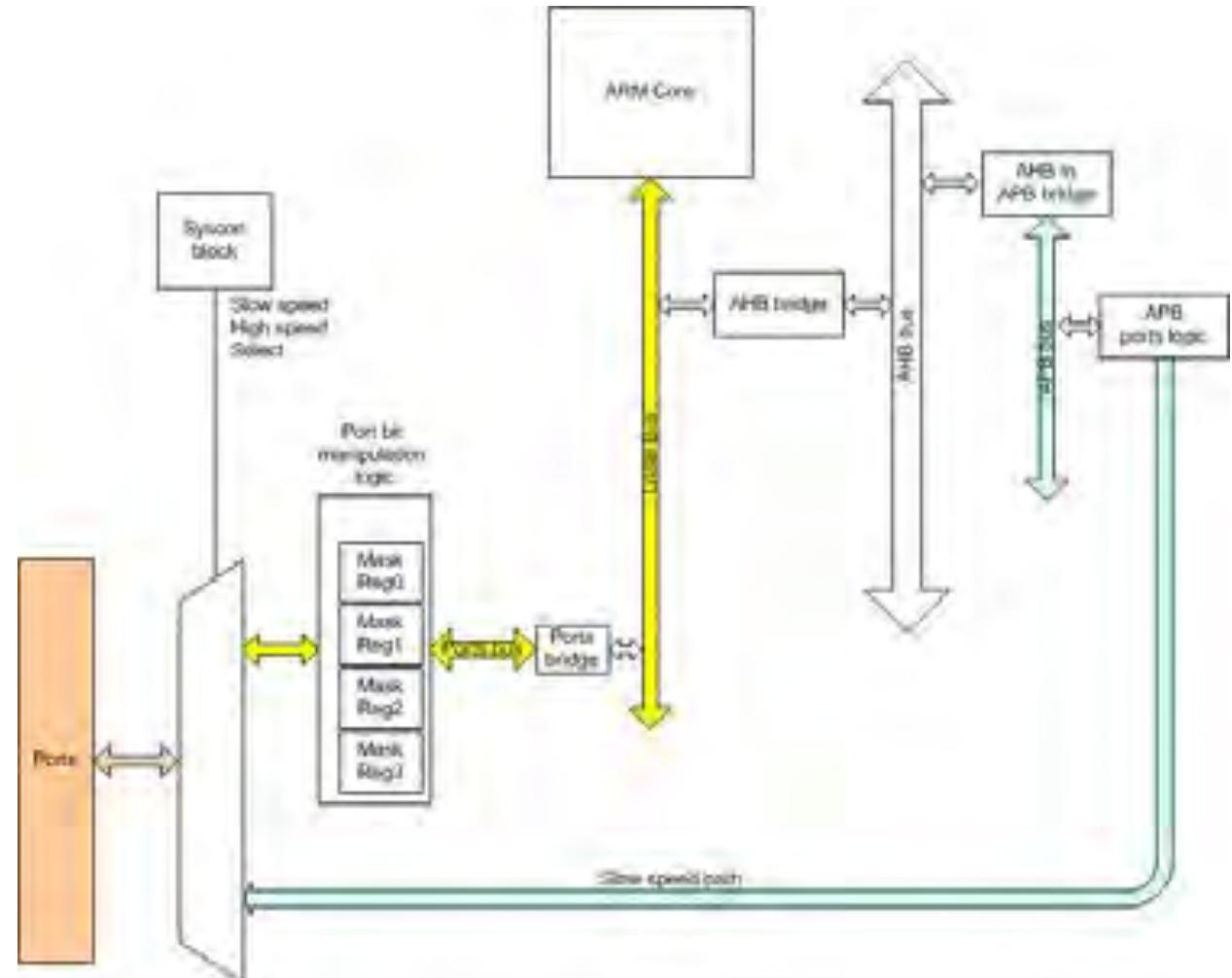
- ⦿ En los anteriores dispositivos LPC21XX, los registros de control GPIO estaban localizados en el bus APB (*Advanced Peripheral Bus*), junto con los de los demás periféricos.
- ⦿ Sin embargo, direccionar estos registros consume un gran número de ciclos para pasar al bus AHB (*Advanced High performance Bus*) a través del APB.
- ⦿ En los siguientes dispositivos LPC213x, NXP introdujo un nuevo conjunto de registros rápidos GPIO situados en el bus local del ARM7.
- ⦿ Estos registros permiten activar un pin del puerto en sólo dos ciclos a una velocidad de 30 MHz, lo que constituye un gran avance.
- ⦿ Además, los registros rápidos GPIO introducen un registro de máscara que mejora la manipulación a nivel de *bit* de cada puerto.

6.1.– Puertos de entrada/salida digitales (GPIO)

- ❖ Los dispositivos LPC23xx disponen de cinco puertos de E/S de propósito general (GPIO) de 32 *bits* cada uno, dando un máximo de 160 pines: PORT0, PORT1, PORT2, PORT3 y PORT4.
- ❖ Para mantener la compatibilidad con los dispositivos anteriores LPC21XX, los puertos PORT0 y PORT1 tienen un conjunto de registros de control heredados en el bus APB (*Advanced Peripheral Bus*). Sin embargo, el control de estos dos puertos por estos registros es bastante lento.
- ❖ La familia de microcontroladores LPC23XX dispone de un segundo conjunto de registros de control de los GPIO situados en el bus local llamados registros de control rápidos GPIO. A no ser que deseemos compatibilizar código existente en versiones anteriores, debemos de usar los registros rápidos en lugar de los registros heredados PORT0 y PORT1.
- ❖ Por otro lado, los puertos PORT0 y PORT2, pueden generar una interrupción cuando detecten un flanco ascendente o descendente en un pin determinado.

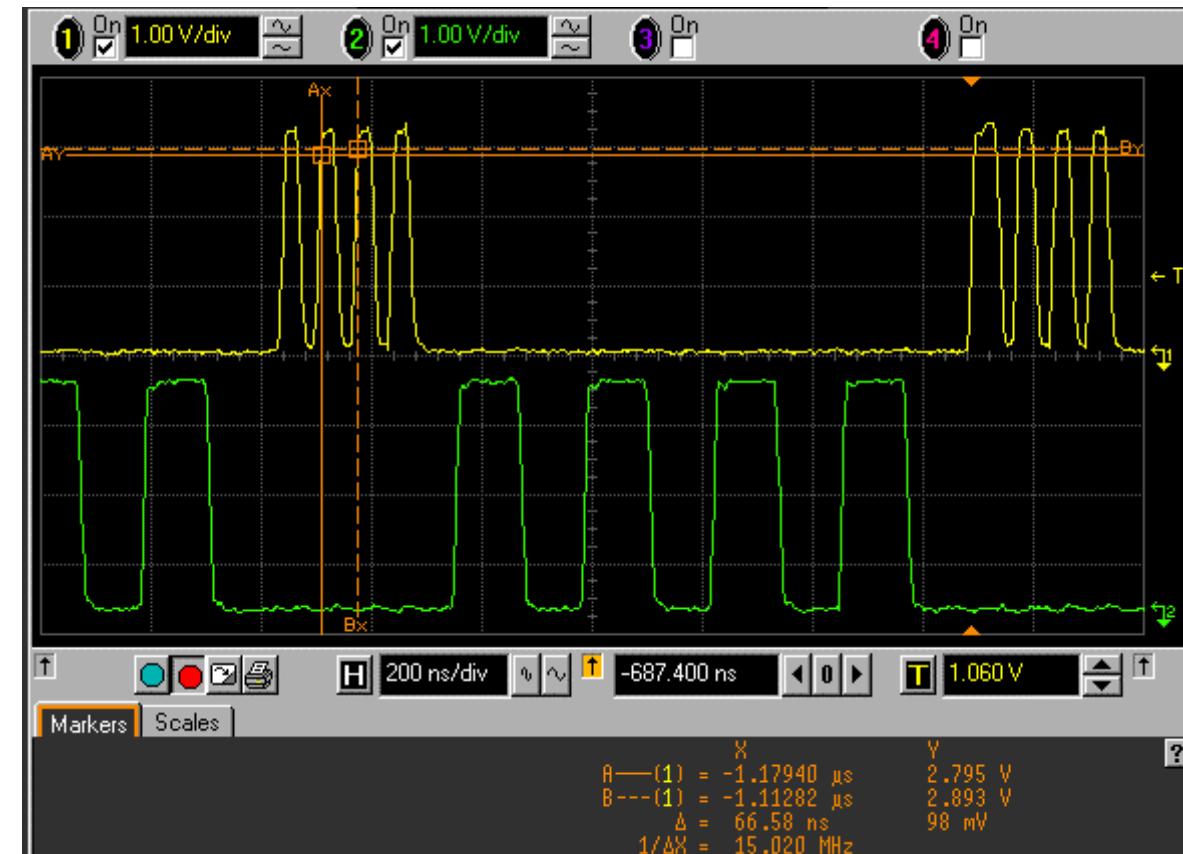
6.1.- Puertos de entrada/salida digitales (GPIO)

- PO y P1 son accesibles a través del conjunto de registros heredados (*legacy port*) o a través del grupo de registros que le permiten una mejora de las características y una mejora de la velocidad.
- P2, P3, P4 son solo accesibles a través del grupo de registros que le permiten una mejora de las características y una mejora de la velocidad.



6.1.– Puertos de entrada/salida digitales (GPIO)

- Puertos rápidos (P2, P3 y P4, color amarillo, señal superior). Conectados al bus local (conmutan hasta 30 MHz).
- Puertos normales (P0 y P1, color verde, señal inferior). Conectados al bus APB (conmutan hasta 4,3 MHz).
- Obsérvese la velocidad de conmutación de cada tipo de puerto.



6.1.– Puertos de entrada/salida digitales (GPIO)

- Tras un *reset* el “*pin connect block*” configura todos los pines del circuito como puertos de entrada salida (GPIO).
- Los registros de configuración son los siguientes:

Registro	Función
FIOxDIR ($x=[0,4]$)	Selecciona en cada <i>bit</i> si un pin es entrada (0) o salida (1)
FIOxSET ($x=[0,4]$)	Escribe uno en los <i>bits</i> seleccionados de un registro IO
FIOxCLR ($x=[0,4]$)	Escribe cero en los <i>bits</i> seleccionados de un registro IO
FIOxPIN ($x=[0,4]$)	Permite escribir o leer en un registro IO
FIOxMASK ($x=[0,4]$)	Cuando se escriben unos en los <i>bits</i> se inhibe cualquier acción de escritura en los registros de entrada/salida

6.1.– Puertos de entrada/salida digitales (GPIO)

- PINSELx ($x = [0,10]$): estos registros escogen una de entre cuatro funciones que puede tener cada pin del microcontrolador.
- PINMODEx ($x = [0,10]$): configura el tipo de salida, permiten habilitar o deshabilitar resistencias de *pull-up* y *pull-down* en cada pin.

Bits de selección	Función
00	<i>Pull-up</i> habilitado (por defecto)
01	Flotante (sin <i>pull-up</i> ni <i>pull-down</i>)
10	<i>Pull-down</i> habilitado
11	Reservado

6.1.– Puertos de entrada/salida digitales (GPIO)

💡 Interrupciones en los puertos de entrada/salida P0 y P2

- 💡 Los puertos de entrada/salida P0 y P2 pueden generar interrupciones al producirse una transición en ellos.
- 💡 Cada pin de los puertos P0 y P2 puede generar una interrupción en el flanco de subida, en el flanco de bajada o en ambos flancos.
- 💡 Para los dos puertos se tiene una única línea de interrupción (EINT3) que se conecta al VIC. Se debe chequear posteriormente los registros de estado de estos puertos para conocer qué pin y qué cambio en él, han provocado la interrupción.

TEMA 6: Periféricos de usuario.

6.1.– Puertos de entrada/salida digitales.

6.2.– Temporizadores.

6.3.– Temporizador de vigilancia (*WatchDog Timer*, WDT).

6.4.– Modulación por anchura de pulsos (PWM).

 6.4.1.– Definición.

 6.4.2.– Tipos.

 6.4.3.– Aplicaciones.

 6.4.4.– Modulador PWM del LPC23xx.

6.5.– Reloj en tiempo real (RTC).

6.6.– Conversor Analógico/Digital.

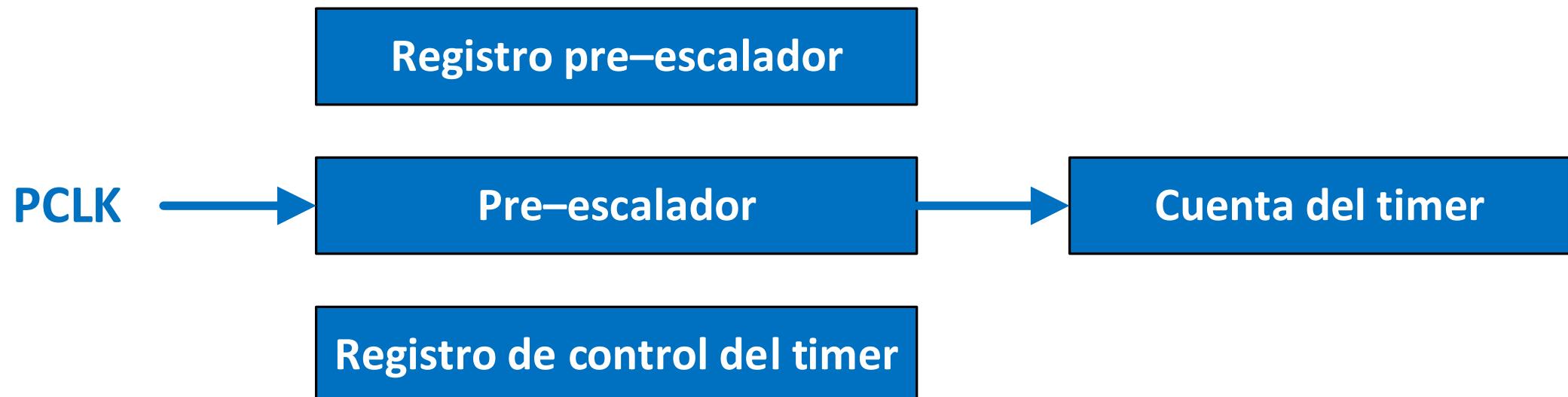
6.7.– Conversor Digital/Analógico.

6.2.– Temporizadores (*timers*)

- ❖ Los dispositivos LPC23xx disponen de cuatro temporizadores de propósito general idénticos en estructura y uso. Están basados en un contador–temporizador de 32 *bits* con un pre–escalador de 32 *bits*.
- ❖ La fuente de reloj por defecto para todos los temporizadores es el reloj periférico PCLK del bus APB (*Advanced Peripheral Bus*).
- ❖ El *tick rate* (tasa de actualizaciones por segundo) del temporizador está controlado por el valor almacenado en el registro pre–escalador (TxPR; x=0..3). El contador del pre–escalador se incrementa en cada pulso de PCLK hasta que alcanza el valor almacenado en el registro pre–escalador.
- ❖ Cuando alcanza el valor del pre–escalador, el contador del *timer* (TxTC; x=0..3) se incrementa en uno, la cuenta del pre–escalador se pone a cero y comienza una nueva cuenta.

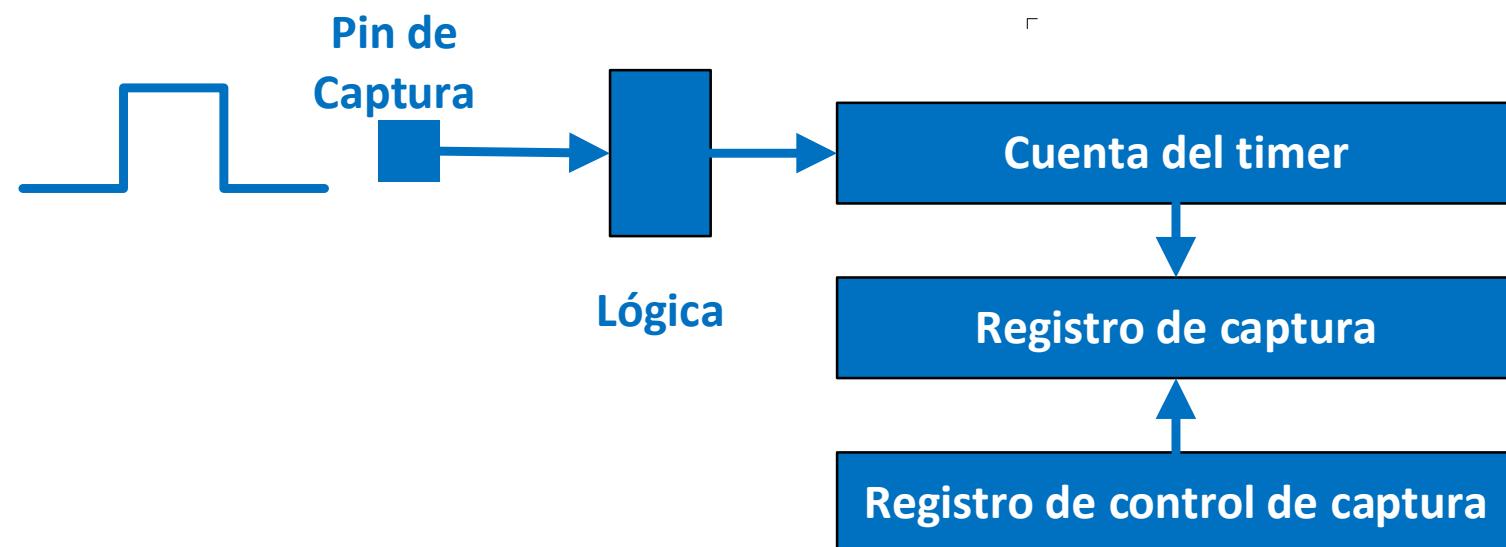
6.2.– Temporizadores (*timers*)

- El registro de control del temporizador (TxTCR; $x=0..3$) contiene sólo dos *bits* que se utilizan para habilitar/deshabilitar el temporizador o para restablecer su cuenta.
- Los cuatro temporizadores pueden trabajar en tres modos de funcionamiento: modo captura, modo contador y modo temporizador general (*match*).



6.2.– Temporizadores (*timers*)

- ❖ **Modo de captura:** Además del contador básico, cada temporizador dispone de cuatro canales de captura que permiten obtener el valor de la cuenta del *timer* cuando se produce una transición en una señal de entrada.
- ❖ Cada canal de captura tiene asociado un pin que puede disparar un evento de captura en un flanco ascendente o descendente. Cuando se produce un evento el valor de la cuenta del temporizador se almacena en un registro de captura asociado y si es necesario se generará una interrupción.

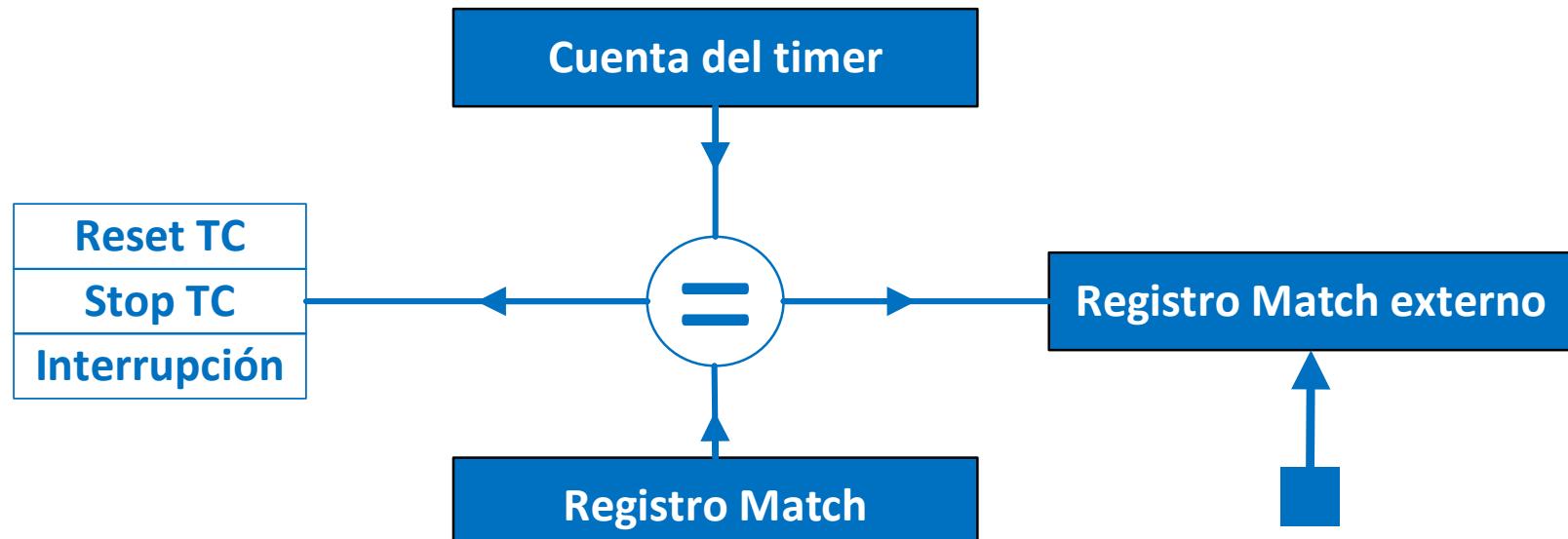


6.2.– Temporizadores (*timers*)

- ❖ **Modo contador:** este modo se utiliza cuando se quiere utilizar una señal externa como reloj del contador.
- ❖ El contador se incrementa en el flanco de subida, de bajada o en ambos.
- ❖ El registro de control de la cuenta (CTCR) permite seleccionar entre utilizar cada temporizador como un contador o un temporizador puro.
- ❖ Este registro permite cambiar la fuente de reloj de PCLK a una fuente de reloj externa aplicada en un pin de captura de temporizador seleccionado.
- ❖ El temporizador puede ser incrementado en un flanco ascendente, descendente o ambos de la señal de reloj externa.

6.2.– Temporizadores (*timers*)

- Modo temporizador general (*match*): Cada temporizador dispone también de cuatro canales tipo *match* cada uno de los cuales tiene un registro *match* de 32 bits. El valor actual de la cuenta del temporizador se compara con este registro *match* (de coincidencia) y cuando los valores coinciden se dispara un evento que puede ser actuar en el temporizador (*reset*, parar o generar una interrupción) o también actuar sobre un pin externo (puesta a uno, puesta a cero o cambio).



6.2.– Temporizadores (*timers*)

- ❖ **Modo temporizador general (*match*):** Para configurar el temporizador para un evento, hay que cargar el registro *match* con el valor deseado.
- ❖ El evento *match* interno se puede configurar a través del MCR (*Match Control Register*).
- ❖ En este registro cada canal tiene un grupo de *bits* que se puede utilizar para generar una interrupción del temporizador, reiniciar el temporizador, detener el temporizador, o cualquier combinación de estos tres eventos.
- ❖ Cada canal *match* tiene un pin *match* asociado que puede ser modificado cuando se produzca un evento.
- ❖ Al igual que con los pines de captura, primero se debe utilizar el bloque de conexión de pines para conectar el pin externo al canal *match*.

6.2.– Temporizadores (*timers*)

- Modo temporizador general (*match*): Los pines *match* son controlados por los cuatro *bits* menos significativos del **EMR (External Match Register)**.
- Este registro contiene un campo de *bits* para configurar cada canal *match* y decidir la acción a realizar en el pin *match* cuando ocurra un evento.

No usado					EMC3		EMC2		EMC1		EMC0		EM3	EM2	EM1	EM0	
15					12	11	10	9	8	7	6	5	4	3	2	1	0

Bits de control EMR (EMCx)	Función
00	Nada
01	Pone a 0 el pin
10	Pone a 1 el pin
11	Complementa el pin

TEMA 6: Periféricos de usuario.

6.1.– Puertos de entrada/salida digitales.

6.2.– Temporizadores.

6.3.– Temporizador de vigilancia (*WatchDog Timer, WDT*).

6.4.– Modulación por anchura de pulsos (PWM).

 6.4.1.– Definición.

 6.4.2.– Tipos.

 6.4.3.– Aplicaciones.

 6.4.4.– Modulador PWM del LPC23xx.

6.5.– Reloj en tiempo real (RTC).

6.6.– Conversor Analógico/Digital.

6.7.– Conversor Digital/Analógico.

6.3.- *WatchDog Timer (WDT)*

- ❖ El *WatchDog Timer (WDT)* es un temporizador de vigilancia de *hardware* que genera automáticamente un restablecimiento del sistema si el programa principal no lo repara periódicamente.
- ❖ Está diseñado para reiniciar el microcontrolador si el programa deja de ejecutarse correctamente. Su propósito principal es recuperar el sistema de fallos o bloqueos, garantizando que el *software* siga funcionando sin intervención manual.
- ❖ A menudo se utiliza para restablecer automáticamente un dispositivo integrado que se cuelga debido a un fallo de *software* o *hardware*.
- ❖ La mayoría de los microcontroladores actuales tienen *hardware* de temporizador de vigilancia.
- ❖ El programa principal normalmente tiene un bucle por el que pasa constantemente realizando varias funciones.

6.3.– *WatchDog Timer (WDT)*

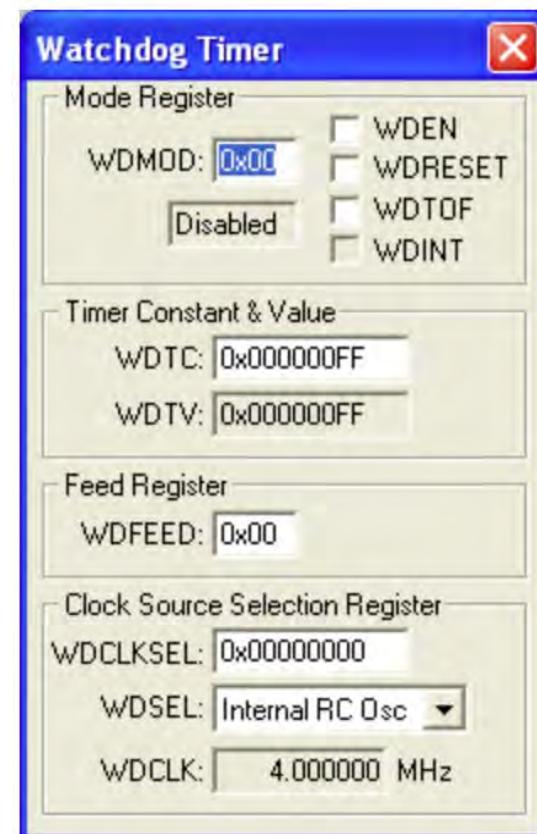
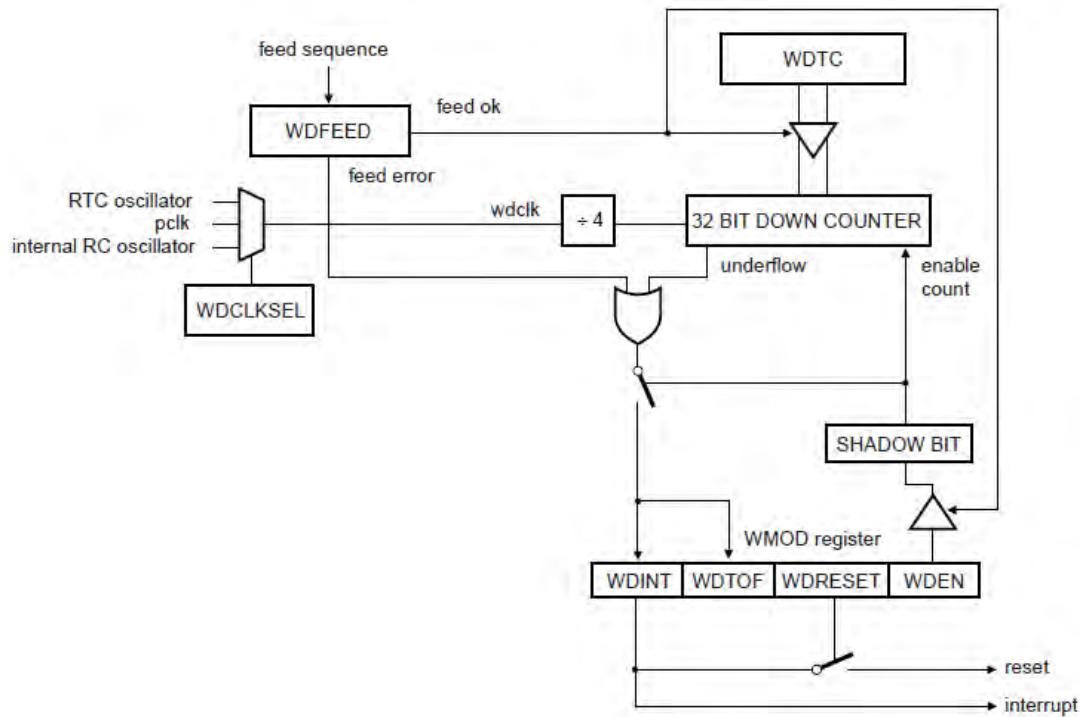
- ⦿ El WDT es un contador decreciente que se carga con un valor inicial mayor que el retardo de tiempo del peor caso a través del ciclo principal del programa.
- ⦿ Cada vez que pasa por el bucle principal, el código debe reiniciar el WDT.
- ⦿ Si ocurre un fallo y el programa principal no regresa para reiniciar el temporizador antes de que comience la cuenta atrás, este contador llega a cero y se genera una interrupción para reiniciar el procesador.
- ⦿ Usado de esta manera, el WDT puede detectar un fallo en un dispositivo integrado desatendido e intentar una acción correctiva con un reinicio.
- ⦿ Normalmente, después del reinicio, también se puede leer un registro para determinar si el WDT generó el reinicio o si fue un reinicio normal.

6.3.– *WhatchDog Timer (WDT)*

- ❖ Cuando el contador de *Watchdog* se desborde, el contador de programa comenzará desde 0x0000 0000 como en el caso del *reset* externo.
- ❖ El indicador de tiempo de espera de *Watchdog* (WDTOF) se puede examinar para determinar si el *Watchdog* ha causado la condición de reinicio.
- ❖ La bandera WDTOF debe ser borrado por *software*.
- ❖ El *WhatchDog Timer* utiliza dos relojes: PCLK y WDCLK.
 - PCLK se utiliza para que el APB acceda a los registros del *Watchdog*.
 - El WDCLK se utiliza para el conteo del *Watchdog Timer*.

6.3.- WhatchDog Timer (WDT)

- Diagrama de bloques y registros del WhatchDog Timer (WDT) del LPC2378.



6.3.– *WatchDog Timer (WDT)*

- ⦿ Registro de modo, WDMOD: contiene el modo del WDT (depurar u operar) y los *bits* de estado del WDT.
 - └ WDEN: configurado para habilitar la interrupción del WDT.
 - └ WDRESET: configurado para reiniciar el WDT.
 - └ WDTOF: se pone a uno cuando se agota el tiempo de espera del WDT.
 - └ WDINT: se pone a uno cuando se produce una interrupción del WDT.
- ⦿ Registro de constante de tiempo, WDTC: contiene el valor de tiempo de espera utilizado para recargar el WDT.
- ⦿ Registro de valor, WDTV: contiene el valor actual del WDT.
- ⦿ Registro de secuencia de alimentación del WDT, WDFEED: se utiliza para recargar el temporizador de vigilancia. Escribir 0xAA seguido de 0x55 en este registro recarga el temporizador de vigilancia a su valor preestablecido.
- ⦿ Registro de selección de la fuente de reloj, WDCLKSEL: selecciona la fuente de reloj entre oscilador interno RC (por defecto), el oscilador RTC o el reloj del bus APB (PCLK).

6.3.– *WatchDog Timer (WDT)*

En resumen, el WDT:

- ✓ El WDT previene bloqueos del sistema forzando un reinicio si el *software* deja de responder.
- ✓ Debe reiniciarse (*feed*) periódicamente para evitar el reinicio.
- ✓ Es configurable, permitiendo elegir el tiempo de cuenta y el tipo de acción (interrupción o reinicio).
- ✓ Fundamental en sistemas empotrados críticos donde la estabilidad es clave.

6.3.- WatchDog Timer (WDT)

- Ejemplo de Configuración en C (Keil µVision 5): el código siguiente configura el WDT con un tiempo de 2 segundos y lo activa:

```
#include <LPC23xx.h>

void WDT_Init(void) {
    WDCLKSEL = 0x1;           // Fuente de reloj: PCLK
    WDTC = 0x1FFFF;           // Configura tiempo de cuenta (ajustable)
    WDMOD = 0x3;              // Habilita WDT y reinicio del sistema
    WDFEED = 0xAA;             // Secuencia de alimentación
    WDFEED = 0x55;
}

void WDT_Feed(void) {
    WDFEED = 0xAA;            // Secuencia obligatoria para resetear el contador
    WDFEED = 0x55;
}

int main(void) {
    WDT_Init();

    while (1) {
        WDT_Feed();           // El microcontrolador evitará el reinicio si se ejecuta este comando
    }
}
```

6.3.– *WatchDog Timer (WDT)*

💡 Explicación de los registros de la función *WDT_Init*:

- └ **WDCLKSEL = 0x1;** selecciona PCLK como fuente de reloj del WDT.
- └ **WDTC = 0x1FFFFF;** Establece el valor de cuenta regresiva del WDT. Un valor alto significa un mayor tiempo antes del reinicio. Se puede ajustar según la aplicación.
- └ **WDMOD = 0x3;** Configura el modo de operación del WDT.
 - 0x1 → Habilita el WDT.
 - 0x2 → Activa el reinicio del sistema cuando el contador llega a 0.
 - 0x3 → Activa ambas opciones (WDT habilitado y reinicio del sistema).
- └ **WDFEED = 0xAA;**
- └ **WDFEED = 0x55;** Esta secuencia especial (*feed*) evita el reinicio inmediato. Es obligatoria y debe repetirse en la ejecución normal del programa.

6.3.– *WatchDog Timer (WDT)*

- ❖ La función *WDT_Feed* reinicia el contador del WDT para evitar el reinicio del sistema. Si no se llama a esta función dentro del tiempo límite, el microcontrolador se reiniciará automáticamente.
- ❖ En el programa principal, se llama a *WDT_Init()* para inicializar el *Watchdog*.
- ❖ En el bucle infinito, la función *WDT_Feed()* mantiene el WDT reiniciado. Si el programa se queda bloqueado y no ejecuta *WDT_Feed()*, el WDT reiniciará el sistema.

TEMA 6: Periféricos de usuario.

6.1.– Puertos de entrada/salida digitales.

6.2.– Temporizadores.

6.3.– Temporizador de vigilancia (*WatchDog Timer*, WDT).

6.4.– Modulación por anchura de pulsos (PWM).

6.4.1.– Definición.

6.4.2.– Tipos.

6.4.3.– Aplicaciones.

6.4.4.– Modulador PWM del LPC23xx.

6.5.– Reloj en tiempo real (RTC).

6.6.– Conversor Analógico/Digital.

6.7.– Conversor Digital/Analógico.

6.4.1.– Modulación por anchura de pulsos (PWM). Definición.

- La modulación de anchura de pulso PWM (*Pulse Width Modulation*) es una codificación analógica, alternativa a la representación habitual en amplitud de tensión, que se acomoda muy fácilmente a las técnicas digitales y permite configurar módulos de control de potencia, conversión DC–DC, potenciómetros digitales, conversión digital–analógica y analógico–digital, etc.
- Este tipo de etapas resulta de gran interés en el diseño microelectrónico puesto que permiten reducir, en forma considerable, la parte analógica que acompaña a la digital.
- El objeto propio de la electrónica es la información; las señales eléctricas son el soporte físico de la información. Generalmente dicha información se representa sobre la amplitud de la tensión eléctrica que conforma la señal: las variaciones de tensión corresponden, en su caso, a las variaciones del valor de la magnitud física representada por la señal.

6.4.1.– Modulación por anchura de pulsos (PWM). Definición.

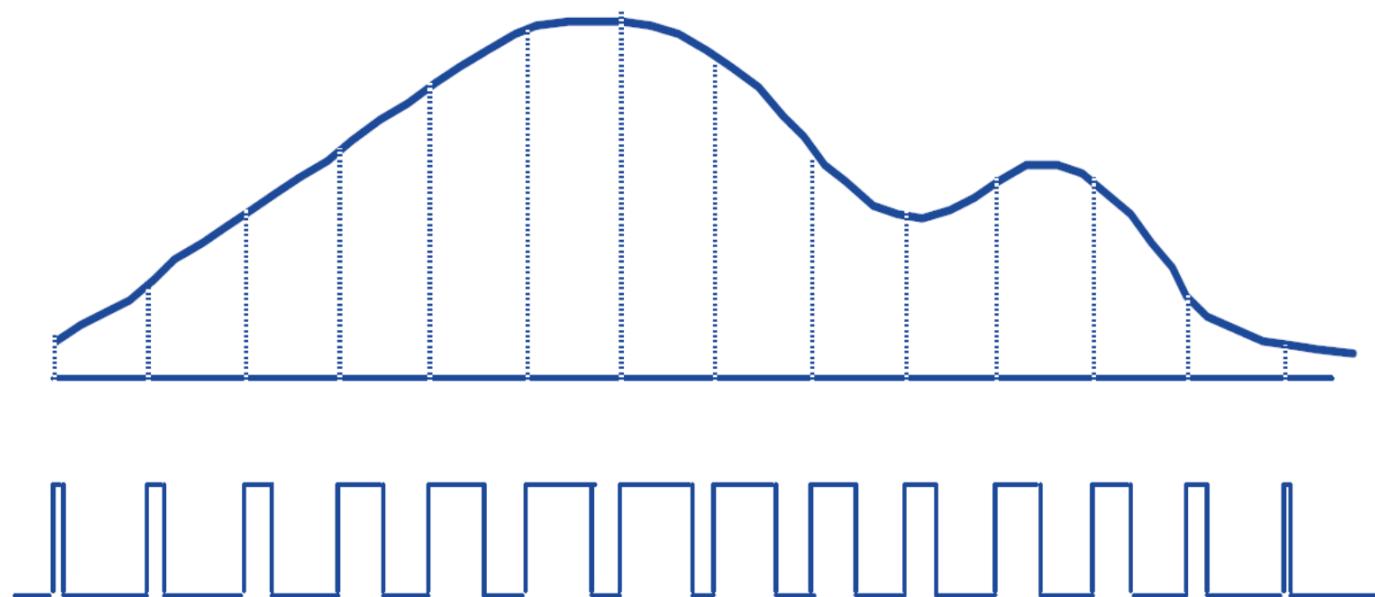
- Pero también es posible representar la misma información sobre la anchura de pulsos de tensión fija: la señal soporte de la información quedará determinada por las variaciones en la duración de los sucesivos pulsos; este procedimiento de codificación de la información se denomina modulación de anchura de pulsos (PWM).
- Ambas codificaciones analógicas, en amplitud de tensión o en anchura de pulso, son equivalentes: el teorema de muestreo de *Shannon* garantiza que ambas señales contienen la misma información, siempre que la frecuencia de los pulsos de anchura modulada sea superior al doble de la frecuencia máxima de la señal.
- Ahora bien, resulta que los pulsos de anchura modulada son relativamente fáciles de manejar mediante técnicas digitales, aprovechando que los contadores son una herramienta adecuada para manejar el tiempo.

6.4.1.– Modulación por anchura de pulsos (PWM). Definición.

- ❖ Contadores y sumadores permiten generar pulsos de anchura controlada por un número y, por otra parte, la carga de condensadores, seguida por comparadores analógicos, proporciona pulsos de anchura controlada por una tensión; de esta forma se configuran circuitos que pueden sustituir (si la velocidad de trabajo requerida no es muy elevada) a etapas típicamente analógicas mucho más complejas.
- ❖ En concreto, la modulación de anchura de pulsos resulta muy útil para el control de potencia (por el mecanismo todo/nada), para la conformación de potenciómetros digitales (para controlar la amplitud de señales o efectuar una conversión DC–DC) y, también, para efectuar sencillas conversiones de señal digital a analógica (de número a tiempo o a tensión) y de analógica a digital (de tensión a número).

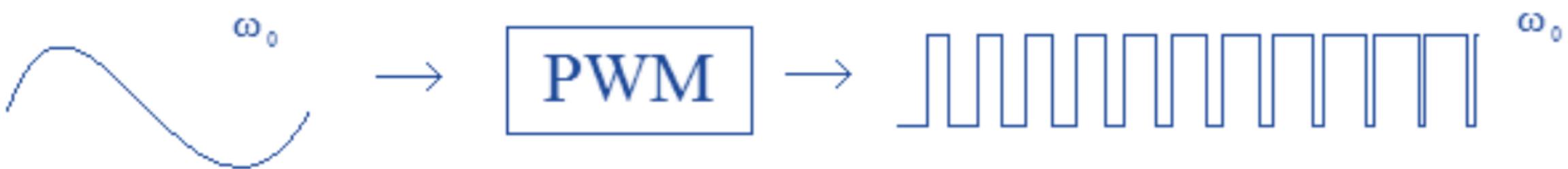
6.4.1.– Modulación por anchura de pulsos (PWM). Definición.

- PWM es una forma de codificar la información que utiliza, para ello, pulsos de anchura variable; es decir, en lugar de una señal que expresa la información a través de su amplitud (variaciones de tensión), se utilizan pulsos de amplitud fija cuya anchura (la duración de los pulsos) es variable, proporcional al valor de la señal en cada momento.



6.4.1.– Modulación por anchura de pulsos (PWM). Definición.

- El proceso de modulación PWM que se lleva a cabo en un convertidor consiste básicamente en transformar una señal de referencia sinusoidal de frecuencia fundamental ω_0 en otra señal modulada de ancho de pulso, como muestra la figura.



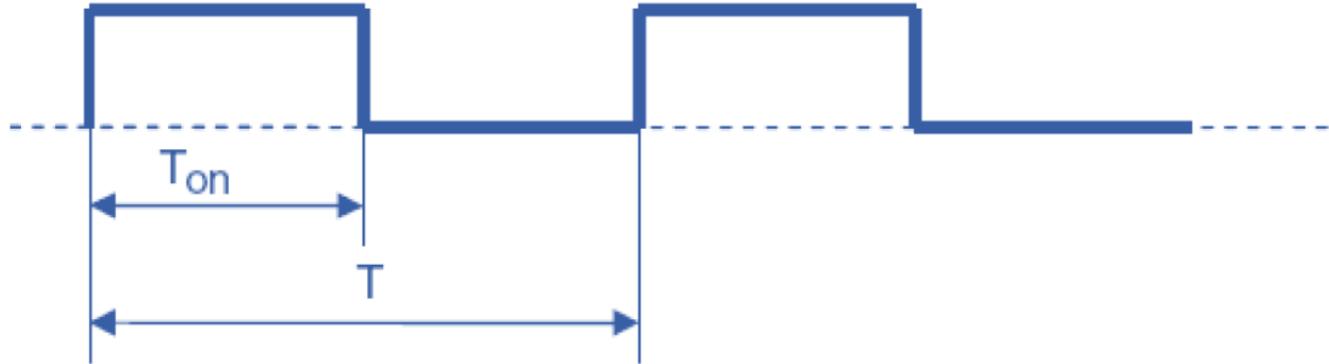
- Esta transformación consiste sencillamente en comparar la referencia con una señal portadora, típicamente triangular o diente de sierra, de modo que la salida esté en nivel alto cuando la referencia sea mayor que la portadora, y en nivel bajo en caso contrario.

6.4.1.– Modulación por anchura de pulsos (PWM). Definición.

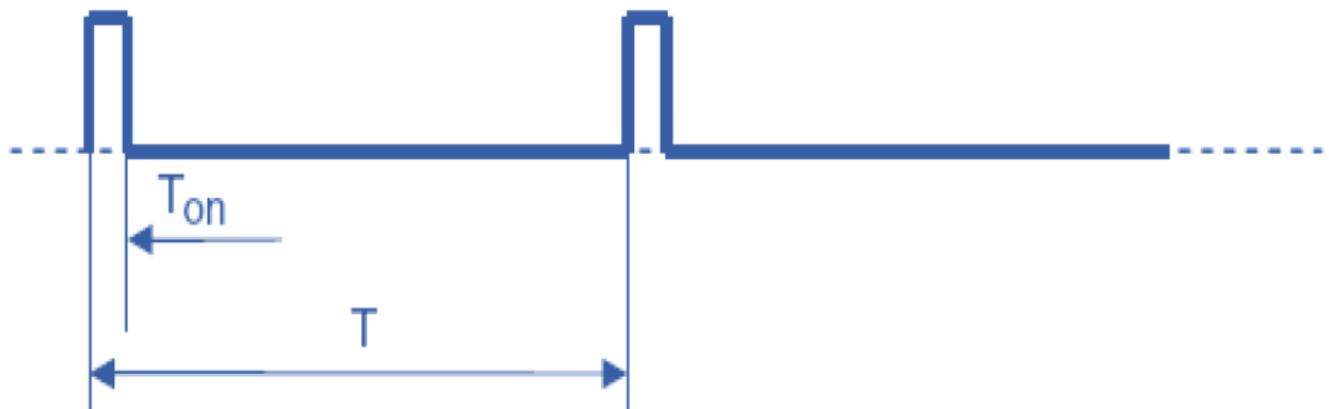
- Al ser distintas las señales de referencia y portadora en frecuencia, forma de onda y amplitud, el tiempo que pasará entre una intersección y la siguiente nunca será el mismo, de modo que los pulsos de la forma de onda de salida serán de distinta anchura, de ahí el nombre modulación de ancho de pulso.
- La técnica PWM consiste en producir un pulso rectangular con un ciclo de trabajo determinado que puede variar de 0 a 100%.
- Un ciclo de trabajo del 0% significa que la señal siempre está nivel bajo; y un ciclo de trabajo del 100% significa la señal siempre en nivel alto, el número de casos intermedios posibles es un número finito llamado resolución del PWM y se expresa como $\text{Log}_2(\text{número de casos})$.
- Por ejemplo, si puede haber 256 ciclos de trabajo posibles se dice que el PWM tiene una resolución de 8 bits.

6.4.1.– Modulación por anchura de pulsos (PWM). Definición.

- Pulso con ciclo de trabajo del 50%: $T_{on}/T = 0.5$



- Pulso con ciclo de trabajo del 10%; $T_{on}/T = 0.1$

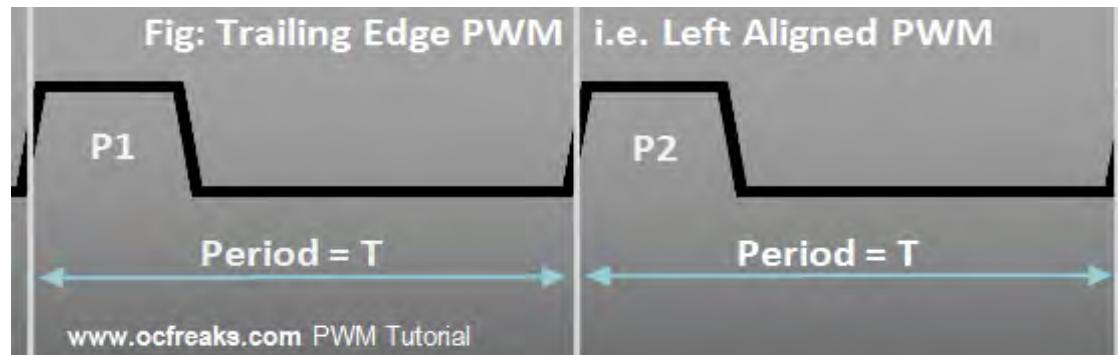


6.4.2.– Modulación por anchura de pulsos (PWM). Tipos.

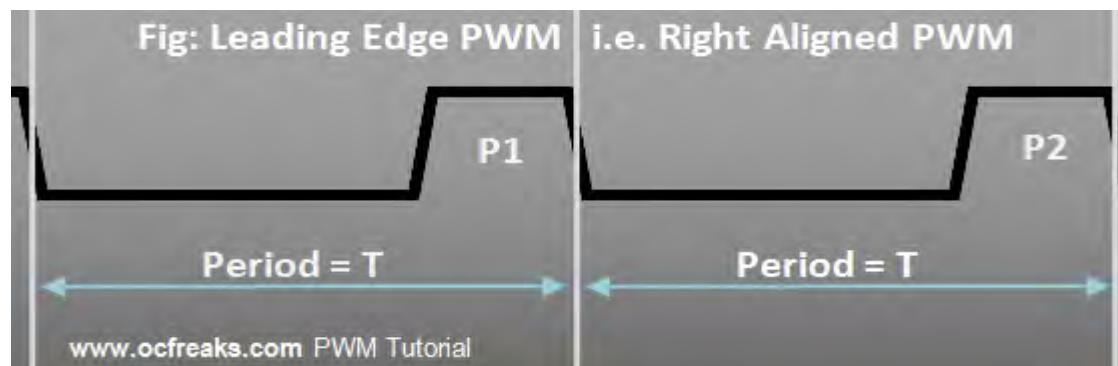
- ❖ La señal PWM se puede clasificar como:
 - *Single Edge PWM*.
 - *Double Edge PWM*.
- ❖ **PWM Single Edge:** (de borde único): el pulso puede ir al principio o al final del período y, por lo tanto, puede clasificarse en *PWM Trailing Edge* (alineado a la izquierda) y *PWM Leading Edge* (alineado a la derecha) .
 - En ***Trailing Edge PWM***, el flanco de subida se fija al comienzo de un período y el flanco de bajada está modulado, es decir, variado.
 - En ***Leading Edge PWM***, el flanco de bajada se fija al final de un período y el flanco de subida está modulado, es decir, variado.

6.4.2.– Modulación por anchura de pulsos (PWM). Tipos.

- Diagrama de PWM *Trailing Edge* (de flanco de bajada, alineado a la izquierda):

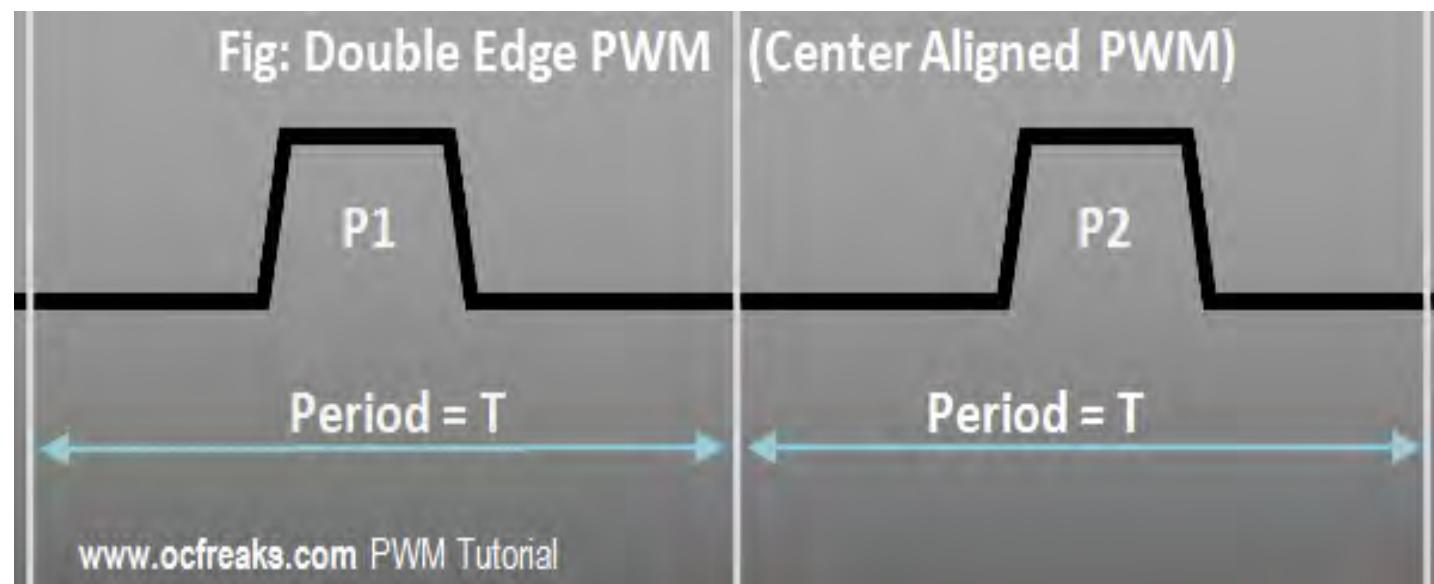


- Diagrama de PWM *Leading Edge* (de flanco de subida, alineado a la derecha)



6.4.2.– Modulación por anchura de pulsos (PWM). Tipos.

- Double Edge PWM: el pulso se puede colocar en cualquier lugar dentro del período. Se llama "Double Edge" porque ambos bordes están modulados o variados.
- En aplicaciones como el control de motor multifásico, Double Edge PWM se utiliza cuando el pulso está alineado al centro para reducir los armónicos.
- El diagrama para Double Edge PWM alineado al centro es el que se muestra a continuación:



6.4.3.– Modulación por anchura de pulsos (PWM). Aplicaciones.

- La modulación PWM es una técnica utilizada para regular la velocidad de giro de los motores eléctricos de inducción o asíncronos. Mantiene el par motor constante y no supone un desaprovechamiento de la energía eléctrica. Se utiliza tanto en corriente continua como en alterna.
- La modulación por ancho de pulsos también se usa para controlar servomotores, los cuales modifican su posición de acuerdo al ancho del pulso enviado cada cierto período que depende de cada servomotor.
- También se puede utilizar el PWM como parte de un conversor ADC, para enviar información de forma analógica. Es útil para comunicarse de forma analógica con sistemas digitales. Para un sistema digital, es relativamente fácil medir cuánto dura una onda cuadrada.
- El PWM en conjunción con un oscilador digital, un contador y una puerta AND como puerta de paso, podrían fácilmente implementar un ADC.

6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

- ❖ A simple vista, el modulador PWM del LPC23xx parece mucho más complicado que los temporizadores de propósito general. Sin embargo, realmente es un *timer* de propósito general con algún *hardware* adicional.
- ❖ El modulador de PWM puede generar hasta seis canales PWM *single edge* o hasta tres canales PWM *dual edge* controlados por este *hardware* adicional.



6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

- ❖ Los generadores de señales PWM son similares a los temporizadores de propósito general con características adicionales. Son dos las diferencias más importantes entre la estructura de los temporizadores básicos y la estructura de los moduladores de PWM:
 - ✓ La comparación del “*timer counter*” no se realiza directamente con el registro *match*, sino con otro registro llamado “*Shadow register*”.
 - ✓ Los canales *match* no controlan directamente la salida de los pines como en el caso de los temporizadores básicos.

6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

💡 ¿Por qué no se pueden utilizar los temporizadores de propósito general para generar la señal PWM en un pin?

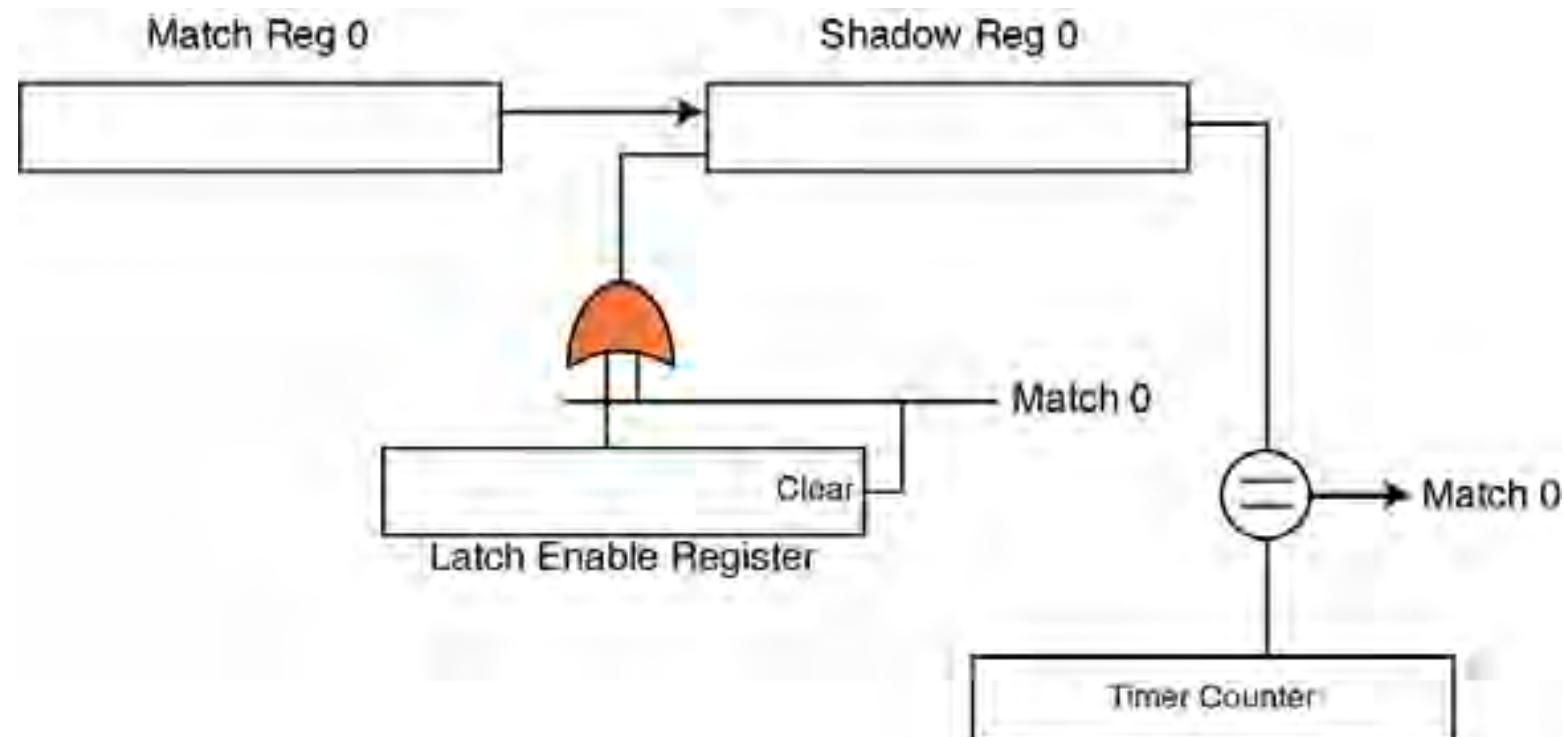
👉 Cuando se actualizan los registros *match* de un *timer* de propósito general, los cambios tienen efectos inmediatos, si no se tiene cuidado en el programa. Aun codificando cuidadosamente, si se quieren tener varios canales, los nuevos valores de PWM se llevarán a efecto, en diferentes puntos del ciclo y puede provocar resultados inesperados. Además, habría que cambiar el valor del pin con una precisión imposible de conseguir por programa.

6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

- ❖ Efectos indeseados al generar la señal PWM con los temporizadores básicos (ejemplo):
 - Se está generando la señal PWM para, por ejemplo, modificar el ángulo que gira un servomotor y queremos modificarlo . En el caso de un servomotor la anchura del pulso tiene relación con el ángulo que gira el servomotor.
 - Supongamos que uno de los registros *match* lo utilizamos para controlar el tiempo a nivel alto de la señal PWM *single edge*. Mantenemos el pulso en alto el tiempo que tarda la cuenta del *timer* en alcanzar el valor del registro *match*.
 - Supongamos ahora que queremos que el pulso sea más estrecho, para provocar un giro en el servomotor, y cargamos el nuevo valor en el registro *match*. Pero, podría haber ocurrido que esta nueva cuenta sea inferior a la que lleve el *timer–counter* en el momento de la actualización. En este caso, hasta que el *timer–counter* no se desborde y alcance la nueva cuenta no se producirá el final del pulso en alto. Como consecuencia el pulso habrá tenido una duración en alto mucho mayor de lo que queremos y como consecuencia el servomotor habrá girado un ángulo indeseado y cuando se tenga el nuevo pulso volverá de nuevo a posicionarse donde se deseaba.

6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

- El modulador PWM tiene un registro de tipo *latch* compartido “*Shadow register*” que permite que los valores PWM no sean actualizados inmediatamente, sino que los nuevos valores sólo se llevan a efecto al comienzo del nuevo ciclo.

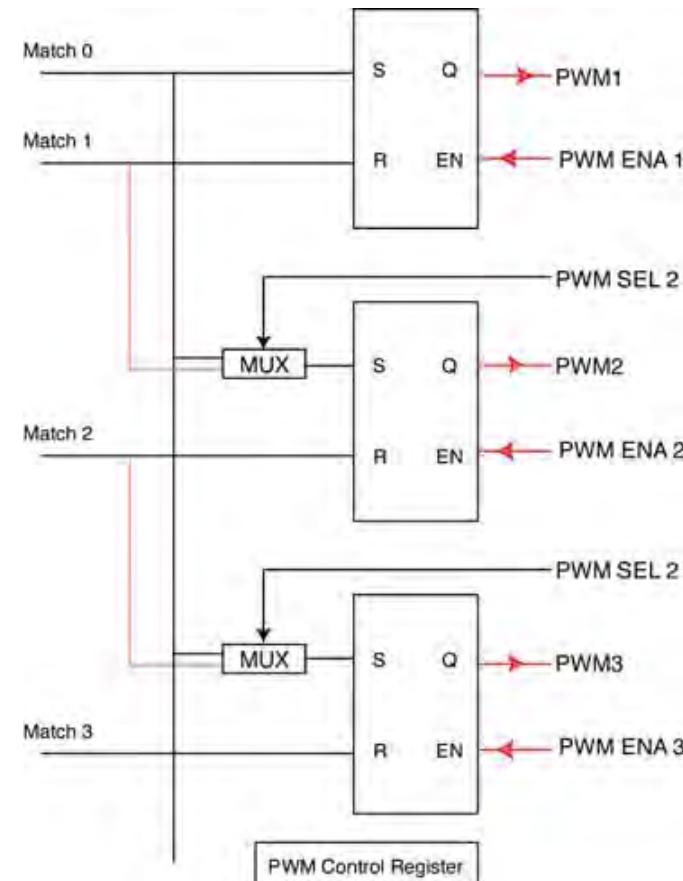


6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

- La comparación del *timer–counter* se realiza con el registro *shadow* en lugar de con el registro *match* como ocurre con los temporizadores básicos. Los nuevos valores en los registros *match* harán que se genere una señal PWM con las nuevas características, pero no tiene efecto hasta que no se carguen en el registro *shadow*. Para que los nuevos valores se carguen en el registro *shadow*, la cuenta debe alcanzar el valor del registro *shadow* y habilitar la carga del nuevo valor del registro *match* escribiendo en el registro “*latch enable register*”.
- Si utilizamos el control del servomotor (ejemplo anterior) con el modulador de PWM, la actualización de la nueva cuenta no se realiza hasta el comienzo del ciclo gracias al registro *shadow*.

6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

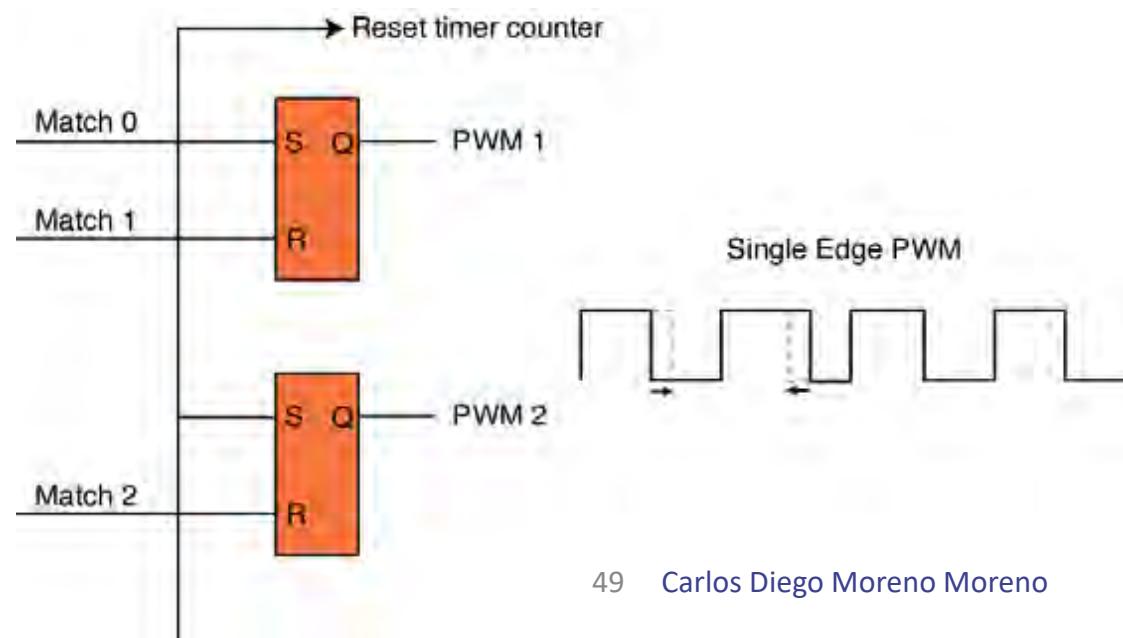
- La circuitería adicional a la salida *match* de los canales permite la generación de seis canales *single edge* o tres canales *double edge* dependiendo del multiplexor.
- Cuando se quiere generar modulación *single edge*, se utiliza el registro *match0* que controla el periodo de la señal y pone a uno la salida PWM para todos los canales, y un registro *match* por cada canal para poner a cero la salida. Como se tienen siete registros *match* se consiguen así los seis canales.
- Cuando se quiere generar modulación *double edge* se necesita el registro *match0* para controlar el periodo de la señal y dos registros más: uno para la puesta a uno de la salida PWM y otro para poner la salida a cero. Se tendrá como máximo tres canales ya que se tienen siete registros *matchs*.



6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

Modulación por anchura de pulsos *single edge*:

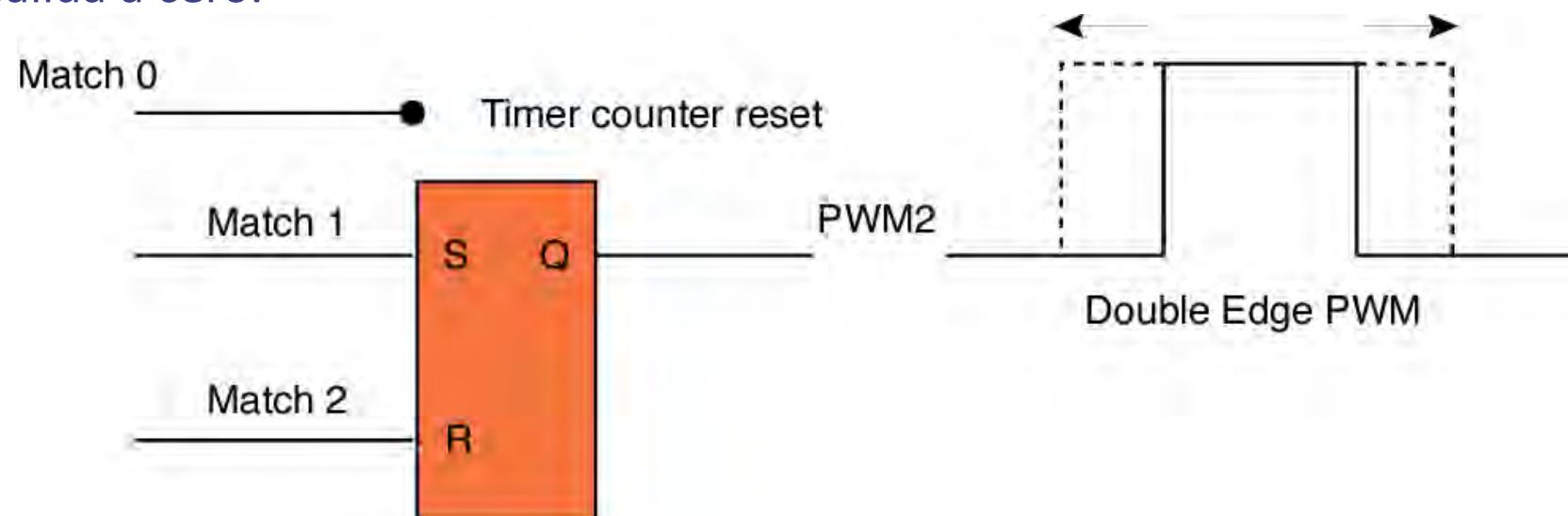
- El registro *match0* es el que fija el momento en el que el *timer–counter* se resetea y por tanto este registro establece el periodo de la señal tanto para *single edge* como *double edge*.
- Cuando se iguale la cuenta del *timer–counter* con el registro *match0* se activará la entrada *set* del *latch S–R* y se pondrá la salida a 1, a la vez que se resetea el *timer–counter*.
- Cuando el *timer–counter* alcance el valor del registro *match1* se activará la entrada *reset* del *latch S–R* y se pondrá la salida a cero.



6.4.4.– Modulación por anchura de pulsos (PWM). Modulador PWM del LPC23xx.

Modulación por anchura de pulsos *double edge*:

- El registro *match0* es el que fija el momento en el que el *timer–counter* se resetea y por tanto este registro establece el periodo de la señal tanto para *single edge* como *double edge*.
- Cuando se iguale la cuenta del *timer–counter* con el registro *match1* se activará la entrada *set* del *latch S–R* y se pondrá la salida a 1.
- Cuando el *timer–counter* alcance el valor del registro *match2* se activará la entrada *reset* del *latch S–R* y se pondrá la salida a cero.



6.4.5.– Modulación por anchura de pulsos (PWM). Registros PWM del LPC23xx.

- Registros de configuración y control:
 - **PWMTCR (Timer Control Register)**: Activa o reinicia el contador del PWM.
 - **PWMPCR (Prescale Register)**: Define el divisor de la frecuencia del reloj para reducir la velocidad del contador.
- Registros de conteo y límite:
 - **PWMTC (Timer Counter)**: Almacena el valor actual del contador.
 - **PWMPMC (Prescale Counter Register)**: Se incrementa hasta el valor de PWMPMR y luego reinicia TC.
 - **PWMMR0–PWMMR6 (Match Registers)**: Determinan los puntos de comparación para generar el ciclo de trabajo de la señal PWM.
- Registros de configuración del ciclo de trabajo:
 - **PWMLER (Latch Enable Register)**: Habilita la actualización de los registros de coincidencia (*Match Registers*).
 - **PWMPCR (PWM Control Register)**: Activa las salidas PWM individuales y configura el modo simple o doble.
- Registros de interrupción:
 - **PWMIR (Interrupt Register)**: Indica qué coincidencias han generado interrupciones.
 - **PWMMCR (Match Control Register)**: Configura qué eventos generan interrupciones o reinician el contador.

6.4.5.– Modulación por anchura de pulsos (PWM). Registros de captura.

- El módulo PWM del LPC2378 también incluye registros de captura, que permiten almacenar el valor del contador cuando ocurre un evento externo en los pines de captura. Esto es útil para medir la duración de señales de entrada, como la frecuencia o el ciclo de trabajo de una señal externa.
- Estos registros permiten capturar el valor del *PWM Timer Counter* (PWMT) cuando ocurre una transición en un pin de captura.
 - **PWM_{CAP0} y PWM_{CAP1} (Capture Registers)**: Almacenan el valor del contador PWMT cuando se detecta un evento en el pin PWM_CAP0 o PWM_CAP1.
 - **PWMCCR (Capture Control Register)**: Controla el comportamiento de los registros de captura, como el flanco activo de la captura o si genera una interrupción.
 - **PWMIR (Interrupt Register)**: Indica si se ha producido una captura en PWM_{CAP0} o PWM_{CAP1}, permitiendo generar interrupciones en la CPU.
- El uso típico de los registros de captura es medir la duración de un pulso o la frecuencia de una señal externa.
- Este mecanismo es útil en aplicaciones como tacómetros, medición de señales PWM externas o sincronización de eventos.

TEMA 6: Periféricos de usuario.

6.1.– Puertos de entrada/salida digitales.

6.2.– Temporizadores.

6.3.– Temporizador de vigilancia (*WatchDog Timer*, WDT).

6.4.– Modulación por anchura de pulsos (PWM).

 6.4.1.– Definición.

 6.4.2.– Tipos.

 6.4.3.– Aplicaciones.

 6.4.4.– Modulador PWM del LPC23xx.

6.5.– Reloj en tiempo real (RTC).

6.6.– Conversor Analógico/Digital.

6.7.– Conversor Digital/Analógico.

6.5.– Reloj en tiempo real (RTC)

6.5.1.– Funcionamiento.

- ❖ El reloj de tiempo real (RTC) es un conjunto de contadores para medir el tiempo cuando la alimentación del sistema es baja si está encendido y, opcionalmente, cuando está apagado.
- ❖ Consumen poca energía en modo de apagado.
- ❖ El RTC puede sincronizarse mediante un oscilador independiente de 32,768 KHz o mediante un divisor de frecuencia programable basado en el reloj PCLK.
- ❖ Además, el RTC funciona con su propio pin de alimentación, VBAT, que se puede conectar a una batería o a la misma 3,3 V utilizada por el resto del dispositivo.

6.5.– Reloj en tiempo real (RTC)

6.5.1.– Funcionamiento.

- ❖ El reloj en tiempo real (RTC) de que dispone el LPC23xx es un reloj–calendario de precisión con alarma válido hasta el año 2099.
- ❖ Tiene asociado 2KB de SRAM de bajo consumo llamada batería RAM.
- ❖ Ambos están conectados al bus APB.
- ❖ El RTC y la batería SRAM tienen un suministro de energía separado, por lo que al suministrar 3,3 V al pin VBAT, el RTC puede mantenerse en funcionamiento y el contenido de la RAM se puede guardar cuando se desconecta el LPC23xx.
- ❖ Tanto el RTC como la batería RAM consumen poca potencia y pueden funcionar con baterías.

6.5.– Reloj en tiempo real (RTC)

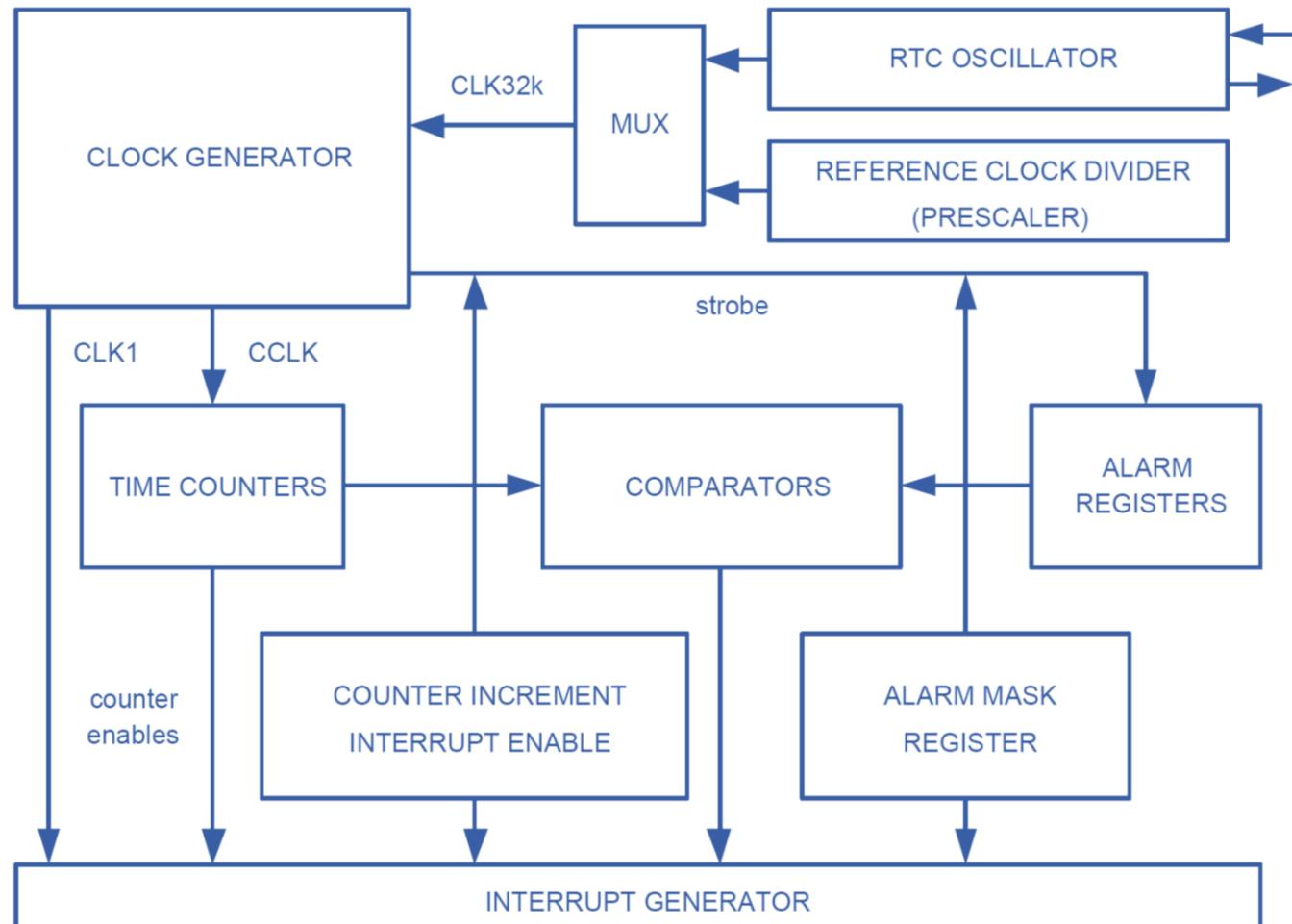
6.5.1.– Funcionamiento.

- 💡 Cuando la CPU y otras funciones del *chip* se detienen y se corta el suministro de energía, el RTC puede suministrar una salida de alarma que puede ser utilizada por algún *hardware* externo para restaurar la energía del *chip* y reanudar la operación.
- 💡 La salida de alarma tiene una oscilación de voltaje nominal de 1,8 V.

6.5.- Reloj en tiempo real (RTC)

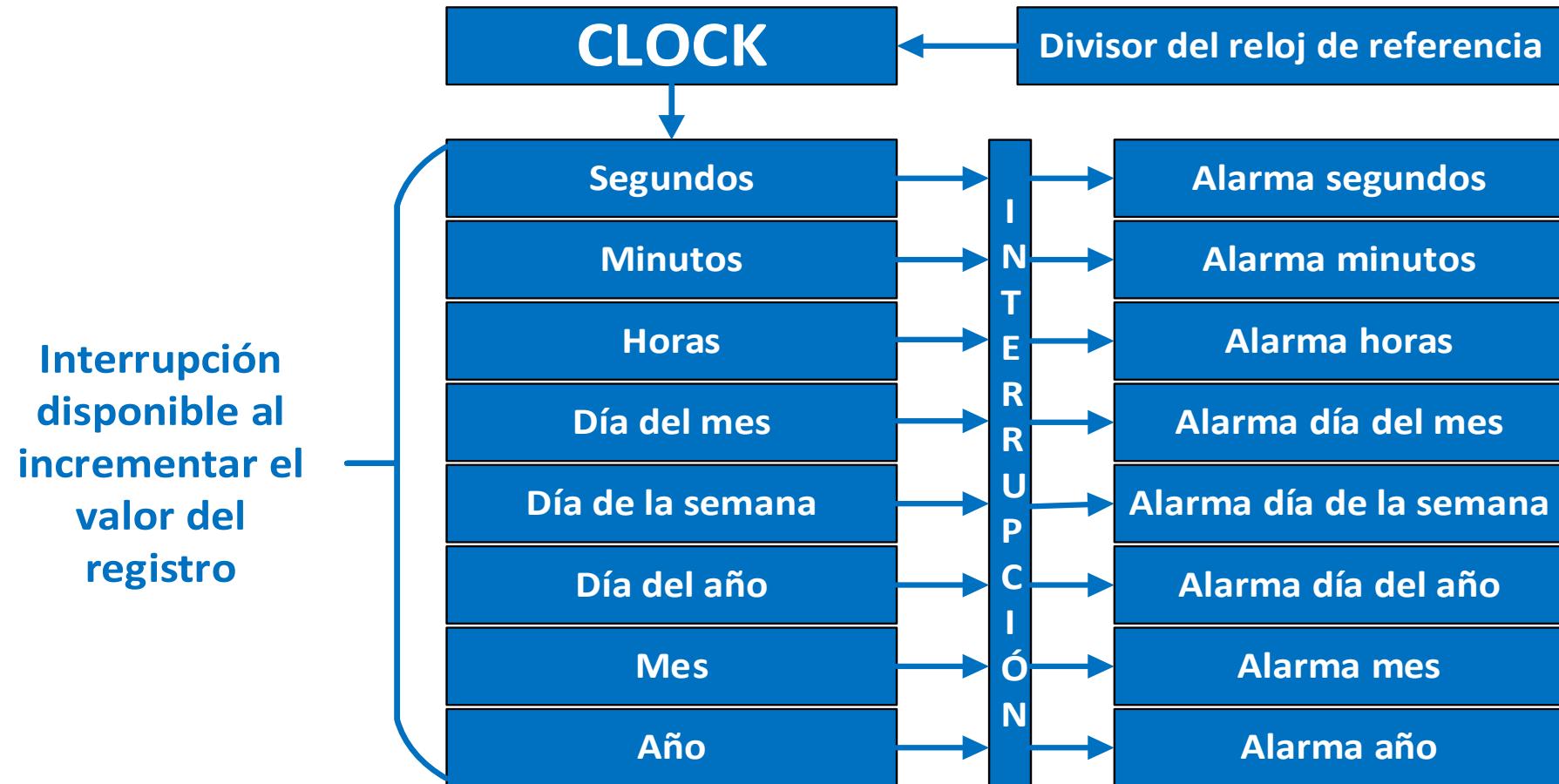
6.5.1.- Funcionamiento.

Diagrama de bloques:



6.5.- Reloj en tiempo real (RTC)

6.5.1.- Funcionamiento.



6.5.– Reloj en tiempo real (RTC)

6.5.1.– Funcionamiento.

- ❖ Esto permite que el RTC proporcione un calendario de reloj perpetuo y, si no es necesario, se puede utilizar para entregar una referencia temporal e interrupciones periódicas sin necesitar un oscilador externo adicional.
- ❖ Hay dos referencias de tiempo disponibles para el RTC: el oscilador externo de 32 KHz y el PCLK interno.
- ❖ El reloj RTC funciona con una frecuencia de cristal de reloj estándar de 32,768 KHz que se obtiene a partir de un cristal de reloj externo conectado a los pines RTCX1 y RTCX2.
- ❖ Si este oscilador está instalado, se puede seleccionar como fuente del reloj configurando el *bit* CLKSRC en el registro de control del reloj.

6.5.– Reloj en tiempo real (RTC)

6.5.1.– Funcionamiento.

- ❖ Si no se necesita un calendario el RTC se puede sincronizar desde PCLK borrando el *bit* CLKSRC.
- ❖ Para derivar la frecuencia de 32,768 KHz, PCLK está conectado al divisor del reloj de referencia.
- ❖ La salida de este divisor luego se pasa al RTC.
- ❖ Este es un pre–escalador que puede dividir con precisión cualquier frecuencia PCLK para producir la frecuencia requerida de 32 KHz.
- ❖ Para garantizar que el reloj RTC se pueda obtener a partir de cualquier PCLK, el pre–escalador es más complejo que los pre–escaladores de los temporizadores genéricos.

6.5.– Reloj en tiempo real (RTC)

6.5.1.– Funcionamiento.

- El pre-escalador está programado por dos registros llamados PREINT y PREFRAC que contienen la parte entera y fraccionaria del divisor. Se calculan así:

$$\text{PREINT} = (\text{int}) (\text{PCLK}/32768) - 1$$

$$\text{PREFRAC} = \text{PCLK} - ((\text{PREINT}+1) \times 32768)$$

- Para PCLK = 30MHz:

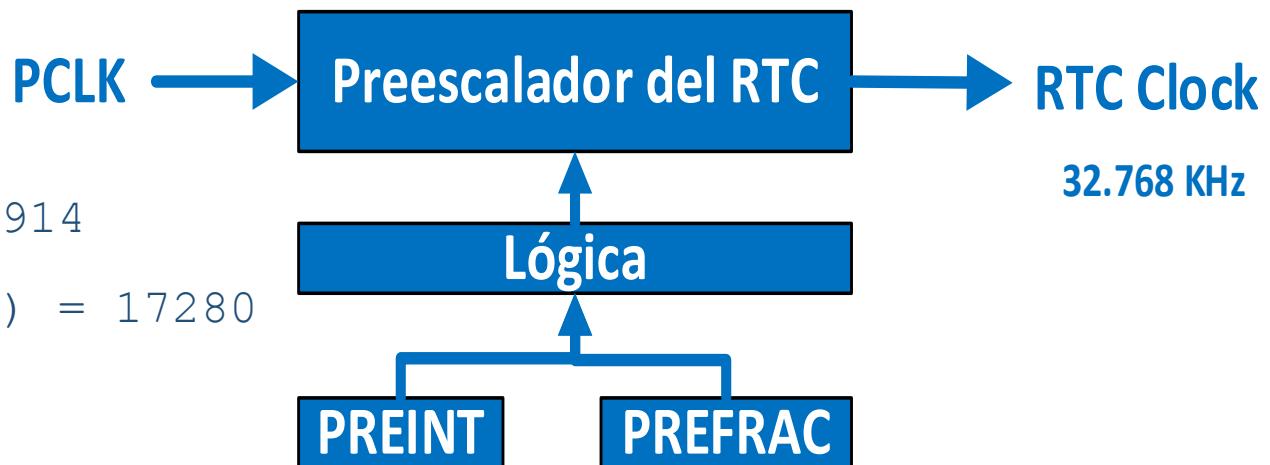
$$\text{PREINT} = (\text{int}) (30.000.000/32768) - 1 = 914$$

$$\text{PREFRAC} = 30.000.000 - ((914+1) \times 32768) = 17280$$

- Programación de los registros:

PREINT = 0x00000392;

PREFRAC = 0x00004380;



6.5.– Reloj en tiempo real (RTC)

6.5.1.– Funcionamiento.

- ❖ Hay ocho registros para contar el tiempo, cada uno de los cuales contiene una única cantidad de tiempo que se puede leer en cualquier momento.
- ❖ Además, existe un conjunto de registros de consolidación que presentan las mismas cantidades de tiempo en tres palabras, lo cual permite leer toda la información del tiempo en tres operaciones.
- ❖ Los registros de tiempo consolidados son de solo lectura. Para escribir nuevos valores en los contadores de tiempo, se deben usar las direcciones de los contadores de tiempo.

6.5.– Reloj en tiempo real (RTC)

6.5.1.– Funcionamiento.

Registro de consolidación 0 CTIME0	Reservado		DÍA DE LA SEMANA							Reservado		HORAS			Reserva do			MINUTOS			Reserva do			SEGUNDOS				
	31					27	26	25	24	23		21	20				16	15	14	13				8	7	6	5	
Registro de consolidación 1 CTIME1	Reservado		AÑO												Reservado			MES			Reservado			DÍA DEL MES				
	31			28	27										16	15		12	11		8	7	6	5	4		0	
Registro de consolidación 2 CTIME2	Reservado												DÍA DEL AÑO															
	31																	12	11									0

6.5.– Reloj en tiempo real (RTC)

6.5.1.– Funcionamiento.

─ Además de mantener un reloj, el RTC también puede generar eventos de alarma como interrupciones. Existen dos mecanismos de interrupción:

- ─ Se puede programar el RTC para generar una interrupción cuando se incrementa cualquier registro del contador de tiempo, por lo tanto, podría generar una interrupción cada segundo cuando se actualice el segundo contador o una vez al año cuando el contador de años se incrementa. El registro de interrupción de incremento del contador le permite habilitar un incremento interrupción para cada uno de los ocho registros del contador de tiempo.
- ─ El segundo método para generar una interrupción RTC es con los registros de alarma. Cada registro contador de tiempo tiene un registro de alarma de coincidencia. Si se desenmascara este registro de alarma *match*, se compara con el registro del contador de tiempo. Si todos los registros de alarma desenmascarados coinciden con los registros del contador de tiempo, se genera una interrupción. Así es posible configurar una alarma entre ahora y 2099 con una precisión de un segundo. El registro de máscara de alarma controla qué registros de alarma se utilizan en la comparación. Como tanto el evento de incremento como el de alarma pueden generar una interrupción es necesario distinguir entre ellos desde dentro de la interrupción. El registro de ubicación de interrupción proporciona dos indicadores que se pueden interrogar para ver qué causó la interrupción del RTC. De nuevo, recuerda que estas banderas deben borrarse para cancelar la interrupción.

6.5.– Reloj en tiempo real (RTC)

6.5.2.– Descripción de los pines.

- 💡 Si no se utiliza el RTC, los pines RTCX1/2 se pueden dejar sin conectar.
- 💡 Si se utiliza el RTC, VBAT debe conectarse a cualquiera de los pines VDD_{3V3} o a una fuente de alimentación independiente (batería externa). En caso contrario, VBAT debería dejarse sin conectar.

Pines	Descripción
RTCX1	Entrada al circuito oscilador RTC
RTCX2	Salida del circuito oscilador RTC
VBAT	Fuente de alimentación del RTC (3,3 V)

6.5.– Reloj en tiempo real (RTC)

6.5.3.– Descripción de los registros.

Grupo de registros variados	
Registro	Descripción
ILR	Registro de localización de la interrupción. Especifica qué bloques están generando una interrupción.
CTC	Registro contador de <i>ticks</i> del reloj. Es de sólo lectura y contiene los bits del contador divisor del reloj.
CCR	Registro de control del reloj. Habilitación, <i>reset</i> y selección del reloj.
CIIR	Registro de interrupción de incremento de contador. Genera una interrupción cada vez que el contador se incrementa.
CISS	Registro de máscara de selección del incremento del contador. Permite obtener interrupciones periódicas de la CPU del orden de ms desde el RTC.
AMR	Registro de máscara de la alarma. Controla cuál de los registros de alarmas están enmascarados.
CTIME0	Registro de tiempo consolidado 0. Contiene los valores de tiempo de orden inferior: segundos, minutos, horas y día de la semana.
CTIME1	Registro de tiempo consolidado 1. Contiene los valores de día del mes, mes y año.
CTIME2	Registro de tiempo consolidado 1. Valor del día del año en el rango de 1 a 365 (366 en año bisiesto).

6.5.– Reloj en tiempo real (RTC)

6.5.3.– Descripción de los registros.

Grupo de contadores de tiempo	
Registro	Descripción
SEC	Valor de los segundos (0 a 59)
MIN	Valor de los minutos (0 a 59)
HOUR	Valor de las horas (0 a 23)
DOM	Valor del día del mes (1 a 28, 29, 30, o 31; dependiendo del mes y si es un año bisiesto).
DOW	Valor del día de la semana (0 a 6)
DOY	Valor del día del año (1 a 365 o 366 si el año es bisiesto)
MONTH	Valor del mes (1 a 12)
YEAR	Valor del año (0 a 4095)

6.5.– Reloj en tiempo real (RTC)

6.5.3.– Descripción de los registros.

Grupo de registros de alarma	
Registro	Descripción
ALSEC	Valor de la alarma para los segundos.
ALMIN	Valor de la alarma para los minutos.
ALHOUR	Valor de la alarma para las horas.
ALDOM	Valor de la alarma para el día del mes.
ALDOW	Valor de la alarma para el día de la semana.
ALDOY	Valor de la alarma para el día del año.
ALMONTH	Valor de la alarma para el mes.
ALYEAR	Valor de la alarma para el año.

TEMA 6: Periféricos de usuario.

6.1.– Puertos de entrada/salida digitales.

6.2.– Temporizadores.

6.3.– Temporizador de vigilancia (*WatchDog Timer*, WDT).

6.4.– Modulación por anchura de pulsos (PWM).

 6.4.1.– Definición.

 6.4.2.– Tipos.

 6.4.3.– Aplicaciones.

 6.4.4.– Modulador PWM del LPC23xx.

6.5.– Reloj en tiempo real (RTC).

6.6.– Conversor Analógico/Digital.

6.7.– Conversor Digital/Analógico.

6.6.– Conversor Analógico/Digital

- El microcontrolador LPC2378 incorpora ocho canales para conversión analógica–digital, multiplexa estas entradas para un solo convertidor analógico–digital de 10 *bits* que funciona con un reloj máximo de 4,3 MHz, el máximo tiempo de conversión con 10 *bits* a esta frecuencia es de 2,44 μ s.
- Se presenta la opción de cambiar la cantidad de *bits* de conversión, y de iniciar la conversión por diferentes métodos.
- Este módulo presenta la característica de poder deshabilitarse a través de un *bit* del registro PCONP.
- El conversor analógico/digital que tienen algunas variantes del LPC2300 es un conversor de aproximaciones sucesivas de 10 *bits* con un tiempo de conversión de 2,44 μ Sec o por debajo de 410 kbps. Dispone de 6 u 8 entradas multiplexadas en función de la variante LPC23xx.

Nombre	Descripción
AD0CR	Registro de control del ADC
AD0GDR	Registro global de datos
AD0STAT	Registro de estado
ADOINTEN	Registro de habilitación de las interrupciones
AD0DR0–7	Registro de datos del canal 0–7

6.6.– Conversor Analógico/Digital

- El **registro de control (ADxCR)** establece la configuración del conversor y controla el inicio de la conversión. El primer paso para configurar el conversor es configurar el reloj. Como en los demás periféricos, el reloj A/D se obtiene a partir del PCLK, dividiéndolo para obtener 4.5MHz. Este es el valor máximo y si PCLK no puede ser dividido para obtener 4.5MHz entonces se debe seleccionar el valor más cercano para conseguirlo por debajo de 4.5MHz.

				ED GE	START			PDN		CLKS			BURST	CLKDIV							SEL				
31			28	27	26	25	24			21		19	18	17	16	15					8	7			0

- SEL:** Selecciona los pines que van a ser convertidos.
- CLKDIV:** El reloj del bus APB (PCLK) se divide por este valor más uno para obtener el reloj del ADC, debe ser menor o igual a 4.5 MHz.
- BURST:** Cuando este *bit* vale cero, la conversión es controlada por *software* y requiere 11 ciclos de reloj.
- CLKS:** Número de ciclos utilizados para cada conversión cuando BURST=1. Si BURST=0 debe estar a 000, utilizando 10 *bits*.
- PDN:** Si está a uno se habilita el ADC, si está a cero está en modo *power-down*.
- START :** Cuando BURST=0, estos *bits* controlan si se produce la conversión o bajo qué condiciones.
- EDGE:** Este *bit* sólo se utiliza cuando START es 010 ó 111.

6.6.– Conversor Analógico/Digital

-  El registro de habilitación de las interrupciones (ADxINTEN) es el siguiente:

No usado																ADGINTEN	ADINTEN 7:0				
31																9	8	7			0

- ADINTEN 7:0: Estos **bits** permiten controlar qué canales A/D generan una interrupción al completar la conversión. Cuando el *bit* 0 es uno, la finalización de una conversión en el canal A/D0 generará una interrupción, cuando el *bit* 1 es uno, la finalización de una conversión en el canal A/D1 generará una interrupción, etc.
 - ADGINTEN: Cuando vale 1, habilita el indicador DONE del ADGDR para generar una interrupción. Cuando vale 0, sólo los canales A/D individuales habilitados por ADINTEN 7:0 generarán una interrupción.

6.6.– Conversor Analógico/Digital

- El registro de datos del ADC (ADxDR0–7): Contiene el resultado de la conversión, error de desbordamiento e indicador de conversión realizada.

DONE	OVERRUN	No usado														V/V _{REF}								No usado							
31	30	29													16	15								6	5					0	

- DONE:** Este *bit* se pone a “1” cuando se completa una conversión A/D. Al leer este registro se coloca a “0”.
- OVERRUN:** Este *bit* es 1 en modo ráfaga si los resultados de una o más conversiones se ha perdido y se sobrescribió antes de la conversión que produjo el resultado en los *bits* menos significativos. Este *bit* se borra al leer este registro.
- V/V_{REF}:** Cuando DONE es 1, este campo contiene la división del voltaje en el pin A_{in} entre el voltaje en el pin V_{REF} . Cuando este campo es 0x0000 indica que el voltaje en el pin A_{in} es menor o igual o cercano a V_{REF} , mientras que 0x03FF indica que el voltaje en A_{in} era cercano, mayor o igual a V_{REF} .

TEMA 6: Periféricos de usuario.

6.1.– Puertos de entrada/salida digitales.

6.2.– Temporizadores.

6.3.– Temporizador de vigilancia (*WatchDog Timer*, WDT).

6.4.– Modulación por anchura de pulsos (PWM).

 6.4.1.– Definición.

 6.4.2.– Tipos.

 6.4.3.– Aplicaciones.

 6.4.4.– Modulador PWM del LPC23xx.

6.5.– Reloj en tiempo real (RTC).

6.6.– Conversor Analógico/Digital.

6.7.– Conversor Digital/Analógico.

6.7.– Conversor Digital/Analógico

- Todas las variantes del LPC23xx disponen de un conversor digital/analógico de 10 bits. Es un periférico muy fácil de utilizar ya que consta solamente de un simple registro (DACR).
- El DAC se habilita con los pines 20 y 21 del PINSEL1 y convirtiendo el pin 0.26 del GPIO a la función AOUT. Hay que señalar que un canal del conversor analógico/digital comparte este pin.
- Una vez habilitada la conversión, ésta comienza al escribir los bits VALUE en el registro de control. El tiempo de conversión depende del bit BIAS: si está a “uno” es 2,5 µs y puede controlar 700 µA; si está a “cero” el tiempo de conversión es 1 µs pero sólo es capaz de entregar 350 µA.

No usado								BIAS	VALOR A CONVERTIR								No usado							
31								17	16	15							6	5						0



Bibliografía.

- “The Insider’s guide to the NXP LPC2300/2400 Based Microcontrollers. An engineer’s introduction to the LPC2300 & LPC2400 series”. <http://www.hitex.co.uk>.
- LPC23XX User manual. UM10211. NXP Semiconductors (Philips). Revision 4.1 5 September 2012. : <http://www.nxp.com>.

¡Muchas gracias por su atención!

Carlos Diego Moreno Moreno



Área de Arquitectura y Tecnología de Computadores
Departamento de Ingeniería Electrónica y de Computadores.

Escuela Politécnica Superior. Universidad de Córdoba