

# LPCXpresso\_1769

## TIMER

## Bloques de LPC1769 vistos hasta ahora

- PIN CONNECT BLOCK (seleccionamos la función de cada pin individual)
- GPIO (utilizamos para controlar pines de entrada/salida digital)
- Clocking and Power Control (Clock en general y encendido de los periféricos)
- NVIC (habilitar/deshabilitar, setear prioridad, ver pendientes/activas, etc de interrupciones)
- System Control (utilizamos para configurar interrupciones externas)

## Timer

Tenemos 4 módulos TIMER0/1/2/3 disponibles en el microcontrolador

Cada contador tiene 32 bits

Cada prescale registers tiene 32 bits

Permite Match

Permite captura

Permite Manipular pines directamente desde el módulo TIMER

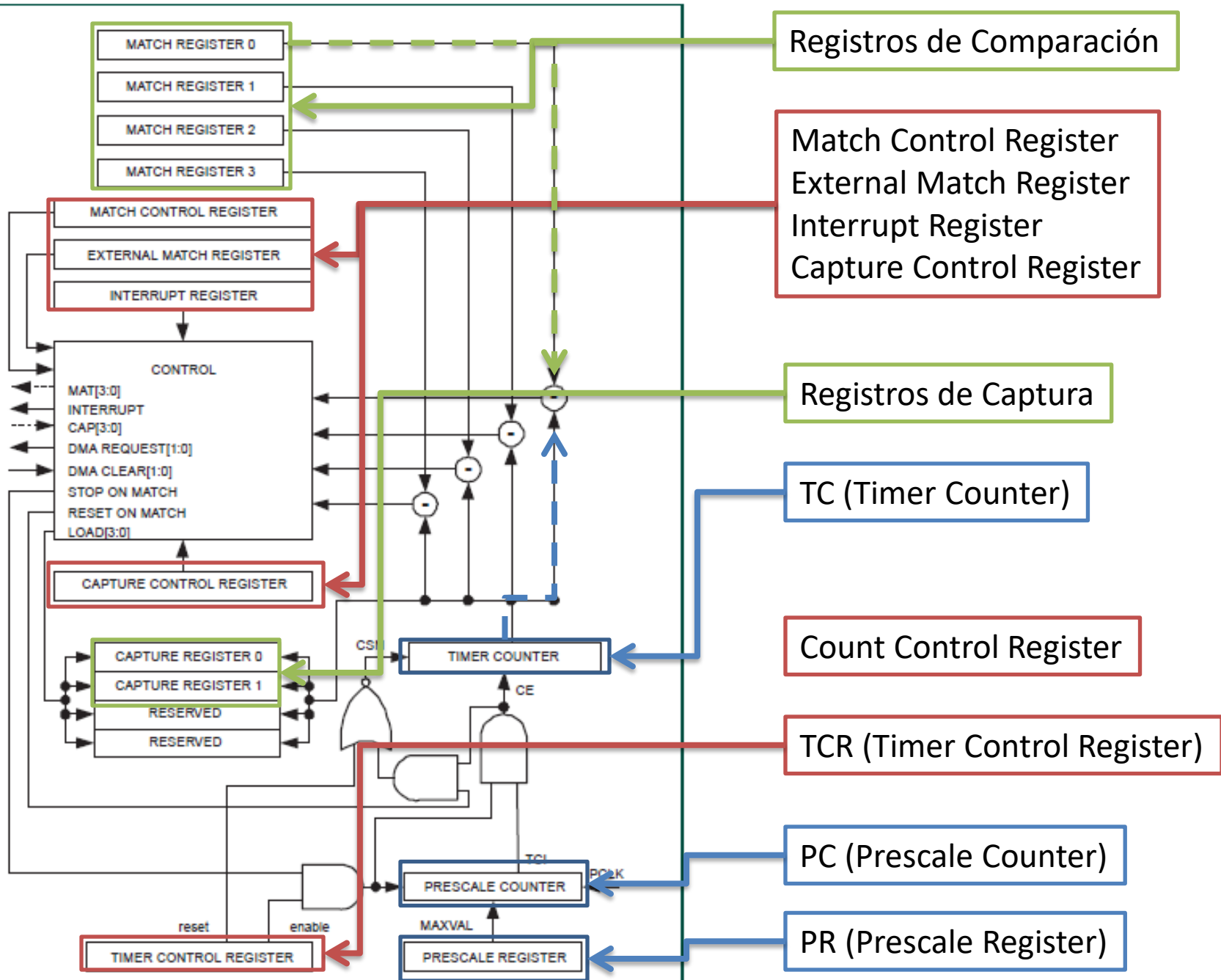


Fig 117. Timer block diagram

TC (Timer Counter) =>32 bits=4.294.967.296

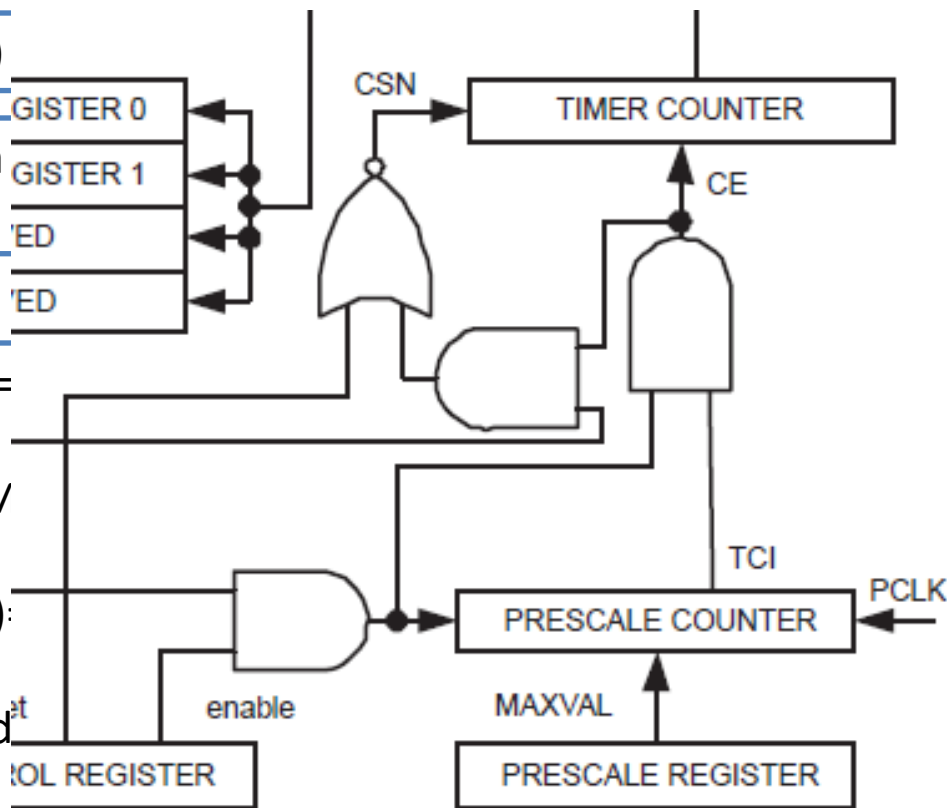
Este registro se incrementa en cada PR+1

PC (Prescale Counter)

Este registro incrementa en 1 por cada ciclo del Pclk (CClk/4=25Mhz)

PR (Prescale Register)

Cada vez que PC llega a cero



TC (Timer Counter) =

Con un PCLK de 25M

$(2^{32}) / (25.000.000)$

Si tenemos que medir  
resolución de 40ns

Si necesitamos medir

pérdida de resolución asociada

1 y PC se resetea

dir con una

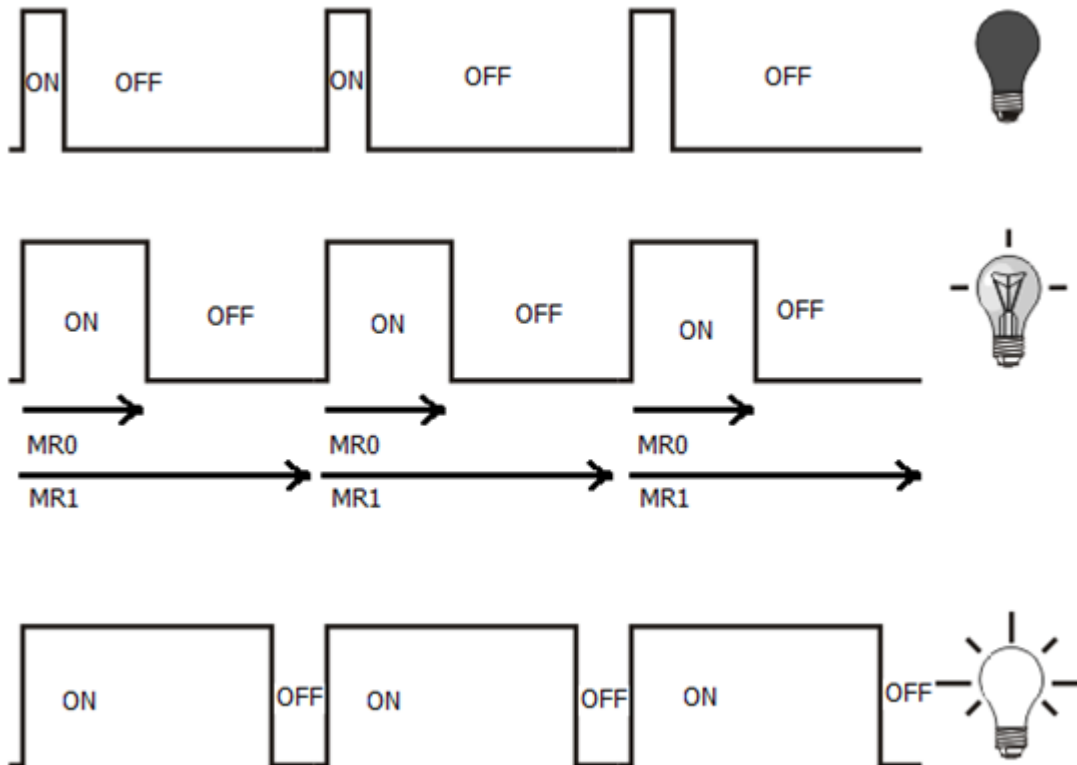
LE (PR) con la

## MR0-MR3 Registros de Comparación

TC se compara constantemente con los MRx, cuando TC iguala alguno de los MRx puedo:

- Disparar una interrupción
- Volver el TC a cero
- Detener el TC
- Controlar un Pin (MATn.x, donde la n indica el número de timer y la x a cual de los 4 match se hace referencia)

Ejemplo de aplicación, generación PWM, dimmer, control de motor DC o paso a paso, etc.



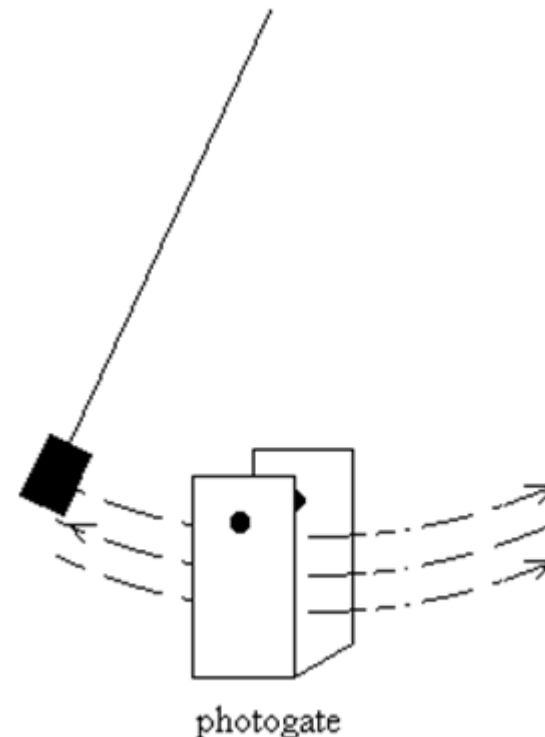
Iniciamos con el pin en '1',  
Cuando TC=MR0 generamos una interrupción y colocamos a '0' el pin  
Cuando TC=MR1 generamos otra interrupción y colocamos el pin en '1', además reseteamos el contador de modo que comenzamos a contar nuevamente.

## Registros de Captura

El valor del TC se almacena en los registros de captura cuando se genera un flanco en el pin de captura.

Ejemplo de aplicación medición de periodo de un péndulo simple, medición de “g”

Conectamos el detector a la pata de captura CAPn.x. En el momento en que el péndulo corta el haz se genera la captura (evitamos latencia), y disparamos una interrupción, en la interrupción almacenamos el valor que tenemos en el registro de captura en otra variable para liberar el registro de captura.



## Count Control Register

Permite configurar el Timer como TIMER propiamente dicho o como contador de eventos, por defecto se comporta como TIMER

## TCR (Timer Control Register)

Tiene solo 2 bits útiles, uno permite habilitar o deshabilitar la cuenta, el otro permite resetear el TC y PC.

## Interrupt Register

Permite identificar cuál es la fuente de interrupción MR0, MR1, MR2, MR3, CR0 (captura 0), CR1 (captura 1)

## Match Control Register

Permite configurar que pasará cuando  $TC=MRx$ , podemos generar una interrupción de Match, podemos volver el TC a cero, o podemos detener el TC

## External Match Register

Permite configurar si queremos manipular un pin de salida (MATn.x, donde n es el número de Timer y x el número de Match)

## Capture Control Register

Permite configurar si la captura será en un flanco de bajada, en un flanco de subida, o ambas en el pin de captura CAPn.x, además permite configurar que se genere una interrupción debido a la captura



## Programa simple

En este caso solo utilizamos el registro MATCH 0, se utilizará el TIMER0 como Timer (registro CTCR), además se cargará el MR0 con 25.000.000 dado que el PCLK =25Mhz, también se carga el registro PR=0, no es necesario porque así queda después un reset, significa que TC se incrementa en cada PCLK y se configura de modo que cuando TC=MR0 se genere una interrupción y simultáneamente se resetee el TC para comenzar nuevamente la cuenta para la próxima interrupción (MCR).

## Programa principal

Configuro CTCR (TIMER0 como Timer), innecesario pero lo aclaro

Cargo MR0 con 25.000.000

Cargo PR con 0 (innecesario)

configuro MCR para generar interrupción cuando TC=MR0, y resetear TC

Habilitamos interrupción en NVIC

Habilito TIMER para que comience a contar (Registro TCR)

Espero en un bucle infinito

## Programa simple

En la Rutina de interrupción solo bajamos la bandera que corresponda a MRO y ejecutamos el código. En caso de tener más fuentes de interrupción del TIMERO utilizar el registro TOIR  $|(1 \ll 0);$

## Programa Rutina Servicio de Interrupción

Bajamos La bandera de Interrupción por MRO (Registro IR)

Ejecuto código Rutina servicio de interrupción

## Actividad práctica

- 1) Implemente el código antes mencionado
- 2) Implemente un código tal que se generen interrupciones en Match0 y Match1, en Match0 encendemos un led, en Match1 apagamos el led y reseteamos el contador del timer.