



# Módulo 2. Entrada/salida

Temporizadores

# Timer



- ARM del laboratorio tiene 6 temporizadores
  - Generan una interrupción por las líneas 8 -13 del controlador de interrupciones
- Son contadores descendentes
  - Se inicializan a un valor
  - Cada ciclo interno se decrementan
  - Cada vez que llegan a cero generan una interrupción
- Pueden utilizarse también para generar una señal cuadrado
  - A partir de los valores de los registros TCNTn y TCMPn

# Gestión del Timer



## ■ Registros

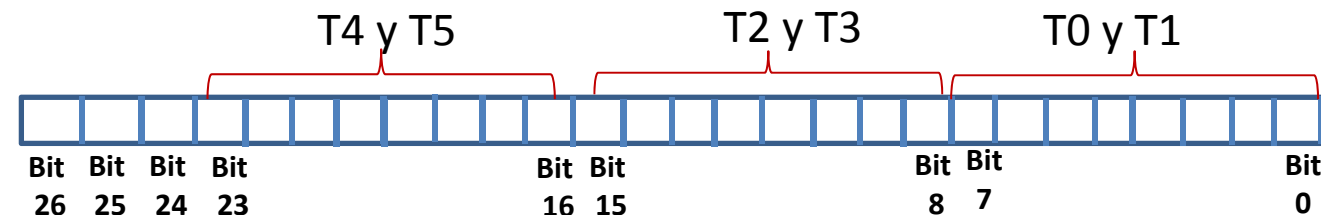
- **TCNTBn** (**registro de cuenta**,  $n=0-5$  depende del temporizador)
  - Se inicializa a un determinado valor y se decrementa en cada ciclo mientras el temporizador esté activo
- **TCMPBn** (**registro de comparación**,  $n=0-5$  depende del temporizador)
  - Se inicializa a otro valor
  - Se emplea para comparar con el registro de cuenta. El resultado de esta comparación se emplea para controlar la señal de salida
- **TCNTOn** (**registro de observación de cuenta**,  $n=0-5$  depende del temporizador)
  - Registro que permite consultar el valor actual del temporizador

# Gestión del Timer

## Registros

### – TCFG0 (registro 0 de configuración)

- Permite configurar el pre-escalado que se quiere aplicar
- El valor de pre-escalado está en el intervalo 0 - 255



Función	Bits	Descripción
Longitud de la zona muerta	[31:24]	Estos 8 bits determinan la zona muerta. La unidad de tiempo de la zona muerta es la misma que la del temporizador 0.
Pre-escalado 2	[23:16]	Estos ocho bits determinan el factor de pre-escalado de los temporizadores 4 y 5.
Pre-escalado 1	[15:8]	Estos ocho bits determinan el factor de pre-escalado de los temporizadores 2 y 3.
Pre-escalado 0	[7:0]	Estos ocho bits determinan el factor de pre-escalado de los temporizadores 0 y 1.

# Gestión del Timer

## Registros

### – TCFG1 (registro 1 de configuración)

- Permite configurar

- Cuál es el **temporizador que usará el DMA**
- Para cada temporizador permite **seleccionar** la salida del **divisor de frecuencia**

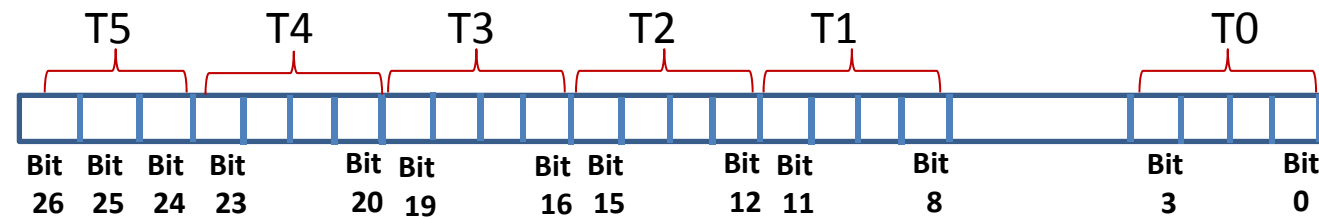
Función	Bits	Descripción
modo DMA	[27:24]	Selecciona el canal de DMA 0000 = No seleccionado      0001 = Temporizador0 0010 = Temporizador1      0011 = Temporizador2 0100 = Temporizador3      0101 = Temporizador4 0110 = Temporizador5      0111 = Reservado
MUX 5	[23:20]	Selecciona el MUX para el Timer5. 0000 = 1/2      0001 = 1/4      0010 = 1/8 0011 = 1/16      01xx = EXTCLK
MUX 4	[19:16]	Selecciona el MUX para el Timer4. 0000 = 1/2      0001 = 1/4      0010 = 1/8 0011 = 1/16      01xx = TCLK
MUX 3	[15:12]	Selecciona el MUX para el Timer3. 0000 = 1/2      0001 = 1/4      0010 = 1/8 0011 = 1/16      01xx = 1/32
MUX 2	[11:8]	Selecciona el MUX para el Timer2. 0000 = 1/2      0001 = 1/4      0010 = 1/8 0011 = 1/16      01xx = 1/32
MUX 1	[7:4]	Selecciona el MUX para el Timer1. 0000 = 1/2      0001 = 1/4      0010 = 1/8 0011 = 1/16      01xx = 1/32
MUX 0	[3:0]	Selecciona el MUX para el Timer0. 0000 = 1/2      0001 = 1/4      0010 = 1/8 0011 = 1/16      01xx = 1/32

# Gestión del Timer

## Registros

### – TCON (registro de control)

- Permite controlar el **comportamiento de los temporizadores** (start/stop, auto-reload, etc.)



Los 4 bits de cada temporizador tiene el siguiente significado

Ejemplo para el bit 4

Timer 4 auto reload on/off	[23]	Este bit determina el auto-reload para el Temporizador 4. 0 = One-shot      1 = Interval mode (auto reload)
Timer 4 output inverter on/off	[22]	Este bit determina el inversor de salida para el Temporizador 4. 0 = Inverter off      1 = Inverter on for TOUT4
Timer 4 manual update	[21]	Este bit determina el <i>manual update</i> del Temporizador 4. 0 = No operation      1 = Update TCNTB4, TCMPB4
Timer 4 start/stop	[20]	Este bit determina el <i>start/stop</i> del Temporizador 4. 0 = Stop      1 = Start for Timer 4

**NOTA: el temporizador 5 no tiene el bit output inverter on/off**

# Ejemplo: Configuración del modo un temporizador



- **Configurar el temporizador 4 en modo auto RELOAD**
  - Hay que **poner a 1 el cuarto bit** de los cuatro que corresponden al **temporizador 4**

## En ensamblador

```
.equ rTCON, 0x1D50008
```

```
ldr r0,=rTCON
ldr r1, [r0]
Or r1, r1, #(0x8 << 20)
str r1, [r0]
```

**0x8=1000**  
un 1 en el cuarto bit

**Desplaza 20 bits=(t+1)x4 para**  
empezar en los bits que  
corresponden al temporizador 4

## En C

```
#define rTCON (*(volatile unsigned *)0x1D50008)
```

Indica el temporizador, en el ejm  
es el temp 4. Para Temp 0 pos=0

↓  
int pos = (t+1)\*4;

```
rTCON |= ~(0x8 << pos);
```

# Configuración del temporizador



- **Configuración** del timer
  - se hace en la función **setup()** de **main.c** (Salvo que se quieran cambiar los valores de pre-escalado, divisor de frecuencia, ... en alguna parte del programa)
    - Poner el valor del pre-escalado que se quiere aplicar (registro **TCFG0**)
    - Seleccionar la salida del divisor de frecuencia (registro **TCFG1**)
    - Darle un valor a los registros **TCNTBn** y **TCMPBn**
    - En el registro **TCON**
      - Poner el bit **manual update** =“1” para que se carguen los registros TCNTBn y TCMPBn
      - Poner el bit **manual update** =“0”
      - Poner el bit **auto reload** =“1”, para que se inicialice el contador cada vez que llegue a 0
      - Poner el bit **start/stop** =“0” para parar el contador



# Configuración del temporizador



- Cada vez que se quiera **inicializar el temporizador** hay que poner:
  - El bit **manual update** =“1”
    - para que se carguen los registros TCNTBn y TCMPBn de nuevo por si se ha parado el temporizador en alguna parte del programa
  - El bit **manual update** =“0”
  - El bit **start/stop** =“1”