



Sistemas Electrónicos Digitales

Tema #3

6. Temporizadores



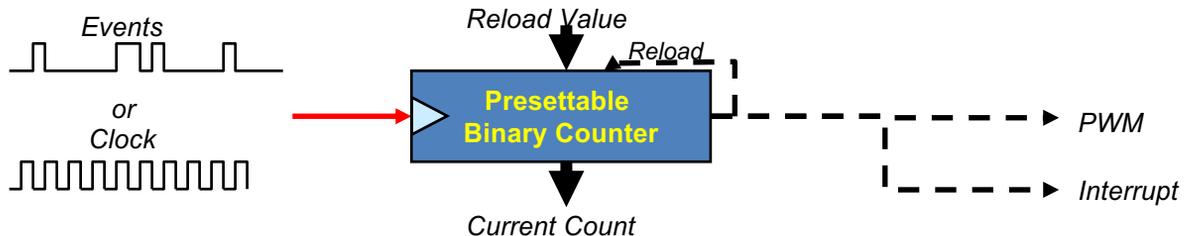
1. Introducción
2. GPIO: General Purpose Input/Output
3. Arquitectura Arm Cortex-M4
4. Interrupciones
5. C en ensamblador
6. Temporizadores (Timers)
7. Direct Memory Access
8. Comunicaciones Serie
9. Conversores A/D y D/A

Tipos de temporizadores

- **Temporizador de interrupción (Interrupt Timer)**
 - Genera interrupciones periódicas o transferencias DMA (acceso directo a la memoria)
- **Módulo PWM**
 - Conectado a los pines de E / S, soporta captura de entrada y comparación de salida
 - Genera señales PWM
 - Genera solicitudes de interrupción
- **Temporizador de bajo consumo (Low-Power Timer)**
 - Puede funcionar como temporizador o contador en todos los modos de alimentación
 - Puede despertar el sistema con una interrupción
 - Puede activar hardware
- **Reloj en tiempo real**
 - Accionado por un cristal externo
 - Cuenta el tiempo transcurrido (segundos) en un registro
 - Puede establecer alarmas
 - Puede generar señal de salida de 1Hz y/o interrupción
 - Puede despertar el sistema con una interrupción
- **SYSTICK**
 - Parte de los periféricos del núcleo de la CPU
 - Puede generar interrupciones periódicas.

Temporizador / Contador

- Periférico común para microcontroladores.
- Basado en contador binario preconfigurado, pero configurable
 - El valor del contador puede ser leído y escrito por MCU
 - La cuenta puede ser hacia arriba o hacia abajo
 - Se puede seleccionar la fuente de reloj del contador
 - **Modo contador:** cuenta pulsos que indican eventos (por ejemplo, pulsos de odómetro)
 - **Modo de temporizador:** la fuente del reloj es periódica, por lo que el valor del contador es proporcional al tiempo transcurrido (por ejemplo, cronómetro)
 - Se puede seleccionar la acción si se produce desbordamiento (arriba o abajo) del contador.
 - Generar **interrupción**
 - **Recargar** el contador con valor especial y continuar contando.
 - **Cambiar señal de salida** de hardware

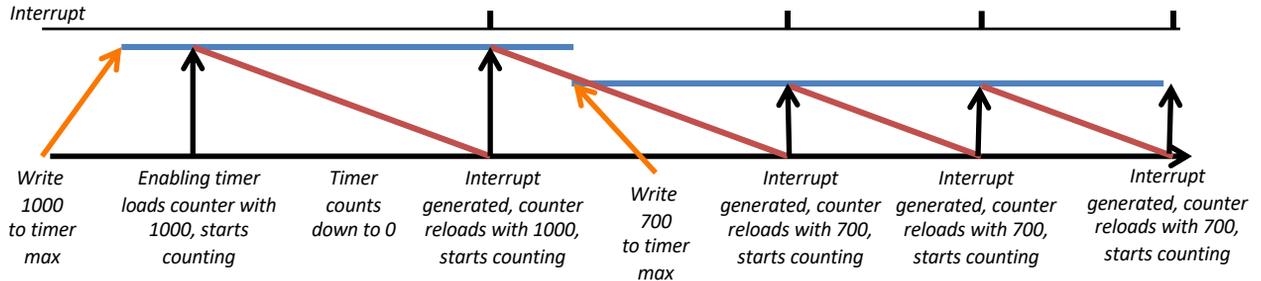




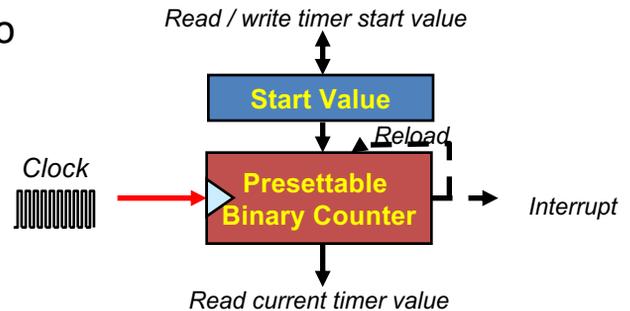
Temporizadores

TEMPORIZADOR DE INTERRUPCIÓN

Interrupt Timer



- Cargar valor de inicio desde registro
- Contador cuenta atrás con cada pulso de reloj
- Cuando el valor del temporizador llega a cero
 - Genera interrupción
 - Recarga el temporizador con valor de inicio.





Cálculo del valor máximo

- Objetivo: generar una interrupción cada T segundos
- Valor máximo = $\text{round}(T * \text{Freq})$
 - Redondear puesto que el registro es un número entero, no un número real
 - El redondeo proporciona el número entero más cercano al valor deseado, lo que resulta en un error de tiempo mínimo
- Ejemplo: interrupción cada 137.41ms, suponiendo $f_{\text{reloj}} = 24 \text{ MHz}$
 - $137.41 \text{ ms} * 24 \text{ MHz} = 3297840$
- Ejemplo: interrupción con $f = 91 \text{ Hz}$ con un reloj de 12MHz
 - $(1/91 \text{ Hz}) * 12 \text{ MHz} = \text{round}(131868.1318) = 131868$
- Macros: interrupción 1000 veces por segundo
 - $\text{CLK_FREQ} / 1000$



Temporizadores

Timer / PWM Module

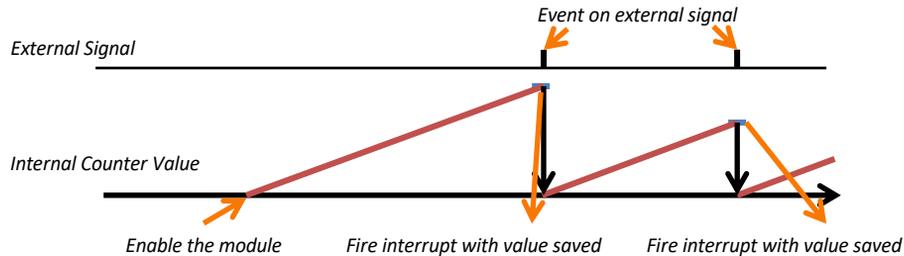
Temporizador / Módulo PWM

- Contador principal (core counter)
 - Opciones de reloj: externo o interno
 - Prescaler para dividir reloj
 - Puede volver a cargar con el valor establecido, o desbordar y dar la vuelta
- N canales
 - 3 modos:
 - **Captura:** captura el valor del temporizador cuando cambia la señal de entrada
 - **Comparación de salida:** cambia una señal de salida cuando el temporizador alcanza cierto valor
 - **PWM:** Genera señal modulada por ancho de pulso (pulse-width-modulation). El ancho del pulso es proporcional al valor especificado.
 - Posible activación de interrupción, activación de hardware en caso de desbordamiento
 - Un pin de E / S por canal

Modos de canal principales

- **Modo de captura de entrada**
 - Captura el valor del temporizador cuando cambia la señal de entrada.
 - Flanco de subida, flanco de bajada, ambos
 - ¿Cuánto tiempo después de que empezó el temporizador cambió la entrada?
 - Medir el retraso de tiempo
- **Modo de comparación de salida**
 - Modificar la señal de salida cuando el temporizador alcanza el valor especificado
 - Set, clear, pulse, toggle (invert)
 - Hacer un pulso de ancho especificado
 - Hacer un pulso después de un retraso especificado
- **PWM (Modulación de ancho de pulso)**
 - Hacer una serie de pulsos de ancho y frecuencia especificados.

Input Capture Mode



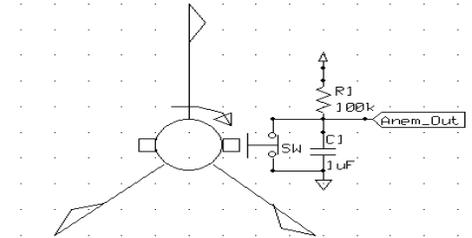
- Los pines de E/S actúan como detectores de flanco
- Cuando se detecta un flanco válido en el pin ...
 - Se almacena el valor actual del contador.
 - Se llama a la interrupción

Indicador de velocidad del viento (anemómetro)

- La velocidad de rotación (y la frecuencia del pulso) es proporcional a la velocidad del viento
- Dos opciones de medición:
 - Frecuencia (mejor para altas velocidades)
 - Ancho (mejor para bajas velocidades)
- Puede calcular la velocidad del viento:

$$v_{wind} = \frac{K * f_{clk}}{T_{anemometer}}$$

- ¿Cómo podemos usar el temporizador para esto?
 - Utilice el modo de captura de entrada para medir el período de la señal de entrada

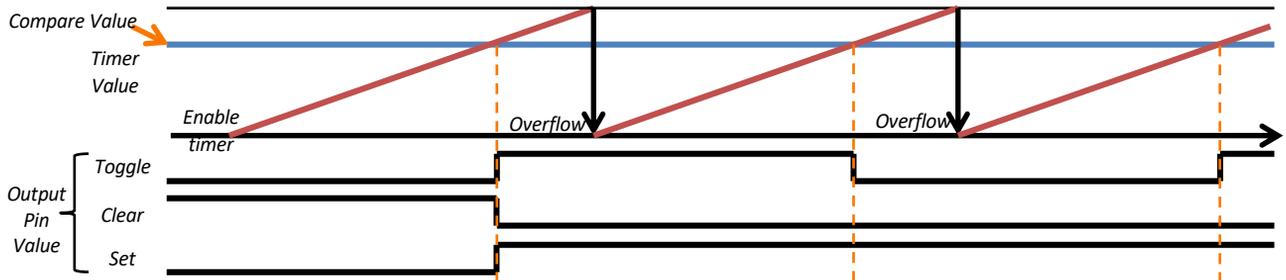




Modo de captura para el anemómetro

- Configuración
 - Configurar el módulo para contar a una velocidad dada desde el reloj interno
 - Configurar canal para captura de entrada en flanco de subida
- Repetir:
 - Primera interrupción - en el flanco de subida
 - Reconfigurar el canal para la captura de entrada en el flanco descendente
 - Borrar contador, empezar a contar
 - Segunda interrupción - en el flanco de bajada
 - Leer el valor de captura y guardarlo para el cálculo de la velocidad del viento
 - Reconfigurar el canal para la captura de entrada en el flanco ascendente
 - Borrar contador, empezar a contar

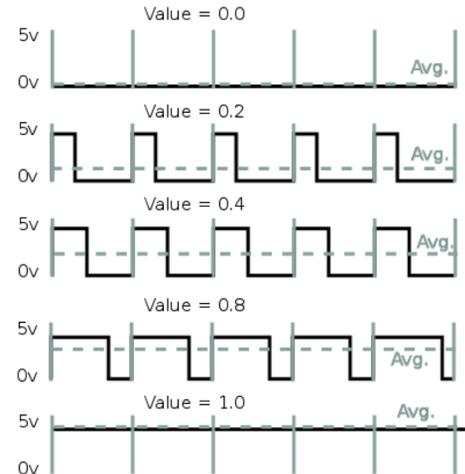
Output Compare Mode



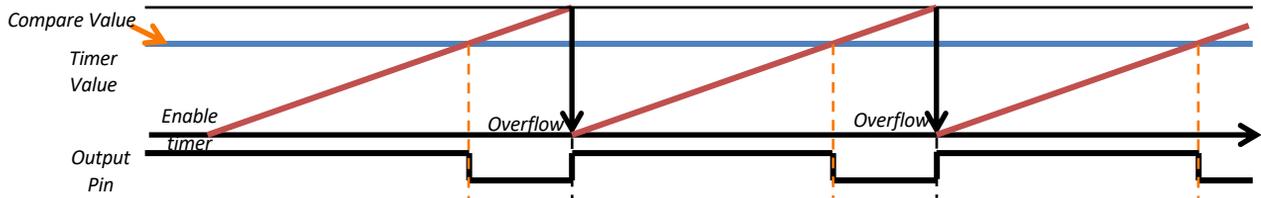
- Acción cuando se alcanza el valor
 - Toggle
 - Clear
 - Set
- Cuando el contador alcanza el valor ...
 - Se genera señal de salida.
 - Se llama la interrupción (si está habilitada)

PWM: Pulse-Width Modulation

- Usos de PWM
 - Los amplificadores de potencia digitales son más eficientes y menos costosos que los amplificadores de potencia analógicos
 - Aplicaciones: control de velocidad del motor, atenuador de luz, conversión de potencia en modo conmutado
 - La carga (motor, luz, etc.) responde lentamente, promedia la señal PWM
 - La comunicación digital es menos sensible al ruido que la analógica.
 - **PWM proporciona una codificación digital de un valor analógico**
 - Mucho menos vulnerable al ruido.
- Características de la señal PWM
 - **Frecuencia de modulación:** cuántos pulsos se producen por segundo (fijo) **TIMx_ARR**
 - **Periodo:** $1 /$ (frecuencia de modulación)
 - **On-time:** cantidad de tiempo que cada pulso está activado (afirmado) **TIMx_CCRx**
 - **Ciclo de trabajo (duty cycle):** on-time/periodo
 - **Ajuste el on-time (por lo tanto, el ciclo de trabajo) para representar el valor analógico**



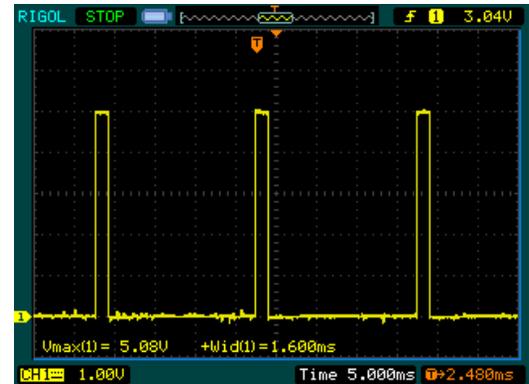
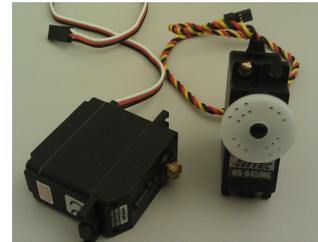
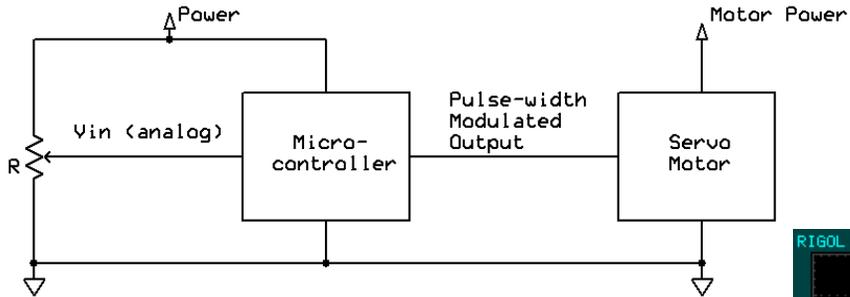
PWM Mode



- Ciclo de trabajo de PWM proporcional al valor (de comparación)
 - Período = valor máximo del temporizador
 - Ancho de pulso = valor (de comparación)

$$Duty\ Cycle = \frac{Compare\ Value}{Max\ Value} \cdot 100\%$$

PWM para controlar un servomotor



- Señal PWM para servo
 - Período: 20 ms period
 - Ancho de pulso: 1 a 2 ms



SysTick

- El procesador tiene un temporizador de sistema de 24 bits, **SysTick**, que realiza una cuenta regresiva desde el valor de recarga a cero, se vuelve a cargar, es decir, carga el valor en el registro SYST_RVR en el siguiente flanco del reloj, y luego cuenta hacia atrás en los flancos de reloj posteriores.

¿Cómo se utilizan los temporizadores?

- **Retraso preciso (i.e. 100)**
 - Lea el temporizador, almacene el valor como K. Haga un bucle hasta que el temporizador alcance $K + 100$.
- **Programar eventos importantes**
 - Configure una comparación de salida para activar una interrupción en un momento preciso
- **Medir el tiempo entre eventos**
 - Cuando ocurre el evento # 1, almacene el valor del temporizador como K
 - Cuando ocurre el evento # 2, lea el valor del temporizador y reste K
 - La diferencia es el tiempo transcurrido entre los dos eventos.

Ejemplo PWM

TIM3
 Slave Mode: Disable
 Trigger Source: Disable
 Clock Source: Disable
 Channel1: PWM Generation CH1
 Channel2: PWM Generation CH2
 Channel3: PWM Generation CH3
 Channel4: PWM Generation CH4
 Combined Channels: Disable
 Use ETR as Clearing Source
 XOR activation
 One Pulse Mode
TIM4

TIM4 Configuration
 Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings
 Configure the below parameters :
 Search : Search (Ctrl+F)
 Counter Settings
 Prescaler (PSC - 16 bits value) : 16
 Counter Mode : Up
 Counter Period (AutoReload Register - 16 bits va... : 100
 Internal Clock Division (CKD) : No Division
 Trigger Output (TRGO) Parameters
 Master/Slave Mode (MSM bit) : Disable (Trigger input effect not delayed)
 Trigger Event Selection : Reset (UG bit from TIMx_EGR)
 PWM Generation Channel 1
 Mode : PWM mode 1
 Pulse (16 bits value) : 50
 Fast Mode : Disable
 CH Polarity : High

```
HAL_TIM_PWM_Start
    (&tim4, TIM_CHANNEL_1);
```

```
while (1)
{
    __HAL_TIM_SET_COMPARE (&tim4, TIM_CHANNEL_1, 100);
}
```



Ejemplo: Cronómetro

- Medir el tiempo con 100us de resolución.
- Muestra el tiempo transcurrido, actualizando la pantalla cada 10 ms.
- Controles
 - S1: alternar inicio / parada
- Usar temporizador de interrupción
 - Contador se incrementa cada 100us
 - Ajustado al temporizador para que expire cada 100us.
 - Calcular el valor máximo, por ejemplo, a 24 MHz = ronda $(100 \text{ us} * 24 \text{ MHz} - 1) = 2399$
 - Actualización de LCD cada 10 ms
 - Actualizar LCD cada “n” interrupciones ISR
 - $n = 10\text{ms}/100\text{us} = 100$
 - ¡No actualices LCD en ISR! Demasiado lento.
 - En su lugar, establezca la bandera en ISR, compruébela (poll) en el bucle principal