

Tema 12

RESUMEN

En el **tema 12** (*Diseño de Soluciones y Funciones Especiales: RTC, Watchdog y Consumo*) se enseña cómo diseñar un proyecto real y algunas funciones especiales interesantes del microcontrolador, dividido en varios puntos y acompañado por sus vídeos explicativos.

Se empieza explicando algunas funciones especiales del microcontrolador. La primera de ellas es la funcionalidad de reloj en tiempo real (RTC). Primero sus características, después su configuración en la perspectiva CubeMX y las funciones HAL para la programación, y finalmente un ejemplo con HAL y CubeMX para probar por el alumno, con su programa correspondiente (inicialización, y funcionamiento continuo). El ejemplo muestra la fecha y hora del RTC en el LCD cada segundo, partiendo de una fecha/hora inicial configurada.

Después del ejemplo se muestra, a título ilustrativo, cómo configurar el RTC con registros. Primeros cómo es la configuración y el modo de trabajar, después los registros de configuración, datos y estado y finalmente el mismo ejemplo que antes pero ahora trabajando con registros.

Los diferentes **registros de control** del bloque RTC, que sirven para configurar su funcionalidad antes de utilizarlo, son los siguientes:

- **RTC→CR** – Control Register, que sólo se va a utilizar el bit 6 (FMT) para indicar si es formato 24h (con un '0') o si es formato 12h (con un '1'). El resto de bits se deja a '0'
- **RTC→PRER** – Prescaler Register, que configura el preescalado asíncrono (PREDIV_A) y el síncrono (PREDIV-S)
- **RTC→WPR** – Write Protection Register, que se utiliza para proteger la configuración

Los dos **registros de datos** del bloque RTC, que sirven para leer/escribir la fecha/hora del bloque, son los siguientes:

- **RTC→TR** – Time Register, que contiene el valor de la hora, en la siguiente estructura (formato BCD):
 - PM: Con un 0 es formato de 24h o AM, y con un 1 es PM
 - HT: Decenas de hora
 - HU: Unidades de hora

- MINT: Decenas de minuto
- MINU: Unidades de minuto
- ST: Decenas de segundo
- SU: Unidades de segundo
- **RTC→DR** – Date Register, que contiene el valor de la fecha, en la siguiente estructura (formato BCD):
 - YT: Decenas de año
 - YU: Unidades de año
 - WDU: Tres bits que indican el día de la semana (000-prohibido; 001-lunes; ...; 111-domingo)
 - MT: Decenas de mes
 - MU: Unidades de mes
 - DT: Decenas de día
 - DU: Unidades de día

El único **registro de estado** del bloque RTC, que sirve para comprobar y consultar la ejecución del RTC, es el siguiente:

- **RTC→ISR** – Initialization and Status Register, que es un registro de 32 bits de lo que sólo interesan los siguientes para el curso:
 - INIT, bit de control:
 - 0: se quita el modo de inicialización, actualizando los nuevos valores y arrancando el RTC
 - 1: se pone el RTC en modo inicialización
 - INITF, bit de estado:
 - 0: el modo de inicialización está desactivado.
 - 1: el modo de inicialización está activado y ya se puede escribir en los registros.

Una vez visto el RTC, se explica la funcionalidad de Watchdog, que es un mecanismo interno de control del microcontrolador para provocar el reset del sistema en caso de que se detecte que el micro ha perdido el control.

Posteriormente se describe la funcionalidad de bajo consumo para el microcontrolador y sus posibilidades.

Una vez vistas esas tres funcionalidades especiales, se habla de los pasos a seguir a la hora de diseñar un proyecto completo y qué puntos hay que tener en cuenta en cuanto a temporización, recursos comunes y desarrollo.

En el último punto se muestra la utilización de las APIs en la programación. Primero los tipos que hay (HAL, Middleware y Language based library) y luego algunos ejemplos para STM32, C y C++ Library en Keil uVision para ARM y finalmente la Standard Peripheral Library de STM32L1, completando lo visto en el tema 9.