

Tema 13: Funciones Especiales y Desarrollo de Proyectos

Sistemas Digitales Basados en Microprocesadores

Universidad Carlos III de Madrid

Dpto. Tecnología Electrónica

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Índice

- Características Especiales
 - Reloj en Tiempo Real
 - Watchdog
 - Bajo Consumo
- Uso de APIs en Programación
- Integración de Periféricos en un Proyecto

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Características Especiales: Reloj en Tiempo Real (RTC)

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

 <http://dte.uc3m.es>

Cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al

Artículo 17 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002.

(Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.)

(C) Raúl Sánchez Reíllo

Características y Funcionamiento

- El Reloj en Tiempo Real (RTC) es un dispositivo que permite medir el tiempo para mantener un calendario y un reloj
 - Evitando el uso de un temporizador para esta función.
- El del STM32L152 ha sido diseñado para un consumo mínimo, pensado para sistemas alimentados por batería
 - Proporciona segundos, minutos, horas, día del mes, mes, año y día de la semana
 - Posee un divisor de reloj para que se ajuste a distintas frecuencias de oscilador
 - Aunque por defecto está configurado para ser usado a través del LSE
 - No tiene ningún mecanismo para mantener la hora tras una pérdida de energía
 - Tiene un mecanismo de protección para evitar modificaciones no voluntarias de los registros de control

Cartagena99

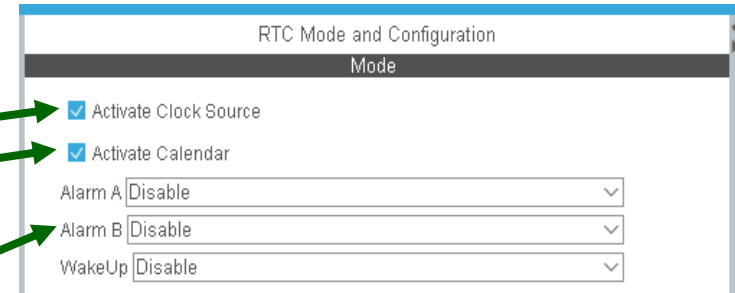
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Guía de Funcionamiento con HAL

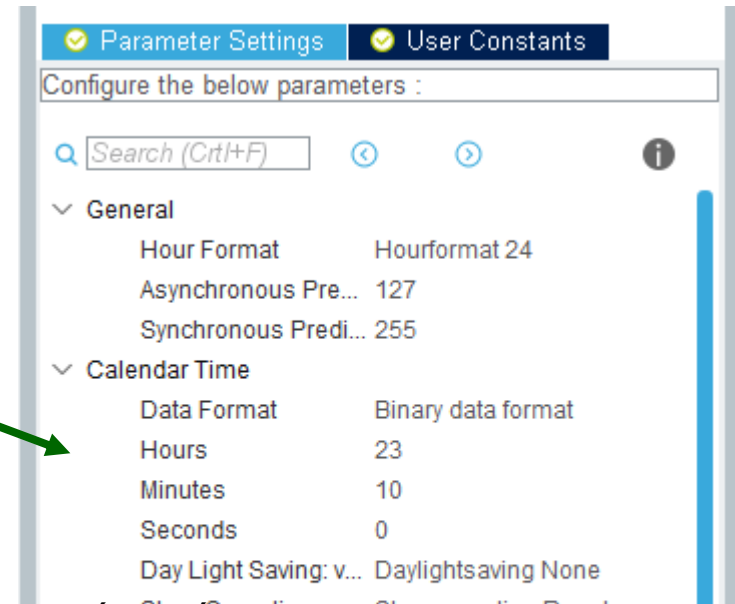
- En CubeMX:

- Activa la fuente de reloj
- Activa el Calendario
- Si se selecciona una alarma, se da la opción de activar la interrupción. Para el ejemplo se deja "Disable"
- Ahora configura los parámetros deseados. Elige los que se muestran en la imagen para el ejemplo con los valores iniciales de la hora y de la fecha



- En Programación:

- Inicialización:
 - HAL_RTC_Init()
 - Se necesita crear una variable de tipo RTC_TimeTypeDef y otra



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Guía de Funcionamiento con HAL

- En Programación (continuación):
 - Fecha y Hora:
 - Parámetros: h – handler; d – datos; f – formato (RTC_FORMAT_BIN, RTC_FORMAT_BCD)
 - HAL_RTC_SetTime(h, d, f)
 - HAL_RTC_SetDate(h, d, f)
 - HAL_RTC_GetTime(h, d, f)
 - HAL_RTC_GetDate(h, d, f)
 - Alarma:
 - Se necesita crear una variable tipo RTC_AlarmTypeDef para tener la estructura con la información de la alarma
 - Parámetros: h – handler; a – alarma; f – formato (RTC_FORMAT_BIN, RTC_FORMAT_BCD); w – timeout
 - HAL_RTC_SetAlarm(h, a, f)
 - HAL_RTC_SetAlarm_IT(h, a, f)
 - HAL_RTC_GetAlarm(h, a, f)

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Ejemplo: Configuración y Uso RTC con HAL

- Se muestra la fecha (día y mes) y la hora (hora y minutos) en el LCD alternativamente cada segundo, empezando a las 23:10 del 24/03 (domingo)
 - Inicialización:

```
// Estructuras necesarias para el RTC
RTC_TimeTypeDef my_time;
RTC_DateTypeDef my_date;

// Variables necesarias para el programa
uint8_t text[6];
uint16_t number;
```

```
// Configuración inicial de la
// fecha y hora del RTC

my_date.Month = 3;
my_date.Date = 24;
my_date.WeekDay = RTC_WEEKDAY_SUNDAY;
my_date.Year = 19;
my_time.DayLightSaving = RTC_DAYLIGHTSAVING_NONE;
my_time.Hours = 23;
my_time.Minutes = 10;
my_time.Seconds = 0;
my_time.TimeFormat = RTC_HOURFORMAT_24;

HAL_RTC_SetDate(&hrtc, &my_date, RTC_FORMAT_BIN);
HAL_RTC_SetTime(&hrtc, &my_time, RTC_FORMAT_BIN);

// Funciones de inicialización del LCD
BSP_LCD_GLASS_Init();
BSP_LCD_GLASS_BarLevelConfig(0);
```

Cartagena99

CLASES PARTICULARES, TUTORIAS TECNICAS ONLINE
LLAMA O ENVIA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Ejemplo: Configuración y Uso RTC con HAL

- Funcionalidad continua:

```
while (1) {

    // Cojo la hora y la manipulo para sacarla por el LCD en 4 dígitos: 00HHMM
    HAL_RTC_GetTime(&hrtc, &my_time, RTC_FORMAT_BIN);
    number = 0;
    number = (my_time.Hours * 100) + (my_time.Minutes);

    // Saco la hora por el LCD y lo muestro 1 segundo
    Bin2Ascii((unsigned short)number, text);
    BSP_LCD_GLASS_DisplayString((uint8_t *)text);
    HAL_Delay(1000);

    // Cojo la fecha y la manipulo para sacarla por el LCD en 4 dígitos: 00DDMM
    HAL_RTC_GetDate(&hrtc, &my_date, RTC_FORMAT_BIN);
    number = 0;
    number = (my_date.Date * 100) + (my_date.Month);

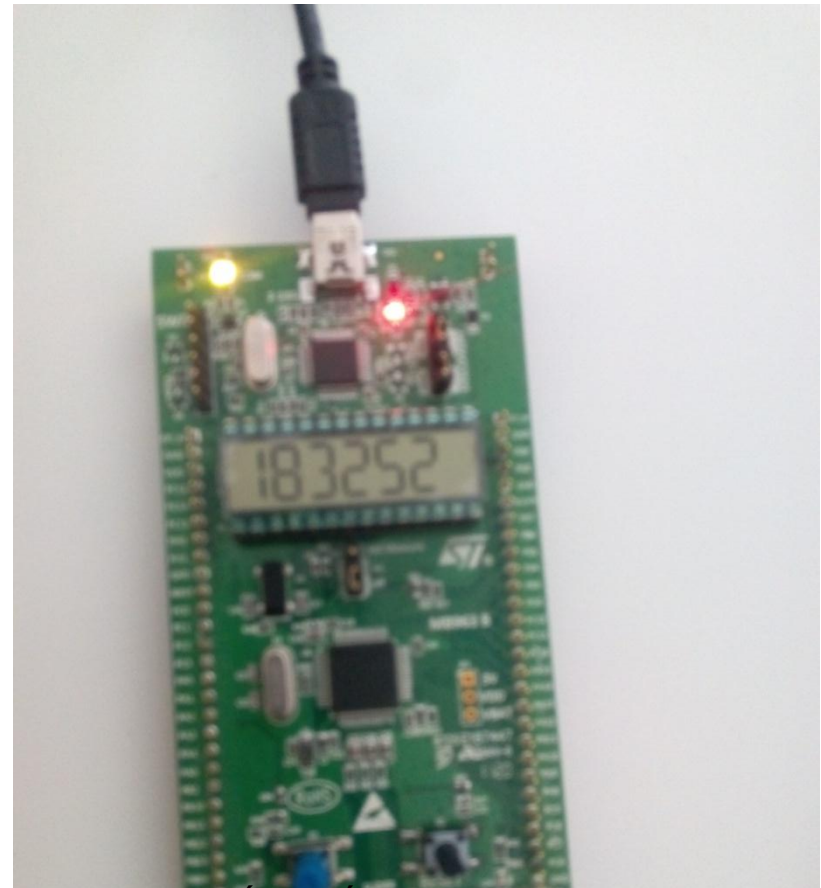
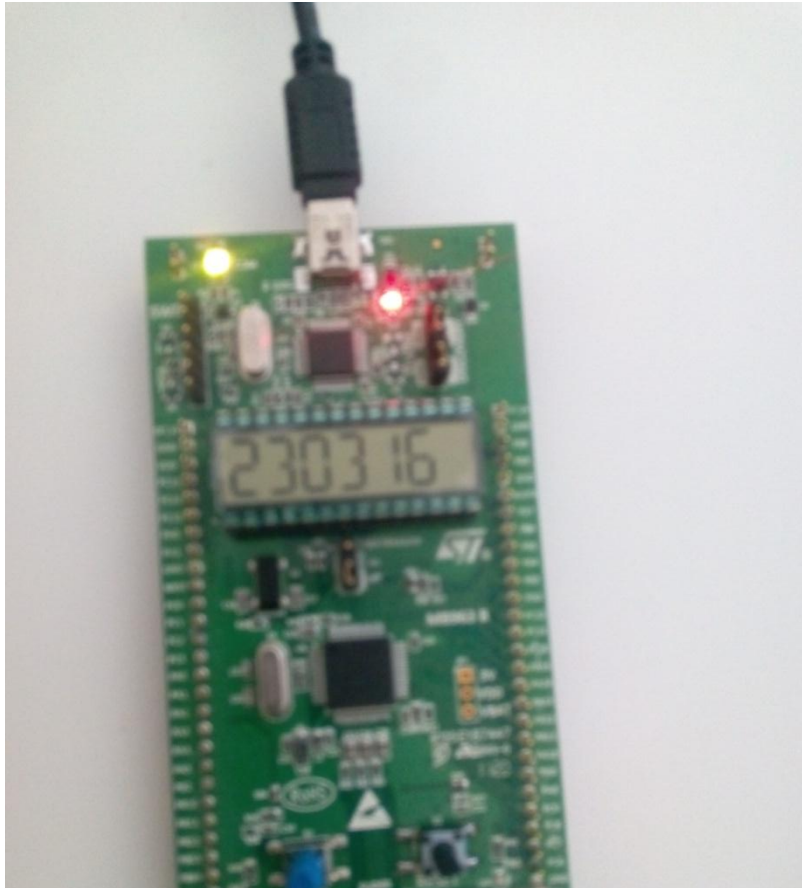
    // Saco la fecha por el LCD y lo muestro 1 segundo
    Bin2Ascii((unsigned short)number, text);
    BSP_LCD_GLASS_DisplayString((uint8_t *)text);
    HAL_Delay(1000);
}
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Ejemplo: Configuración y Uso RTC con HAL



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Características Especiales: Watchdog

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Características Generales

- Un Watchdog 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
- El sistema de Watchdog tiene las siguientes características:
 - Resetea el chip internamente si no se actualiza periódicamente
 - Es decir, el programa debe, de forma periódica y antes de que pase un determinado tiempo límite, escribir en un registro del watchdog una secuencia de valores, de tal forma que le indica al watchdog que sigue teniendo el control del sistema
 - Se habilita por software, pero sólo se resetea por reset o por una interrupción de Watchdog
 - Un uso incorrecto o incompleto, provoca también el reset
 - Utiliza internamente un temporizador

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Características Especiales: Bajo Consumo

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Bajo Consumo

- Al crear un sistema electrónico, uno de los requisitos principales es que su consumo sea mínimo:
 - Por motivos de calificación energética
 - Cuando el sistema es portátil y ha de ser alimentado con baterías
- Todos los microcontroladores actuales tienen la posibilidad de definir distintos modos de bajo consumo:
 - Totalmente operativo (máximo consumo)
 - Determinados periféricos desconectados
 - Todos los periféricos desconectados (salvo algún PIN)
 - La mayor parte de la CPU desconectada (mínimo consumo)
- El ubicar al dispositivo en uno de los modos de bajo consumo, lo realiza el programador de la aplicación cuando se cumplen determinados requisitos
- El “despertar” al micro de un estado de bajo consumo se realiza mediante una

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Integración de Periféricos en un Proyecto

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Pasos para el Diseño de un Proyecto

1. Analizar con detalle el problema a resolver
2. Localizar funcionalidades necesarias por parte del micro
 1. Realizar el **Diagrama de Bloques**, seleccionando los recursos (periféricos, pines...) a utilizar
3. Definir la funcionalidad de cada uno de los periféricos del micro involucrados
4. Determinar la prioridad de cada uno de ellos y realizar un análisis temporal de su petición de uso por parte del micro
 1. **Determinar las IRQ a usar**, y las prioridades entre ellas
 2. Definir las RAI correspondientes
5. Diseñar la solución completa utilizando **diagramas de flujo**

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Puntos a tener muy en cuenta

- Temporización:
 - Muchas veces hay necesidades de temporización que no están específicamente planteadas en el problema a resolver:
 - Suelen tener que ver con usabilidad (interacción con el usuario final), o con compartición del tiempo.
- Recursos comunes:
 - No sólo los pines, sino también otros periféricos
 - Especialmente los temporizadores
- Desarrollo:
 - Se recomienda siempre utilizar una estrategia de Divide y Vencerás. Dos posibilidades:
 - **Aproximación Top-Down:**
 - Se hace el programa principal con las llamadas a las distintas funciones, implementando estas como simples funciones vacías.
 - Se implementa cada una de las funciones por separado
 - **Aproximación Bottom-Up:**
 - Se desarrollan las funciones individuales
 - Posteriormente se implementa el programa principal uniendo las funciones individuales

Cartagena99

CLASES PARTICULARES, TUTORIAS TECNICAS ONLINE
LLAMA O ENVIA WHATSAPP: 689 45 44 70


ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Uso de APIs en Programación

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

 <http://dte.uc3m.es>

Cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al

Artículo 17 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002.

Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

(C) Raúl Sánchez Reíllo

Uso de APIs en Programación

- Cuando la CPU o los periféricos son excesivamente complicados, o simplemente la aplicación a desarrollar requiere de programación avanzada, suele ser necesario utilizar **Application Programming Interfaces (APIs)** para simplificar nuestra tarea
- También pueden ser conocidas como **Libraries** (traducido como Biblioteca o como Librería)
- Las APIs pueden ser de muy distinto tipo, e incluso tener nombres distintos al de API dependiendo del sector.
 - Por ejemplo, en el caso de microcontroladores, se habla de:
 - HAL – Hardware Abstraction Layer: para hablar de aquellas funciones definidas que simplifiquen la configuración y el acceso a los periféricos
 - También llamada Standard Peripheral Library
 - Muchas veces suele ser proporcionado por el mismo fabricante, pero a veces hay algunas de terceras partes
 - Middleware: que se refiere a aquellas APIs que intentan dar soporte a una funcionalidad compleja que todavía no se encuentra a nivel de aplicación final (por ejemplo, la pila de TCP/IP, el interfaz completo de USB, o un sistema de ficheros)
 - Pueden haber varias disponibles, procedentes de terceras partes

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

<http://dte.uc3m.es>

Cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al

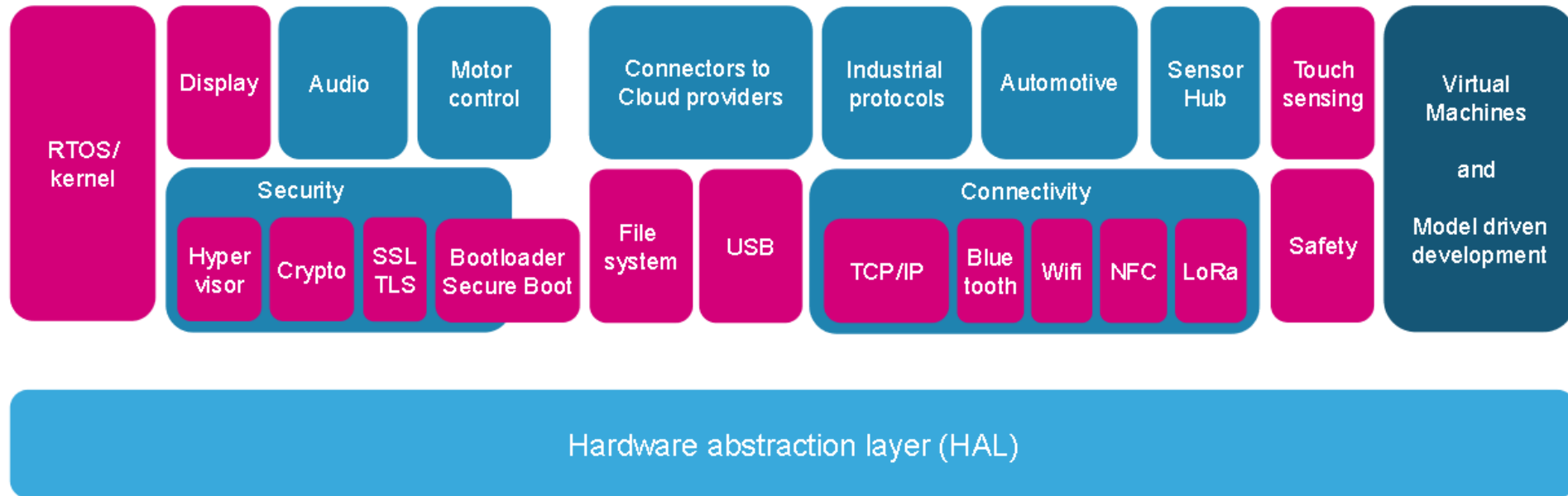
Artículo 17 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002,

si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

(C) Raúl Sánchez Reíllo

Uso de APIs en Programación

Middleware / Application fields



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Uso de APIs en Programación

STM32 – Crypto



Provider	Solution name	Model	Cost	Availability									
				F0	F1	F2	F3	F4	F7	H7	L0	L1	L4
Cypherbridge	uVPN SDK IKE v1/IKE v2/IPsec	Source	License	N	N	N	N	Y	Y	N ³	N	N	N
HCC	Verifiable Encryption manager AES, 3DES, DSS, EDH, MD5, RSA, SHA1, SHA256	Source	License	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Rowebots	UNISON SSL/TLS Stack AES, Blowfish, Triple-DES (3DES), DES, ARC4, Camellia, XTEA ECB, CBC, CFB, CTR, GCM, CCM MD2, MD4, MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, RIPMD-160 ECC	Source	License	N	Y	Y	Y	Y	Y	Y	N	Y	Y
SEGGER	emSecure signatures	Source	License	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SEGGER	emLib AES and emLib DES	Source	License	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SEGGER	emFile encryption	Source	License	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ST	STM32 Cryptographic library ^{1,2} AES, DES, 3DES, ARC4, MD5, SHA1, SHA2, RSAsig, ECC Key gen, ECDSA, ...	Binaries	Free	N	Y	Y	Y	Y	N	N	N	Y	N
ST	X-CUBE-CRYPTOLIB	Binaries	Free	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
wolfSSL	wolfCrypt ¹ , part of wolfSSL MD2, MD4, MD5, SHA-1, SHA-256, SHA-384, SHA-512, BLAKE2b, RIPMD-160, Poly1305 AES (CBC, CTR, GCM, CCM), Camellia, DES, 3DES, ARC4, RABBIT, HC- 128, ChaCha20	Open source (GPL2) or Source	Free or license	N	N	Y	N	Y	Y	Y	Y	Y	Y

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Uso de APIs en Programación



STM32 – USB solutions (1/2)

Provider	Solution name	Model	Cost	Availability										
				F0	F1	F2	F3	F4	F7	H7	L0	L1	L4	
Chibios	ChibiOS/HAL	Open source (GPL3) or Source	Free or License	Y	Y	Y	Y	Y	Y	Y	N ¹	Y	Y	Y
CMX	CMX-USB Device, Host	Source	License	Y	Y	Y	Y	Y	Y	Y	N ¹	Y	Y	N
eCosCentric	eCosPro-Host, Device	Source	License	N	Y	Y	Y	Y	Y	Y	N ¹	N	Y	Y
EUROS	USB Host & Device	Binaries	License	N	Y	Y	Y	Y	Y	Y	N ¹	N	Y	Y
EmCraft	Linux USB Host	Open source (GPL)	Free	N	N	Y	N	Y	N	N	N ¹	N	N	N
Express Logic	USBX	Source	License	Y	Y	Y	Y	Y	Y	Y	N ¹	Y	Y	Y
HCC	HCC-USB Host, Device	Source	License	Y	Y	Y	Y	Y	Y	Y	N ¹	Y	Y	Y
Wittenstein - High Integrity Systems	CONNECT USB Device, USB Host	Source	License	Y	Y	Y	Y	Y	Y	Y	N ¹	Y	Y	Y
Keil / arm	MDK-ARM USB	Source	License	Y	Y	Y	Y	Y	Y	Y	N ¹	Y	Y	Y
Mentor Embedded	Nucleus USB	Source	License	N	Y	Y	Y	Y	Y	Y	N ¹	N	Y	Y

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Uso de APIs en Programación



STM32 – USB solutions (2/2)

Provider	Solution name	Model	Cost	Availability											
				F0	F1		F2	F3	F4	F7	H7	L0	L1	L4	
					Others	F105 F107									
Micro Digital	smxUSB	Source	License	Y	Y	Y	Y	Y	Y	Y	Y	N ¹	Y	Y	Y
Quadros	RTXCusb	Source	License	N ¹	Y	Y	Y	Y	N ¹	N ¹	N ¹	N ¹	N ¹	N ¹	N ¹
Rowebots	Unison USB System	Source	License	N	Y	Y	Y	Y	Y	N	N ¹	Y	N	Y	
SEgger	emUSB Device, emUSB Host	Source	License	Y	Y	Y	Y	Y	Y	Y	N ¹	Y	Y	Y	
ST	USB FS device library	Source	Free	<u>Y</u>	<u>Y</u>	N	N	<u>Y</u>	N	N	N ¹	N	<u>Y</u>	N	
ST	USB FS&HS Host&Device lib	Source	Free	N	N	<u>Y</u>	<u>Y</u>	N	<u>Y</u>	N	N ¹	N	N	N	
ST	STM32Cube – USB Host&Device	Source	Free	Y ²	Y ²		Y	Y ²	Y	Y	N ¹	Y ²	Y ²	Y ²	
Thesycon	Embedded USB Device	Source	License	N ¹	N ¹		Y	N ¹	Y	Y	N ¹	N ¹	N ¹	N ¹	
Zephyr	USB device stack	Source	Free ³	Y	Y		N	Y	Y	N	N	N	N	Y	

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

C y C++ Library en Keil uVision para ARM

Summary of the C and C++ runtime libraries

1.2.1 Summary of the C and C++ runtime libraries

A summary of the C and C++ runtime libraries provided by ARM®.

C standardlib
This is a C library consisting of:

- All functions defined by the ISO C99 library standard.
- Target-dependent functions that implement the C library functions in the semihosted execution environment. You can redefine these functions in your own application.
- Functions called implicitly by the compiler.
- ARM extensions that are not defined by the ISO C library standard, but are included in the library.

C microlib
This is a C library that can be used as an alternative to C standardlib. It is a micro-library that is ideally suited for deeply embedded applications that have to fit within small-sized memory. The C micro-library, microlib, consists of:

- Functions that are highly optimized to achieve the minimum code size.
- Functions that are not compliant with the ISO C library standard.
- Functions that are not compliant with the 1985 IEEE 754 standard for binary floating-point arithmetic.

C++
This is a C++ library that can be used with C standardlib. It consists of:

- Functions defined by the ISO C++ library standard.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

C y C++ Library en Keil uVision para ARM

The screenshot shows the ARM Development Tools interface. The left pane displays a tree view of the documentation, with 'C++ and C libraries and the std namespace' selected. The right pane shows the content of this document, including a section header '1.4 C++ and C libraries and the std namespace', a paragraph explaining that all C++ standard library names are defined in the namespace `std`, and a list of methods to qualify library names: specifying the namespace (e.g., `std::printf`), using the `using` keyword, and using the compiler option `--using_std`. A note mentions that `errno` is a macro and does not need a namespace.

ARM Development Tools

Ocultar Buscar Atrás Adelante Imprimir Opciones

Contenido [índice] [Buscar] [Favoritos]

- Libraries and Floating Point Support Guide
 - Preface
 - The ARM C and C++ Libraries
 - Mandatory linkage with the C library
 - C and C++ runtime libraries
 - Summary of the C and C++ runtime
 - Compliance with the Application E
 - Increasing portability of object file
 - ARM C and C++ library directory s
 - Selection of ARM C and C++ libra
 - Thumb C libraries
 - C and C++ library features
 - C++ and C libraries and the std names**
 - Multithreaded support in ARM C librar
 - Support for building an application wit
 - Support for building an application wit
 - Tailoring the C library to a new execut
 - Assembler macros that tailor locale fu
 - Modification of C library functions for e
 - Stack and heap memory allocation ar
 - Tailoring input/output functions in the
 - Target dependencies on low-level fur
 - The C library printf family of functions
 - The C library scanf family of functions
 - Redefining low-level library functions l
 - The C library functions fread(), fgets()

C++ and C libraries and the std namespace

1.4 C++ and C libraries and the std namespace

All C++ standard library names, including the C library names, if you include them, are defined in the namespace `std`.

Standard library names are defined using the following C++ syntax:

```
#include <cstdlib> // instead of stdlib.h
```

This means that you must qualify all the library names using one of the following methods:

- Specify the standard namespace, for example:

```
std::printf("example\n");
```
- Use the C++ keyword `using` to import a name to the global namespace:

```
using namespace std;  
printf("example\n");
```
- Use the compiler option `--using_std`.

Note

`errno` is a macro, so it is not necessary to qualify it with a namespace.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

La Standard Peripheral Library de STM32L1

- Estos son los módulos HAL para STM32L1:

- misc.h
- stm32l1xx_adc.h
- stm32l1xx_aes.h
- stm32l1xx_comp.h
- stm32l1xx_crc.h
- stm32l1xx_dac.h
- stm32l1xx_dbgmcu.h
- stm32l1xx_dma.h
- stm32l1xx_exti.h
- stm32l1xx_flash.h
- stm32l1xx_fsmc.h
- stm32l1xx_i2c.h
- stm32l1xx_iwdg.h
- stm32l1xx_lcd.h
- stm32l1xx_opamp.h
- stm32l1xx_pwr.h
- stm32l1xx_rcc.h
- stm32l1xx_rtc.h
- stm32l1xx_sdio.h
- stm32l1xx_spi.h
- stm32l1xx_syscfg.h
- stm32l1xx_tim.h

Cartagena99

CLASES PARTICULARES, TUTORIAS TECNICAS ONLINE
LLAMA O ENVIA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

[Extra] Ejemplo de uso de RTC con registros

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Guía de Funcionamiento con registros

- Procedimiento de inicialización:
 - Se **desprotegen los registros de control** (RTC→WPR)
 - Se escribe primero **0xCA** y seguidamente **0x53**
 - Escribir cualquier cosa vuelve a bloquear los registros
 - El desbloqueo se mantiene
 - Se pone a **1** el bit **INIT** (RTC→ISR)
 - Se espera a que se ponga a **1** el bit **INITF** (RTC→ISR)
 - Se programa el **divisor de reloj, tanto síncrono como asíncrono** (RTC→PRER)
 - Se hace en dos pasos. Primero el síncrono (16 lsb) y luego el asíncrono (16msb)
 - Para el uso con el LSE se ponen de valores **255 para el síncrono** y **127 para el asíncrono**
 - Se escribe el valor de la **hora inicial** (RTC→TR)
 - Se escribe el valor de la **fecha inicial** (RTC→DR)
 - Se selecciona el **modo de 12 o 24 horas** en el bit **FMT** (RTC→CR)
 - Se pone a **0** el bit **INIT** (RTC→ISR)
 - Se **protegen los registros** (RTC→WPR = 0)
- Procedimiento de consulta de la hora:

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

RTC: Registros de Control

- **RTC→CR** – Control Register:

- 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'

Reserved										COE	OSEL[1:0]		POL	Reserved	BKP	SUB1H	ADD1H
										rw	rw	rw	rw		rw	w	w
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	DC	FMT	Reserved	REFCKON	TSEDGE	WUICKSEL[2:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

- **RTC→PRER** – Prescaler Register:

- Configura el preescalado asíncrono (PREDIV_A) y el síncrono (PREDIV-S). Para el uso con el LSE se ponen los valores **255 para el síncrono y 127 para el asíncrono**
- La frecuencia de funcionamiento del RTC será:

$$f_{CK_SPRE} = \frac{f_{RTCCLK}}{(PREDIV_S + 1) \times (PREVID_A + 1)}$$

Cartagena99

CLASES PARTICULARES, TUTORIAS TÉCNICAS ONLINE
LLAMA O ENVIA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



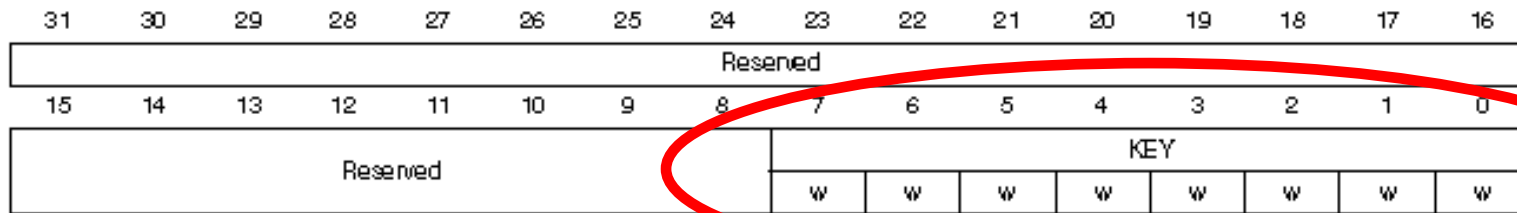
<http://dte.uc3m.es>

RTC: Registros de Control

- **RTC**→**WPR** – Write Protection Register:

- Se utiliza como se ha comentado anteriormente

- Primero se desprotegen los registros de control escribiendo primero **0xCA** y seguidamente **0x53**
- Cuando se haya terminado de configurar todo, protegen los registros escribiendo **0x00**



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

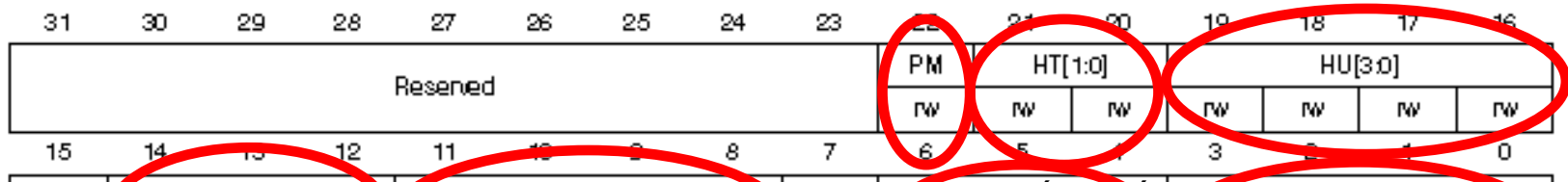
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

RTC: Registros de Datos

- **RTC**→**TR** – Time Register:

- 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



Cartagena99

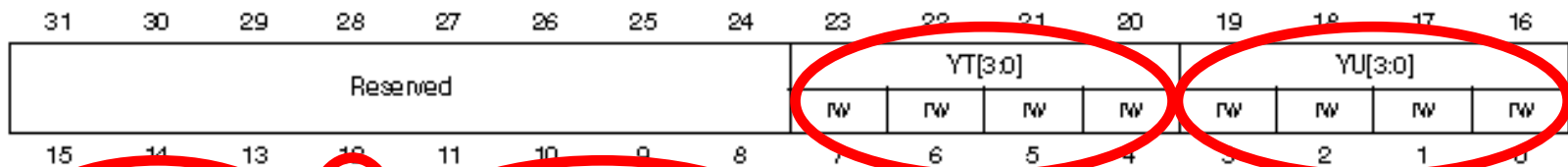
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

RTC: Registros de Datos

- **RTC**→**DR** – Date Register:

- 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



Cartagena99

CLASES PARTICULARES, TUTORIAS TÉCNICAS ONLINE
LLAMA O ENVIA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

RTC: Registros de Estado

- **RTC**→**ISR** – Initialization and Status Register:

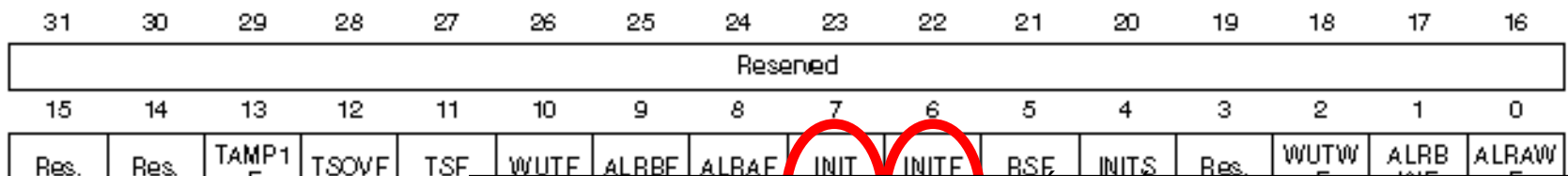
- De todos los bits sólo interesan:

- **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.



CLASES PARTICULARES, TUTORIAS TECNICAS ONLINE
LLAMA O ENVIA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

Ejemplo a nivel de registros

- A continuación se muestra el mismo ejemplo que el mostrado con HAL, pero ahora configurado con registros (en este caso también se muestran los segundos en la hora y el año en la fecha).
- Las configuraciones en CUBE no vale la misma que para el ejemplo con HAL. Consulta la transparencia 21 para ver qué hay que ajustar para que funcione el ejemplo con registros.
- Los programa para μ Vision es el mostrado a continuación, teniendo en cuenta colocar cada parte del código en su sitio, es decir:
 - Variables globales entre `/* USER CODE BEGIN PV */ y /* USER CODE END PV */`
 - Inicialización entre `/* USER CODE BEGIN 2 */ y /* USER CODE END 2 */`
 - Funcionamiento cíclico entre `/* USER CODE BEGIN WHILE */ y /* USER CODE END WHILE */`

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

<http://dte.uc3m.es>

Cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al

Artículo 17 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002,

si la información contenida en el documento es ficticia o lesiona bienes o derechos de un tercero haganoslo saber y será retirada.

(C) Raúl Sánchez Reíllo

Ejemplo a nivel de registros

- El mismo ejemplo que el mostrado con HAL, pero ahora con registros: la fecha es 21/03/18 y la hora empieza en 18:30:20
 - Inicialización:

```
int main(void) {

    unsigned char cadena[6];
    unsigned valor;

    // Funciones de inicialización del LCD
    BSP_LCD_GLASS_Init();
    BSP_LCD_GLASS_BarLevelConfig(0);
    BSP_LCD_GLASS_Clear();

    // Si se utiliza la V5.22 de µVision en
    adelante hay que añadir esta fila al
    principio para activar el RTC en la
    placa

    RCC->CSR |= (1<<22);
    // Inicialización del RTC
    RTC->WPR=0xCA;
    RTC->WPR=0x53;
    RTC->ISR |= (1<<7);
    while ((RTC->ISR & (1<<6))==0);
    RTC->PRER=255;
    RTC->PRER|=127<<16;
    RTC->TR = 0x00183020;
    RTC->DR = 0x00186321;
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Ejemplo a nivel de registros

- Funcionamiento continuo:

```
while (1) {
    // Obtengo la hora y la voy filtrando para sacarla por el LCD en 6 dígitos: HHMMSS
    valor = RTC->TR;
    cadena[5]=(valor & 0x0000000F)+'0';
    valor = valor >> 4;
    cadena[4]=(valor & 0x00000007)+'0';
    valor = valor >> 4;
    cadena[3]=(valor & 0x0000000F)+'0';
    valor = valor >> 4;
    cadena[2]=(valor & 0x00000007)+'0';
    valor = valor >> 4;
    cadena[1]=(valor & 0x0000000F)+'0';
    valor = valor >> 4;
    cadena[0]=(valor & 0x00000003)+'0';
    valor = valor >> 4;
    BSP_LCD_GLASS_DisplayString((uint8_t *)cadena);
    espera(10000000);
    // Obtengo la fecha y la voy filtrando para sacarla por el LCD en 6 dígitos: DDMMAA
    valor = RTC->DR;
    cadena[1]=(valor & 0x0000000F)+'0';
    valor = valor >> 4;
    cadena[0]=(valor & 0x00000003)+'0';
    valor = valor >> 4;
    cadena[3]=(valor & 0x0000000F)+'0';
    valor = valor >> 4;
    cadena[2]=(valor & 0x00000001)+'0';
    valor = valor >> 4;
    cadena[5]=(valor & 0x0000000F)+'0';
    valor = valor >> 4;
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

<http://dt>

Cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al

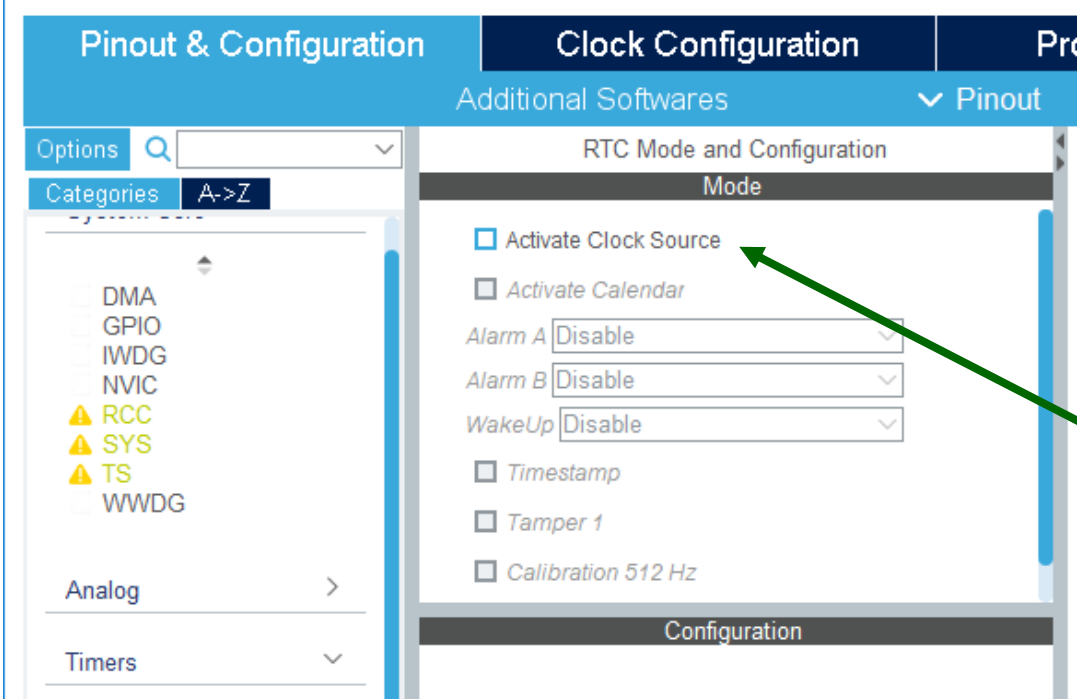
Artículo 17 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002,

si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

(C) Raúl Sánchez Reíllo

STM32 CubeMX: RTC con registros

- CUIDADO: En este caso, a diferencia del ejemplo mostrado con HAL, hay que desactivar la fuente de reloj y el calendario en CubeMX. Si no, aparece un fallo de compilación por duplicación de ajuste



Desactiva la fuente de reloj y el calendario en CubeMX



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70