

# *Ingeniería de Sistemas Electrónicos*

## *Programación avanzada de microcontroladores (II)*

- 
- Integración de referencia temporal en el sistema
  - Desarrollo de aplicaciones “confiables”
  - Reducción del consumo
  - Desarrollo de software. Control de versiones
  - Prácticas de laboratorio

# *Ingeniería de Sistemas Electrónicos*

## *Programación avanzada de microcontroladores (II)*

- 
- Integración de referencia temporal en el sistema
  - Desarrollo de aplicaciones “confiables”
  - Reducción del consumo
  - Desarrollo de software. Control de versiones
  - Prácticas de laboratorio

# Integración de referencia temporal en el sistema



# *Formas de mantener una referencia temporal*

---

## □ Reloj software “local”

- Timer con interrupciones periódicas (cada segundo?)
- Puesta en hora inicial
- Mantenimiento de la hora
- Desventajas??

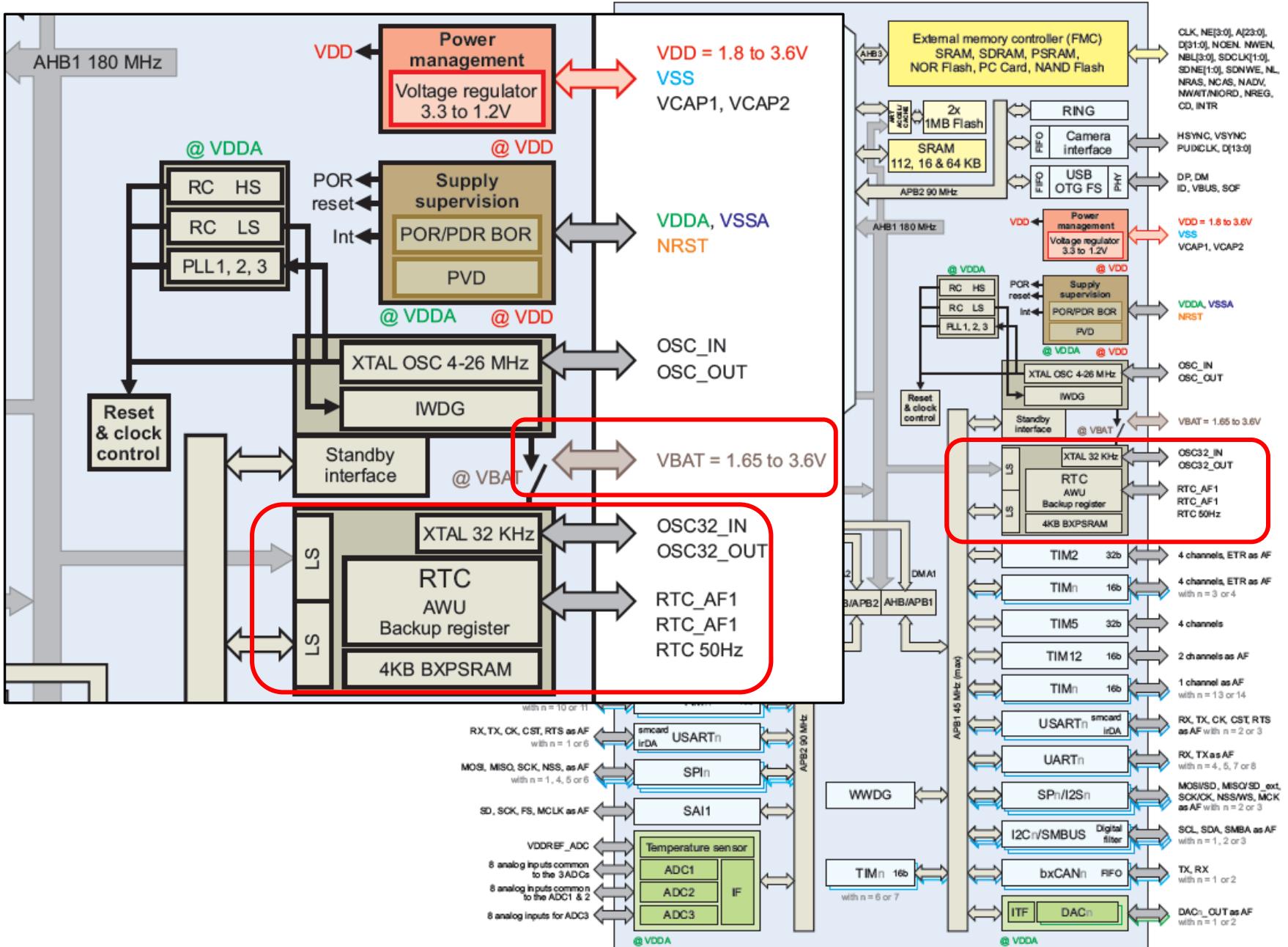
## □ Reloj hardware (RTC, real time clock)

- Hardware específico (interno o externo)
- Puesta en hora
- Mantenimiento de la hora automático
- Desventajas??

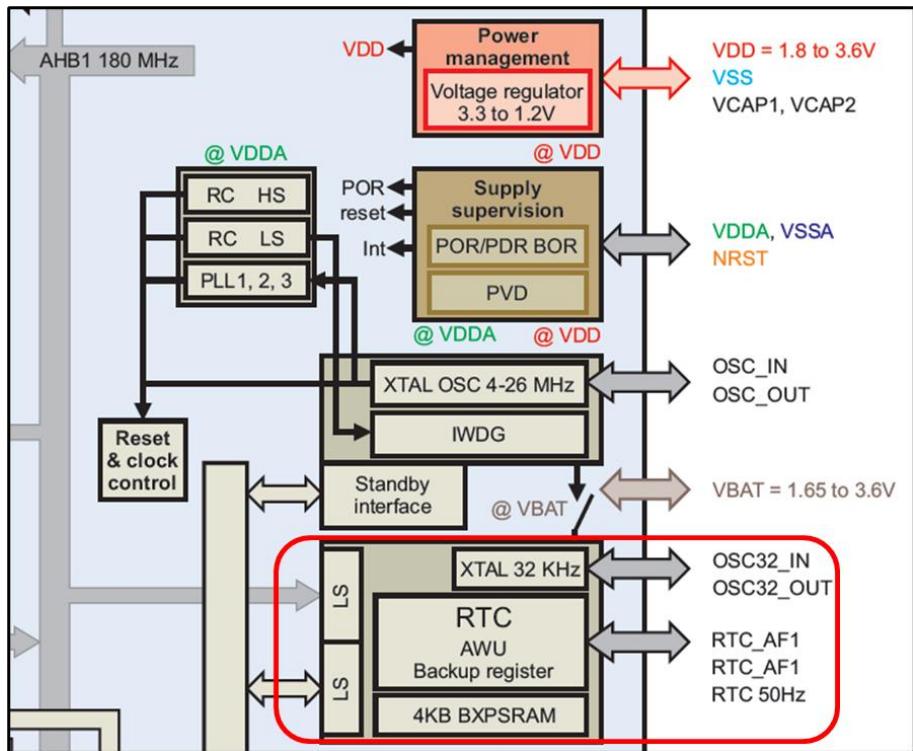
## □ Servidor de tiempo en red

- Proporciona la fecha y hora actuales
- Protocolo SNTP (Simple Network Time Protocol), versión simplificada de NTP
- Desventajas??

# *STM32F4 Real Time Clock (RTC)*



# STM32F4 Real Time Clock (RTC)



## RTC:

- ✓ Special peripheral
- ✓ Independent power domain
- ✓ Independent oscillator
- ✓ Special timer with dedicated registers to store calendar and time data such as years, months, days, hours, minutes, and seconds.
- ✓ RTC plays a major role in power-saving strategies as it can be the only domain to stay awake while every other peripheral has been turned into standby or sleeping modes.

- ✓ Very low power consumption
- ✓ You can even turn the STM32 device off VDD and still keep the RTC running as long as you power supply the VBAT pin (using a button battery for instance).

# *STM32F4 Real Time Clock (RTC)*

- ✓ All STM32 microcontrollers provide an integrated RTC unit that is not limited to keeping track of the current date/time. In fact, RTC provides some additional and relevant features such as antitampering detection, generation of alarm events and the ability to wake-up the MCU from deeper low-power modes.

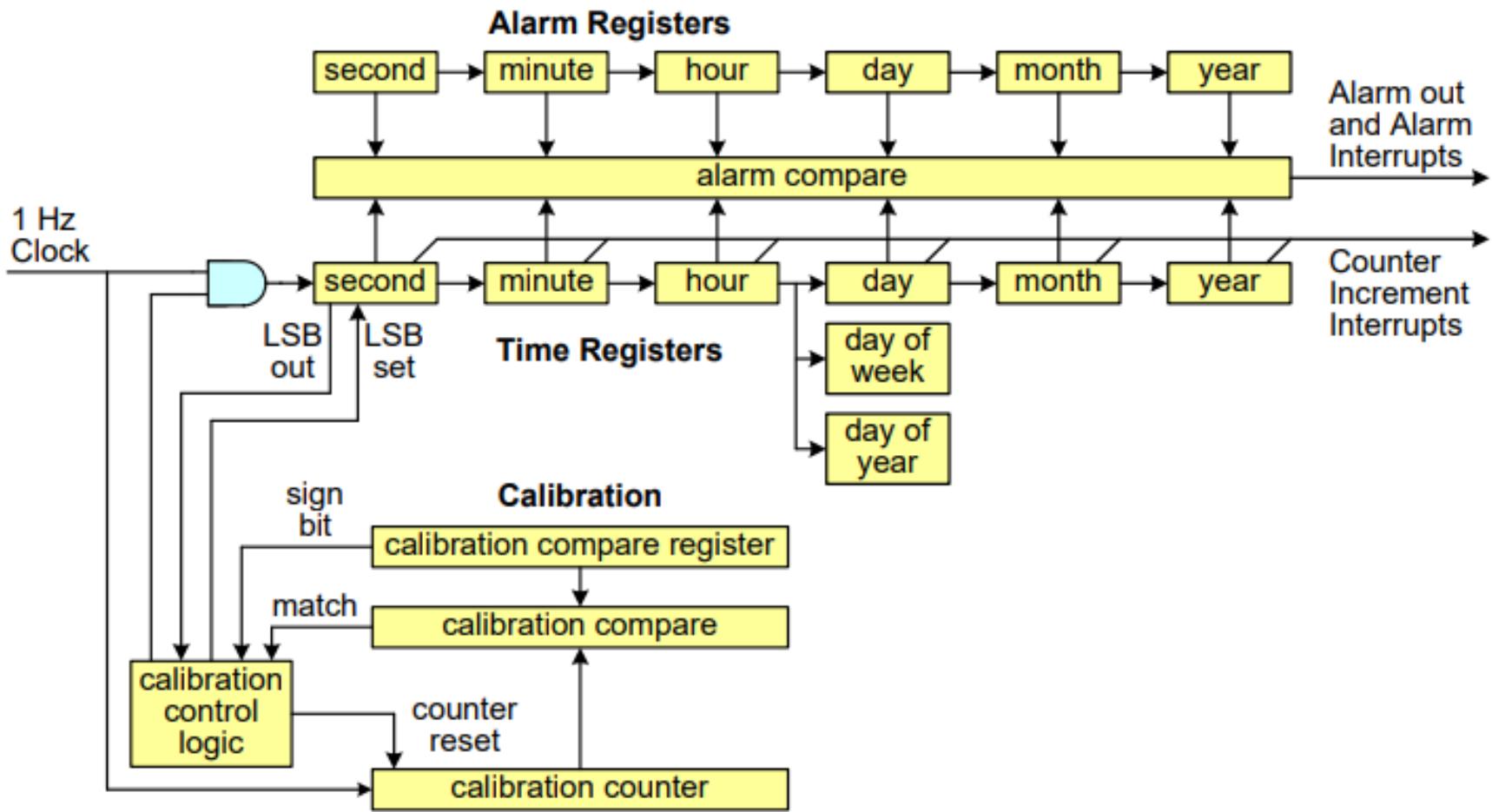
The screenshot shows a PDF document titled "STM32F429-REFERENCE-MANUAL.pdf" open in Adobe Acrobat Pro 2020. The document is a reference manual for the STM32F429 microcontroller, specifically focusing on the Real-time clock (RTC).

The left sidebar contains a table of contents with the following items:

- 15 Digital camera interface (DCMI)
- 16 LCD-TFT controller (LTDC)
- 17 Advanced-control timers (TIM1 and TIM8)
- 18 General-purpose timers (TIM2 to TIM5)
- 19 General-purpose timers (TIM9 to TIM14)
- 20 Basic timers (TIM6 and TIM7)
- 21 Independent watchdog (IWDG)
- 22 Window watchdog (WWDG)
- 23 Cryptographic processor (CRYP)
- 24 Random number generator (RNG)
- 25 Hash processor (HASH)
- 26 Real-time clock (RTC)** (This item is highlighted in gray)
- 27 Inter-integrated circuit (I2C) interface
- 28 Serial peripheral interface (SPI)
- 29 Serial audio interface (SAI)
- 30 Universal synchronous asynchronous

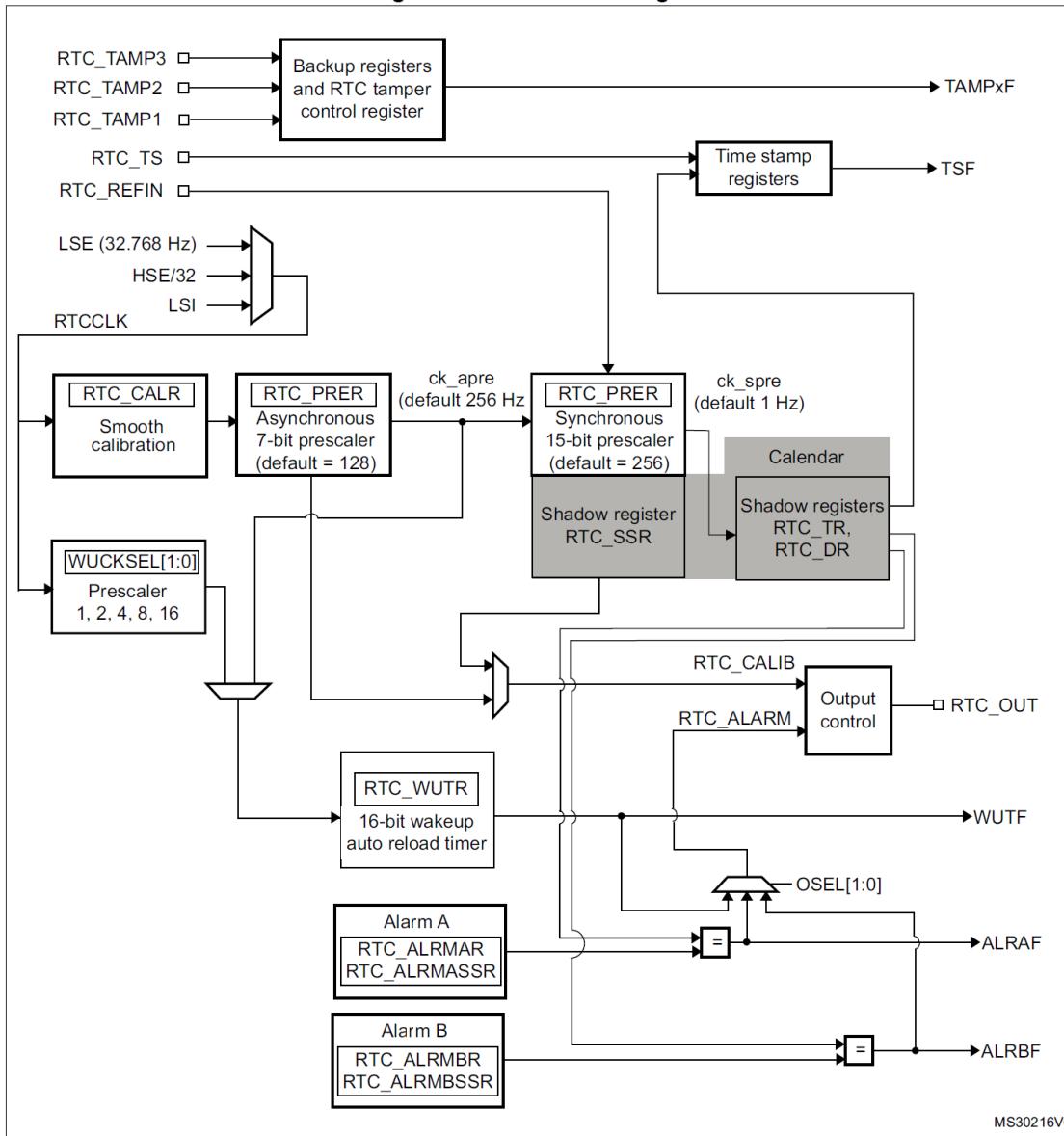
The main content area displays the "Real-time clock (RTC)" chapter. The title "Real-time clock (RTC)" is at the top, followed by the identifier "RM0090". The chapter begins with the heading "26 Real-time clock (RTC)". A note below the heading states: "This section applies to the whole STM32F4xx family, unless otherwise specified." The "Introduction" section follows, which includes a detailed description of the RTC's functions and capabilities. Below the introduction, there are several paragraphs explaining various features and how they are implemented. At the bottom of the page, there is a small note about digital calibration.

# *NXP LPC1768 (Cortex M3) Real Time Clock (RTC)*



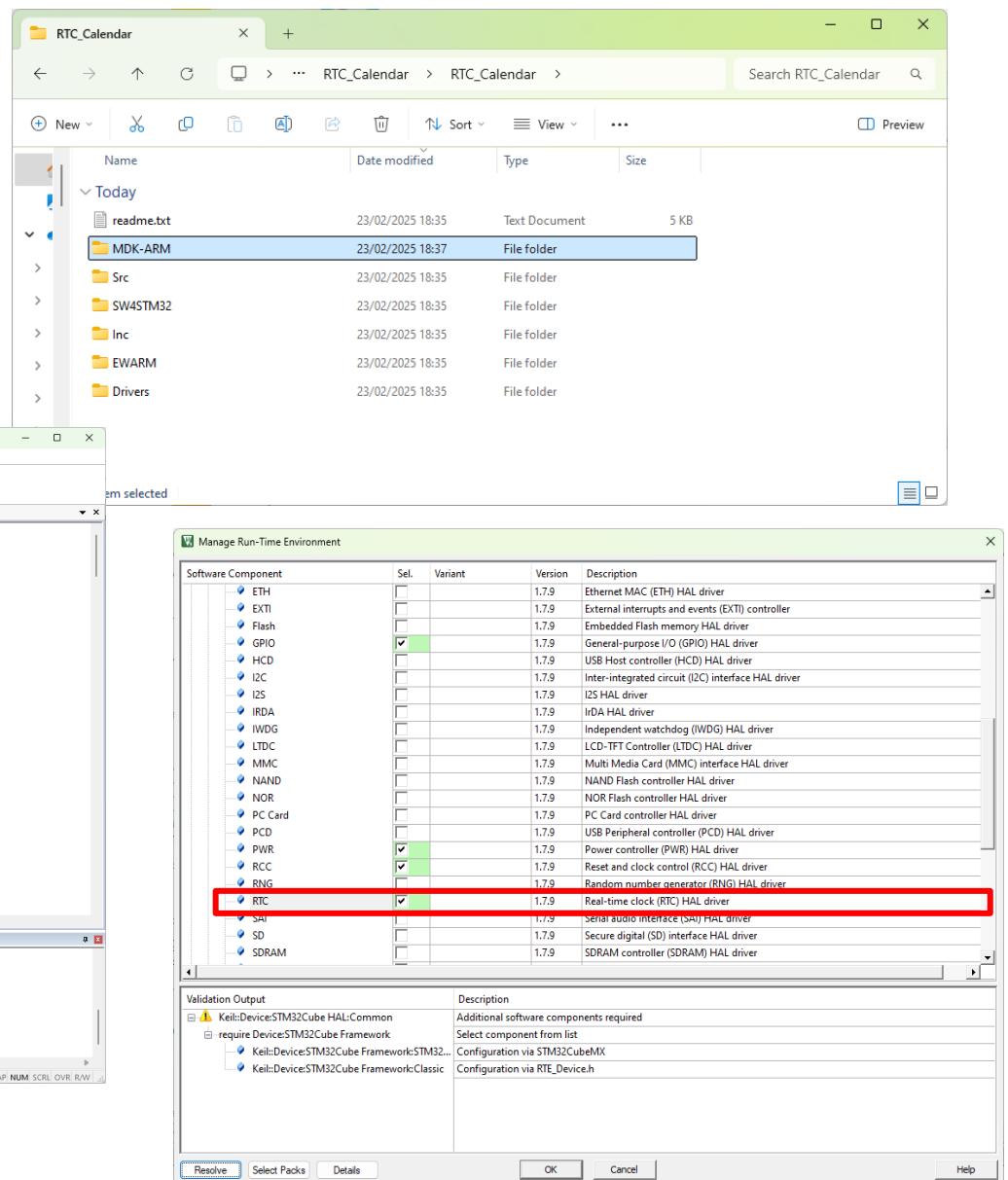
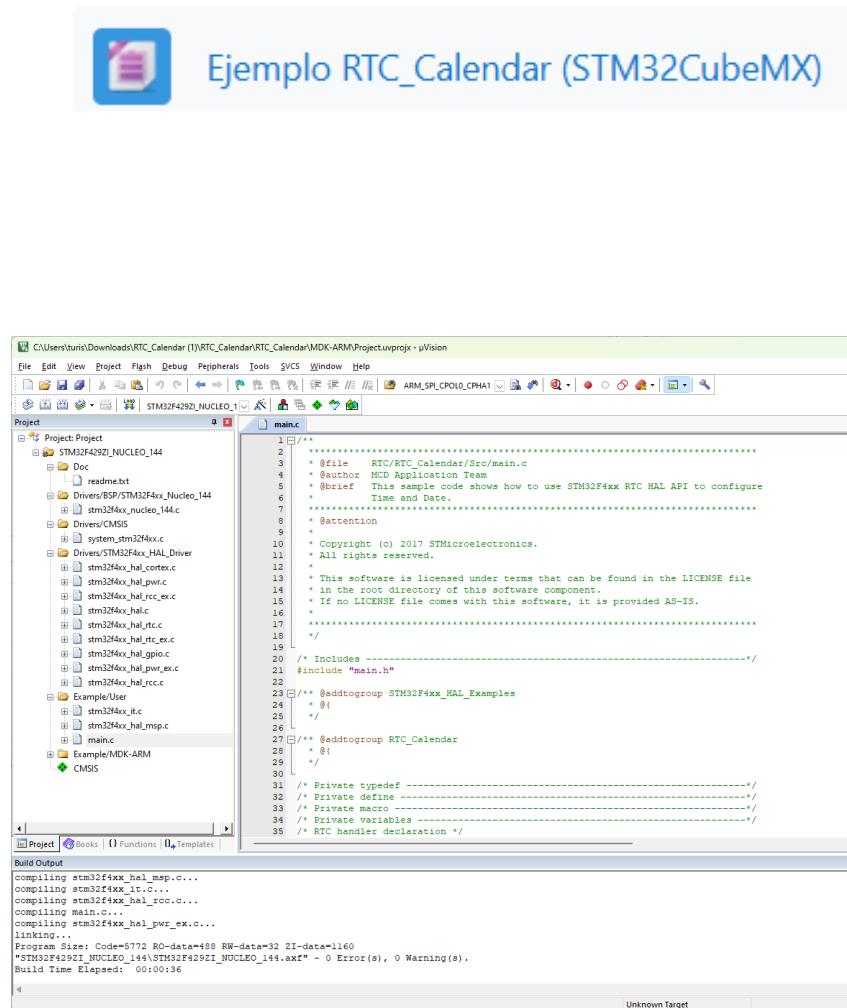
# Real Time Clock (RTC) STM32F429

Figure 237. RTC block diagram



- On STM32F4xx devices, the RTC\_AF1 and RTC\_AF2 alternate functions are connected to PC13 and PI8, respectively.

# *STM32F4 Real Time Clock (RTC)*



## □ Reloj software “local”

- Timer con interrupciones periódicas (cada segundo?)
- Puesta en hora inicial
- Mantenimiento de la hora
- Desventajas??

## □ Reloj hardware (RTC, real time clock)

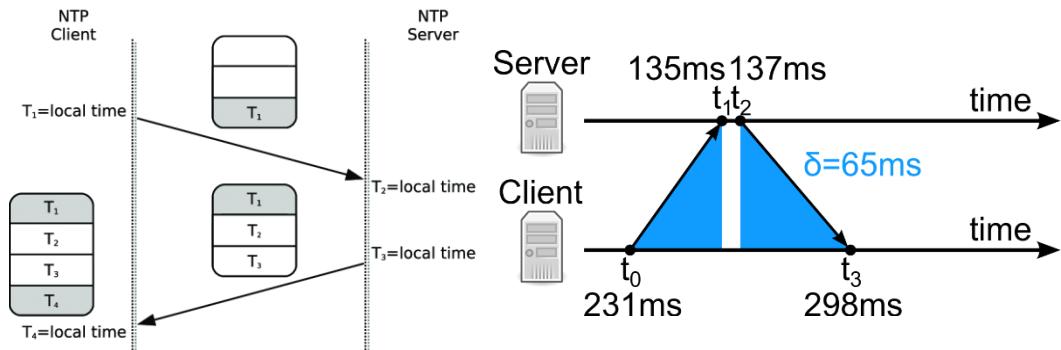
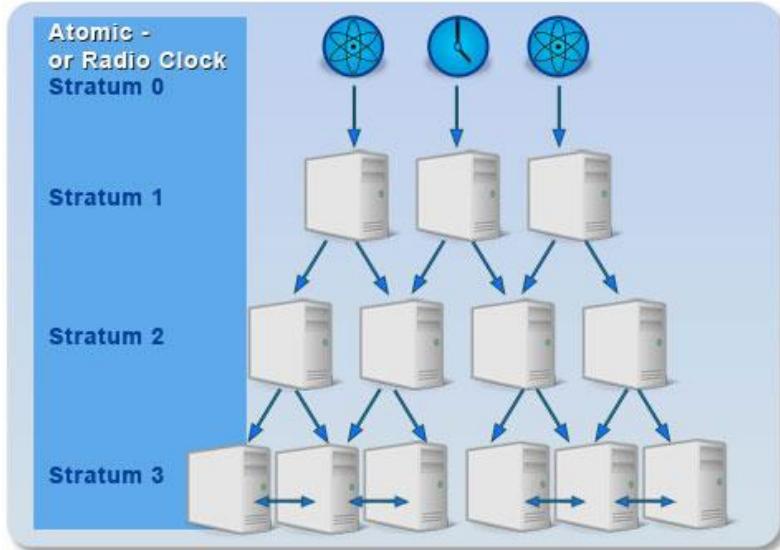
- Hardware específico (interno o externo)
- Puesta en hora
- Mantenimiento de la hora automático
- Desventajas??

## □ Servidor de tiempo en red

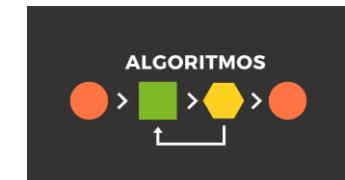
- Proporciona la fecha y hora actuales
- Protocolo SNTP (Simple Network Time Protocol), versión simplificada de NTP
- Desventajas??

# Servidor de tiempo en red. Protocolos NTP y SNTP

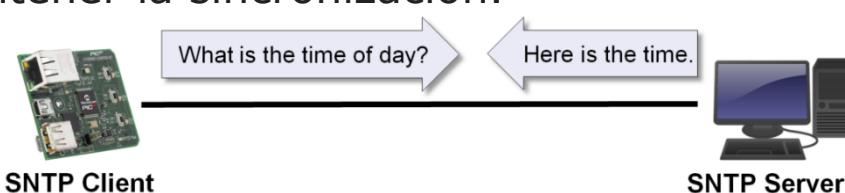
**Network Time Protocol** (NTP) es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable. NTP utiliza UDP como su capa de transporte, usando el puerto 123. Está diseñado para resistir los efectos de la latencia variable.



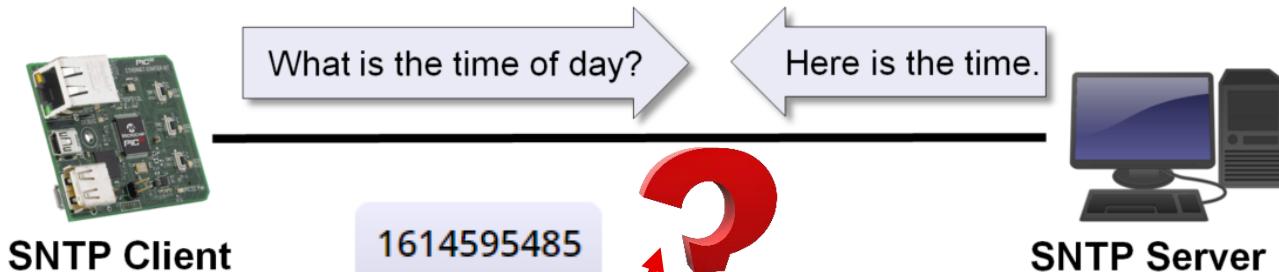
$$\text{offset} = [(T_2 - T_1) + (T_3 - T_4)] / 2, \\ \text{delay} = (T_4 - T_1) - (T_3 - T_2).$$



**Simple Network Time Protocol** (**SNTP**) es una versión simplificada de NTP para sistemas con menor capacidad de procesamiento y menores requerimientos de tiempo. Es compatible con NTP, los paquetes intercambiados entre el cliente y el servidor de tiempo son los mismos, pero el cliente no necesita ejecutar algoritmos para determinar y mantener la sincronización.



# Protocolo SNTP



**Network Time Foundation** <http://www.ntp.org/>



**NTP Pool  
Project**

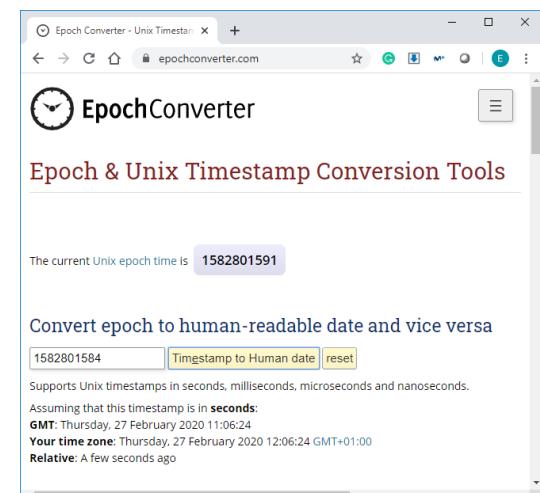
<https://www.ntppool.org/> → Lista de servidores NTP

The current Unix epoch time is

1582801685

Segundos transcurridos desde  
el 1 de enero de 1970

<https://www.epochconverter.com/>



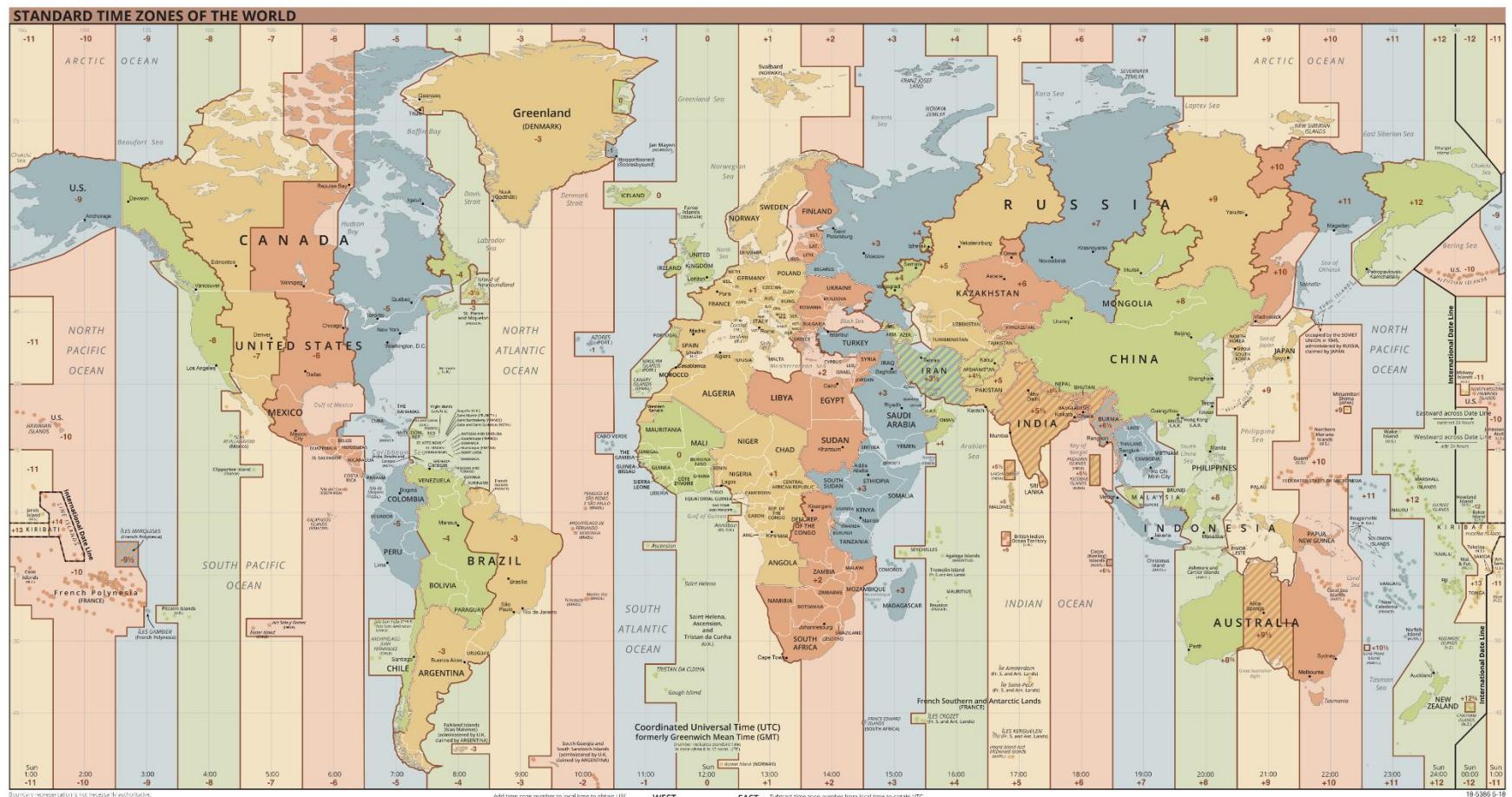
## *Protocolo SNTP*

The current Unix epoch time is

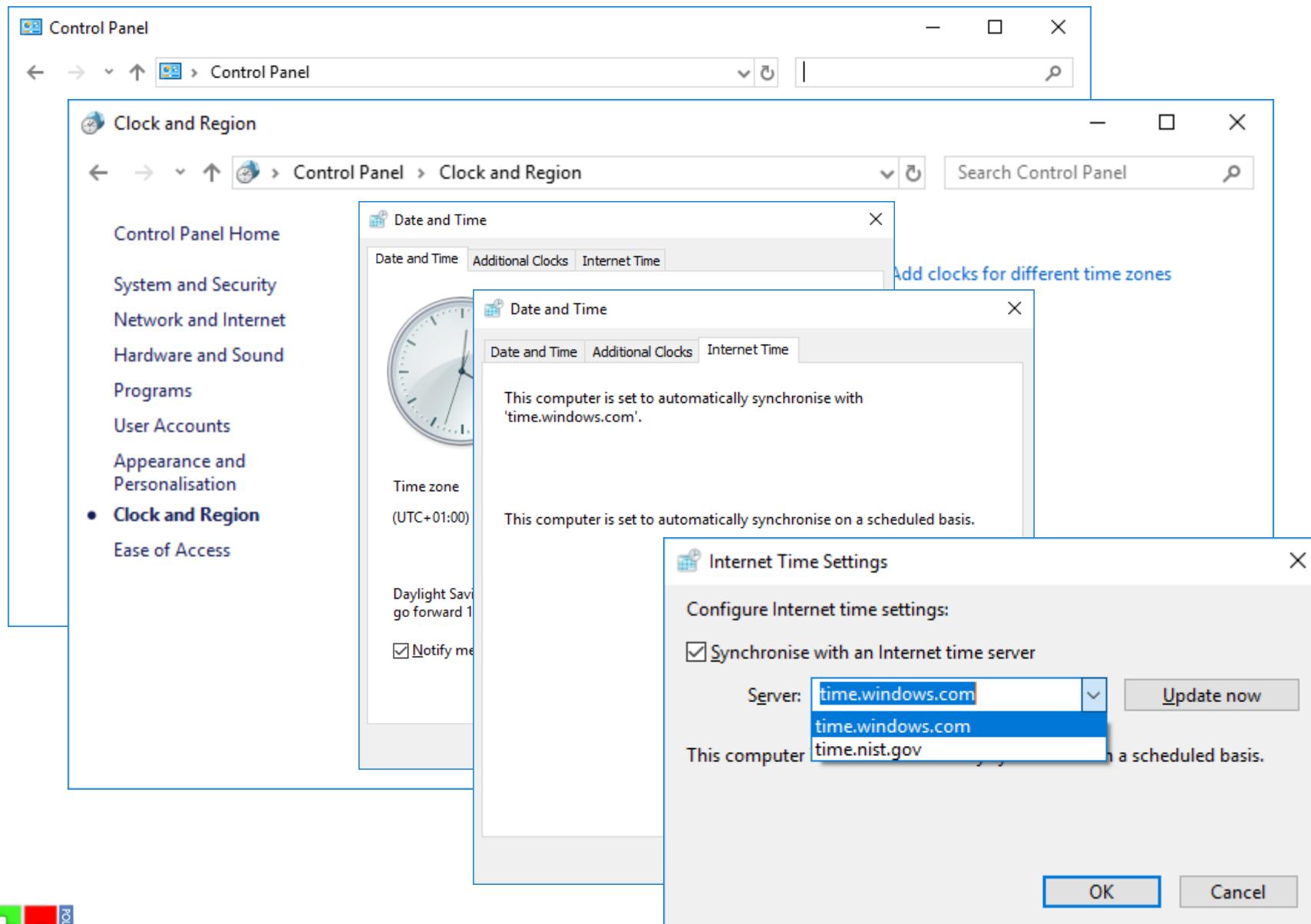
1582801685

Segundos transcurridos desde el 1 de enero de 1970

# Hora UTC (Coordinated Universal Time)



# Protocolo SNTP. Uso en ordenadores, móviles...



# Protocolo SNTP. Integración en aplicación servidor Web

Manage Run-Time Environment

Software Component	Sel.	Variant	Version	Description
Board Support		32F469IDISCOVERY	1.0.0	<a href="#">STMicroelectronics 32F469IDISCOVERY Kit</a>
CMSIS				<a href="#">Cortex Microcontroller Software Interface Components</a>
CMSIS Driver				<a href="#">Unified Device Drivers compliant to CMSIS-Driver Specifications</a>
Compiler		ARM Compiler	1.6.0	<a href="#">Compiler Extensions for ARM Compiler 5 and ARM Compiler 6</a>
Device				<a href="#">Startup, System Setup</a>
File System		MDK-Plus	6.14.1	<a href="#">File Access on various storage devices</a>
Graphics		MDK-Plus	6.16.3	<a href="#">User Interface on graphical LCD displays</a>
Graphics Display				Display Interface including configuration for emWIN
Network		MDK-Pro	7.15.0	<a href="#">IPv4/IPv6 Networking using Ethernet or Serial protocols</a>
CORE		IPv4/IPv6 Debug	7.15.0	IPv4/IPv6 Networking Core for Cortex-M (Debug)
Legacy API			7.15.0	Network Legacy API support
Interface				Connection Mechanism
Service				Network Services
DNS Client			7.15.0	<a href="#">DNS Client</a>
FTP Client			7.15.0	<a href="#">FTP Client</a>
FTP Server			7.15.0	<a href="#">FTP Server</a>
SMTP Client		SMTP	7.15.0	<a href="#">Email Client (SMTP)</a>
SNMP Agent			7.15.0	<a href="#">SNMP Agent</a>
<b>SNTP Client</b>	<b>✓</b>		7.15.0	<b>SNTP Client</b>
TFTP Client			7.15.0	<a href="#">TFTP Client</a>
TFTP Server			7.15.0	<a href="#">TFTP Server</a>
Telnet Server			7.15.0	<a href="#">Telnet Server</a>
Web Server Compact	<b>✓</b>	HTTP	7.15.0	<b>Web Server (HTTP) with Read-only Web Resources (Compact)</b>

Validation Output

	Description

Net\_Config\_SNTP\_Client.h

Option	Value
SNTP Client	
Broadcast Mode	<input type="checkbox"/>
NTP Server	217.79.179.106

Text Editor Configuration Wizard

# Protocolo SNTP. Integración en aplicación

The screenshot shows a web browser window displaying the ARM KEIL Network Component documentation. The title bar reads "SNTP Client". The address bar shows the URL: "C:/Users/Eduardo/AppData/Local/Arm/Packs/Keil/MDK-Middleware/7.13.0/Doc/Network/html/group\_\_net\_s\_n\_t\_pc\_\_func.html". The main content area is titled "Network Component Version 7.15.0" and "MDK Middleware for IPv4 and IPv6 Networking". The "Network" tab is selected in the navigation bar. The left sidebar contains a tree view of documentation topics, with "SNTP Client" currently selected. The right panel displays the "SNTCP Client Services" page, which includes sections for "Content", "Control Interface", "Configuration", "Enumerations", "Description", and a "Note" section. The note states that the SNTP Client supports IPv4 connections only and provides UTC time information from an NTP server.

An SNTP Client delivers precise timing information over an IP network. More...

### Content

**Control Interface**  
Functions to start the SNTP Client.

**Configuration**  
Configuration of the SNTP client in µVision.

**Enumerations**  
Enumerations of the SNTP Client.

### Description

An SNTP Client delivers precise timing information over an IP network.

Simple Network Time Protocol (**SNTP**) is a networking protocol for **clock synchronization** between computer systems over IP based, variable-latency data networks. SNTP does not require the storage of the state over extended periods of time, so that it is used mostly in memory limited embedded devices and in applications where high accuracy timing is not required.

**Note**  
The Network Component's SNTP Client supports IPv4 connections only.

The protocol provides **Coordinated Universal Time (UTC)** including scheduled leap second adjustments. No information about time zones or daylight saving time is transmitted. This information must be obtained separately.

The SNTP Client enables the retrieval of UTC coded time information from a given NTP server. The IP address of the server can be specified separately.

Generated on Tue May 25 2021 13:57:38 for Network Component by ARM Ltd. All rights reserved.

# Protocolo SNTP. Integración en aplicación

Control Interface x +  
File | C:/Users/Eduardo/AppData/Local/Arm/Packs/Keil/MDK-Middleware/7.13.0/Doc/Network/html/group\_starting\_sntp.html#ga5fb2de3e22a5aac8c29e1e7efdb567ca

arm KEIL Network Component Version 7.15.0  
MDK Middleware for IPv4 and IPv6 Networking

General File System Graphic Network USB Board Support

Main Page Usage and Description Reference Search

Secure Communication  
Cyber Security  
Network Examples  
Migration  
Resource Requirements  
Function Overview  
Reference  
Overview  
Services  
HTTP Server  
FTP Server  
FTP Client  
TFTP Server  
TFTP Client  
Telnet Server  
SNMP Agent  
DNS Client  
SNTP Client  
Control Interface  
netSNTPc\_cb\_t  
netSNTPc\_GetTime  
netSNTPc\_GetTimeX  
netSNTPc\_SetMode  
Configuration  
Enumerations  
SMTP Client  
Ping Client  
Sockets  
Interfaces  
IP Address  
System Functions (User)  
Status and Error Codes  
Modem Driver Functions  
Debug Events  
Data Structures  
Data Structure Index

**netStatus netSNTPc\_GetTime ( const NET\_ADDR \* addr,  
                                  netSNTPc\_cb\_t    cb\_func  
                                  )**

Determine current time from NTP or SNTP time server. [thread-safe].

**Parameters**

[in] **addr** structure containing IP address of NTP or SNTP server.  
• NULL to use NTP server IP address from system configuration.

[in] **cb\_func** callback function to call, when the session ends.

**Returns**  
status code that indicates the execution status of the function.

In **unicast** mode, the non-blocking function **netSNTPc\_GetTime** sends a get time request to NTP server specified with the argument **addr**. If the IP address is not specified (**addr** is **NULL**), then the IP address of the NTP Server configured in **Net\_Config\_SNTP\_Client.h** is used.

In **broadcast** mode, it opens an UDP socket for receiving broadcast NTP messages. You can use broadcast mode, if you have a broadcasting NTP server in your local network. NTP messages that come from the specified NTP server are accepted. Messages that come from a different NTP server are ignored. If the specified NTP server address is unknown address (0.0.0.0), then messages from any NTP server are accepted.

The argument **addr** is a pointer to the structure containing the IP address and port of a local or public NTP server.

If the IP address is not specified (**addr** is **NULL**), then the IP address of the NTP Server configured in **Net\_Config\_SNTP\_Client.h** is used. **Broadcast Mode** gets also enabled here.

The argument **cb\_func** points to a user-defined callback function called by the SNTP client when an NTP message is received from the server. Refer to **netSNTPc\_cb\_t**.

Possible **netStatus** return values:

- **netOK**: Operation completed successfully.
- **netInvalidParameter**: Invalid or not supported parameter.
- **netWrongState**: SNTP client state incorrect for the current operating mode.
- **netServerError**: NTP or SNTP server IP address undefined.

**Code Example**

```
const NET_ADDR4 ntp_server = { .NET_ADDR_IP4, 0, 217, 79, 179, 106 };
static void time_callback (uint32_t seconds, uint32_t seconds_fraction);

void get_time (void) {
    if (netSNTPc_GetTime ((NET_ADDR *)&ntp_server, time_callback) == netOK) {
        printf ("SNTP request sent.\n");
    }
    else {
        printf ("SNTP not ready or bad parameters.\n");
    }
}

static void time_callback (uint32_t seconds, uint32_t seconds_fraction) {
    if (seconds == 0) {
        printf ("Received no response or bad response.\n");
    }
}
```

Generated on Tue May 25 2021 13:57:38 for Network Component by ARM Ltd. All rights reserved.

Segundos desde 01/01/1970, utilizar función para su conversión.

# Protocolo SNTP. Integración en aplicación

<https://www.epochconverter.com/programming/c>

The screenshot shows a web browser with two tabs open. The left tab, titled 'Epoch Converter - Unix Timestamp', contains code examples for various programming languages to convert epoch timestamps. The right tab, titled 'C Epoch Converter Routines', shows a detailed C code example for epoch conversion.

**Left Tab: Convert from epoch to human-readable date**

- PHP:** `date(output format, epoch);` Output format example: 'r' = RFC 2822 date, [more PHP examples](#)
- Python:** `import time; time.strftime("%a, %d %b %Y %H:%M:%S +0000", time.localtime(epoch))`  
Replace `time.localtime` with `time.gmtime` for GMT time. Or using `datetime`: `import datetime;`  
`datetime.datetime.utcfromtimestamp(epoch).replace(tzinfo=datetime.timezone.utc)`
- Ruby:** `Time.at(epoch)`
- C#:** `private string epoch2string(int epoch) { return new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc).AddSeconds(epoch).ToShortDateString(); }`
- Perl:** Use the Perl Epoch routines
- Java:** `String date = new java.text.SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new java.util.Date (epoch*1000));` Epoch in seconds
- Lua:** `datestring = os.date([format[,epoch]])`
- VBScript/ASP:** `DateAdd("s", epoch, "01/01/1970 00:00:00")`
- AutoIT:** `_DateAdd("s", $EpochSeconds , "1970/01/01 00:00:00")`
- Delphi:** `myString := DateTimeToStr(UnixToDate(DateTime(Epoch)))`
- C:** Use the C Epoch Converter routines
- Objective-C:** `NSDate * myDate = [NSDate dateWithTimeIntervalSince1970:(epoch)];`
- R:** `as.POSIXct(epoch, origin="1970-01-01", tz="UTC")`
- Go:** [Example code ↗](#)
- Adobe ColdFusion:** `DateAdd("s",epoch,"1/1/1970");`
- MySQL:** `FROM_UNIXTIME(epoch, optional output format)`  
If you need support for negative timestamps: DATE

**Right Tab: Convert from epoch to human-readable date**

How to convert epoch / UNIX timestamps to normal readable date/time using C.

Example C routine using `strftime`. `strftime` converts information from a time structure to a string form, and writes the string into the memory area pointed to by "string".

```
#include <stdio.h>
#include <time.h>

int main(void)
{
    time_t now;
    struct tm ts;
    char buf[80];

    // Get current time
    time(&now);

    // Format time, "ddd yyyy-mm-dd hh:mm:ss zzz"
    ts = *localtime(&now);
    strftime(buf, sizeof(buf), "%a %Y-%m-%d %H:%M:%S %Z", &ts);
    printf("%s\n", buf);
    return 0;
}
```

For more information on `strftime` [click here](#).

With a custom epoch (here 1262304000):

**Pages**

- Home
- Preferences
- Toggle theme ↗

**Tools ↗**

- Epoch converter
- Batch converter
- Time zone converter
- Timestamp list
- LDAP converter
- WebKit/Chrome timestamp
- Unix hex timestamp
- Cocoa Core Data timestamp
- Mac HFS+ timestamp
- SAS timestamp
- Seconds/days since year 0
- Bin/Oct/Hex converter
- Countdown in seconds
- Epoch clock

**Date and Time ↗**

- Week numbers
- Weeks by year
- Day numbers
- Days by year
- Years/leap years
- Calculate the difference between two dates

# *Formas de mantener una referencia temporal*

---

## □ Reloj software “local”

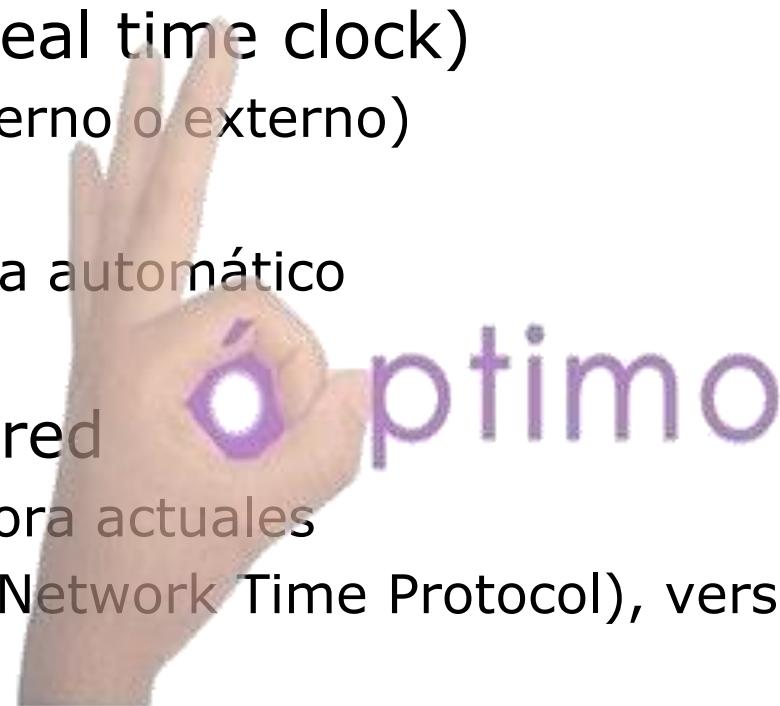
- Timer con interrupciones periódicas (cada segundo?)
- Puesta en hora inicial
- Mantenimiento de la hora
- Desventajas??

## □ Reloj hardware (RTC, real time clock)

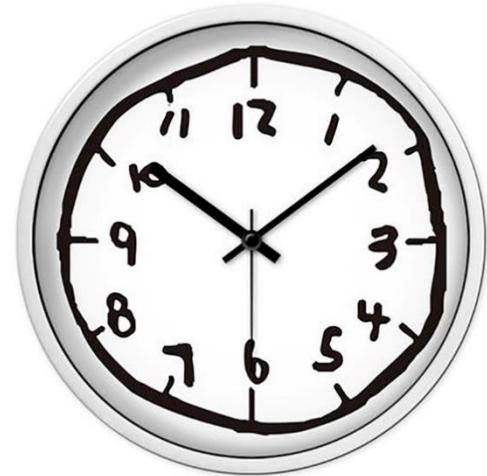
- Hardware específico (interno o externo)
- Puesta en hora
- Mantenimiento de la hora automático
- Desventajas??

## + Servidor de tiempo en red

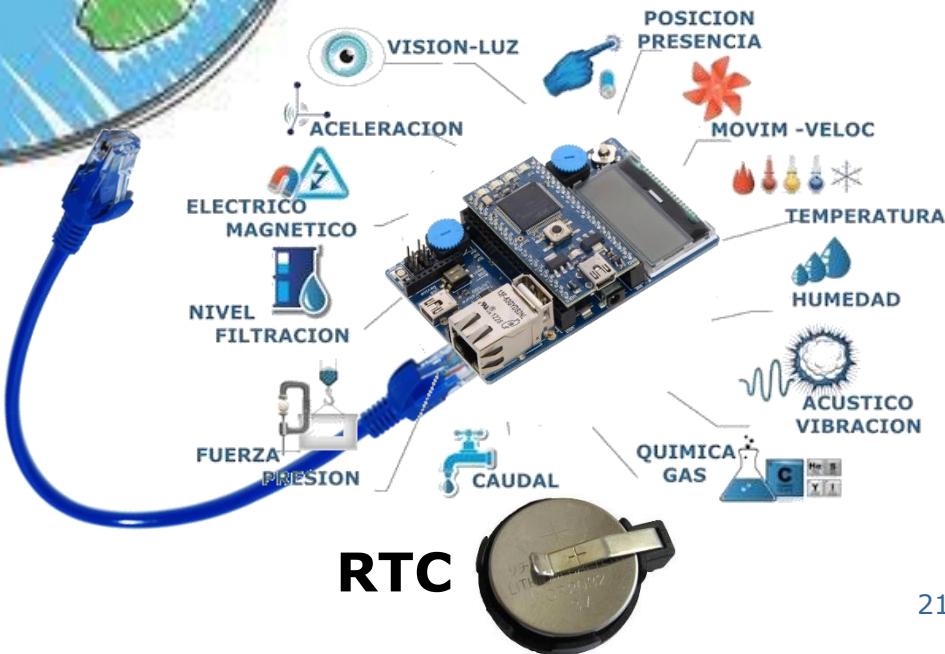
- Proporciona la fecha y hora actuales
- Protocolo SNTP (Simple Network Time Protocol), versión simplificada de NTP
- Desventajas??



# Implementación práctica



**NTP server**



- **Práctica 2:** Soporte de referencias de tiempo:
  - Integración de RTC y SNTP en servidor Web
- ❖ 1 semana

# *Ingeniería de Sistemas Electrónicos*

## *Programación avanzada de microcontroladores (II)*

- 
- Integración de referencia temporal en el sistema
  - Desarrollo de aplicaciones “confiables”
  - Reducción del consumo
  - Desarrollo de software. Control de versiones
  - Prácticas de laboratorio

# *Desarrollo de aplicaciones “confiables”*

---



- Alguna vez os ha fallado una aplicación que habéis desarrollado para un microcontrolador??
- Os suenan de algo estas frases???
  - Te juro que ayer me funcionaba!!!!
  - Uhmmm, creo que eso no lo había probado nunca...
  - Pues esto antes iba...
  - Se ha quedado “colgado”, resetéalo
  - ....

# *Desarrollo de aplicaciones “confiables”*

---



Los tres incorporan  
microcontroladores, pero...  
¿dónde es más crítico un error?

## Desarrollo de firmware de alta calidad y con criterios functional safety

Seminarios gratuitos

MARZO 2020

**24**  
Martes

MADRID

MARZO 2020

**31**  
Martes

BARCELONA

Organiza:

**CAPEL**

CAPTURA ELECTRÓNICA

**arm**

**LAUTERBACH**  
DEVELOPMENT TOOLS

**RAZORCAT**



IEC 61508



ISO 26262



IEC 62304



EN 50128



DO-178C

# Desarrollo de aplicaciones “confiables”



**14th Annual  
AUTOMOTIVE FUNCTIONAL SAFETY WEEK 2024  
Implementation of ISO 26262 & SOTIF**

15th - 18th April, 2024 | Berlin, Germany

[Download Agenda](#) [Partnership Opportunities](#) [Register to Attend](#)

Home Agenda ▾ Speakers Partners ▾ \*NEW\* - Gala Dinner Content Hub Why Attend? ▾ Venue Register [✉](#) [LinkedIn](#) #FUSAWeek2024

**HIGHEST SAVING ON ATTENDEE REGISTRATION ENDS  
16TH FEBRUARY 2024**

Secure Your Place at the Lowest Rate!  
OEM & Group Discounts available - Contact [enquire@automotive-iq.com](mailto:enquire@automotive-iq.com).

[Register to Attend](#)

Get Ready for a One-of-a-Kind Experience at Automotive IQ's **Functional Safety Week 2024!**



**4 Full Days**  
Four Days of Technical Content & Networking



**45+ Expert Speakers**  
Including 26262 Club Committee Members & Experts  
From Leading OEM, Tier-1 and Tier-2 Companies



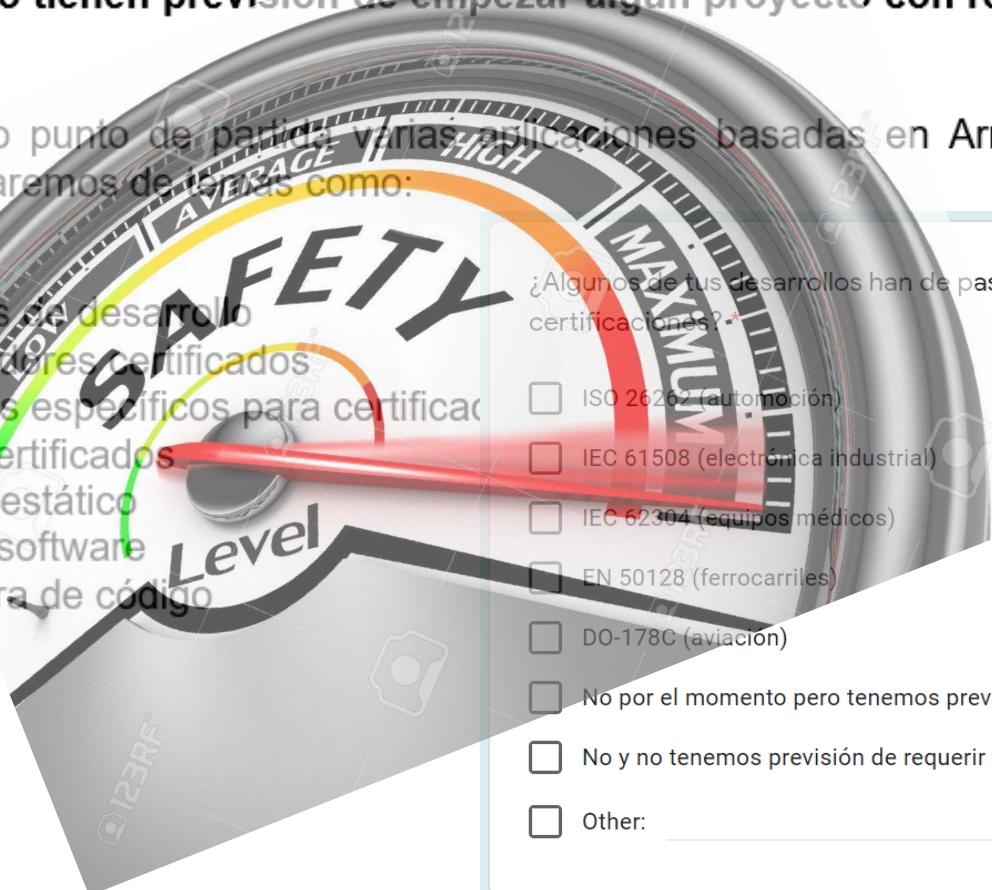
**220+ Attendees**  
From Across Europe, Asia & US

# Desarrollo de aplicaciones “confiables”

El seminario está enfocado tanto para aquellos que ya certifican o siguen esta metodología como para aquellos que aún no lo hacen pero que tienen interés en mejorar sus procesos o tienen previsión de empezar algún proyecto con requisitos de functional safety.

Tomando como punto de partida varias aplicaciones basadas en Arm Cortex-M, durante el seminario hablaremos de temas como:

- Entornos de desarrollo
- Compiladores certificados
- Paquetes específicos para certificación
- RTC5 certificados
- Análisis estático
- Test de software
- Cobertura de código



<http://www2.keil.com/fusa-rts>

[http://www2.keil.com/mdk5/safety?utm\\_campaign=arm\\_seminarios-safety-2020&utm\\_medium=email&utm\\_source=acumbamail](http://www2.keil.com/mdk5/safety?utm_campaign=arm_seminarios-safety-2020&utm_medium=email&utm_source=acumbamail)

<http://www2.keil.com/mdk5/verification-tools#lint>

<http://www2.keil.com/mdk5/debug/coverage>

# Desarrollo de aplicaciones “confiables”

## Niveles de integridad de seguridad

Probability	Mapping of Risk to SIL	Severity of Consequence				
		Insignificant	Minor	Severe	Major	Catastrophic
Rare	-	-	SIL1	SIL2	SIL3	
Unlikely	-	SIL1	SIL2	SIL3	SIL4	
Likely	SIL1	SIL2	SIL3	SIL4	x	
Very Likely	SIL2	SIL3	SIL4	x	x	
Certain	SIL3	SIL4	x	x	x	



arm

**CAPEL**  
CAPTURA ELECTRÓNICA

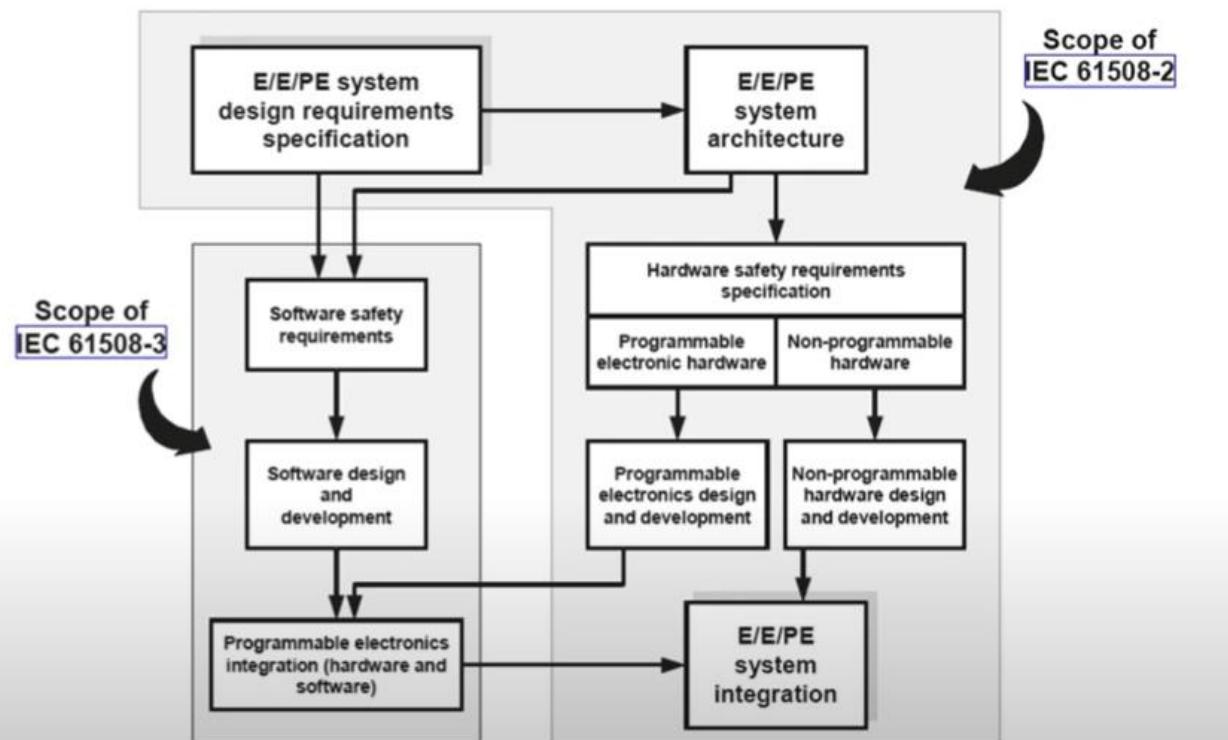
# Desarrollo de aplicaciones “confiables”

## E/E/PE Safety System Lifecycle

Search with your voice

Cada normativa define requisitos para:

- Procesos
- Documentación
- Hardware
- Software
- Herramientas
- ...



arm

CAPEL  
CAPTURA ELECTRÓNICA

Estrategias y herramientas “más a mano” que podemos utilizar para desarrollar aplicaciones más o menos confiables:

## Estrategias:

- Buenas prácticas de desarrollo de código (uso de librerías, código estructurado, cumplimiento de guías de desarrollo, etc.)
- Uso de herramientas de depuración 
- Uso de herramientas de control de versiones
- Plan de test de los módulos independientes
- Test de código (code coverage?)
- ...

Qué herramientas nos permiten determinar un fallo “no evidente” en el sistema:

Durante el desarrollo de la aplicación:

- Plan de test de cada módulo desarrollado
  - Análisis de especificaciones → funcionalidades a probar
  - Definir estrategia y tipos de pruebas. Criterios de aceptación
- Pruebas de integración
- Uso intensivo de herramientas de depuración
- Consulta de excepciones del sistema (¿?)
- ...

Durante la “vida útil” de la aplicación:

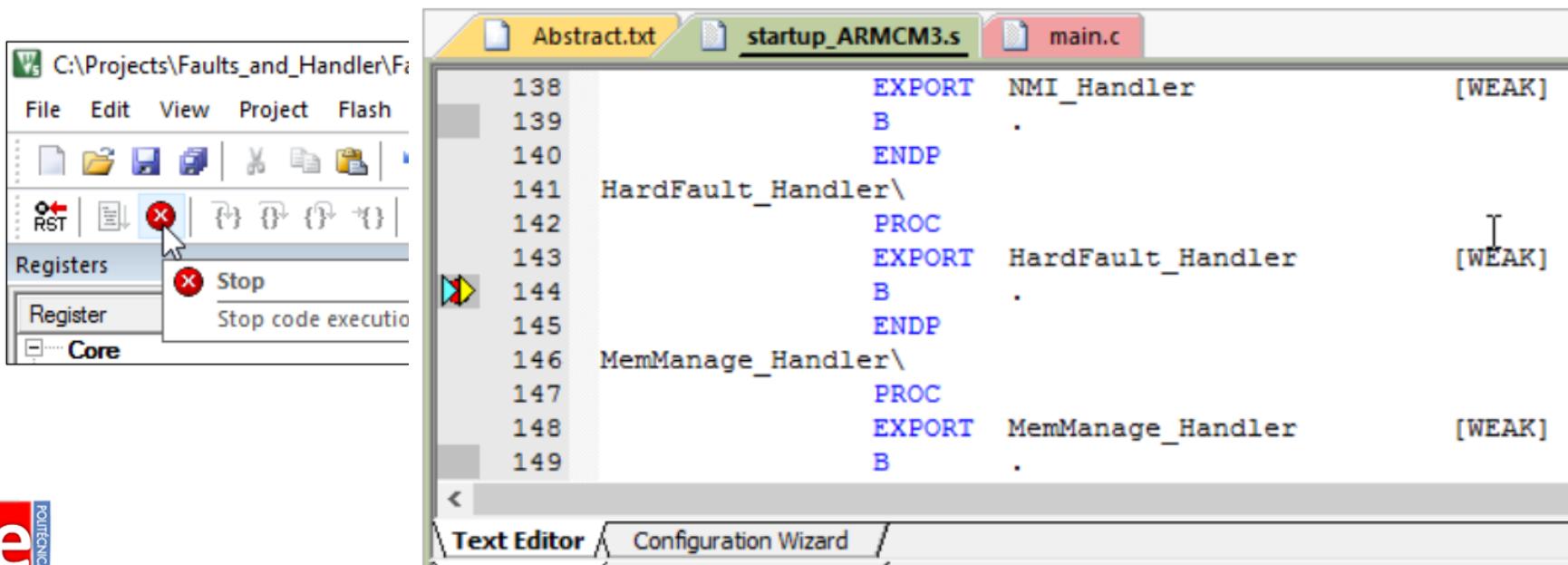
- Watchdog (Capítulos 21 y 22 del Reference Manual)
- Consulta de causa de reset del sistema (¿?)
- ...

# Fault Exceptions

Cuando una aplicación “se cuelga”, es probable que se haya producido una *Fault Exception*.

Parar la ejecución del programa en el depurador y tratar de ver el tipo de excepción (y su causa) puede dar pistas del problema.

<http://www.keil.com/appnotes/files/apnt209.pdf>

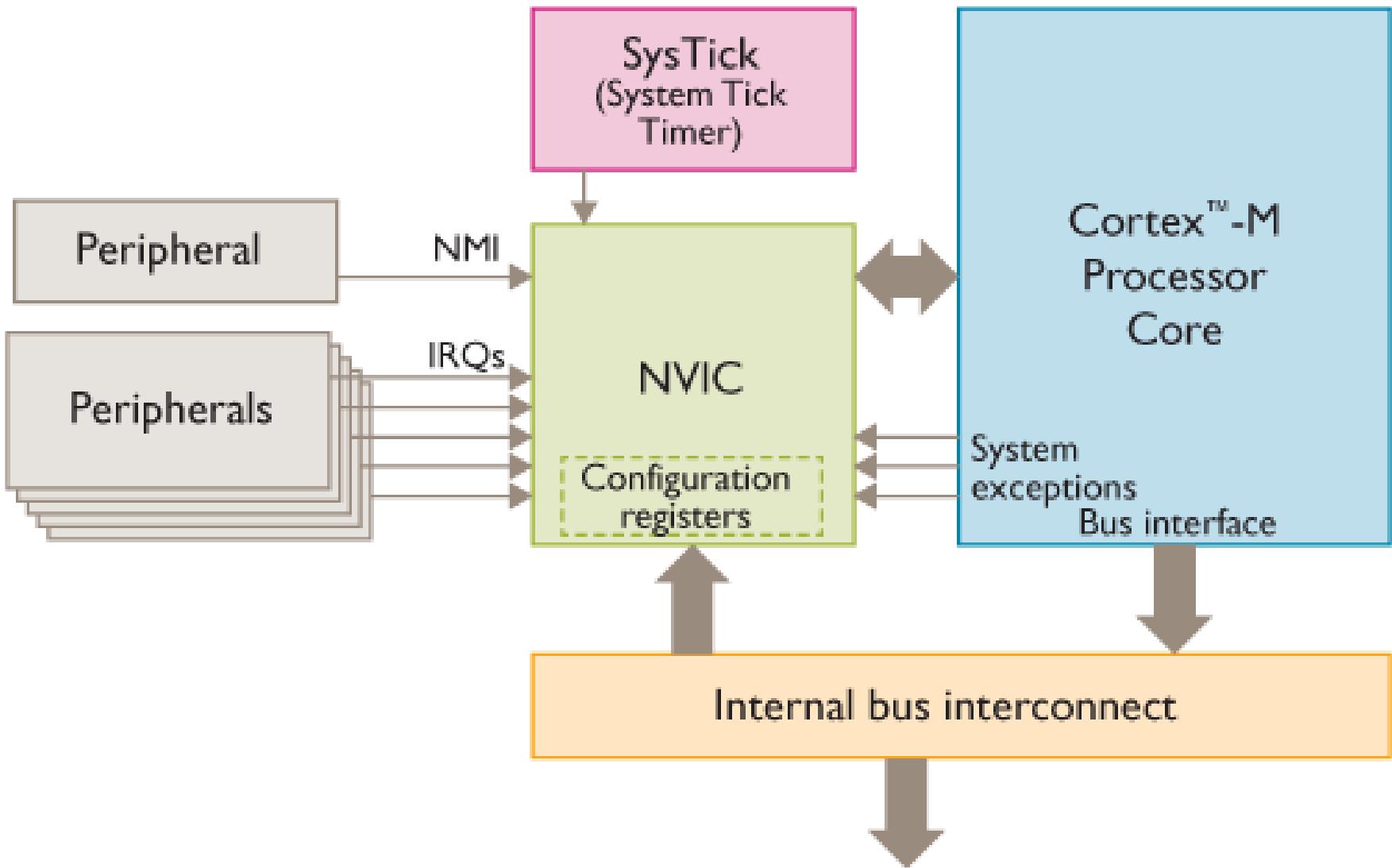


# *Excepciones*

---

- What is an exception?
  - Any event (internal or external) that stops µP current process to switch to different task.
  - The exception priority is used to manage when it is attended by the µP, and to nest them.
  
- In Cortex-M
  - **System Exceptions** are numbered 1–15.
  - **External Interrupt** (to the µP, known as Interrupt Requests, IRQs) are numbered 16 and above.
  - Most of the exceptions have programmable priority, and a few have fixed priority.

# *NVIC (Nested Vector Interrupt Controller)*



# Excepciones

Exception Type	Name	Priority (Level Address)	Enable
1	Reset	-3	Always
2	NMI	-2	Always
3	Hard fault	-1	Always
4	MemManage	Programmable (0xE000ED18)	NVIC SHCSR (0xE000ED24) bit[16]
5	BusFault	Programmable (0xE000ED19)	NVIC SHCSR (0xE000ED24) bit[17]
6	Usage fault	Programmable (0xE000ED1A)	NVIC SHCSR (0xE000ED24) bit[18]
7-10	-	-	-
11	SVC	Programmable (0xE000ED1F)	Always
12	Debug monitor	Programmable (0xE000ED20)	NVIC DEMCR (0xE000EDFC) bit[16]
13	-	-	-
14	PendSV	Programmable (0xE000ED22)	Always
15	SysTick	Programmable (0xE000ED23)	SYSTICK CTRLSTAT (0xE000E010) bit[1]
16-255	IRQ	Programmable (0xE000E400)	NVIC SETEN (0xE000E100)

Fault exceptions

## □ Hard Fault Exception

- ✓ Usage faults, bus faults, and memory management faults if their handler cannot be executed
- ✓ A bus fault during vector fetch
- ✓ Hard Fault Status Register (HFSR) is used in the exception handler to determine the cause of the fault

## □ Bus Fault Exception

- ✓ Bus faults are produced when an error response is received during a transfer on the AHB interfaces
- ✓ Bus Fault Status Register (BFSR) is used in the exception handler to determine the cause of the fault

## □ Memory Management Fault Exception

- ✓ Memory Management faults, as access to memory regions not defined in MPU setup, execute code from non executable memory regions, writing to read-only regions, ...
- ✓ Memory Management Fault Status Register (MMFSR) is used in the exception handler to determine the cause of the fault

## □ Usage Fault Exception

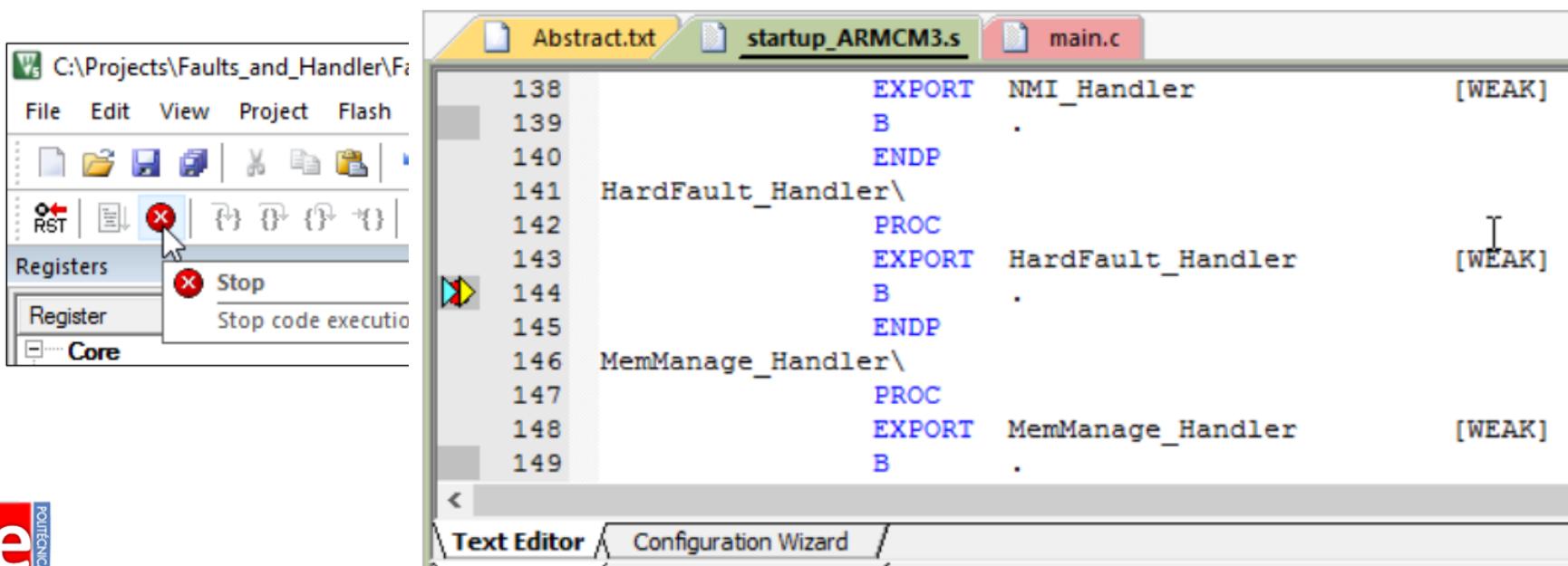
- ✓ Usage faults can be caused by undefined instructions, invalid interrupt return, divide by zero, etc.
- ✓ Usage Fault Status Register (UFSR) is used in the exception handler to determine the cause of the fault

# Fault Exceptions

Cuando una aplicación “se cuelga”, es probable que se haya producido una *Fault Exception*.

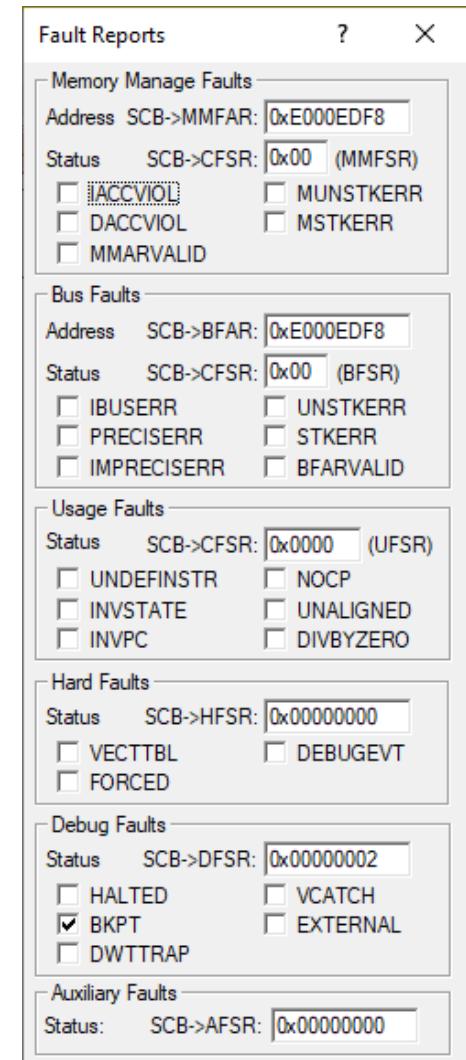
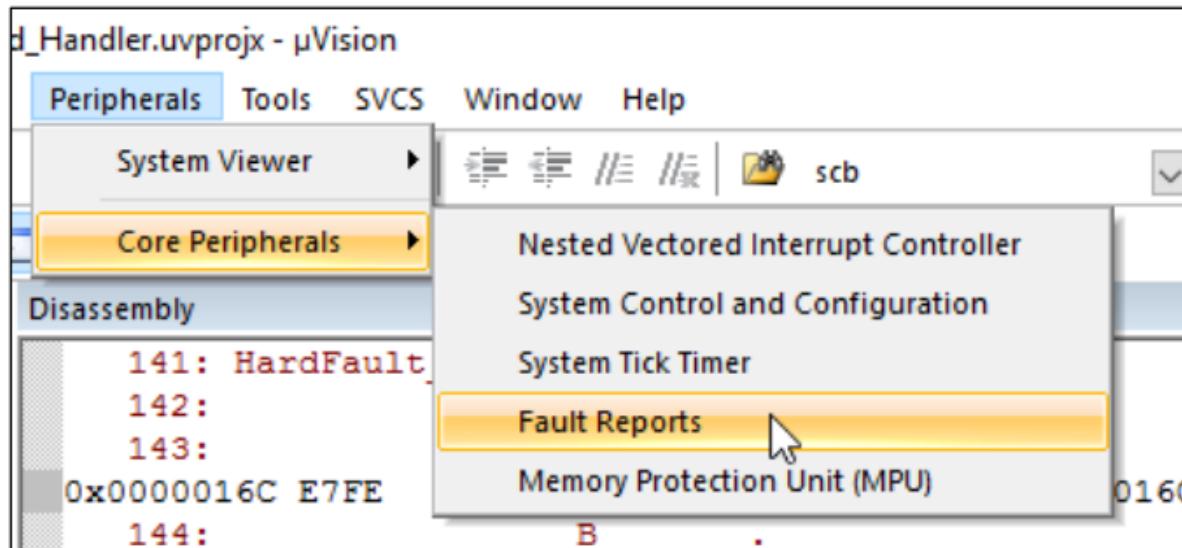
Parar la ejecución del programa en el depurador y tratar de ver el tipo de excepción (y su causa) puede dar pistas del problema.

<http://www.keil.com/appnotes/files/apnt209.pdf>

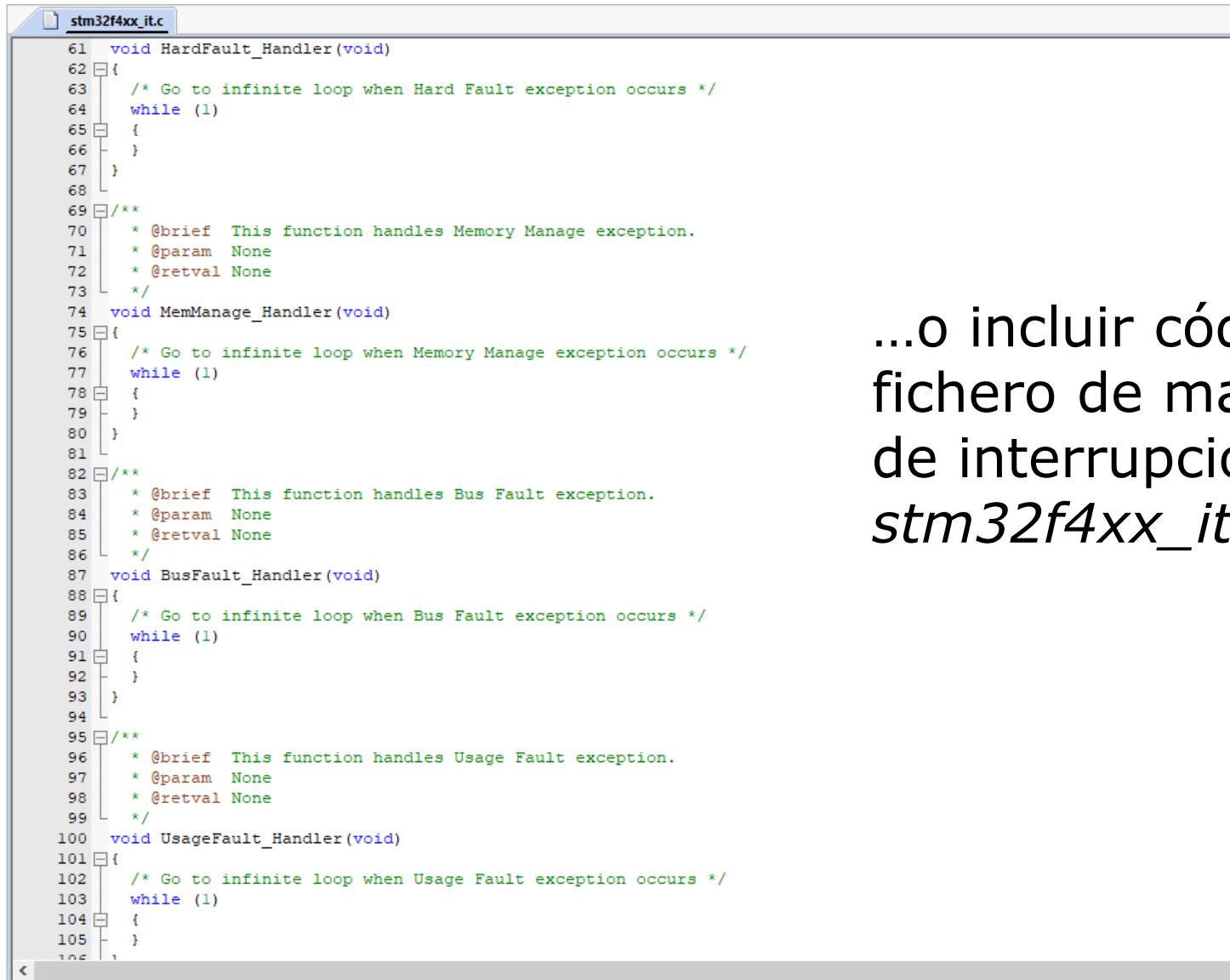


# Fault Exceptions

También es útil revisar la información del Fault Reports.



# Fault Exceptions

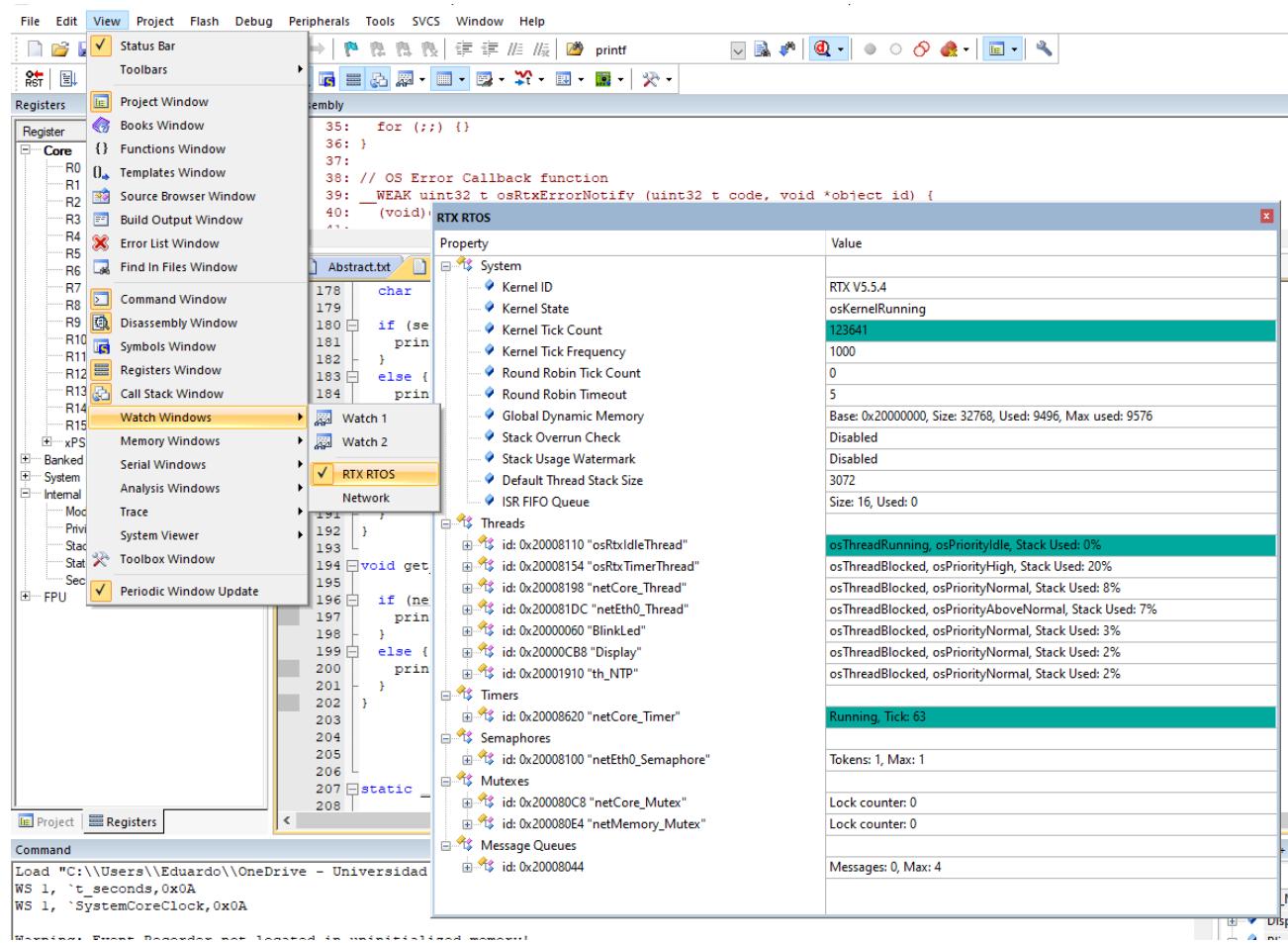


```
stm32f4xx_it.c
61 void HardFault_Handler(void)
62 {
63     /* Go to infinite loop when Hard Fault exception occurs */
64     while (1)
65     {
66     }
67 }
68
69 /**
70  * @brief This function handles Memory Manage exception.
71  * @param None
72  * @retval None
73 */
74 void MemManage_Handler(void)
75 {
76     /* Go to infinite loop when Memory Manage exception occurs */
77     while (1)
78     {
79     }
80 }
81
82 /**
83  * @brief This function handles Bus Fault exception.
84  * @param None
85  * @retval None
86 */
87 void BusFault_Handler(void)
88 {
89     /* Go to infinite loop when Bus Fault exception occurs */
90     while (1)
91     {
92     }
93 }
94
95 /**
96  * @brief This function handles Usage Fault exception.
97  * @param None
98  * @retval None
99 */
100 void UsageFault_Handler(void)
101 {
102     /* Go to infinite loop when Usage Fault exception occurs */
103     while (1)
104     {
105     }
106 }
```

...o incluir código en el  
fichero de manejadores  
de interrupciones  
*stm32f4xx\_it.c*.

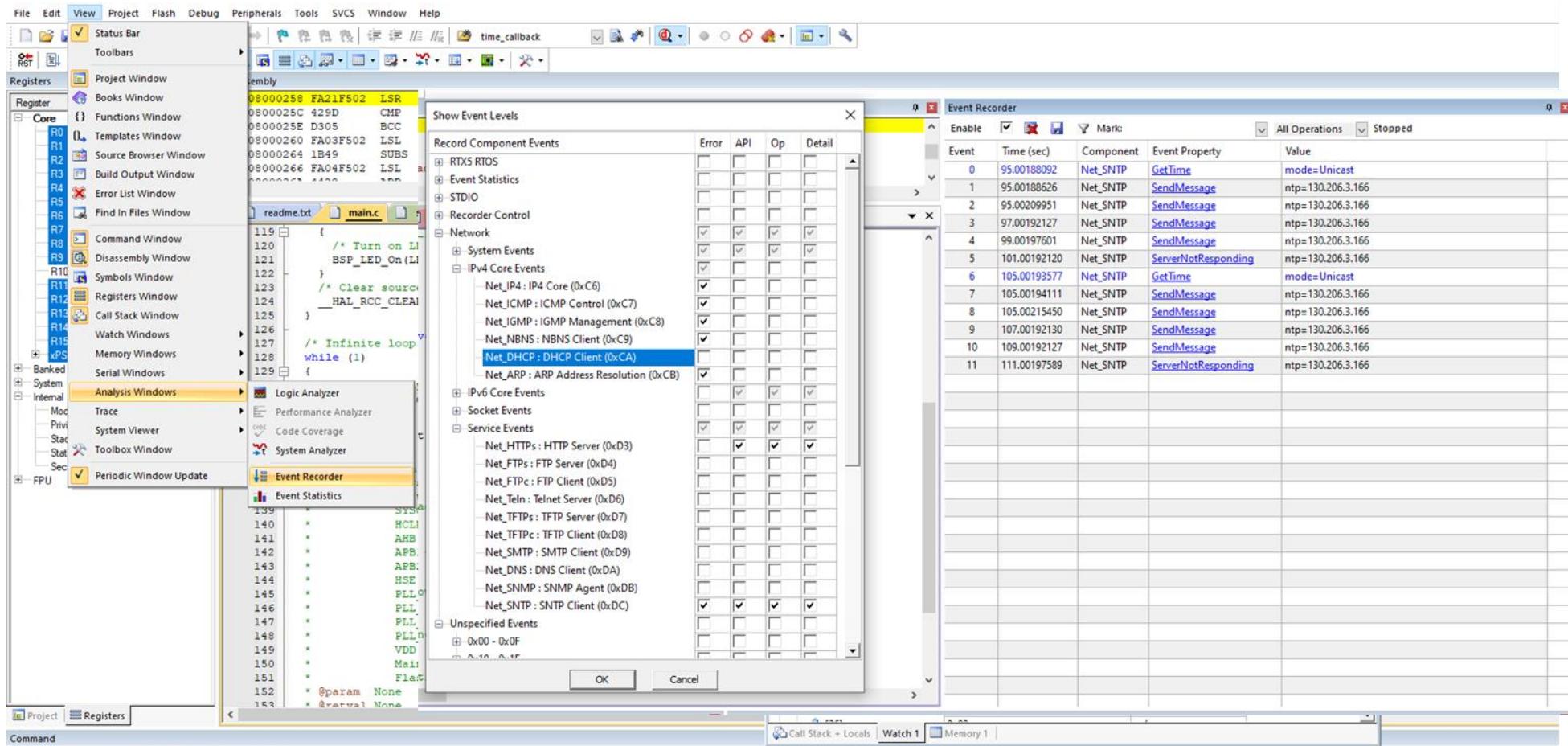
En ocasiones, el fallo se produce por una mala utilización de los recursos que proporciona el Sistema Operativo.

Utilizar las herramientas del entorno puede ayudar a detectar el problema



# Event recorder

The software component **Event Recorder** provides an API (function calls) for event annotations in the application code or software component libraries. These API functions record event timing and data information while the program is executing.



# Event recorder setup

Manage Run-Time Environment

Software Component	Sel.	Variant	Version	Description
Board Support		32F469IDISCOVERY	1.0.0	<a href="#">STMicroelectronics 32F469IDISCOVERY Kit</a>
CMSIS				<a href="#">Cortex Microcontroller Software Interface Components</a>
CMSIS Driver				<a href="#">Unified Device Drivers compliant to CMSIS-Driver Specifications</a>
Compiler		ARM Compiler	1.6.0	<a href="#">Compiler Extensions for Cortex-M</a>
Event Recorder	<input checked="" type="checkbox"/>	DAP	1.4.0	<a href="#">Event Recording and Configuration</a>
I/O				<a href="#">Retarget Input/Output</a>
Device				<a href="#">Startup, System Setup</a>
File System		MDK-Plus	6.14.1	<a href="#">File Access on various targets</a>
Graphics		MDK-Plus	6.16.3	<a href="#">User Interface on graphical displays</a>
Graphics Display				<a href="#">Display Interface including OpenGL ES 2.0</a>
Network		MDK-Driver	7.15.0	<a href="#">Intra/Interfacing Networking</a>

Cortex-M Target Driver Setup

Debug | Trace | Flash Download | Pack

Core Clock: 168.000000 MHz  iTrace Enable

Trace Clock: 168.000000 MHz  Use Core Clock

Trace Port

Serial Wire Output - UART/NRZ

SWO Clock Prescaler: 14  Configure target  Set max

SWO Clock: 12.000000 MHz

Timestamps

Enable Prescaler: 1

Trace Events

CPI: Cycles per Instruction  
 EXC: Exception overhead  
 SLEEP: Sleep Cycles  
 LSU: Load Store Unit Cycles  
 FOLD: Folded Instructions  
 EXCTRC: Exception Tracing

PC Sampling

Prescaler: 1024\*16  Periodic Period: <Disabled>  
 on Data R/W Sample

ITM Stimulus Ports

Enable: 0xFFFFFFFF	31	Port	24	23	Port	16	15	Port	8	7	Port	0
Privilege: 0x00000008			Port 31..24	<input checked="" type="checkbox"/>	Port 23..16	<input type="checkbox"/>	Port 15..8	<input type="checkbox"/>	Port 7..0	<input type="checkbox"/>		

Advanced settings

Ignore packets with no SYNC  
 Overwrite CYCCNT

OK Cancel Apply

RTX\_Config.h

Expand All Collapse All Help Show Grid

Option

System Configuration

- Global Dynamic Memory size [bytes] 32768
- Kernel Tick Frequency [Hz] 1000
- Round-Robin Thread switching
- ISR FIFO Queue
- Object Memory usage counters

Thread Configuration

Timer Configuration

Event Flags Configuration

Mutex Configuration

Semaphore Configuration

Memory Pool Configuration

Message Queue Configuration

Event Recorder Configuration

- Global Initialization
  - Start recording
  - Global Event Filter Setup
  - RTOS Event Filter Setup
- RTOS Event Generation
  - Memory Management
  - Kernel
  - Thread
  - Generic Wait
  - Thread Flags
  - Event Flags
  - Timer
  - Mutex
  - Semaphore
  - Memory Pool

RTOS Event Filter Setup

Recording levels for RTX components.  
Only applicable if events for the respective component are generated.

dte POLITECNICA

Text Editor Configuration Wizard

44

# I/O Retargeting. Debug (printf) Viewer

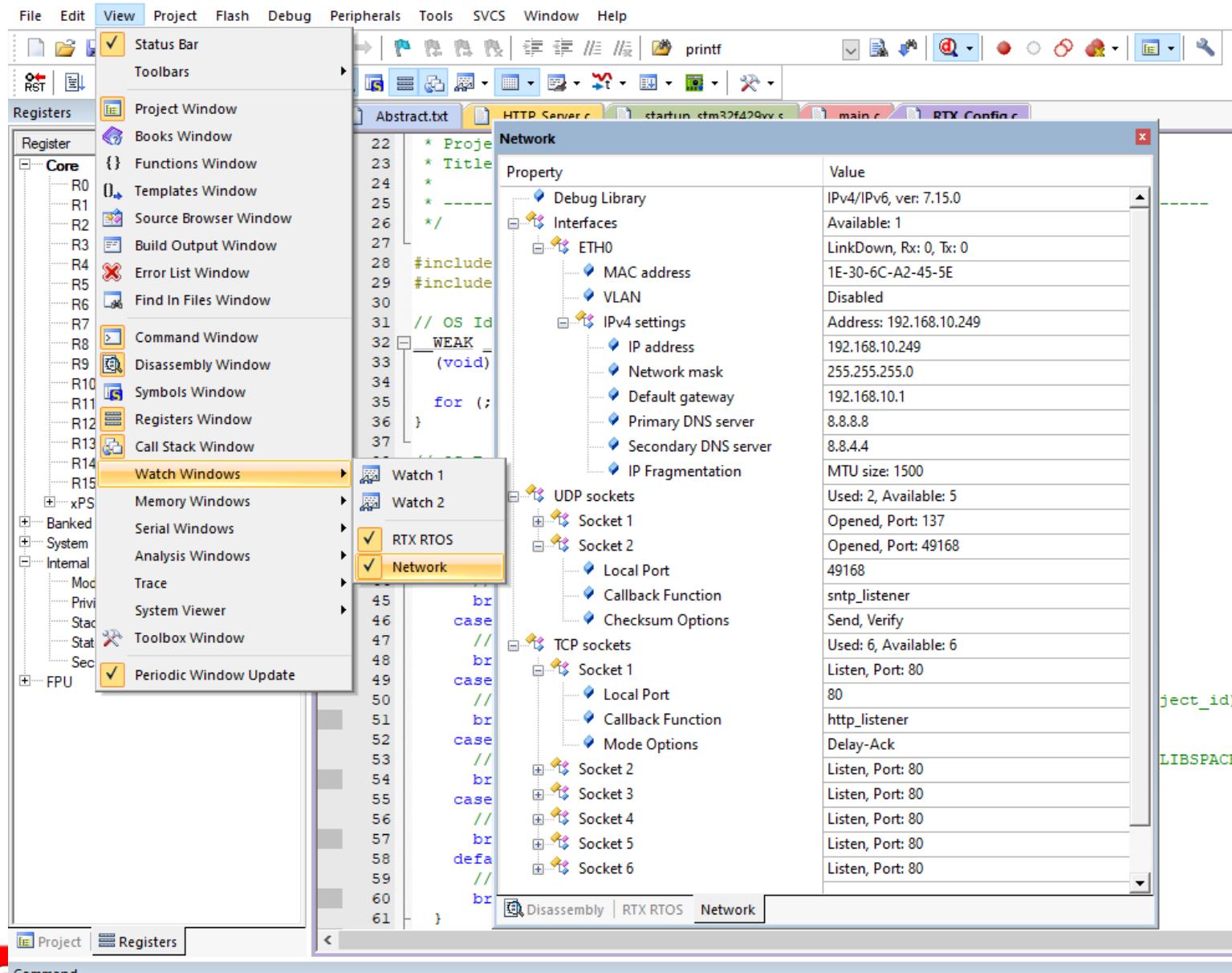
Software Component	Sel.	Variant	Version	Description
Board Support		32F469IDISCOVERY	1.0.0	STMicroelectronics 32F469IDISCOVERY Kit
CMSIS				Cortex Microcontroller Software Interface Components
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
Compiler		ARM Compiler	1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
Event Recorder	<input checked="" type="checkbox"/>	DAP	1.4.0	Event Recording and Component Viewer via Debug Access Port (DAP)
I/O				Retarget Input/Output
File	<input type="checkbox"/>	File System	1.2.0	Use retargeting together with the File System component
STDERR	<input type="checkbox"/>	Breakpoint	1.2.0	Stop program execution at a breakpoint when using STDERR
STDIN	<input type="checkbox"/>	Breakpoint	1.2.0	Stop program execution at a breakpoint when using STDIN
STDOUT	<input checked="" type="checkbox"/>	ITM	1.2.0	Redirect STDOUT to a debug output window using ITM
TTY	<input type="checkbox"/>	Breakpoint	1.2.0	Stop program execution at a breakpoint when using TTY

The screenshot shows the ST-Connect IDE interface. The top window is titled "Manage Run-Time Environment" and displays the "I/O" configuration. The "STDOUT" row is selected and highlighted with a red box, indicating it is set to "ITM". The main workspace shows a project with files like "HTTP\_Server.c", "main.c", and "RTX\_Config.c". The assembly view shows a snippet of code with a yellow highlight. The bottom right window is a "Debug (printf) Viewer" showing log entries:

```
SNTP request sent.  
Server not responding or bad response.  
SNTP request sent.  
Server not responding or bad response.  
SNTP request sent.  
Server not responding or bad response.  
SNTP request sent.  
Server not responding or bad response.
```

The bottom left corner features the dte logo.

# Network Watch Window

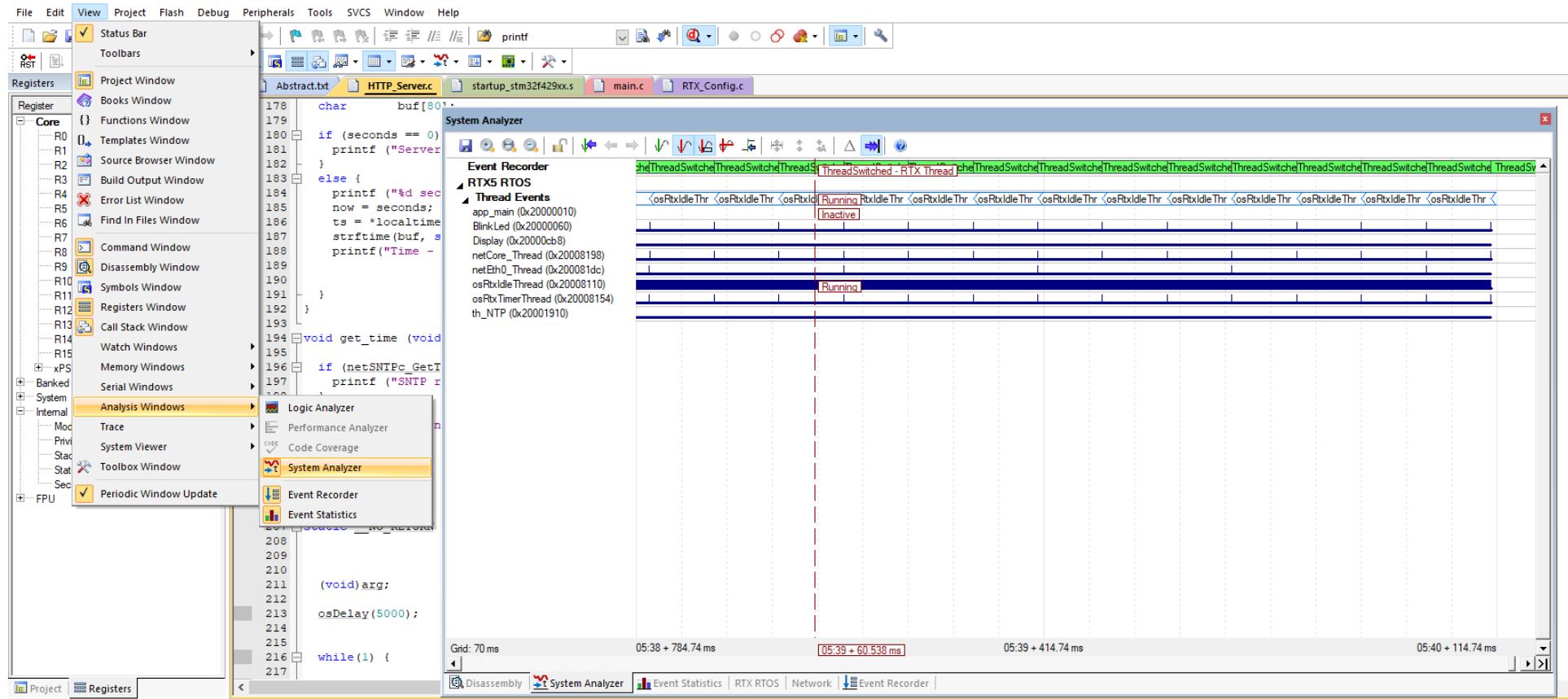


# Event statistics

The screenshot shows a software development environment with a menu bar (File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help) and a toolbar with various icons. The left sidebar contains a tree view of windows: Status Bar (checked), Toolbars, Registers, Project Window, Books Window, Functions Window, Templates Window, Source Browser Window, Build Output Window, Error List Window, Find In Files Window, Command Window, Disassembly Window, Symbols Window, Registers Window, Call Stack Window, Watch Windows, Memory Windows, Serial Windows, Analysis Windows (checked), Logic Analyzer, Trace, System Viewer, Toolbox Window, and FPU. A checkbox for Periodic Window Update is also checked. The main area shows a code editor with the file `HTTP_Server.c` open, displaying C code related to a server and NTP. Below the code editor is a tab bar with Disassembly, Event Statistics (checked), RTX RTOS, Network, Event Recorder, and Call Stack + Locals. An `Event Statistics` window is open, showing a table of events and their execution timing. The table has columns for Source, Count, and Filter Enable / Execution Timing.

Source	Count	Filter Enable / Execution Timing
Event Start/Stop Group A - ...		
Event Start/Stop Group B - ...		
Event Start/Stop Group C - ...		
Event Start/Stop Group D - ...		
RTX5 RTOS		
Thread Events		
app_main	6	Running: [0.00%] min=3.9167 us, max=6.1488 us, avg=5.0327 us
osRtxIdleThread	929 (+1)	Running: [99.95%] min=99.495 ms, max=1.098e+11 s, avg=232.61 ms
osRtxTimerThread	930	Running: [0.01%] min=8.4583 us, max=15.5 us, avg=15.385 us
"netEth0_Thread"	619	Running: [0.01%] min=6.4048 us, max=55.506 us, avg=36.441 us
"netCore_Thread"	930	Running: [0.02%] min=17.661 us, max=407.48 us, avg=57.86 us
BlinkLed	1542	Running: [0.01%] min=6.8631 us, max=17.827 us, avg=11.364 us
Display	3	Running: [0.00%] min=23.637 us, max=23.643 us, avg=23.639 us
th_NTP	19	Running: [0.00%] min=6.5833 us, max=176.87 us, avg=91.303 us

# System Analyzer



Qué herramientas nos permiten determinar un fallo “no evidente” en el sistema:

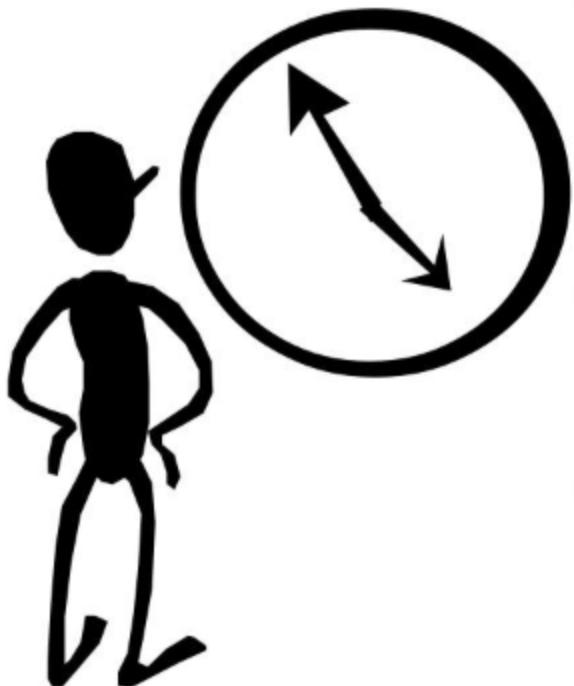
Durante el desarrollo de la aplicación:

- Plan de test de cada módulo desarrollado
  - Análisis de especificaciones → funcionalidades a probar
  - Definir estrategia y tipos de pruebas. Criterios de aceptación
- Pruebas de integración
- Uso intensivo de herramientas de depuración
- Consulta de excepciones del sistema (¿?)
- ...

Durante la “vida útil” de la aplicación:

- Watchdog (Capítulos 21 y 22 del Reference Manual)
- Consulta de causa de reset del sistema (¿?)
- ...

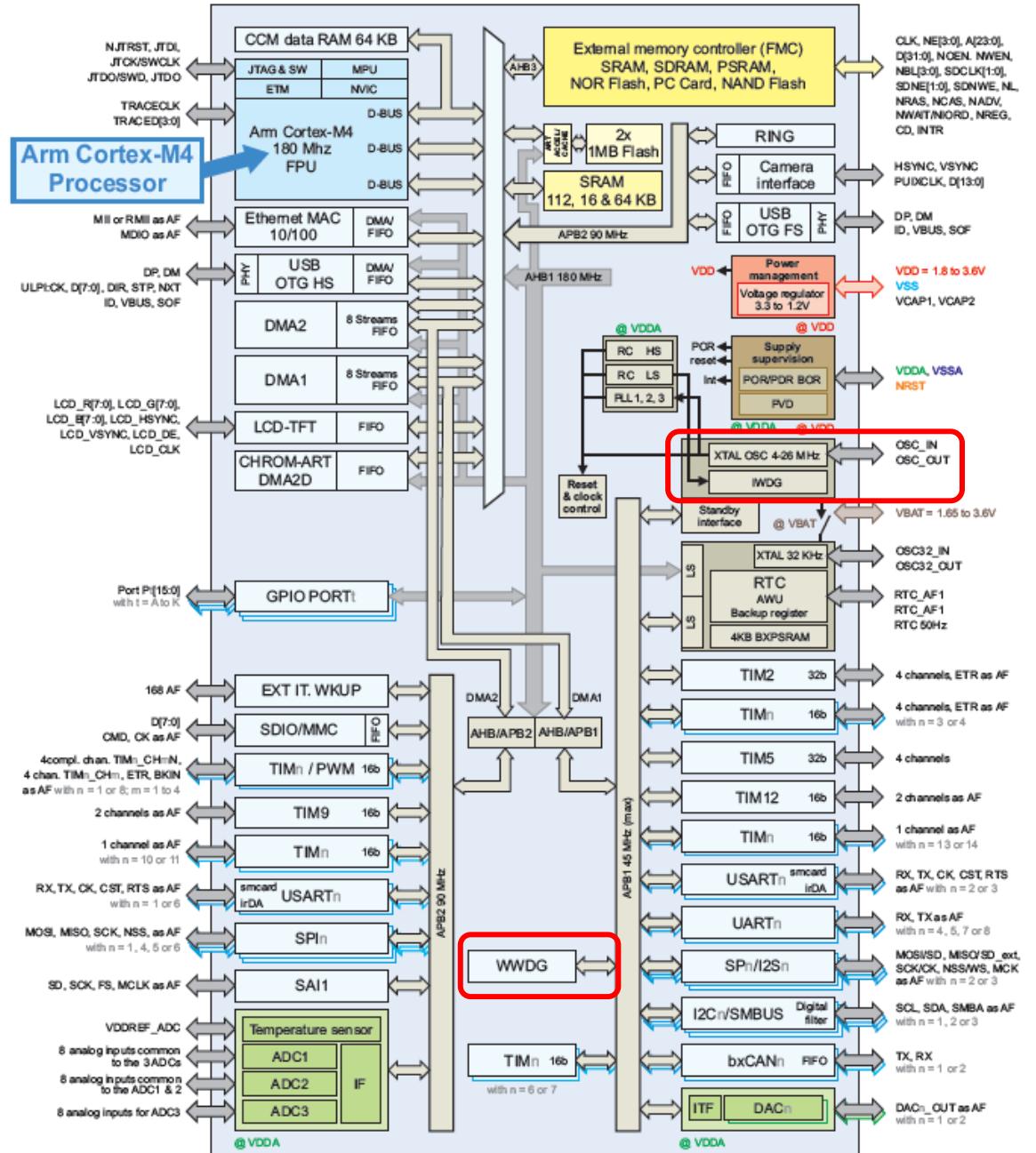
## What is the watchdog timer?



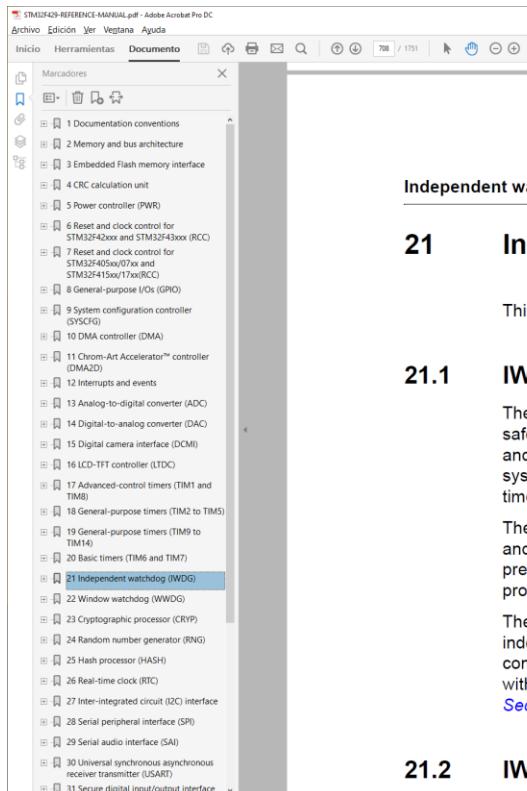
- A special timer that, once configured and enabled, counts down to 0
- When expired, causes the CPU to reset
- The countdown can be restarted at any time before expiration
- If periodically restarted, the expiration will never occur, and the reset will never happen.



Timer/s que “vigilan” el sistema. Ante un comportamiento “anormal”, puede provocar bien una **interrupción**, bien un **reset del sistema**.



# Watchdogs en STM32F4



## Independent watchdog (IWDG)

RM0090

## 21 Independent watchdog (IWDG)

This section applies to the whole STM32F4xx family, unless otherwise specified.

### 21.1 IWDG introduction

The devices have two embedded watchdog peripherals which offer a combination of high safety level, timing accuracy and flexibility of use. Both watchdog peripherals (Independent and Window) serve to detect and resolve malfunctions due to software failure, and to trigger system reset or an interrupt (window watchdog only) when the counter reaches a given timeout value.

The independent watchdog is based on a free-running prescaler programed by the user.

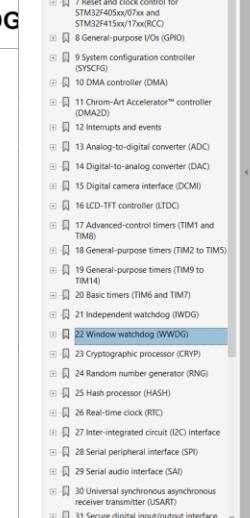
The IWDG has a 16-bit downcounter. It is independent from the window watchdog, and it is constrained by the user-defined prescaler programed within a section of the memory.

## 21.2 IWDG

RM0090

## Window watchdog (WWDG)

This section applies to the whole STM32F4xx family, unless otherwise specified.



## 22 Window watchdog (WWDG)

This section applies to the whole STM32F4xx family, unless otherwise specified.

### 22.1 WWDG introduction

The window watchdog is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the contents of the downcounter before the T6 bit becomes cleared. An MCU reset is also generated if the 7-bit downcounter value (in the control register) is refreshed before the downcounter has reached the window register value. This implies that the counter must be refreshed in a limited window.

### 22.2 WWDG main features

- Programmable free-running downcounter
- Conditional reset
  - Reset (if watchdog activated) when the downcounter value becomes less than 0x40
  - Reset (if watchdog activated) if the downcounter is reloaded outside the window

➤ **Práctica 3:** Modos de bajo consumo y Watchdog

- Apartado 4 (opcional) Watchdog e identificación de causa de reset del sistema

❖ 1 semana

# *Ingeniería de Sistemas Electrónicos*

## *Programación avanzada de microcontroladores (II)*

- 
- Integración de referencia temporal en el sistema
  - Desarrollo de aplicaciones “confiables”
  - **Reducción del consumo**
  - Desarrollo de software. Control de versiones
  - Prácticas de laboratorio

## 5 Power controller (PWR)

### 5.3 Low-power modes

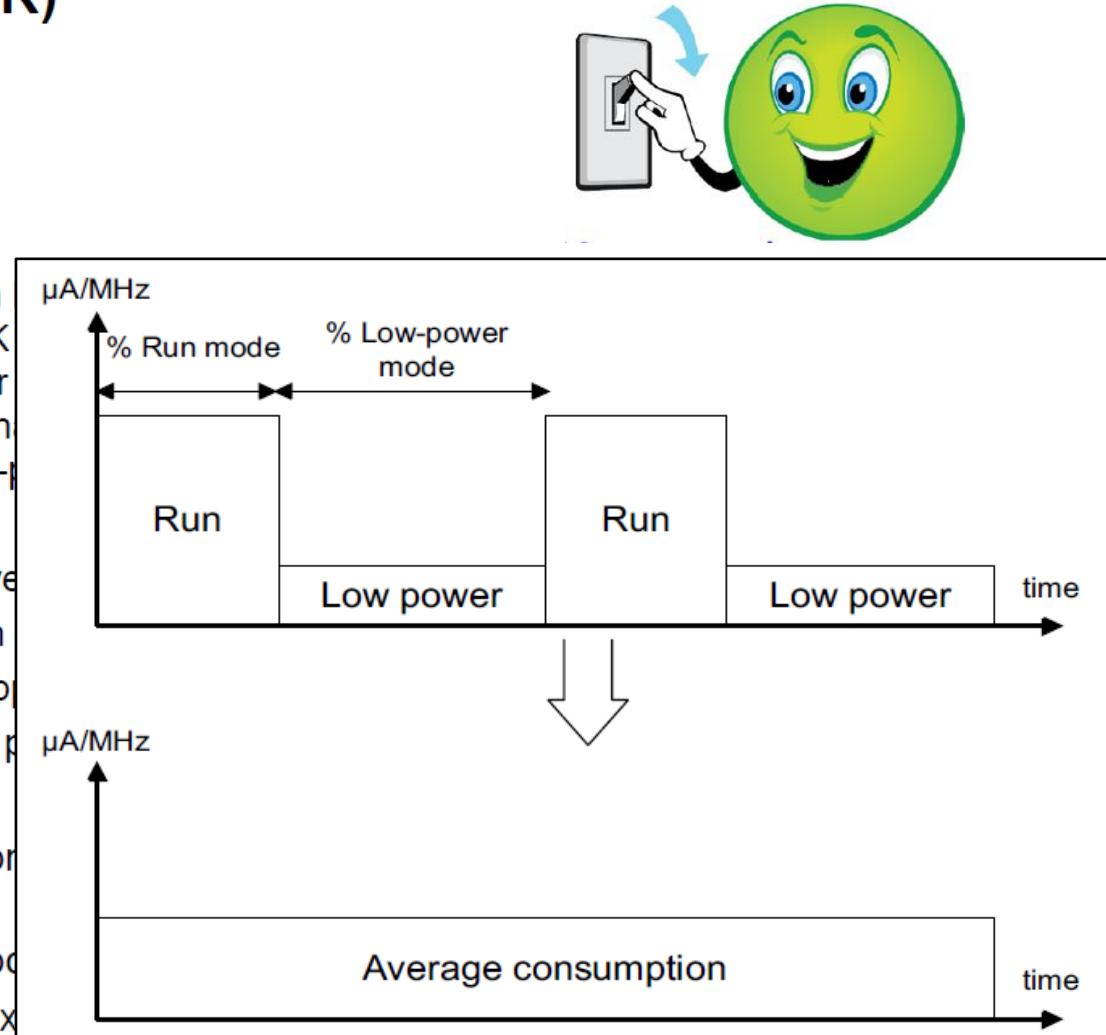
By default, the microcontroller is in mode the CPU is clocked by HCLK modes are available to save power example when waiting for an external event or a timer interrupt, the best compromise between low-power consumption and fast wakeup sources.

The devices feature three low-power modes:

- Sleep mode (Cortex<sup>®</sup>-M4 with HSE oscillator)
- Stop mode (all clocks are stopped)
- Standby mode (1.2 V domain powered)

In addition, the power consumption can be reduced by other means:

- Slowing down the system clock
- Gating the clocks to the APBx



# Modos de bajo consumo en STM32F4



## AN4365 Application note

Using STM32F4 MCU power modes with best dynamic efficiency

[https://www.st.com/resource/en/application\\_note/an4365-using-stm32f4-mcu-power-modes-with-best-dynamic-efficiency-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an4365-using-stm32f4-mcu-power-modes-with-best-dynamic-efficiency-stmicroelectronics.pdf)

STM32F4 devices feature four main low-power modes:

- **Sleep mode:**

- Only the CPU clock is stopped.

The Cortex-M4 clock is stopped and the peripherals are kept running. The current consumption increases with the clock frequency. As in Run mode, the user should be aware of system configuration rules that concern the system clock and voltage regulator scales.

- **Stop mode:**

- Lowest power consumption while all the SRAM and registers are kept.
  - PLL, HSI, HSE are disabled.
  - All clocks in 1.2 V domain are switched-off.
  - Voltage regulator is working in Normal mode or Low-power mode.
  - Flash memory is working in Stop mode or Deep-power down mode.

The Cortex-M4 core is stopped and the clocks are switched off (the PLL, the HSI and the HSE are disabled). SRAM and register content are kept. All I/O pins keep the same state as the Run mode. The voltage regulator is working in Normal or Low-power mode and the Flash memory can be configured in Stop mode or Deep-power down mode to save more static power.

# *Modos de bajo consumo en STM32F4*

---

- **Standby mode:**

- The lowest power consumption.
- The 1.2 domain is powered off (regulator is disabled).
- SRAM and register contents are lost except in the backup domain.

The Cortex-M4 core is stopped and the clocks are switched off. The voltage regulator is disabled and the 1.2 V domain is powered-off. SRAM and register contents are lost except for registers in the Backup domain (RTC registers, RTC backup register and backup SRAM), and Standby circuitry. This mode has the lowest current consumption, which depends on the configuration of both backup SRAM (not available for STM32F401x) and RTC.

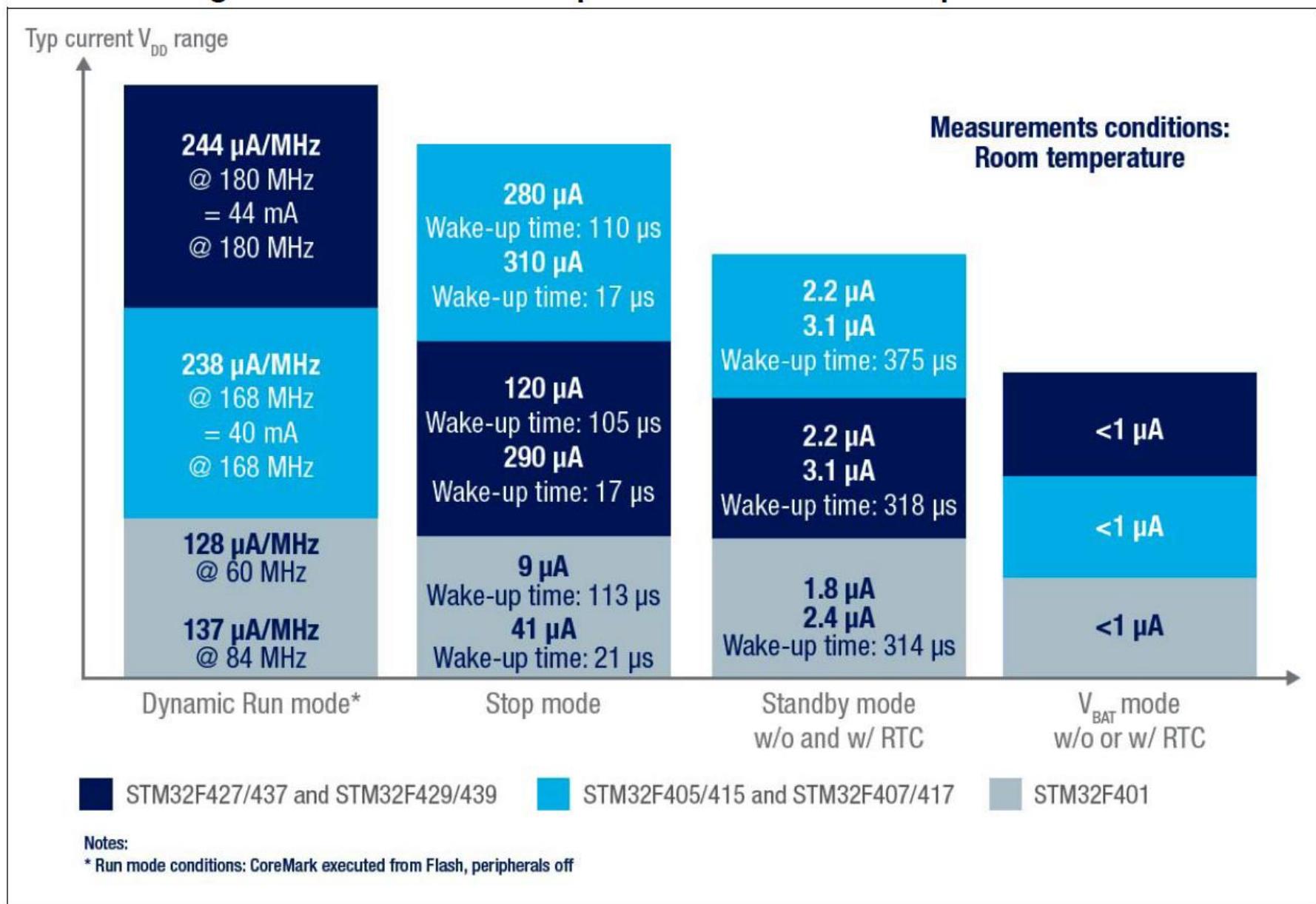
- **$V_{BAT}$  mode:**

- The main digital supply ( $V_{DD}$ ) is turned off.
- The circuit is supplied through  $V_{BAT}$  pin which should be connected to an external supply voltage (a battery or any other source).

This mode is used only when the main digital supply ( $V_{DD}$ ) is turned off and the  $V_{BAT}$  pin is connected to an external supply voltage (a battery or any other source). The  $V_{BAT}$  pin powers the Backup domain (RTC registers, RTC backup register and backup SRAM).

# Modos de bajo consumo en STM32F4

Figure 2. Current consumption of STM32F4 MCU power modes



# *Modos de bajo consumo. Entrada*

---

## **Entering low-power mode**

Low-power modes are entered by the MCU by executing the WFI (Wait For Interrupt), or WFE (Wait for Event) instructions, or when the SLEEPONEXIT bit in the Cortex®-M4 with FPU System Control register is set on Return from ISR.

Entering Low-power mode through WFI or WFE will be executed only if no interrupt is pending or no event is pending.

WFI (Wait For Interrupt) instruction → `_WFI();`

WFE (Wait For Exception) instruction → `_WFE();`

- El uso de las instrucciones anteriores hace que el “programa principal” se detenga a la espera de una interrupción (o una excepción)
- Útil para las denominadas “interrupt driven applications”
  
- Nota de aplicación “*AN3430. How to achieve the lowest current consumption with STM32F2xx*”

[https://www.st.com/resource/en/application\\_note/an3430-how-to-achieve-the-lowest-current-consumption-with-stm32f2xx-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an3430-how-to-achieve-the-lowest-current-consumption-with-stm32f2xx-stmicroelectronics.pdf)

# Modos de bajo consumo. Salida

## Exiting low-power mode

The MCU exits from Sleep and Stop modes low-power mode depending on the way the low-power mode was entered:

- If the WFI instruction or Return from ISR was used to enter the low-power mode, any peripheral interrupt acknowledged by the NVIC can wake up the device.
- If the WFE instruction is used to enter the low-power mode, the MCU exits the low-power mode as soon as an event occurs. The wakeup event can be generated either by:
  - NVIC IRQ interrupt:

When SEVONPEND = 0 in the Cortex®-M4 with FPU System Control register: by enabling an interrupt in the peripheral control register and in the NVIC. When the MCU resumes from WFE, the peripheral interrupt pending bit and the NVIC peripheral IRQ channel pending bit (in the NVIC interrupt clear pending register) have to be cleared. Only NVIC interrupts with sufficient priority will wakeup and interrupt the MCU.

When SEVONPEND = 1 in the Cortex®-M4 with FPU System Control register: by enabling an interrupt in the peripheral control register and optionally in the NVIC. When the MCU resumes from WFE, the peripheral interrupt pending bit and when enabled the NVIC peripheral IRQ channel pending bit (in the NVIC interrupt clear pending register) have to be cleared. All NVIC interrupts will wakeup the MCU, even the disabled ones. Only enabled NVIC interrupts with sufficient priority will wakeup and interrupt the MCU.

- Event

This is done by configuring a EXTI line in event mode. When the CPU resumes from WFE, it is not necessary to clear the EXTI peripheral interrupt pending bit or the NVIC IRQ channel pending bit as the pending bits corresponding to the event line is not set. It may be necessary to clear the interrupt flag in the peripheral.

The MCU exits from Standby low-power mode through an external reset (NRST pin), an IWDG reset, a rising edge on one of the enabled WKUPx pins or a RTC event occurs (see [Figure 237: RTC block diagram](#)).

After waking up from Standby mode, program execution restarts in the same way as after a Reset (boot pin sampling, option bytes loading, reset vector is fetched, etc.).

# Modos de bajo consumo. Resumen

Table 23. Low-power mode summary

Mode name	Entry	Wakeup	Effect on 1.2 V domain clocks	Effect on V <sub>DD</sub> domain clocks	Voltage regulator
<b>Sleep (Sleep now or Sleep-on-exit)</b>	WFI or Return from ISR	Any interrupt	CPU CLK OFF no effect on other clocks or analog clock sources	None	ON
	WFE	Wakeup event			
<b>Stop</b>	PDDS and LPDS bits + SLEEPDEEP bit + WFI, Return from ISR or WFE	Any EXTI line (configured in the EXTI registers, internal and external lines)	All 1.2 V domain clocks OFF	HSI and HSE oscillator s OFF	ON or in low-power mode (depends on <i>PWR power control register (PWR_CR)</i> for STM32F405xx/07x x and STM32F415xx/17x x and <i>PWR power control register (PWR_CR)</i> for STM32F405xx/07x x and STM32F415xx/17x x <i>PWR power control register (PWR_CR)</i> for STM32F42xxx and STM32F43xxx)
<b>Standby</b>	PDDS bit + SLEEPDEEP bit + WFI, Return from ISR or WFE	WKUP pin rising edge, RTC alarm (Alarm A or Alarm B), RTC Wakeup event, RTC tamper events, RTC time stamp event, external reset in NRST pin, IWDG reset			OFF

## *Otras recomendaciones para reducir el consumo*

---

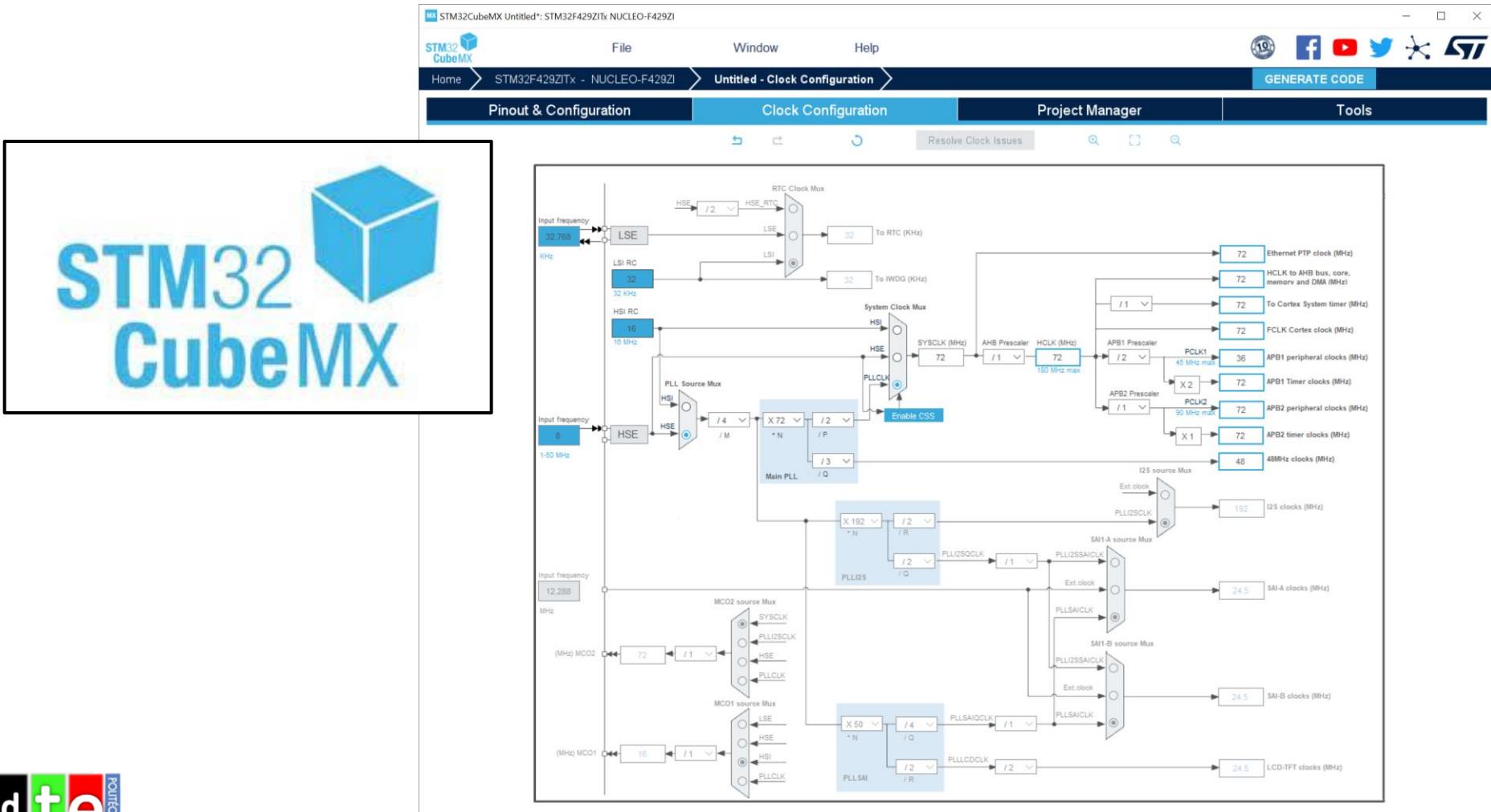
- Reducir al máximo la frecuencia de reloj del sistema
    - Algunos periféricos (Ethernet, USB, I2S and SDIO) necesitan frecuencias específicas de reloj que limitan el valor mínimo de la frecuencia de reloj del sistema
  - Reducir al máximo la frecuencia de reloj de los periféricos
  - Desactivar todos los periféricos no utilizados
  - Configurar todos los pines del GPIO no utilizados (configurados por defecto como entradas digitales) como entradas analógicas (AIN)
- 
- IMPORTANTE:
    - Cuando se hace entrar al sistema en un modo de bajo consumo no pueden utilizarse las herramientas de depuración (el core del Cortex-M4 no tiene reloj durante el modo de bajo consumo, por lo que las funciones de depuración están desactivadas)



Permitido

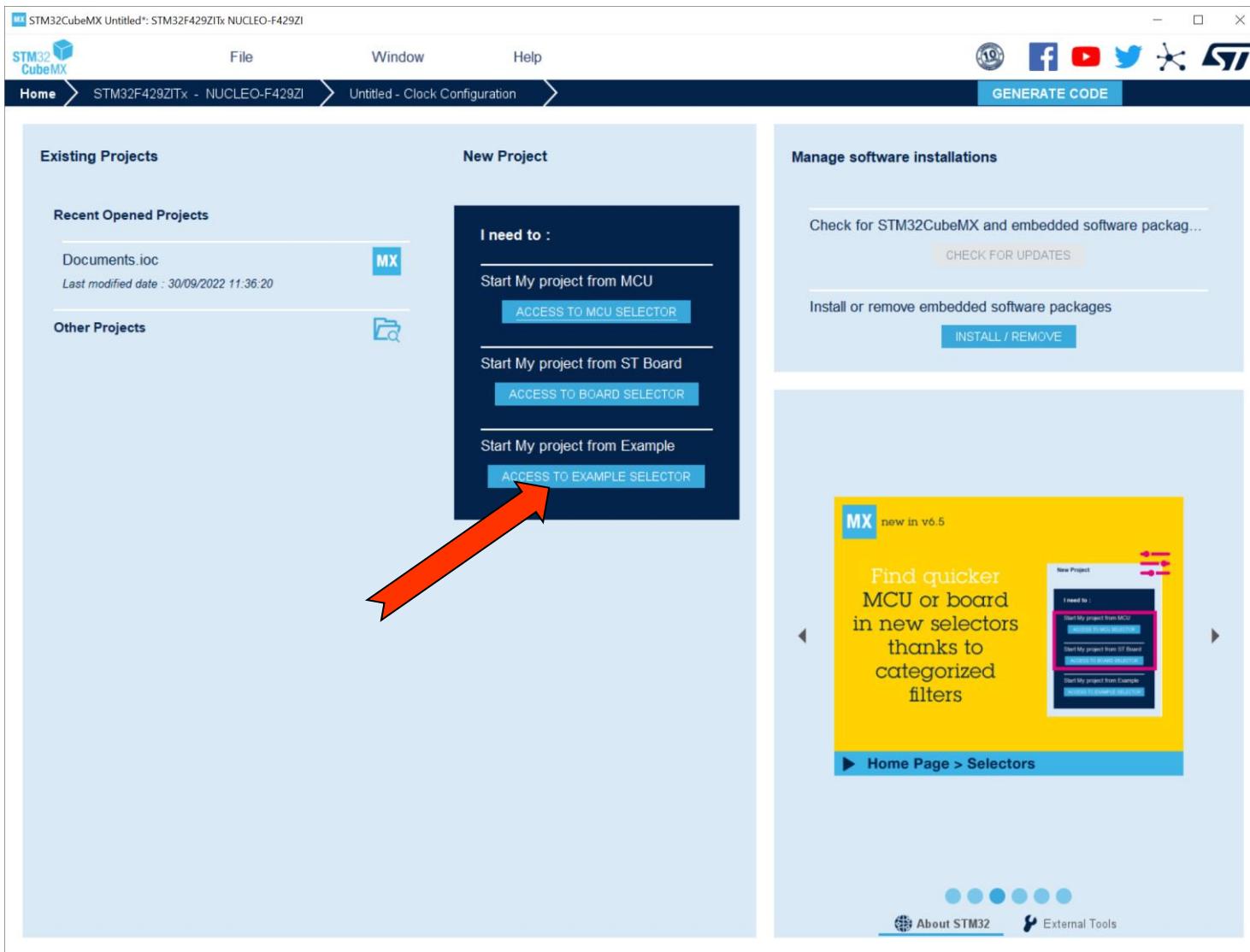
# Modos de bajo consumo. Ejemplo aplicación

- Pueden encontrarse ejemplos de aplicación en la Web de ST o en STMCubeMX



# Modos de bajo consumo. Ejemplo aplicación

- Pueden encontrarse ejemplos de aplicación en la Web de ST o en STMCubeMX



# Modos de bajo consumo. Ejemplo aplicación



Ejemplo PWR\_CurrentConsumption (STM32CubeMX)

Disponible en Moodle

A red arrow points from the text "Ejemplo PWR\_CurrentConsumption (STM32CubeMX)" to the STM32CubeMX interface. The interface shows the "Example Selector" tab selected. In the "Board" dropdown, "Name NUCLEO-F429ZI" is highlighted with a red box. In the "Examples List" table, the row for "PWR\_CurrentConsumption" is also highlighted with a red box.

**STM32CubeMX Example Selection Screenshot:**

- MCU/MPU Selector: STM32F4
- Board Selector: NUCLEO-F429ZI
- Example Selector: PWR\_CurrentConsumption
- Cross Selector

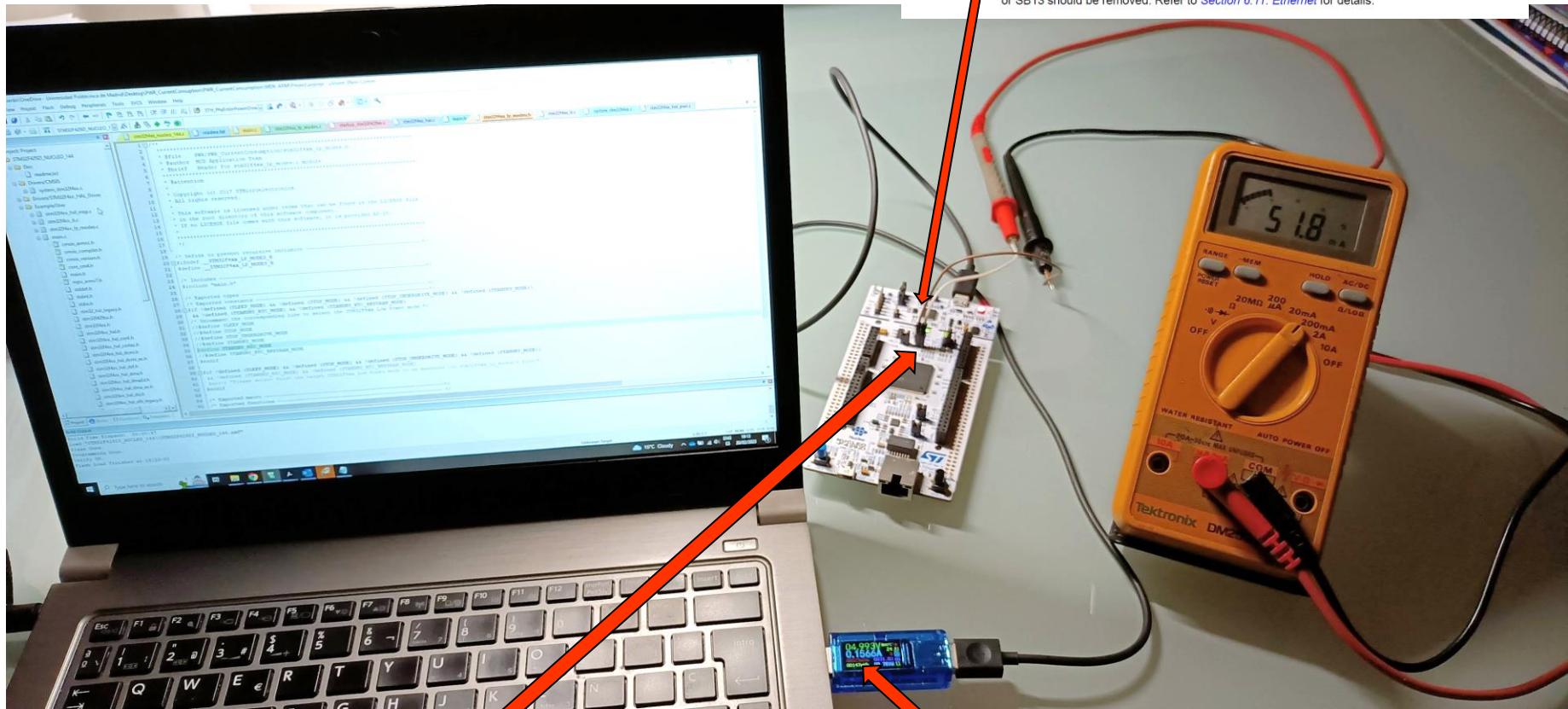
**Selected Project Details:**

- STM32F4**
- Required Software Package:** STM32Cube\_FW\_F4\_V1.27.1 (size: 773.0 MB) ✓
- Vendor:** STMicroelectronics
- Supported Toolchain/IDE:** EWARM, MDK-ARM, SW4STM32
- STM32CubeMX Minimum Compatible Version:** NA ❌
- Keywords:** System, RTC, Calendar, Backup Domain, Reset

**Examples List: 63 items**

Name	Board	Board Type	STM32CubeMX Compat...	STM32CubeMX Ver...	SW Package Insta...
NX_TTIP_Client	NUCLEO-F429ZI	Nucleo-144	✓	6.6.0	✓
Nx_UDP_Echo_Client	NUCLEO-F429ZI	Nucleo-144	✓	6.6.0	✓
Nx_UDP_Echo_Server	NUCLEO-F429ZI	Nucleo-144	✓	6.6.0	✓
Nx_UDP_Echo_Client	NUCLEO-F429ZI	Nucleo-144	✓	6.6.0	✓
Nx_UDP_Echo_Server	NUCLEO-F429ZI	Nucleo-144	✓	6.6.0	✓
Nx_WebServer	NUCLEO-F429ZI	Nucleo-144	✓	6.6.0	✓
<b>PWR_CurrentConsumption</b>	<b>NUCLEO-F429ZI</b>	<b>Nucleo-144</b>	✗	NA	✓
RCC_ClockConfig	NUCLEO-F429ZI	Nucleo-144	✗	NA	✓
RNG_GenerateRandomNumbers	NUCLEO-F429ZI	Nucleo-144	✗	NA	✓
RNG_GenerateRandomNumbers_IT	NUCLEO-F429ZI	Nucleo-144	✗	NA	✓
<b>RTC_Calendar</b>	<b>NUCLEO-F429ZI</b>	<b>Nucleo-144</b>	✗	NA	✓
RTC_Tamper	NUCLEO-F429ZI	Nucleo-144	✗	NA	✓
TIM_DMA	NUCLEO-F429ZI	Nucleo-144	✗	NA	✓
TIM_InputCapture	NUCLEO-F429ZI	Nucleo-144	✗	NA	✓
TIM_OCActive	NUCLEO-F429ZI	Nucleo-144	✗	NA	✓
TIM_OCTrigger	NUCLEO-F429ZI	Nucleo-144	✗	NA	✓

## ➤ Cómo mido el consumo de mi sistema?



### 6.6 Push-buttons

**B1 USER:** the user button is connected to the I/O PC13 by default (Tamper support, SB173 ON and SB180 OFF) or PA0 (Wakeup support, SB180 ON and SB173 OFF) of the STM32 microcontroller.

**B2 RESET:** this push-button is connected to NRST and is used to RESET the STM32 microcontroller.

### 6.7 JP5 (IDD)

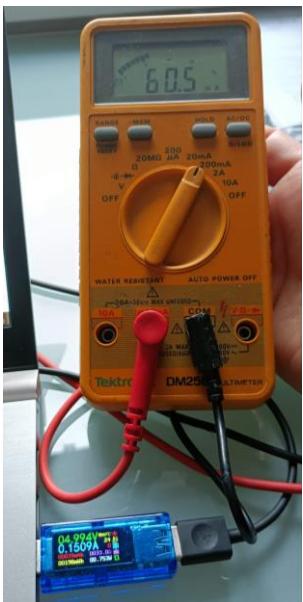
Jumper JP5, labeled IDD, is used to measure the STM32 microcontroller consumption by removing the jumper and by connecting an ammeter:

- JP5 ON: STM32 is powered (default)
- JP5 OFF: an ammeter must be connected to measure the STM32 current. If there is no ammeter, the STM32 is not powered

To get a correct current consumption, the Ethernet PHY should be set in power-down mode or SB13 should be removed. Refer to [Section 6.11: Ethernet](#) for details.

# Algunas medidas experimentales

Sistema a 180 MHz modo **RUN**



Sistema a 25 MHz modo **RUN**

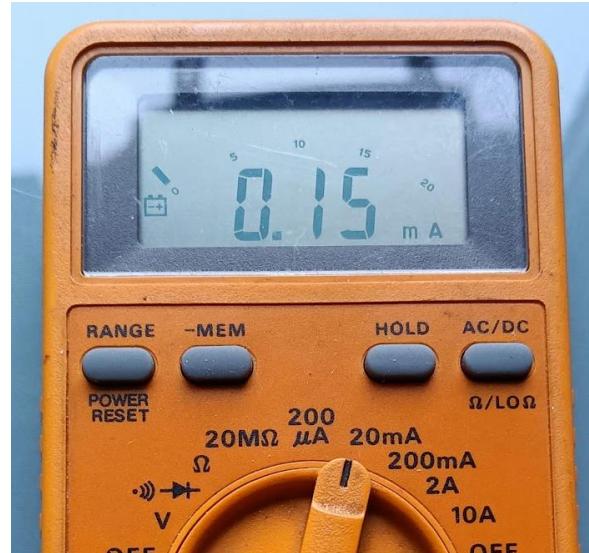


Modo	Consumo	Wake-up time
DYNAMIC RUN mode*	244 $\mu$ A/MHz @ 180 MHz = 44 mA @ 180 MHz	110 $\mu$ s
STOP mode	280 $\mu$ A Wake-up time: 105 $\mu$ s 310 $\mu$ A Wake-up time: 17 $\mu$ s	
STANDBY mode	120 $\mu$ A Wake-up time: 105 $\mu$ s 290 $\mu$ A Wake-up time: 17 $\mu$ s	
	9 $\mu$ A Wake-up time: 113 $\mu$ s 137 $\mu$ A/MHz @ 84 MHz	
	41 $\mu$ A Wake-up time: 21 $\mu$ s	
	1.8 $\mu$ A 2.4 $\mu$ A	
	2.2 $\mu$ A 3.1 $\mu$ A	
	2.2 $\mu$ A 3.1 $\mu$ A	
	1.8 $\mu$ A 2.4 $\mu$ A	
	2.2 $\mu$ A 3.1 $\mu$ A	
	1.8 $\mu$ A 2.4 $\mu$ A	

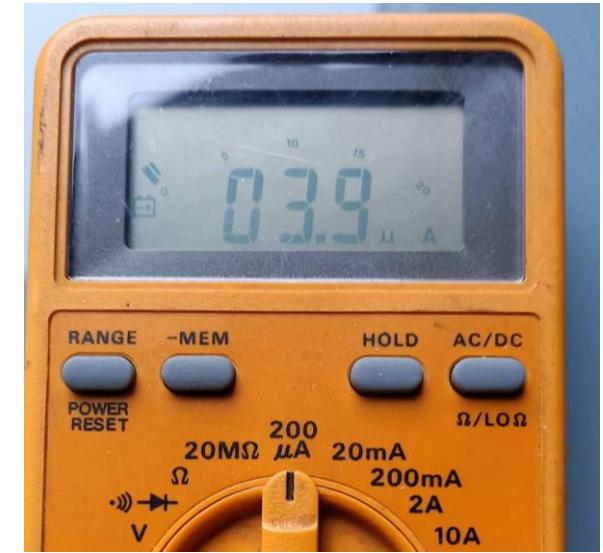
Modo **SLEEP**



Modo **STOP**



Modo **STANDBY**



# *Ingeniería de Sistemas Electrónicos*

## *Programación avanzada de microcontroladores (II)*

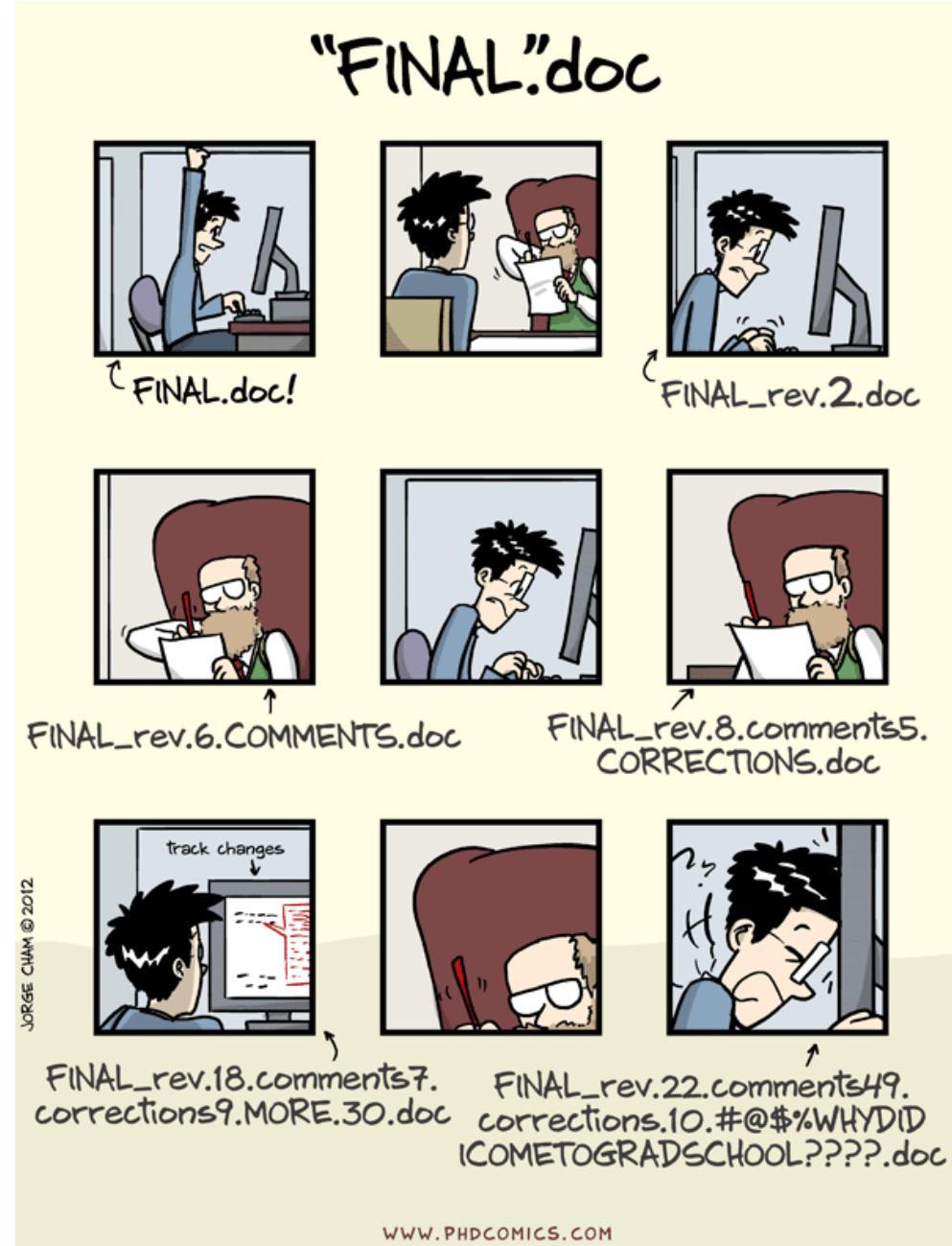
- 
- Integración de referencia temporal en el sistema
  - Desarrollo de aplicaciones “confiables”
  - Reducción del consumo
  - Desarrollo de software. Control de versiones
  - Prácticas de laboratorio

# Control de versiones

A alguien le suena??

Nombre

- main (3rd copy).cpp
- main (another copy).cpp
- main (copy).cpp
- main.cpp
- main\_old.cpp
- main\_v1.cpp
- main\_v1\_old.cpp
- main\_v2.cpp
- main\_v2\_old.cpp



## □ **Gestión manual de copias**

- Dificulta organización y versionado de ficheros
  - Complejidad creciente
  - Mucho uso de memoria
  - Fácil cometer errores
  - Difícil compartir
  - Gestión manual, sin herramientas de ayuda

## □ **Sistemas de almacenamiento online**

- Dropbox, Owncloud, Onedrive, Google Drive...
- Pros
  - Sincronización continua
  - Cierta gestión de versiones de ficheros
  - Fácil compartir
- Cons
  - Trabajo concurrente sobre ficheros
  - Control de las versiones (cuándo?, identificación, seguimiento)
  - Desarrollo de nuevas funcionalidades sobre un código estable

## □ Organizar versiones

- Version

- “Estado” de un proyecto en un momento dado

- Organizar

- Historial de cambios
  - Identificar cada versión
  - Recuperar versiones anteriores/diferentes
  - Facilitar la creación de nuevas versiones (con mínimas o múltiples diferencias)
  - Combinar varias versiones

## □ Compartir proyectos

- Permite acceso a múltiples desarrolladores a un proyecto común
- Facilita la colaboración
  - Herramienta estándar de gestión de proyectos
  - Seguimiento y trabajo concurrente sobre el mismo o diferentes ficheros

# *Sistemas de control de versiones: Version Control Systems (VCS)*

- Diferentes herramientas de control de versiones disponibles

- CVS (1986, GNU): Centralized, file-oriented. Deprecated



- SVN (2000, Apache): Centralized, project-oriented, heavy branches.



- Git (2005, Linus Torvalds): Distributed, project oriented, light branches.



- Mercurial (2005, Matt Mackall): Similar to git. Simpler, but less flexible.



- Sistema de Control de Versiones (VCS) distribuido
- Creado originalmente en 2005
  - Surge de la comunidad de desarrollo de Linux (Linus Torvalds)
  - Diseñado para hacer control de versiones en el kernel de Linux, pero útil en otros entornos
- Características
  - Rápido, sencillo, totalmente distribuido, gran soporte de trabajo concurrente y muchas ramas de desarrollo, adecuado para proyectos muy grandes/pequeños
- Aceptación
  - **MUY usado**
  - **Recomendación:** aprended a usar Git, os lo valorarán positivamente y os ahorrará muchos quebraderos de cabeza
  - Estándar de facto actual

# Git: documentación

- Sitio web de Git
  - <http://git-scm.com/>
- Libro online gratuito
  - <http://git-scm.com/book>
- Tutorial Git
  - <http://git-scm.com/docs/gittutorial>
- Línea de comandos
  - git help <command>



```
MINGW64:/c/Users/Eduardo/OneDrive - Universidad Politécnica de Madrid/Desktop/GitISE/P1/ise2022_1 (main)$ git help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:
start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  diff      Show changes between commits, commit and working tree, etc
  grep      Print lines matching a pattern
  log       Show commit logs
  show      Show various types of objects
  status    Show the working tree status

grow, mark and tweak your common history
  branch   List, create, or delete branches
  commit   Record changes to the repository
  merge    Join two or more development histories together
  rebase   Reapply commits on top of another base tip
  reset   Reset current HEAD to the specified state
  switch  Switch branches
  tag     Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch   Download objects and refs from another repository
  pull    Fetch from and integrate with another repository or a local branch
  push    Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

Eduardo@LAPTOP-K5I5CK58 MINGW64 ~/OneDrive - Universidad Politécnica de Madrid/Desktop/GitISE/P1/ise2022_1 (main)$ |
```

# *Git hosting*

---

- Existen diferentes servicios con diferentes condiciones
- Los más usados
  - Github
    - <https://github.com/> 
  - GitLab
    - <https://about.gitlab.com/> 
  - Bitbucket
    - <https://bitbucket.org/> 
- Primeros pasos para trabajar con Github:
  - <https://guides.github.com/activities/hello-world/>
  - <https://www.youtube.com/watch?v=HVsySz-h9r4>

# *Git Basic Concepts*

---

## □ Repository: Container of the project

- Contains the different files of the project
- Contains the multiple versions of each file
- Normally, there are several repositories for each project

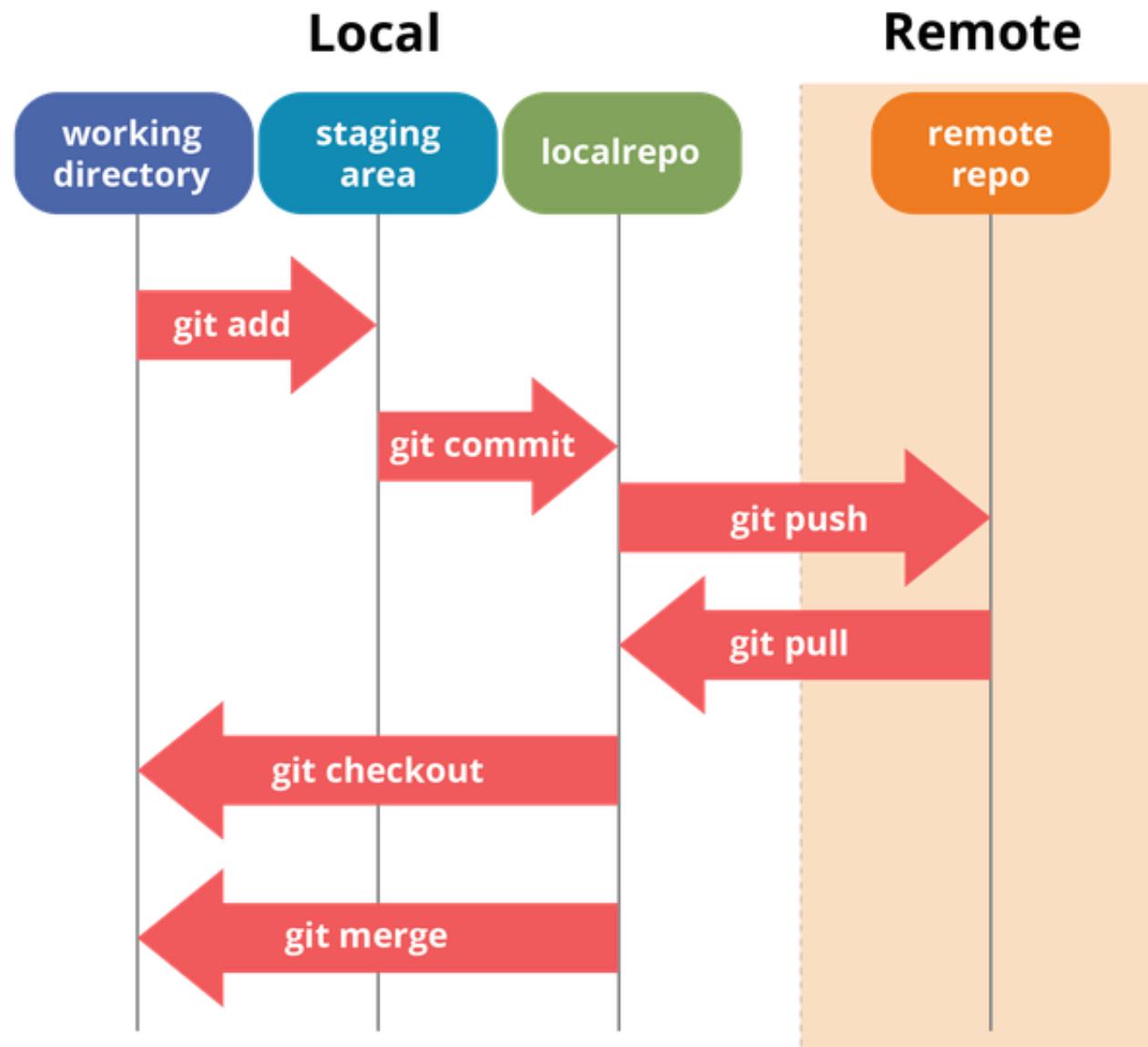
## ➤ Remote repository

- Normally in a server. Available for all the developers.

## ➤ Local repository

- Developers start downloading a copy of the remote repository (**clone**)
- The developer works in their local repository.
- The developer saves the versions to their local repository (**commit**)
- When changes are ready to be shared, developers upload their changes to the remote repository (**push**)
- For obtaining the latest changes in the remote repository, developers can download the latest changes (**pull**)

# Áreas Git



# Flujo de trabajo básico con Git (en local)

## □ 1.- Crear/Modificar

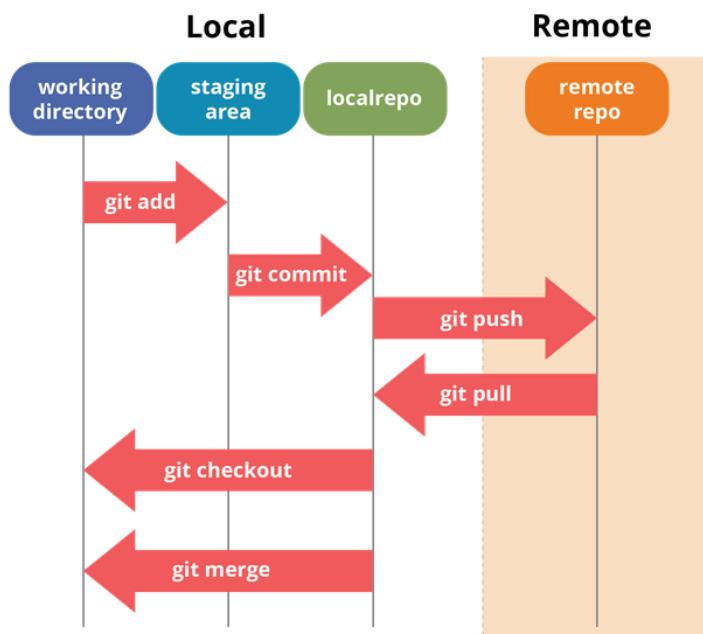
- Se crean/modifican los ficheros en el working directory

## □ 2.- Stage

- Selecciona ("add") los ficheros a incluir en el siguiente *commit*
- Añade snapshots de esos ficheros al staging area
  - Únicamente incluye esos ficheros (no otros modificados pero sobre los que no se hace stage)

## □ 3.- Commit

- Almacena **permanentemente** el snapshot con los ficheros del staging area en el directorio .git



# Qué es un commit ?

- Un **snapshot** del proyecto
  - Contiene **todos los ficheros** de un proyecto en un momento determinado, no las diferencias con versiones previas
- Un **checkpoint** en el proyecto
  - Puedo volver en cualquier momento
- Cada commit lleva asociado un **commit message**
  - Comentario **descriptivo e inequívoco** del snapshot

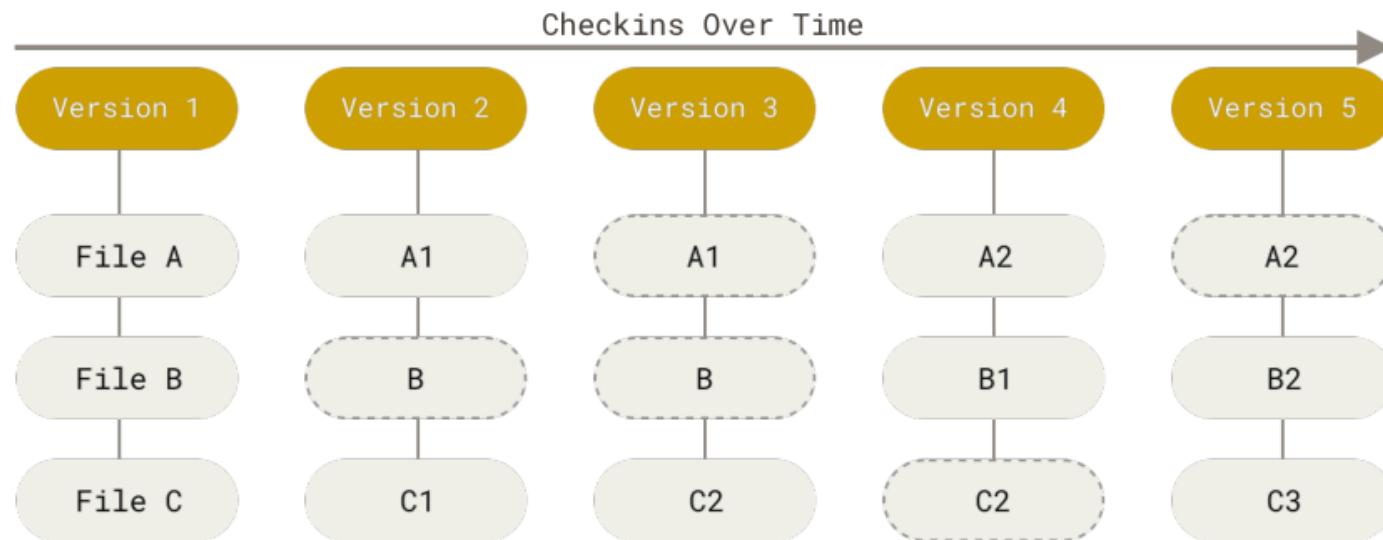


Figure 5. Storing data as snapshots of the project over time

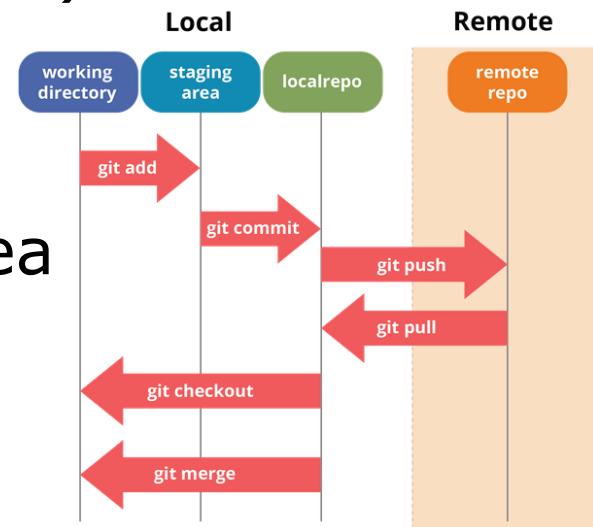
# *Flujo de trabajo con Git: set-up*

---

- Crear un repositorio en la Web de github
- Clonar el repositorio
  - git clone \_\_\_\_\_ → Comandos git en terminal (GitBash)
- Añadir ficheros/directorios al repo local
  - copiarlos al modo tradicional
  - git add (de los que se quiera hacer commit)
- Primer commit al repo local
  - git commit -m "Initial commit"
- Push al repo remoto
  - git push
- ...Fase de desarrollo en local...

# Flujo de trabajo con Git 2: trabajo local + remoto

- Recuperar versión actual del repo remoto (actualiza “Working copy”)
  - `git pull`
  - `git fetch`
  - `git merge`
- Modificar ficheros (del “Working copy” o copia local)
- Revisar cambios (respecto del repositorio)
  - `git status`
  - `git diff`
  - `git log`
- Añadir ficheros modificados al stage area
  - `git add`
- Commit cambios
  - `git commit`
- Incorporar los cambios en el repo remoto
  - `git push`



# *Using GIT with Keil environment*

---

[https://www.keil.com/appnotes/files/apnt\\_279.pdf](https://www.keil.com/appnotes/files/apnt_279.pdf)

## **□ Project Files under Version Control**

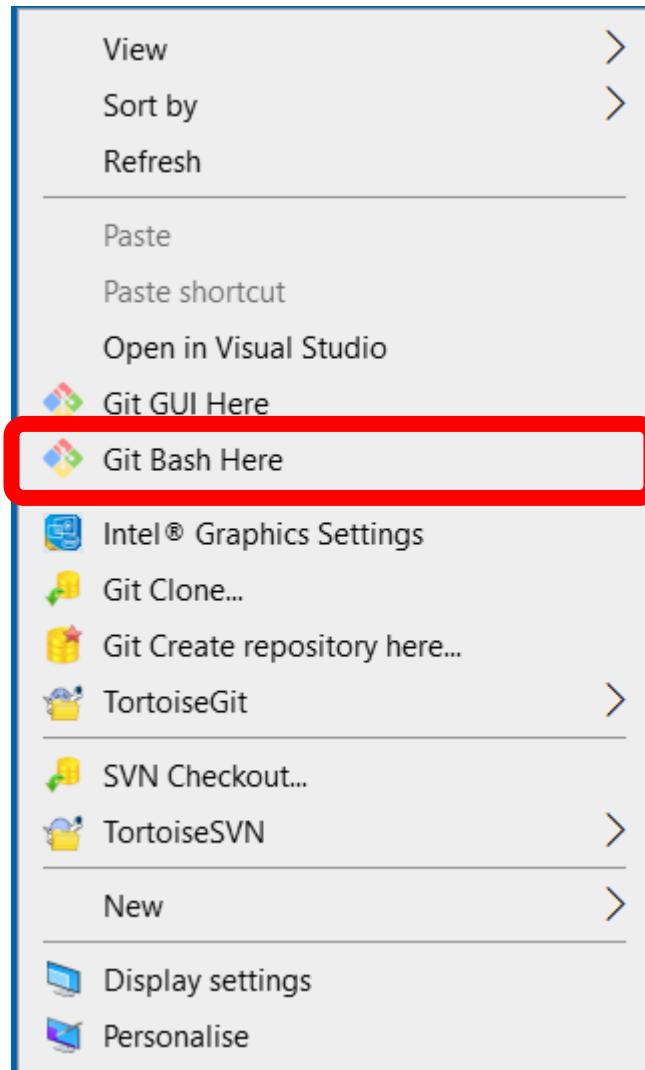
- Before setting up the workflow, the project manager should be absolutely clear about the files that need to be version controlled. Of course all source code files need to be versioned, but there are a couple of files that are special to µVision that need to be monitored as well.
- All user generated source files (\*.c, \*.cpp, \*.h, \*.inc, \*.s)
- Project file: *Project.uvprojx* (is used to build the project from scratch)
- Project options file: *Project.uvoptx* (contains information about the debugger and trace configuration)
- Configuration files for the run-time environment that are copied to the project (all files below .\RTE)
- List of #includes created by software components: RTE\RTE\_Components.h file
- Device configuration file: for example RTE\Device\LPC1857\RTE\_Device.h
- Linker control file (*Project.sct*) if created manually
- All relevant Pack files (for example ARM::CMSIS, Keil::Middleware, Device Family Packs, etc.)

## **□ Files that do not need to be monitored**

- Project screen layout file: *Project.uvguix.username*
- All files that are part of a Pack (the complete Pack will be revision controlled and is available to every user as soon as he is installing it using Pack Installer)
- Generated output files in the sub-directories .\Listings and .\Objects
- INI files for debug adapters

# *Steps to create your first repository*

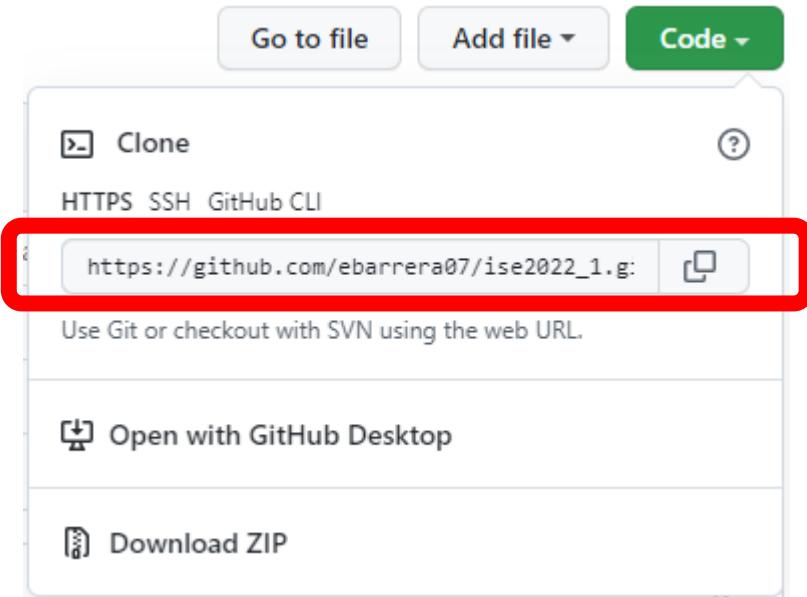
1. Access to [github.com](https://github.com)
2. Create an account
3. Sign-in
4. Create a new repository
5. Install git for your operating system
  - Windows(<https://git-scm.com/download/win>)
6. In the windows explorer, select the folder where you want to have your repository, right click and select Git Bash Here



# *Steps to create your first repository*

---

7. In the online github page navigate to your repository an click on Code button
8. Using the command window execute
  - `git clone <complete repo url>`
9. After some seconds you will have a copy of your project in a local folder.
10. Start working with your code an learn to use the commands *git add*, *git commit* and *git push (pull)*



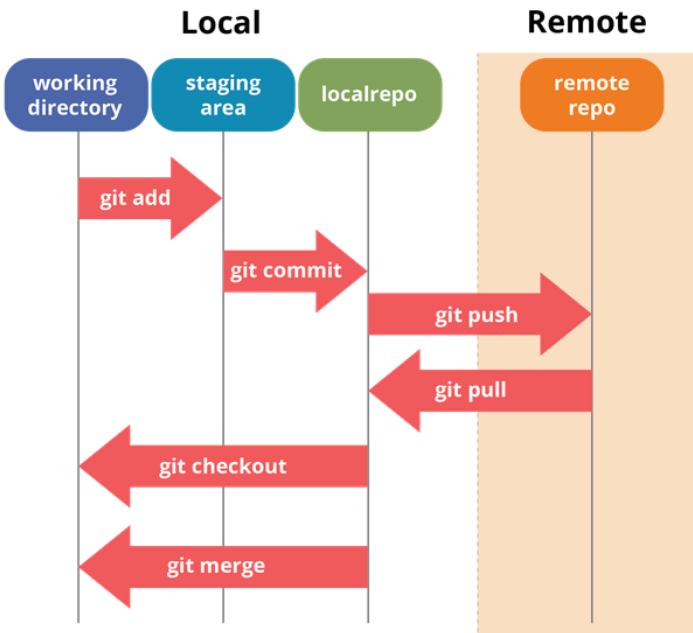
# *Pasos para trabajar con GIT en ISE*

---

- Crear cuenta en github con correo electrónico de alumnos UPM
- Crear un repositorio para cada estudiante
  - Nombre del repo: ISE2025\_grupoLAB\_ApellidoNombre
- Añada como colaborador a los profesores de la asignatura ([eduardo.barrera@upm.es](mailto:eduardo.barrera@upm.es), [julian.nieto.valhondo@upm.es](mailto:julian.nieto.valhondo@upm.es))
- Crear carpeta/s local/es para alojar el/los repositorio/s local/es (laboratorio, portátil, casa)
- Abrir Git Bash (click derecho) en dicha carpeta
- Configurar entorno -usuario y email- (solo una vez)
  - git config user.name "Mona Lisa"
  - git config user.email "email@example.com"

# Pasos para trabajar con GIT en ISE

- Clonar el repositorio en la carpeta local (solo una vez)
  - `git clone <repository_url>`
- Copiar el ejemplo del servidor Web original (sin modificar), tras hacer un clean del proyecto
- Añadir los ficheros de los que se desee hacer commit (puede hacerse uso de `.gitignore`)
  - `git add <path>`
- Hacer commit (con comentarios) del ejemplo original
  - `git commit -m "comentarios"`
- Hacer push para subir el repositorio local a remote
  - `git push`



# *Pasos para trabajar con GIT en ISE*

---

- Revisar el repositorio remoto en la Web de github
- Añadir las **modificaciones** al ejemplo del servidor en el repositorio local
- Añadir los **ficheros** de los que se desee hacer commit (puede hacerse uso de .gitignore)
  - `git add <path>`
- Hacer uno o varios **commit** (con comentarios) de las modificaciones que se han realizado
  - `git commit -m "comentarios"`
- Hacer **push** para subir el repositorio local a remote
  - `git push`
- Revisar nuevamente el repositorio remoto en la Web de github

# *Pasos para trabajar con GIT en ISE*

---

- Todo el código desarrollado en la asignatura deberá estar en github
- En cada sesión de laboratorio se harán varios commit (con comentarios descriptivos de los cambios) y al menos un push al finalizar la sesión
- En el trabajo no presencial deberán realizarse igualmente commit con comentarios y al menos un push por cada día que se trabaje en la asignatura
- Cuando comience el Bloque 2 de la asignatura se deberá crear un repositorio por cada Equipo de Trabajo
- Añada como colaborador a su profesor de la asignatura

# *Ingeniería de Sistemas Electrónicos*

## *Programación avanzada de microcontroladores (II)*

- 
- Integración de referencia temporal en el sistema
  - Desarrollo de aplicaciones “confiables”
  - Reducción del consumo
  - Desarrollo de software. Control de versiones
  - Prácticas de laboratorio

- Preliminar: Integración de Práctica 1 en sistema de Control de Versiones (GIT) ([libre acceso](#))
  - **Práctica 2:** Integración de SNTP y RTC en servidor
    - ❖ 1 semana (usar repo Git individual)
  - **Práctica 3:** Modos de bajo consumo y Watchdog
    - ❖ 1 semana (usar repo Git individual)
- 
- ❖ **Entrega P1, P2 y P3** en Moodle → 28 feb, 7 mar, 14 mar (solo proyectos Keil)
  - ❖ Entrega **documentación** completa del **Bloque 1** → viernes, **21 de marzo**
  - ❖ **Evaluación individual Bloque 1** → lunes, **31 de marzo**

# *Desarrollo de la asignatura*

## □ Entregas Bloque 1

	L	M	X	J	V	S	D
Feb	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
Mar	24	25	26	27	28	1	2
	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
Abr	24	25	26	27	28	29	30
	31	1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
May	21	22	23	24	25	26	27
	28	29	30	1	2	3	4
	5	6	7	8	9	10	11
	12	Hor J.13	14	15	16	17	18
	19	20	21	22	23	24	25
	26	27	28	29	30	31	1
Jun	30	1	2	3	4	5	6

## PROYECTOS DISEÑO BLOQUE 2

Lunes 17 febrero:

- Equipos de 3/4 estudiantes (del mismo grupo de laboratorio)
- Representante de cada Equipo

Lunes 3 de marzo:

- Propuestas de diseño Bloque 2

Lunes 17 de marzo:

- Presentaciones propuestas Bloque 2

## Prácticas BLOQUE 1

Viernes 28 de febrero:

- Práctica 1

Viernes 7 de marzo:

- Práctica 2

Viernes 14 de marzo:

- Práctica 3

Viernes 21 de marzo:

- Documentación Bloque 1