

Doxy html pdf
1.0

Generado por Doxygen 1.9.8

1 Topic Index	1
1.1 Topics	1
2 Índice de archivos	3
2.1 Lista de archivos	3
3 Topic Documentation	5
3.1 CMSIS	5
3.1.1 Descripción detallada	5
3.1.2 Stm32g4xx_system	5
3.1.2.1 Descripción detallada	5
3.1.2.2 STM32G4xx_System_Private_Includes	5
3.1.2.3 STM32G4xx_System_Private_TypesDefinitions	6
3.1.2.4 STM32G4xx_System_Private_Defines	6
3.1.2.5 STM32G4xx_System_Private_Macros	6
3.1.2.6 STM32G4xx_System_Private_Variables	6
3.1.2.7 STM32G4xx_System_Private_FunctionPrototypes	6
3.1.2.8 STM32G4xx_System_Private_Functions	6
4 Documentación de archivos	9
4.1 Referencia del archivo Core/Src/main.c	9
4.1.1 Descripción detallada	9
4.1.2 Documentación de funciones	9
4.1.2.1 Error_Handler()	9
4.1.2.2 main()	10
4.1.2.3 SystemClock_Config()	10
4.2 Referencia del archivo Core/Src/stm32g4xx_hal_msp.c	10
4.2.1 Descripción detallada	11
4.2.2 Documentación de funciones	11
4.2.2.1 HAL_MspInit()	11
4.2.2.2 HAL_UART_MspDeInit()	11
4.2.2.3 HAL_UART_MspInit()	11
4.3 Referencia del archivo Core/Src/stm32g4xx_it.c	12
4.3.1 Descripción detallada	12
4.4 Referencia del archivo Core/Src/syscalls.c	13
4.4.1 Descripción detallada	13
4.5 Referencia del archivo Core/Src/sysmem.c	14
4.5.1 Descripción detallada	14
4.5.2 Documentación de funciones	14
4.5.2.1 _sbrk()	14
4.6 Referencia del archivo Core/Src/system_stm32g4xx.c	15
4.6.1 Descripción detallada	15
4.6.2 This file configures the system clock as follows:	16

4.6.2.1 System Clock source HSI	16
4.6.2.2 SYSCLK(Hz) 16000000	16
4.6.2.3 HCLK(Hz) 16000000	16
4.6.2.4 AHB Prescaler 1	16
4.6.2.5 APB1 Prescaler 1	16
4.6.2.6 APB2 Prescaler 1	16
4.6.2.7 PLL_M 1	16
4.6.2.8 PLL_N 16	16
4.6.2.9 PLL_P 7	16
4.6.2.10 PLL_Q 2	16
4.6.2.11 PLL_R 2	16
4.6.2.12 Require 48MHz for RNG Disabled	16

Índice alfabético	17
--------------------------	-----------

Capítulo 1

Topic Index

1.1. Topics

Here is a list of all topics with brief descriptions:

CMSIS	5
Stm32g4xx_system	5
STM32G4xx_System_Private_Includes	5
STM32G4xx_System_Private_TypesDefinitions	6
STM32G4xx_System_Private_Defines	6
STM32G4xx_System_Private_Macros	6
STM32G4xx_System_Private_Variables	6
STM32G4xx_System_Private_FunctionPrototypes	6
STM32G4xx_System_Private_Functions	6

Capítulo 2

Índice de archivos

2.1. Lista de archivos

Lista de todos los archivos documentados y con breves descripciones:

Core/Src/ main.c	
: Main program body	9
Core/Src/ stm32g4xx_hal_msp.c	
This file provides code for the MSP Initialization and de-Initialization codes	10
Core/Src/ stm32g4xx_it.c	
Interrupt Service Routines	12
Core/Src/ syscalls.c	
STM32CubeIDE Minimal System calls file	13
Core/Src/ sysmem.c	
STM32CubeIDE System Memory calls file	14
Core/Src/ system_stm32g4xx.c	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File	15

Capítulo 3

Topic Documentation

3.1. CMSIS

Módulos

- [Stm32g4xx_system](#)

3.1.1. Descripción detallada

3.1.2. Stm32g4xx_system

Módulos

- [STM32G4xx_System_Private_Includes](#)
- [STM32G4xx_System_Private_TypesDefinitions](#)
- [STM32G4xx_System_Private_Defines](#)
- [STM32G4xx_System_Private_Macros](#)
- [STM32G4xx_System_Private_Variables](#)
- [STM32G4xx_System_Private_FunctionPrototypes](#)
- [STM32G4xx_System_Private_Functions](#)

3.1.2.1. Descripción detallada

3.1.2.2. STM32G4xx_System_Private_Includes

defines

- #define [HSE_VALUE](#) 24000000U
- #define [HSI_VALUE](#) 16000000U

3.1.2.2.1. Descripción detallada

3.1.2.2.2. Documentación de «define»

3.1.2.2.2.1. HSE_VALUE

```
#define HSE_VALUE 24000000U
```

Value of the External oscillator in Hz

3.1.2.2.2.2. HSI_VALUE

```
#define HSI_VALUE 16000000U
```

Value of the Internal oscillator in Hz

3.1.2.3. STM32G4xx_System_Private_TypesDefinitions

3.1.2.4. STM32G4xx_System_Private_Defines

3.1.2.5. STM32G4xx_System_Private_Macros

3.1.2.6. STM32G4xx_System_Private_Variables

Variables

- uint32_t **SystemCoreClock** = [HSI_VALUE](#)
- const uint8_t **AHBPrescTable** [16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}
- const uint8_t **APBPrescTable** [8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}

3.1.2.6.1. Descripción detallada

3.1.2.7. STM32G4xx_System_Private_FunctionPrototypes

3.1.2.8. STM32G4xx_System_Private_Functions

Funciones

- void [SystemInit](#) (void)
Setup the microcontroller system.
- void [SystemCoreClockUpdate](#) (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

3.1.2.8.1. Descripción detallada

3.1.2.8.2. Documentación de funciones

3.1.2.8.2.1. SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (
    void )
```

Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Nota

Each time the core clock (HCLK) changes, this function must be called to update SystemCoreClock variable value. Otherwise, any configuration based on this variable will be incorrect.

- The system frequency computed by this function is not the real frequency in the chip. It is calculated based on the predefined constant and the selected clock source:

- If SYSCLK source is HSI, SystemCoreClock will contain the [HSI_VALUE\(**\)](#)
- If SYSCLK source is HSE, SystemCoreClock will contain the [HSE_VALUE\(***\)](#)
- If SYSCLK source is PLL, SystemCoreClock will contain the [HSE_VALUE\(***\)](#) or [HSI_VALUE\(*\)](#) multiplied/divided by the PLL factors.

(**) HSI_VALUE is a constant defined in stm32g4xx_hal.h file (default value 16 MHz) but the real value may vary depending on the variations in voltage and temperature.

(***) HSE_VALUE is a constant defined in stm32g4xx_hal.h file (default value 24 MHz), user has to ensure that HSE_VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

- The result of this function could be not correct when using fractional value for HSE crystal.

Parámetros

None	
------	--

Valores devueltos

None	
------	--

3.1.2.8.2.2. SystemInit()

```
void SystemInit (
    void )
```

Setup the microcontroller system.

Parámetros

<i>None</i>	
-------------	--

Valores devueltos

<i>None</i>	
-------------	--

Capítulo 4

Documentación de archivos

4.1. Referencia del archivo Core/Src/main.c

: Main program body

```
#include "main.h"
```

Funciones

- void [SystemClock_Config](#) (void)
System Clock Configuration.
- int [main](#) (void)
The application entry point.
- void [Error_Handler](#) (void)
This function is executed in case of error occurrence.

Variables

- UART_HandleTypeDef **hlpuart1**

4.1.1. Descripción detallada

: Main program body

Atención

Copyright (c) 2023 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

4.1.2. Documentación de funciones

4.1.2.1. Error_Handler()

```
void Error_Handler (  
    void )
```

This function is executed in case of error occurrence.

Valores devueltos

<i>None</i>	
-------------	--

4.1.2.2. main()

```
int main (
        void )
```

The application entry point.

Valores devueltos

<i>int</i>	
------------	--

4.1.2.3. SystemClock_Config()

```
void SystemClock_Config (
        void )
```

System Clock Configuration.

Valores devueltos

<i>None</i>	
-------------	--

Configure the main internal regulator output voltage

Initializes the RCC Oscillators according to the specified parameters in the RCC_OscInitTypeDef structure.

Initializes the CPU, AHB and APB buses clocks

4.2. Referencia del archivo Core/Src/stm32g4xx_hal_msp.c

This file provides code for the MSP Initialization and de-Initialization codes.

```
#include "main.h"
```

Funciones

- void [HAL_MspInit](#) (void)
- void [HAL_UART_MspInit](#) (UART_HandleTypeDef *huart)
UART MSP Initialization This function configures the hardware resources used in this example.
- void [HAL_UART_MspDeInit](#) (UART_HandleTypeDef *huart)
UART MSP De-Initialization This function freeze the hardware resources used in this example.

4.2.1. Descripción detallada

This file provides code for the MSP Initialization and de-Initialization codes.

Atención

Copyright (c) 2023 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

4.2.2. Documentación de funciones

4.2.2.1. HAL_MspInit()

```
void HAL_MspInit (
    void )
```

Initializes the Global MSP. Disable the internal Pull-Up in Dead Battery pins of UCPD peripheral

4.2.2.2. HAL_UART_MspDeInit()

```
void HAL_UART_MspDeInit (
    UART_HandleTypeDef * huart )
```

UART MSP De-Initialization This function freeze the hardware resources used in this example.

Parámetros

<i>huart</i>	UART handle pointer
--------------	---------------------

Valores devueltos

<i>None</i>	
-------------	--

LPUART1 GPIO Configuration PA2 ----> LPUART1_TX PA3 ----> LPUART1_RX

4.2.2.3. HAL_UART_MspInit()

```
void HAL_UART_MspInit (
    UART_HandleTypeDef * huart )
```

UART MSP Initialization This function configures the hardware resources used in this example.

Parámetros

<i>huart</i>	UART handle pointer
--------------	---------------------

Valores devueltos

<i>None</i>	
-------------	--

Initializes the peripherals clocks

LPUART1 GPIO Configuration PA2 ----> LPUART1_TX PA3 ----> LPUART1_RX

4.3. Referencia del archivo Core/Src/stm32g4xx_it.c

Interrupt Service Routines.

```
#include "main.h"
#include "stm32g4xx_it.h"
```

Funciones

- void **NMI_Handler** (void)
This function handles Non maskable interrupt.
- void **HardFault_Handler** (void)
This function handles Hard fault interrupt.
- void **MemManage_Handler** (void)
This function handles Memory management fault.
- void **BusFault_Handler** (void)
This function handles Prefetch fault, memory access fault.
- void **UsageFault_Handler** (void)
This function handles Undefined instruction or illegal state.
- void **SVC_Handler** (void)
This function handles System service call via SWI instruction.
- void **DebugMon_Handler** (void)
This function handles Debug monitor.
- void **PendSV_Handler** (void)
This function handles Pendable request for system service.
- void **SysTick_Handler** (void)
This function handles System tick timer.

4.3.1. Descripción detallada

Interrupt Service Routines.

Atención

Copyright (c) 2023 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

4.4. Referencia del archivo Core/Src/syscalls.c

STM32CubeIDE Minimal System calls file.

```
#include <sys/stat.h>
#include <stdlib.h>
#include <errno.h>
#include <stdio.h>
#include <signal.h>
#include <time.h>
#include <sys/time.h>
#include <sys/times.h>
```

Funciones

- `int __io_putchar (int ch) __attribute__((weak))`
- `int __io_getchar (void)`
- `void initialise_monitor_handles ()`
- `int _getpid (void)`
- `int _kill (int pid, int sig)`
- `void _exit (int status)`
- `__attribute__((weak))`
- `int _close (int file)`
- `int _fstat (int file, struct stat *st)`
- `int _isatty (int file)`
- `int _lseek (int file, int ptr, int dir)`
- `int _open (char *path, int flags,...)`
- `int _wait (int *status)`
- `int _unlink (char *name)`
- `int _times (struct tms *buf)`
- `int _stat (char *file, struct stat *st)`
- `int _link (char *old, char *new)`
- `int _fork (void)`
- `int _execve (char *name, char **argv, char **env)`

Variables

- `char ** environ = __env`

4.4.1. Descripción detallada

STM32CubeIDE Minimal System calls file.

Autor

Auto-generated by STM32CubeIDE

For more information about which c-functions
need which of these lowlevel functions
please consult the Newlib libc-manual

Atención

Copyright (c) 2020-2023 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

4.5. Referencia del archivo Core/Src/sysmem.c

STM32CubeIDE System Memory calls file.

```
#include <errno.h>
#include <stdint.h>
```

Funciones

- void * [_sbrk](#) (ptrdiff_t incr)
[_sbrk\(\)](#) allocates memory to the newlib heap and is used by malloc and others from the C library

4.5.1. Descripción detallada

STM32CubeIDE System Memory calls file.

Autor

Generated by STM32CubeIDE

For more information about which C functions
need which of these lowlevel functions
please consult the newlib libc manual

Atención

Copyright (c) 2023 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

4.5.2. Documentación de funciones

4.5.2.1. [_sbrk\(\)](#)

```
void * _sbrk (
    ptrdiff_t incr )
```

[_sbrk\(\)](#) allocates memory to the newlib heap and is used by malloc and others from the C library

```
* #####
* # .data # .bss #          newlib heap          #          MSP stack          #
* #          #          #          #          # Reserved by _Min_Stack_Size #
* #####
* ^-- RAM start          ^-- _end          _estack, RAM end --^
*
```

This implementation starts allocating at the '_end' linker symbol The '_Min_Stack_Size' linker symbol reserves a memory for the MSP stack The implementation considers '_estack' linker symbol to be RAM end NOTE: If the MSP stack, at any point during execution, grows larger than the reserved size, please increase the '_Min_Stack_Size'.

Parámetros

<i>incr</i>	Memory size
-------------	-------------

Devuelve

Pointer to allocated memory

4.6. Referencia del archivo Core/Src/system_stm32g4xx.c

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File.

```
#include "stm32g4xx.h"
```

defines

- #define **HSE_VALUE** 24000000U
- #define **HSI_VALUE** 16000000U

Funciones

- void **SystemInit** (void)
Setup the microcontroller system.
- void **SystemCoreClockUpdate** (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Variables

- uint32_t **SystemCoreClock** = **HSI_VALUE**
- const uint8_t **AHBPrescTable** [16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}
- const uint8_t **APBPrescTable** [8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}

4.6.1. Descripción detallada

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File.

Autor

MCD Application Team

This file provides two functions and one global variable to be called from user application:

- **SystemInit()**: This function is called at startup just after reset and before branch to main program. This call is made inside the "startup_stm32g4xx.s" file.
- **SystemCoreClock** variable: Contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.
- **SystemCoreClockUpdate()**: Updates the variable SystemCoreClock and must be called whenever the core clock is changed during program execution.

After each device reset the HSI (16 MHz) is used as system clock source. Then **SystemInit()** function is called, in "startup_stm32g4xx.s" file, to configure the system clock before to branch to main program.

4.6.2. This file configures the system clock as follows:

4.6.2.1. System Clock source | HSI

4.6.2.2. SYSCLK(Hz) | 16000000

4.6.2.3. HCLK(Hz) | 16000000

4.6.2.4. AHB Prescaler | 1

4.6.2.5. APB1 Prescaler | 1

4.6.2.6. APB2 Prescaler | 1

4.6.2.7. PLL_M | 1

4.6.2.8. PLL_N | 16

4.6.2.9. PLL_P | 7

4.6.2.10. PLL_Q | 2

4.6.2.11. PLL_R | 2

4.6.2.12. Require 48MHz for RNG | Disabled

=====

Atención

Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

Índice alfabético

`_sbrk`
 `sysmem.c`, [14](#)

`CMSIS`, [5](#)
`Core/Src/main.c`, [9](#)
`Core/Src/stm32g4xx_hal_msp.c`, [10](#)
`Core/Src/stm32g4xx_it.c`, [12](#)
`Core/Src/syscalls.c`, [13](#)
`Core/Src/sysmem.c`, [14](#)
`Core/Src/system_stm32g4xx.c`, [15](#)

`Error_Handler`
 `main.c`, [9](#)

`HAL_MspInit`
 `stm32g4xx_hal_msp.c`, [11](#)
`HAL_UART_MspDeInit`
 `stm32g4xx_hal_msp.c`, [11](#)
`HAL_UART_MspInit`
 `stm32g4xx_hal_msp.c`, [11](#)

`HSE_VALUE`
 `STM32G4xx_System_Private_Includes`, [6](#)

`HSI_VALUE`
 `STM32G4xx_System_Private_Includes`, [6](#)

`main`
 `main.c`, [10](#)

`main.c`
 `Error_Handler`, [9](#)
 `main`, [10](#)
 `SystemClock_Config`, [10](#)

`stm32g4xx_hal_msp.c`
 `HAL_MspInit`, [11](#)
 `HAL_UART_MspDeInit`, [11](#)
 `HAL_UART_MspInit`, [11](#)

`Stm32g4xx_system`, [5](#)
`STM32G4xx_System_Private_Defines`, [6](#)
`STM32G4xx_System_Private_FunctionPrototypes`, [6](#)
`STM32G4xx_System_Private_Functions`, [6](#)
 `SystemCoreClockUpdate`, [7](#)
 `SystemInit`, [7](#)
`STM32G4xx_System_Private_Includes`, [5](#)
 `HSE_VALUE`, [6](#)
 `HSI_VALUE`, [6](#)
`STM32G4xx_System_Private_Macros`, [6](#)
`STM32G4xx_System_Private_TypesDefinitions`, [6](#)
`STM32G4xx_System_Private_Variables`, [6](#)
`sysmem.c`
 `_sbrk`, [14](#)

`SystemClock_Config`
 `main.c`, [10](#)
`SystemCoreClockUpdate`
 `STM32G4xx_System_Private_Functions`, [7](#)
`SystemInit`
 `STM32G4xx_System_Private_Functions`, [7](#)