

SysAcq_Pertuet_Schini

Generated by Doxygen 1.9.5

1 Module Index	1
1.1 Modules	1
2 File Index	3
2.1 File List	3
3 Module Documentation	5
3.1 CMSIS	5
3.1.1 Detailed Description	5
3.2 Stm32g4xx_system	5
3.2.1 Detailed Description	5
3.3 STM32G4xx_System_Private_Includes	5
3.3.1 Detailed Description	6
3.3.2 Macro Definition Documentation	6
3.3.2.1 HSE_VALUE	6
3.3.2.2 HSI_VALUE	6
3.4 STM32G4xx_System_Private_TypesDefinitions	6
3.5 STM32G4xx_System_Private_Defines	6
3.6 STM32G4xx_System_Private_Macros	6
3.7 STM32G4xx_System_Private_Variables	6
3.7.1 Detailed Description	6
3.8 STM32G4xx_System_Private_FunctionPrototypes	6
3.9 STM32G4xx_System_Private_Functions	6
3.9.1 Detailed Description	7
3.9.2 Function Documentation	7
3.9.2.1 SystemCoreClockUpdate()	7
3.9.2.2 SystemInit()	7
4 File Documentation	9
4.1 main.c File Reference	9
4.1.1 Detailed Description	11
4.1.2 Function Documentation	11
4.1.2.1 delete()	11
4.1.2.2 Error_Handler()	11
4.1.2.3 HAL_ADC_ConvCpltCallback()	12
4.1.2.4 help()	12
4.1.2.5 main()	12
4.1.2.6 new_carac()	12
4.1.2.7 speed()	12
4.1.2.8 start()	13
4.1.2.9 stop()	13
4.1.2.10 SystemClock_Config()	13
4.1.3 Variable Documentation	13

4.1.3.1 started	13
4.1.3.2 vitesse	14
4.2 stm32g4xx_hal_msp.c File Reference	14
4.2.1 Detailed Description	14
4.2.2 Function Documentation	15
4.2.2.1 HAL_ADC_MspDeInit()	15
4.2.2.2 HAL_ADC_MspInit()	15
4.2.2.3 HAL_MspInit()	15
4.2.2.4 HAL_TIM_Base_MspDeInit()	16
4.2.2.5 HAL_TIM_Base_MspInit()	16
4.2.2.6 HAL_TIM_Encoder_MspDeInit()	16
4.2.2.7 HAL_TIM_Encoder_MspInit()	17
4.2.2.8 HAL_TIM_MspPostInit()	17
4.2.2.9 HAL_UART_MspDeInit()	17
4.2.2.10 HAL_UART_MspInit()	17
4.3 stm32g4xx_it.c File Reference	18
4.3.1 Detailed Description	19
4.3.2 Variable Documentation	19
4.3.2.1 vitesse	19
4.4 syscalls.c File Reference	19
4.4.1 Detailed Description	20
4.5 systemem.c File Reference	20
4.5.1 Detailed Description	21
4.5.2 Function Documentation	21
4.5.2.1 _sbrk()	21
4.6 system_stm32g4xx.c File Reference	22
4.6.1 Detailed Description	22
4.6.2 This file configures the system clock as follows:	23
4.6.2.1 System Clock source HSI	23
4.6.2.2 SYCLK(Hz) 16000000	23
4.6.2.3 HCLK(Hz) 16000000	23
4.6.2.4 AHB Prescaler 1	23
4.6.2.5 APB1 Prescaler 1	23
4.6.2.6 APB2 Prescaler 1	23
4.6.2.7 PLL_M 1	23
4.6.2.8 PLL_N 16	23
4.6.2.9 PLL_P 7	23
4.6.2.10 PLL_Q 2	23
4.6.2.11 PLL_R 2	23
4.6.2.12 Require 48MHz for RNG Disabled	23

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

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

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

main.c	: Main program body	9
stm32g4xx_hal_msp.c	This file provides code for the MSP Initialization and de-Initialization codes	14
stm32g4xx_it.c	Interrupt Service Routines	18
syscalls.c	STM32CubeIDE Minimal System calls file	19
systemem.c	STM32CubeIDE System Memory calls file	20
system_stm32g4xx.c	CMSIS Cortex-M4 Device Peripheral Access Layer System Source File	22

Chapter 3

Module Documentation

3.1 CMSIS

Modules

- [Stm32g4xx_system](#)

3.1.1 Detailed Description

3.2 Stm32g4xx_system

Modules

- [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](#)

3.2.1 Detailed Description

3.3 STM32G4xx_System_Private_Includes

Macros

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

3.3.1 Detailed Description

3.3.2 Macro Definition Documentation

3.3.2.1 HSE_VALUE

```
#define HSE_VALUE 24000000U
```

Value of the External oscillator in Hz

3.3.2.2 HSI_VALUE

```
#define HSI_VALUE 16000000U
```

Value of the Internal oscillator in Hz

3.4 STM32G4xx_System_Private_TypeDefinitions

3.5 STM32G4xx_System_Private_Defines

3.6 STM32G4xx_System_Private_Macros

3.7 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}

3.7.1 Detailed Description

3.8 STM32G4xx_System_Private_FunctionPrototypes

3.9 STM32G4xx_System_Private_Functions

Functions

- 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.

3.9.1 Detailed Description

3.9.2 Function Documentation

3.9.2.1 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.

Note

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.

Parameters

None	
------	--

Return values

None	
------	--

3.9.2.2 SystemInit()

```
void SystemInit (
```

```
void  )
```

Setup the microcontroller system.

Parameters

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

Return values

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

Chapter 4

File Documentation

4.1 main.c File Reference

: Main program body

```
#include "main.h"
#include "corrPI.h"
#include "commandeMCC.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Macros

- `#define UART_TX_BUFFER_SIZE 64`
- `#define UART_RX_BUFFER_SIZE 1`
- `#define CMD_BUFFER_SIZE 64`
- `#define MAX_ARGS 9`
- `#define ASCII_LF 0x0A`
- `#define ASCII_CR 0x0D`
- `#define ASCII_DEL 0x7F`
- `#define ARR_VAL 1024`
- `#define ALPHA 600 /** Rapport cyclique permettant de gérer les PWM entre 0 et 1023 */`
- `#define KP_ALPHA 0.10`
- `#define KI_ALPHA 0.8`
- `#define ALPHA_OUT_MAX_VALUE 0.99`
- `#define ALPHA_OUT_MIN_VALUE 0.01`
- `#define TIM1_PERIOD 0.0000625`
- `#define KP_CURRENT 0.1`
- `#define KI_CURRENT 2.0`
- `#define CURRENT_OUT_MAX_VALUE 2.0`
- `#define TIM3_PERIOD 0.1`
- `#define TIM4_PERIOD 0.0064`
- `#define TICK2SPEED_TIM3 0.0146`
- `#define TICK2SPEED_TIM4 (60.0/4096.0)/TIM4_PERIOD`

Functions

- void `SystemClock_Config` (void)
System Clock Configuration.
- void `newcom` (void)
- void `delete` (void)
- void `new_carac` (void)
- void `get` (void)
- void `set` (void)
- void `help` (void)
Présente les fonctions et leur utilités dans la console.
- void `start` (void)
Initialise le hacheur et lance les PWM avec alpha = 50%.
- void `stop` (void)
Arrête les moteurs.
- void `speed` (void)
Règle le rapport cyclique des PWM, avec une valeur entre 0 et 100%.
- void `changeSpeed` (int a)
- void `HAL_ADC_ConvCpltCallback` (ADC_HandleTypeDef *hadc)
Permet de réinitialiser le flag adcDmaFlag après l'exécution automatique de l'interruption dans le shell.
- int `main` (void)
Lance le Shell et initialise les périphériques.
- void `HAL_UART_RxCpltCallback` (UART_HandleTypeDef *huart)
- void `Error_Handler` (void)
This function is executed in case of error occurrence.

Variables

- ADC_HandleTypeDef `hadc1`
- DMA_HandleTypeDef `hdma_adc1`
- TIM_HandleTypeDef `htim1`
- TIM_HandleTypeDef `htim2`
- TIM_HandleTypeDef `htim3`
- UART_HandleTypeDef `huart2`
- uint8_t `adcDmaFlag` = 0
- uint32_t `adcRawValue` [1]
- uint8_t `prompt` [] = "user@Nucleo-STM32G474>>"
- uint8_t `started` []
- uint8_t `newline` [] = "\r\n"
- uint8_t `cmdNotFound` [] = "Command not found\r\n"
- uint32_t `uartRxReceived`
- uint8_t `uartRxBuffer` [UART_RX_BUFFER_SIZE]
- uint8_t `uartTxBuffer` [UART_TX_BUFFER_SIZE]
- uint8_t `powerOn` [] = "Allumage du moteur\r\n"
- uint8_t `powerOff` [] = "Extinction du moteur\r\n"
- int `a`
- int `adcValue`
- long `vitesse`
- int `PIFlag`
- int `PIUpdateFlag`
- float `i_n` [2]
- float `w_n` [2]

- float **i_consigne**
- float **v_consigne**
- uint16_t **AD_RES** = 0
- char **cmdBuffer** [CMD_BUFFER_SIZE]
- int **idx_cmd**
- char * **argv** [MAX_ARGS]
- int **argc** = 0
- char * **token**
- int **newCmdReady** = 0

4.1.1 Detailed Description

: Main program body

Attention

Copyright (c) 2022 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 Function Documentation

4.1.2.1 delete()

```
void delete (
    void )
```

Supprime le dernier caractère après avoir appuyé sur BackSpace

4.1.2.2 Error_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

None	
------	--

4.1.2.3 HAL_ADC_ConvCpltCallback()

```
void HAL_ADC_ConvCpltCallback (
    ADC_HandleTypeDef * hadc )
```

Permet de réinitialiser le flag `adcDmaFlag` après l'exécution automatique de l'interruption dans le shell.

Parameters

<i>hadc</i>	
-------------	--

4.1.2.4 help()

```
void help (
    void )
```

Présente les fonctions et leur utilités dans la console.

4.1.2.5 main()

```
int main (
    void )
```

Lance le Shell et initialise les périphériques.

4.1.2.6 new_carac()

```
void new_carac (
    void )
```

Ecrit le caractère tapé dans la console

4.1.2.7 speed()

```
void speed (
    void )
```

Règle le rapport cyclique des PWM, avec une valeur entre 0 et 100%.

4.1.2.8 start()

```
void start (
    void )
```

Initialise le hacheur et lance les PWM avec alpha = 50%.

4.1.2.9 stop()

```
void stop (
    void )
```

Arrête les moteurs.

4.1.2.10 SystemClock_Config()

```
void SystemClock_Config (
    void )
```

System Clock Configuration.

Return values

None	
------	--

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

4.1.3 Variable Documentation

4.1.3.1 started

```
uint8_t started[]
```

Initial value:

```
=
    "\r\n*-----*"
    "\r\n| Welcome on Nucleo-STM32G474 |"
    "\r\n*-----*"
    "\r\n"
```

4.1.3.2 vitesse

```
long vitesse
```

Valeur de courant relevée après conversion en ampères

4.2 stm32g4xx_hal_msp.c File Reference

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

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

Functions

- void [HAL_TIM_MspPostInit](#) (TIM_HandleTypeDef *htim)
- void [HAL_MspInit](#) (void)
- void [HAL_ADC_MspInit](#) (ADC_HandleTypeDef *hadc)
ADC MSP Initialization This function configures the hardware resources used in this example.
- void [HAL_ADC_MspDeInit](#) (ADC_HandleTypeDef *hadc)
ADC MSP De-Initialization This function freeze the hardware resources used in this example.
- void [HAL_TIM_Base_MspInit](#) (TIM_HandleTypeDef *htim_base)
TIM_Base MSP Initialization This function configures the hardware resources used in this example.
- void [HAL_TIM_Encoder_MspInit](#) (TIM_HandleTypeDef *htim_encoder)
TIM_Encoder MSP Initialization This function configures the hardware resources used in this example.
- void [HAL_TIM_Base_MspDeInit](#) (TIM_HandleTypeDef *htim_base)
TIM_Base MSP De-Initialization This function freeze the hardware resources used in this example.
- void [HAL_TIM_Encoder_MspDeInit](#) (TIM_HandleTypeDef *htim_encoder)
TIM_Encoder MSP De-Initialization This function freeze the hardware resources used in this example.
- 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.

Variables

- DMA_HandleTypeDef **hdma_adc1**

4.2.1 Detailed Description

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

Attention

Copyright (c) 2022 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 Function Documentation

4.2.2.1 HAL_ADC_MspDeInit()

```
void HAL_ADC_MspDeInit (
    ADC_HandleTypeDef * hadc )
```

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

Parameters

<i>hadc</i>	ADC handle pointer
-------------	--------------------

Return values

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

ADC1 GPIO Configuration PA0 ----> ADC1_IN1

4.2.2.2 HAL_ADC_MspInit()

```
void HAL_ADC_MspInit (
    ADC_HandleTypeDef * hadc )
```

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

Parameters

<i>hadc</i>	ADC handle pointer
-------------	--------------------

Return values

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

Initializes the peripherals clocks

ADC1 GPIO Configuration PA0 ----> ADC1_IN1

4.2.2.3 HAL_MspInit()

```
void HAL_MspInit (
    void )
```

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

4.2.2.4 HAL_TIM_Base_MspDeInit()

```
void HAL_TIM_Base_MspDeInit (
    TIM_HandleTypeDef * htim_base )
```

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

Parameters

<i>htim_base</i>	TIM_Base handle pointer
------------------	-------------------------

Return values

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

4.2.2.5 HAL_TIM_Base_MspInit()

```
void HAL_TIM_Base_MspInit (
    TIM_HandleTypeDef * htim_base )
```

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

Parameters

<i>htim_base</i>	TIM_Base handle pointer
------------------	-------------------------

Return values

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

4.2.2.6 HAL_TIM_Encoder_MspDeInit()

```
void HAL_TIM_Encoder_MspDeInit (
    TIM_HandleTypeDef * htim_encoder )
```

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

Parameters

<i>htim_encoder</i>	TIM_Encoder handle pointer
---------------------	----------------------------

Return values

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

TIM2 GPIO Configuration PA1 ----> TIM2_CH2 PA15 ----> TIM2_CH1

4.2.2.7 HAL_TIM_Encoder_MspInit()

```
void HAL_TIM_Encoder_MspInit (
    TIM_HandleTypeDef * htim_encoder )
```

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

Parameters

<i>htim_encoder</i>	TIM_Encoder handle pointer
---------------------	----------------------------

Return values

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

TIM2 GPIO Configuration PA1 ----> TIM2_CH2 PA15 ----> TIM2_CH1

4.2.2.8 HAL_TIM_MspPostInit()

```
void HAL_TIM_MspPostInit (
    TIM_HandleTypeDef * htim )
```

TIM1 GPIO Configuration PA8 ----> TIM1_CH1 PA9 ----> TIM1_CH2 PA11 ----> TIM1_CH1N PA12 ----> TIM1_CH2N

4.2.2.9 HAL_UART_MspDeInit()

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

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

Parameters

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

Return values

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

USART2 GPIO Configuration PA2 ----> USART2_TX PA3 ----> USART2_RX

4.2.2.10 HAL_UART_MspInit()

```
void HAL_UART_MspInit (
```

```
UART_HandleTypeDef * huart )
```

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

Parameters

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

Return values

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

Initializes the peripherals clocks

USART2 GPIO Configuration PA2 ----> USART2_TX PA3 ----> USART2_RX

4.3 stm32g4xx_it.c File Reference

Interrupt Service Routines.

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

Functions

- 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.
- void **DMA1_Channel1_IRQHandler** (void)
This function handles DMA1 channel1 global interrupt.
- void **TIM2_IRQHandler** (void)
This function handles TIM2 global interrupt.
- void **TIM3_IRQHandler** (void)
This function handles TIM3 global interrupt.
- void **USART2_IRQHandler** (void)
This function handles USART2 global interrupt / USART2 wake-up interrupt through EXTI line 26.
- void **EXTI15_10_IRQHandler** (void)
This function handles EXTI line[15:10] interrupts.

Variables

- int **PIFlag**
- uint32_t [vitesse](#)
- int **speedFlag**
- DMA_HandleTypeDef **hdma_adc1**
- TIM_HandleTypeDef **htim2**
- TIM_HandleTypeDef **htim3**
- UART_HandleTypeDef **huart2**

4.3.1 Detailed Description

Interrupt Service Routines.

Attention

Copyright (c) 2022 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.

4.3.2 Variable Documentation

4.3.2.1 vitesse

```
uint32_t vitesse [extern]
```

Valeur de courant relevée après conversion en ampères

4.4 syscalls.c File Reference

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>
```

Functions

- `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 Detailed Description

STM32CubeIDE Minimal System calls file.

Author

Auto-generated by STM32CubeIDE

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

Attention

Copyright (c) 2022 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 sysmem.c File Reference

STM32CubeIDE System Memory calls file.

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


Functions

- 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 Detailed Description

STM32CubeIDE System Memory calls file.

Author

Generated by STM32CubeIDE

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

Attention

Copyright (c) 2022 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 Function Documentation

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'.

Parameters

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

Returns

Pointer to allocated memory

4.6 system_stm32g4xx.c File Reference

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

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

Macros

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

Functions

- 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 Detailed Description

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

Author

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

=====
Attention

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.

Index

- [_sbrk](#)
 - [sysmem.c, 21](#)
- [CMSIS, 5](#)
- [delete](#)
 - [main.c, 11](#)
- [Error_Handler](#)
 - [main.c, 11](#)
- [HAL_ADC_ConvCpltCallback](#)
 - [main.c, 11](#)
- [HAL_ADC_MspDeInit](#)
 - [stm32g4xx_hal_msp.c, 15](#)
- [HAL_ADC_MspltInit](#)
 - [stm32g4xx_hal_msp.c, 15](#)
- [HAL_MspltInit](#)
 - [stm32g4xx_hal_msp.c, 15](#)
- [HAL_TIM_Base_MspDeInit](#)
 - [stm32g4xx_hal_msp.c, 15](#)
- [HAL_TIM_Base_MspltInit](#)
 - [stm32g4xx_hal_msp.c, 16](#)
- [HAL_TIM_Encoder_MspDeInit](#)
 - [stm32g4xx_hal_msp.c, 16](#)
- [HAL_TIM_Encoder_MspltInit](#)
 - [stm32g4xx_hal_msp.c, 17](#)
- [HAL_TIM_MspltPostInit](#)
 - [stm32g4xx_hal_msp.c, 17](#)
- [HAL_UART_MspDeInit](#)
 - [stm32g4xx_hal_msp.c, 17](#)
- [HAL_UART_MspltInit](#)
 - [stm32g4xx_hal_msp.c, 17](#)
- [help](#)
 - [main.c, 12](#)
- [HSE_VALUE](#)
 - [STM32G4xx_System_Private_Includes, 6](#)
- [HSI_VALUE](#)
 - [STM32G4xx_System_Private_Includes, 6](#)
- [main](#)
 - [main.c, 12](#)
- [main.c, 9](#)
 - [delete, 11](#)
 - [Error_Handler, 11](#)
 - [HAL_ADC_ConvCpltCallback, 11](#)
 - [help, 12](#)
 - [main, 12](#)
 - [new_carac, 12](#)
 - [speed, 12](#)
 - [start, 12](#)
 - [started, 13](#)
 - [stop, 13](#)
 - [SystemClock_Config, 13](#)
 - [vitesse, 13](#)
- [new_carac](#)
 - [main.c, 12](#)
- [speed](#)
 - [main.c, 12](#)
- [start](#)
 - [main.c, 12](#)
- [started](#)
 - [main.c, 13](#)
- [stm32g4xx_hal_msp.c, 14](#)
 - [HAL_ADC_MspDeInit, 15](#)
 - [HAL_ADC_MspltInit, 15](#)
 - [HAL_MspltInit, 15](#)
 - [HAL_TIM_Base_MspDeInit, 15](#)
 - [HAL_TIM_Base_MspltInit, 16](#)
 - [HAL_TIM_Encoder_MspDeInit, 16](#)
 - [HAL_TIM_Encoder_MspltInit, 17](#)
 - [HAL_TIM_MspltPostInit, 17](#)
 - [HAL_UART_MspDeInit, 17](#)
 - [HAL_UART_MspltInit, 17](#)
- [stm32g4xx_it.c, 18](#)
 - [vitesse, 19](#)
- [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](#)
- [stop](#)
 - [main.c, 13](#)
- [syscalls.c, 19](#)
- [sysmem.c, 20](#)
 - [_sbrk, 21](#)
- [system_stm32g4xx.c, 22](#)
- [SystemClock_Config](#)
 - [main.c, 13](#)
- [SystemCoreClockUpdate](#)
 - [STM32G4xx_System_Private_Functions, 7](#)
- [SystemInit](#)

STM32G4xx_System_Private_Functions, [7](#)

vitesse

main.c, [13](#)

stm32g4xx_it.c, [19](#)