My Project

Generated by Doxygen 1.9.1

1	Module Index	1
	1.1 Modules	1
2	Class Index	3
	2.1 Class List	3
3	File Index	5
	3.1 File List	5
4	Module Documentation	7
	4.1 CMSIS	7
	4.1.1 Detailed Description	8
	4.2 Stm32f4xx_system	8
	4.2.1 Detailed Description	9
	4.3 STM32F4xx_System_Private_Includes	9
	4.3.1 Detailed Description	9
	4.3.2 Macro Definition Documentation	9
	4.3.2.1 HSE_VALUE	9
	4.3.2.2 HSI_VALUE	9
	4.4 STM32F4xx_System_Private_TypesDefinitions	9
	4.5 STM32F4xx_System_Private_Defines	10
	4.6 STM32F4xx_System_Private_Macros	10
	4.7 STM32F4xx_System_Private_Variables	10
	4.7.1 Detailed Description	11
	4.8 STM32F4xx_System_Private_FunctionPrototypes	11
	4.9 STM32F4xx_System_Private_Functions	11
	4.9.1 Detailed Description	11
	4.9.2 Function Documentation	11
	4.9.2.1 SystemCoreClockUpdate()	12
	4.9.2.2 SystemInit()	12
5	Class Documentation	15
	5.1 gps Struct Reference	15
6	File Documentation	17
Ŭ	6.1 Core/Inc/main.h File Reference	17
	6.1.1 Detailed Description	19
	6.1.2 Function Documentation	19
	6.1.2.1 Error_Handler()	19
	6.2 Core/Inc/stm32f4xx_it.h File Reference	19
	6.2.1 Detailed Description	20 21
	6.3 Core/Src/stm32f4xx_hal_msp.c File Reference	
	6.3.1 Detailed Description	21
	6.3.2 Function Documentation	22

6.3.2.1 HAL_ETH_MspDeInit()	22
6.3.2.2 HAL_ETH_MspInit()	22
6.3.2.3 HAL_MspInit()	22
6.3.2.4 HAL_PCD_MspDeInit()	23
6.3.2.5 HAL_PCD_MspInit()	23
6.3.2.6 HAL_UART_MspDeInit()	23
6.3.2.7 HAL_UART_MspInit()	24
6.4 Core/Src/stm32f4xx_hal_timebase_tim.c File Reference	24
6.4.1 Detailed Description	25
6.4.2 Function Documentation	25
6.4.2.1 HAL_InitTick()	25
6.4.2.2 HAL_ResumeTick()	26
6.4.2.3 HAL_SuspendTick()	26
6.5 Core/Src/stm32f4xx_it.c File Reference	27
6.5.1 Detailed Description	28
6.6 Core/Src/syscalls.c File Reference	28
•	29
•	29
The second pro-	30
	30
	30
	31
6.8.1 Detailed Description	31
Index	33

# **Chapter 1**

# **Module Index**

# 1.1 Modules

Here is a list of all modules:

CMSIS	7
Stm32f4xx_system	8
STM32F4xx_System_Private_Includes	Ş
STM32F4xx_System_Private_TypesDefinitions	Ş
STM32F4xx_System_Private_Defines	10
STM32F4xx_System_Private_Macros	10
STM32F4xx_System_Private_Variables	10
STM32F4xx_System_Private_FunctionPrototypes	11
STM32F4xx System Private Functions	11

2 Module Index

# Chapter 2

# **Class Index**

# 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:	
gps	15

4 Class Index

# **Chapter 3**

# File Index

# 3.1 File List

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

Core/Inc/ <b>config.h</b>	??
Core/Inc/FreeRTOSConfig.h	??
Core/Inc/gps.h	??
Core/Inc/main.h	
: Header for main.c file. This file contains the common defines of the application	17
Core/Inc/stm32f4xx_hal_conf.h	??
Core/Inc/stm32f4xx_it.h	
This file contains the headers of the interrupt handlers	19
Core/Src/stm32f4xx_hal_msp.c	
This file provides code for the MSP Initialization and de-Initialization codes	21
Core/Src/stm32f4xx_hal_timebase_tim.c	
HAL time base based on the hardware TIM	24
Core/Src/stm32f4xx_it.c	
Interrupt Service Routines	27
Core/Src/syscalls.c	
STM32CubeIDE Minimal System calls file	28
Core/Src/sysmem.c	
STM32CubeIDE System Memory calls file	29
Core/Src/system_stm32f4xx.c	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File	31

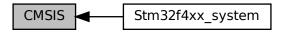
6 File Index

# **Chapter 4**

# **Module Documentation**

## 4.1 CMSIS

Collaboration diagram for CMSIS:



## **Modules**

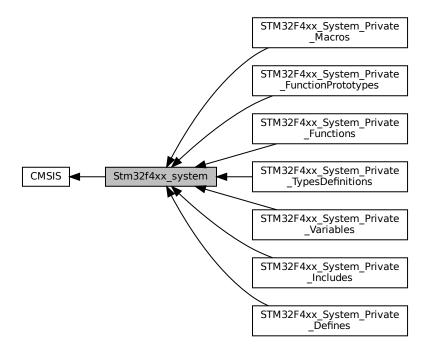
• Stm32f4xx\_system

8 Module Documentation

### 4.1.1 Detailed Description

## 4.2 Stm32f4xx\_system

Collaboration diagram for Stm32f4xx system:



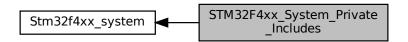
### **Modules**

- STM32F4xx\_System\_Private\_Includes
- STM32F4xx\_System\_Private\_TypesDefinitions
- STM32F4xx\_System\_Private\_Defines
- STM32F4xx\_System\_Private\_Macros
- STM32F4xx\_System\_Private\_Variables
- STM32F4xx\_System\_Private\_FunctionPrototypes
- STM32F4xx\_System\_Private\_Functions

#### 4.2.1 Detailed Description

# 4.3 STM32F4xx\_System\_Private\_Includes

Collaboration diagram for STM32F4xx\_System\_Private\_Includes:



#### **Macros**

- #define HSE\_VALUE ((uint32\_t)25000000)
- #define HSI\_VALUE ((uint32\_t)16000000)

### 4.3.1 Detailed Description

#### 4.3.2 Macro Definition Documentation

#### 4.3.2.1 HSE\_VALUE

#define HSE\_VALUE ((uint32\_t)25000000)

Default value of the External oscillator in Hz

#### 4.3.2.2 HSI\_VALUE

#define HSI\_VALUE ((uint32\_t)16000000)

Value of the Internal oscillator in Hz

# 4.4 STM32F4xx\_System\_Private\_TypesDefinitions

Collaboration diagram for STM32F4xx\_System\_Private\_TypesDefinitions:



10 Module Documentation

# 4.5 STM32F4xx\_System\_Private\_Defines

Collaboration diagram for STM32F4xx\_System\_Private\_Defines:



## 4.6 STM32F4xx\_System\_Private\_Macros

Collaboration diagram for STM32F4xx\_System\_Private\_Macros:



# 4.7 STM32F4xx\_System\_Private\_Variables

Collaboration diagram for STM32F4xx\_System\_Private\_Variables:



### **Variables**

- uint32\_t SystemCoreClock = 16000000
- const uint8\_t **AHBPrescTable** [16] =  $\{0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9\}$
- const uint8\_t **APBPrescTable** [8] = {0, 0, 0, 0, 1, 2, 3, 4}

#### 4.7.1 Detailed Description

# 4.8 STM32F4xx\_System\_Private\_FunctionPrototypes

Collaboration diagram for STM32F4xx System Private FunctionPrototypes:



## 4.9 STM32F4xx\_System\_Private\_Functions

Collaboration diagram for STM32F4xx\_System\_Private\_Functions:



### **Functions**

- void SystemInit (void)
  - Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.
- 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.

#### 4.9.1 Detailed Description

#### 4.9.2 Function Documentation

12 Module Documentation

#### 4.9.2.1 SystemCoreClockUpdate()

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 stm32f4xx\_hal\_conf.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 stm32f4xx\_hal\_conf.h file (its value depends on the application requirements), 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

#### 4.9.2.2 SystemInit()

```
void SystemInit (
     void )
```

Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.

Parameters		
None		
Return values		
None		

14 Module Documentation

# **Chapter 5**

# **Class Documentation**

# 5.1 gps Struct Reference

### **Public Attributes**

- char gps\_buffer [GPS\_BUFFER]
- uint8\_t \_longitude\_attitude
- float \_longitude
- uint8\_t \_latitude\_attitude
- float \_latitude

The documentation for this struct was generated from the following file:

· Core/Inc/gps.h

16 Class Documentation

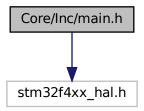
# **Chapter 6**

# **File Documentation**

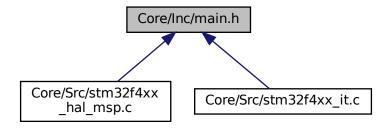
## 6.1 Core/Inc/main.h File Reference

: Header for main.c file. This file contains the common defines of the application.

#include "stm32f4xx\_hal.h"
Include dependency graph for main.h:



This graph shows which files directly or indirectly include this file:



#### **Macros**

- #define USER\_Btn\_Pin GPIO\_PIN\_13
- · #define USER Btn GPIO Port GPIOC
- #define MCO Pin GPIO PIN 0
- #define MCO GPIO Port GPIOH
- #define RMII MDC Pin GPIO PIN 1
- · #define RMII MDC GPIO Port GPIOC
- #define RMII\_REF\_CLK\_Pin GPIO\_PIN\_1
- · #define RMII REF CLK GPIO Port GPIOA
- #define RMII MDIO Pin GPIO PIN 2
- #define RMII MDIO GPIO Port GPIOA
- #define RMII CRS DV Pin GPIO PIN 7
- · #define RMII CRS DV GPIO Port GPIOA
- #define RMII\_RXD0\_Pin GPIO\_PIN\_4
- #define RMII\_RXD0\_GPIO\_Port GPIOC
- #define RMII RXD1 Pin GPIO PIN 5
- #define RMII\_RXD1\_GPIO\_Port GPIOC
- #define LD1\_Pin GPIO PIN 0
- #define LD1 GPIO Port GPIOB
- #define RMII TXD1 Pin GPIO PIN 13
- #define RMII TXD1 GPIO Port GPIOB
- #define LD3 Pin GPIO PIN 14
- #define LD3\_GPIO\_Port GPIOB
- #define STLK\_RX\_Pin GPIO\_PIN\_8
- · #define STLK RX GPIO Port GPIOD
- #define STLK TX Pin GPIO PIN 9
- #define STLK TX GPIO Port GPIOD
- #define USB\_PowerSwitchOn\_Pin GPIO\_PIN\_6
- #define USB PowerSwitchOn GPIO Port GPIOG
- #define USB OverCurrent Pin GPIO PIN 7
- #define USB OverCurrent GPIO Port GPIOG
- #define USB SOF Pin GPIO PIN 8
- · #define USB SOF GPIO Port GPIOA
- #define USB VBUS Pin GPIO PIN 9
- #define USB\_VBUS\_GPIO\_Port GPIOA
- #define USB ID Pin GPIO PIN 10
- · #define USB ID GPIO Port GPIOA
- #define USB DM Pin GPIO PIN 11
- #define USB\_DM\_GPIO\_Port GPIOA
- #define USB\_DP\_Pin GPIO\_PIN\_12
- #define USB\_DP\_GPIO\_Port GPIOA
- #define TMS\_Pin GPIO\_PIN\_13
- #define TMS\_GPIO\_Port GPIOA
- #define TCK\_Pin GPIO\_PIN\_14
- #define TCK GPIO Port GPIOA
- #define RMII\_TX\_EN\_Pin GPIO\_PIN\_11
- #define RMII\_TX\_EN\_GPIO\_Port GPIOG
- #define RMII TXD0 Pin GPIO PIN 13
- · #define RMII TXD0 GPIO Port GPIOG
- #define LD2 Pin GPIO PIN 7
- #define LD2 GPIO Port GPIOB

#### **Functions**

void Error\_Handler (void)

This function is executed in case of error occurrence.

### 6.1.1 Detailed Description

: Header for main.c file. This file contains the common defines of the application.

Attention

Copyright (c) 2024 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.

#### 6.1.2 Function Documentation

#### 6.1.2.1 Error\_Handler()

This function is executed in case of error occurrence.

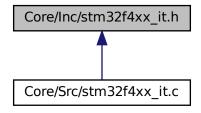
**Return values** 

None

# 6.2 Core/Inc/stm32f4xx\_it.h File Reference

This file contains the headers of the interrupt handlers.

This graph shows which files directly or indirectly include this file:



#### **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 Pre-fetch fault, memory access fault.

void UsageFault\_Handler (void)

This function handles Undefined instruction or illegal state.

void DebugMon\_Handler (void)

This function handles Debug monitor.

• void TIM1\_UP\_TIM10\_IRQHandler (void)

This function handles TIM1 update interrupt and TIM10 global interrupt.

#### 6.2.1 Detailed Description

This file contains the headers of the interrupt handlers.

Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

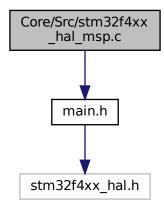
This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

## 6.3 Core/Src/stm32f4xx\_hal\_msp.c File Reference

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

#include "main.h"

Include dependency graph for stm32f4xx\_hal\_msp.c:



#### **Functions**

- void HAL\_MspInit (void)
- void HAL\_ETH\_MspInit (ETH\_HandleTypeDef \*heth)

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

• void HAL\_ETH\_MspDeInit (ETH\_HandleTypeDef \*heth)

ETH 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.

• void HAL\_PCD\_MspInit (PCD\_HandleTypeDef \*hpcd)

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

void HAL PCD MspDeInit (PCD HandleTypeDef \*hpcd)

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

#### 6.3.1 Detailed Description

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

Attention

Copyright (c) 2024 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.

## 6.3.2 Function Documentation

#### 6.3.2.1 HAL\_ETH\_MspDeInit()

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

#### **Parameters**

heth ETH handle pointer

#### Return values

None

ETH GPIO Configuration PC1 ----> ETH\_MDC PA1 ----> ETH\_REF\_CLK PA2 ----> ETH\_MDIO PA7 ----> ETH\_CRS\_DV PC4 ----> ETH\_RXD0 PC5 ----> ETH\_RXD1 PB13 ----> ETH\_TXD1 PG11 ----> ETH\_TXD1 PG13 ----> ETH\_TXD0

#### 6.3.2.2 HAL\_ETH\_MspInit()

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

#### **Parameters**

heth ETH handle pointer

#### Return values

None

ETH GPIO Configuration PC1 ----> ETH\_MDC PA1 ----> ETH\_REF\_CLK PA2 ----> ETH\_MDIO PA7 ----> ETH\_CRS\_DV PC4 ----> ETH\_RXD0 PC5 ----> ETH\_RXD1 PB13 ----> ETH\_TXD1 PG11 ----> ETH\_TXD1 PG13 ----> ETH\_TXD0

#### 6.3.2.3 HAL\_MspInit()

```
void HAL_MspInit (
     void )
```

Initializes the Global MSP.

#### 6.3.2.4 HAL\_PCD\_MspDeInit()

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

#### **Parameters**

hpcd PCD handle pointer

#### Return values

None

USB\_OTG\_FS GPIO Configuration PA8 ----> USB\_OTG\_FS\_SOF PA9 ----> USB\_OTG\_FS\_VBUS PA10 ----> USB\_OTG\_FS\_ID PA11 ----> USB\_OTG\_FS\_DM PA12 ----> USB\_OTG\_FS\_DP

#### 6.3.2.5 HAL\_PCD\_MspInit()

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

#### **Parameters**

hpcd PCD handle pointer

#### Return values

None

#### 6.3.2.6 HAL\_UART\_MspDeInit()

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

#### **Parameters**

huart UART handle pointer

#### Return values

None

USART3 GPIO Configuration PD8 ----> USART3\_TX PD9 ----> USART3\_RX

#### 6.3.2.7 HAL UART MspInit()

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

#### **Parameters**

huart UART handle pointer

#### **Return values**

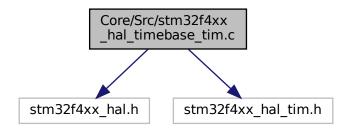
None

USART3 GPIO Configuration PD8 ----> USART3\_TX PD9 ----> USART3\_RX

# 6.4 Core/Src/stm32f4xx\_hal\_timebase\_tim.c File Reference

HAL time base based on the hardware TIM.

```
#include "stm32f4xx_hal.h"
#include "stm32f4xx_hal_tim.h"
Include dependency graph for stm32f4xx_hal_timebase_tim.c:
```



#### **Functions**

• HAL\_StatusTypeDef HAL\_InitTick (uint32\_t TickPriority)

This function configures the TIM1 as a time base source. The time source is configured to have 1ms time base with a dedicated Tick interrupt priority.

void HAL\_SuspendTick (void)

Suspend Tick increment.

• void HAL\_ResumeTick (void)

Resume Tick increment.

#### **Variables**

• TIM\_HandleTypeDef htim1

### 6.4.1 Detailed Description

HAL time base based on the hardware TIM.

Attention

Copyright (c) 2024 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.

#### 6.4.2 Function Documentation

#### 6.4.2.1 HAL\_InitTick()

This function configures the TIM1 as a time base source. The time source is configured to have 1ms time base with a dedicated Tick interrupt priority.

Note

This function is called automatically at the beginning of program after reset by HAL\_Init() or at any time when clock is configured, by HAL\_RCC\_ClockConfig().

#### **Parameters**

TickPriority	Tick interrupt priority.

Reti	11410	1/0	
Reli	ILU	Va	HIPS

#### 6.4.2.2 HAL\_ResumeTick()

```
void HAL_ResumeTick (
     void )
```

Resume Tick increment.

Note

Enable the tick increment by Enabling TIM1 update interrupt.

**Parameters** 

None

Return values

None

#### 6.4.2.3 HAL\_SuspendTick()

```
void HAL_SuspendTick (
     void )
```

Suspend Tick increment.

Note

Disable the tick increment by disabling TIM1 update interrupt.

**Parameters** 

None

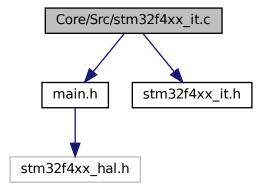
**Return values** 

None

## 6.5 Core/Src/stm32f4xx\_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"
#include "stm32f4xx_it.h"
Include dependency graph for stm32f4xx it.c:
```



#### **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 Pre-fetch fault, memory access fault.

void UsageFault\_Handler (void)

This function handles Undefined instruction or illegal state.

void DebugMon\_Handler (void)

This function handles Debug monitor.

• void TIM1\_UP\_TIM10\_IRQHandler (void)

This function handles TIM1 update interrupt and TIM10 global interrupt.

#### **Variables**

• TIM\_HandleTypeDef htim1

#### 6.5.1 Detailed Description

Interrupt Service Routines.

Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

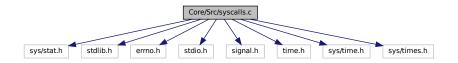
This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

### 6.6 Core/Src/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>
```

Include dependency graph for syscalls.c:



#### **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

### 6.6.1 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

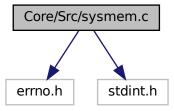
Copyright (c) 2020-2024 STMicroelectronics. All rights reserved.

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.

# 6.7 Core/Src/sysmem.c File Reference

STM32CubeIDE System Memory calls file.

#include <errno.h>
#include <stdint.h>
Include dependency graph for sysmem.c:



#### **Functions**

void \* \_sbrk (ptrdiff\_t incr)

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

#### 6.7.1 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) 2024 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.

#### 6.7.2 Function Documentation

#### 6.7.2.1 \_sbrk()

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

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**

```
incr Memory size
```

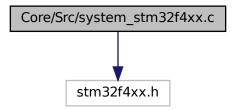
### Returns

Pointer to allocated memory

## 6.8 Core/Src/system stm32f4xx.c File Reference

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

#include "stm32f4xx.h"
Include dependency graph for system\_stm32f4xx.c:



#### **Macros**

- #define HSE\_VALUE ((uint32\_t)25000000)
- #define HSI\_VALUE ((uint32\_t)16000000)

#### **Functions**

void SystemInit (void)

Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.

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 = 16000000
- const uint8\_t **AHBPrescTable** [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
- const uint8\_t **APBPrescTable** [8] = {0, 0, 0, 0, 1, 2, 3, 4}

#### 6.8.1 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 stm32f4xx.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.

#### Attention

Copyright (c) 2017 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	HAL_PCD_MspInit, 23
sysmem.c, 30	HAL_UART_MspDeInit, 23
	HAL_UART_MspInit, 24
CMSIS, 7	stm32f4xx_hal_timebase_tim.c
Core/Inc/main.h, 17	HAL_InitTick, 25
Core/Inc/stm32f4xx_it.h, 19	HAL_ResumeTick, 26
Core/Src/stm32f4xx_hal_msp.c, 21	HAL_SuspendTick, 26
Core/Src/stm32f4xx_hal_timebase_tim.c, 24	Stm32f4xx_system, 8
Core/Src/stm32f4xx_it.c, 27	STM32F4xx_System_Private_Defines, 10
Core/Src/syscalls.c, 28	STM32F4xx_System_Private_FunctionPrototypes, 11
Core/Src/sysmem.c, 29	STM32F4xx_System_Private_Functions, 11
Core/Src/system_stm32f4xx.c, 31	SystemCoreClockUpdate, 11 SystemInit, 12
Error_Handler	STM32F4xx_System_Private_Includes, 9
main.h, 19	HSE_VALUE, 9 HSI_VALUE, 9
gps, 15	STM32F4xx System Private Macros, 10
	— · — — ·
HAL_ETH_MspDeInit	STM32F4xx_System_Private_TypesDefinitions, 9 STM32F4xx System Private Variables, 10
stm32f4xx_hal_msp.c, 22	— · — — —
HAL_ETH_MspInit	sysmem.c
stm32f4xx_hal_msp.c, 22	_sbrk, 30 SystemCoreClockUpdate
HAL_InitTick	·
stm32f4xx_hal_timebase_tim.c, 25	STM32F4xx_System_Private_Functions, 11
HAL_MspInit	SystemInit
stm32f4xx_hal_msp.c, 22	STM32F4xx_System_Private_Functions, 12
HAL_PCD_MspDeInit	
stm32f4xx_hal_msp.c, 22	
HAL_PCD_MspInit	
stm32f4xx_hal_msp.c, 23	
HAL_ResumeTick	
stm32f4xx_hal_timebase_tim.c, 26	
HAL_SuspendTick	
stm32f4xx_hal_timebase_tim.c, 26	
HAL_UART_MspDeInit	
stm32f4xx_hal_msp.c, 23	
HAL_UART_MspInit	
stm32f4xx_hal_msp.c, 24	
HSE_VALUE	
STM32F4xx_System_Private_Includes, 9	
HSI_VALUE	
STM32F4xx_System_Private_Includes, 9	
main.h	
Error_Handler, 19	
stm32f4xx_hal_msp.c	
HAL_ETH_MspDeInit, 22	
HAL_ETH_Msplnit, 22	
HAL_MspInit, 22	
HAL_PCD_MspDeInit, 22	