# VOLTMETER

Grzegorz Grund

Generated by Doxygen 1.9.6

# Chapter 1

# Module Index

## 1.1 Modules

Here is a list of all modules:

# Chapter 2

# File Index

## 2.1 File List

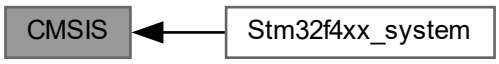
Here is a list of all files with brief descriptions:

# Chapter 3

# Module Documentation

## 3.1 CMSIS

Collaboration diagram for CMSIS:



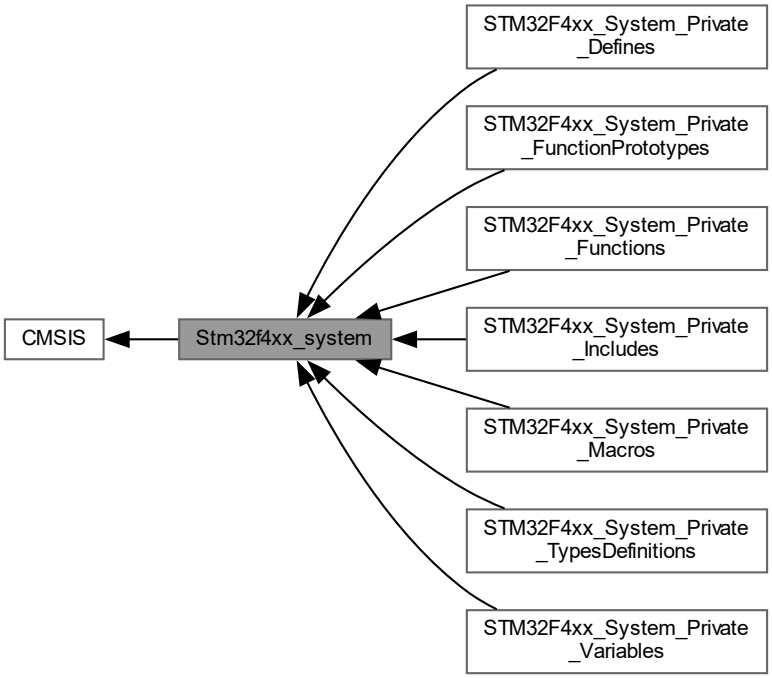

### Modules

- Stm32f4xx_system

### 3.1.1 Detailed Description

## 3.2 Stm32f4xx_system

Collaboration diagram for Stm32f4xx_system:



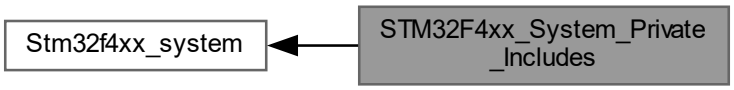

### Modules

- 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

### 3.2.1 Detailed Description

## 3.3 STM32F4xx_System_Private_Includes

Collaboration diagram for STM32F4xx_System_Private_Includes:




### Macros

- #define HSE_VALUE ((uint32_t)25000000)
- #define HSI_VALUE ((uint32_t)16000000)

### 3.3.1 Detailed Description

### 3.3.2 Macro Definition Documentation

#### 3.3.2.1 HSE_VALUE

```
#define HSE_VALUE ((uint32_t)25000000)
```

Default value of the External oscillator in Hz

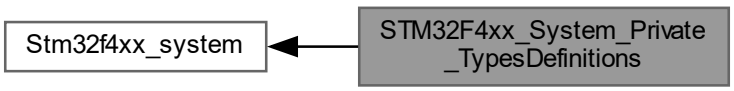
#### 3.3.2.2 HSI_VALUE

```
#define HSI_VALUE ((uint32_t)16000000)
```

Value of the Internal oscillator in Hz

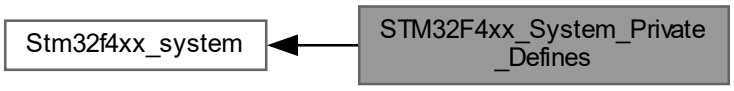
## 3.4 STM32F4xx_System_Private_TypesDefinitions

Collaboration diagram for STM32F4xx_System_Private_TypesDefinitions:



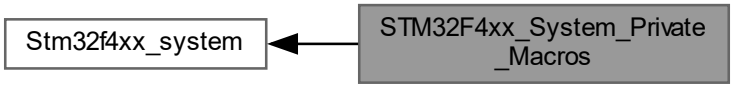

## 3.5 STM32F4xx_System_Private_Defines

Collaboration diagram for STM32F4xx_System_Private_Defines:



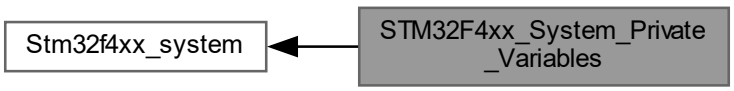

## 3.6 STM32F4xx_System_Private_Macros

Collaboration diagram for STM32F4xx_System_Private_Macros:



## 3.7 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}

### 3.7.1 Detailed Description

### 3.7.2 Variable Documentation

#### 3.7.2.1 AHBPrescTable

```
const uint8_t AHBPrescTable[16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
```

#### 3.7.2.2 APBPrescTable

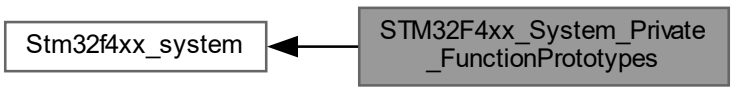
```
const uint8_t APBPrescTable[8] = {0, 0, 0, 0, 1, 2, 3, 4}
```

#### 3.7.2.3 SystemCoreClock

```
uint32_t SystemCoreClock = 16000000
```

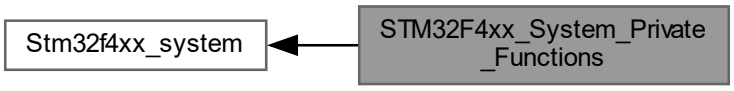
## 3.8 STM32F4xx_System_Private_FunctionPrototypes

Collaboration diagram for STM32F4xx_System_Private_FunctionPrototypes:




## 3.9 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.*

### 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 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* | |

#### 3.9.2.2 SystemInit()

```
void SystemInit (
            void  )
```

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

**Parameters**

| | |
|---|---|
| *None* | |

**Return values**

| | |
|---|---|
| *None* | |

# Chapter 4

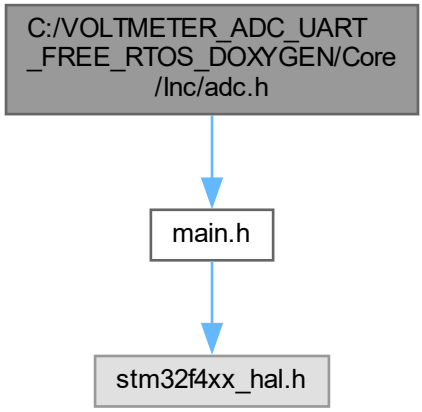
# File Documentation

## 4.1 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/adc.h File Reference

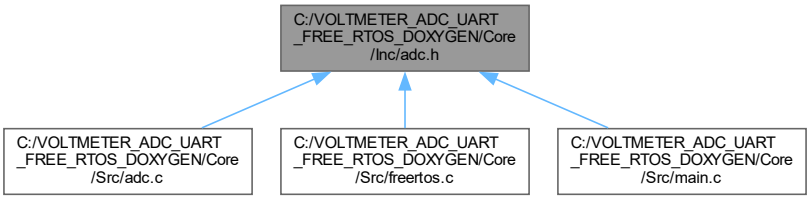
This file contains all the function prototypes for the adc.c file.

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

Include dependency graph for adc.h:




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



## Functions

- void MX_ADC1_Init (void)

## Variables

- ADC_HandleTypeDef hadc1

### 4.1.1 Detailed Description

This file contains all the function prototypes for the adc.c file.

**Attention**



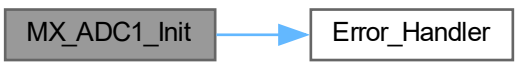
### 4.1.2 Function Documentation

#### 4.1.2.1 MX_ADC1_Init()

```
void MX_ADC1_Init (
             void  )
```

Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of conversion)

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time.Here is the call graph for this function:




Here is the caller graph for this function:

### 4.1.3 Variable Documentation

#### 4.1.3.1 hadc1

```
ADC_HandleTypeDef hadc1  [extern]
```

## 4.2 adc.h

Go to the documentation of this file.

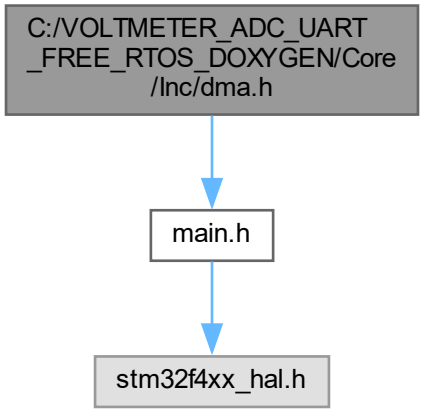
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __ADC_H__
00022 #define __ADC_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern ADC_HandleTypeDef hadc1;
00036
00037 /* USER CODE BEGIN Private defines */
00038
00039 /* USER CODE END Private defines */
00040
00041 void MX_ADC1_Init(void);
00042
00043 /* USER CODE BEGIN Prototypes */
00044
00045 /* USER CODE END Prototypes */
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* __ADC_H__ */
00052
```

## 4.3 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Inc/dma.h File Reference

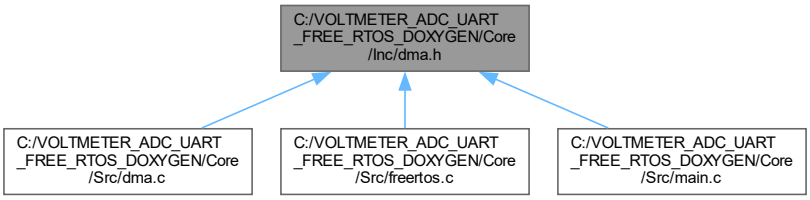
This file contains all the function prototypes for the dma.c file.

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

Include dependency graph for dma.h:




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




## Functions

- void MX_DMA_Init (void)

  *Enable DMA controller clock.*

### 4.3.1 Detailed Description

This file contains all the function prototypes for the dma.c file.

**Attention**

### 4.3.2 Function Documentation

#### 4.3.2.1 MX_DMA_Init()

```
void MX_DMA_Init (
             void  )
```

Enable DMA controller clock.

Here is the caller graph for this function:


## 4.4 dma.h

Go to the documentation of this file.

```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __DMA_H__
00022 #define __DMA_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/
00029 #include "main.h"
00030
00031 /* DMA memory to memory transfer handles -------------------------------------*/
00032
00033 /* USER CODE BEGIN Includes */
00034
00035 /* USER CODE END Includes */
00036
00037 /* USER CODE BEGIN Private defines */
00038
00039 /* USER CODE END Private defines */
00040
00041 void MX_DMA_Init(void);
00042
00043 /* USER CODE BEGIN Prototypes */
00044
00045 /* USER CODE END Prototypes */
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* __DMA_H__ */
00052
```

## 4.5 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/FreeRTOSConfig.h File Reference

### Macros

- #define configENABLE_FPU 0
- #define configENABLE_MPU 0
- #define configUSE_PREEMPTION 1
- #define configSUPPORT_STATIC_ALLOCATION 1
- #define configSUPPORT_DYNAMIC_ALLOCATION 1
- #define configUSE_IDLE_HOOK 1
- #define configUSE_TICK_HOOK 1
- #define configCPU_CLOCK_HZ ( SystemCoreClock )
- #define configTICK_RATE_HZ ((TickType_t)1000)
- #define configMAX_PRIORITIES ( 7 )
- #define configMINIMAL_STACK_SIZE ((uint16_t)128)
- #define configTOTAL_HEAP_SIZE ((size_t)15360)
- #define configMAX_TASK_NAME_LEN ( 16 )
- #define configUSE_16_BIT_TICKS 0
- #define configUSE_MUTEXES 1
- #define configQUEUE_REGISTRY_SIZE 8
- #define configUSE_PORT_OPTIMISED_TASK_SELECTION 1
- #define configMESSAGE_BUFFER_LENGTH_TYPE size_t
- #define configUSE_CO_ROUTINES 0
- #define configMAX_CO_ROUTINE_PRIORITIES ( 2 )
- #define configUSE_TIMERS 1
- #define configTIMER_TASK_PRIORITY ( 2 )
- #define configTIMER_QUEUE_LENGTH 10
- #define configTIMER_TASK_STACK_DEPTH 256
- #define INCLUDE_vTaskPrioritySet 1
- #define INCLUDE_uxTaskPriorityGet 1
- #define INCLUDE_vTaskDelete 1
- #define INCLUDE_vTaskCleanUpResources 0
- #define INCLUDE_vTaskSuspend 1
- #define INCLUDE_vTaskDelayUntil 0
- #define INCLUDE_vTaskDelay 1
- #define INCLUDE_xTaskGetSchedulerState 1
- #define configPRIO_BITS 4
- #define configLIBRARY_LOWEST_INTERRUPT_PRIORITY 15
- #define configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY 5
- #define configKERNEL_INTERRUPT_PRIORITY ( configLIBRARY_LOWEST_INTERRUPT_PRIORITY << (8 - configPRIO_BITS) )
- #define configMAX_SYSCALL_INTERRUPT_PRIORITY ( configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY << (8 - configPRIO_BITS) )
- #define configASSERT(x) if ((x) == 0) {taskDISABLE_INTERRUPTS(); for( ;; );}
- #define vPortSVCHandler SVC_Handler
- #define xPortPendSVHandler PendSV_Handler
- #define xPortSysTickHandler SysTick_Handler

### 4.5.1 Macro Definition Documentation

#### 4.5.1.1 configASSERT

```
#define configASSERT(
             x ) if ((x) == 0) {taskDISABLE_INTERRUPTS(); for( ;; );}
```

#### 4.5.1.2 configCPU_CLOCK_HZ

```
#define configCPU_CLOCK_HZ ( SystemCoreClock )
```

#### 4.5.1.3 configENABLE_FPU

```
#define configENABLE_FPU 0
```

#### 4.5.1.4 configENABLE_MPU

```
#define configENABLE_MPU 0
```

#### 4.5.1.5 configKERNEL_INTERRUPT_PRIORITY

```
#define configKERNEL_INTERRUPT_PRIORITY ( configLIBRARY_LOWEST_INTERRUPT_PRIORITY << (8 -
configPRIO_BITS) )
```

#### 4.5.1.6 configLIBRARY_LOWEST_INTERRUPT_PRIORITY

```
#define configLIBRARY_LOWEST_INTERRUPT_PRIORITY 15
```

#### 4.5.1.7 configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY

```
#define configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY 5
```

#### 4.5.1.8 configMAX_CO_ROUTINE_PRIORITIES

```
#define configMAX_CO_ROUTINE_PRIORITIES ( 2 )
```

#### 4.5.1.9 configMAX_PRIORITIES

```
#define configMAX_PRIORITIES ( 7 )
```

#### 4.5.1.10 configMAX_SYSCALL_INTERRUPT_PRIORITY

```
#define configMAX_SYSCALL_INTERRUPT_PRIORITY ( configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY <<
(8 - configPRIO_BITS) )
```

#### 4.5.1.11 configMAX_TASK_NAME_LEN

```
#define configMAX_TASK_NAME_LEN ( 16 )
```

#### 4.5.1.12 configMESSAGE_BUFFER_LENGTH_TYPE

```
#define configMESSAGE_BUFFER_LENGTH_TYPE size_t
```

#### 4.5.1.13 configMINIMAL_STACK_SIZE

```
#define configMINIMAL_STACK_SIZE ((uint16_t)128)
```

#### 4.5.1.14 configPRIO_BITS

```
#define configPRIO_BITS 4
```

#### 4.5.1.15 configQUEUE_REGISTRY_SIZE

```
#define configQUEUE_REGISTRY_SIZE 8
```

#### 4.5.1.16 configSUPPORT_DYNAMIC_ALLOCATION

```
#define configSUPPORT_DYNAMIC_ALLOCATION 1
```

#### 4.5.1.17 configSUPPORT_STATIC_ALLOCATION

```
#define configSUPPORT_STATIC_ALLOCATION 1
```

#### 4.5.1.18 configTICK_RATE_HZ

```
#define configTICK_RATE_HZ ((TickType_t)1000)
```

#### 4.5.1.19 configTIMER_QUEUE_LENGTH

```
#define configTIMER_QUEUE_LENGTH 10
```

#### 4.5.1.20 configTIMER_TASK_PRIORITY

```
#define configTIMER_TASK_PRIORITY ( 2 )
```

#### 4.5.1.21 configTIMER_TASK_STACK_DEPTH

```
#define configTIMER_TASK_STACK_DEPTH 256
```

#### 4.5.1.22 configTOTAL_HEAP_SIZE

```
#define configTOTAL_HEAP_SIZE ((size_t)15360)
```

#### 4.5.1.23 configUSE_16_BIT_TICKS

```
#define configUSE_16_BIT_TICKS 0
```

#### 4.5.1.24 configUSE_CO_ROUTINES

```
#define configUSE_CO_ROUTINES 0
```

#### 4.5.1.25 configUSE_IDLE_HOOK

```
#define configUSE_IDLE_HOOK 1
```

#### 4.5.1.26 configUSE_MUTEXES

```
#define configUSE_MUTEXES 1
```

#### 4.5.1.27 configUSE_PORT_OPTIMISED_TASK_SELECTION

```
#define configUSE_PORT_OPTIMISED_TASK_SELECTION 1
```

#### 4.5.1.28 configUSE_PREEMPTION

```
#define configUSE_PREEMPTION 1
```

#### 4.5.1.29 configUSE_TICK_HOOK

```
#define configUSE_TICK_HOOK 1
```

#### 4.5.1.30 configUSE_TIMERS

```
#define configUSE_TIMERS 1
```

#### 4.5.1.31 INCLUDE_uxTaskPriorityGet

```
#define INCLUDE_uxTaskPriorityGet 1
```

#### 4.5.1.32 INCLUDE_vTaskCleanUpResources

```
#define INCLUDE_vTaskCleanUpResources 0
```

#### 4.5.1.33 INCLUDE_vTaskDelay

```
#define INCLUDE_vTaskDelay 1
```

#### 4.5.1.34 INCLUDE_vTaskDelayUntil

```
#define INCLUDE_vTaskDelayUntil 0
```

#### 4.5.1.35 INCLUDE_vTaskDelete

```
#define INCLUDE_vTaskDelete 1
```

#### 4.5.1.36 INCLUDE_vTaskPrioritySet

```
#define INCLUDE_vTaskPrioritySet 1
```

#### 4.5.1.37 INCLUDE_vTaskSuspend

```
#define INCLUDE_vTaskSuspend 1
```

#### 4.5.1.38 INCLUDE_xTaskGetSchedulerState

```
#define INCLUDE_xTaskGetSchedulerState 1
```

#### 4.5.1.39 vPortSVCHandler

```
#define vPortSVCHandler SVC_Handler
```

#### 4.5.1.40 xPortPendSVHandler

```
#define xPortPendSVHandler PendSV_Handler
```

#### 4.5.1.41 xPortSysTickHandler

```
#define xPortSysTickHandler SysTick_Handler
```

## 4.6 FreeRTOSConfig.h

Go to the documentation of this file.

```
00001 /* USER CODE BEGIN Header */
00002 /*
00003  * FreeRTOS Kernel V10.3.1
00004  * Portion Copyright (C) 2017 Amazon.com, Inc. or its affiliates.  All Rights Reserved.
00005  * Portion Copyright (C) 2019 StMicroelectronics, Inc.  All Rights Reserved.
00006  *
00007  * Permission is hereby granted, free of charge, to any person obtaining a copy of
00008  * this software and associated documentation files (the "Software"), to deal in
00009  * the Software without restriction, including without limitation the rights to
00010  * use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
00011  * the Software, and to permit persons to whom the Software is furnished to do so,
00012  * subject to the following conditions:
00013  *
00014  * The above copyright notice and this permission notice shall be included in all
00015  * copies or substantial portions of the Software.
00016  *
00017  * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
00019  * FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
00020  * COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
00021  * IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
00022  * CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
00023  *
00024  * http://www.FreeRTOS.org
00025  * http://aws.amazon.com/freertos
00026  *
00027  * 1 tab == 4 spaces!
00028  */
00029 /* USER CODE END Header */
00030
00031 #ifndef FREERTOS_CONFIG_H
00032 #define FREERTOS_CONFIG_H
00033
00034 /*-----------------------------------------------------------
00035  * Application specific definitions.
00036  *
00037  * These definitions should be adjusted for your particular hardware and
00038  * application requirements.
00039  *
00040  * These parameters and more are described within the 'configuration' section of the
00041  * FreeRTOS API documentation available on the FreeRTOS.org web site.
00042  *
00043  * See http://www.freertos.org/a00110.html
00044  *----------------------------------------------------------*/
00045
00046 /* USER CODE BEGIN Includes */
00047 /* Section where include file can be added */
00048 /* USER CODE END Includes */
00049
```

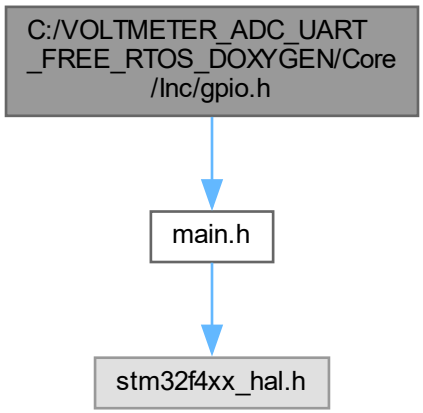
```
00050 /* Ensure definitions are only used by the compiler, and not by the assembler. */
00051 #if defined(__ICCARM__) || defined(__CC_ARM) || defined(__GNUC__)
00052   #include <stdint.h>
00053   extern uint32_t SystemCoreClock;
00054 #endif
00055 #define configENABLE_FPU                        0
00056 #define configENABLE_MPU                        0
00057
00058 #define configUSE_PREEMPTION                    1
00059 #define configSUPPORT_STATIC_ALLOCATION         1
00060 #define configSUPPORT_DYNAMIC_ALLOCATION        1
00061 #define configUSE_IDLE_HOOK                     1
00062 #define configUSE_TICK_HOOK                     1
00063 #define configCPU_CLOCK_HZ                      ( SystemCoreClock )
00064 #define configTICK_RATE_HZ                      ((TickType_t)1000)
00065 #define configMAX_PRIORITIES                    ( 7 )
00066 #define configMINIMAL_STACK_SIZE                ((uint16_t)128)
00067 #define configTOTAL_HEAP_SIZE                   ((size_t)15360)
00068 #define configMAX_TASK_NAME_LEN                 ( 16 )
00069 #define configUSE_16_BIT_TICKS                  0
00070 #define configUSE_MUTEXES                       1
00071 #define configQUEUE_REGISTRY_SIZE               8
00072 #define configUSE_PORT_OPTIMISED_TASK_SELECTION 1
00073 /* USER CODE BEGIN MESSAGE_BUFFER_LENGTH_TYPE */
00074 /* Defaults to size_t for backward compatibility, but can be changed
00075    if lengths will always be less than the number of bytes in a size_t. */
00076 #define configMESSAGE_BUFFER_LENGTH_TYPE        size_t
00077 /* USER CODE END MESSAGE_BUFFER_LENGTH_TYPE */
00078
00079 /* Co-routine definitions. */
00080 #define configUSE_CO_ROUTINES                   0
00081 #define configMAX_CO_ROUTINE_PRIORITIES         ( 2 )
00082
00083 /* Software timer definitions. */
00084 #define configUSE_TIMERS                        1
00085 #define configTIMER_TASK_PRIORITY               ( 2 )
00086 #define configTIMER_QUEUE_LENGTH                10
00087 #define configTIMER_TASK_STACK_DEPTH            256
00088
00089 /* Set the following definitions to 1 to include the API function, or zero
00090 to exclude the API function. */
00091 #define INCLUDE_vTaskPrioritySet            1
00092 #define INCLUDE_uxTaskPriorityGet           1
00093 #define INCLUDE_vTaskDelete                 1
00094 #define INCLUDE_vTaskCleanUpResources       0
00095 #define INCLUDE_vTaskSuspend                1
00096 #define INCLUDE_vTaskDelayUntil             0
00097 #define INCLUDE_vTaskDelay                  1
00098 #define INCLUDE_xTaskGetSchedulerState      1
00099
00100 /* Cortex-M specific definitions. */
00101 #ifdef __NVIC_PRIO_BITS
00102  /* __BVIC_PRIO_BITS will be specified when CMSIS is being used. */
00103  #define configPRIO_BITS         __NVIC_PRIO_BITS
00104 #else
00105  #define configPRIO_BITS         4
00106 #endif
00107
00108 /* The lowest interrupt priority that can be used in a call to a "set priority"
00109 function. */
00110 #define configLIBRARY_LOWEST_INTERRUPT_PRIORITY   15
00111
00112 /* The highest interrupt priority that can be used by any interrupt service
00113 routine that makes calls to interrupt safe FreeRTOS API functions.  DO NOT CALL
00114 INTERRUPT SAFE FREERTOS API FUNCTIONS FROM ANY INTERRUPT THAT HAS A HIGHER
00115 PRIORITY THAN THIS! (higher priorities are lower numeric values. */
00116 #define configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY 5
00117
00118 /* Interrupt priorities used by the kernel port layer itself.  These are generic
00119 to all Cortex-M ports, and do not rely on any particular library functions. */
00120 #define configKERNEL_INTERRUPT_PRIORITY         ( configLIBRARY_LOWEST_INTERRUPT_PRIORITY « (8 -
      configPRIO_BITS) )
00121 /* !!!! configMAX_SYSCALL_INTERRUPT_PRIORITY must not be set to zero !!!!
00122 See http://www.FreeRTOS.org/RTOS-Cortex-M3-M4.html. */
00123 #define configMAX_SYSCALL_INTERRUPT_PRIORITY    ( configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY « (8 -
      configPRIO_BITS) )
00124
00125 /* Normal assert() semantics without relying on the provision of an assert.h
00126 header file. */
00127 /* USER CODE BEGIN 1 */
00128 #define configASSERT( x ) if ((x) == 0) {taskDISABLE_INTERRUPTS(); for( ;; );}
00129 /* USER CODE END 1 */
00130
00131 /* Definitions that map the FreeRTOS port interrupt handlers to their CMSIS
00132 standard names. */
00133 #define vPortSVCHandler    SVC_Handler
00134 #define xPortPendSVHandler PendSV_Handler
```

```
00135
00136 /* IMPORTANT: This define is commented when used with STM32Cube firmware, when the timebase source is
      SysTick,
00137               to prevent overwriting SysTick_Handler defined within STM32Cube HAL */
00138
00139 #define xPortSysTickHandler SysTick_Handler
00140
00141 /* USER CODE BEGIN Defines */
00142 /* Section where parameter definitions can be added (for instance, to override default ones in
      FreeRTOS.h) */
00143 /* USER CODE END Defines */
00144
00145 #endif /* FREERTOS_CONFIG_H */
```

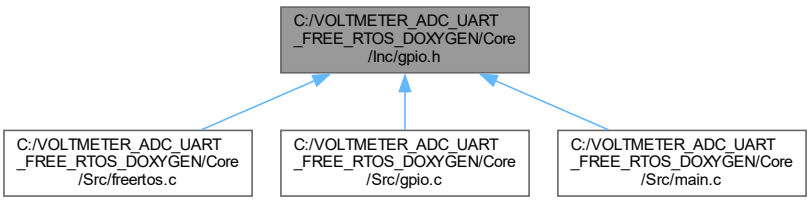
## 4.7 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Inc/gpio.h File Reference

This file contains all the function prototypes for the gpio.c file.

`#include "main.h"`
Include dependency graph for gpio.h:




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



## Functions

- void MX_GPIO_Init (void)

  *Configure pins.*

### 4.7.1 Detailed Description

This file contains all the function prototypes for the gpio.c file.

**Attention**



### 4.7.2 Function Documentation

#### 4.7.2.1 MX_GPIO_Init()

```
void MX_GPIO_Init (
             void  )
```

Configure pins.

Here is the caller graph for this function:

## 4.8 gpio.h

Go to the documentation of this file.

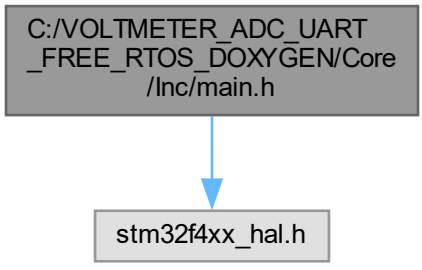
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __GPIO_H__
00022 #define __GPIO_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 /* USER CODE BEGIN Private defines */
00036
00037 /* USER CODE END Private defines */
00038
00039 void MX_GPIO_Init(void);
00040
00041 /* USER CODE BEGIN Prototypes */
00042
00043 /* USER CODE END Prototypes */
00044
00045 #ifdef __cplusplus
00046 }
00047 #endif
00048 #endif /*__ GPIO_H__ */
00049
```

## 4.9 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Inc/main.h File Reference

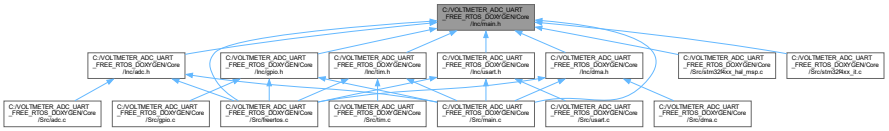
: Header for main.c file.

`#include "stm32f4xx_hal.h"`

Include dependency graph for main.h:



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



## Functions

- void Error_Handler (void)

  *This function is executed in case of error occurrence.*

### 4.9.1 Detailed Description

: Header for main.c file.

This file contains the common defines of the application.

**Attention**



### 4.9.2 Function Documentation

#### 4.9.2.1 Error_Handler()

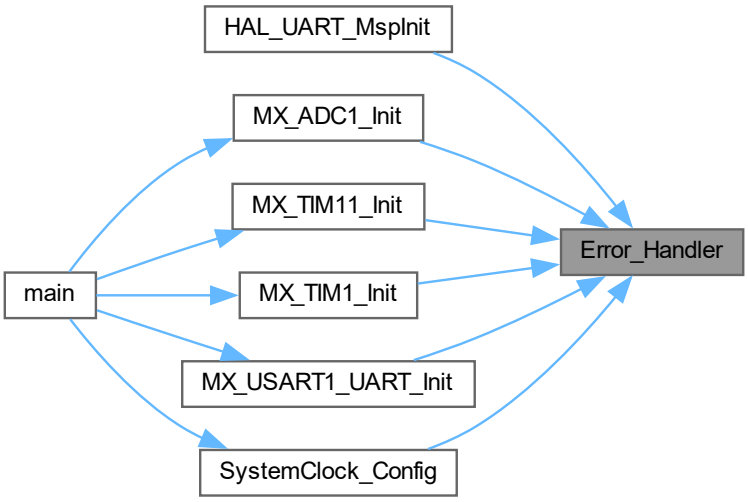
```
void Error_Handler (
             void  )
```

This function is executed in case of error occurrence.

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:




## 4.10 main.h

Go to the documentation of this file.

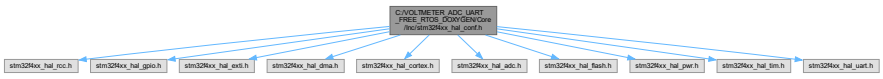
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020
00021 /* Define to prevent recursive inclusion -------------------------------------*/
00022 #ifndef __MAIN_H
00023 #define __MAIN_H
00024
00025 #ifdef __cplusplus
00026 extern "C" {
00027 #endif
00028
00029 /* Includes ------------------------------------------------------------------*/
00030 #include "stm32f4xx_hal.h"
00031
00032 /* Private includes ----------------------------------------------------------*/
00033 /* USER CODE BEGIN Includes */
00034
00035 /* USER CODE END Includes */
00036
00037 /* Exported types ------------------------------------------------------------*/
00038 /* USER CODE BEGIN ET */
00039
00040 /* USER CODE END ET */
00041
00042 /* Exported constants --------------------------------------------------------*/
00043 /* USER CODE BEGIN EC */
00044
00045 /* USER CODE END EC */
00046
00047 /* Exported macro ------------------------------------------------------------*/
00048 /* USER CODE BEGIN EM */
00049
00050 /* USER CODE END EM */
00051
00052 /* Exported functions prototypes ---------------------------------------------*/
00053 void Error_Handler(void);
00054
00055 /* USER CODE BEGIN EFP */
00056
00057 /* USER CODE END EFP */
00058
```

```
00059 /* Private defines -----------------------------------------------------------*/
00060
00061 /* USER CODE BEGIN Private defines */
00062
00063
00064
00065 /* USER CODE END Private defines */
00066
00067 #ifdef __cplusplus
00068 }
00069 #endif
00070
00071 #endif /* __MAIN_H */
```

## 4.11 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Inc/stm32f4xx_hal_conf.h File Reference

```
#include "stm32f4xx_hal_rcc.h"
#include "stm32f4xx_hal_gpio.h"
#include "stm32f4xx_hal_exti.h"
#include "stm32f4xx_hal_dma.h"
#include "stm32f4xx_hal_cortex.h"
#include "stm32f4xx_hal_adc.h"
#include "stm32f4xx_hal_flash.h"
#include "stm32f4xx_hal_pwr.h"
#include "stm32f4xx_hal_tim.h"
#include "stm32f4xx_hal_uart.h"
```

Include dependency graph for stm32f4xx_hal_conf.h:




### Macros

- #define HAL_MODULE_ENABLED

  *This is the list of modules to be used in the HAL driver.*
- #define HAL_ADC_MODULE_ENABLED
- #define HAL_TIM_MODULE_ENABLED
- #define HAL_UART_MODULE_ENABLED
- #define HAL_GPIO_MODULE_ENABLED
- #define HAL_EXTI_MODULE_ENABLED
- #define HAL_DMA_MODULE_ENABLED
- #define HAL_RCC_MODULE_ENABLED
- #define HAL_FLASH_MODULE_ENABLED
- #define HAL_PWR_MODULE_ENABLED
- #define HAL_CORTEX_MODULE_ENABLED
- #define HSE_VALUE 25000000U

  *Adjust the value of External High Speed oscillator (HSE) used in your application.*
- #define HSE_STARTUP_TIMEOUT 100U
- #define HSI_VALUE ((uint32_t)16000000U)

  *Internal High Speed oscillator (HSI) value.*
- #define LSI_VALUE 32000U

  *Internal Low Speed oscillator (LSI) value.*

- #define LSE_VALUE 32768U

  *External Low Speed oscillator (LSE) value.*
- #define LSE_STARTUP_TIMEOUT 5000U
- #define EXTERNAL_CLOCK_VALUE 12288000U

  *External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S_CKIN pad.*
- #define VDD_VALUE 3300U

  *This is the HAL system configuration section.*
- #define TICK_INT_PRIORITY 5U
- #define USE_RTOS 0U
- #define PREFETCH_ENABLE 1U
- #define INSTRUCTION_CACHE_ENABLE 1U
- #define DATA_CACHE_ENABLE 1U
- #define USE_HAL_ADC_REGISTER_CALLBACKS 0U /∗ ADC register callback disabled ∗/
- #define USE_HAL_CAN_REGISTER_CALLBACKS 0U /∗ CAN register callback disabled ∗/
- #define USE_HAL_CEC_REGISTER_CALLBACKS 0U /∗ CEC register callback disabled ∗/
- #define USE_HAL_CRYP_REGISTER_CALLBACKS 0U /∗ CRYP register callback disabled ∗/
- #define USE_HAL_DAC_REGISTER_CALLBACKS 0U /∗ DAC register callback disabled ∗/
- #define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /∗ DCMI register callback disabled ∗/
- #define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /∗ DFSDM register callback disabled ∗/
- #define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /∗ DMA2D register callback disabled ∗/
- #define USE_HAL_DSI_REGISTER_CALLBACKS 0U /∗ DSI register callback disabled ∗/
- #define USE_HAL_ETH_REGISTER_CALLBACKS 0U /∗ ETH register callback disabled ∗/
- #define USE_HAL_HASH_REGISTER_CALLBACKS 0U /∗ HASH register callback disabled ∗/
- #define USE_HAL_HCD_REGISTER_CALLBACKS 0U /∗ HCD register callback disabled ∗/
- #define USE_HAL_I2C_REGISTER_CALLBACKS 0U /∗ I2C register callback disabled ∗/
- #define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /∗ FMPI2C register callback disabled ∗/
- #define USE_HAL_FMPSMBUS_REGISTER_CALLBACKS 0U /∗ FMPSMBUS register callback disabled ∗/
- #define USE_HAL_I2S_REGISTER_CALLBACKS 0U /∗ I2S register callback disabled ∗/
- #define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /∗ IRDA register callback disabled ∗/
- #define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /∗ LPTIM register callback disabled ∗/
- #define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /∗ LTDC register callback disabled ∗/
- #define USE_HAL_MMC_REGISTER_CALLBACKS 0U /∗ MMC register callback disabled ∗/
- #define USE_HAL_NAND_REGISTER_CALLBACKS 0U /∗ NAND register callback disabled ∗/
- #define USE_HAL_NOR_REGISTER_CALLBACKS 0U /∗ NOR register callback disabled ∗/
- #define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /∗ PCCARD register callback disabled ∗/
- #define USE_HAL_PCD_REGISTER_CALLBACKS 0U /∗ PCD register callback disabled ∗/
- #define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /∗ QSPI register callback disabled ∗/
- #define USE_HAL_RNG_REGISTER_CALLBACKS 0U /∗ RNG register callback disabled ∗/
- #define USE_HAL_RTC_REGISTER_CALLBACKS 0U /∗ RTC register callback disabled ∗/
- #define USE_HAL_SAI_REGISTER_CALLBACKS 0U /∗ SAI register callback disabled ∗/
- #define USE_HAL_SD_REGISTER_CALLBACKS 0U /∗ SD register callback disabled ∗/
- #define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /∗ SMARTCARD register callback disabled ∗/
- #define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /∗ SDRAM register callback disabled ∗/
- #define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /∗ SRAM register callback disabled ∗/
- #define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /∗ SPDIFRX register callback disabled ∗/
- #define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /∗ SMBUS register callback disabled ∗/
- #define USE_HAL_SPI_REGISTER_CALLBACKS 0U /∗ SPI register callback disabled ∗/
- #define USE_HAL_TIM_REGISTER_CALLBACKS 0U /∗ TIM register callback disabled ∗/
- #define USE_HAL_UART_REGISTER_CALLBACKS 0U /∗ UART register callback disabled ∗/
- #define USE_HAL_USART_REGISTER_CALLBACKS 0U /∗ USART register callback disabled ∗/
- #define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /∗ WWDG register callback disabled ∗/
- #define MAC_ADDR0 2U

*Uncomment the line below to expanse the "assert_param" macro in the HAL drivers code.*

- #define MAC_ADDR1 0U
- #define MAC_ADDR2 0U
- #define MAC_ADDR3 0U
- #define MAC_ADDR4 0U
- #define MAC_ADDR5 0U
- #define ETH_RX_BUF_SIZE /∗ buffer size for receive ∗/
- #define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /∗ buffer size for transmit ∗/
- #define ETH_RXBUFNB 4U /∗ 4 Rx buffers of size ETH_RX_BUF_SIZE ∗/
- #define ETH_TXBUFNB 4U /∗ 4 Tx buffers of size ETH_TX_BUF_SIZE ∗/
- #define DP83848_PHY_ADDRESS 0x01U
- #define PHY_RESET_DELAY 0x000000FFU
- #define PHY_CONFIG_DELAY 0x00000FFFU
- #define PHY_READ_TO 0x0000FFFFU
- #define PHY_WRITE_TO 0x0000FFFFU
- #define PHY_BCR ((uint16_t)0x0000U)
- #define PHY_BSR ((uint16_t)0x0001U)
- #define PHY_RESET ((uint16_t)0x8000U)
- #define PHY_LOOPBACK ((uint16_t)0x4000U)
- #define PHY_FULLDUPLEX_100M ((uint16_t)0x2100U)
- #define PHY_HALFDUPLEX_100M ((uint16_t)0x2000U)
- #define PHY_FULLDUPLEX_10M ((uint16_t)0x0100U)
- #define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
- #define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
- #define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
- #define PHY_POWERDOWN ((uint16_t)0x0800U)
- #define PHY_ISOLATE ((uint16_t)0x0400U)
- #define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
- #define PHY_LINKED_STATUS ((uint16_t)0x0004U)
- #define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
- #define PHY_SR ((uint16_t)0x10U)
- #define PHY_SPEED_STATUS ((uint16_t)0x0002U)
- #define PHY_DUPLEX_STATUS ((uint16_t)0x0004U)
- #define USE_SPI_CRC 0U
- #define assert_param(expr) ((void)0U)

*Include module's header file.*

## 4.11.1 Macro Definition Documentation

### 4.11.1.1 assert_param

```
#define assert_param(
              expr ) ((void)0U)
```

Include module's header file.

#### 4.11.1.2 DATA_CACHE_ENABLE

```
#define DATA_CACHE_ENABLE 1U
```

#### 4.11.1.3 DP83848_PHY_ADDRESS

```
#define DP83848_PHY_ADDRESS 0x01U
```

#### 4.11.1.4 ETH_RX_BUF_SIZE

```
#define ETH_RX_BUF_SIZE /* buffer size for receive */
```

#### 4.11.1.5 ETH_RXBUFNB

```
#define ETH_RXBUFNB 4U /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
```

#### 4.11.1.6 ETH_TX_BUF_SIZE

```
#define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for transmit */
```

#### 4.11.1.7 ETH_TXBUFNB

```
#define ETH_TXBUFNB 4U /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
```

#### 4.11.1.8 EXTERNAL_CLOCK_VALUE

```
#define EXTERNAL_CLOCK_VALUE 12288000U
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S_CKIN pad.

Value of the External audio frequency in Hz

#### 4.11.1.9 HAL_ADC_MODULE_ENABLED

```
#define HAL_ADC_MODULE_ENABLED
```

#### 4.11.1.10 HAL_CORTEX_MODULE_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

#### 4.11.1.11 HAL_DMA_MODULE_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

#### 4.11.1.12 HAL_EXTI_MODULE_ENABLED

```
#define HAL_EXTI_MODULE_ENABLED
```

#### 4.11.1.13 HAL_FLASH_MODULE_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

#### 4.11.1.14 HAL_GPIO_MODULE_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

#### 4.11.1.15 HAL_MODULE_ENABLED

```
#define HAL_MODULE_ENABLED
```

This is the list of modules to be used in the HAL driver.

#### 4.11.1.16 HAL_PWR_MODULE_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

#### 4.11.1.17 HAL_RCC_MODULE_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

#### 4.11.1.18 HAL_TIM_MODULE_ENABLED

```
#define HAL_TIM_MODULE_ENABLED
```

#### 4.11.1.19 HAL_UART_MODULE_ENABLED

```
#define HAL_UART_MODULE_ENABLED
```

#### 4.11.1.20 HSE_STARTUP_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT 100U
```

Time out for HSE start up, in ms

#### 4.11.1.21 HSE_VALUE

```
#define HSE_VALUE 25000000U
```

Adjust the value of External High Speed oscillator (HSE) used in your application.

This value is used by the RCC HAL module to compute the system frequency (when HSE is used as system clock source, directly or through the PLL). Value of the External oscillator in Hz

#### 4.11.1.22 HSI_VALUE

```
#define HSI_VALUE ((uint32_t)16000000U)
```

Internal High Speed oscillator (HSI) value.

This value is used by the RCC HAL module to compute the system frequency (when HSI is used as system clock source, directly or through the PLL). Value of the Internal oscillator in Hz

#### 4.11.1.23 INSTRUCTION_CACHE_ENABLE

```
#define INSTRUCTION_CACHE_ENABLE 1U
```

#### 4.11.1.24 LSE_STARTUP_TIMEOUT

```
#define LSE_STARTUP_TIMEOUT 5000U
```

Time out for LSE start up, in ms

#### 4.11.1.25 LSE_VALUE

```
#define LSE_VALUE 32768U
```

External Low Speed oscillator (LSE) value.

< Value of the Internal Low Speed oscillator in Hz The real value may vary depending on the variations in voltage and temperature. Value of the External Low Speed oscillator in Hz

#### 4.11.1.26 LSI_VALUE

```
#define LSI_VALUE 32000U
```

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

#### 4.11.1.27 MAC_ADDR0

```
#define MAC_ADDR0 2U
```

Uncomment the line below to expanse the "assert_param" macro in the HAL drivers code.

#### 4.11.1.28 MAC_ADDR1

```
#define MAC_ADDR1 0U
```

#### 4.11.1.29 MAC_ADDR2

```
#define MAC_ADDR2 0U
```

#### 4.11.1.30 MAC_ADDR3

```
#define MAC_ADDR3 0U
```

#### 4.11.1.31 MAC_ADDR4

```
#define MAC_ADDR4 0U
```

#### 4.11.1.32 MAC_ADDR5

```
#define MAC_ADDR5 0U
```

#### 4.11.1.33 PHY_AUTONEGO_COMPLETE

```
#define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
```

Auto-Negotiation process completed

#### 4.11.1.34 PHY_AUTONEGOTIATION

```
#define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
```

Enable auto-negotiation function

#### 4.11.1.35 PHY_BCR

```
#define PHY_BCR ((uint16_t)0x0000U)
```

Transceiver Basic Control Register

#### 4.11.1.36 PHY_BSR

```
#define PHY_BSR ((uint16_t)0x0001U)
```

Transceiver Basic Status Register

#### 4.11.1.37 PHY_CONFIG_DELAY

```
#define PHY_CONFIG_DELAY 0x00000FFFU
```

#### 4.11.1.38 PHY_DUPLEX_STATUS

```
#define PHY_DUPLEX_STATUS ((uint16_t)0x0004U)
```

PHY Duplex mask

#### 4.11.1.39 PHY_FULLDUPLEX_100M

```
#define PHY_FULLDUPLEX_100M ((uint16_t)0x2100U)
```

Set the full-duplex mode at 100 Mb/s

#### 4.11.1.40 PHY_FULLDUPLEX_10M

```
#define PHY_FULLDUPLEX_10M ((uint16_t)0x0100U)
```

Set the full-duplex mode at 10 Mb/s

#### 4.11.1.41 PHY_HALFDUPLEX_100M

```
#define PHY_HALFDUPLEX_100M ((uint16_t)0x2000U)
```

Set the half-duplex mode at 100 Mb/s

#### 4.11.1.42 PHY_HALFDUPLEX_10M

```
#define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
```

Set the half-duplex mode at 10 Mb/s

#### 4.11.1.43 PHY_ISOLATE

```
#define PHY_ISOLATE ((uint16_t)0x0400U)
```

Isolate PHY from MII

#### 4.11.1.44 PHY_JABBER_DETECTION

```
#define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
```

Jabber condition detected

#### 4.11.1.45 PHY_LINKED_STATUS

```
#define PHY_LINKED_STATUS ((uint16_t)0x0004U)
```

Valid link established

#### 4.11.1.46 PHY_LOOPBACK

```
#define PHY_LOOPBACK ((uint16_t)0x4000U)
```

Select loop-back mode

#### 4.11.1.47 PHY_POWERDOWN

```
#define PHY_POWERDOWN ((uint16_t)0x0800U)
```

Select the power down mode

#### 4.11.1.48 PHY_READ_TO

```
#define PHY_READ_TO 0x0000FFFFU
```

#### 4.11.1.49 PHY_RESET

```
#define PHY_RESET ((uint16_t)0x8000U)
```

PHY Reset

#### 4.11.1.50 PHY_RESET_DELAY

```
#define PHY_RESET_DELAY 0x000000FFU
```

#### 4.11.1.51 PHY_RESTART_AUTONEGOTIATION

```
#define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
```

Restart auto-negotiation function

#### 4.11.1.52 PHY_SPEED_STATUS

```
#define PHY_SPEED_STATUS ((uint16_t)0x0002U)
```

PHY Speed mask

#### 4.11.1.53 PHY_SR

```
#define PHY_SR ((uint16_t)0x10U)
```

PHY status register Offset

#### 4.11.1.54 PHY_WRITE_TO

```
#define PHY_WRITE_TO 0x0000FFFFU
```

#### 4.11.1.55 PREFETCH_ENABLE

```
#define PREFETCH_ENABLE 1U
```

#### 4.11.1.56 TICK_INT_PRIORITY

```
#define TICK_INT_PRIORITY 5U
```

tick interrupt priority

#### 4.11.1.57 USE_HAL_ADC_REGISTER_CALLBACKS

```
#define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
```

#### 4.11.1.58 USE_HAL_CAN_REGISTER_CALLBACKS

```
#define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
```

#### 4.11.1.59 USE_HAL_CEC_REGISTER_CALLBACKS

```
#define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
```

#### 4.11.1.60 USE_HAL_CRYP_REGISTER_CALLBACKS

```
#define USE_HAL_CRYP_REGISTER_CALLBACKS 0U /* CRYP register callback disabled */
```

#### 4.11.1.61 USE_HAL_DAC_REGISTER_CALLBACKS

```
#define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
```

#### 4.11.1.62 USE_HAL_DCMI_REGISTER_CALLBACKS

```
#define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
```

#### 4.11.1.63 USE_HAL_DFSDM_REGISTER_CALLBACKS

```
#define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
```

#### 4.11.1.64 USE_HAL_DMA2D_REGISTER_CALLBACKS

```
#define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
```

#### 4.11.1.65 USE_HAL_DSI_REGISTER_CALLBACKS

```
#define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
```

#### 4.11.1.66 USE_HAL_ETH_REGISTER_CALLBACKS

```
#define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
```

#### 4.11.1.67 USE_HAL_FMPI2C_REGISTER_CALLBACKS

```
#define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /* FMPI2C register callback disabled */
```

#### 4.11.1.68 USE_HAL_FMPSMBUS_REGISTER_CALLBACKS

```
#define USE_HAL_FMPSMBUS_REGISTER_CALLBACKS 0U /* FMPSMBUS register callback disabled */
```

#### 4.11.1.69 USE_HAL_HASH_REGISTER_CALLBACKS

```
#define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
```

#### 4.11.1.70 USE_HAL_HCD_REGISTER_CALLBACKS

```
#define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
```

#### 4.11.1.71 USE_HAL_I2C_REGISTER_CALLBACKS

```
#define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
```

#### 4.11.1.72 USE_HAL_I2S_REGISTER_CALLBACKS

```
#define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
```

#### 4.11.1.73 USE_HAL_IRDA_REGISTER_CALLBACKS

```
#define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
```

#### 4.11.1.74 USE_HAL_LPTIM_REGISTER_CALLBACKS

```
#define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
```

#### 4.11.1.75 USE_HAL_LTDC_REGISTER_CALLBACKS

```
#define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
```

#### 4.11.1.76 USE_HAL_MMC_REGISTER_CALLBACKS

```
#define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
```

#### 4.11.1.77 USE_HAL_NAND_REGISTER_CALLBACKS

```
#define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
```

#### 4.11.1.78 USE_HAL_NOR_REGISTER_CALLBACKS

```
#define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
```

#### 4.11.1.79 USE_HAL_PCCARD_REGISTER_CALLBACKS

```
#define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /* PCCARD register callback disabled */
```

#### 4.11.1.80 USE_HAL_PCD_REGISTER_CALLBACKS

```
#define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
```

#### 4.11.1.81 USE_HAL_QSPI_REGISTER_CALLBACKS

```
#define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
```

#### 4.11.1.82 USE_HAL_RNG_REGISTER_CALLBACKS

```
#define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
```

#### 4.11.1.83 USE_HAL_RTC_REGISTER_CALLBACKS

```
#define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
```

#### 4.11.1.84 USE_HAL_SAI_REGISTER_CALLBACKS

```
#define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
```

#### 4.11.1.85 USE_HAL_SD_REGISTER_CALLBACKS

```
#define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
```

#### 4.11.1.86 USE_HAL_SDRAM_REGISTER_CALLBACKS

```
#define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
```

#### 4.11.1.87 USE_HAL_SMARTCARD_REGISTER_CALLBACKS

```
#define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
```

#### 4.11.1.88 USE_HAL_SMBUS_REGISTER_CALLBACKS

```
#define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
```

#### 4.11.1.89 USE_HAL_SPDIFRX_REGISTER_CALLBACKS

```
#define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
```

#### 4.11.1.90 USE_HAL_SPI_REGISTER_CALLBACKS

```
#define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
```

#### 4.11.1.91 USE_HAL_SRAM_REGISTER_CALLBACKS

```
#define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
```

#### 4.11.1.92 USE_HAL_TIM_REGISTER_CALLBACKS

```
#define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
```

#### 4.11.1.93 USE_HAL_UART_REGISTER_CALLBACKS

```
#define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
```

#### 4.11.1.94 USE_HAL_USART_REGISTER_CALLBACKS

```
#define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
```

#### 4.11.1.95 USE_HAL_WWDG_REGISTER_CALLBACKS

```
#define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
```

#### 4.11.1.96 USE_RTOS

```
#define USE_RTOS 0U
```

#### 4.11.1.97 USE_SPI_CRC

```
#define USE_SPI_CRC 0U
```

#### 4.11.1.98 VDD_VALUE

```
#define VDD_VALUE 3300U
```

This is the HAL system configuration section.

Value of VDD in mv

## 4.12 stm32f4xx_hal_conf.h

Go to the documentation of this file.

```
00001 /* USER CODE BEGIN Header */
00021 /* USER CODE END Header */
00022
00023 /* Define to prevent recursive inclusion -------------------------------------*/
00024 #ifndef __STM32F4xx_HAL_CONF_H
00025 #define __STM32F4xx_HAL_CONF_H
00026
00027 #ifdef __cplusplus
00028  extern "C" {
00029 #endif
00030
00031 /* Exported types ------------------------------------------------------------*/
00032 /* Exported constants --------------------------------------------------------*/
00033
00034 /* ########################## Module Selection ############################## */
00038 #define HAL_MODULE_ENABLED
00039
00040   /* #define HAL_CRYP_MODULE_ENABLED */
00041 #define HAL_ADC_MODULE_ENABLED
00042 /* #define HAL_CAN_MODULE_ENABLED */
00043 /* #define HAL_CRC_MODULE_ENABLED */
00044 /* #define HAL_CAN_LEGACY_MODULE_ENABLED */
00045 /* #define HAL_DAC_MODULE_ENABLED */
00046 /* #define HAL_DCMI_MODULE_ENABLED */
00047 /* #define HAL_DMA2D_MODULE_ENABLED */
00048 /* #define HAL_ETH_MODULE_ENABLED */
00049 /* #define HAL_NAND_MODULE_ENABLED */
00050 /* #define HAL_NOR_MODULE_ENABLED */
00051 /* #define HAL_PCCARD_MODULE_ENABLED */
00052 /* #define HAL_SRAM_MODULE_ENABLED */
00053 /* #define HAL_SDRAM_MODULE_ENABLED */
00054 /* #define HAL_HASH_MODULE_ENABLED */
00055 /* #define HAL_I2C_MODULE_ENABLED */
00056 /* #define HAL_I2S_MODULE_ENABLED */
00057 /* #define HAL_IWDG_MODULE_ENABLED */
00058 /* #define HAL_LTDC_MODULE_ENABLED */
00059 /* #define HAL_RNG_MODULE_ENABLED */
00060 /* #define HAL_RTC_MODULE_ENABLED */
00061 /* #define HAL_SAI_MODULE_ENABLED */
00062 /* #define HAL_SD_MODULE_ENABLED */
00063 /* #define HAL_MMC_MODULE_ENABLED */
00064 /* #define HAL_SPI_MODULE_ENABLED */
00065 #define HAL_TIM_MODULE_ENABLED
00066 #define HAL_UART_MODULE_ENABLED
00067 /* #define HAL_USART_MODULE_ENABLED */
00068 /* #define HAL_IRDA_MODULE_ENABLED */
00069 /* #define HAL_SMARTCARD_MODULE_ENABLED */
00070 /* #define HAL_SMBUS_MODULE_ENABLED */
00071 /* #define HAL_WWDG_MODULE_ENABLED */
00072 /* #define HAL_PCD_MODULE_ENABLED */
00073 /* #define HAL_HCD_MODULE_ENABLED */
00074 /* #define HAL_DSI_MODULE_ENABLED */
00075 /* #define HAL_QSPI_MODULE_ENABLED */
00076 /* #define HAL_QSPI_MODULE_ENABLED */
00077 /* #define HAL_CEC_MODULE_ENABLED */
00078 /* #define HAL_FMPI2C_MODULE_ENABLED */
00079 /* #define HAL_FMPSMBUS_MODULE_ENABLED */
00080 /* #define HAL_SPDIFRX_MODULE_ENABLED */
00081 /* #define HAL_DFSDM_MODULE_ENABLED */
00082 /* #define HAL_LPTIM_MODULE_ENABLED */
00083 #define HAL_GPIO_MODULE_ENABLED
00084 #define HAL_EXTI_MODULE_ENABLED
00085 #define HAL_DMA_MODULE_ENABLED
00086 #define HAL_RCC_MODULE_ENABLED
00087 #define HAL_FLASH_MODULE_ENABLED
00088 #define HAL_PWR_MODULE_ENABLED
```

```
00089 #define HAL_CORTEX_MODULE_ENABLED
00090
00091 /* ########################## HSE/HSI Values adaptation ##################### */
00097 #if !defined  (HSE_VALUE)
00098   #define HSE_VALUE    25000000U
00099 #endif /* HSE_VALUE */
00100
00101 #if !defined  (HSE_STARTUP_TIMEOUT)
00102   #define HSE_STARTUP_TIMEOUT    100U
00103 #endif /* HSE_STARTUP_TIMEOUT */
00104
00110 #if !defined  (HSI_VALUE)
00111   #define HSI_VALUE    ((uint32_t)16000000U)
00112 #endif /* HSI_VALUE */
00113
00117 #if !defined  (LSI_VALUE)
00118  #define LSI_VALUE  32000U
00119 #endif /* LSI_VALUE */
00125 #if !defined  (LSE_VALUE)
00126  #define LSE_VALUE  32768U
00127 #endif /* LSE_VALUE */
00128
00129 #if !defined  (LSE_STARTUP_TIMEOUT)
00130   #define LSE_STARTUP_TIMEOUT    5000U
00131 #endif /* LSE_STARTUP_TIMEOUT */
00132
00138 #if !defined  (EXTERNAL_CLOCK_VALUE)
00139   #define EXTERNAL_CLOCK_VALUE    12288000U
00140 #endif /* EXTERNAL_CLOCK_VALUE */
00141
00142 /* Tip: To avoid modifying this file each time you need to use different HSE,
00143    ===  you can define the HSE value in your toolchain compiler preprocessor. */
00144
00145 /* ########################### System Configuration ######################### */
00149 #define  VDD_VALUE            3300U
00150 #define  TICK_INT_PRIORITY            5U
00151 #define  USE_RTOS                     0U
00152 #define  PREFETCH_ENABLE              1U
00153 #define  INSTRUCTION_CACHE_ENABLE     1U
00154 #define  DATA_CACHE_ENABLE            1U
00155
00156 #define  USE_HAL_ADC_REGISTER_CALLBACKS         0U /* ADC register callback disabled       */
00157 #define  USE_HAL_CAN_REGISTER_CALLBACKS         0U /* CAN register callback disabled       */
00158 #define  USE_HAL_CEC_REGISTER_CALLBACKS         0U /* CEC register callback disabled       */
00159 #define  USE_HAL_CRYP_REGISTER_CALLBACKS        0U /* CRYP register callback disabled      */
00160 #define  USE_HAL_DAC_REGISTER_CALLBACKS         0U /* DAC register callback disabled       */
00161 #define  USE_HAL_DCMI_REGISTER_CALLBACKS        0U /* DCMI register callback disabled      */
00162 #define  USE_HAL_DFSDM_REGISTER_CALLBACKS       0U /* DFSDM register callback disabled     */
00163 #define  USE_HAL_DMA2D_REGISTER_CALLBACKS       0U /* DMA2D register callback disabled     */
00164 #define  USE_HAL_DSI_REGISTER_CALLBACKS         0U /* DSI register callback disabled       */
00165 #define  USE_HAL_ETH_REGISTER_CALLBACKS         0U /* ETH register callback disabled       */
00166 #define  USE_HAL_HASH_REGISTER_CALLBACKS        0U /* HASH register callback disabled      */
00167 #define  USE_HAL_HCD_REGISTER_CALLBACKS         0U /* HCD register callback disabled       */
00168 #define  USE_HAL_I2C_REGISTER_CALLBACKS         0U /* I2C register callback disabled       */
00169 #define  USE_HAL_FMPI2C_REGISTER_CALLBACKS      0U /* FMPI2C register callback disabled    */
00170 #define  USE_HAL_FMPSMBUS_REGISTER_CALLBACKS    0U /* FMPSMBUS register callback disabled  */
00171 #define  USE_HAL_I2S_REGISTER_CALLBACKS         0U /* I2S register callback disabled       */
00172 #define  USE_HAL_IRDA_REGISTER_CALLBACKS        0U /* IRDA register callback disabled      */
00173 #define  USE_HAL_LPTIM_REGISTER_CALLBACKS       0U /* LPTIM register callback disabled     */
00174 #define  USE_HAL_LTDC_REGISTER_CALLBACKS        0U /* LTDC register callback disabled      */
00175 #define  USE_HAL_MMC_REGISTER_CALLBACKS         0U /* MMC register callback disabled       */
00176 #define  USE_HAL_NAND_REGISTER_CALLBACKS        0U /* NAND register callback disabled      */
00177 #define  USE_HAL_NOR_REGISTER_CALLBACKS         0U /* NOR register callback disabled       */
00178 #define  USE_HAL_PCCARD_REGISTER_CALLBACKS      0U /* PCCARD register callback disabled    */
00179 #define  USE_HAL_PCD_REGISTER_CALLBACKS         0U /* PCD register callback disabled       */
00180 #define  USE_HAL_QSPI_REGISTER_CALLBACKS        0U /* QSPI register callback disabled      */
00181 #define  USE_HAL_RNG_REGISTER_CALLBACKS         0U /* RNG register callback disabled       */
00182 #define  USE_HAL_RTC_REGISTER_CALLBACKS         0U /* RTC register callback disabled       */
00183 #define  USE_HAL_SAI_REGISTER_CALLBACKS         0U /* SAI register callback disabled       */
00184 #define  USE_HAL_SD_REGISTER_CALLBACKS          0U /* SD register callback disabled        */
00185 #define  USE_HAL_SMARTCARD_REGISTER_CALLBACKS   0U /* SMARTCARD register callback disabled */
00186 #define  USE_HAL_SDRAM_REGISTER_CALLBACKS       0U /* SDRAM register callback disabled     */
00187 #define  USE_HAL_SRAM_REGISTER_CALLBACKS        0U /* SRAM register callback disabled      */
00188 #define  USE_HAL_SPDIFRX_REGISTER_CALLBACKS     0U /* SPDIFRX register callback disabled   */
00189 #define  USE_HAL_SMBUS_REGISTER_CALLBACKS       0U /* SMBUS register callback disabled     */
00190 #define  USE_HAL_SPI_REGISTER_CALLBACKS         0U /* SPI register callback disabled       */
00191 #define  USE_HAL_TIM_REGISTER_CALLBACKS         0U /* TIM register callback disabled       */
00192 #define  USE_HAL_UART_REGISTER_CALLBACKS        0U /* UART register callback disabled      */
00193 #define  USE_HAL_USART_REGISTER_CALLBACKS       0U /* USART register callback disabled     */
00194 #define  USE_HAL_WWDG_REGISTER_CALLBACKS        0U /* WWDG register callback disabled      */
00195
00196 /* ########################## Assert Selection ############################## */
00201 /* #define USE_FULL_ASSERT    1U */
00202
00203 /* ################### Ethernet peripheral configuration ##################### */
00204
00205 /* Section 1 : Ethernet peripheral configuration */
```

```
00206
00207 /* MAC ADDRESS: MAC_ADDR0:MAC_ADDR1:MAC_ADDR2:MAC_ADDR3:MAC_ADDR4:MAC_ADDR5 */
00208 #define MAC_ADDR0   2U
00209 #define MAC_ADDR1   0U
00210 #define MAC_ADDR2   0U
00211 #define MAC_ADDR3   0U
00212 #define MAC_ADDR4   0U
00213 #define MAC_ADDR5   0U
00214
00215 /* Definition of the Ethernet driver buffers size and count */
00216 #define ETH_RX_BUF_SIZE                /* buffer size for receive               */
00217 #define ETH_TX_BUF_SIZE               ETH_MAX_PACKET_SIZE /* buffer size for transmit             */
00218 #define ETH_RXBUFNB                   4U       /* 4 Rx buffers of size ETH_RX_BUF_SIZE  */
00219 #define ETH_TXBUFNB                   4U       /* 4 Tx buffers of size ETH_TX_BUF_SIZE  */
00220
00221 /* Section 2: PHY configuration section */
00222
00223 /* DP83848_PHY_ADDRESS Address*/
00224 #define DP83848_PHY_ADDRESS           0x01U
00225 /* PHY Reset delay these values are based on a 1 ms Systick interrupt*/
00226 #define PHY_RESET_DELAY                0x000000FFU
00227 /* PHY Configuration delay */
00228 #define PHY_CONFIG_DELAY               0x00000FFFU
00229
00230 #define PHY_READ_TO                    0x0000FFFFU
00231 #define PHY_WRITE_TO                   0x0000FFFFU
00232
00233 /* Section 3: Common PHY Registers */
00234
00235 #define PHY_BCR                        ((uint16_t)0x0000U)
00236 #define PHY_BSR                        ((uint16_t)0x0001U)
00238 #define PHY_RESET                      ((uint16_t)0x8000U)
00239 #define PHY_LOOPBACK                   ((uint16_t)0x4000U)
00240 #define PHY_FULLDUPLEX_100M            ((uint16_t)0x2100U)
00241 #define PHY_HALFDUPLEX_100M            ((uint16_t)0x2000U)
00242 #define PHY_FULLDUPLEX_10M             ((uint16_t)0x0100U)
00243 #define PHY_HALFDUPLEX_10M             ((uint16_t)0x0000U)
00244 #define PHY_AUTONEGOTIATION            ((uint16_t)0x1000U)
00245 #define PHY_RESTART_AUTONEGOTIATION    ((uint16_t)0x0200U)
00246 #define PHY_POWERDOWN                  ((uint16_t)0x0800U)
00247 #define PHY_ISOLATE                    ((uint16_t)0x0400U)
00249 #define PHY_AUTONEGO_COMPLETE          ((uint16_t)0x0020U)
00250 #define PHY_LINKED_STATUS              ((uint16_t)0x0004U)
00251 #define PHY_JABBER_DETECTION           ((uint16_t)0x0002U)
00253 /* Section 4: Extended PHY Registers */
00254 #define PHY_SR                         ((uint16_t)0x10U)
00256 #define PHY_SPEED_STATUS               ((uint16_t)0x0002U)
00257 #define PHY_DUPLEX_STATUS              ((uint16_t)0x0004U)
00259 /* ################### SPI peripheral configuration ########################### */
00260
00261 /* CRC FEATURE: Use to activate CRC feature inside HAL SPI Driver
00262 * Activated: CRC code is present inside driver
00263 * Deactivated: CRC code cleaned from driver
00264 */
00265
00266 #define USE_SPI_CRC                    0U
00267
00268 /* Includes ------------------------------------------------------------------*/
00273 #ifdef HAL_RCC_MODULE_ENABLED
00274   #include "stm32f4xx_hal_rcc.h"
00275 #endif /* HAL_RCC_MODULE_ENABLED */
00276
00277 #ifdef HAL_GPIO_MODULE_ENABLED
00278   #include "stm32f4xx_hal_gpio.h"
00279 #endif /* HAL_GPIO_MODULE_ENABLED */
00280
00281 #ifdef HAL_EXTI_MODULE_ENABLED
00282   #include "stm32f4xx_hal_exti.h"
00283 #endif /* HAL_EXTI_MODULE_ENABLED */
00284
00285 #ifdef HAL_DMA_MODULE_ENABLED
00286   #include "stm32f4xx_hal_dma.h"
00287 #endif /* HAL_DMA_MODULE_ENABLED */
00288
00289 #ifdef HAL_CORTEX_MODULE_ENABLED
00290   #include "stm32f4xx_hal_cortex.h"
00291 #endif /* HAL_CORTEX_MODULE_ENABLED */
00292
00293 #ifdef HAL_ADC_MODULE_ENABLED
00294   #include "stm32f4xx_hal_adc.h"
00295 #endif /* HAL_ADC_MODULE_ENABLED */
00296
00297 #ifdef HAL_CAN_MODULE_ENABLED
00298   #include "stm32f4xx_hal_can.h"
00299 #endif /* HAL_CAN_MODULE_ENABLED */
00300
00301 #ifdef HAL_CAN_LEGACY_MODULE_ENABLED
```

```
00302   #include "stm32f4xx_hal_can_legacy.h"
00303 #endif /* HAL_CAN_LEGACY_MODULE_ENABLED */
00304
00305 #ifdef HAL_CRC_MODULE_ENABLED
00306   #include "stm32f4xx_hal_crc.h"
00307 #endif /* HAL_CRC_MODULE_ENABLED */
00308
00309 #ifdef HAL_CRYP_MODULE_ENABLED
00310   #include "stm32f4xx_hal_cryp.h"
00311 #endif /* HAL_CRYP_MODULE_ENABLED */
00312
00313 #ifdef HAL_DMA2D_MODULE_ENABLED
00314   #include "stm32f4xx_hal_dma2d.h"
00315 #endif /* HAL_DMA2D_MODULE_ENABLED */
00316
00317 #ifdef HAL_DAC_MODULE_ENABLED
00318   #include "stm32f4xx_hal_dac.h"
00319 #endif /* HAL_DAC_MODULE_ENABLED */
00320
00321 #ifdef HAL_DCMI_MODULE_ENABLED
00322   #include "stm32f4xx_hal_dcmi.h"
00323 #endif /* HAL_DCMI_MODULE_ENABLED */
00324
00325 #ifdef HAL_ETH_MODULE_ENABLED
00326   #include "stm32f4xx_hal_eth.h"
00327 #endif /* HAL_ETH_MODULE_ENABLED */
00328
00329 #ifdef HAL_FLASH_MODULE_ENABLED
00330   #include "stm32f4xx_hal_flash.h"
00331 #endif /* HAL_FLASH_MODULE_ENABLED */
00332
00333 #ifdef HAL_SRAM_MODULE_ENABLED
00334   #include "stm32f4xx_hal_sram.h"
00335 #endif /* HAL_SRAM_MODULE_ENABLED */
00336
00337 #ifdef HAL_NOR_MODULE_ENABLED
00338   #include "stm32f4xx_hal_nor.h"
00339 #endif /* HAL_NOR_MODULE_ENABLED */
00340
00341 #ifdef HAL_NAND_MODULE_ENABLED
00342   #include "stm32f4xx_hal_nand.h"
00343 #endif /* HAL_NAND_MODULE_ENABLED */
00344
00345 #ifdef HAL_PCCARD_MODULE_ENABLED
00346   #include "stm32f4xx_hal_pccard.h"
00347 #endif /* HAL_PCCARD_MODULE_ENABLED */
00348
00349 #ifdef HAL_SDRAM_MODULE_ENABLED
00350   #include "stm32f4xx_hal_sdram.h"
00351 #endif /* HAL_SDRAM_MODULE_ENABLED */
00352
00353 #ifdef HAL_HASH_MODULE_ENABLED
00354  #include "stm32f4xx_hal_hash.h"
00355 #endif /* HAL_HASH_MODULE_ENABLED */
00356
00357 #ifdef HAL_I2C_MODULE_ENABLED
00358  #include "stm32f4xx_hal_i2c.h"
00359 #endif /* HAL_I2C_MODULE_ENABLED */
00360
00361 #ifdef HAL_SMBUS_MODULE_ENABLED
00362  #include "stm32f4xx_hal_smbus.h"
00363 #endif /* HAL_SMBUS_MODULE_ENABLED */
00364
00365 #ifdef HAL_I2S_MODULE_ENABLED
00366  #include "stm32f4xx_hal_i2s.h"
00367 #endif /* HAL_I2S_MODULE_ENABLED */
00368
00369 #ifdef HAL_IWDG_MODULE_ENABLED
00370  #include "stm32f4xx_hal_iwdg.h"
00371 #endif /* HAL_IWDG_MODULE_ENABLED */
00372
00373 #ifdef HAL_LTDC_MODULE_ENABLED
00374  #include "stm32f4xx_hal_ltdc.h"
00375 #endif /* HAL_LTDC_MODULE_ENABLED */
00376
00377 #ifdef HAL_PWR_MODULE_ENABLED
00378  #include "stm32f4xx_hal_pwr.h"
00379 #endif /* HAL_PWR_MODULE_ENABLED */
00380
00381 #ifdef HAL_RNG_MODULE_ENABLED
00382  #include "stm32f4xx_hal_rng.h"
00383 #endif /* HAL_RNG_MODULE_ENABLED */
00384
00385 #ifdef HAL_RTC_MODULE_ENABLED
00386  #include "stm32f4xx_hal_rtc.h"
00387 #endif /* HAL_RTC_MODULE_ENABLED */
00388
```

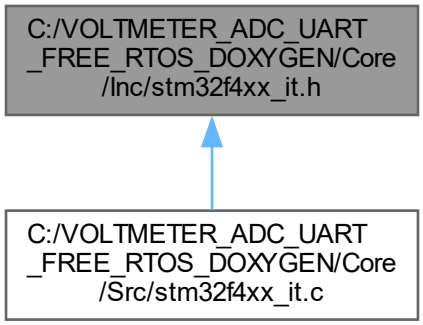
```
00389 #ifdef HAL_SAI_MODULE_ENABLED
00390  #include "stm32f4xx_hal_sai.h"
00391 #endif /* HAL_SAI_MODULE_ENABLED */
00392
00393 #ifdef HAL_SD_MODULE_ENABLED
00394  #include "stm32f4xx_hal_sd.h"
00395 #endif /* HAL_SD_MODULE_ENABLED */
00396
00397 #ifdef HAL_SPI_MODULE_ENABLED
00398  #include "stm32f4xx_hal_spi.h"
00399 #endif /* HAL_SPI_MODULE_ENABLED */
00400
00401 #ifdef HAL_TIM_MODULE_ENABLED
00402  #include "stm32f4xx_hal_tim.h"
00403 #endif /* HAL_TIM_MODULE_ENABLED */
00404
00405 #ifdef HAL_UART_MODULE_ENABLED
00406  #include "stm32f4xx_hal_uart.h"
00407 #endif /* HAL_UART_MODULE_ENABLED */
00408
00409 #ifdef HAL_USART_MODULE_ENABLED
00410  #include "stm32f4xx_hal_usart.h"
00411 #endif /* HAL_USART_MODULE_ENABLED */
00412
00413 #ifdef HAL_IRDA_MODULE_ENABLED
00414  #include "stm32f4xx_hal_irda.h"
00415 #endif /* HAL_IRDA_MODULE_ENABLED */
00416
00417 #ifdef HAL_SMARTCARD_MODULE_ENABLED
00418  #include "stm32f4xx_hal_smartcard.h"
00419 #endif /* HAL_SMARTCARD_MODULE_ENABLED */
00420
00421 #ifdef HAL_WWDG_MODULE_ENABLED
00422  #include "stm32f4xx_hal_wwdg.h"
00423 #endif /* HAL_WWDG_MODULE_ENABLED */
00424
00425 #ifdef HAL_PCD_MODULE_ENABLED
00426  #include "stm32f4xx_hal_pcd.h"
00427 #endif /* HAL_PCD_MODULE_ENABLED */
00428
00429 #ifdef HAL_HCD_MODULE_ENABLED
00430  #include "stm32f4xx_hal_hcd.h"
00431 #endif /* HAL_HCD_MODULE_ENABLED */
00432
00433 #ifdef HAL_DSI_MODULE_ENABLED
00434  #include "stm32f4xx_hal_dsi.h"
00435 #endif /* HAL_DSI_MODULE_ENABLED */
00436
00437 #ifdef HAL_QSPI_MODULE_ENABLED
00438  #include "stm32f4xx_hal_qspi.h"
00439 #endif /* HAL_QSPI_MODULE_ENABLED */
00440
00441 #ifdef HAL_CEC_MODULE_ENABLED
00442  #include "stm32f4xx_hal_cec.h"
00443 #endif /* HAL_CEC_MODULE_ENABLED */
00444
00445 #ifdef HAL_FMPI2C_MODULE_ENABLED
00446  #include "stm32f4xx_hal_fmpi2c.h"
00447 #endif /* HAL_FMPI2C_MODULE_ENABLED */
00448
00449 #ifdef HAL_FMPSMBUS_MODULE_ENABLED
00450  #include "stm32f4xx_hal_fmpsmbus.h"
00451 #endif /* HAL_FMPSMBUS_MODULE_ENABLED */
00452
00453 #ifdef HAL_SPDIFRX_MODULE_ENABLED
00454  #include "stm32f4xx_hal_spdifrx.h"
00455 #endif /* HAL_SPDIFRX_MODULE_ENABLED */
00456
00457 #ifdef HAL_DFSDM_MODULE_ENABLED
00458  #include "stm32f4xx_hal_dfsdm.h"
00459 #endif /* HAL_DFSDM_MODULE_ENABLED */
00460
00461 #ifdef HAL_LPTIM_MODULE_ENABLED
00462  #include "stm32f4xx_hal_lptim.h"
00463 #endif /* HAL_LPTIM_MODULE_ENABLED */
00464
00465 #ifdef HAL_MMC_MODULE_ENABLED
00466  #include "stm32f4xx_hal_mmc.h"
00467 #endif /* HAL_MMC_MODULE_ENABLED */
00468
00469 /* Exported macro ------------------------------------------------------------*/
00470 #ifdef  USE_FULL_ASSERT
00479   #define assert_param(expr) ((expr) ? (void)0U : assert_failed((uint8_t *)__FILE__, __LINE__))
00480 /* Exported functions ------------------------------------------------------- */
00481   void assert_failed(uint8_t* file, uint32_t line);
00482 #else
00483   #define assert_param(expr) ((void)0U)
```

```
00484 #endif /* USE_FULL_ASSERT */
00485
00486 #ifdef __cplusplus
00487 }
00488 #endif
00489
00490 #endif /* __STM32F4xx_HAL_CONF_H */
```

## 4.13 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/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 ADC_IRQHandler (void)

  *This function handles ADC1 global interrupt.*
- void TIM1_UP_TIM10_IRQHandler (void)

  *This function handles TIM1 update interrupt and TIM10 global interrupt.*
- void TIM1_TRG_COM_TIM11_IRQHandler (void)

  *This function handles TIM1 trigger and commutation interrupts and TIM11 global interrupt.*

- void USART1_IRQHandler (void)

  *This function handles USART1 global interrupt.*
- void DMA2_Stream2_IRQHandler (void)

  *This function handles DMA2 stream2 global interrupt.*
- void DMA2_Stream7_IRQHandler (void)

  *This function handles DMA2 stream7 global interrupt.*

## 4.13.1 Detailed Description

This file contains the headers of the interrupt handlers.

**Attention**



## 4.13.2 Function Documentation

### 4.13.2.1 ADC_IRQHandler()

```
void ADC_IRQHandler (
             void  )
```

This function handles ADC1 global interrupt.

### 4.13.2.2 BusFault_Handler()

```
void BusFault_Handler (
             void  )
```

This function handles Pre-fetch fault, memory access fault.

### 4.13.2.3 DebugMon_Handler()

```
void DebugMon_Handler (
             void  )
```

This function handles Debug monitor.

#### 4.13.2.4 DMA2_Stream2_IRQHandler()

```
void DMA2_Stream2_IRQHandler (
            void  )
```

This function handles DMA2 stream2 global interrupt.

#### 4.13.2.5 DMA2_Stream7_IRQHandler()

```
void DMA2_Stream7_IRQHandler (
            void  )
```

This function handles DMA2 stream7 global interrupt.

#### 4.13.2.6 HardFault_Handler()

```
void HardFault_Handler (
            void  )
```

This function handles Hard fault interrupt.

#### 4.13.2.7 MemManage_Handler()

```
void MemManage_Handler (
            void  )
```

This function handles Memory management fault.

#### 4.13.2.8 NMI_Handler()

```
void NMI_Handler (
            void  )
```

This function handles Non maskable interrupt.

#### 4.13.2.9 TIM1_TRG_COM_TIM11_IRQHandler()

```
void TIM1_TRG_COM_TIM11_IRQHandler (
            void  )
```

This function handles TIM1 trigger and commutation interrupts and TIM11 global interrupt.

#### 4.13.2.10 TIM1_UP_TIM10_IRQHandler()

```
void TIM1_UP_TIM10_IRQHandler (
             void  )
```

This function handles TIM1 update interrupt and TIM10 global interrupt.

#### 4.13.2.11 UsageFault_Handler()

```
void UsageFault_Handler (
             void  )
```

This function handles Undefined instruction or illegal state.

#### 4.13.2.12 USART1_IRQHandler()

```
void USART1_IRQHandler (
             void  )
```

This function handles USART1 global interrupt.

## 4.14 stm32f4xx_it.h

Go to the documentation of this file.

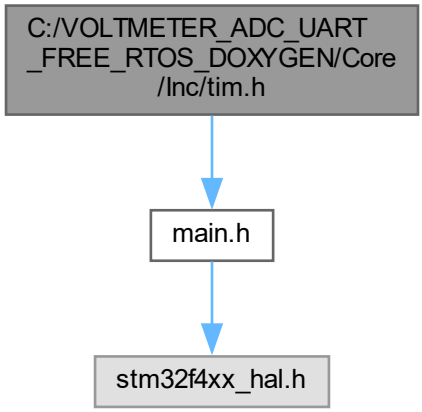
```
00001 /* USER CODE BEGIN Header */
00018 /* USER CODE END Header */
00019
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __STM32F4xx_IT_H
00022 #define __STM32F4xx_IT_H
00023
00024 #ifdef __cplusplus
00025  extern "C" {
00026 #endif
00027
00028 /* Private includes ----------------------------------------------------------*/
00029 /* USER CODE BEGIN Includes */
00030
00031 /* USER CODE END Includes */
00032
00033 /* Exported types ------------------------------------------------------------*/
00034 /* USER CODE BEGIN ET */
00035
00036 /* USER CODE END ET */
00037
00038 /* Exported constants --------------------------------------------------------*/
00039 /* USER CODE BEGIN EC */
00040
00041 /* USER CODE END EC */
00042
00043 /* Exported macro ------------------------------------------------------------*/
00044 /* USER CODE BEGIN EM */
00045
00046 /* USER CODE END EM */
00047
00048 /* Exported functions prototypes ---------------------------------------------*/
00049 void NMI_Handler(void);
00050 void HardFault_Handler(void);
00051 void MemManage_Handler(void);
00052 void BusFault_Handler(void);
```

```
00053 void UsageFault_Handler(void);
00054 void DebugMon_Handler(void);
00055 void ADC_IRQHandler(void);
00056 void TIM1_UP_TIM10_IRQHandler(void);
00057 void TIM1_TRG_COM_TIM11_IRQHandler(void);
00058 void USART1_IRQHandler(void);
00059 void DMA2_Stream2_IRQHandler(void);
00060 void DMA2_Stream7_IRQHandler(void);
00061 /* USER CODE BEGIN EFP */
00062
00063 /* USER CODE END EFP */
00064
00065 #ifdef __cplusplus
00066 }
00067 #endif
00068
00069 #endif /* __STM32F4xx_IT_H */
```

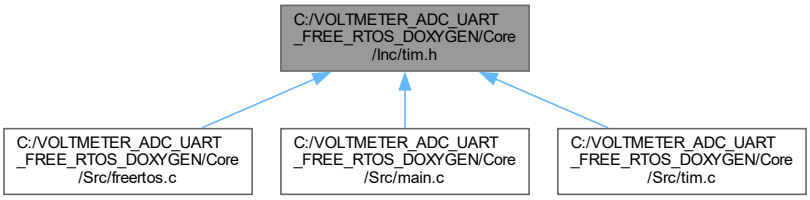
## 4.15 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Inc/tim.h File Reference

This file contains all the function prototypes for the tim.c file.

`#include "main.h"`
Include dependency graph for tim.h:




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



## Functions

- void MX_TIM1_Init (void)
- void MX_TIM11_Init (void)

## Variables

- TIM_HandleTypeDef htim1
- TIM_HandleTypeDef htim11

### 4.15.1 Detailed Description

This file contains all the function prototypes for the tim.c file.

**Attention**

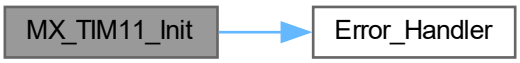


### 4.15.2 Function Documentation

#### 4.15.2.1 MX_TIM11_Init()

```
void MX_TIM11_Init (
            void  )
```

Here is the call graph for this function:

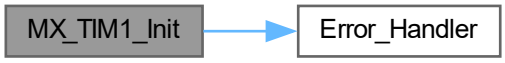



Here is the caller graph for this function:

#### 4.15.2.2 MX_TIM1_Init()

```
void MX_TIM1_Init (
            void  )
```

Here is the call graph for this function:




Here is the caller graph for this function:


### 4.15.3 Variable Documentation

#### 4.15.3.1 htim1

```
TIM_HandleTypeDef htim1  [extern]
```

#### 4.15.3.2 htim11

```
TIM_HandleTypeDef htim11  [extern]
```

## 4.16 tim.h

Go to the documentation of this file.

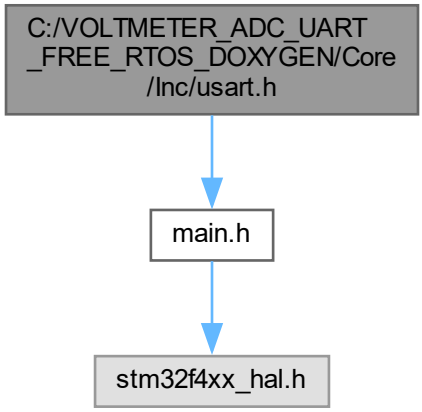
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __TIM_H__
00022 #define __TIM_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern TIM_HandleTypeDef htim1;
00036
00037 extern TIM_HandleTypeDef htim11;
00038
00039 /* USER CODE BEGIN Private defines */
00040
00041 /* USER CODE END Private defines */
00042
00043 void MX_TIM1_Init(void);
00044 void MX_TIM11_Init(void);
00045
00046 /* USER CODE BEGIN Prototypes */
00047
00048 /* USER CODE END Prototypes */
00049
00050 #ifdef __cplusplus
00051 }
00052 #endif
00053
00054 #endif /* __TIM_H__ */
00055
```

## 4.17 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Inc/usart.h File Reference

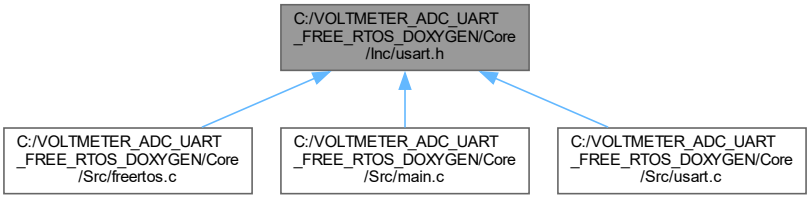
This file contains all the function prototypes for the usart.c file.

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

Include dependency graph for usart.h:




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




## Functions

- void MX_USART1_UART_Init (void)

## Variables

- UART_HandleTypeDef huart1

### 4.17.1 Detailed Description

This file contains all the function prototypes for the usart.c file.

**Attention**

## 4.17.2 Function Documentation

### 4.17.2.1 MX_USART1_UART_Init()

```
void MX_USART1_UART_Init (
            void  )
```

Here is the call graph for this function:


Here is the caller graph for this function:


## 4.17.3 Variable Documentation

### 4.17.3.1 huart1

```
UART_HandleTypeDef huart1  [extern]
```

## 4.18 usart.h

Go to the documentation of this file.

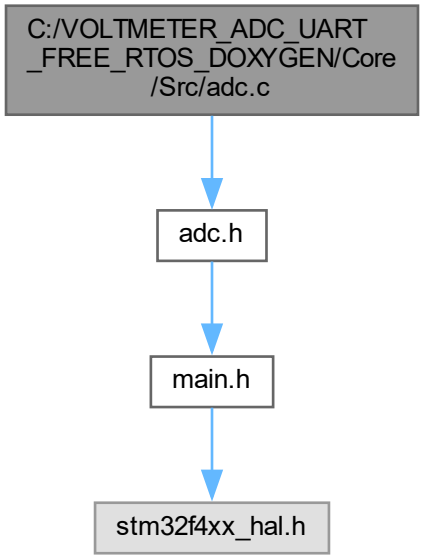
```
00001 /* USER CODE BEGIN Header */
00019 /* USER CODE END Header */
00020 /* Define to prevent recursive inclusion -------------------------------------*/
00021 #ifndef __USART_H__
00022 #define __USART_H__
00023
00024 #ifdef __cplusplus
00025 extern "C" {
00026 #endif
00027
00028 /* Includes ------------------------------------------------------------------*/
00029 #include "main.h"
00030
00031 /* USER CODE BEGIN Includes */
00032
00033 /* USER CODE END Includes */
00034
00035 extern UART_HandleTypeDef huart1;
00036
00037 /* USER CODE BEGIN Private defines */
00038
00039 /* USER CODE END Private defines */
00040
00041 void MX_USART1_UART_Init(void);
00042
00043 /* USER CODE BEGIN Prototypes */
00044
00045 /* USER CODE END Prototypes */
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* __USART_H__ */
00052
```

## 4.19 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Src/adc.c File Reference

This file provides code for the configuration of the ADC instances.

```
#include "adc.h"
```

Include dependency graph for adc.c:




## Functions

- void MX_ADC1_Init (void)
- void HAL_ADC_MspInit (ADC_HandleTypeDef *adcHandle)
- void HAL_ADC_MspDeInit (ADC_HandleTypeDef *adcHandle)

## Variables

- ADC_HandleTypeDef hadc1

### 4.19.1 Detailed Description

This file provides code for the configuration of the ADC instances.

**Attention**

### 4.19.2 Function Documentation

#### 4.19.2.1 HAL_ADC_MspDeInit()

```
void HAL_ADC_MspDeInit (
            ADC_HandleTypeDef * adcHandle )
```

ADC1 GPIO Configuration PA0-WKUP ----—> ADC1_IN0

#### 4.19.2.2 HAL_ADC_MspInit()

```
void HAL_ADC_MspInit (
            ADC_HandleTypeDef * adcHandle )
```

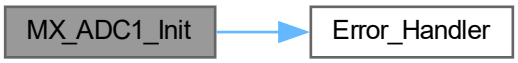
ADC1 GPIO Configuration PA0-WKUP ----—> ADC1_IN0

#### 4.19.2.3 MX_ADC1_Init()

```
void MX_ADC1_Init (
            void  )
```

Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of conversion)

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time.Here is the call graph for this function:




Here is the caller graph for this function:

### 4.19.3 Variable Documentation

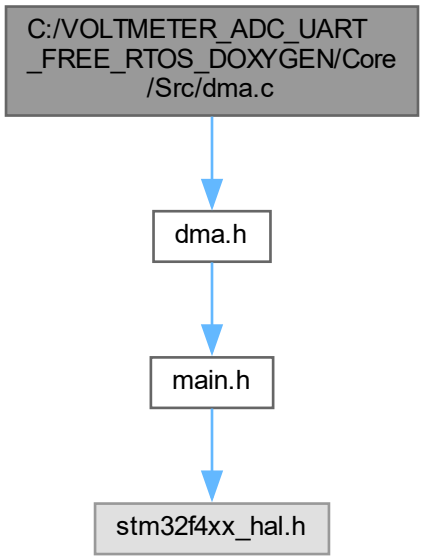
#### 4.19.3.1 hadc1

```
ADC_HandleTypeDef hadc1
```

## 4.20 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Src/dma.c File Reference

This file provides code for the configuration of all the requested memory to memory DMA transfers.

```
#include "dma.h"
```
Include dependency graph for dma.c:




## Functions

- void MX_DMA_Init (void)

  *Enable DMA controller clock.*

### 4.20.1 Detailed Description

This file provides code for the configuration of all the requested memory to memory DMA transfers.

**Attention**



### 4.20.2 Function Documentation

#### 4.20.2.1 MX_DMA_Init()

```
void MX_DMA_Init (
             void  )
```

Enable DMA controller clock.

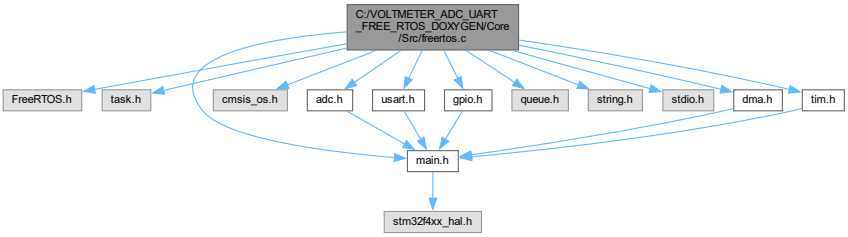
Here is the caller graph for this function:


## 4.21 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Src/freertos.c File Reference

```
#include "FreeRTOS.h"
#include "task.h"
#include "main.h"
#include "cmsis_os.h"
#include "adc.h"
#include "usart.h"
#include "gpio.h"
#include "queue.h"
#include <string.h>
#include <stdio.h>
#include "dma.h"
```

```
#include "tim.h"
```

Include dependency graph for freertos.c:




## Macros

- #define RxBuf_SIZE 512

  *File Name : freertos.c Description : Code for freertos applications.*
- #define MainBuf_SIZE 2048

## Functions

- void IDLE_DEBUG_TASK_INIT (void const ∗argument)

  *Function implementing the IDLE_DEBUG_TASK thread.*
- void ADC_TAKE_TASK_INIT (void const ∗argument)

  *Function implementing the ADC_TAKE_TASK thread.*
- void UART_TX_TASK_INIT (void const ∗argument)

  *Function implementing the UART_TX_TASK thread.*
- void UART_RX_TASK_INIT (void const ∗argument)

  *Function implementing the UART_RX_TASK thread.*
- void ADC_CONFIG_TASK_INIT (void const ∗argument)

  *Function implementing the ADC_CONFIG_TASK thread.*
- void INT_TIMER_TASK_INIT (void const ∗argument)

  *Function implementing the INT_TIMER_TASK thread.*
- void INT_UART_TASK_INIT (void const ∗argument)

  *Function implementing the INT_UART_TASK thread.*
- void TIMER_CALLBACK (void const ∗argument)
- void MX_FREERTOS_Init (void)

  *FreeRTOS initialization.*
- void vApplicationGetIdleTaskMemory (StaticTask_t ∗∗ppxIdleTaskTCBBuffer, StackType_t ∗∗ppxIdleTask↩ StackBuffer, uint32_t ∗pulIdleTaskStackSize)
- void vApplicationGetTimerTaskMemory (StaticTask_t ∗∗ppxTimerTaskTCBBuffer, StackType_t ∗∗ppxTimer↩ TaskStackBuffer, uint32_t ∗pulTimerTaskStackSize)
- void vApplicationIdleHook (void)
- void vApplicationTickHook (void)
- void HAL_TIM_PeriodicElapsedCallback (TIM_HandleTypeDef ∗htim)
- void HAL_UARTEx_RxEventCallback (UART_HandleTypeDef ∗huart, uint16_t Size)

### Variables

- volatile uint8_t RxBuf [RxBuf_SIZE]
- volatile uint8_t MainBuf [MainBuf_SIZE]
- volatile uint16_t oldPos = 0
- volatile uint16_t newPos = 0
- osThreadId IDLE_DEBUG_TASKHandle
- osThreadId ADC_TAKE_TASKHandle
- osThreadId UART_TX_TASKHandle
- osThreadId UART_RX_TASKHandle
- osThreadId ADC_CONFIG_TASKHandle
- osThreadId INT_TIMER_TASKHandle
- osThreadId INT_UART_TASKHandle
- osMessageQId ADC_QUEUE_UART_TXHandle
- osMessageQId UART_RX_QUEUE_ADC_TAKHandle
- osMessageQId UART_RX_QUEUE_UART_TXHandle
- osMessageQId UART_RX_QUEUE_ADC_CONFIGHandle
- osMessageQId INT_UART_QUEUE_UART_RXHandle
- osTimerId RTOS_TIMER_1Handle

## 4.21.1 Macro Definition Documentation

### 4.21.1.1 MainBuf_SIZE

```
#define MainBuf_SIZE 2048
```

### 4.21.1.2 RxBuf_SIZE

```
#define RxBuf_SIZE 512
```

File Name : freertos.c Description : Code for freertos applications.

**Attention**



## 4.21.2 Function Documentation

### 4.21.2.1 ADC_CONFIG_TASK_INIT()

```
void ADC_CONFIG_TASK_INIT (
            void const * argument )
```

Function implementing the ADC_CONFIG_TASK thread.

**Parameters**

| | |
|---|---|
| *argument* | Not used |

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:


#### 4.21.2.2 ADC_TAKE_TASK_INIT()

```
void ADC_TAKE_TASK_INIT (
            void const * argument )
```

Function implementing the ADC_TAKE_TASK thread.

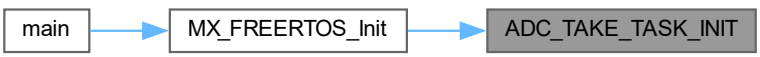
**Parameters**

| | |
|---|---|
| *argument* | Not used |

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:



#### 4.21.2.3 HAL_TIM_PeriodicElapsedCallback()

```
void HAL_TIM_PeriodicElapsedCallback (
            TIM_HandleTypeDef * htim )
```

#### 4.21.2.4 HAL_UARTEx_RxEventCallback()

```
void HAL_UARTEx_RxEventCallback (
            UART_HandleTypeDef * huart,
            uint16_t Size )
```

#### 4.21.2.5 IDLE_DEBUG_TASK_INIT()

```
void IDLE_DEBUG_TASK_INIT (
            void const * argument )
```

Function implementing the IDLE_DEBUG_TASK thread.

**Parameters**

| | |
|---|---|
| *argument* | Not used |

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:


#### 4.21.2.6 INT_TIMER_TASK_INIT()

```
void INT_TIMER_TASK_INIT (
            void const * argument )
```

Function implementing the INT_TIMER_TASK thread.

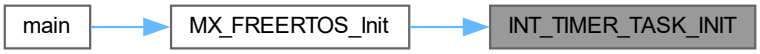
**Parameters**

| | |
|---|---|
| *argument* | Not used |

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:




#### 4.21.2.7 INT_UART_TASK_INIT()

```
void INT_UART_TASK_INIT (
            void const * argument )
```

Function implementing the INT_UART_TASK thread.

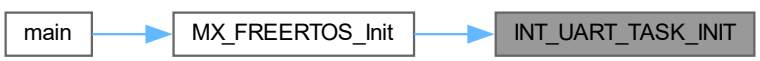
**Parameters**

| | |
|---|---|
| *argument* | Not used |

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:



#### 4.21.2.8 MX_FREERTOS_Init()

```
void MX_FREERTOS_Init (
            void  )
```

FreeRTOS initialization.

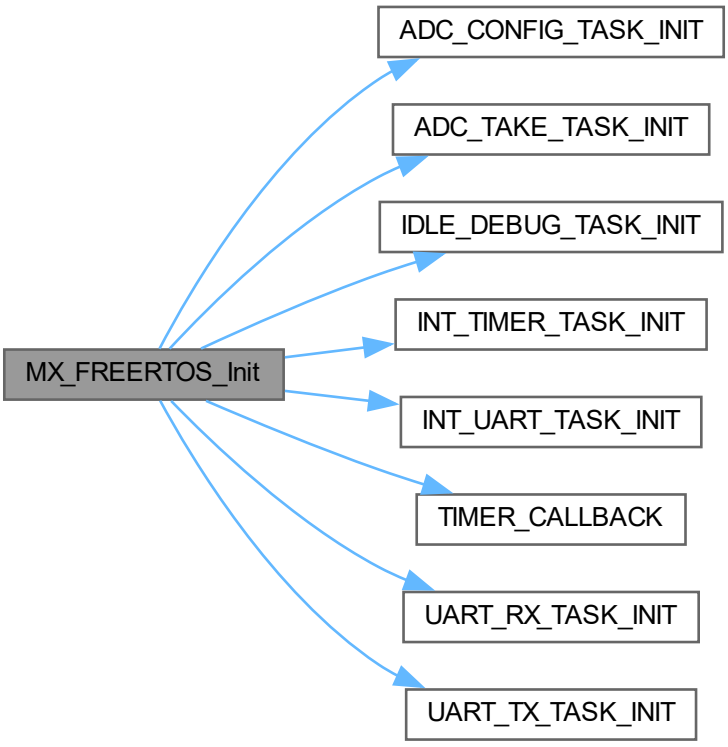
**Parameters**

| | |
|---|---|
| *None* | |

**Return values**

| | |
|---|---|
| *None* | |

Here is the call graph for this function:

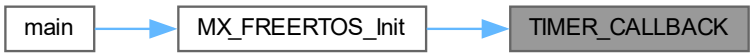


Here is the caller graph for this function:


#### 4.21.2.9 TIMER_CALLBACK()

```
void TIMER_CALLBACK (
             void const * argument )
```

Here is the caller graph for this function:




#### 4.21.2.10 UART_RX_TASK_INIT()

```
void UART_RX_TASK_INIT (
             void const * argument )
```

Function implementing the UART_RX_TASK thread.

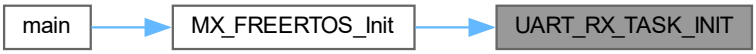
**Parameters**

| | |
|---|---|
| *argument* | Not used |

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:




#### 4.21.2.11 UART_TX_TASK_INIT()

```
void UART_TX_TASK_INIT (
            void const * argument )
```

Function implementing the UART_TX_TASK thread.

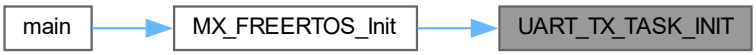
**Parameters**

| | |
|---|---|
| *argument* | Not used |

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:




#### 4.21.2.12 vApplicationGetIdleTaskMemory()

```
void vApplicationGetIdleTaskMemory (
            StaticTask_t ** ppxIdleTaskTCBBuffer,
            StackType_t ** ppxIdleTaskStackBuffer,
            uint32_t * pulIdleTaskStackSize )
```

#### 4.21.2.13 vApplicationGetTimerTaskMemory()

```
void vApplicationGetTimerTaskMemory (
             StaticTask_t ** ppxTimerTaskTCBBuffer,
             StackType_t ** ppxTimerTaskStackBuffer,
             uint32_t * pulTimerTaskStackSize )
```

#### 4.21.2.14 vApplicationIdleHook()

```
__weak void vApplicationIdleHook (
             void  )
```

#### 4.21.2.15 vApplicationTickHook()

```
__weak void vApplicationTickHook (
             void  )
```

### 4.21.3 Variable Documentation

#### 4.21.3.1 ADC_CONFIG_TASKHandle

```
osThreadId ADC_CONFIG_TASKHandle
```

#### 4.21.3.2 ADC_QUEUE_UART_TXHandle

```
osMessageQId ADC_QUEUE_UART_TXHandle
```

#### 4.21.3.3 ADC_TAKE_TASKHandle

```
osThreadId ADC_TAKE_TASKHandle
```

#### 4.21.3.4 IDLE_DEBUG_TASKHandle

```
osThreadId IDLE_DEBUG_TASKHandle
```

#### 4.21.3.5 INT_TIMER_TASKHandle

```
osThreadId INT_TIMER_TASKHandle
```

#### 4.21.3.6 INT_UART_QUEUE_UART_RXHandle

```
osMessageQId INT_UART_QUEUE_UART_RXHandle
```

#### 4.21.3.7 INT_UART_TASKHandle

```
osThreadId INT_UART_TASKHandle
```

#### 4.21.3.8 MainBuf

```
volatile uint8_t MainBuf[MainBuf_SIZE]
```

#### 4.21.3.9 newPos

```
volatile uint16_t newPos = 0
```

#### 4.21.3.10 oldPos

```
volatile uint16_t oldPos = 0
```

#### 4.21.3.11 RTOS_TIMER_1Handle

```
osTimerId RTOS_TIMER_1Handle
```

#### 4.21.3.12 RxBuf

```
volatile uint8_t RxBuf[RxBuf_SIZE]
```

#### 4.21.3.13 UART_RX_QUEUE_ADC_CONFIGHandle

```
osMessageQId UART_RX_QUEUE_ADC_CONFIGHandle
```

#### 4.21.3.14 UART_RX_QUEUE_ADC_TAKHandle

```
osMessageQId UART_RX_QUEUE_ADC_TAKHandle
```

#### 4.21.3.15 UART_RX_QUEUE_UART_TXHandle

```
osMessageQId UART_RX_QUEUE_UART_TXHandle
```

#### 4.21.3.16 UART_RX_TASKHandle

```
osThreadId UART_RX_TASKHandle
```

#### 4.21.3.17 UART_TX_TASKHandle

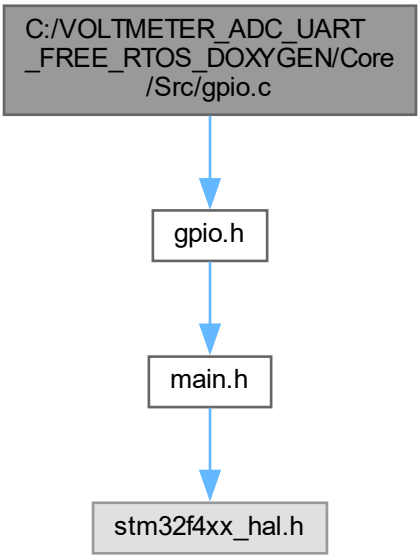
```
osThreadId UART_TX_TASKHandle
```

## 4.22 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Src/gpio.c File Reference

This file provides code for the configuration of all used GPIO pins.

```
#include "gpio.h"
```

Include dependency graph for gpio.c:




### Functions

- void MX_GPIO_Init (void)

  *Configure pins.*

### 4.22.1 Detailed Description

This file provides code for the configuration of all used GPIO pins.

**Attention**

### 4.22.2 Function Documentation

#### 4.22.2.1 MX_GPIO_Init()

```
void MX_GPIO_Init (
            void  )
```

Configure pins.

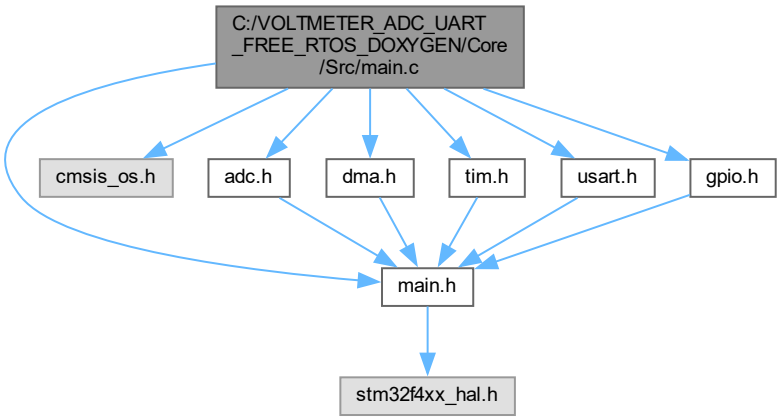
Here is the caller graph for this function:


## 4.23 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Src/main.c File Reference

: Main program body

```
#include "main.h"
#include "cmsis_os.h"
#include "adc.h"
#include "dma.h"
#include "tim.h"
#include "usart.h"
#include "gpio.h"
```

Include dependency graph for main.c:



## Functions

- void SystemClock_Config (void)

  *System Clock Configuration.*
- void MX_FREERTOS_Init (void)

  *FreeRTOS initialization.*
- int main (void)

  *The application entry point.*
- void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef *htim)

  *Period elapsed callback in non blocking mode.*
- void Error_Handler (void)

  *This function is executed in case of error occurrence.*

### 4.23.1 Detailed Description

: Main program body

**Attention**



### 4.23.2 Function Documentation

#### 4.23.2.1 Error_Handler()

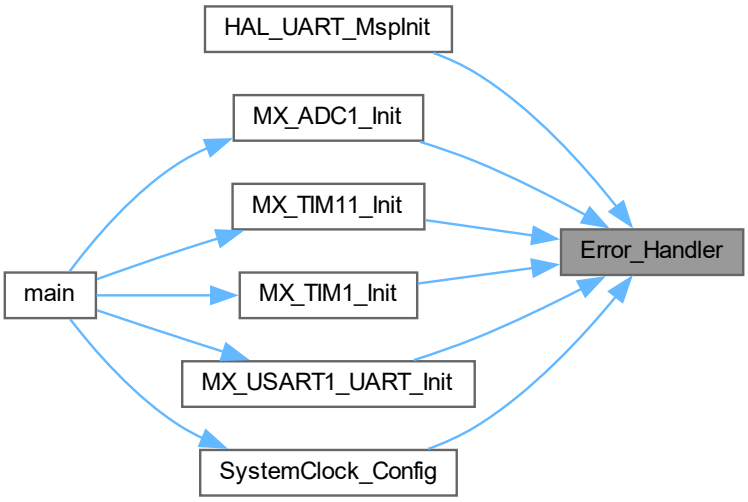
```
void Error_Handler (
            void  )
```

This function is executed in case of error occurrence.

**Return values**

| | |
|---|---|
| *None* | |

Here is the caller graph for this function:




#### 4.23.2.2 HAL_TIM_PeriodElapsedCallback()

```
void HAL_TIM_PeriodElapsedCallback (
            TIM_HandleTypeDef * htim )
```

Period elapsed callback in non blocking mode.

**Note**

This function is called when TIM10 interrupt took place, inside HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to increment a global variable "uwTick" used as application time base.

**Parameters**

| | |
|---|---|
| *htim* | : TIM handle |

**Return values**

| | |
|---|---|
| *None* | |

#### 4.23.2.3 main()

```
int main (
```

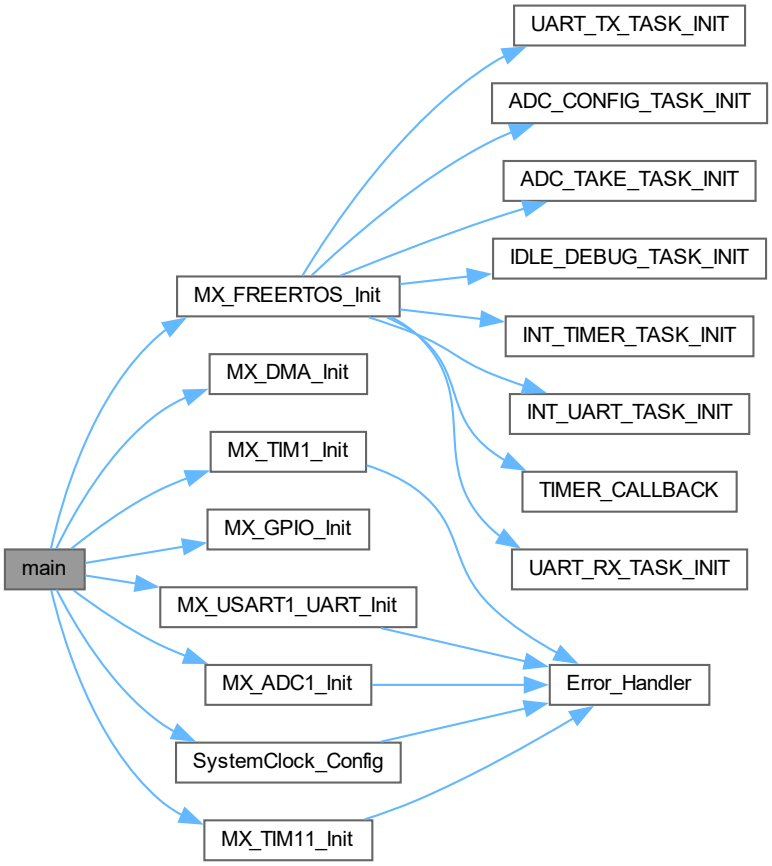
```
            void  )
```

The application entry point.

**Return values**

| | |
|---|---|
| *int* | |

Here is the call graph for this function:




#### 4.23.2.4 MX_FREERTOS_Init()

```
void MX_FREERTOS_Init (
            void  )
```

FreeRTOS initialization.

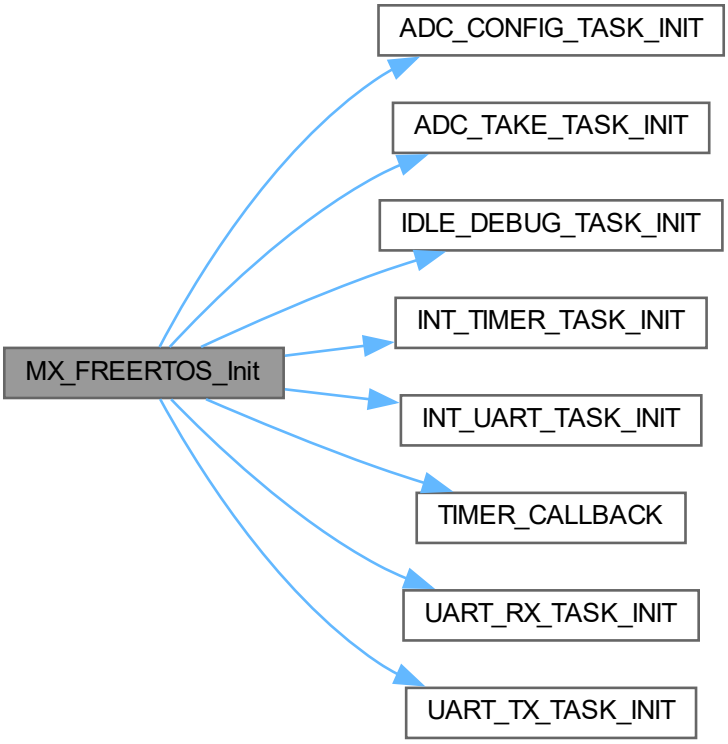
**Parameters**

| | |
|---|---|
| *None* | |

**Return values**

| | |
|---|---|
| *None* | |

Here is the call graph for this function:




Here is the caller graph for this function:

#### 4.23.2.5 SystemClock_Config()

```
void SystemClock_Config (
            void  )
```

System Clock Configuration.

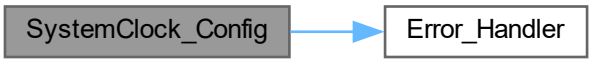
**Return values**

| | |
|---|---|
| *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 clocksHere is the call graph for this function:



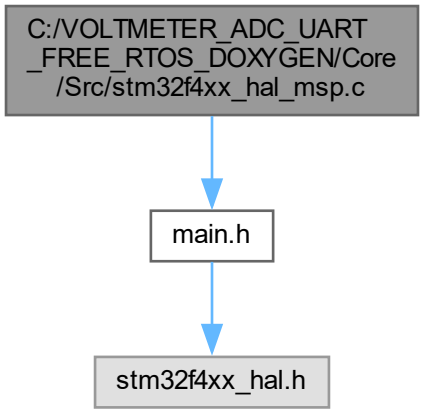

Here is the caller graph for this function:


## 4.24 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/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)

  *Initializes the Global MSP.*

### 4.24.1 Detailed Description

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

**Attention**



### 4.24.2 Function Documentation

#### 4.24.2.1 HAL_MspInit()

```
void HAL_MspInit (
             void  )
```

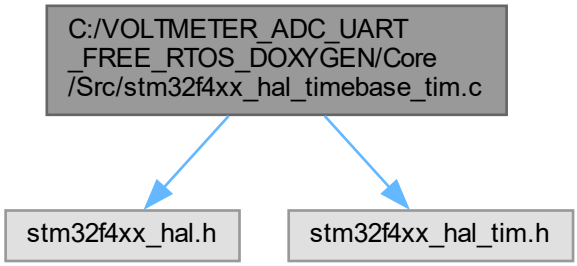
Initializes the Global MSP.

## 4.25 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/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 TIM10 as a time base source.*
- void HAL_SuspendTick (void)
  *Suspend Tick increment.*
- void HAL_ResumeTick (void)
  *Resume Tick increment.*

### Variables

- TIM_HandleTypeDef htim10

### 4.25.1 Detailed Description

HAL time base based on the hardware TIM.

**Attention**

## 4.25.2 Function Documentation

### 4.25.2.1 HAL_InitTick()

```
HAL_StatusTypeDef HAL_InitTick (
            uint32_t TickPriority )
```

This function configures the TIM10 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. |

**Return values**

| | |
|---|---|
| *HAL* | status |

### 4.25.2.2 HAL_ResumeTick()

```
void HAL_ResumeTick (
            void  )
```

Resume Tick increment.

**Note**

Enable the tick increment by Enabling TIM10 update interrupt.

**Parameters**

| | |
|---|---|
| *None* | |

**Return values**

| | |
|---|---|
| *None* | |

#### 4.25.2.3 HAL_SuspendTick()

```
void HAL_SuspendTick (
             void  )
```

Suspend Tick increment.

**Note**

Disable the tick increment by disabling TIM10 update interrupt.

**Parameters**

| | |
|---|---|
| *None* | |

**Return values**

| | |
|---|---|
| *None* | |

### 4.25.3 Variable Documentation

#### 4.25.3.1 htim10

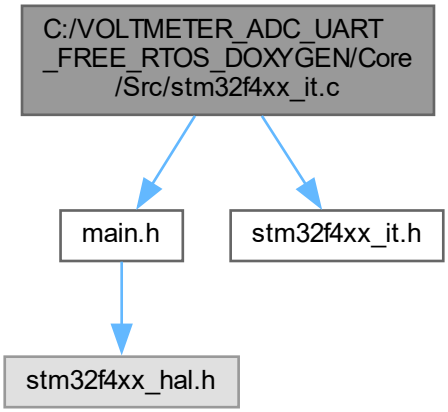
```
TIM_HandleTypeDef htim10
```

## 4.26 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/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 ADC_IRQHandler (void)

  *This function handles ADC1 global interrupt.*
- void TIM1_UP_TIM10_IRQHandler (void)

  *This function handles TIM1 update interrupt and TIM10 global interrupt.*
- void TIM1_TRG_COM_TIM11_IRQHandler (void)

  *This function handles TIM1 trigger and commutation interrupts and TIM11 global interrupt.*
- void USART1_IRQHandler (void)

  *This function handles USART1 global interrupt.*
- void DMA2_Stream2_IRQHandler (void)

  *This function handles DMA2 stream2 global interrupt.*
- void DMA2_Stream7_IRQHandler (void)

  *This function handles DMA2 stream7 global interrupt.*

## Variables

- ADC_HandleTypeDef hadc1
- TIM_HandleTypeDef htim1
- TIM_HandleTypeDef htim11
- DMA_HandleTypeDef hdma_usart1_rx
- DMA_HandleTypeDef hdma_usart1_tx
- UART_HandleTypeDef huart1
- TIM_HandleTypeDef htim10

### 4.26.1 Detailed Description

Interrupt Service Routines.

**Attention**



### 4.26.2 Function Documentation

#### 4.26.2.1 ADC_IRQHandler()

```
void ADC_IRQHandler (
            void  )
```

This function handles ADC1 global interrupt.

#### 4.26.2.2 BusFault_Handler()

```
void BusFault_Handler (
            void  )
```

This function handles Pre-fetch fault, memory access fault.

#### 4.26.2.3 DebugMon_Handler()

```
void DebugMon_Handler (
             void  )
```

This function handles Debug monitor.

#### 4.26.2.4 DMA2_Stream2_IRQHandler()

```
void DMA2_Stream2_IRQHandler (
             void  )
```

This function handles DMA2 stream2 global interrupt.

#### 4.26.2.5 DMA2_Stream7_IRQHandler()

```
void DMA2_Stream7_IRQHandler (
             void  )
```

This function handles DMA2 stream7 global interrupt.

#### 4.26.2.6 HardFault_Handler()

```
void HardFault_Handler (
             void  )
```

This function handles Hard fault interrupt.

#### 4.26.2.7 MemManage_Handler()

```
void MemManage_Handler (
             void  )
```

This function handles Memory management fault.

#### 4.26.2.8 NMI_Handler()

```
void NMI_Handler (
             void  )
```

This function handles Non maskable interrupt.

#### 4.26.2.9 TIM1_TRG_COM_TIM11_IRQHandler()

```
void TIM1_TRG_COM_TIM11_IRQHandler (
             void  )
```

This function handles TIM1 trigger and commutation interrupts and TIM11 global interrupt.

#### 4.26.2.10 TIM1_UP_TIM10_IRQHandler()

```
void TIM1_UP_TIM10_IRQHandler (
             void  )
```

This function handles TIM1 update interrupt and TIM10 global interrupt.

#### 4.26.2.11 UsageFault_Handler()

```
void UsageFault_Handler (
             void  )
```

This function handles Undefined instruction or illegal state.

#### 4.26.2.12 USART1_IRQHandler()

```
void USART1_IRQHandler (
             void  )
```

This function handles USART1 global interrupt.

### 4.26.3 Variable Documentation

#### 4.26.3.1 hadc1

```
ADC_HandleTypeDef hadc1  [extern]
```

#### 4.26.3.2 hdma_usart1_rx

```
DMA_HandleTypeDef hdma_usart1_rx  [extern]
```

#### 4.26.3.3 hdma_usart1_tx

```
DMA_HandleTypeDef hdma_usart1_tx  [extern]
```

#### 4.26.3.4 htim1

```
TIM_HandleTypeDef htim1  [extern]
```

#### 4.26.3.5 htim10

```
TIM_HandleTypeDef htim10  [extern]
```

#### 4.26.3.6 htim11

```
TIM_HandleTypeDef htim11  [extern]
```

#### 4.26.3.7 huart1

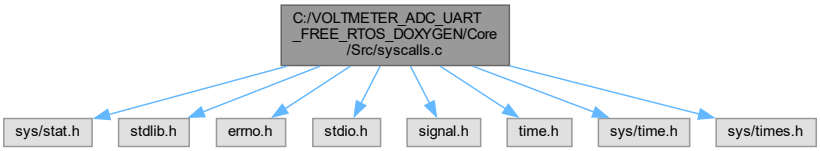
```
UART_HandleTypeDef huart1  [extern]
```

## 4.27 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/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

### 4.27.1 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**

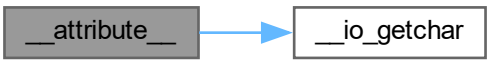


### 4.27.2 Function Documentation

#### 4.27.2.1 __attribute__()

```
__attribute__ (
             (weak)  )
```

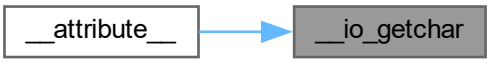
Here is the call graph for this function:




#### 4.27.2.2 __io_getchar()

```
int __io_getchar (
             void  )
```

Here is the caller graph for this function:




#### 4.27.2.3 __io_putchar()

```
int __io_putchar (
             int ch )
```

#### 4.27.2.4 _close()

```
int _close (
             int file )
```

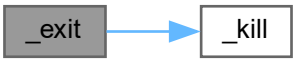
#### 4.27.2.5 _execve()

```
int _execve (
             char * name,
             char ** argv,
             char ** env )
```

#### 4.27.2.6 _exit()

```
void _exit (
             int status )
```

Here is the call graph for this function:




#### 4.27.2.7 _fork()

```
int _fork (
             void  )
```

#### 4.27.2.8 _fstat()

```
int _fstat (
             int file,
             struct stat * st )
```

#### 4.27.2.9 _getpid()

```
int _getpid (
             void  )
```

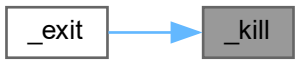
#### 4.27.2.10 _isatty()

```
int _isatty (
              int file )
```

#### 4.27.2.11 _kill()

```
int _kill (
              int pid,
              int sig )
```

Here is the caller graph for this function:




#### 4.27.2.12 _link()

```
int _link (
              char * old,
              char * new )
```

#### 4.27.2.13 _lseek()

```
int _lseek (
              int file,
              int ptr,
              int dir )
```

#### 4.27.2.14 _open()

```
int _open (
              char * path,
              int flags,
               ...  )
```

#### 4.27.2.15 _stat()

```
int _stat (
              char * file,
              struct stat * st )
```

#### 4.27.2.16 _times()

```
int _times (
              struct tms * buf )
```

#### 4.27.2.17 _unlink()

```
int _unlink (
              char * name )
```

#### 4.27.2.18 _wait()

```
int _wait (
              int * status )
```

#### 4.27.2.19 initialise_monitor_handles()

```
void initialise_monitor_handles ( )
```

### 4.27.3 Variable Documentation

#### 4.27.3.1 environ

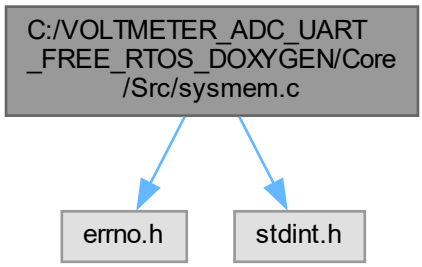
```
char** environ = __env
```

# 4.28 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/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*

### 4.28.1 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**

### 4.28.2 Function Documentation

#### 4.28.2.1 _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**

| | |
|---|---|
| *incr* | Memory size |

**Returns**

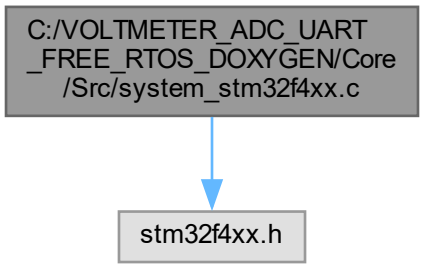
Pointer to allocated memory

## 4.29 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/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.*

## 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}

### 4.29.1 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.

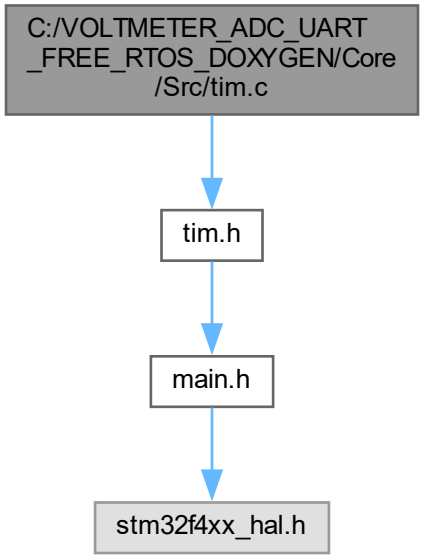
**Attention**

## 4.30 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Src/tim.c File Reference

This file provides code for the configuration of the TIM instances.

```
#include "tim.h"
```

Include dependency graph for tim.c:




### Functions

- void MX_TIM1_Init (void)
- void MX_TIM11_Init (void)
- void HAL_TIM_Base_MspInit (TIM_HandleTypeDef *tim_baseHandle)
- void HAL_TIM_Base_MspDeInit (TIM_HandleTypeDef *tim_baseHandle)

### Variables

- TIM_HandleTypeDef htim1
- TIM_HandleTypeDef htim11

### 4.30.1 Detailed Description

This file provides code for the configuration of the TIM instances.

**Attention**

## 4.30.2 Function Documentation

### 4.30.2.1 HAL_TIM_Base_MspDeInit()

```
void HAL_TIM_Base_MspDeInit (
            TIM_HandleTypeDef * tim_baseHandle )
```

Uncomment the line below to disable the "TIM1_TRG_COM_TIM11_IRQn" interrupt Be aware, disabling shared interrupt may affect other IPs

Uncomment the line below to disable the "TIM1_TRG_COM_TIM11_IRQn" interrupt Be aware, disabling shared interrupt may affect other IPs

### 4.30.2.2 HAL_TIM_Base_MspInit()

```
void HAL_TIM_Base_MspInit (
            TIM_HandleTypeDef * tim_baseHandle )
```

### 4.30.2.3 MX_TIM11_Init()

```
void MX_TIM11_Init (
            void  )
```

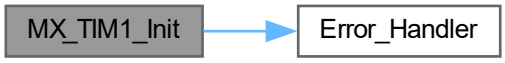
Here is the call graph for this function:


Here is the caller graph for this function:

#### 4.30.2.4 MX_TIM1_Init()

```
void MX_TIM1_Init (
             void  )
```

Here is the call graph for this function:




Here is the caller graph for this function:


### 4.30.3 Variable Documentation

#### 4.30.3.1 htim1

```
TIM_HandleTypeDef htim1
```

#### 4.30.3.2 htim11

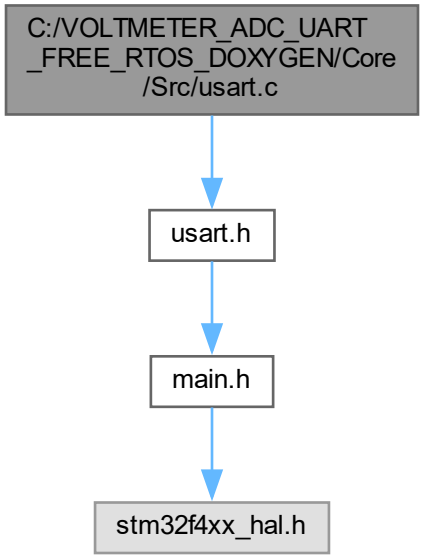
```
TIM_HandleTypeDef htim11
```

# 4.31 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/↩ Src/usart.c File Reference

This file provides code for the configuration of the USART instances.

```
#include "usart.h"
```

Include dependency graph for usart.c:




## Functions

- void MX_USART1_UART_Init (void)
- void HAL_UART_MspInit (UART_HandleTypeDef ∗uartHandle)
- void HAL_UART_MspDeInit (UART_HandleTypeDef ∗uartHandle)

## Variables

- UART_HandleTypeDef huart1
- DMA_HandleTypeDef hdma_usart1_rx
- DMA_HandleTypeDef hdma_usart1_tx

### 4.31.1 Detailed Description

This file provides code for the configuration of the USART instances.

**Attention**

### 4.31.2 Function Documentation

#### 4.31.2.1 HAL_UART_MspDeInit()

```
void HAL_UART_MspDeInit (
            UART_HandleTypeDef * uartHandle )
```

USART1 GPIO Configuration PA9 ------> USART1_TX PA10 ------> USART1_RX

#### 4.31.2.2 HAL_UART_MspInit()

```
void HAL_UART_MspInit (
            UART_HandleTypeDef * uartHandle )
```

USART1 GPIO Configuration PA9 ------> USART1_TX PA10 ------> USART1_RXHere is the call graph for this function:


#### 4.31.2.3 MX_USART1_UART_Init()

```
void MX_USART1_UART_Init (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:


### 4.31.3 Variable Documentation

#### 4.31.3.1 hdma_usart1_rx

```
DMA_HandleTypeDef hdma_usart1_rx
```

#### 4.31.3.2 hdma_usart1_tx

```
DMA_HandleTypeDef hdma_usart1_tx
```

#### 4.31.3.3 huart1

```
UART_HandleTypeDef huart1
```

# Index