

## STM32 Project

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Module Documentation</b>	<b>5</b>
3.1	CMSIS . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.2	Stm32f3xx_system . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.3	STM32F3xx_System_Private_Includes . . . . .	7
3.4	STM32F3xx_System_Private_TypesDefinitions . . . . .	8
3.5	STM32F3xx_System_Private_Defines . . . . .	9
3.5.1	Detailed Description . . . . .	9
3.5.2	Macro Definition Documentation . . . . .	9
3.5.2.1	HSE_VALUE . . . . .	9
3.5.2.2	HSI_VALUE . . . . .	9
3.5.2.3	VECT_TAB_OFFSET . . . . .	9
3.6	STM32F3xx_System_Private_Macros . . . . .	10
3.7	STM32F3xx_System_Private_Variables . . . . .	11
3.7.1	Detailed Description . . . . .	11
3.7.2	Variable Documentation . . . . .	11
3.7.2.1	AHBPrescTable . . . . .	11
3.7.2.2	APBPrescTable . . . . .	11
3.7.2.3	SystemCoreClock . . . . .	11
3.8	STM32F3xx_System_Private_FunctionPrototypes . . . . .	12
3.9	STM32F3xx_System_Private_Functions . . . . .	13
3.9.1	Detailed Description . . . . .	13
3.9.2	Function Documentation . . . . .	13
3.9.2.1	SystemCoreClockUpdate() . . . . .	13
3.9.2.2	SystemInit() . . . . .	14

<b>4 File Documentation</b>	<b>15</b>
4.1 Inc/main.h File Reference	15
4.1.1 Detailed Description	16
4.1.2 Macro Definition Documentation	16
4.1.2.1 B1_EXTI_IRQn	17
4.1.2.2 B1_GPIO_Port	17
4.1.2.3 B1_Pin	17
4.1.2.4 LD2_GPIO_Port	17
4.1.2.5 LD2_Pin	17
4.1.2.6 RCC_OSC32_OUT_GPIO_Port	17
4.1.2.7 RCC_OSC32_OUT_Pin	17
4.1.2.8 SWO_GPIO_Port	17
4.1.2.9 SWO_Pin	18
4.1.2.10 TCK_GPIO_Port	18
4.1.2.11 TCK_Pin	18
4.1.2.12 TMS_GPIO_Port	18
4.1.2.13 TMS_Pin	18
4.1.2.14 USART_RX_GPIO_Port	18
4.1.2.15 USART_RX_Pin	18
4.1.2.16 USART_TX_GPIO_Port	18
4.1.2.17 USART_TX_Pin	19
4.1.3 Function Documentation	19
4.1.3.1 Error_Handler()	19
4.2 Inc/stm32f3xx_hal_conf.h File Reference	19
4.2.1 Detailed Description	20
4.2.2 Macro Definition Documentation	21
4.2.2.1 assert_param	21
4.2.2.2 DATA_CACHE_ENABLE	21
4.2.2.3 EXTERNAL_CLOCK_VALUE	22
4.2.2.4 HAL_ADC_MODULE_ENABLED	22

4.2.2.5	HAL_CORTEX_MODULE_ENABLED . . . . .	22
4.2.2.6	HAL_DMA_MODULE_ENABLED . . . . .	22
4.2.2.7	HAL_FLASH_MODULE_ENABLED . . . . .	22
4.2.2.8	HAL_GPIO_MODULE_ENABLED . . . . .	22
4.2.2.9	HAL_I2C_MODULE_ENABLED . . . . .	22
4.2.2.10	HAL_MODULE_ENABLED . . . . .	23
4.2.2.11	HAL_OPAMP_MODULE_ENABLED . . . . .	23
4.2.2.12	HAL_PWR_MODULE_ENABLED . . . . .	23
4.2.2.13	HAL_RCC_MODULE_ENABLED . . . . .	23
4.2.2.14	HAL_TIM_MODULE_ENABLED . . . . .	23
4.2.2.15	HAL_UART_MODULE_ENABLED . . . . .	23
4.2.2.16	HSE_STARTUP_TIMEOUT . . . . .	23
4.2.2.17	HSE_VALUE . . . . .	24
4.2.2.18	HSI_STARTUP_TIMEOUT . . . . .	24
4.2.2.19	HSI_VALUE . . . . .	24
4.2.2.20	INSTRUCTION_CACHE_ENABLE . . . . .	24
4.2.2.21	LSE_STARTUP_TIMEOUT . . . . .	24
4.2.2.22	LSE_VALUE . . . . .	24
4.2.2.23	LSI_VALUE . . . . .	25
4.2.2.24	PREFETCH_ENABLE . . . . .	25
4.2.2.25	TICK_INT_PRIORITY . . . . .	25
4.2.2.26	USE_RTOS . . . . .	25
4.2.2.27	USE_SPI_CRC . . . . .	25
4.2.2.28	VDD_VALUE . . . . .	25
4.3	Inc/stm32f3xx_it.h File Reference . . . . .	26
4.3.1	Detailed Description . . . . .	27
4.3.2	Function Documentation . . . . .	27
4.3.2.1	ADC1_2_IRQHandler() . . . . .	27
4.3.2.2	BusFault_Handler() . . . . .	27
4.3.2.3	DebugMon_Handler() . . . . .	27

4.3.2.4	DMA1_Channel1_IRQHandler()	28
4.3.2.5	EXTI15_10_IRQHandler()	28
4.3.2.6	HardFault_Handler()	28
4.3.2.7	I2C1_ER_IRQHandler()	28
4.3.2.8	I2C1_EV_IRQHandler()	28
4.3.2.9	MemManage_Handler()	28
4.3.2.10	NMI_Handler()	29
4.3.2.11	PendSV_Handler()	29
4.3.2.12	SVC_Handler()	29
4.3.2.13	SysTick_Handler()	29
4.3.2.14	TIM1_UP_TIM16_IRQHandler()	29
4.3.2.15	UsageFault_Handler()	29
4.3.2.16	USART2_IRQHandler()	30
4.4	Src/main.c File Reference	30
4.4.1	Detailed Description	31
4.4.2	Macro Definition Documentation	32
4.4.2.1	BUFFER_SIZE	32
4.4.2.2	DATA_SIZE	32
4.4.2.3	IN_SIZE	32
4.4.2.4	INA219_ADDRESS	32
4.4.2.5	INA219_REG_BUSVOLTAGE	32
4.4.2.6	INA219_REG_CALIBRATION	33
4.4.2.7	INA219_REG_CONFIG	33
4.4.2.8	INA219_REG_CURRENT	33
4.4.2.9	INA219_REG_POWER	33
4.4.2.10	INA219_REG_SHUNTVOLTAGE	33
4.4.3	Function Documentation	33
4.4.3.1	Error_Handler()	33
4.4.3.2	getBusVoltage_V()	34
4.4.3.3	getCurrent_mA()	34

4.4.3.4	<a href="#">getCurrentDivider_mA()</a>	34
4.4.3.5	<a href="#">getPower_mW()</a>	34
4.4.3.6	<a href="#">getPowerMultiplier_mW()</a>	35
4.4.3.7	<a href="#">getShuntVoltage_mV()</a>	35
4.4.3.8	<a href="#">main()</a>	35
4.4.3.9	<a href="#">setCalibration_16V_400mA()</a>	36
4.4.3.10	<a href="#">SystemClock_Config()</a>	36
4.4.4	<a href="#">Variable Documentation</a>	36
4.4.4.1	<a href="#">adc_val</a>	36
4.4.4.2	<a href="#">buffer</a>	36
4.4.4.3	<a href="#">data</a>	37
4.4.4.4	<a href="#">hadc1</a>	37
4.4.4.5	<a href="#">hadc2</a>	37
4.4.4.6	<a href="#">hdma_adc1</a>	37
4.4.4.7	<a href="#">hi2c1</a>	37
4.4.4.8	<a href="#">hopamp2</a>	37
4.4.4.9	<a href="#">htim1</a>	38
4.4.4.10	<a href="#">huart2</a>	38
4.4.4.11	<a href="#">ina219_currentDivider_mA</a>	38
4.4.4.12	<a href="#">ina219_powerMultiplier_mW</a>	38
4.4.4.13	<a href="#">uart1</a>	38
4.4.4.14	<a href="#">uart2</a>	38
4.4.4.15	<a href="#">uart_ptr</a>	38
4.5	<a href="#">Src/stm32f3xx_hal_msp.c File Reference</a>	39
4.5.1	<a href="#">Function Documentation</a>	40
4.5.1.1	<a href="#">HAL_ADC_MspDeInit()</a>	40
4.5.1.2	<a href="#">HAL_ADC_MspInit()</a>	40
4.5.1.3	<a href="#">HAL_I2C_MspDeInit()</a>	41
4.5.1.4	<a href="#">HAL_I2C_MspInit()</a>	41
4.5.1.5	<a href="#">HAL_MspInit()</a>	41

4.5.1.6	HAL_OPAMP_MspDeInit()	41
4.5.1.7	HAL_OPAMP_MspInit()	42
4.5.1.8	HAL_TIM_Base_MspDeInit()	42
4.5.1.9	HAL_TIM_Base_MspInit()	43
4.5.1.10	HAL_UART_MspDeInit()	43
4.5.1.11	HAL_UART_MspInit()	43
4.5.2	Variable Documentation	44
4.5.2.1	hdma_adc1	44
4.6	Src/stm32f3xx_it.c File Reference	44
4.6.1	Detailed Description	47
4.6.2	Macro Definition Documentation	47
4.6.2.1	BUFFER_SIZE	47
4.6.2.2	DATA_SIZE	47
4.6.2.3	DMA_SIZE	48
4.6.2.4	IN_SIZE	48
4.6.2.5	INA219_ADDRESS	48
4.6.2.6	INA219_REG_BUSVOLTAGE	48
4.6.2.7	INA219_REG_CALIBRATION	48
4.6.2.8	INA219_REG_CONFIG	48
4.6.2.9	INA219_REG_CURRENT	48
4.6.2.10	INA219_REG_POWER	48
4.6.2.11	INA219_REG_SHUNTVOLTAGE	49
4.6.3	Enumeration Type Documentation	49
4.6.3.1	COMMAND	49
4.6.3.2	MEASUREMENT	49
4.6.4	Function Documentation	49
4.6.4.1	ADC1_2_IRQHandler()	49
4.6.4.2	BusFault_Handler()	50
4.6.4.3	cmdI2C()	50
4.6.4.4	cmdParser()	50



4.6.4.5	<a href="#">convertToCommand()</a>	50
4.6.4.6	<a href="#">DebugMon_Handler()</a>	50
4.6.4.7	<a href="#">DMA1_Channel1_IRQHandler()</a>	51
4.6.4.8	<a href="#">EXTI15_10_IRQHandler()</a>	51
4.6.4.9	<a href="#">getBusVoltage_V()</a>	51
4.6.4.10	<a href="#">getCurrent_mA()</a>	51
4.6.4.11	<a href="#">getPower_mW()</a>	52
4.6.4.12	<a href="#">getShuntVoltage_mV()</a>	52
4.6.4.13	<a href="#">HAL_ADC_ConvCpltCallback()</a>	52
4.6.4.14	<a href="#">HAL_GPIO_EXTI_Callback()</a>	54
4.6.4.15	<a href="#">HAL_I2C_ErrorCallback()</a>	54
4.6.4.16	<a href="#">HAL_I2C_MasterRxCpltCallback()</a>	54
4.6.4.17	<a href="#">HAL_I2C_MasterTxCpltCallback()</a>	55
4.6.4.18	<a href="#">HAL_TIM_PeriodElapsedCallback()</a>	55
4.6.4.19	<a href="#">HAL_UART_ErrorCallback()</a>	55
4.6.4.20	<a href="#">HAL_UART_RxCpltCallback()</a>	55
4.6.4.21	<a href="#">HAL_UART_TxCpltCallback()</a>	56
4.6.4.22	<a href="#">HardFault_Handler()</a>	56
4.6.4.23	<a href="#">I2C1_ER_IRQHandler()</a>	56
4.6.4.24	<a href="#">I2C1_EV_IRQHandler()</a>	56
4.6.4.25	<a href="#">MemManage_Handler()</a>	56
4.6.4.26	<a href="#">NMI_Handler()</a>	57
4.6.4.27	<a href="#">PendSV_Handler()</a>	57
4.6.4.28	<a href="#">setupTimerDMA()</a>	57
4.6.4.29	<a href="#">setupTimerIRQ()</a>	57
4.6.4.30	<a href="#">SVC_Handler()</a>	57
4.6.4.31	<a href="#">SysTick_Handler()</a>	57
4.6.4.32	<a href="#">TIM1_UP_TIM16_IRQHandler()</a>	58
4.6.4.33	<a href="#">UsageFault_Handler()</a>	58
4.6.4.34	<a href="#">USART2_IRQHandler()</a>	58

4.6.5	Variable Documentation	58
4.6.5.1	_cnt	58
4.6.5.2	_freq	58
4.6.5.3	adc_buffer	58
4.6.5.4	adc_val	59
4.6.5.5	buffer	59
4.6.5.6	cmd	59
4.6.5.7	cnt_tim	59
4.6.5.8	data	59
4.6.5.9	hadc1	59
4.6.5.10	hadc2	59
4.6.5.11	hdma_adc1	60
4.6.5.12	hi2c1	60
4.6.5.13	htim1	60
4.6.5.14	huart2	60
4.6.5.15	measurement	60
4.6.5.16	tx_flag	60
4.6.5.17	uart1	61
4.6.5.18	uart2	61
4.6.5.19	uart_ptr	61
4.7	Src/system_stm32f3xx.c File Reference	61
4.7.1	Detailed Description	62

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

CMSIS . . . . .	5
Stm32f3xx_system . . . . .	6
STM32F3xx_System_Private_Includes . . . . .	7
STM32F3xx_System_Private_TypesDefinitions . . . . .	8
STM32F3xx_System_Private_Defines . . . . .	9
STM32F3xx_System_Private_Macros . . . . .	10
STM32F3xx_System_Private_Variables . . . . .	11
STM32F3xx_System_Private_FunctionPrototypes . . . . .	12
STM32F3xx_System_Private_Functions . . . . .	13



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

Inc/ <a href="#">main.h</a>	
: Header for <a href="#">main.c</a> file. This file contains the common defines of the application . . . . .	15
Inc/ <a href="#">stm32f3xx_hal_conf.h</a>	
HAL configuration file . . . . .	19
Inc/ <a href="#">stm32f3xx_it.h</a>	
This file contains the headers of the interrupt handlers . . . . .	26
Src/ <a href="#">main.c</a>	
: Main program body . . . . .	30
Src/ <a href="#">stm32f3xx_hal_msp.c</a>	
. . . . .	39
Src/ <a href="#">stm32f3xx_it.c</a>	
Interrupt Service Routines . . . . .	44
Src/ <a href="#">system_stm32f3xx.c</a>	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File . . . . .	61

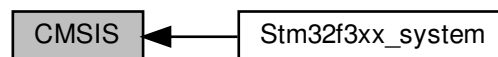


## Chapter 3

# Module Documentation

### 3.1 CMSIS

Collaboration diagram for CMSIS:



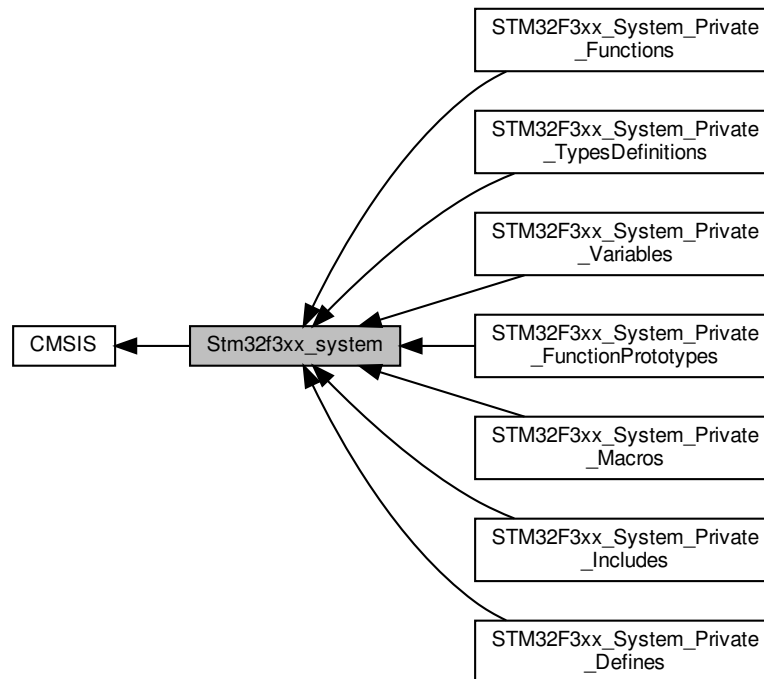
#### Modules

- [Stm32f3xx\\_system](#)

#### 3.1.1 Detailed Description

## 3.2 Stm32f3xx\_system

Collaboration diagram for Stm32f3xx\_system:



### Modules

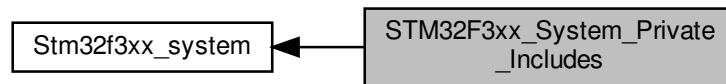
- [STM32F3xx\\_System\\_Private\\_Includes](#)
- [STM32F3xx\\_System\\_Private\\_TypesDefinitions](#)
- [STM32F3xx\\_System\\_Private\\_Defines](#)
- [STM32F3xx\\_System\\_Private\\_Macros](#)
- [STM32F3xx\\_System\\_Private\\_Variables](#)
- [STM32F3xx\\_System\\_Private\\_FunctionPrototypes](#)
- [STM32F3xx\\_System\\_Private\\_Functions](#)

### 3.2.1 Detailed Description



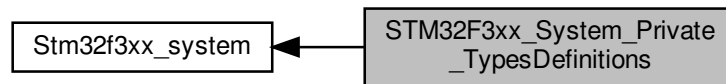
### 3.3 STM32F3xx\_System\_Private\_Includes

Collaboration diagram for STM32F3xx\_System\_Private\_Includes:



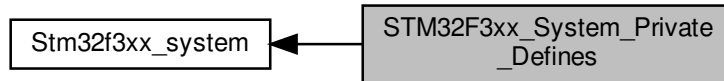
### 3.4 STM32F3xx\_System\_Private\_TypesDefinitions

Collaboration diagram for STM32F3xx\_System\_Private\_TypesDefinitions:



## 3.5 STM32F3xx\_System\_Private\_Defines

Collaboration diagram for STM32F3xx\_System\_Private\_Defines:



### Macros

- `#define HSE_VALUE ((uint32_t)8000000)`
- `#define HSI_VALUE ((uint32_t)8000000)`
- `#define VECT_TAB_OFFSET 0x0`

### 3.5.1 Detailed Description

### 3.5.2 Macro Definition Documentation

#### 3.5.2.1 HSE\_VALUE

```
#define HSE_VALUE ((uint32_t)8000000)
```

Default value of the External oscillator in Hz. This value can be provided and adapted by the user application.

#### 3.5.2.2 HSI\_VALUE

```
#define HSI_VALUE ((uint32_t)8000000)
```

Default value of the Internal oscillator in Hz. This value can be provided and adapted by the user application.

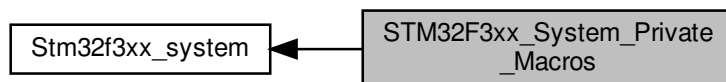
#### 3.5.2.3 VECT\_TAB\_OFFSET

```
#define VECT_TAB_OFFSET 0x0
```

< Uncomment the following line if you need to relocate your vector Table in Internal SRAM. Vector Table base offset field. This value must be a multiple of 0x200.

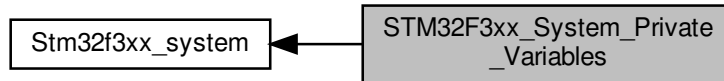
### 3.6 STM32F3xx\_System\_Private\_Macros

Collaboration diagram for STM32F3xx\_System\_Private\_Macros:



## 3.7 STM32F3xx\_System\_Private\_Variables

Collaboration diagram for STM32F3xx\_System\_Private\_Variables:



### Variables

- uint32\_t [SystemCoreClock](#) = 8000000
- 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

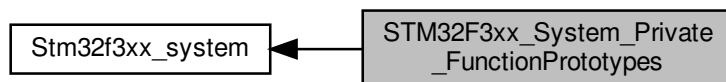
```
const uint8_t APBPrescTable[8] = {0, 0, 0, 0, 1, 2, 3, 4}
```

#### 3.7.2.3 SystemCoreClock

```
uint32_t SystemCoreClock = 8000000
```

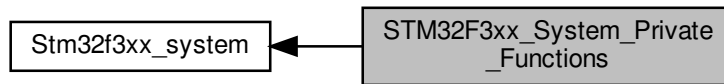
### 3.8 STM32F3xx\_System\_Private\_FunctionPrototypes

Collaboration diagram for STM32F3xx\_System\_Private\_FunctionPrototypes:



## 3.9 STM32F3xx\_System\_Private\_Functions

Collaboration diagram for STM32F3xx\_System\_Private\_Functions:



### Functions

- void [SystemInit](#) (void)  
*Setup the microcontroller system Initialize the FPU setting, vector table location and the PLL configuration is reset.*
- void [SystemCoreClockUpdate](#) (void)  
*Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.*

#### 3.9.1 Detailed Description

#### 3.9.2 Function Documentation

##### 3.9.2.1 SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (
    void )
```

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

#### Note

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

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

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

(\*) HSI\_VALUE is a constant defined in stm32f3xx\_hal.h file (default value 8 MHz) but the real value may vary depending on the variations in voltage and temperature.

(\*\*) HSE\_VALUE is a constant defined in stm32f3xx\_hal.h file (default value 8 MHz), user has to ensure that HSE\_VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

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

**Parameters**

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

**Return values**

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

**3.9.2.2 SystemInit()**

```
void SystemInit (
    void )
```

Setup the microcontroller system Initialize the FPU setting, vector table location and the PLL configuration is reset.

**Parameters**

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

**Return values**

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



## Chapter 4

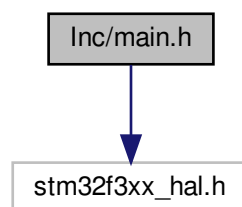
# File Documentation

### 4.1 Inc/main.h File Reference

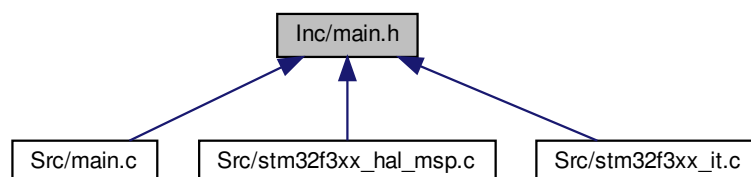
: Header for [main.c](#) file. This file contains the common defines of the application.

```
#include "stm32f3xx_hal.h"
```

Include dependency graph for main.h:



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



## Macros

- `#define B1_Pin` GPIO\_PIN\_13
- `#define B1_GPIO_Port` GPIOC
- `#define B1_EXTI_IRQn` EXTI15\_10\_IRQn
- `#define RCC_OSC32_OUT_Pin` GPIO\_PIN\_15
- `#define RCC_OSC32_OUT_GPIO_Port` GPIOC
- `#define USART_TX_Pin` GPIO\_PIN\_2
- `#define USART_TX_GPIO_Port` GPIOA
- `#define USART_RX_Pin` GPIO\_PIN\_3
- `#define USART_RX_GPIO_Port` GPIOA
- `#define LD2_Pin` GPIO\_PIN\_5
- `#define LD2_GPIO_Port` GPIOA
- `#define TMS_Pin` GPIO\_PIN\_13
- `#define TMS_GPIO_Port` GPIOA
- `#define TCK_Pin` GPIO\_PIN\_14
- `#define TCK_GPIO_Port` GPIOA
- `#define SWO_Pin` GPIO\_PIN\_3
- `#define SWO_GPIO_Port` GPIOB

## Functions

- void `Error_Handler` (void)

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

### 4.1.1 Detailed Description

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

## Attention

© Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: [opensource.org/licenses/BSD-3-Clause](https://opensource.org/licenses/BSD-3-Clause)

### 4.1.2 Macro Definition Documentation

#### 4.1.2.1 B1\_EXTI\_IRQn

```
#define B1_EXTI_IRQn EXTI15_10_IRQn
```

#### 4.1.2.2 B1\_GPIO\_Port

```
#define B1_GPIO_Port GPIOC
```

#### 4.1.2.3 B1\_Pin

```
#define B1_Pin GPIO_PIN_13
```

#### 4.1.2.4 LD2\_GPIO\_Port

```
#define LD2_GPIO_Port GPIOA
```

#### 4.1.2.5 LD2\_Pin

```
#define LD2_Pin GPIO_PIN_5
```

#### 4.1.2.6 RCC\_OSC32\_OUT\_GPIO\_Port

```
#define RCC_OSC32_OUT_GPIO_Port GPIOC
```

#### 4.1.2.7 RCC\_OSC32\_OUT\_Pin

```
#define RCC_OSC32_OUT_Pin GPIO_PIN_15
```

#### 4.1.2.8 SWO\_GPIO\_Port

```
#define SWO_GPIO_Port GPIOB
```

#### 4.1.2.9 SWO\_Pin

```
#define SWO_Pin GPIO_PIN_3
```

#### 4.1.2.10 TCK\_GPIO\_Port

```
#define TCK_GPIO_Port GPIOA
```

#### 4.1.2.11 TCK\_Pin

```
#define TCK_Pin GPIO_PIN_14
```

#### 4.1.2.12 TMS\_GPIO\_Port

```
#define TMS_GPIO_Port GPIOA
```

#### 4.1.2.13 TMS\_Pin

```
#define TMS_Pin GPIO_PIN_13
```

#### 4.1.2.14 USART\_RX\_GPIO\_Port

```
#define USART_RX_GPIO_Port GPIOA
```

#### 4.1.2.15 USART\_RX\_Pin

```
#define USART_RX_Pin GPIO_PIN_3
```

#### 4.1.2.16 USART\_TX\_GPIO\_Port

```
#define USART_TX_GPIO_Port GPIOA
```

## 4.1.2.17 USART\_TX\_Pin

```
#define USART_TX_Pin GPIO_PIN_2
```

## 4.1.3 Function Documentation

## 4.1.3.1 Error\_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

Return values

None	
------	--

## 4.2 Inc/stm32f3xx\_hal\_conf.h File Reference

HAL configuration file.

```
#include "stm32f3xx_hal_rcc.h"
#include "stm32f3xx_hal_gpio.h"
#include "stm32f3xx_hal_dma.h"
#include "stm32f3xx_hal_cortex.h"
#include "stm32f3xx_hal_adc.h"
#include "stm32f3xx_hal_flash.h"
#include "stm32f3xx_hal_i2c.h"
#include "stm32f3xx_hal_opamp.h"
#include "stm32f3xx_hal_pwr.h"
#include "stm32f3xx_hal_tim.h"
#include "stm32f3xx_hal_uart.h"
```

Include dependency graph for stm32f3xx\_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_OPAMP_MODULE_ENABLED`
- `#define HAL_TIM_MODULE_ENABLED`

- `#define HAL_UART_MODULE_ENABLED`
- `#define HAL_GPIO_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 HAL_I2C_MODULE_ENABLED`
- `#define HSE_VALUE ((uint32_t)8000000)`  
*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).*
- `#define HSE_STARTUP_TIMEOUT ((uint32_t)100)`  
*In the following line adjust the External High Speed oscillator (HSE) Startup Timeout value.*
- `#define HSI_VALUE ((uint32_t)8000000)`  
*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).*
- `#define HSI_STARTUP_TIMEOUT ((uint32_t)5000)`  
*In the following line adjust the Internal High Speed oscillator (HSI) Startup Timeout value.*
- `#define LSI_VALUE ((uint32_t)40000)`  
*Internal Low Speed oscillator (LSI) value.*
- `#define LSE_VALUE ((uint32_t)32768)`  
*External Low Speed oscillator (LSE) value.*
- `#define LSE_STARTUP_TIMEOUT ((uint32_t)5000)`  
*Time out for LSE start up value in ms.*
- `#define EXTERNAL_CLOCK_VALUE ((uint32_t)8000000)`  
*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 ((uint32_t)3300)`  
*This is the HAL system configuration section.*
- `#define TICK_INT_PRIORITY ((uint32_t)0)`
- `#define USE_RTOS 0`
- `#define PREFETCH_ENABLE 1`
- `#define INSTRUCTION_CACHE_ENABLE 0`
- `#define DATA_CACHE_ENABLE 0`
- `#define USE_SPI_CRC 0U`  
*Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.*
- `#define assert_param(expr) ((void)0U)`  
*Include module's header file.*

#### 4.2.1 Detailed Description

HAL configuration file.

Attention

© COPYRIGHT(c) 2019 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 4.2.2 Macro Definition Documentation

### 4.2.2.1 assert\_param

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

Include module's header file.

### 4.2.2.2 DATA\_CACHE\_ENABLE

```
#define DATA_CACHE_ENABLE 0
```

#### 4.2.2.3 EXTERNAL\_CLOCK\_VALUE

```
#define EXTERNAL_CLOCK_VALUE ((uint32_t)8000000)
```

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.

- External clock generated through external PLL component on EVAL 303 (based on MCO or crystal)
- External clock not generated on EVAL 373 Value of the External oscillator in Hz

#### 4.2.2.4 HAL\_ADC\_MODULE\_ENABLED

```
#define HAL_ADC_MODULE_ENABLED
```

#### 4.2.2.5 HAL\_CORTEX\_MODULE\_ENABLED

```
#define HAL_CORTEX_MODULE_ENABLED
```

#### 4.2.2.6 HAL\_DMA\_MODULE\_ENABLED

```
#define HAL_DMA_MODULE_ENABLED
```

#### 4.2.2.7 HAL\_FLASH\_MODULE\_ENABLED

```
#define HAL_FLASH_MODULE_ENABLED
```

#### 4.2.2.8 HAL\_GPIO\_MODULE\_ENABLED

```
#define HAL_GPIO_MODULE_ENABLED
```

#### 4.2.2.9 HAL\_I2C\_MODULE\_ENABLED

```
#define HAL_I2C_MODULE_ENABLED
```



#### 4.2.2.10 HAL\_MODULE\_ENABLED

```
#define HAL_MODULE_ENABLED
```

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

#### 4.2.2.11 HAL\_OPAMP\_MODULE\_ENABLED

```
#define HAL_OPAMP_MODULE_ENABLED
```

#### 4.2.2.12 HAL\_PWR\_MODULE\_ENABLED

```
#define HAL_PWR_MODULE_ENABLED
```

#### 4.2.2.13 HAL\_RCC\_MODULE\_ENABLED

```
#define HAL_RCC_MODULE_ENABLED
```

#### 4.2.2.14 HAL\_TIM\_MODULE\_ENABLED

```
#define HAL_TIM_MODULE_ENABLED
```

#### 4.2.2.15 HAL\_UART\_MODULE\_ENABLED

```
#define HAL_UART_MODULE_ENABLED
```

#### 4.2.2.16 HSE\_STARTUP\_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT ((uint32_t)100)
```

In the following line adjust the External High Speed oscillator (HSE) Startup Timeout value.

Time out for HSE start up, in ms

#### 4.2.2.17 HSE\_VALUE

```
#define HSE_VALUE ((uint32_t)8000000)
```

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

#### 4.2.2.18 HSI\_STARTUP\_TIMEOUT

```
#define HSI_STARTUP_TIMEOUT ((uint32_t)5000)
```

In the following line adjust the Internal High Speed oscillator (HSI) Startup Timeout value.

Time out for HSI start up

#### 4.2.2.19 HSI\_VALUE

```
#define HSI_VALUE ((uint32_t)8000000)
```

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

#### 4.2.2.20 INSTRUCTION\_CACHE\_ENABLE

```
#define INSTRUCTION_CACHE_ENABLE 0
```

#### 4.2.2.21 LSE\_STARTUP\_TIMEOUT

```
#define LSE_STARTUP_TIMEOUT ((uint32_t)5000)
```

Time out for LSE start up value in ms.

Time out for LSE start up, in ms

#### 4.2.2.22 LSE\_VALUE

```
#define LSE_VALUE ((uint32_t)32768)
```

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

#### 4.2.2.23 LSI\_VALUE

```
#define LSI_VALUE ((uint32_t)40000)
```

Internal Low Speed oscillator (LSI) value.

#### 4.2.2.24 PREFETCH\_ENABLE

```
#define PREFETCH_ENABLE 1
```

#### 4.2.2.25 TICK\_INT\_PRIORITY

```
#define TICK_INT_PRIORITY ((uint32_t)0)
```

tick interrupt priority (lowest by default)

#### 4.2.2.26 USE\_RTOS

```
#define USE_RTOS 0
```

#### 4.2.2.27 USE\_SPI\_CRC

```
#define USE_SPI_CRC 0U
```

Uncomment the line below to expanse the "assert\_param" macro in the HAL drivers code.

#### 4.2.2.28 VDD\_VALUE

```
#define VDD_VALUE ((uint32_t)3300)
```

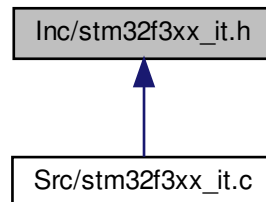
This is the HAL system configuration section.

Value of VDD in mv

### 4.3 Inc/stm32f3xx\_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 [SVC\\_Handler](#) (void)  
*This function handles System service call via SWI instruction.*
- void [DebugMon\\_Handler](#) (void)  
*This function handles Debug monitor.*
- void [PendSV\\_Handler](#) (void)  
*This function handles Pendable request for system service.*
- void [SysTick\\_Handler](#) (void)  
*This function handles System tick timer.*
- void [DMA1\\_Channel1\\_IRQHandler](#) (void)  
*This function handles DMA1 channel1 global interrupt.*
- void [ADC1\\_2\\_IRQHandler](#) (void)  
*This function handles ADC1 and ADC2 interrupts.*
- void [TIM1\\_UP\\_TIM16\\_IRQHandler](#) (void)  
*This function handles TIM1 update and TIM16 interrupts.*
- void [I2C1\\_EV\\_IRQHandler](#) (void)  
*This function handles I2C1 event global interrupt / I2C1 wake-up interrupt through EXT line 23.*
- void [I2C1\\_ER\\_IRQHandler](#) (void)  
*This function handles I2C1 error interrupt.*
- void [USART2\\_IRQHandler](#) (void)  
*This function handles USART2 global interrupt / USART2 wake-up interrupt through EXT line 26.*
- void [EXTI15\\_10\\_IRQHandler](#) (void)  
*This function handles EXTI line[15:10] interrupts.*

### 4.3.1 Detailed Description

This file contains the headers of the interrupt handlers.

#### Attention

© Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: [opensource.org/licenses/BSD-3-Clause](https://opensource.org/licenses/BSD-3-Clause)

### 4.3.2 Function Documentation

#### 4.3.2.1 ADC1\_2\_IRQHandler()

```
void ADC1_2_IRQHandler (  
    void )
```

This function handles ADC1 and ADC2 interrupts.

#### 4.3.2.2 BusFault\_Handler()

```
void BusFault_Handler (  
    void )
```

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

#### 4.3.2.3 DebugMon\_Handler()

```
void DebugMon_Handler (  
    void )
```

This function handles Debug monitor.

#### 4.3.2.4 DMA1\_Channel1\_IRQHandler()

```
void DMA1_Channel1_IRQHandler (
    void )
```

This function handles DMA1 channel1 global interrupt.

#### 4.3.2.5 EXTI15\_10\_IRQHandler()

```
void EXTI15_10_IRQHandler (
    void )
```

This function handles EXTI line[15:10] interrupts.

#### 4.3.2.6 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

#### 4.3.2.7 I2C1\_ER\_IRQHandler()

```
void I2C1_ER_IRQHandler (
    void )
```

This function handles I2C1 error interrupt.

#### 4.3.2.8 I2C1\_EV\_IRQHandler()

```
void I2C1_EV_IRQHandler (
    void )
```

This function handles I2C1 event global interrupt / I2C1 wake-up interrupt through EXT line 23.

#### 4.3.2.9 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 4.3.2.10 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

#### 4.3.2.11 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

#### 4.3.2.12 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

#### 4.3.2.13 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

#### 4.3.2.14 TIM1\_UP\_TIM16\_IRQHandler()

```
void TIM1_UP_TIM16_IRQHandler (
    void )
```

This function handles TIM1 update and TIM16 interrupts.

#### 4.3.2.15 UsageFault\_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Undefined instruction or illegal state.

#### 4.3.2.16 USART2\_IRQHandler()

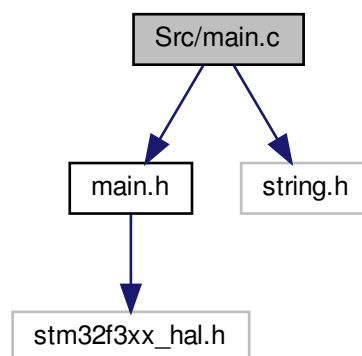
```
void USART2_IRQHandler (
    void )
```

This function handles USART2 global interrupt / USART2 wake-up interrupt through EXT line 26.

## 4.4 Src/main.c File Reference

: Main program body

```
#include "main.h"
#include <string.h>
Include dependency graph for main.c:
```



## Macros

- #define BUFFER\_SIZE 32
- #define DATA\_SIZE 2
- #define IN\_SIZE 32
- #define INA219\_ADDRESS (0x40<<1)
- #define INA219\_REG\_CONFIG (0x00)
- #define INA219\_REG\_SHUNTVOLTAGE (0x01)
- #define INA219\_REG\_BUSVOLTAGE (0x02)
- #define INA219\_REG\_POWER (0x03)
- #define INA219\_REG\_CURRENT (0x04)
- #define INA219\_REG\_CALIBRATION (0x05)



## Functions

- void [SystemClock\\_Config](#) (void)  
*System Clock Configuration.*
- void [setCalibration\\_16V\\_400mA](#) (void)  
*Sets up INA219 configuration (0x019F) and calibration (0x2000) registers.*
- float [getCurrent\\_mA](#) (uint16\_t value)  
*Calculates current value.*
- float [getPower\\_mW](#) (uint16\_t value)  
*Calculates power value.*
- float [getBusVoltage\\_V](#) (uint16\_t value)  
*Calculates bus voltage.*
- float [getShuntVoltage\\_mV](#) (uint16\_t value)  
*Calculates shunt voltage.*
- int [getCurrentDivider\\_mA](#) (void)  
*Getter for current divider.*
- int [getPowerMultiplier\\_mW](#) (void)  
*Getter for power multiplier.*
- int [main](#) (void)  
*The application entry point.*
- void [Error\\_Handler](#) (void)  
*This function is executed in case of error occurrence.*

## Variables

- ADC\_HandleTypeDef [hadc1](#)
- ADC\_HandleTypeDef [hadc2](#)
- DMA\_HandleTypeDef [hdma\\_adc1](#)
- I2C\_HandleTypeDef [hi2c1](#)
- OPAMP\_HandleTypeDef [hopamp2](#)
- TIM\_HandleTypeDef [htim1](#)
- UART\_HandleTypeDef [huart2](#)
- uint8\_t [buffer](#) [BUFFER\_SIZE]
- uint8\_t [data](#) [DATA\_SIZE]
- uint8\_t [uart1](#) [IN\_SIZE]
- uint8\_t [uart2](#) [IN\_SIZE]
- uint8\_t \* [uart\\_ptr](#)
- uint16\_t [adc\\_val](#)
- uint32\_t [ina219\\_currentDivider\\_mA](#)
- uint32\_t [ina219\\_powerMultiplier\\_mW](#)

### 4.4.1 Detailed Description

: Main program body

#### Attention

© Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: [opensource.org/licenses/BSD-3-Clause](https://opensource.org/licenses/BSD-3-Clause)

## 4.4.2 Macro Definition Documentation

### 4.4.2.1 BUFFER\_SIZE

```
#define BUFFER_SIZE 32
```

### 4.4.2.2 DATA\_SIZE

```
#define DATA_SIZE 2
```

### 4.4.2.3 IN\_SIZE

```
#define IN_SIZE 32
```

### 4.4.2.4 INA219\_ADDRESS

```
#define INA219_ADDRESS (0x40<<1)
```

### 4.4.2.5 INA219\_REG\_BUSVOLTAGE

```
#define INA219_REG_BUSVOLTAGE (0x02)
```

#### 4.4.2.6 INA219\_REG\_CALIBRATION

```
#define INA219_REG_CALIBRATION (0x05)
```

#### 4.4.2.7 INA219\_REG\_CONFIG

```
#define INA219_REG_CONFIG (0x00)
```

#### 4.4.2.8 INA219\_REG\_CURRENT

```
#define INA219_REG_CURRENT (0x04)
```

#### 4.4.2.9 INA219\_REG\_POWER

```
#define INA219_REG_POWER (0x03)
```

#### 4.4.2.10 INA219\_REG\_SHUNTVOLTAGE

```
#define INA219_REG_SHUNTVOLTAGE (0x01)
```

### 4.4.3 Function Documentation

#### 4.4.3.1 Error\_Handler()

```
void Error_Handler (  
    void )
```

This function is executed in case of error occurrence.

##### Return values

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

#### 4.4.3.2 getBusVoltage\_V()

```
float getBusVoltage_V (
    uint16_t value )
```

Calculates bus voltage.

##### Parameters

2	bytes raw value
---	-----------------

##### Returns

returns measured bus voltage in V

#### 4.4.3.3 getCurrent\_mA()

```
float getCurrent_mA (
    uint16_t value )
```

Calculates current value.

##### Parameters

2	bytes raw value
---	-----------------

##### Returns

returns measured current in mA

#### 4.4.3.4 getCurrentDivider\_mA()

```
int getCurrentDivider_mA (
    void )
```

Getter for current divider.

##### Returns

returns current divider in mA

#### 4.4.3.5 getPower\_mW()

```
float getPower_mW (
    uint16_t value )
```

Calculates power value.

**Parameters**

2	bytes raw value
---	-----------------

**Returns**

returns measured power in mW

**4.4.3.6 getPowerMultiplier\_mW()**

```
int getPowerMultiplier_mW (  
    void )
```

Getter for power multiplier.

**Returns**

returns power multiplier in mW

**4.4.3.7 getShuntVoltage\_mV()**

```
float getShuntVoltage_mV (  
    uint16_t value )
```

Calculates shunt voltage.

**Parameters**

2	bytes raw value
---	-----------------

**Returns**

returns measured shunt voltage in mV

**4.4.3.8 main()**

```
int main (  
    void )
```

The application entry point.

**Return values**

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

**4.4.3.9 setCalibration\_16V\_400mA()**

```
void setCalibration_16V_400mA (  
    void )
```

Sets up INA219 configuration (0x019F) and calibration (0x2000) registers.

**4.4.3.10 SystemClock\_Config()**

```
void SystemClock_Config (  
    void )
```

System Clock Configuration.

**Return values**

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

Initializes the CPU, AHB and APB busses clocks

Initializes the CPU, AHB and APB busses clocks

**4.4.4 Variable Documentation****4.4.4.1 adc\_val**

```
uint16_t adc_val
```

**4.4.4.2 buffer**

```
uint8_t buffer[BUFFER_SIZE]
```

#### 4.4.4.3 data

```
uint8_t data[DATA_SIZE]
```

#### 4.4.4.4 hadc1

```
ADC_HandleTypeDef hadc1
```

#### 4.4.4.5 hadc2

```
ADC_HandleTypeDef hadc2
```

#### 4.4.4.6 hdma\_adc1

```
DMA_HandleTypeDef hdma_adc1
```

File Name : [stm32f3xx\\_hal\\_msp.c](#) Description : This file provides code for the MSP Initialization and de-Initialization codes.

#### Attention

© Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: [opensource.org/licenses/BSD-3-Clause](https://opensource.org/licenses/BSD-3-Clause)

#### 4.4.4.7 hi2c1

```
I2C_HandleTypeDef hi2c1
```

#### 4.4.4.8 hopamp2

```
OPAMP_HandleTypeDef hopamp2
```

#### 4.4.4.9 htim1

TIM\_HandleTypeDef htim1

#### 4.4.4.10 huart2

UART\_HandleTypeDef huart2

#### 4.4.4.11 ina219\_currentDivider\_mA

uint32\_t ina219\_currentDivider\_mA

#### 4.4.4.12 ina219\_powerMultiplier\_mW

uint32\_t ina219\_powerMultiplier\_mW

#### 4.4.4.13 uart1

uint8\_t uart1[IN\_SIZE]

#### 4.4.4.14 uart2

uint8\_t uart2[IN\_SIZE]

#### 4.4.4.15 uart\_ptr

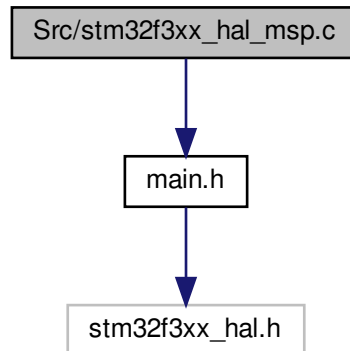
uint8\_t\* uart\_ptr



## 4.5 Src/stm32f3xx\_hal\_msp.c File Reference

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

Include dependency graph for stm32f3xx\_hal\_msp.c:



### Functions

- void [HAL\\_MspInit](#) (void)
- void [HAL\\_ADC\\_MspInit](#) (ADC\_HandleTypeDef \*hadc)  
*ADC MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_ADC\\_MspDeInit](#) (ADC\_HandleTypeDef \*hadc)  
*ADC MSP De-Initialization This function freeze the hardware resources used in this example.*
- void [HAL\\_I2C\\_MspInit](#) (I2C\_HandleTypeDef \*hi2c)  
*I2C MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_I2C\\_MspDeInit](#) (I2C\_HandleTypeDef \*hi2c)  
*I2C MSP De-Initialization This function freeze the hardware resources used in this example.*
- void [HAL\\_OPAMP\\_MspInit](#) (OPAMP\_HandleTypeDef \*hopamp)  
*OPAMP MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_OPAMP\\_MspDeInit](#) (OPAMP\_HandleTypeDef \*hopamp)  
*OPAMP MSP De-Initialization This function freeze the hardware resources used in this example.*
- void [HAL\\_TIM\\_Base\\_MspInit](#) (TIM\_HandleTypeDef \*htim\_base)  
*TIM\_Base MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_TIM\\_Base\\_MspDeInit](#) (TIM\_HandleTypeDef \*htim\_base)  
*TIM\_Base MSP De-Initialization This function freeze the hardware resources used in this example.*
- void [HAL\\_UART\\_MspInit](#) (UART\_HandleTypeDef \*huart)  
*UART MSP Initialization This function configures the hardware resources used in this example.*
- void [HAL\\_UART\\_MspDeInit](#) (UART\_HandleTypeDef \*huart)  
*UART MSP De-Initialization This function freeze the hardware resources used in this example.*

### Variables

- DMA\_HandleTypeDef [hdma\\_adc1](#)

## 4.5.1 Function Documentation

### 4.5.1.1 HAL\_ADC\_MspDeInit()

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

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

#### Parameters

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

#### Return values

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

ADC1 GPIO Configuration PA0 —> ADC1\_IN1

Uncomment the line below to disable the "ADC1\_2\_IRQn" interrupt Be aware, disabling shared interrupt may affect other IPs

ADC2 GPIO Configuration PC2 —> ADC2\_IN8

Uncomment the line below to disable the "ADC1\_2\_IRQn" interrupt Be aware, disabling shared interrupt may affect other IPs

### 4.5.1.2 HAL\_ADC\_MspInit()

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

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

#### Parameters

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

#### Return values

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

ADC1 GPIO Configuration PA0 —> ADC1\_IN1

ADC2 GPIO Configuration PC2 ----> ADC2\_IN8

#### 4.5.1.3 HAL\_I2C\_MspDeInit()

```
void HAL_I2C_MspDeInit (
    I2C_HandleTypeDef * hi2c )
```

I2C MSP De-Initialization This function freeze the hardware resources used in this example.

##### Parameters

<i>hi2c</i>	I2C handle pointer
-------------	--------------------

##### Return values

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

I2C1 GPIO Configuration PA15 ----> I2C1\_SCL PB7 ----> I2C1\_SDA

#### 4.5.1.4 HAL\_I2C\_MspInit()

```
void HAL_I2C_MspInit (
    I2C_HandleTypeDef * hi2c )
```

I2C MSP Initialization This function configures the hardware resources used in this example.

##### Parameters

<i>hi2c</i>	I2C handle pointer
-------------	--------------------

##### Return values

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

I2C1 GPIO Configuration PA15 ----> I2C1\_SCL PB7 ----> I2C1\_SDA

#### 4.5.1.5 HAL\_MspInit()

```
void HAL_MspInit (
    void )
```

Initializes the Global MSP.

#### 4.5.1.6 HAL\_OPAMP\_MspDeInit()

```
void HAL_OPAMP_MspDeInit (
    OPAMP_HandleTypeDef * hopamp )
```

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

**Parameters**

<i>hopamp</i>	OPAMP handle pointer
---------------	----------------------

**Return values**

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

OPAMP2 GPIO Configuration PA6 ----> OPAMP2\_VOUT PA7 ----> OPAMP2\_VINP PC5 ----> OPAMP2\_VINM

**4.5.1.7 HAL\_OPAMP\_MspInit()**

```
void HAL_OPAMP_MspInit (
    OPAMP_HandleTypeDef * hopamp )
```

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

**Parameters**

<i>hopamp</i>	OPAMP handle pointer
---------------	----------------------

**Return values**

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

OPAMP2 GPIO Configuration PA6 ----> OPAMP2\_VOUT PA7 ----> OPAMP2\_VINP PC5 ----> OPAMP2\_VINM

**4.5.1.8 HAL\_TIM\_Base\_MspDeInit()**

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

TIM\_Base MSP De-Initialization This function freeze the hardware resources used in this example.

**Parameters**

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

**Return values**

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

#### 4.5.1.9 HAL\_TIM\_Base\_MspInit()

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

TIM\_Base MSP Initialization This function configures the hardware resources used in this example.

##### Parameters

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

##### Return values

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

#### 4.5.1.10 HAL\_UART\_MspDeInit()

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

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

##### Parameters

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

##### Return values

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

USART2 GPIO Configuration PA2 ----> USART2\_TX PA3 ----> USART2\_RX

#### 4.5.1.11 HAL\_UART\_MspInit()

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

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

##### Parameters

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

##### Return values

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

USART2 GPIO Configuration PA2 -----> USART2\_TX PA3 -----> USART2\_RX

## 4.5.2 Variable Documentation

### 4.5.2.1 hdma\_adc1

DMA\_HandleTypeDef hdma\_adc1

File Name : [stm32f3xx\\_hal\\_msp.c](#) Description : This file provides code for the MSP Initialization and de-Initialization codes.

Attention

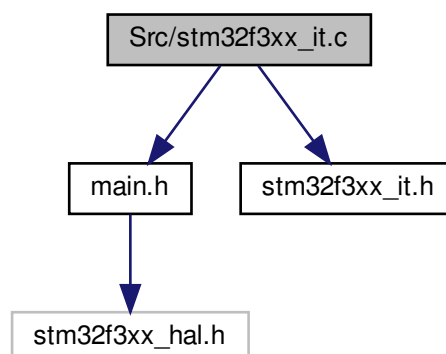
© Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: [opensource.org/licenses/BSD-3-Clause](https://opensource.org/licenses/BSD-3-Clause)

## 4.6 Src/stm32f3xx\_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"
#include "stm32f3xx_it.h"
Include dependency graph for stm32f3xx_it.c:
```



## Macros

- #define `BUFFER_SIZE` 32
- #define `DATA_SIZE` 2
- #define `IN_SIZE` 32
- #define `DMA_SIZE` 4
- #define `INA219_ADDRESS` (0x40<<1)
- #define `INA219_REG_CONFIG` (0x00)
- #define `INA219_REG_SHUNTVOLTAGE` (0x01)
- #define `INA219_REG_BUSVOLTAGE` (0x02)
- #define `INA219_REG_POWER` (0x03)
- #define `INA219_REG_CURRENT` (0x04)
- #define `INA219_REG_CALIBRATION` (0x05)

## Enumerations

- enum `COMMAND` {  
`CMD_SHUNT_VOLTAGE` = 0x01, `CMD_BUS_VOLTAGE` = 0x02, `CMD_POWER` = 0x03, `CMD_CURRENT` = 0x04,  
`CMD_STOP` = 0x05, `CMD_ADC` = 0x06, `CMD_UNKNOWN` = 0x07 }
- enum `MEASUREMENT` { `MEASURE_INA219` = 0x01, `MEASURE_CUSTOM` = 0x02 }

## Functions

- float `getCurrent_mA` (uint16\_t value)  
*Calculates current value.*
- float `getPower_mW` (uint16\_t value)  
*Calculates power value.*
- float `getBusVoltage_V` (uint16\_t value)  
*Calculates bus voltage.*
- float `getShuntVoltage_mV` (uint16\_t value)  
*Calculates shunt voltage.*
- void `cmdParser` (uint8\_t \*in, uint8\_t cnt)  
*Handles received commands.*
- void `setupTimerDMA` ()  
*Sets up timer counter and frequency for DMA.*
- void `setupTimerIRQ` ()  
*Sets up timer counter and frequency for IRQ.*
- void `convertToCommand` (uint8\_t \*in)  
*Tries to convert received message to command and obtains data.*
- void `cmdI2C` ()  
*Triggers INA219 actions depending on cmd variable.*
- void `HAL_GPIO_EXTI_Callback` (uint16\_t GPIO\_Pin)  
*Switches way of measurements.*
- void `HAL_UART_RxCpltCallback` (UART\_HandleTypeDef \*huart)  
*Handles receiving bytes on UART.*
- void `HAL_ADC_ConvCpltCallback` (ADC\_HandleTypeDef \*hadc)  
*Gets measurement values and send them back to PC via UART.*
- void `HAL_TIM_PeriodElapsedCallback` (TIM\_HandleTypeDef \*htim)  
*Triggers measurements.*
- void `HAL_I2C_MasterRxCpltCallback` (I2C\_HandleTypeDef \*hi2c)

- Handles received bytes on I2C.*
- void [HAL\\_I2C\\_MasterTxCpltCallback](#) (I2C\_HandleTypeDef \*hi2c)
- void [HAL\\_I2C\\_ErrorCallback](#) (I2C\_HandleTypeDef \*hi2c)
- Sends errors.*
- void [HAL\\_UART\\_ErrorCallback](#) (UART\_HandleTypeDef \*huart)
- Sends errors.*
- 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 [SVC\\_Handler](#) (void)
- This function handles System service call via SWI instruction.*
- void [DebugMon\\_Handler](#) (void)
- This function handles Debug monitor.*
- void [PendSV\\_Handler](#) (void)
- This function handles Pendable request for system service.*
- void [SysTick\\_Handler](#) (void)
- This function handles System tick timer.*
- void [DMA1\\_Channel1\\_IRQHandler](#) (void)
- This function handles DMA1 channel1 global interrupt.*
- void [ADC1\\_2\\_IRQHandler](#) (void)
- This function handles ADC1 and ADC2 interrupts.*
- void [TIM1\\_UP\\_TIM16\\_IRQHandler](#) (void)
- This function handles TIM1 update and TIM16 interrupts.*
- void [I2C1\\_EV\\_IRQHandler](#) (void)
- This function handles I2C1 event global interrupt / I2C1 wake-up interrupt through EXT line 23.*
- void [I2C1\\_ER\\_IRQHandler](#) (void)
- This function handles I2C1 error interrupt.*
- void [USART2\\_IRQHandler](#) (void)
- This function handles USART2 global interrupt / USART2 wake-up interrupt through EXT line 26.*
- void [EXTI15\\_10\\_IRQHandler](#) (void)
- This function handles EXTI line[15:10] interrupts.*
- void [HAL\\_UART\\_TxCpltCallback](#) (UART\_HandleTypeDef \*huart)
- Reset tx\_flag when transmit completed.*

## Variables

- uint8\_t [data](#) [DATA\_SIZE]
- uint8\_t [buffer](#) [BUFFER\_SIZE]
- uint8\_t [uart1](#) [IN\_SIZE]
- uint8\_t [uart2](#) [IN\_SIZE]
- uint8\_t \* [uart\\_ptr](#)
- uint16\_t [adc\\_val](#)
- volatile uint16\_t [cnt\\_tim](#)
- volatile uint8\_t [tx\\_flag](#)



- volatile int `_cnt`
- volatile int `_freq`
- uint16\_t `adc_buffer` [`DMA_SIZE`]
- enum `COMMAND cmd`
- enum `MEASUREMENT measurement` = `MEASURE_INA219`
- DMA\_HandleTypeDef `hdma_adc1`
- ADC\_HandleTypeDef `hadc1`
- ADC\_HandleTypeDef `hadc2`
- I2C\_HandleTypeDef `hi2c1`
- TIM\_HandleTypeDef `htim1`
- UART\_HandleTypeDef `huart2`

### 4.6.1 Detailed Description

Interrupt Service Routines.

Attention

© Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: [opensource.org/licenses/BSD-3-Clause](https://opensource.org/licenses/BSD-3-Clause)

### 4.6.2 Macro Definition Documentation

#### 4.6.2.1 BUFFER\_SIZE

```
#define BUFFER_SIZE 32
```

#### 4.6.2.2 DATA\_SIZE

```
#define DATA_SIZE 2
```

#### 4.6.2.3 DMA\_SIZE

```
#define DMA_SIZE 4
```

#### 4.6.2.4 IN\_SIZE

```
#define IN_SIZE 32
```

#### 4.6.2.5 INA219\_ADDRESS

```
#define INA219_ADDRESS (0x40<<1)
```

#### 4.6.2.6 INA219\_REG\_BUSVOLTAGE

```
#define INA219_REG_BUSVOLTAGE (0x02)
```

#### 4.6.2.7 INA219\_REG\_CALIBRATION

```
#define INA219_REG_CALIBRATION (0x05)
```

#### 4.6.2.8 INA219\_REG\_CONFIG

```
#define INA219_REG_CONFIG (0x00)
```

#### 4.6.2.9 INA219\_REG\_CURRENT

```
#define INA219_REG_CURRENT (0x04)
```

#### 4.6.2.10 INA219\_REG\_POWER

```
#define INA219_REG_POWER (0x03)
```

#### 4.6.2.11 INA219\_REG\_SHUNTVOLTAGE

```
#define INA219_REG_SHUNTVOLTAGE (0x01)
```

### 4.6.3 Enumeration Type Documentation

#### 4.6.3.1 COMMAND

```
enum COMMAND
```

##### Enumerator

CMD_SHUNT_VOLTAGE	
CMD_BUS_VOLTAGE	
CMD_POWER	
CMD_CURRENT	
CMD_STOP	
CMD_ADC	
CMD_UNKNOWN	

#### 4.6.3.2 MEASUREMENT

```
enum MEASUREMENT
```

##### Enumerator

MEASURE_INA219	
MEASURE_CUSTOM	

### 4.6.4 Function Documentation

#### 4.6.4.1 ADC1\_2\_IRQHandler()

```
void ADC1_2_IRQHandler (  
    void )
```

This function handles ADC1 and ADC2 interrupts.

#### 4.6.4.2 BusFault\_Handler()

```
void BusFault_Handler (
    void )
```

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

#### 4.6.4.3 cmdI2C()

```
void cmdI2C ( )
```

Triggers INA219 actions depending on cmd variable.

#### 4.6.4.4 cmdParser()

```
void cmdParser (
    uint8_t * in,
    uint8_t cnt )
```

Handles received commands.

##### Parameters

<i>uint8_t</i>	buffer
----------------	--------

#### 4.6.4.5 convertToCommand()

```
void convertToCommand (
    uint8_t * in )
```

Tries to convert received message to command and obtains data.

##### Parameters

<i>uint8_t</i>	buffer
----------------	--------

#### 4.6.4.6 DebugMon\_Handler()

```
void DebugMon_Handler (
```

```
void )
```

This function handles Debug monitor.

#### 4.6.4.7 DMA1\_Channel1\_IRQHandler()

```
void DMA1_Channel1_IRQHandler (
    void )
```

This function handles DMA1 channel1 global interrupt.

#### 4.6.4.8 EXTI15\_10\_IRQHandler()

```
void EXTI15_10_IRQHandler (
    void )
```

This function handles EXTI line[15:10] interrupts.

#### 4.6.4.9 getBusVoltage\_V()

```
float getBusVoltage_V (
    uint16_t value )
```

Calculates bus voltage.

##### Parameters

2	bytes raw value
---	-----------------

##### Returns

returns measured bus voltage in V

#### 4.6.4.10 getCurrent\_mA()

```
float getCurrent_mA (
    uint16_t value )
```

Calculates current value.

**Parameters**

2	bytes raw value
---	-----------------

**Returns**

returns measured current in mA

**4.6.4.11 getPower\_mW()**

```
float getPower_mW (
    uint16_t value )
```

Calculates power value.

**Parameters**

2	bytes raw value
---	-----------------

**Returns**

returns measured power in mW

**4.6.4.12 getShuntVoltage\_mV()**

```
float getShuntVoltage_mV (
    uint16_t value )
```

Calculates shunt voltage.

**Parameters**

2	bytes raw value
---	-----------------

**Returns**

returns measured shunt voltage in mV

**4.6.4.13 HAL\_ADC\_ConvCpltCallback()**

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

Gets measurement values and send them back to PC via UART.

## Parameters

<i>ADC_HandleTypeDef</i>	
--------------------------	--

**4.6.4.14 HAL\_GPIO\_EXTI\_Callback()**

```
void HAL_GPIO_EXTI_Callback (
    uint16_t GPIO_Pin )
```

Switches way of measurements.

## Parameters

<i>GPIO_Pin</i>	
-----------------	--

**4.6.4.15 HAL\_I2C\_ErrorCallback()**

```
void HAL_I2C_ErrorCallback (
    I2C_HandleTypeDef * hi2c )
```

Sends errors.

## Parameters

<i>I2C_HandleTypeDef</i>	
--------------------------	--

**4.6.4.16 HAL\_I2C\_MasterRxCpltCallback()**

```
void HAL_I2C_MasterRxCpltCallback (
    I2C_HandleTypeDef * hi2c )
```

Handles received bytes on I2C.

## Parameters

<i>I2C_HandleTypeDef</i>	
--------------------------	--



#### 4.6.4.17 HAL\_I2C\_MasterTxCpltCallback()

```
void HAL_I2C_MasterTxCpltCallback (
    I2C_HandleTypeDef * hi2c )
```

#### 4.6.4.18 HAL\_TIM\_PeriodElapsedCallback()

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

Triggers measurements.

##### Parameters

<i>TIM_HandleTypeDef</i>	
--------------------------	--

#### 4.6.4.19 HAL\_UART\_ErrorCallback()

```
void HAL_UART_ErrorCallback (
    UART_HandleTypeDef * huart )
```

Sends errors.

##### Parameters

<i>UART_HandleTypeDef</i>	
---------------------------	--

#### 4.6.4.20 HAL\_UART\_RxCpltCallback()

```
void HAL_UART_RxCpltCallback (
    UART_HandleTypeDef * huart )
```

Handles receiving bytes on UART.

##### Parameters

<i>UART_HandleTypeDef</i>	
---------------------------	--

#### 4.6.4.21 HAL\_UART\_TxCpltCallback()

```
void HAL_UART_TxCpltCallback (
    UART_HandleTypeDef * huart )
```

Reset tx\_flag when transmit completed.

##### Parameters

UART_HandleTypeDef	
--------------------	--

#### 4.6.4.22 HardFault\_Handler()

```
void HardFault_Handler (
    void )
```

This function handles Hard fault interrupt.

#### 4.6.4.23 I2C1\_ER\_IRQHandler()

```
void I2C1_ER_IRQHandler (
    void )
```

This function handles I2C1 error interrupt.

#### 4.6.4.24 I2C1\_EV\_IRQHandler()

```
void I2C1_EV_IRQHandler (
    void )
```

This function handles I2C1 event global interrupt / I2C1 wake-up interrupt through EXT line 23.

#### 4.6.4.25 MemManage\_Handler()

```
void MemManage_Handler (
    void )
```

This function handles Memory management fault.

#### 4.6.4.26 NMI\_Handler()

```
void NMI_Handler (
    void )
```

This function handles Non maskable interrupt.

#### 4.6.4.27 PendSV\_Handler()

```
void PendSV_Handler (
    void )
```

This function handles Pendable request for system service.

#### 4.6.4.28 setupTimerDMA()

```
void setupTimerDMA ( )
```

Sets up timer counter and frequency for DMA.

#### 4.6.4.29 setupTimerIRQ()

```
void setupTimerIRQ ( )
```

Sets up timer counter and frequency for IRQ.

#### 4.6.4.30 SVC\_Handler()

```
void SVC_Handler (
    void )
```

This function handles System service call via SWI instruction.

#### 4.6.4.31 SysTick\_Handler()

```
void SysTick_Handler (
    void )
```

This function handles System tick timer.

#### 4.6.4.32 TIM1\_UP\_TIM16\_IRQHandler()

```
void TIM1_UP_TIM16_IRQHandler (  
    void )
```

This function handles TIM1 update and TIM16 interrupts.

#### 4.6.4.33 UsageFault\_Handler()

```
void UsageFault_Handler (  
    void )
```

This function handles Undefined instruction or illegal state.

#### 4.6.4.34 USART2\_IRQHandler()

```
void USART2_IRQHandler (  
    void )
```

This function handles USART2 global interrupt / USART2 wake-up interrupt through EXT line 26.

### 4.6.5 Variable Documentation

#### 4.6.5.1 \_cnt

```
volatile int _cnt
```

#### 4.6.5.2 \_freq

```
volatile int _freq
```

#### 4.6.5.3 adc\_buffer

```
uint16_t adc_buffer[DMA_SIZE]
```

#### 4.6.5.4 adc\_val

```
uint16_t adc_val
```

#### 4.6.5.5 buffer

```
uint8_t buffer[BUFFER\_SIZE]
```

#### 4.6.5.6 cmd

```
enum COMMAND cmd
```

#### 4.6.5.7 cnt\_tim

```
volatile uint16_t cnt_tim
```

#### 4.6.5.8 data

```
uint8_t data[DATA\_SIZE]
```

#### 4.6.5.9 hadc1

```
ADC_HandleTypeDef hadc1
```

#### 4.6.5.10 hadc2

```
ADC_HandleTypeDef hadc2
```

#### 4.6.5.11 hdma\_adc1

DMA\_HandleTypeDef hdma\_adc1

File Name : [stm32f3xx\\_hal\\_msp.c](#) Description : This file provides code for the MSP Initialization and de-Initialization codes.

#### Attention

© Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: [opensource.org/licenses/BSD-3-Clause](https://opensource.org/licenses/BSD-3-Clause)

#### 4.6.5.12 hi2c1

I2C\_HandleTypeDef hi2c1

#### 4.6.5.13 htim1

TIM\_HandleTypeDef htim1

#### 4.6.5.14 huart2

UART\_HandleTypeDef huart2

#### 4.6.5.15 measurement

enum [MEASUREMENT](#) measurement = [MEASURE\\_INA219](#)

#### 4.6.5.16 tx\_flag

volatile uint8\_t tx\_flag

## 4.6.5.17 uart1

```
uint8_t uart1[IN_SIZE]
```

## 4.6.5.18 uart2

```
uint8_t uart2[IN_SIZE]
```

## 4.6.5.19 uart\_ptr

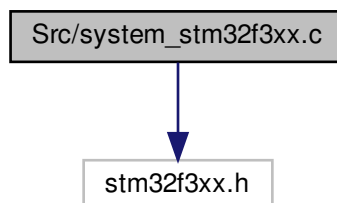
```
uint8_t* uart_ptr
```

## 4.7 Src/system\_stm32f3xx.c File Reference

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

```
#include "stm32f3xx.h"
```

Include dependency graph for system\_stm32f3xx.c:



### Macros

- #define `HSE_VALUE` ((uint32\_t)8000000)
- #define `HSI_VALUE` ((uint32\_t)8000000)
- #define `VECT_TAB_OFFSET` 0x0

### Functions

- void `SystemInit` (void)  
*Setup the microcontroller system Initialize the FPU setting, vector table location and the PLL configuration is reset.*
- void `SystemCoreClockUpdate` (void)  
*Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.*

## Variables

- uint32\_t [SystemCoreClock](#) = 8000000
- const uint8\_t [AHBPrescTable](#) [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
- const uint8\_t [APBPrescTable](#) [8] = {0, 0, 0, 0, 1, 2, 3, 4}

### 4.7.1 Detailed Description

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

#### Author

MCD Application Team

1. 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\_stm32f3xx.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.
2. After each device reset the HSI (8 MHz) is used as system clock source. Then [SystemInit\(\)](#) function is called, in "startup\_stm32f3xx.s" file, to configure the system clock before to branch to main program.

### 3. This file configures the system clock as follows:

Supported STM32F3xx device

System Clock source | HSI

SYSCLK(Hz) | 8000000

HCLK(Hz) | 8000000

AHB Prescaler | 1

APB2 Prescaler | 1

APB1 Prescaler | 1

USB Clock | DISABLE

=====

Attention



© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

