

# VOLTMETER

Grzegorz Grund

Generated by Doxygen 1.9.6



<b>1 Module Index</b>	<b>1</b>
1.1 Modules	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Module Documentation</b>	<b>5</b>
3.1 CMSIS	5
3.1.1 Detailed Description	6
3.2 Stm32f4xx_system	6
3.2.1 Detailed Description	7
3.3 STM32F4xx_System_Private_Includes	7
3.3.1 Detailed Description	7
3.3.2 Macro Definition Documentation	7
3.3.2.1 HSE_VALUE	7
3.3.2.2 HSI_VALUE	7
3.4 STM32F4xx_System_Private_TypesDefinitions	8
3.5 STM32F4xx_System_Private_Defines	8
3.6 STM32F4xx_System_Private_Macros	8
3.7 STM32F4xx_System_Private_Variables	9
3.7.1 Detailed Description	9
3.7.2 Variable Documentation	9
3.7.2.1 AHBPrescTable	9
3.7.2.2 APBPrescTable	9
3.7.2.3 SystemCoreClock	9
3.8 STM32F4xx_System_Private_FunctionPrototypes	10
3.9 STM32F4xx_System_Private_Functions	10
3.9.1 Detailed Description	10
3.9.2 Function Documentation	10
3.9.2.1 SystemCoreClockUpdate()	11
3.9.2.2 SystemInit()	11
<b>4 File Documentation</b>	<b>13</b>
4.1 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/adc.h File Reference	13
4.1.1 Detailed Description	14
4.1.2 Function Documentation	14
4.1.2.1 MX_ADC1_Init()	14
4.1.3 Variable Documentation	15
4.1.3.1 hadc1	15
4.2 adc.h	15
4.3 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/dma.h File Reference	15
4.3.1 Detailed Description	16
4.3.2 Function Documentation	17

4.3.2.1 MX_DMA_Init()	17
4.4 dma.h	17
4.5 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/FreeRTOSConfig.h File Reference	18
4.5.1 Macro Definition Documentation	18
4.5.1.1 configASSERT	19
4.5.1.2 configCPU_CLOCK_HZ	19
4.5.1.3 configENABLE_FPU	19
4.5.1.4 configENABLE_MPU	19
4.5.1.5 configKERNEL_INTERRUPT_PRIORITY	19
4.5.1.6 configLIBRARY_LOWEST_INTERRUPT_PRIORITY	19
4.5.1.7 configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY	19
4.5.1.8 configMAX_CO_ROUTINE_PRIORITIES	20
4.5.1.9 configMAX_PRIORITIES	20
4.5.1.10 configMAX_SYSCALL_INTERRUPT_PRIORITY	20
4.5.1.11 configMAX_TASK_NAME_LEN	20
4.5.1.12 configMESSAGE_BUFFER_LENGTH_TYPE	20
4.5.1.13 configMINIMAL_STACK_SIZE	20
4.5.1.14 configPRIO_BITS	20
4.5.1.15 configQUEUE_REGISTRY_SIZE	21
4.5.1.16 configSUPPORT_DYNAMIC_ALLOCATION	21
4.5.1.17 configSUPPORT_STATIC_ALLOCATION	21
4.5.1.18 configTICK_RATE_HZ	21
4.5.1.19 configTIMER_QUEUE_LENGTH	21
4.5.1.20 configTIMER_TASK_PRIORITY	21
4.5.1.21 configTIMER_TASK_STACK_DEPTH	21
4.5.1.22 configTOTAL_HEAP_SIZE	21
4.5.1.23 configUSE_16_BIT_TICKS	22
4.5.1.24 configUSE_CO_ROUTINES	22
4.5.1.25 configUSE_IDLE_HOOK	22
4.5.1.26 configUSE_MUTEXES	22
4.5.1.27 configUSE_PORT_OPTIMISED_TASK_SELECTION	22
4.5.1.28 configUSE_PREEMPTION	22
4.5.1.29 configUSE_TICK_HOOK	22
4.5.1.30 configUSE_TIMERS	22
4.5.1.31 INCLUDE_uxTaskPriorityGet	23
4.5.1.32 INCLUDE_vTaskCleanUpResources	23
4.5.1.33 INCLUDE_vTaskDelay	23
4.5.1.34 INCLUDE_vTaskDelayUntil	23
4.5.1.35 INCLUDE_vTaskDelete	23
4.5.1.36 INCLUDE_vTaskPrioritySet	23
4.5.1.37 INCLUDE_vTaskSuspend	23
4.5.1.38 INCLUDE_xTaskGetSchedulerState	23

4.5.1.39 vPortSVCHandler . . . . .	24
4.5.1.40 xPortPendSVHandler . . . . .	24
4.5.1.41 xPortSysTickHandler . . . . .	24
4.6 FreeRTOSConfig.h . . . . .	24
4.7 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/gpio.h File Reference . . . . .	26
4.7.1 Detailed Description . . . . .	27
4.7.2 Function Documentation . . . . .	27
4.7.2.1 MX_GPIO_Init() . . . . .	27
4.8 gpio.h . . . . .	28
4.9 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/main.h File Reference . . . . .	28
4.9.1 Detailed Description . . . . .	29
4.9.2 Function Documentation . . . . .	29
4.9.2.1 Error_Handler() . . . . .	29
4.10 main.h . . . . .	30
4.11 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/stm32f4xx_hal_conf.h File Reference . . . . .	31
4.11.1 Macro Definition Documentation . . . . .	33
4.11.1.1 assert_param . . . . .	33
4.11.1.2 DATA_CACHE_ENABLE . . . . .	34
4.11.1.3 DP83848_PHY_ADDRESS . . . . .	34
4.11.1.4 ETH_RX_BUF_SIZE . . . . .	34
4.11.1.5 ETH_RXBUFNB . . . . .	34
4.11.1.6 ETH_TX_BUF_SIZE . . . . .	34
4.11.1.7 ETH_TXBUFNB . . . . .	34
4.11.1.8 EXTERNAL_CLOCK_VALUE . . . . .	34
4.11.1.9 HAL_ADC_MODULE_ENABLED . . . . .	35
4.11.1.10 HAL_CORTEX_MODULE_ENABLED . . . . .	35
4.11.1.11 HAL_DMA_MODULE_ENABLED . . . . .	35
4.11.1.12 HAL_EXTI_MODULE_ENABLED . . . . .	35
4.11.1.13 HAL_FLASH_MODULE_ENABLED . . . . .	35
4.11.1.14 HAL_GPIO_MODULE_ENABLED . . . . .	35
4.11.1.15 HAL_MODULE_ENABLED . . . . .	35
4.11.1.16 HAL_PWR_MODULE_ENABLED . . . . .	36
4.11.1.17 HAL_RCC_MODULE_ENABLED . . . . .	36
4.11.1.18 HAL_TIM_MODULE_ENABLED . . . . .	36
4.11.1.19 HAL_UART_MODULE_ENABLED . . . . .	36
4.11.1.20 HSE_STARTUP_TIMEOUT . . . . .	36
4.11.1.21 HSE_VALUE . . . . .	36
4.11.1.22 HSI_VALUE . . . . .	36
4.11.1.23 INSTRUCTION_CACHE_ENABLE . . . . .	37
4.11.1.24 LSE_STARTUP_TIMEOUT . . . . .	37
4.11.1.25 LSE_VALUE . . . . .	37

4.11.1.26 LSI_VALUE . . . . .	37
4.11.1.27 MAC_ADDR0 . . . . .	37
4.11.1.28 MAC_ADDR1 . . . . .	37
4.11.1.29 MAC_ADDR2 . . . . .	37
4.11.1.30 MAC_ADDR3 . . . . .	38
4.11.1.31 MAC_ADDR4 . . . . .	38
4.11.1.32 MAC_ADDR5 . . . . .	38
4.11.1.33 PHY_AUTONEGO_COMPLETE . . . . .	38
4.11.1.34 PHY_AUTONEGOTIATION . . . . .	38
4.11.1.35 PHY_BCR . . . . .	38
4.11.1.36 PHY_BSR . . . . .	38
4.11.1.37 PHY_CONFIG_DELAY . . . . .	39
4.11.1.38 PHY_DUPLEX_STATUS . . . . .	39
4.11.1.39 PHY_FULLDUPLEX_100M . . . . .	39
4.11.1.40 PHY_FULLDUPLEX_10M . . . . .	39
4.11.1.41 PHY_HALFDUPLEX_100M . . . . .	39
4.11.1.42 PHY_HALFDUPLEX_10M . . . . .	39
4.11.1.43 PHY_ISOLATE . . . . .	39
4.11.1.44 PHY_JABBER_DETECTION . . . . .	40
4.11.1.45 PHY_LINKED_STATUS . . . . .	40
4.11.1.46 PHY_LOOPBACK . . . . .	40
4.11.1.47 PHY_POWERDOWN . . . . .	40
4.11.1.48 PHY_READ_TO . . . . .	40
4.11.1.49 PHY_RESET . . . . .	40
4.11.1.50 PHY_RESET_DELAY . . . . .	40
4.11.1.51 PHY_RESTART_AUTONEGOTIATION . . . . .	41
4.11.1.52 PHY_SPEED_STATUS . . . . .	41
4.11.1.53 PHY_SR . . . . .	41
4.11.1.54 PHY_WRITE_TO . . . . .	41
4.11.1.55 PREFETCH_ENABLE . . . . .	41
4.11.1.56 TICK_INT_PRIORITY . . . . .	41
4.11.1.57 USE_HAL_ADC_REGISTER_CALLBACKS . . . . .	41
4.11.1.58 USE_HAL_CAN_REGISTER_CALLBACKS . . . . .	42
4.11.1.59 USE_HAL_CEC_REGISTER_CALLBACKS . . . . .	42
4.11.1.60 USE_HAL_Cryp_REGISTER_CALLBACKS . . . . .	42
4.11.1.61 USE_HAL_DAC_REGISTER_CALLBACKS . . . . .	42
4.11.1.62 USE_HAL_DCMI_REGISTER_CALLBACKS . . . . .	42
4.11.1.63 USE_HAL_DFSDM_REGISTER_CALLBACKS . . . . .	42
4.11.1.64 USE_HAL_DMA2D_REGISTER_CALLBACKS . . . . .	42
4.11.1.65 USE_HAL_DSI_REGISTER_CALLBACKS . . . . .	42
4.11.1.66 USE_HAL_ETH_REGISTER_CALLBACKS . . . . .	43
4.11.1.67 USE_HAL_FMPI2C_REGISTER_CALLBACKS . . . . .	43

4.11.1.68 USE_HAL_FMPMBUS_REGISTER_CALLBACKS . . . . .	43
4.11.1.69 USE_HAL_HASH_REGISTER_CALLBACKS . . . . .	43
4.11.1.70 USE_HAL_HCD_REGISTER_CALLBACKS . . . . .	43
4.11.1.71 USE_HAL_I2C_REGISTER_CALLBACKS . . . . .	43
4.11.1.72 USE_HAL_I2S_REGISTER_CALLBACKS . . . . .	43
4.11.1.73 USE_HAL_IRDA_REGISTER_CALLBACKS . . . . .	43
4.11.1.74 USE_HAL_LPTIM_REGISTER_CALLBACKS . . . . .	44
4.11.1.75 USE_HAL_LTDC_REGISTER_CALLBACKS . . . . .	44
4.11.1.76 USE_HAL_MMC_REGISTER_CALLBACKS . . . . .	44
4.11.1.77 USE_HAL_NAND_REGISTER_CALLBACKS . . . . .	44
4.11.1.78 USE_HAL_NOR_REGISTER_CALLBACKS . . . . .	44
4.11.1.79 USE_HAL_PCCARD_REGISTER_CALLBACKS . . . . .	44
4.11.1.80 USE_HAL_PCD_REGISTER_CALLBACKS . . . . .	44
4.11.1.81 USE_HAL_QSPI_REGISTER_CALLBACKS . . . . .	44
4.11.1.82 USE_HAL_RNG_REGISTER_CALLBACKS . . . . .	45
4.11.1.83 USE_HAL_RTC_REGISTER_CALLBACKS . . . . .	45
4.11.1.84 USE_HAL_SAI_REGISTER_CALLBACKS . . . . .	45
4.11.1.85 USE_HAL_SD_REGISTER_CALLBACKS . . . . .	45
4.11.1.86 USE_HAL_SDRAM_REGISTER_CALLBACKS . . . . .	45
4.11.1.87 USE_HAL_SMARTCARD_REGISTER_CALLBACKS . . . . .	45
4.11.1.88 USE_HAL_SMBUS_REGISTER_CALLBACKS . . . . .	45
4.11.1.89 USE_HAL_SPDIFRX_REGISTER_CALLBACKS . . . . .	45
4.11.1.90 USE_HAL_SPI_REGISTER_CALLBACKS . . . . .	46
4.11.1.91 USE_HAL_SRAM_REGISTER_CALLBACKS . . . . .	46
4.11.1.92 USE_HAL_TIM_REGISTER_CALLBACKS . . . . .	46
4.11.1.93 USE_HAL_UART_REGISTER_CALLBACKS . . . . .	46
4.11.1.94 USE_HAL_USART_REGISTER_CALLBACKS . . . . .	46
4.11.1.95 USE_HAL_WWDG_REGISTER_CALLBACKS . . . . .	46
4.11.1.96 USE_RTOS . . . . .	46
4.11.1.97 USE_SPI_CRC . . . . .	46
4.11.1.98 VDD_VALUE . . . . .	47
4.12 stm32f4xx_hal_conf.h . . . . .	47
4.13 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/stm32f4xx_it.h File Reference . .	52
4.13.1 Detailed Description . . . . .	53
4.13.2 Function Documentation . . . . .	53
4.13.2.1 ADC_IRQHandler() . . . . .	53
4.13.2.2 BusFault_Handler() . . . . .	53
4.13.2.3 DebugMon_Handler() . . . . .	53
4.13.2.4 DMA2_Stream2_IRQHandler() . . . . .	54
4.13.2.5 DMA2_Stream7_IRQHandler() . . . . .	54
4.13.2.6 HardFault_Handler() . . . . .	54
4.13.2.7 MemManage_Handler() . . . . .	54

4.13.2.8 NMI_Handler()	54
4.13.2.9 TIM1_TRG_COM_TIM11_IRQHandler()	54
4.13.2.10 TIM1_UP_TIM10_IRQHandler()	55
4.13.2.11 UsageFault_Handler()	55
4.13.2.12 USART1_IRQHandler()	55
4.14 stm32f4xx_it.h	55
4.15 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/tim.h File Reference	56
4.15.1 Detailed Description	57
4.15.2 Function Documentation	57
4.15.2.1 MX_TIM11_Init()	57
4.15.2.2 MX_TIM1_Init()	58
4.15.3 Variable Documentation	58
4.15.3.1 htim1	58
4.15.3.2 htim11	58
4.16 tim.h	59
4.17 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/usart.h File Reference	59
4.17.1 Detailed Description	60
4.17.2 Function Documentation	61
4.17.2.1 MX_USART1_UART_Init()	61
4.17.3 Variable Documentation	61
4.17.3.1 huart1	61
4.18 usart.h	62
4.19 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/adc.c File Reference	62
4.19.1 Detailed Description	63
4.19.2 Function Documentation	64
4.19.2.1 HAL_ADC_MspDeInit()	64
4.19.2.2 HAL_ADC_MspInit()	64
4.19.2.3 MX_ADC1_Init()	64
4.19.3 Variable Documentation	65
4.19.3.1 hadc1	65
4.20 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/dma.c File Reference	65
4.20.1 Detailed Description	66
4.20.2 Function Documentation	66
4.20.2.1 MX_DMA_Init()	66
4.21 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/freertos.c File Reference	66
4.21.1 Macro Definition Documentation	68
4.21.1.1 MainBuf_SIZE	68
4.21.1.2 RxBuf_SIZE	68
4.21.2 Function Documentation	68
4.21.2.1 ADC_CONFIG_TASK_INIT()	68
4.21.2.2 ADC_TAKE_TASK_INIT()	69
4.21.2.3 HAL_TIM_PeriodicElapsedCallback()	70



4.21.2.4 HAL_UARTEx_RxEventCallback()	70
4.21.2.5 IDLE_DEBUG_TASK_INIT()	70
4.21.2.6 INT_TIMER_TASK_INIT()	70
4.21.2.7 INT_UART_TASK_INIT()	71
4.21.2.8 MX_FREERTOS_Init()	72
4.21.2.9 TIMER_CALLBACK()	73
4.21.2.10 UART_RX_TASK_INIT()	73
4.21.2.11 UART_TX_TASK_INIT()	74
4.21.2.12 vApplicationGetIdleTaskMemory()	74
4.21.2.13 vApplicationGetTimerTaskMemory()	75
4.21.2.14 vApplicationIdleHook()	75
4.21.2.15 vApplicationTickHook()	75
4.21.3 Variable Documentation	75
4.21.3.1 ADC_CONFIG_TASKHandle	75
4.21.3.2 ADC_QUEUE_UART_TXHandle	75
4.21.3.3 ADC_TAKE_TASKHandle	75
4.21.3.4 IDLE_DEBUG_TASKHandle	76
4.21.3.5 INT_TIMER_TASKHandle	76
4.21.3.6 INT_UART_QUEUE_UART_RXHandle	76
4.21.3.7 INT_UART_TASKHandle	76
4.21.3.8 MainBuf	76
4.21.3.9 newPos	76
4.21.3.10 oldPos	76
4.21.3.11 RTOS_TIMER_1Handle	76
4.21.3.12 RxBuf	77
4.21.3.13 UART_RX_QUEUE_ADC_CONFIGHandle	77
4.21.3.14 UART_RX_QUEUE_ADC_TAKHandle	77
4.21.3.15 UART_RX_QUEUE_UART_TXHandle	77
4.21.3.16 UART_RX_TASKHandle	77
4.21.3.17 UART_TX_TASKHandle	77
4.22 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/gpio.c File Reference	78
4.22.1 Detailed Description	78
4.22.2 Function Documentation	79
4.22.2.1 MX_GPIO_Init()	79
4.23 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/main.c File Reference	79
4.23.1 Detailed Description	80
4.23.2 Function Documentation	80
4.23.2.1 Error_Handler()	80
4.23.2.2 HAL_TIM_PeriodElapsedCallback()	81
4.23.2.3 main()	81
4.23.2.4 MX_FREERTOS_Init()	82
4.23.2.5 SystemClock_Config()	84

4.24 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/stm32f4xx_hal_msp.c File Reference . . . . .	84
4.24.1 Detailed Description . . . . .	85
4.24.2 Function Documentation . . . . .	85
4.24.2.1 HAL_MspInit() . . . . .	85
4.25 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/stm32f4xx_hal_timebase_tim.c File Reference . . . . .	86
4.25.1 Detailed Description . . . . .	86
4.25.2 Function Documentation . . . . .	87
4.25.2.1 HAL_InitTick() . . . . .	87
4.25.2.2 HAL_ResumeTick() . . . . .	87
4.25.2.3 HAL_SuspendTick() . . . . .	88
4.25.3 Variable Documentation . . . . .	88
4.25.3.1 htim10 . . . . .	88
4.26 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/stm32f4xx_it.c File Reference . . . . .	88
4.26.1 Detailed Description . . . . .	90
4.26.2 Function Documentation . . . . .	90
4.26.2.1 ADC_IRQHandler() . . . . .	90
4.26.2.2 BusFault_Handler() . . . . .	90
4.26.2.3 DebugMon_Handler() . . . . .	91
4.26.2.4 DMA2_Stream2_IRQHandler() . . . . .	91
4.26.2.5 DMA2_Stream7_IRQHandler() . . . . .	91
4.26.2.6 HardFault_Handler() . . . . .	91
4.26.2.7 MemManage_Handler() . . . . .	91
4.26.2.8 NMI_Handler() . . . . .	91
4.26.2.9 TIM1_TRG_COM_TIM11_IRQHandler() . . . . .	92
4.26.2.10 TIM1_UP_TIM10_IRQHandler() . . . . .	92
4.26.2.11 UsageFault_Handler() . . . . .	92
4.26.2.12 USART1_IRQHandler() . . . . .	92
4.26.3 Variable Documentation . . . . .	92
4.26.3.1 hadc1 . . . . .	92
4.26.3.2 hdma_usart1_rx . . . . .	92
4.26.3.3 hdma_usart1_tx . . . . .	93
4.26.3.4 htim1 . . . . .	93
4.26.3.5 htim10 . . . . .	93
4.26.3.6 htim11 . . . . .	93
4.26.3.7 huart1 . . . . .	93
4.27 C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/syscalls.c File Reference . . . . .	93
4.27.1 Detailed Description . . . . .	94
4.27.2 Function Documentation . . . . .	94
4.27.2.1 __attribute__() . . . . .	95
4.27.2.2 __io_getchar() . . . . .	95
4.27.2.3 __io_putchar() . . . . .	95

4.27.2.4	<a href="#">_close()</a>	95
4.27.2.5	<a href="#">_execve()</a>	96
4.27.2.6	<a href="#">_exit()</a>	96
4.27.2.7	<a href="#">_fork()</a>	96
4.27.2.8	<a href="#">_fstat()</a>	96
4.27.2.9	<a href="#">_getpid()</a>	96
4.27.2.10	<a href="#">_isatty()</a>	97
4.27.2.11	<a href="#">_kill()</a>	97
4.27.2.12	<a href="#">_link()</a>	97
4.27.2.13	<a href="#">_lseek()</a>	97
4.27.2.14	<a href="#">_open()</a>	97
4.27.2.15	<a href="#">_stat()</a>	98
4.27.2.16	<a href="#">_times()</a>	98
4.27.2.17	<a href="#">_unlink()</a>	98
4.27.2.18	<a href="#">_wait()</a>	98
4.27.2.19	<a href="#">initialise_monitor_handles()</a>	98
4.27.3	Variable Documentation	98
4.27.3.1	<a href="#">environ</a>	98
4.28	<a href="#">C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/systemem.c File Reference</a>	99
4.28.1	Detailed Description	99
4.28.2	Function Documentation	100
4.28.2.1	<a href="#">_sbrk()</a>	100
4.29	<a href="#">C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/system_stm32f4xx.c File Reference</a>	100
4.29.1	Detailed Description	101
4.30	<a href="#">C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/tim.c File Reference</a>	102
4.30.1	Detailed Description	102
4.30.2	Function Documentation	103
4.30.2.1	<a href="#">HAL_TIM_Base_MspDeInit()</a>	103
4.30.2.2	<a href="#">HAL_TIM_Base_MspInit()</a>	103
4.30.2.3	<a href="#">MX_TIM11_Init()</a>	103
4.30.2.4	<a href="#">MX_TIM1_Init()</a>	104
4.30.3	Variable Documentation	104
4.30.3.1	<a href="#">htim1</a>	104
4.30.3.2	<a href="#">htim11</a>	104
4.31	<a href="#">C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/usart.c File Reference</a>	105
4.31.1	Detailed Description	105
4.31.2	Function Documentation	106
4.31.2.1	<a href="#">HAL_UART_MspDeInit()</a>	106
4.31.2.2	<a href="#">HAL_UART_MspInit()</a>	106
4.31.2.3	<a href="#">MX_USART1_UART_Init()</a>	106
4.31.3	Variable Documentation	107

4.31.3.1 <a href="#">hdma_usart1_rx</a> . . . . .	107
4.31.3.2 <a href="#">hdma_usart1_tx</a> . . . . .	107
4.31.3.3 <a href="#">huart1</a> . . . . .	107

<b><a href="#">Index</a></b>	<b>109</b>
------------------------------	------------

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

CMSIS . . . . .	5
Stm32f4xx_system . . . . .	6
STM32F4xx_System_Private_Includes . . . . .	7
STM32F4xx_System_Private_TypesDefinitions . . . . .	8
STM32F4xx_System_Private_Defines . . . . .	8
STM32F4xx_System_Private_Macros . . . . .	8
STM32F4xx_System_Private_Variables . . . . .	9
STM32F4xx_System_Private_FunctionPrototypes . . . . .	10
STM32F4xx_System_Private_Functions . . . . .	10



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">adc.h</a>	
This file contains all the function prototypes for the <a href="#">adc.c</a> file	13
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">dma.h</a>	
This file contains all the function prototypes for the <a href="#">dma.c</a> file	15
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">FreeRTOSConfig.h</a>	18
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">gpio.h</a>	
This file contains all the function prototypes for the <a href="#">gpio.c</a> file	26
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">main.h</a>	
: Header for <a href="#">main.c</a> file	28
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">stm32f4xx_hal_conf.h</a>	31
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">stm32f4xx_it.h</a>	
This file contains the headers of the interrupt handlers	52
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">tim.h</a>	
This file contains all the function prototypes for the <a href="#">tim.c</a> file	56
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/ <a href="#">usart.h</a>	
This file contains all the function prototypes for the <a href="#">usart.c</a> file	59
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">adc.c</a>	
This file provides code for the configuration of the ADC instances	62
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">dma.c</a>	
This file provides code for the configuration of all the requested memory to memory DMA transfers	65
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">freertos.c</a>	66
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">gpio.c</a>	
This file provides code for the configuration of all used GPIO pins	78
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">main.c</a>	
: Main program body	79
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">stm32f4xx_hal_msp.c</a>	
This file provides code for the MSP Initialization and de-Initialization codes	84
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">stm32f4xx_hal_timebase_tim.c</a>	
HAL time base based on the hardware TIM	86
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">stm32f4xx_it.c</a>	
Interrupt Service Routines	88
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">syscalls.c</a>	
STM32CubeIDE Minimal System calls file	93
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">sysmem.c</a>	
STM32CubeIDE System Memory calls file	99

C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">system_stm32f4xx.c</a>	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File . . . . .	100
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">tim.c</a>	
This file provides code for the configuration of the TIM instances . . . . .	102
C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/ <a href="#">usart.c</a>	
This file provides code for the configuration of the USART instances . . . . .	105

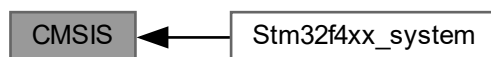


## Chapter 3

# Module Documentation

### 3.1 CMSIS

Collaboration diagram for CMSIS:



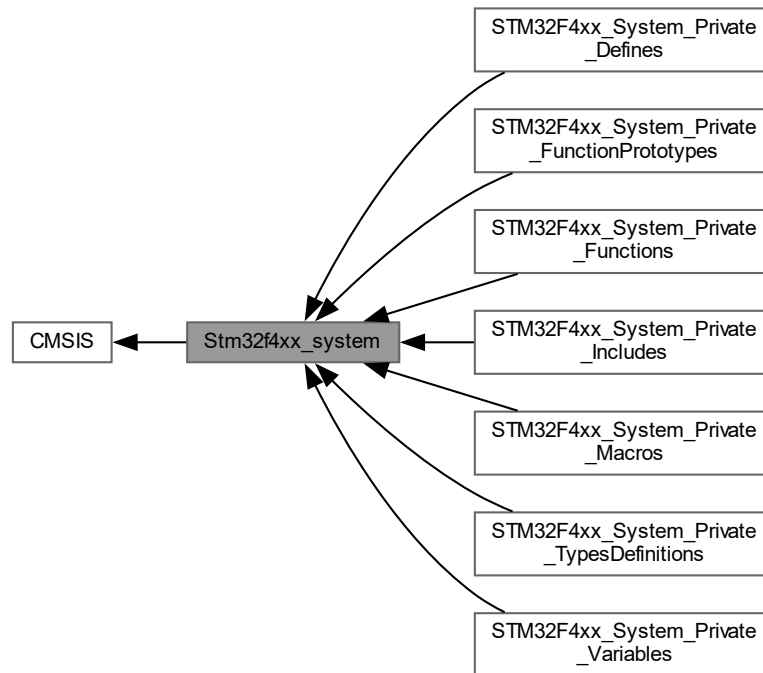
#### Modules

- [Stm32f4xx\\_system](#)

### 3.1.1 Detailed Description

## 3.2 Stm32f4xx\_system

Collaboration diagram for Stm32f4xx\_system:



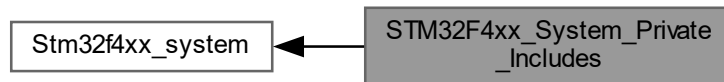
## Modules

- [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

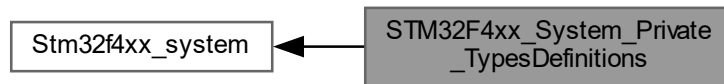
#### 3.3.2.2 HSI\_VALUE

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

Value of the Internal oscillator in Hz

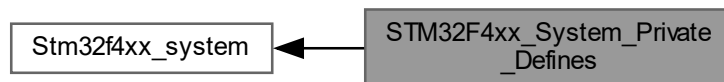
### 3.4 STM32F4xx\_System\_Private\_TypeDefinitions

Collaboration diagram for STM32F4xx\_System\_Private\_TypeDefinitions:



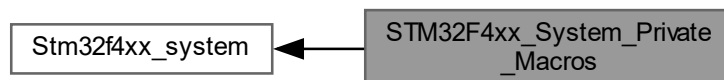
### 3.5 STM32F4xx\_System\_Private\_Defines

Collaboration diagram for STM32F4xx\_System\_Private\_Defines:



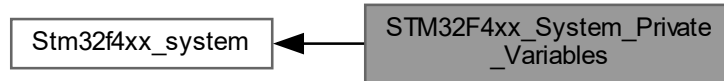
### 3.6 STM32F4xx\_System\_Private\_Macros

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

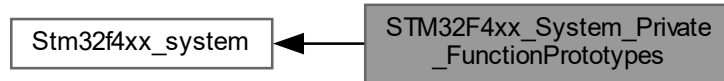
```
const uint8_t APBPrescTable[8] = {0, 0, 0, 0, 1, 2, 3, 4}
```

##### 3.7.2.3 SystemCoreClock

```
uint32_t SystemCoreClock = 16000000
```

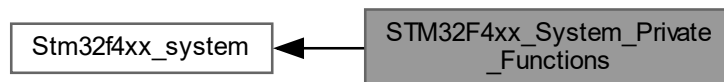
### 3.8 STM32F4xx\_System\_Private\_FunctionPrototypes

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**

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

**Return values**

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



## Chapter 4

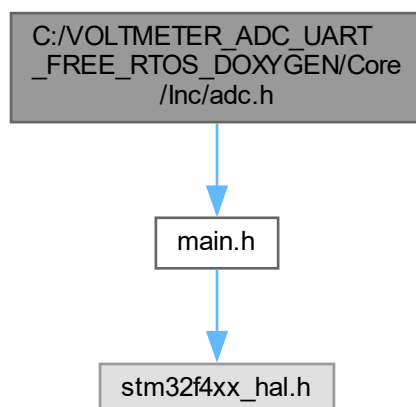
# File Documentation

### 4.1 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/Inc/adc.h File Reference

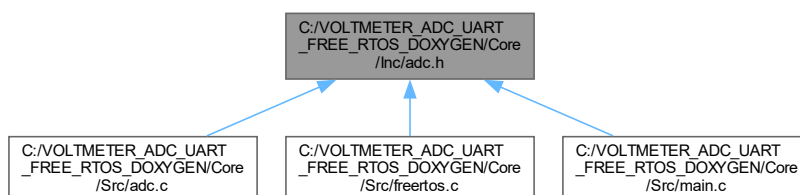
This file contains all the function prototypes for the [adc.c](#) file.

```
#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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.1.2 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

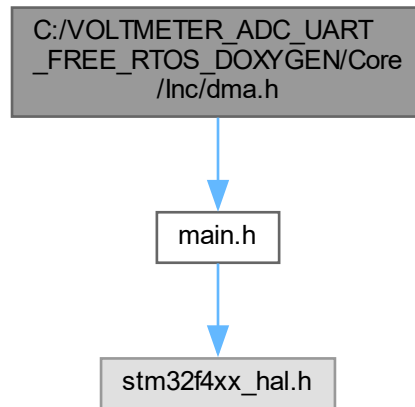
```

## 4.3 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/↵ Inc/dma.h File Reference

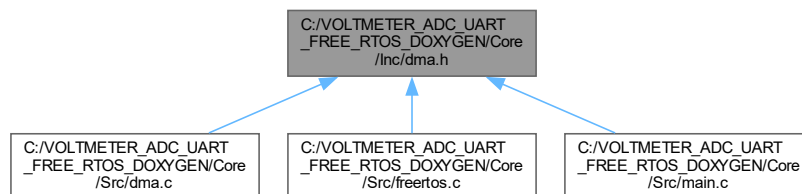
This file contains all the function prototypes for the [dma.c](#) file.

```
#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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.3.2 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     /* __NVIC_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 */

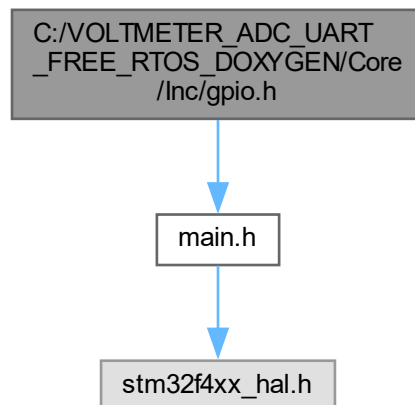
```

## 4.7 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/↵ Inc/gpio.h File Reference

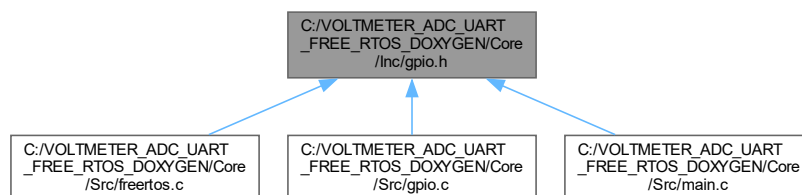
This file contains all the function prototypes for the [gpio.c](#) file.

```
#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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.7.2 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

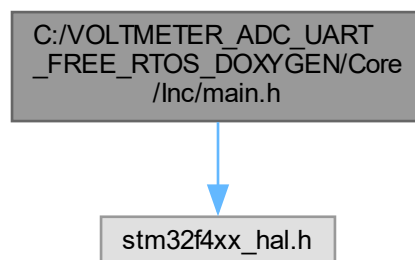
```

## 4.9 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/Inc/main.h File Reference

: Header for [main.c](#) file.

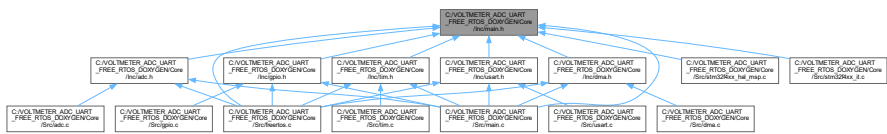
```
#include "stm32f4xx_hal.h"
```

Include dependency graph for main.h:





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



- 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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

4.9.2 Function Documentation

4.9.2.1 Error\_Handler()

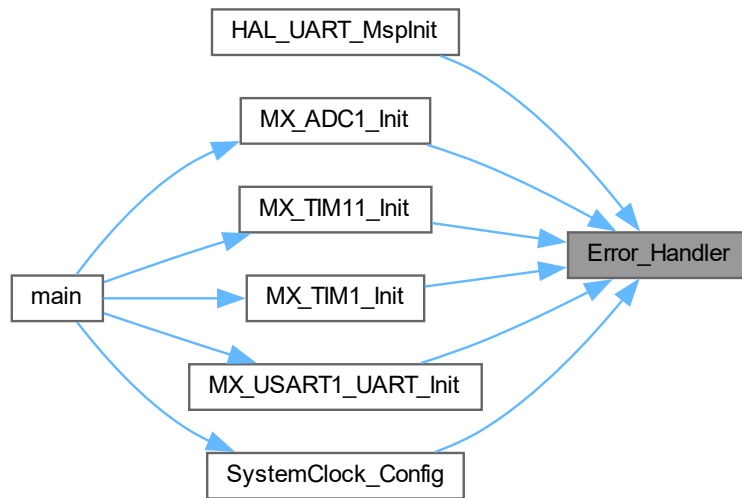
```
void Error_Handler (  
    void )
```

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 */

```

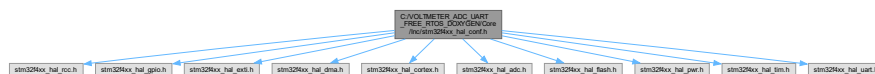
## 4.11 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/Inc/stm32f4xx\_hal\_conf.h File Reference

```

#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_FMPMBUS_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\_CRYPT\_REGISTER\_CALLBACKS

```
#define USE_HAL_CRYPT_REGISTER_CALLBACKS 0U /* CRYPT 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\_FMPMBUS\_REGISTER\_CALLBACKS**

```
#define USE_HAL_FMPMBUS_REGISTER_CALLBACKS 0U /* FMPMBUS 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_FMPSPMBUS_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 ##### */
00092 #if !defined (HSE_VALUE)
00093     #define HSE_VALUE 25000000U
00094 #endif /* HSE_VALUE */
00095
00096 #if !defined (HSE_STARTUP_TIMEOUT)
00097     #define HSE_STARTUP_TIMEOUT 100U
00098 #endif /* HSE_STARTUP_TIMEOUT */
00099
00100 #if !defined (HSI_VALUE)
00101     #define HSI_VALUE ((uint32_t)16000000U)
00102 #endif /* HSI_VALUE */
00103
00104 #if !defined (LSI_VALUE)
00105     #define LSI_VALUE 32000U
00106 #endif /* LSI_VALUE */
00107
00108 #if !defined (LSE_VALUE)
00109     #define LSE_VALUE 32768U
00110 #endif /* LSE_VALUE */
00111
00112 #if !defined (LSE_STARTUP_TIMEOUT)
00113     #define LSE_STARTUP_TIMEOUT 5000U
00114 #endif /* LSE_STARTUP_TIMEOUT */
00115
00116 #if !defined (EXTERNAL_CLOCK_VALUE)
00117     #define EXTERNAL_CLOCK_VALUE 12288000U
00118 #endif /* EXTERNAL_CLOCK_VALUE */
00119
00120 /* Tip: To avoid modifying this file each time you need to use different HSE,
00121    == you can define the HSE value in your toolchain compiler preprocessor. */
00122
00123 /* ##### System Configuration ##### */
00124 #define VDD_VALUE 3300U
00125 #define TICK_INT_PRIORITY 5U
00126 #define USE_RTOS 0U
00127 #define PREFETCH_ENABLE 1U
00128 #define INSTRUCTION_CACHE_ENABLE 1U
00129 #define DATA_CACHE_ENABLE 1U
00130
00131 #define USE_HAL_ADC_REGISTER_CALLBACKS 0U /* ADC register callback disabled */
00132 #define USE_HAL_CAN_REGISTER_CALLBACKS 0U /* CAN register callback disabled */
00133 #define USE_HAL_CEC_REGISTER_CALLBACKS 0U /* CEC register callback disabled */
00134 #define USE_HAL_CRYP_REGISTER_CALLBACKS 0U /* CRYP register callback disabled */
00135 #define USE_HAL_DAC_REGISTER_CALLBACKS 0U /* DAC register callback disabled */
00136 #define USE_HAL_DCMI_REGISTER_CALLBACKS 0U /* DCMI register callback disabled */
00137 #define USE_HAL_DFSDM_REGISTER_CALLBACKS 0U /* DFSDM register callback disabled */
00138 #define USE_HAL_DMA2D_REGISTER_CALLBACKS 0U /* DMA2D register callback disabled */
00139 #define USE_HAL_DSI_REGISTER_CALLBACKS 0U /* DSI register callback disabled */
00140 #define USE_HAL_ETH_REGISTER_CALLBACKS 0U /* ETH register callback disabled */
00141 #define USE_HAL_HASH_REGISTER_CALLBACKS 0U /* HASH register callback disabled */
00142 #define USE_HAL_HCD_REGISTER_CALLBACKS 0U /* HCD register callback disabled */
00143 #define USE_HAL_I2C_REGISTER_CALLBACKS 0U /* I2C register callback disabled */
00144 #define USE_HAL_FMPI2C_REGISTER_CALLBACKS 0U /* FMPI2C register callback disabled */
00145 #define USE_HAL_FMPUSB_REGISTER_CALLBACKS 0U /* FMPUSB register callback disabled */
00146 #define USE_HAL_I2S_REGISTER_CALLBACKS 0U /* I2S register callback disabled */
00147 #define USE_HAL_IRDA_REGISTER_CALLBACKS 0U /* IRDA register callback disabled */
00148 #define USE_HAL_LPTIM_REGISTER_CALLBACKS 0U /* LPTIM register callback disabled */
00149 #define USE_HAL_LTDC_REGISTER_CALLBACKS 0U /* LTDC register callback disabled */
00150 #define USE_HAL_MMC_REGISTER_CALLBACKS 0U /* MMC register callback disabled */
00151 #define USE_HAL_NAND_REGISTER_CALLBACKS 0U /* NAND register callback disabled */
00152 #define USE_HAL_NOR_REGISTER_CALLBACKS 0U /* NOR register callback disabled */
00153 #define USE_HAL_PCCARD_REGISTER_CALLBACKS 0U /* PCCARD register callback disabled */
00154 #define USE_HAL_PCD_REGISTER_CALLBACKS 0U /* PCD register callback disabled */
00155 #define USE_HAL_QSPI_REGISTER_CALLBACKS 0U /* QSPI register callback disabled */
00156 #define USE_HAL_RNG_REGISTER_CALLBACKS 0U /* RNG register callback disabled */
00157 #define USE_HAL_RTC_REGISTER_CALLBACKS 0U /* RTC register callback disabled */
00158 #define USE_HAL_SAI_REGISTER_CALLBACKS 0U /* SAI register callback disabled */
00159 #define USE_HAL_SD_REGISTER_CALLBACKS 0U /* SD register callback disabled */
00160 #define USE_HAL_SMARTCARD_REGISTER_CALLBACKS 0U /* SMARTCARD register callback disabled */
00161 #define USE_HAL_SDRAM_REGISTER_CALLBACKS 0U /* SDRAM register callback disabled */
00162 #define USE_HAL_SRAM_REGISTER_CALLBACKS 0U /* SRAM register callback disabled */
00163 #define USE_HAL_SPDIFRX_REGISTER_CALLBACKS 0U /* SPDIFRX register callback disabled */
00164 #define USE_HAL_SMBUS_REGISTER_CALLBACKS 0U /* SMBUS register callback disabled */
00165 #define USE_HAL_SPI_REGISTER_CALLBACKS 0U /* SPI register callback disabled */
00166 #define USE_HAL_TIM_REGISTER_CALLBACKS 0U /* TIM register callback disabled */
00167 #define USE_HAL_UART_REGISTER_CALLBACKS 0U /* UART register callback disabled */
00168 #define USE_HAL_USART_REGISTER_CALLBACKS 0U /* USART register callback disabled */
00169 #define USE_HAL_WWDG_REGISTER_CALLBACKS 0U /* WWDG register callback disabled */
00170
00171 /* ##### Assert Selection ##### */
00172 #define USE_FULL_ASSERT 1U
00173
00174 /* ##### Ethernet peripheral configuration ##### */
00175
00176 /* 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)
00237 #define PHY_RESET                     ((uint16_t)0x8000U)
00238 #define PHY_LOOPBACK                  ((uint16_t)0x4000U)
00239 #define PHY_FULLDUPLEX_100M           ((uint16_t)0x2100U)
00240 #define PHY_HALFDUPLEX_100M           ((uint16_t)0x2000U)
00241 #define PHY_FULLDUPLEX_10M            ((uint16_t)0x0100U)
00242 #define PHY_HALFDUPLEX_10M            ((uint16_t)0x0000U)
00243 #define PHY_AUTONEGOTIATION            ((uint16_t)0x1000U)
00244 #define PHY_AUTONEGOTIATION_COMPLETE ((uint16_t)0x0200U)
00245 #define PHY_POWERDOWN                 ((uint16_t)0x0800U)
00246 #define PHY_ISOLATE                   ((uint16_t)0x0400U)
00247 #define PHY_AUTONEGO_COMPLETE          ((uint16_t)0x0020U)
00248 #define PHY_LINKED_STATUS              ((uint16_t)0x0004U)
00249 #define PHY_JABBER_DETECTION           ((uint16_t)0x0002U)
00250
00251 /* Section 4: Extended PHY Registers */
00252 #define PHY_SR                        ((uint16_t)0x10U)
00253 #define PHY_SPEED_STATUS               ((uint16_t)0x0002U)
00254 #define PHY_DUPLEX_STATUS              ((uint16_t)0x0004U)
00255
00256 /* ##### SPI peripheral configuration ##### */
00257
00258 /* CRC FEATURE: Use to activate CRC feature inside HAL SPI Driver
00259 * Activated: CRC code is present inside driver
00260 * Deactivated: CRC code cleaned from driver
00261 */
00262
00263 #define USE_SPI_CRC                    0U
00264
00265 /* Includes -----*/
00266
00267 #ifndef HAL_RCC_MODULE_ENABLED
00268 #include "stm32f4xx_hal_rcc.h"
00269 #endif /* HAL_RCC_MODULE_ENABLED */
00270
00271 #ifndef HAL_GPIO_MODULE_ENABLED
00272 #include "stm32f4xx_hal_gpio.h"
00273 #endif /* HAL_GPIO_MODULE_ENABLED */
00274
00275 #ifndef HAL_EXTI_MODULE_ENABLED
00276 #include "stm32f4xx_hal_exti.h"
00277 #endif /* HAL_EXTI_MODULE_ENABLED */
00278
00279 #ifndef HAL_DMA_MODULE_ENABLED
00280 #include "stm32f4xx_hal_dma.h"
00281 #endif /* HAL_DMA_MODULE_ENABLED */
00282
00283 #ifndef HAL_CORTEX_MODULE_ENABLED
00284 #include "stm32f4xx_hal_cortex.h"
00285 #endif /* HAL_CORTEX_MODULE_ENABLED */
00286
00287 #ifndef HAL_ADC_MODULE_ENABLED
00288 #include "stm32f4xx_hal_adc.h"
00289 #endif /* HAL_ADC_MODULE_ENABLED */
00290
00291 #ifndef HAL_CAN_MODULE_ENABLED
00292 #include "stm32f4xx_hal_can.h"
00293 #endif /* HAL_CAN_MODULE_ENABLED */
00294
00295 #ifndef HAL_CAN_LEGACY_MODULE_ENABLED
00296 #include "stm32f4xx_hal_legacy.h"
00297 #endif /* 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_FMPMBUS_MODULE_ENABLED
00450 #include "stm32f4xx_hal_fmpmbus.h"
00451 #endif /* HAL_FMPMBUS_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
00471 #define assert_param(expr) ((expr) ? (void)0U : assert_failed((uint8_t *)__FILE__, __LINE__))
00472 #else
00473 #define assert_param(expr) ((void)0U)
00474 #endif
00475 /* Exported functions ----- */
00476 void assert_failed(uint8_t* file, uint32_t line);
00477 #else
00478 #define assert_param(expr) ((void)0U)

```

```

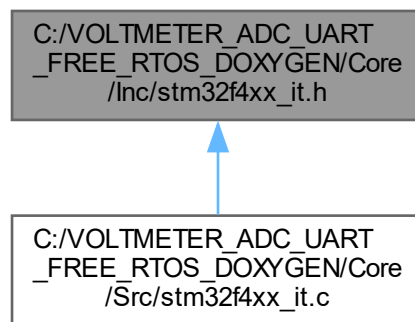
00484 #endif /* USE_FULL_ASSERT */
00485
00486 #ifdef __cplusplus
00487 }
00488 #endif
00489
00490 #endif /* __STM32F4xx_HAL_CONF_H */

```

## 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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.13.2 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 */

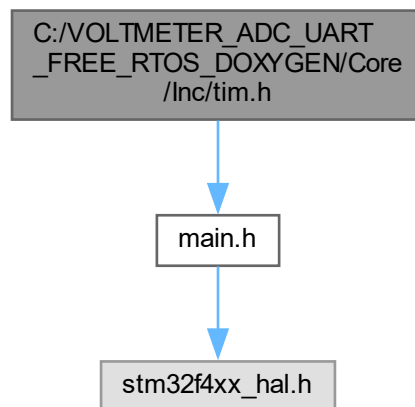
```

## 4.15 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/Inc/tim.h File Reference

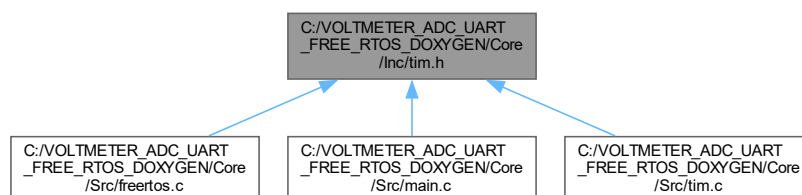
This file contains all the function prototypes for the [tim.c](#) file.

```
#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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.15.2 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

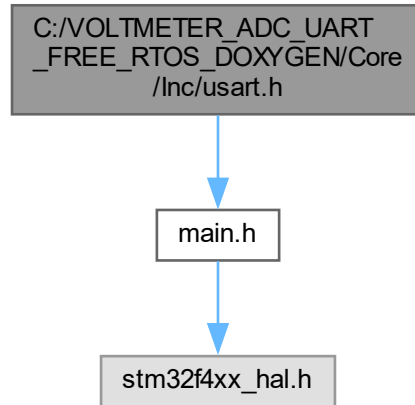
```

## 4.17 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/↵ Inc/usart.h File Reference

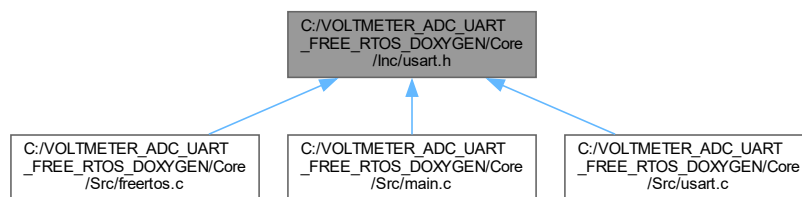
This file contains all the function prototypes for the [usart.c](#) file.

```
#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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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



## 4.17.2 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

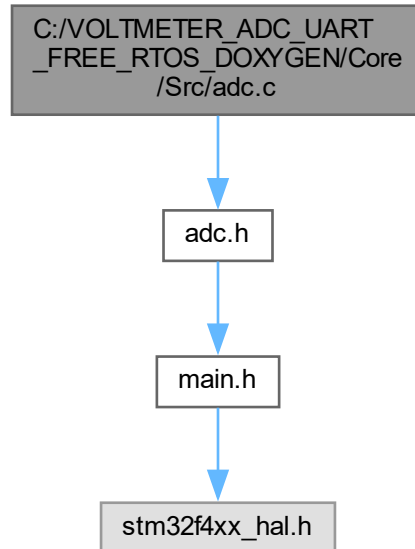
```

## 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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 4.19.2 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

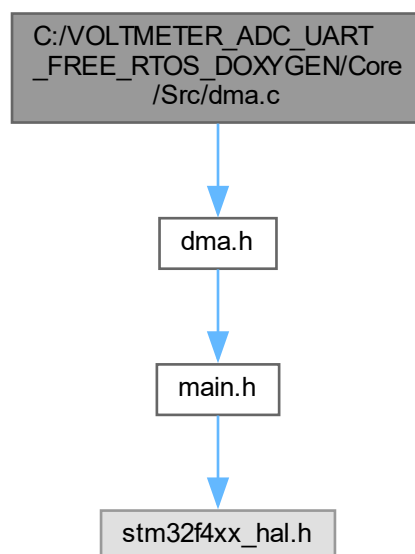
```
ADC_HandleTypeDef hadc1
```

## 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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.20.2 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:

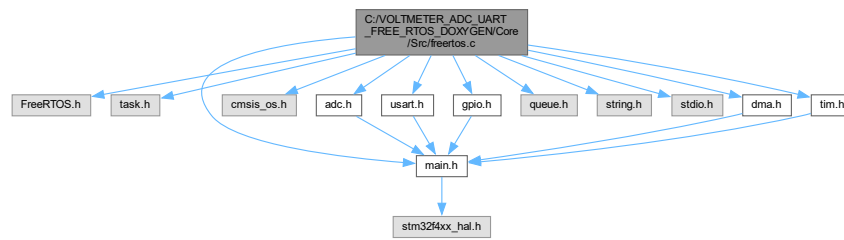


## 4.21 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/↵ Src/freertos.c File Reference

```
#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 \*\*ppxIdleTaskStackBuffer, uint32\_t \*pulIdleTaskStackSize)
- void `vApplicationGetTimerTaskMemory` (StaticTask\_t \*\*ppxTimerTaskTCBBuffer, StackType\_t \*\*ppxTimerTaskStackBuffer, 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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 4.21.2 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.



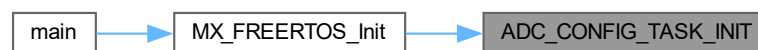
## Parameters

<i>argument</i>	Not used
-----------------	----------

## Return values

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

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.

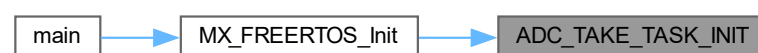
## Parameters

<i>argument</i>	Not used
-----------------	----------

## Return values

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

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.

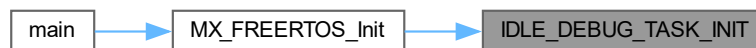
##### Parameters

<i>argument</i>	Not used
-----------------	----------

##### Return values

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

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.

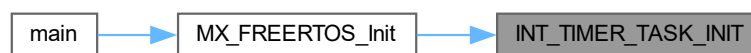
## Parameters

<i>argument</i>	Not used
-----------------	----------

## Return values

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

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.

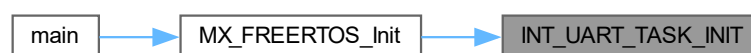
## Parameters

<i>argument</i>	Not used
-----------------	----------

## Return values

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

Here is the caller graph for this function:



#### 4.21.2.8 MX\_FREERTOS\_Init()

```
void MX_FREERTOS_Init (  
    void )
```

FreeRTOS initialization.

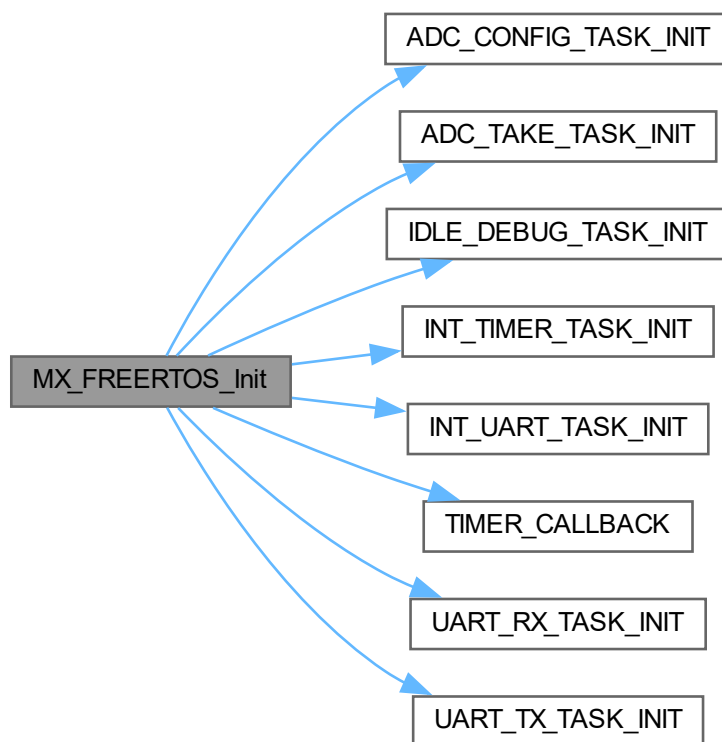
##### Parameters

None	
------	--

##### Return values

None	
------	--

Here is the call graph for this function:



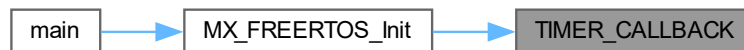
Here is the caller graph for this function:



#### 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.

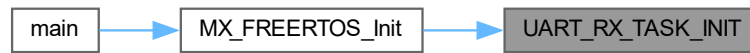
##### Parameters

<i>argument</i>	Not used
-----------------	----------

##### Return values

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

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.

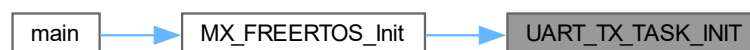
##### Parameters

<i>argument</i>	Not used
-----------------	----------

##### Return values

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

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
```

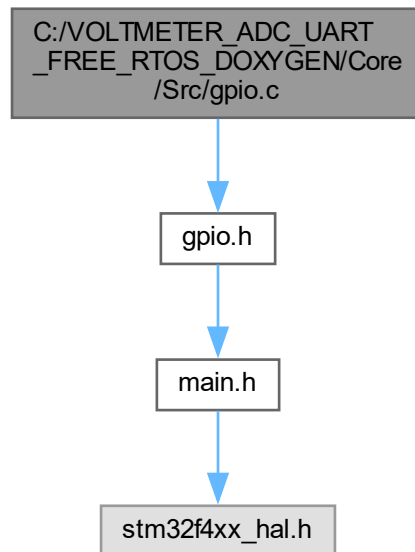
#### 4.21.3.17 UART\_TX\_TASKHandle

```
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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 4.22.2 Function Documentation

### 4.22.2.1 MX\_GPIO\_Init()

```
void MX_GPIO_Init (
    void )
```

Configure pins.

Here is the caller graph for this function:

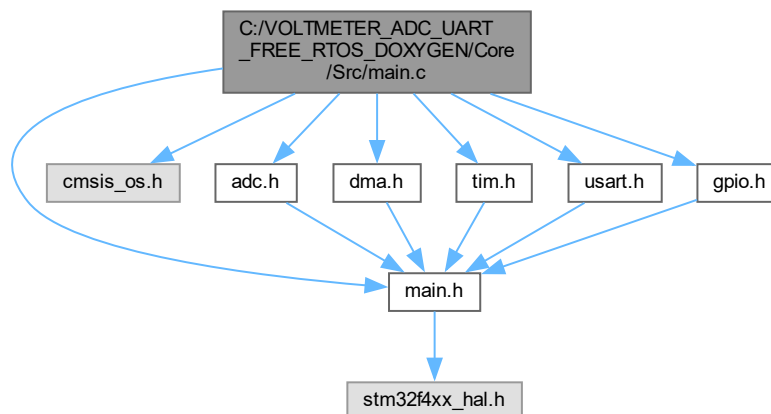


## 4.23 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/Src/main.c File Reference

: 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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.23.2 Function Documentation

#### 4.23.2.1 Error\_Handler()

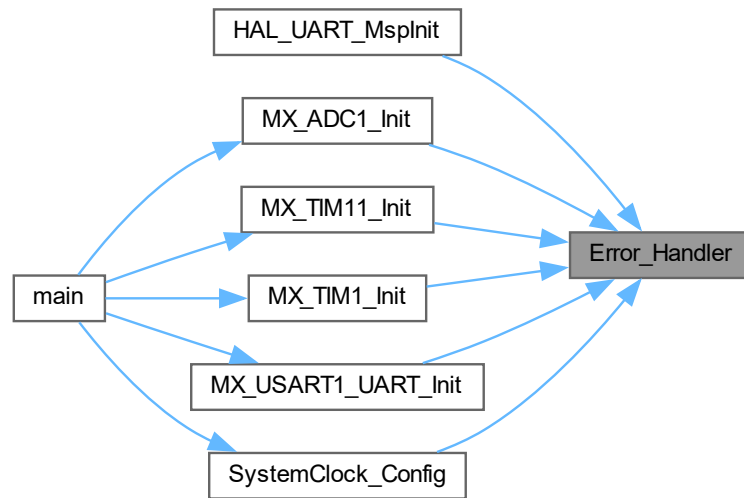
```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

#### Return values

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

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

<i>htim</i>	: TIM handle
-------------	--------------

##### Return values

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

#### 4.23.2.3 main()

```
int main (
```

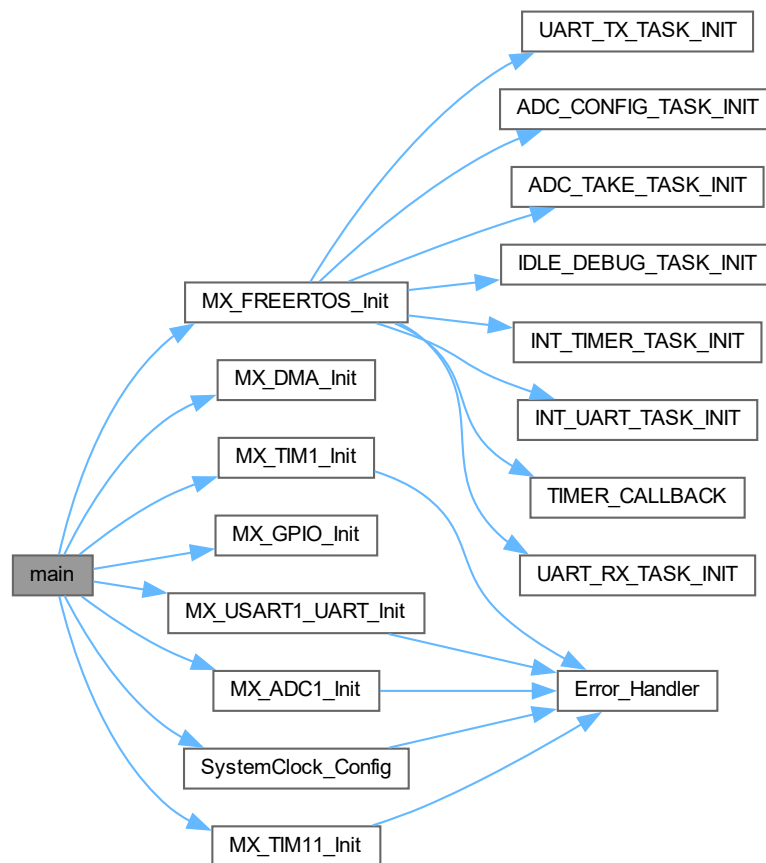
```
void )
```

The application entry point.

Return values

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

Here is the call graph for this function:



#### 4.23.2.4 MX\_FREERTOS\_Init()

```
void MX_FREERTOS_Init (
    void )
```

FreeRTOS initialization.

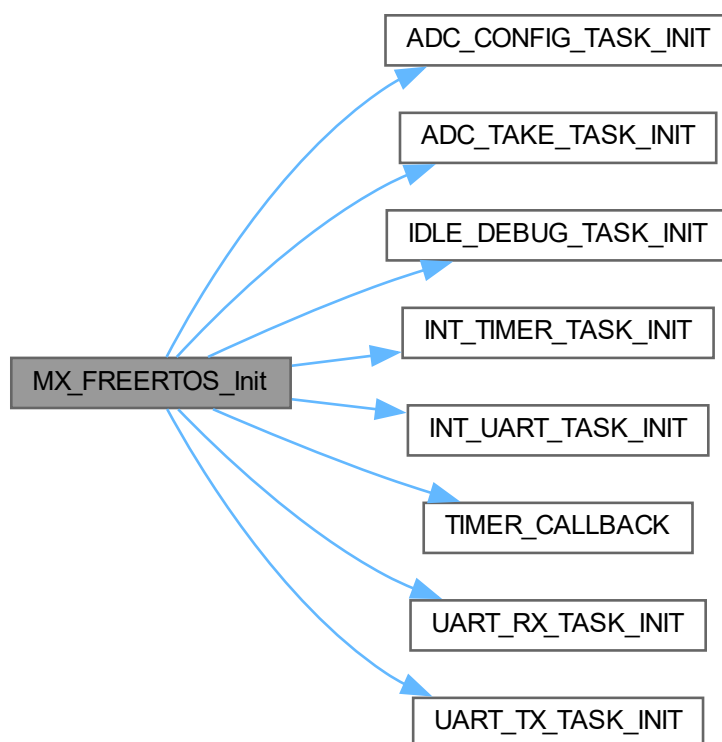
## Parameters

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.

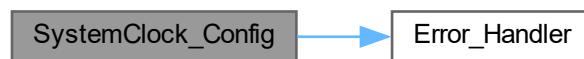
Return values

None	
------	--

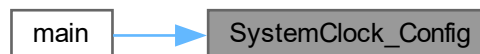
Configure the main internal regulator output voltage

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:



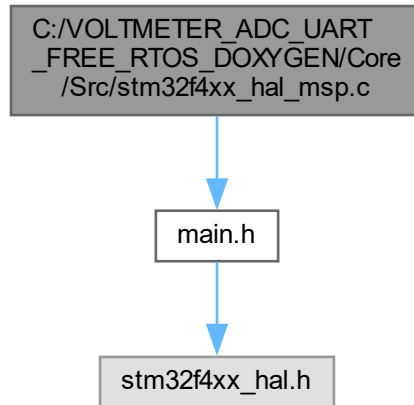
## 4.24 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/↵ Src/stm32f4xx\_hal\_msp.c File Reference

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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.24.2 Function Documentation

#### 4.24.2.1 HAL\_MspInit()

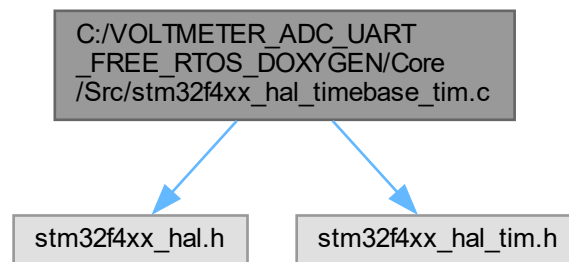
```
void HAL_MspInit (  
    void )
```

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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 4.25.2 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

<i>TickPriority</i>	Tick interrupt priority.
---------------------	--------------------------

#### Return values

<i>HAL</i>	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

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

#### Return values

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

#### 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

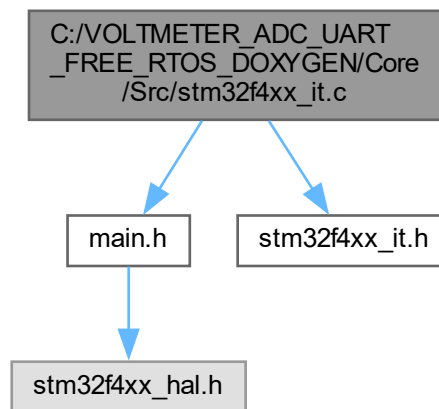
```
TIM_HandleTypeDef htim10
```

## 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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

### 4.26.2 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

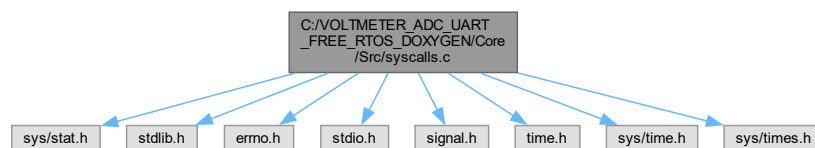
```
UART_HandleTypeDef huart1 [extern]
```

## 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

Copyright (c) 2020-2022 STMicroelectronics. All rights reserved.

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

### 4.27.2 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

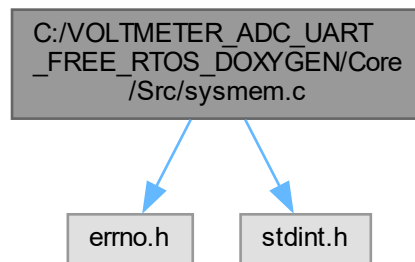
#### 4.27.3.1 `environ`

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

## 4.28 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/Src/systemem.c File Reference

STM32CubeIDE System Memory calls file.

```
#include <errno.h>
#include <stdint.h>
Include dependency graph for systemem.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

Copyright (c) 2022 STMicroelectronics. All rights reserved.

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

## 4.28.2 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

<code>incr</code>	Memory size
-------------------	-------------

#### Returns

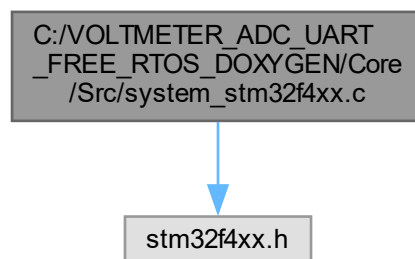
Pointer to allocated memory

## 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

Copyright (c) 2017 STMicroelectronics. All rights reserved.

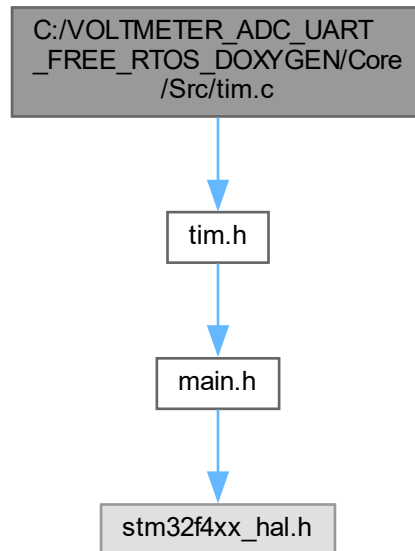
This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

## 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**

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 4.30.2 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

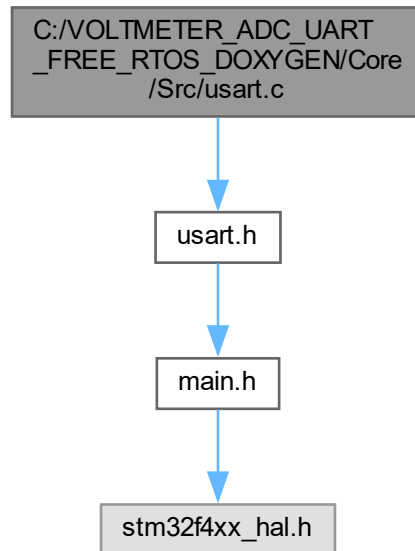
```
TIM_HandleTypeDef htim11
```

## 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

Copyright (c) 2023 STMicroelectronics. All rights reserved.

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

## 4.31.2 Function Documentation

### 4.31.2.1 HAL\_UART\_MspDeInit()

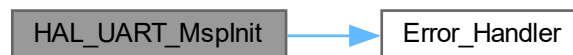
```
void HAL_UART_MspDeInit (
    UART_HandleTypeDef * uartHandle )
```

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\_RX Here is the call graph for this function:



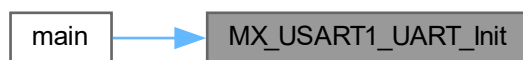
### 4.31.2.3 MX\_USART1\_UART\_Init()

```
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

`__attribute__`  
    syscalls.c, 94  
`__io_getchar`  
    syscalls.c, 95  
`__io_putchar`  
    syscalls.c, 95  
`_close`  
    syscalls.c, 95  
`_execve`  
    syscalls.c, 95  
`_exit`  
    syscalls.c, 96  
`_fork`  
    syscalls.c, 96  
`_fstat`  
    syscalls.c, 96  
`_getpid`  
    syscalls.c, 96  
`_isatty`  
    syscalls.c, 96  
`_kill`  
    syscalls.c, 97  
`_link`  
    syscalls.c, 97  
`_lseek`  
    syscalls.c, 97  
`_open`  
    syscalls.c, 97  
`_sbrk`  
    sysmem.c, 100  
`_stat`  
    syscalls.c, 97  
`_times`  
    syscalls.c, 98  
`_unlink`  
    syscalls.c, 98  
`_wait`  
    syscalls.c, 98

`adc.c`  
    hadc1, 65  
    HAL\_ADC\_MspDeInit, 64  
    HAL\_ADC\_MspInit, 64  
    MX\_ADC1\_Init, 64  
`adc.h`  
    hadc1, 15  
    MX\_ADC1\_Init, 14  
`ADC_CONFIG_TASK_INIT`  
    freertos.c, 68  
`ADC_CONFIG_TASKHandle`

    freertos.c, 75  
`ADC_IRQHandler`  
    stm32f4xx\_it.c, 90  
    stm32f4xx\_it.h, 53  
`ADC_QUEUE_UART_TXHandle`  
    freertos.c, 75  
`ADC_TAKE_TASK_INIT`  
    freertos.c, 69  
`ADC_TAKE_TASKHandle`  
    freertos.c, 75  
`AHBPrescTable`  
    STM32F4xx\_System\_Private\_Variables, 9  
`APBPrescTable`  
    STM32F4xx\_System\_Private\_Variables, 9  
`assert_param`  
    stm32f4xx\_hal\_conf.h, 33  
  
`BusFault_Handler`  
    stm32f4xx\_it.c, 90  
    stm32f4xx\_it.h, 53

`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/adc.h`,  
    13, 15  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/dma.h`,  
    15, 17  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/FreeRTOSConfig.h`,  
    18, 24  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/gpio.h`,  
    26, 28  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/main.h`,  
    28, 30  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/stm32f4xx_it.h`,  
    31, 47  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/stm32f4xx.h`,  
    52, 55  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/tim.h`,  
    56, 59  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Inc/usart.h`,  
    59, 62  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/adc.c`,  
    62  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/dma.c`,  
    65  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/freertos.c`,  
    66  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/gpio.c`,  
    78  
`C:/VOLTMETER_ADC_UART_FREE_RTOS_DOXYGEN/Core/Src/main.c`,  
    79

C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/FreeRTOSConfig.h, 21  
 84  
 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/FreeRTOSConfig.h, 21  
 86  
 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/FreeRTOSConfig.h, 21  
 88  
 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/FreeRTOSConfig.h, 22  
 93  
 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/FreeRTOSConfig.h, 22  
 99  
 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/FreeRTOSConfig.h, 22  
 100  
 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/FreeRTOSConfig.h, 22  
 102  
 C:/VOLTMETER\_ADC\_UART\_FREE\_RTOS\_DOXYGEN/Core/FreeRTOSConfig.h, 22  
 105  
 CMSIS, 5  
 configASSERT  
     FreeRTOSConfig.h, 18  
 configCPU\_CLOCK\_HZ  
     FreeRTOSConfig.h, 19  
 configENABLE\_FPU  
     FreeRTOSConfig.h, 19  
 configENABLE\_MPU  
     FreeRTOSConfig.h, 19  
 configKERNEL\_INTERRUPT\_PRIORITY  
     FreeRTOSConfig.h, 19  
 configLIBRARY\_LOWEST\_INTERRUPT\_PRIORITY  
     FreeRTOSConfig.h, 19  
 configLIBRARY\_MAX\_SYSCALL\_INTERRUPT\_PRIORITY  
     FreeRTOSConfig.h, 19  
 configMAX\_CO\_ROUTINE\_PRIORITIES  
     FreeRTOSConfig.h, 19  
 configMAX\_PRIORITIES  
     FreeRTOSConfig.h, 20  
 configMAX\_SYSCALL\_INTERRUPT\_PRIORITY  
     FreeRTOSConfig.h, 20  
 configMAX\_TASK\_NAME\_LEN  
     FreeRTOSConfig.h, 20  
 configMESSAGE\_BUFFER\_LENGTH\_TYPE  
     FreeRTOSConfig.h, 20  
 configMINIMAL\_STACK\_SIZE  
     FreeRTOSConfig.h, 20  
 configPRIO\_BITS  
     FreeRTOSConfig.h, 20  
 configQUEUE\_REGISTRY\_SIZE  
     FreeRTOSConfig.h, 20  
 configSUPPORT\_DYNAMIC\_ALLOCATION  
     FreeRTOSConfig.h, 21  
 configSUPPORT\_STATIC\_ALLOCATION  
     FreeRTOSConfig.h, 21  
 configTICK\_RATE\_HZ  
     FreeRTOSConfig.h, 21  
 configTIMER\_QUEUE\_LENGTH  
     FreeRTOSConfig.h, 21  
 configTIMER\_TASK\_PRIORITY  
     FreeRTOSConfig.h, 21  
 configTIMER\_TASK\_STACK\_DEPTH  
     FreeRTOSConfig.h, 21  
 configTOTAL\_HEAP\_SIZE  
     FreeRTOSConfig.h, 21  
 configUSE\_16\_BIT\_TICKS  
     FreeRTOSConfig.h, 21  
 configUSE\_CO\_ROUTINES  
     FreeRTOSConfig.h, 22  
 configUSE\_IDLE\_HOOK  
     FreeRTOSConfig.h, 22  
 configUSE\_MUTEXES  
     FreeRTOSConfig.h, 22  
 configUSE\_PORT\_OPTIMISED\_TASK\_SELECTION  
     FreeRTOSConfig.h, 22  
 configUSE\_PREEMPTION  
     FreeRTOSConfig.h, 22  
 configUSE\_TICK\_HOOK  
     FreeRTOSConfig.h, 22  
 configUSE\_TIMERS  
     FreeRTOSConfig.h, 22  
 DATA\_CACHE\_ENABLE  
     stm32f4xx\_hal\_conf.h, 33  
 DebugMon\_Handler  
     stm32f4xx\_it.c, 90  
     stm32f4xx\_it.h, 53  
 dma.c  
     MX\_DMA\_Init, 66  
 dma.h  
     MX\_DMA\_Init, 17  
 DMA2\_Stream2\_IRQHandler  
     stm32f4xx\_it.c, 91  
     stm32f4xx\_it.h, 53  
 DMA2\_Stream7\_IRQHandler  
     stm32f4xx\_it.c, 91  
     stm32f4xx\_it.h, 54  
 DP83848\_PHY\_ADDRESS  
     stm32f4xx\_hal\_conf.h, 34  
 environ  
     syscalls.c, 98  
 Error\_Handler  
     main.c, 80  
     main.h, 29  
 ETH\_RX\_BUF\_SIZE  
     stm32f4xx\_hal\_conf.h, 34  
 ETH\_RXBUFNB  
     stm32f4xx\_hal\_conf.h, 34  
 ETH\_TX\_BUF\_SIZE  
     stm32f4xx\_hal\_conf.h, 34  
 ETH\_TXBUFNB  
     stm32f4xx\_hal\_conf.h, 34  
 EXTERNAL\_CLOCK\_VALUE  
     stm32f4xx\_hal\_conf.h, 34  
 freertos.c  
     ADC\_CONFIG\_TASK\_INIT, 68  
     ADC\_CONFIG\_TASKHandle, 75  
     ADC\_QUEUE\_UART\_TXHandle, 75  
     ADC\_TAKE\_TASK\_INIT, 69

- ADC\_TAKE\_TASKHandle, 75
- HAL\_TIM\_PeriodicElapsedCallback, 69
- HAL\_UARTEx\_RxEventCallback, 70
- IDLE\_DEBUG\_TASK\_INIT, 70
- IDLE\_DEBUG\_TASKHandle, 75
- INT\_TIMER\_TASK\_INIT, 70
- INT\_TIMER\_TASKHandle, 76
- INT\_UART\_QUEUE\_UART\_RXHandle, 76
- INT\_UART\_TASK\_INIT, 71
- INT\_UART\_TASKHandle, 76
- MainBuf, 76
- MainBuf\_SIZE, 68
- MX\_FREERTOS\_Init, 71
- newPos, 76
- oldPos, 76
- RTOS\_TIMER\_1Handle, 76
- RxBuf, 76
- RxBuf\_SIZE, 68
- TIMER\_CALLBACK, 73
- UART\_RX\_QUEUE\_ADC\_CONFIGHandle, 77
- UART\_RX\_QUEUE\_ADC\_TAKHandle, 77
- UART\_RX\_QUEUE\_UART\_TXHandle, 77
- UART\_RX\_TASK\_INIT, 73
- UART\_RX\_TASKHandle, 77
- UART\_TX\_TASK\_INIT, 74
- UART\_TX\_TASKHandle, 77
- vApplicationGetIdleTaskMemory, 74
- vApplicationGetTimerTaskMemory, 74
- vApplicationIdleHook, 75
- vApplicationTickHook, 75
- FreeRTOSConfig.h
  - configASSERT, 18
  - configCPU\_CLOCK\_HZ, 19
  - configENABLE\_FPU, 19
  - configENABLE\_MPU, 19
  - configKERNEL\_INTERRUPT\_PRIORITY, 19
  - configLIBRARY\_LOWEST\_INTERRUPT\_PRIORITY, 19
  - configLIBRARY\_MAX\_SYSCALL\_INTERRUPT\_PRIORITY, 19
  - configMAX\_CO\_ROUTINE\_PRIORITIES, 19
  - configMAX\_PRIORITIES, 20
  - configMAX\_SYSCALL\_INTERRUPT\_PRIORITY, 20
  - configMAX\_TASK\_NAME\_LEN, 20
  - configMESSAGE\_BUFFER\_LENGTH\_TYPE, 20
  - configMINIMAL\_STACK\_SIZE, 20
  - configPRIO\_BITS, 20
  - configQUEUE\_REGISTRY\_SIZE, 20
  - configSUPPORT\_DYNAMIC\_ALLOCATION, 21
  - configSUPPORT\_STATIC\_ALLOCATION, 21
  - configTICK\_RATE\_HZ, 21
  - configTIMER\_QUEUE\_LENGTH, 21
  - configTIMER\_TASK\_PRIORITY, 21
  - configTIMER\_TASK\_STACK\_DEPTH, 21
  - configTOTAL\_HEAP\_SIZE, 21
  - configUSE\_16\_BIT\_TICKS, 21
  - configUSE\_CO\_ROUTINES, 22
  - configUSE\_IDLE\_HOOK, 22
  - configUSE\_MUTEXES, 22
  - configUSE\_PORT\_OPTIMISED\_TASK\_SELECTION, 22
  - configUSE\_PREEMPTION, 22
  - configUSE\_TICK\_HOOK, 22
  - configUSE\_TIMERS, 22
  - INCLUDE\_uxTaskPriorityGet, 22
  - INCLUDE\_vTaskCleanUpResources, 23
  - INCLUDE\_vTaskDelay, 23
  - INCLUDE\_vTaskDelayUntil, 23
  - INCLUDE\_vTaskDelete, 23
  - INCLUDE\_vTaskPrioritySet, 23
  - INCLUDE\_vTaskSuspend, 23
  - INCLUDE\_xTaskGetSchedulerState, 23
  - vPortSVCHandler, 23
  - xPortPendSVHandler, 24
  - xPortSysTickHandler, 24
- gpio.c
  - MX\_GPIO\_Init, 79
- gpio.h
  - MX\_GPIO\_Init, 27
- hadc1
  - adc.c, 65
  - adc.h, 15
  - stm32f4xx\_it.c, 92
- HAL\_ADC\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 34
- HAL\_ADC\_MspDeInit
  - adc.c, 64
- HAL\_ADC\_MspInit
  - adc.c, 64
- HAL\_CORTEX\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 35
- HAL\_DMA\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 35
- HAL\_EXTI\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 35
- HAL\_FLASH\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 35
- HAL\_GPIO\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 35
- HAL\_InitTick
  - stm32f4xx\_hal\_timebase\_tim.c, 87
- HAL\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 35
- HAL\_MspInit
  - stm32f4xx\_hal\_msp.c, 85
- HAL\_PWR\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 35
- HAL\_RCC\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, 36
- HAL\_ResumeTick
  - stm32f4xx\_hal\_timebase\_tim.c, 87
- HAL\_SuspendTick
  - stm32f4xx\_hal\_timebase\_tim.c, 88
- HAL\_TIM\_Base\_MspDeInit

- tim.c, [103](#)
- HAL\_TIM\_Base\_MspInit
  - tim.c, [103](#)
- HAL\_TIM\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [36](#)
- HAL\_TIM\_PeriodElapsedCallback
  - main.c, [81](#)
- HAL\_TIM\_PeriodicElapsedCallback
  - freertos.c, [69](#)
- HAL\_UART\_MODULE\_ENABLED
  - stm32f4xx\_hal\_conf.h, [36](#)
- HAL\_UART\_MspDelInit
  - usart.c, [106](#)
- HAL\_UART\_MspInit
  - usart.c, [106](#)
- HAL\_UARTEx\_RxEventCallback
  - freertos.c, [70](#)
- HardFault\_Handler
  - stm32f4xx\_it.c, [91](#)
  - stm32f4xx\_it.h, [54](#)
- hdma\_usart1\_rx
  - stm32f4xx\_it.c, [92](#)
  - usart.c, [107](#)
- hdma\_usart1\_tx
  - stm32f4xx\_it.c, [92](#)
  - usart.c, [107](#)
- HSE\_STARTUP\_TIMEOUT
  - stm32f4xx\_hal\_conf.h, [36](#)
- HSE\_VALUE
  - stm32f4xx\_hal\_conf.h, [36](#)
  - STM32F4xx\_System\_Private\_Includes, [7](#)
- HSI\_VALUE
  - stm32f4xx\_hal\_conf.h, [36](#)
  - STM32F4xx\_System\_Private\_Includes, [7](#)
- htim1
  - stm32f4xx\_it.c, [93](#)
  - tim.c, [104](#)
  - tim.h, [58](#)
- htim10
  - stm32f4xx\_hal\_timebase\_tim.c, [88](#)
  - stm32f4xx\_it.c, [93](#)
- htim11
  - stm32f4xx\_it.c, [93](#)
  - tim.c, [104](#)
  - tim.h, [58](#)
- huart1
  - stm32f4xx\_it.c, [93](#)
  - usart.c, [107](#)
  - usart.h, [61](#)
- IDLE\_DEBUG\_TASK\_INIT
  - freertos.c, [70](#)
- IDLE\_DEBUG\_TASKHandle
  - freertos.c, [75](#)
- INCLUDE\_uxTaskPriorityGet
  - FreeRTOSConfig.h, [22](#)
- INCLUDE\_vTaskCleanUpResources
  - FreeRTOSConfig.h, [23](#)
- INCLUDE\_vTaskDelay
  - FreeRTOSConfig.h, [23](#)
- INCLUDE\_vTaskDelayUntil
  - FreeRTOSConfig.h, [23](#)
- INCLUDE\_vTaskDelete
  - FreeRTOSConfig.h, [23](#)
- INCLUDE\_vTaskPrioritySet
  - FreeRTOSConfig.h, [23](#)
- INCLUDE\_vTaskSuspend
  - FreeRTOSConfig.h, [23](#)
- INCLUDE\_xTaskGetSchedulerState
  - FreeRTOSConfig.h, [23](#)
- initialise\_monitor\_handles
  - syscalls.c, [98](#)
- INSTRUCTION\_CACHE\_ENABLE
  - stm32f4xx\_hal\_conf.h, [36](#)
- INT\_TIMER\_TASK\_INIT
  - freertos.c, [70](#)
- INT\_TIMER\_TASKHandle
  - freertos.c, [76](#)
- INT\_UART\_QUEUE\_UART\_RXHandle
  - freertos.c, [76](#)
- INT\_UART\_TASK\_INIT
  - freertos.c, [71](#)
- INT\_UART\_TASKHandle
  - freertos.c, [76](#)
- LSE\_STARTUP\_TIMEOUT
  - stm32f4xx\_hal\_conf.h, [37](#)
- LSE\_VALUE
  - stm32f4xx\_hal\_conf.h, [37](#)
- LSI\_VALUE
  - stm32f4xx\_hal\_conf.h, [37](#)
- MAC\_ADDR0
  - stm32f4xx\_hal\_conf.h, [37](#)
- MAC\_ADDR1
  - stm32f4xx\_hal\_conf.h, [37](#)
- MAC\_ADDR2
  - stm32f4xx\_hal\_conf.h, [37](#)
- MAC\_ADDR3
  - stm32f4xx\_hal\_conf.h, [37](#)
- MAC\_ADDR4
  - stm32f4xx\_hal\_conf.h, [38](#)
- MAC\_ADDR5
  - stm32f4xx\_hal\_conf.h, [38](#)
- main
  - main.c, [81](#)
- main.c
  - Error\_Handler, [80](#)
  - HAL\_TIM\_PeriodElapsedCallback, [81](#)
  - main, [81](#)
  - MX\_FREERTOS\_Init, [82](#)
  - SystemClock\_Config, [83](#)
- main.h
  - Error\_Handler, [29](#)
- MainBuf
  - freertos.c, [76](#)
- MainBuf\_SIZE
  - freertos.c, [68](#)

MemManage\_Handler  
     stm32f4xx\_it.c, [91](#)  
     stm32f4xx\_it.h, [54](#)  
 MX\_ADC1\_Init  
     adc.c, [64](#)  
     adc.h, [14](#)  
 MX\_DMA\_Init  
     dma.c, [66](#)  
     dma.h, [17](#)  
 MX\_FREERTOS\_Init  
     freertos.c, [71](#)  
     main.c, [82](#)  
 MX\_GPIO\_Init  
     gpio.c, [79](#)  
     gpio.h, [27](#)  
 MX\_TIM11\_Init  
     tim.c, [103](#)  
     tim.h, [57](#)  
 MX\_TIM1\_Init  
     tim.c, [103](#)  
     tim.h, [57](#)  
 MX\_USART1\_UART\_Init  
     usart.c, [106](#)  
     usart.h, [61](#)  
  
 newPos  
     freertos.c, [76](#)  
 NMI\_Handler  
     stm32f4xx\_it.c, [91](#)  
     stm32f4xx\_it.h, [54](#)  
  
 oldPos  
     freertos.c, [76](#)  
  
 PHY\_AUTONEGO\_COMPLETE  
     stm32f4xx\_hal\_conf.h, [38](#)  
 PHY\_AUTONEGOTIATION  
     stm32f4xx\_hal\_conf.h, [38](#)  
 PHY\_BCR  
     stm32f4xx\_hal\_conf.h, [38](#)  
 PHY\_BSR  
     stm32f4xx\_hal\_conf.h, [38](#)  
 PHY\_CONFIG\_DELAY  
     stm32f4xx\_hal\_conf.h, [38](#)  
 PHY\_DUPLEX\_STATUS  
     stm32f4xx\_hal\_conf.h, [39](#)  
 PHY\_FULLDUPLEX\_100M  
     stm32f4xx\_hal\_conf.h, [39](#)  
 PHY\_FULLDUPLEX\_10M  
     stm32f4xx\_hal\_conf.h, [39](#)  
 PHY\_HALFDUPLEX\_100M  
     stm32f4xx\_hal\_conf.h, [39](#)  
 PHY\_HALFDUPLEX\_10M  
     stm32f4xx\_hal\_conf.h, [39](#)  
 PHY\_ISOLATE  
     stm32f4xx\_hal\_conf.h, [39](#)  
 PHY\_JABBER\_DETECTION  
     stm32f4xx\_hal\_conf.h, [39](#)  
 PHY\_LINKED\_STATUS  
     stm32f4xx\_hal\_conf.h, [40](#)  
 PHY\_LOOPBACK  
     stm32f4xx\_hal\_conf.h, [40](#)  
 PHY\_POWERDOWN  
     stm32f4xx\_hal\_conf.h, [40](#)  
 PHY\_READ\_TO  
     stm32f4xx\_hal\_conf.h, [40](#)  
 PHY\_RESET  
     stm32f4xx\_hal\_conf.h, [40](#)  
 PHY\_RESET\_DELAY  
     stm32f4xx\_hal\_conf.h, [40](#)  
 PHY\_RESTART\_AUTONEGOTIATION  
     stm32f4xx\_hal\_conf.h, [40](#)  
 PHY\_SPEED\_STATUS  
     stm32f4xx\_hal\_conf.h, [41](#)  
 PHY\_SR  
     stm32f4xx\_hal\_conf.h, [41](#)  
 PHY\_WRITE\_TO  
     stm32f4xx\_hal\_conf.h, [41](#)  
 PREFETCH\_ENABLE  
     stm32f4xx\_hal\_conf.h, [41](#)  
  
 RTOS\_TIMER\_1Handle  
     freertos.c, [76](#)  
 RxBuf  
     freertos.c, [76](#)  
 RxBuf\_SIZE  
     freertos.c, [68](#)  
  
 stm32f4xx\_hal\_conf.h  
     assert\_param, [33](#)  
     DATA\_CACHE\_ENABLE, [33](#)  
     DP83848\_PHY\_ADDRESS, [34](#)  
     ETH\_RX\_BUF\_SIZE, [34](#)  
     ETH\_RXBUFNB, [34](#)  
     ETH\_TX\_BUF\_SIZE, [34](#)  
     ETH\_TXBUFNB, [34](#)  
     EXTERNAL\_CLOCK\_VALUE, [34](#)  
     HAL\_ADC\_MODULE\_ENABLED, [34](#)  
     HAL\_CORTEX\_MODULE\_ENABLED, [35](#)  
     HAL\_DMA\_MODULE\_ENABLED, [35](#)  
     HAL\_EXTI\_MODULE\_ENABLED, [35](#)  
     HAL\_FLASH\_MODULE\_ENABLED, [35](#)  
     HAL\_GPIO\_MODULE\_ENABLED, [35](#)  
     HAL\_MODULE\_ENABLED, [35](#)  
     HAL\_PWR\_MODULE\_ENABLED, [35](#)  
     HAL\_RCC\_MODULE\_ENABLED, [36](#)  
     HAL\_TIM\_MODULE\_ENABLED, [36](#)  
     HAL\_UART\_MODULE\_ENABLED, [36](#)  
     HSE\_STARTUP\_TIMEOUT, [36](#)  
     HSE\_VALUE, [36](#)  
     HSI\_VALUE, [36](#)  
     INSTRUCTION\_CACHE\_ENABLE, [36](#)  
     LSE\_STARTUP\_TIMEOUT, [37](#)  
     LSE\_VALUE, [37](#)  
     LSI\_VALUE, [37](#)  
     MAC\_ADDR0, [37](#)  
     MAC\_ADDR1, [37](#)  
     MAC\_ADDR2, [37](#)

- MAC\_ADDR3, [37](#)
- MAC\_ADDR4, [38](#)
- MAC\_ADDR5, [38](#)
- PHY\_AUTONEGO\_COMPLETE, [38](#)
- PHY\_AUTONEGOTIATION, [38](#)
- PHY\_BCR, [38](#)
- PHY\_BSR, [38](#)
- PHY\_CONFIG\_DELAY, [38](#)
- PHY\_DUPLEX\_STATUS, [39](#)
- PHY\_FULLDUPLEX\_100M, [39](#)
- PHY\_FULLDUPLEX\_10M, [39](#)
- PHY\_HALFDUPLEX\_100M, [39](#)
- PHY\_HALFDUPLEX\_10M, [39](#)
- PHY\_ISOLATE, [39](#)
- PHY\_JABBER\_DETECTION, [39](#)
- PHY\_LINKED\_STATUS, [40](#)
- PHY\_LOOPBACK, [40](#)
- PHY\_POWERDOWN, [40](#)
- PHY\_READ\_TO, [40](#)
- PHY\_RESET, [40](#)
- PHY\_RESET\_DELAY, [40](#)
- PHY\_RESTART\_AUTONEGOTIATION, [40](#)
- PHY\_SPEED\_STATUS, [41](#)
- PHY\_SR, [41](#)
- PHY\_WRITE\_TO, [41](#)
- PREFETCH\_ENABLE, [41](#)
- TICK\_INT\_PRIORITY, [41](#)
- USE\_HAL\_ADC\_REGISTER\_CALLBACKS, [41](#)
- USE\_HAL\_CAN\_REGISTER\_CALLBACKS, [41](#)
- USE\_HAL\_CEC\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_CRYPT\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DAC\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DCMI\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_DSI\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_ETH\_REGISTER\_CALLBACKS, [42](#)
- USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_FMPSMBUS\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_HASH\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_HCD\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_I2C\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_I2S\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_IRDA\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS, [43](#)
- USE\_HAL\_LTDC\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_MMC\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_NAND\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_NOR\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_PCD\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_QSPI\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_RNG\_REGISTER\_CALLBACKS, [44](#)
- USE\_HAL\_RTC\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SAI\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SD\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SPI\_REGISTER\_CALLBACKS, [45](#)
- USE\_HAL\_SRAM\_REGISTER\_CALLBACKS, [46](#)
- USE\_HAL\_TIM\_REGISTER\_CALLBACKS, [46](#)
- USE\_HAL\_UART\_REGISTER\_CALLBACKS, [46](#)
- USE\_HAL\_USART\_REGISTER\_CALLBACKS, [46](#)
- USE\_HAL\_WWDG\_REGISTER\_CALLBACKS, [46](#)
- USE\_RTOS, [46](#)
- USE\_SPI\_CRC, [46](#)
- VDD\_VALUE, [46](#)
- stm32f4xx\_hal\_msp.c
  - HAL\_MspInit, [85](#)
- stm32f4xx\_hal\_timebase\_tim.c
  - HAL\_InitTick, [87](#)
  - HAL\_ResumeTick, [87](#)
  - HAL\_SuspendTick, [88](#)
  - htim10, [88](#)
- stm32f4xx\_it.c
  - ADC\_IRQHandler, [90](#)
  - BusFault\_Handler, [90](#)
  - DebugMon\_Handler, [90](#)
  - DMA2\_Stream2\_IRQHandler, [91](#)
  - DMA2\_Stream7\_IRQHandler, [91](#)
  - hadc1, [92](#)
  - HardFault\_Handler, [91](#)
  - hdma\_usart1\_rx, [92](#)
  - hdma\_usart1\_tx, [92](#)
  - htim1, [93](#)
  - htim10, [93](#)
  - htim11, [93](#)
  - huart1, [93](#)
  - MemManage\_Handler, [91](#)
  - NMI\_Handler, [91](#)
  - TIM1\_TRG\_COM\_TIM11\_IRQHandler, [91](#)
  - TIM1\_UP\_TIM10\_IRQHandler, [92](#)
  - UsageFault\_Handler, [92](#)
  - USART1\_IRQHandler, [92](#)
- stm32f4xx\_it.h
  - ADC\_IRQHandler, [53](#)
  - BusFault\_Handler, [53](#)
  - DebugMon\_Handler, [53](#)
  - DMA2\_Stream2\_IRQHandler, [53](#)
  - DMA2\_Stream7\_IRQHandler, [54](#)
  - HardFault\_Handler, [54](#)
  - MemManage\_Handler, [54](#)
  - NMI\_Handler, [54](#)
  - TIM1\_TRG\_COM\_TIM11\_IRQHandler, [54](#)
  - TIM1\_UP\_TIM10\_IRQHandler, [54](#)
  - UsageFault\_Handler, [55](#)
  - USART1\_IRQHandler, [55](#)
- Stm32f4xx\_system, [6](#)
- STM32F4xx\_System\_Private\_Defines, [8](#)
- STM32F4xx\_System\_Private\_FunctionPrototypes, [10](#)

- STM32F4xx\_System\_Private\_Functions, 10
  - SystemCoreClockUpdate, 10
  - SystemInit, 11
- STM32F4xx\_System\_Private\_Includes, 7
  - HSE\_VALUE, 7
  - HSI\_VALUE, 7
- STM32F4xx\_System\_Private\_Macros, 8
- STM32F4xx\_System\_Private\_TypesDefinitions, 8
- STM32F4xx\_System\_Private\_Variables, 9
  - AHBPrescTable, 9
  - APBPrescTable, 9
  - SystemCoreClock, 9
- syscalls.c
  - \_\_attribute\_\_, 94
  - \_\_io\_getchar, 95
  - \_\_io\_putchar, 95
  - \_close, 95
  - \_execve, 95
  - \_exit, 96
  - \_fork, 96
  - \_fstat, 96
  - \_getpid, 96
  - \_isatty, 96
  - \_kill, 97
  - \_link, 97
  - \_lseek, 97
  - \_open, 97
  - \_stat, 97
  - \_times, 98
  - \_unlink, 98
  - \_wait, 98
  - environ, 98
  - initialise\_monitor\_handles, 98
- sysmem.c
  - \_sbrk, 100
- SystemClock\_Config
  - main.c, 83
- SystemCoreClock
  - STM32F4xx\_System\_Private\_Variables, 9
- SystemCoreClockUpdate
  - STM32F4xx\_System\_Private\_Functions, 10
- SystemInit
  - STM32F4xx\_System\_Private\_Functions, 11
- TICK\_INT\_PRIORITY
  - stm32f4xx\_hal\_conf.h, 41
- tim.c
  - HAL\_TIM\_Base\_MspDeInit, 103
  - HAL\_TIM\_Base\_MspInit, 103
  - htim1, 104
  - htim11, 104
  - MX\_TIM11\_Init, 103
  - MX\_TIM1\_Init, 103
- tim.h
  - htim1, 58
  - htim11, 58
  - MX\_TIM11\_Init, 57
  - MX\_TIM1\_Init, 57
- TIM1\_TRG\_COM\_TIM11\_IRQHandler
  - stm32f4xx\_it.c, 91
  - stm32f4xx\_it.h, 54
- TIM1\_UP\_TIM10\_IRQHandler
  - stm32f4xx\_it.c, 92
  - stm32f4xx\_it.h, 54
- TIMER\_CALLBACK
  - freertos.c, 73
- UART\_RX\_QUEUE\_ADC\_CONFIGHandle
  - freertos.c, 77
- UART\_RX\_QUEUE\_ADC\_TAKHandle
  - freertos.c, 77
- UART\_RX\_QUEUE\_UART\_TXHandle
  - freertos.c, 77
- UART\_RX\_TASK\_INIT
  - freertos.c, 73
- UART\_RX\_TASKHandle
  - freertos.c, 77
- UART\_TX\_TASK\_INIT
  - freertos.c, 74
- UART\_TX\_TASKHandle
  - freertos.c, 77
- UsageFault\_Handler
  - stm32f4xx\_it.c, 92
  - stm32f4xx\_it.h, 55
- usart.c
  - HAL\_UART\_MspDeInit, 106
  - HAL\_UART\_MspInit, 106
  - hdma\_usart1\_rx, 107
  - hdma\_usart1\_tx, 107
  - huart1, 107
  - MX\_USART1\_UART\_Init, 106
- usart.h
  - huart1, 61
  - MX\_USART1\_UART\_Init, 61
- USART1\_IRQHandler
  - stm32f4xx\_it.c, 92
  - stm32f4xx\_it.h, 55
- USE\_HAL\_ADC\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 41
- USE\_HAL\_CAN\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 41
- USE\_HAL\_CEC\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 42
- USE\_HAL\_Cryp\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 42
- USE\_HAL\_DAC\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 42
- USE\_HAL\_DCMI\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 42
- USE\_HAL\_DFSDM\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 42
- USE\_HAL\_DMA2D\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 42
- USE\_HAL\_DSI\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 42
- USE\_HAL\_ETH\_REGISTER\_CALLBACKS
  - stm32f4xx\_hal\_conf.h, 42
- USE\_HAL\_FMPI2C\_REGISTER\_CALLBACKS



stm32f4xx\_hal\_conf.h, [43](#)  
 USE\_HAL\_FMPMBUS\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [43](#)  
 USE\_HAL\_HASH\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [43](#)  
 USE\_HAL\_HCD\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [43](#)  
 USE\_HAL\_I2C\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [43](#)  
 USE\_HAL\_I2S\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [43](#)  
 USE\_HAL\_IRDA\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [43](#)  
 USE\_HAL\_LPTIM\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [43](#)  
 USE\_HAL\_LTDC\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [44](#)  
 USE\_HAL\_MMC\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [44](#)  
 USE\_HAL\_NAND\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [44](#)  
 USE\_HAL\_NOR\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [44](#)  
 USE\_HAL\_PCCARD\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [44](#)  
 USE\_HAL\_PCD\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [44](#)  
 USE\_HAL\_QSPI\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [44](#)  
 USE\_HAL\_RNG\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [44](#)  
 USE\_HAL\_RTC\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [45](#)  
 USE\_HAL\_SAI\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [45](#)  
 USE\_HAL\_SD\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [45](#)  
 USE\_HAL\_SDRAM\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [45](#)  
 USE\_HAL\_SMARTCARD\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [45](#)  
 USE\_HAL\_SMBUS\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [45](#)  
 USE\_HAL\_SPDIFRX\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [45](#)  
 USE\_HAL\_SPI\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [45](#)  
 USE\_HAL\_SRAM\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [46](#)  
 USE\_HAL\_TIM\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [46](#)  
 USE\_HAL\_UART\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [46](#)  
 USE\_HAL\_USART\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [46](#)  
 USE\_HAL\_WWDG\_REGISTER\_CALLBACKS  
   stm32f4xx\_hal\_conf.h, [46](#)  
 USE\_RTOS  
   stm32f4xx\_hal\_conf.h, [46](#)  
 USE\_SPI\_CRC  
   stm32f4xx\_hal\_conf.h, [46](#)  
 vApplicationGetIdleTaskMemory  
   freertos.c, [74](#)  
 vApplicationGetTimerTaskMemory  
   freertos.c, [74](#)  
 vApplicationIdleHook  
   freertos.c, [75](#)  
 vApplicationTickHook  
   freertos.c, [75](#)  
 VDD\_VALUE  
   stm32f4xx\_hal\_conf.h, [46](#)  
 vPortSVCHandler  
   FreeRTOSConfig.h, [23](#)  
 xPortPendSVHandler  
   FreeRTOSConfig.h, [24](#)  
 xPortSysTickHandler  
   FreeRTOSConfig.h, [24](#)